i.MX 6Solo/6DualLite Applications Processor Reference Manual

Document Number: IMX6SDLRM
Rev. 5, 05/2020



Contents

Section number **Title** **Page**

Chapter 1
Introduction

1.1 About This Document....193

1.1.1 Audience....193

1.1.2 Organization....194

1.1.3 Suggested Reading....194

1.1.3.1 General Information....194

1.1.3.2 Related Documentation....194

1.1.4 Conventions....194

1.1.5 Register Access....196

1.1.5.1 Register Diagram Field Access Type Legend....196

1.1.5.2 Register Macro Usage....197

1.1.6 Signal Conventions....198

1.1.7 Acronyms and Abbreviations....198

1.2 Introduction....201

1.3 Target Applications....201

1.4 Features....202

1.5 Architectural Overview....206

1.5.1 Block Diagram....206

1.5.1.1 Simplified Block Diagram....206

1.5.2 Architectural Partitioning....207

1.5.3 Endianness Support....209

1.5.4 Memory Interfaces....209

Chapter 2
Memory Maps

2.1 Memory system overview....211

2.2 Arm Platform Memory Map....211

2.3 DDR mapping to MMDC controller ports....216

2.4 DMA memory map....217

Chapter 3
Interrupts and DMA Events

3.1 Overview....219
3.2 Cortex A9 interrupts....219
3.3 SDMA event mapping....223

Chapter 4
External Signals and Pin Multiplexing

4.1 Overview....225
4.1.1 Pin Assignments....225
4.1.2 Muxing Options....287

Chapter 5
Fusemap

5.1 Boot Fusemap....329
5.2 Fusemap Description Table....338

Chapter 6
External Memory Controllers

6.1 Overview....347
6.2 Multi-mode DDR controller (MMDC) overview and feature summary....347
6.3 Raw NAND flash controller overview....349
6.3.1 NAND interface features....350
6.3.2 NAND control features....351
6.3.3 Internal interface features....351
6.3.4 APBH-DMA overview....351
6.3.5 ECC-BCH features....351
6.4 EIM-PSRAM/NOR flash controller overview....352
6.4.1 EIM features....352
6.4.2 EIM boot scenarios....353
6.4.3 EIM boot configuration....353
6.4.4 OneNAND requirements....354

Chapter 7
System Debug

7.1 Overview....355

7.2 Chip and Arm Platform Debug Architecture....355

7.2.1 Debug Features....356

7.2.2 Debug system components....357

7.2.2.1 AMBA Trace Bus (ATB)....358

7.2.2.2 ATB replicator....358

7.2.2.3 Embedded Cross Triggering....358

7.2.2.3.1 Cross-Trigger Matrix (CTM)....359

7.2.2.3.2 Cross-Trigger Interface (CTI)....360

7.2.2.4 Debug Access Port (DAP)....360

7.2.2.5 Coresight Embedded Trace Buffer (ETB)....361

7.2.2.5.1 Performance Profiling with the ETB....362

7.2.2.5.2 ETB Counter Control....362

7.2.3 Chip-Specific SJC Features....363

7.2.3.1 JTAG Disable Mode....363

7.2.3.2 JTAG ID....363

7.2.4 System JTAG Controller - SJC....364

7.2.5 System JTAG controller main features....364

7.2.6 TAP Port....364

7.2.7 SJC main blocks....365

7.3 Smart DMA (SDMA) core....365

7.3.1 SDMA On Chip Emulation Module (OnCE) Feature Summary....366

7.3.1.1 Other SDMA Debug Functionality....367

7.3.1.2 SDMA ROM Patching....367

7.4 Miscellaneous....367

7.4.1 Clock/Reset/Power....367

7.5 Supported tools....368

Chapter 8
System Boot

8.1 Overview....369

8.2 Boot modes....371

8.2.1 Boot mode pin settings....371

8.2.2 High-level boot sequence....372

8.2.3 Boot From Fuses mode (BOOT_MODE[1:0] = 00b)....373

8.2.4 Serial Downloader....374

8.2.5 Internal Boot mode (BOOT_MODE[1:0] = 0b10)....375

8.2.6 Boot security settings....376

8.3 Device configuration....376

8.3.1 Boot eFUSE descriptions....377

8.3.2 GPIO boot overrides....379

8.3.3 Device Configuration Data (DCD)....380

8.4 Device initialization....380

8.4.1 Internal ROM/RAM memory map....380

8.4.2 Boot block activation....381

8.4.3 Clocks at boot time....382

8.4.4 Enabling MMU and caches....384

8.4.5 Exception handling....384

8.4.6 Interrupt handling during boot....385

8.4.7 Persistent bits....385

8.5 Boot devices (internal boot)....386

8.5.1 NOR flash/OneNAND using EIM interface....387

8.5.1.1 NOR flash boot operation....387

8.5.1.2 OneNAND flash boot operation....387

8.5.1.3 IOMUX configuration for EIM devices....388

8.5.2 NAND flash....389

8.5.2.1 NAND eFUSE configuration....390

8.5.2.2 NAND flash boot flow and Boot Control Blocks (BCB)..... 391
8.5.2.3 Firmware configuration block..... 395
8.5.2.4 Discovered Bad Block Table (DBBT)..... 398
8.5.2.5 Bad block handling in ROM..... 398
8.5.2.6 Toggle mode DDR NAND boot..... 399
8.5.2.6.1 GPMI and BCH clocks configuration..... 399
8.5.2.6.2 Setup DMA for DDR transfers..... 400
8.5.2.6.3 Reconfigure timing and speed using values in FCB..... 400
8.5.2.7 Typical NAND page organization..... 401
8.5.2.7.1 BCH ECC page organization..... 401
8.5.2.7.2 Metadata..... 402
8.5.2.8 IOMUX configuration for NAND..... 402
8.5.3 Expansion device..... 403
8.5.3.1 Expansion device eFUSE configuration..... 403
8.5.3.2 MMC and eMMC boot..... 406
8.5.3.3 SD, eSD, and SDXC..... 414
8.5.3.4 IOMUX configuration for SD/MMC..... 414
8.5.3.5 Redundant boot support for expansion device..... 415
8.5.4 Serial ROM through SPI and I2C..... 417
8.5.4.1 Serial ROM eFUSE configuration..... 417
8.5.4.2 I2C boot..... 418
8.5.4.2.1 I2C IOMUX pin configuration..... 419
8.5.4.3 ECSPI boot..... 419
8.5.4.3.1 ECSPI IOMUX pin configuration..... 421
8.6 Program image..... 422
8.6.1 Image Vector Table and Boot Data..... 422
8.6.1.1 Image vector table structure..... 423
8.6.1.2 Boot data structure..... 424
8.6.2 Device Configuration Data (DCD)..... 424

8.6.2.1 Write data command.... 426
8.6.2.2 Check data command.... 427
8.6.2.3 NOP command.... 429
8.6.2.4 Unlock command.... 429
8.7 Plugin image.... 430
8.8 Serial Downloader.... 430
8.8.1 USB.... 431
8.8.1.1 USB configuration details.... 432
8.8.1.2 IOMUX configuration for USB.... 433
8.8.2 Serial Download Protocol (SDP).... 433
8.8.2.1 SDP commands.... 434
8.8.2.1.1 READ_REGISTER.... 434
8.8.2.1.2 WRITE_REGISTER.... 435
8.8.2.1.3 WRITE_FILE.... 435
8.8.2.1.4 ERROR_STATUS.... 437
8.8.2.1.5 DCD_WRITE.... 437
8.8.2.1.6 JUMP_ADDRESS.... 438
8.9 Recovery devices.... 439
8.10 USB low-power boot.... 439
8.11 SD/MMC manufacture mode.... 441
8.11.1 Using manufacture mode / serial download mode with eMMC.... 442
8.12 High-Assurance Boot (HAB).... 443
8.12.1 HAB API vector table addresses.... 444

Chapter 9
Multimedia

9.1 Video Graphics Sub System.... 445
9.1.1 Display outputs.... 446
9.1.2 Video input.... 448
9.1.3 Synchronization Mechanisms.... 449

9.1.4 Supported applications....449
9.2 Image Processing Unit (IPU)....452
9.2.1 IPU External Ports....454
9.2.1.1 Camera Ports....455
9.2.1.2 Display Ports....456
9.2.1.2.1 Access Modes....456
9.2.1.2.2 Synchronous Access....456
9.2.1.2.3 Asynchronous Access....457
9.2.1.2.4 Interface details....457
9.2.1.2.5 Connecting To Display Devices....458
9.2.2 Processing....458
9.2.2.1 Display Processor (DP)....459
9.2.2.2 Video de-interlacer (VDIC)....460
9.2.2.3 Image Converter (IC)....460
9.2.2.4 Image Rotator (IRT)....461
9.2.3 Automatic Procedures....461
9.3 LVDS Display Bridge (LDB)....463
9.3.1 LDB Overview....463
9.3.2 LDB External Ports....464
9.3.3 Input Parallel Display Ports....464
9.3.3.1 Output LVDS Ports....464
9.3.3.2 Control signals....465
9.4 Video Data Order Adapter (VDOA)....465
9.5 Display Content Integrity Checker (DCIC)....465
9.6 Video Processing Unit (VPUv6)....465
9.6.1 Basic Structure....467
9.6.2 Feature summary....468
9.7 OpenGL ES 3D Graphics Processing Unit (GPU3Dv5)....468
9.7.1 OpenGL Overview....468

9.7.2 OpenGL Features ... 468
9.7.3 OpenGL Block Diagram ... 469
9.7.4 OpenGL Performance ... 469
9.7.5 OpenGL Software ... 470
9.8 2D Graphics Processing Unit (GPU2Dv2) ... 470
9.8.1 2D feature summary ... 470
9.8.2 2D Block Diagram ... 471
9.8.3 2D Performance ... 472
9.8.4 2D Software ... 472
9.9 HDMI TX - HD Multimedia Interface transmitter ... 472
9.9.1 HDMI Introduction ... 473
9.9.2 Features ... 474
9.10 Display / Sensor MIPI interfaces ... 475
9.10.1 Introduction ... 475
9.10.2 MIPI DSI ... 476
9.10.3 MIPI CSI-2 ... 477
9.10.4 D - PHY ... 478
9.11 Audio subsystem ... 479
9.11.1 Audio Subsystem Module Overview ... 480
9.11.2 Synchronous Serial Interface (SSI) ... 482
9.11.3 Digital Audio MUX (AUDMUX) ... 483
9.11.4 Enhanced Serial Audio Interface (ESAI) ... 485
9.11.5 Sony/Philips Digital Interface (SPDIF) ... 485
9.11.6 Asynchronous Sample Rate Converter (ASRC) ... 487

Chapter 10
Clock and Power Management

10.1 Introduction ... 491
10.2 Device Power Management Architecture Components ... 491
10.2.1 Centralized components of clock generation and management ... 492

10.2.2 Centralized components of power generation, distribution and management ... 493
10.2.3 Reset generation and distribution system ... 493
10.2.4 Power and clock management framework ... 493
10.3 Clock Management ... 494
10.3.1 Centralized components of clock management system ... 494
10.3.2 Clock generation ... 497
10.3.2.1 Crystal Oscillator (XTALOSC) ... 497
10.3.2.2 Low Voltage Differential Signaling (LVDS) I/O ports ... 497
10.3.2.3 PLLs ... 497
10.3.2.3.1 General PLL Control and Status Functions ... 499
10.3.2.4 CCM ... 500
10.3.2.5 Low Power Clock Gating unit (LPCG) ... 500
10.3.3 Peripheral components of clock management system ... 501
10.3.3.1 Interface and functional clock ... 501
10.3.3.2 Block level clock management ... 502
10.3.3.2.1 Master clock protocol ... 503
10.3.3.2.2 Slave clock protocol ... 503
10.3.3.3 Clock Domain(s) ... 504
10.3.3.4 Domain level clock management ... 504
10.3.3.5 Domain dependencies ... 504
10.4 Power management ... 504
10.4.1 Centralized Components of Power Management System ... 505
10.4.1.1 Integrated PMU ... 505
10.4.1.1.1 Digital LDO Regulators ... 507
10.4.1.1.2 Analog LDO regulators ... 508
10.4.1.1.3 USB LDO ... 508
10.4.1.1.4 SNVS regulator ... 508
10.4.1.1.5 Reverse well biasing ... 509
10.4.1.2 GPC - General Power Controller ... 509

10.4.1.3 SRC - System reset Controller 509
10.4.1.4 Power domain(s) 510
10.4.1.4.1 Power distribution 511
10.4.1.4.2 Domain Memory and domain logic state retention in case of Power Gating. 512
10.4.1.4.3 Power Gating Domain Management 512
10.4.1.4.3.1 Arm Core Platform 512
10.4.1.4.3.2 GPU3D, GPU2D and VPU 513
10.4.1.4.3.3 SoC 514
10.4.1.4.4 Power Gating domain dependencies 514
10.4.1.5 Voltage domains 515
10.4.1.6 Voltage domain management 515
10.4.1.6.1 Dynamic 515
10.4.1.6.1.1 Voltage Scaling 515
10.4.1.6.2 Static 515
10.4.1.6.2.1 Standby Leakage reduction (SLR) 515
10.4.1.6.2.2 ANALOG PHYs IPs 516
10.4.1.7 System domains layout 516
10.4.2 Power management techniques 518
10.4.2.1 Power saving techniques 519
10.4.2.2 Thermal-aware power management 520
10.4.2.3 Peripheral Power management 520
10.4.2.3.1 Main memory power management 520
10.4.2.3.2 IO power reduction 521
10.4.2.4 Examples of External Power Supply Interface 522
10.5 ONOFF (Button) 525

Chapter 11
System Security

11.1 Overview 527
11.2 Central Security Unit (CSU) 528

11.2.1 CSU Overview 528
11.2.2 CSU Features 528
11.2.3 CSU Functional Description 529
11.2.3.1 CSU Peripheral Access Policy 529
11.3 Cryptographic Acceleration and Assurance Module (CAAM) 530
11.3.1 CAAM Overview 530
11.4 Secure Non-Volatile Storage (SNVS) 531
11.4.1 SNVS Overview 531
11.4.2 Tamper Detection 531
11.5 Data Co-Processor (DCP) 532
11.6 High-Assurance Boot (HAB) 532
11.7 System JTAG Controller (SJC) 533

Chapter 12
ARM Cortex A9 MPCore Platform (ARM)

12.1 Overview 535
12.2 External Signals 535
12.3 Platform configuration 536
12.3.1 Platform and SCU configuration 537
12.3.2 Core configuration 537
12.3.3 PL310 L2 Cache configuration 537
12.3.4 Endian Modes 538
12.3.5 Memory Parity error support 538
12.4 Performance and Power 538
12.4.1 Low-Power design 538
12.4.1.1 SRPG (State Retention Power Gating) 539
12.4.1.2 Dynamic Voltage and Frequency Scaling (DVFS) 539
12.4.2 Clocks, frequency goals 539
12.4.2.1 ARM Clock 540
12.4.2.2 Bus Clocks 540

12.4.2.3 Debug Clocks....540
12.5 Core Platform Sub-Blocks details....540
12.5.1 ARM Cortex A9 MPCore Processor....540
12.5.2 Media Processing Engine (MPE - NEON)....541
12.5.3 Generic Interrupt Controller (GIC)....541
12.5.3.1 Interrupt Controller Features....541
12.5.3.2 About the Interrupt Controller....542
12.5.3.3 Interrupt Controller Clock frequency....542
12.5.3.4 TrustZone support....542
12.5.4 Instruction and data caches (L1)....542
12.5.4.1 L1 features....543
12.5.5 L2 Cache and controller (PL310)....543
12.6 Debug and Trace Sub-blocks (CoreSight components)....543
12.6.1 Debug Access Port (DAP)....543
12.6.2 Program Trace Macrocell (PTM)....544
12.6.2.1 Program Flow Trace (PFT)....544
12.6.3 Cross Trigger Interface (CTI)....545
12.6.4 Embedded Trace Buffer (ETB)....546
12.6.4.1 AMBA Trace Bus (ATB) Replicator....546

Chapter 13
AHB to IP Bridge (AIPSTZ)

13.1 Overview....547
13.1.1 Features....547
13.2 Clocks....547
13.3 Functional Description....548
13.4 Access Protections....549
13.5 Access Support....549
13.6 Initialization Information....550
13.6.1 Security Block....550

13.7 Off-Platform Peripherals Index....551
13.8 AIPSTZ Memory Map/Register Definition....554
13.8.1 Master Priviledge Registers (AIPSTZ*x*_MPR)....554
13.8.2 Off-Platform Peripheral Access Control Registers (AIPSTZ*x*_OPACR)....557
13.8.3 Off-Platform Peripheral Access Control Registers (AIPSTZ*x*_OPACR1)....560
13.8.4 Off-Platform Peripheral Access Control Registers (AIPSTZ*x*_OPACR2)....563
13.8.5 Off-Platform Peripheral Access Control Registers (AIPSTZ*x*_OPACR3)....566
13.8.6 Off-Platform Peripheral Access Control Registers (AIPSTZ*x*_OPACR4)....569

Chapter 14
AHB-to-APBH Bridge with DMA (APBH-Bridge-DMA)

14.1 Overview....571
14.2 Clocks....572
14.3 APBH DMA....573
14.4 NAND Read Status Polling Example....578
14.5 APBH Memory Map/Register Definition....580
14.5.1 AHB to APBH Bridge Control and Status Register 0 (APBH_CTRL0*n*)....586
14.5.2 AHB to APBH Bridge Control and Status Register 1 (APBH_CTRL1*n*)....588
14.5.3 AHB to APBH Bridge Control and Status Register 2 (APBH_CTRL2*n*)....591
14.5.4 AHB to APBH Bridge Channel Register (APBH_CHANNEL_CTRL*n*)....596
14.5.5 AHB to APBH DMA Device Assignment Register (APBH_DEVSEL)....597
14.5.6 AHB to APBH DMA burst size (APBH_DMA_BURST_SIZE)....598
14.5.7 AHB to APBH DMA Debug Register (APBH_DEBUG)....599
14.5.8 APBH DMA Channel n Current Command Address Register (APBH_CH*n*_CURCMDAR)....600
14.5.9 APBH DMA Channel n Next Command Address Register (APBH_CH*n*_NXTCMDAR)....601
14.5.10 APBH DMA Channel n Command Register (APBH_CH*n*_CMD)....601
14.5.11 APBH DMA Channel n Buffer Address Register (APBH_CH*n*_BAR)....603
14.5.12 APBH DMA Channel n Semaphore Register (APBH_CH*n*_SEMA)....604
14.5.13 AHB to APBH DMA Channel n Debug Information (APBH_CH*n*_DEBUG1)....605
14.5.14 AHB to APBH DMA Channel n Debug Information (APBH_CH*n*_DEBUG2)....608

14.5.15 APBH Bridge Version Register (APBH_VERSION)..... 609

Chapter 15
Asynchronous Sample Rate Converter (ASRC)

15.1 Overview.....611
15.1.1 Features..... 613
15.1.2 Modes of Operation..... 614
15.1.2.1 Data Transfer Schemes..... 614
15.1.2.1.1 Data Input Modes.....615
15.1.2.1.2 Data Output Modes.....616
15.1.2.2 Word Alignment Supported 617
15.1.2.2.1 Input Data Alignment Modes 617
15.1.2.2.2 Output Data Alignment Modes 617
15.2 Clocks..... 618
15.3 Interrupts.....618
15.4 DMA requests..... 619
15.5 Functional Description.....619
15.5.1 Algorithm Description..... 619
15.5.1.1 Signal processing flow..... 619
15.5.1.2 Operation of the Filter.....623
15.5.1.2.1 Support of Physical Clocks.....623
15.6 Startup Procedure.....623
15.7 ASRC Memory Map/Register Definition..... 628
15.7.1 ASRC Control Register (ASRC_ASRCTR)..... 630
15.7.2 ASRC Interrupt Enable Register (ASRC_ASRIER)..... 633
15.7.3 ASRC Channel Number Configuration Register (ASRC_ASRCNCR)..... 634
15.7.4 ASRC Filter Configuration Status Register (ASRC_ASRCFG)..... 636
15.7.5 ASRC Clock Source Register (ASRC_ASRCSR).....638
15.7.6 ASRC Clock Divider Register 1 (ASRC_ASRCDR1)..... 641
15.7.7 ASRC Clock Divider Register 2 (ASRC_ASRCDR2)..... 642

15.7.8 ASRC Status Register (ASRC_ASRSTR)....643
15.7.9 ASRC Parameter Register n (ASRC_ASRPMn*n*)....646
15.7.10 ASRC ASRC Task Queue FIFO Register 1 (ASRC_ASRTFR1)....647
15.7.11 ASRC Channel Counter Register (ASRC_ASRCCR)....648
15.7.12 ASRC Data Input Register for Pair *x* (ASRC_ASRDI*n*)....649
15.7.13 ASRC Data Output Register for Pair *x* (ASRC_ASRDO*n*)....649
15.7.14 ASRC Ideal Ratio for Pair A-High Part (ASRC_ASRIDRHA)....650
15.7.15 ASRC Ideal Ratio for Pair A -Low Part (ASRC_ASRIDRLA)....650
15.7.16 ASRC Ideal Ratio for Pair B-High Part (ASRC_ASRIDRHB)....651
15.7.17 ASRC Ideal Ratio for Pair B-Low Part (ASRC_ASRIDRLB)....651
15.7.18 ASRC Ideal Ratio for Pair C-High Part (ASRC_ASRIDRHC)....652
15.7.19 ASRC Ideal Ratio for Pair C-Low Part (ASRC_ASRIDRLC)....652
15.7.20 ASRC 76 kHz Period in terms of ASRC processing clock (ASRC_ASR76K)....653
15.7.21 ASRC 56 kHz Period in terms of ASRC processing clock (ASRC_ASR56K)....654
15.7.22 ASRC Misc Control Register for Pair A (ASRC_ASRMCRA)....655
15.7.23 ASRC FIFO Status Register for Pair A (ASRC_ASRFSTA)....657
15.7.24 ASRC Misc Control Register for Pair B (ASRC_ASRMCRB)....658
15.7.25 ASRC FIFO Status Register for Pair B (ASRC_ASRFSTB)....660
15.7.26 ASRC Misc Control Register for Pair C (ASRC_ASRMCRC)....661
15.7.27 ASRC FIFO Status Register for Pair C (ASRC_ASRFSTC)....663
15.7.28 ASRC Misc Control Register 1 for Pair X (ASRC_ASRMCR1*n*)....664

Chapter 16
Digital Audio Multiplexer (AUDMUX)

16.1 Overview....667
16.1.1 Features....669
16.1.2 Modes and Operations....669
16.2 External Signals....669
16.3 Clocks....670
16.3.1 Clock Inputs....671

16.3.2 Clock Diagram....671
16.3.3 Clocking Restrictions....672
16.4 Default Register Configuration....672
16.4.1 Default Port Configuration....672
16.5 Functional Description....672
16.5.1 Operating Modes....672
16.5.1.1 Port Receive Data Modes....674
16.5.1.1.1 Normal Mode....675
16.5.1.1.2 Internal Network Mode....676
16.5.1.1.3 Transmit Data Output Enable Assertion....682
16.5.1.2 Tx/Rx Switch and External Network Mode....683
16.5.1.3 Timing Modes....684
16.5.1.3.1 Synchronous Mode (4-Wire Interface)....684
16.5.1.3.2 Asynchronous Mode (6-Wire Interface)....686
16.5.2 Connectivity Between Ports....689
16.5.2.1 Internal Port to External Port Connectivity....690
16.5.2.2 External Port to External Port Connectivity....691
16.5.2.3 Internal Port to Internal Port Connectivity....691
16.5.2.4 Loopback Connectivity....692
16.6 AUDMUX Memory Map/Register Definition....692
16.6.1 Port Timing Control Register 1 (AUDMUX_PTCR1)....693
16.6.2 Port Data Control Register 1 (AUDMUX_PDCR1)....695
16.6.3 Port Timing Control Register 2 (AUDMUX_PTCR2)....696
16.6.4 Port Data Control Register 2 (AUDMUX_PDCR2)....698
16.6.5 Port Timing Control Register 3 (AUDMUX_PTCR3)....699
16.6.6 Port Data Control Register 3 (AUDMUX_PDCR3)....701
16.6.7 Port Timing Control Register 4 (AUDMUX_PTCR4)....702
16.6.8 Port Data Control Register 4 (AUDMUX_PDCR4)....704
16.6.9 Port Timing Control Register 5 (AUDMUX_PTCR5)....705

16.6.10 Port Data Control Register 5 (AUDMUX_PDCR5)....707
16.6.11 Port Timing Control Register 6 (AUDMUX_PTCR6)....708
16.6.12 Port Data Control Register 6 (AUDMUX_PDCR6)....710
16.6.13 Port Timing Control Register 7 (AUDMUX_PTCR7)....711
16.6.14 Port Data Control Register 7 (AUDMUX_PDCR7)....713

Chapter 17
40-BIT Correcting ECC Accelerator (BCH)

17.1 Overview....715
17.2 Operation....717
17.2.1 BCH Limitations and Assumptions....718
17.2.2 Flash Page Layout....719
17.2.3 Determining the ECC layout for a device....721
17.2.3.1 4K+218 flash, 10 bytes metadata, 512 byte data blocks, separate metadata, Assuming GF(213)....721
17.2.3.2 4K+128 flash, 10 bytes metadata, 1024 byte data blocks, separate metadata, assuming GF(213) for data and GF(214) for metadata....722
17.2.4 Data Buffers in System Memory....722
17.3 Memory to Memory (Loopback) Operation....724
17.4 Programming the BCH/GPMI Interfaces....725
17.4.1 BCH Encoding for NAND Writes....725
17.4.1.1 DMA Structure Code Example....728
17.4.1.2 Using the BCH Encoder....733
17.4.2 BCH Decoding for NAND Reads....734
17.4.2.1 DMA Structure Code Example....738
17.4.2.2 Using the Decoder....741
17.4.3 Interrupts....743
17.5 Behavior During Reset....744
17.6 BCH Memory Map/Register Definition....745
17.6.1 Hardware BCH ECC Accelerator Control Register (BCH_CTRL*n*)....746
17.6.2 Hardware ECC Accelerator Status Register 0 (BCH_STATUS0)....749

17.6.3 Hardware ECC Accelerator Mode Register (BCH_MODE)..... 750
17.6.4 Hardware BCH ECC Loopback Encode Buffer Register (BCH_ENCODEPTR).....751
17.6.5 Hardware BCH ECC Loopback Data Buffer Register (BCH_DATAPTR)..... 751
17.6.6 Hardware BCH ECC Loopback Metadata Buffer Register (BCH_METAPTR).....752
17.6.7 Hardware ECC Accelerator Layout Select Register (BCH_LAYOUTSELECT)..... 752
17.6.8 Hardware BCH ECC Flash 0 Layout 0 Register (BCH_FLASH0LAYOUT0).....754
17.6.9 Hardware BCH ECC Flash 0 Layout 1 Register (BCH_FLASH0LAYOUT1).....756
17.6.10 Hardware BCH ECC Flash 1 Layout 0 Register (BCH_FLASH1LAYOUT0).....757
17.6.11 Hardware BCH ECC Flash 1 Layout 1 Register (BCH_FLASH1LAYOUT1).....759
17.6.12 Hardware BCH ECC Flash 2 Layout 0 Register (BCH_FLASH2LAYOUT0).....760
17.6.13 Hardware BCH ECC Flash 2 Layout 1 Register (BCH_FLASH2LAYOUT1).....762
17.6.14 Hardware BCH ECC Flash 3 Layout 0 Register (BCH_FLASH3LAYOUT0).....763
17.6.15 Hardware BCH ECC Flash 3 Layout 1 Register (BCH_FLASH3LAYOUT1).....765
17.6.16 Hardware BCH ECC Debug Register0 (BCH_DEBUG0*n*)..... 766
17.6.17 KES Debug Read Register (BCH_DBGKESREAD)..... 768
17.6.18 Chien Search Debug Read Register (BCH_DBGCSFEREAD)..... 769
17.6.19 Syndrome Generator Debug Read Register (BCH_DBGSYNDGENREAD).....769
17.6.20 Bus Master and ECC Controller Debug Read Register (BCH_DBGAHBMREAD).....770
17.6.21 Block Name Register (BCH_BLOCKNAME)..... 770
17.6.22 BCH Version Register (BCH_VERSION)..... 771
17.6.23 Hardware BCH ECC Debug Register 1 (BCH_DEBUG1)..... 772

Chapter 18
Clock Controller Module (CCM)

18.1 Overview.....775
18.1.1 Features..... 775
18.1.2 CCM Block Diagram..... 776
18.2 External Signals..... 778
18.3 CCM Clock Tree.....778
18.4 System Clocks.....782

18.5 Functional Description....793
18.5.1 Clock Generation....793
18.5.1.1 External Low Frequency Clock - CKIL....793
18.5.1.1.1 CKIL synchronizing to IPG_CLK....793
18.5.1.2 External High Frequency Clock - CKIH and internal oscillator....794
18.5.1.3 PLL reference clock....794
18.5.1.3.1 ARM PLL (PLL1)....794
18.5.1.3.2 System PLL (PLL2)....794
18.5.1.3.3 USB1 PLL (PLL3)....795
18.5.1.3.4 Audio PLL (PLL4)....795
18.5.1.3.5 Video PLL (PLL5)....796
18.5.1.3.6 MLB PLL....796
18.5.1.3.7 Ethernet PLL (PLL6)....796
18.5.1.3.8 USB2 PLL (PLL7)....796
18.5.1.4 Phase Fractional Dividers (PFD)....796
18.5.1.5 CCM internal clock generation....797
18.5.1.5.1 Clock Switcher....797
18.5.1.5.2 PLL bypass procedure....799
18.5.1.5.3 PLL clock change....799
18.5.1.5.4 Clock Root Generator....799
18.5.1.5.5 Initial values controlled by the System JTAG Controller (SJC)....811
18.5.1.5.6 Divider change handshake....812
18.5.1.6 Disabling / Enabling PLLs....812
18.5.1.7 Clock Switching Multiplexers....812
18.5.1.8 Low Power Clock Gating module (LPCG)....815
18.5.1.9 MMDC handshake....816
18.5.2 DVFS support....817
18.5.3 Power modes....817
18.5.3.1 RUN mode....817

18.5.3.2 WAIT mode ... 817
18.5.3.2.1 Entering WAIT mode ... 817
18.5.3.2.2 Exiting WAIT mode ... 818
18.5.3.3 STOP mode ... 818
18.5.3.3.1 Entering STOP mode ... 818
18.5.3.3.2 Exiting STOP mode ... 819
18.6 CCM Memory Map/Register Definition ... 820
18.6.1 CCM Control Register (CCM_CCR) ... 822
18.6.2 CCM Control Divider Register (CCM_CCDR) ... 824
18.6.3 CCM Status Register (CCM_CSR) ... 825
18.6.4 CCM Clock Switcher Register (CCM_CCSR) ... 826
18.6.5 CCM Arm Clock Root Register (CCM_CACRR) ... 828
18.6.6 CCM Bus Clock Divider Register (CCM_CBCDR) ... 829
18.6.7 CCM Bus Clock Multiplexer Register (CCM_CBCMR) ... 832
18.6.8 CCM Serial Clock Multiplexer Register 1 (CCM_CSCMR1) ... 834
18.6.9 CCM Serial Clock Multiplexer Register 2 (CCM_CSCMR2) ... 837
18.6.10 CCM Serial Clock Divider Register 1 (CCM_CSCDR1) ... 838
18.6.11 CCM SSI1 Clock Divider Register (CCM_CS1CDR) ... 841
18.6.12 CCM SSI2 Clock Divider Register (CCM_CS2CDR) ... 842
18.6.13 CCM D1 Clock Divider Register (CCM_CDCDR) ... 844
18.6.14 CCM HSC Clock Divider Register (CCM_CHSCCDR) ... 846
18.6.15 CCM Serial Clock Divider Register 2 (CCM_CSCDR2) ... 849
18.6.16 CCM Serial Clock Divider Register 3 (CCM_CSCDR3) ... 851
18.6.17 CCM Divider Handshake In-Process Register (CCM_CDHIPR) ... 853
18.6.18 CCM Low Power Control Register (CCM_CLPCR) ... 856
18.6.19 CCM Interrupt Status Register (CCM_CISR) ... 859
18.6.20 CCM Interrupt Mask Register (CCM_CIMR) ... 862
18.6.21 CCM Clock Output Source Register (CCM_CCOSR) ... 865
18.6.22 CCM General Purpose Register (CCM_CGPR) ... 868

18.6.23 CCM Clock Gating Register 0 (CCM_CCGR0)....869
18.6.24 CCM Clock Gating Register 1 (CCM_CCGR1)....871
18.6.25 CCM Clock Gating Register 2 (CCM_CCGR2)....872
18.6.26 CCM Clock Gating Register 3 (CCM_CCGR3)....874
18.6.27 CCM Clock Gating Register 4 (CCM_CCGR4)....875
18.6.28 CCM Clock Gating Register 5 (CCM_CCGR5)....876
18.6.29 CCM Clock Gating Register 6 (CCM_CCGR6)....878
18.6.30 CCM Module Enable Overide Register (CCM_CMEOR).... 879
18.7 CCM Analog Memory Map/Register Definition....880
18.7.1 Analog ARM PLL control Register (CCM_ANALOG_PLL_ARM*n*).... 884
18.7.2 Analog USB1 480MHz PLL Control Register (CCM_ANALOG_PLL_USB1*n*).... 886
18.7.3 Analog USB2 480MHz PLL Control Register (CCM_ANALOG_PLL_USB2*n*).... 889
18.7.4 Analog System PLL Control Register (CCM_ANALOG_PLL_SYS*n*)....891
18.7.5 528MHz System PLL Spread Spectrum Register (CCM_ANALOG_PLL_SYS_SS).... 893
18.7.6 Numerator of 528MHz System PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_SYS_NUM).... 893
18.7.7 Denominator of 528MHz System PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_SYS_DENOM).... 894
18.7.8 Analog Audio PLL control Register (CCM_ANALOG_PLL_AUDIO*n*)....895
18.7.9 Numerator of Audio PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_AUDIO_NUM).. 897
18.7.10 Denominator of Audio PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_AUDIO_DENOM)....897
18.7.11 Analog Video PLL control Register (CCM_ANALOG_PLL_VIDEO*n*).... 899
18.7.12 Numerator of Video PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_VIDEO_NUM)...901
18.7.13 Denominator of Video PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_VIDEO_DENOM).... 901
18.7.14 MLB PLL Control Register (CCM_ANALOG_PLL_MLB*n*).... 902
18.7.15 Analog ENET PLL Control Register (CCM_ANALOG_PLL_ENET*n*).... 904
18.7.16 480MHz Clock (PLL3) Phase Fractional Divider Control Register (CCM_ANALOG_PFD_480*n*)..... 905
18.7.17 528MHz Clock (PLL2) Phase Fractional Divider Control Register (CCM_ANALOG_PFD_528*n*)..... 907

18.7.18 Miscellaneous Register 0 (CCM_ANALOG_MISC0*n*)..... 910
18.7.19 Miscellaneous Register 1 (CCM_ANALOG_MISC1*n*)..... 914
18.7.20 Miscellaneous Register 2 (CCM_ANALOG_MISC2*n*)..... 917

Chapter 19
MIPI CSI to IPU Gasket (CSI2IPU)

19.1 Overview..... 923
19.1.1 CSI2IPU feature summary..... 924
19.1.2 CSI2IPU architectural description..... 924
19.2 CSI2IPU signals..... 925
19.3 Timing interface..... 926
19.4 Payload data output formats..... 927
19.5 CSI2IPU Memory Map/Register Definition..... 934
19.5.1 CSI 2 IPU Gasket Software Reset (CSI2IPU_SW_RST)..... 935

Chapter 20
Display Content Integrity Checker (DCIC)

20.1 Overview..... 937
20.1.1 Block Diagram..... 938
20.1.2 Features..... 939
20.2 Functional Description..... 939
20.2.1 Generic synchronous parallel display interface..... 939
20.2.2 CRC Polynomial..... 940
20.2.3 Mode of operation..... 941
20.2.4 Interrupts..... 941
20.2.5 Software reset..... 941
20.2.6 Clock domains..... 942
20.2.7 External controller mismatch indication signal..... 942
20.2.8 Power saving..... 942
20.2.9 System Considerations..... 942
20.3 DCIC Memory Map/Register Definition..... 943

20.3.1 DCIC Control Register (DCIC*x*_DCICC)..... 944
20.3.2 DCIC Interrupt Control Register (DCIC*x*_DCICIC)..... 945
20.3.3 DCIC Status Register (DCIC*x*_DCICS)..... 946
20.3.4 DCIC ROI Config Register m (DCIC*x*_DCICRC)..... 947
20.3.5 DCIC ROI Size Register m (DCIC*x*_DCICRS)..... 948
20.3.6 DCIC ROI Reference Signature Register m (DCIC*x*_DCICRRS)..... 949
20.3.7 DCIC ROI Calculated Signature m (DCIC*x*_DCICRCS)..... 949

Chapter 21
Enhanced Configurable SPI (ECSPI)

21.1 Overview..... 951
21.1.1 Features..... 952
21.1.2 Modes and Operations..... 952
21.2 External Signals..... 953
21.3 Clocks..... 956
21.4 Functional Description..... 956
21.4.1 Master Mode..... 957
21.4.2 Slave Mode..... 957
21.4.3 Low Power Modes..... 957
21.4.4 Operations..... 957
21.4.4.1 Typical Master Mode..... 957
21.4.4.1.1 Master Mode with SPI_RDY..... 958
21.4.4.1.2 Master Mode with Wait States..... 960
21.4.4.1.3 Master Mode with SS_CTL[3:0] Control..... 960
21.4.4.1.4 Master Mode with Phase Control..... 961
21.4.4.2 Typical Slave Mode..... 962
21.4.5 Reset..... 963
21.4.6 Interrupts..... 963
21.4.7 DMA..... 965
21.4.8 Byte Order..... 966

21.5 Initialization....966
21.6 Applications....966
21.7 ECSPI Memory Map/Register Definition....967
21.7.1 Receive Data Register (ECSPI*x*_RXDATA)....969
21.7.2 Transmit Data Register (ECSPI*x*_TXDATA)....970
21.7.3 Control Register (ECSPI*x*_CONREG)....970
21.7.4 Config Register (ECSPI*x*_CONFIGREG)....973
21.7.5 Interrupt Control Register (ECSPI*x*_INTREG)....975
21.7.6 DMA Control Register (ECSPI*x*_DMAREG)....976
21.7.7 Status Register (ECSPI*x*_STATREG)....978
21.7.8 Sample Period Control Register (ECSPI*x*_PERIODREG)....979
21.7.9 Test Control Register (ECSPI*x*_TESTREG)....981
21.7.10 Message Data Register (ECSPI*x*_MSGDATA)....982

Chapter 22
External Interface Module (EIM)

22.1 Overview....983
22.1.1 Features....985
22.1.2 Modes of Operation....985
22.1.2.1 Asynchronous Mode....986
22.1.2.2 Asynchronous Page Read Mode....986
22.1.2.3 Multiplexed Address/Data Mode....986
22.1.2.4 Burst Clock Mode....987
22.1.2.5 Low Power Modes....988
22.1.2.6 Boot Mode....988
22.2 External Signals....988
22.2.1 Other Important Block I/O Signals Internal to the SoC....993
22.3 Clocks....994
22.4 Chip Select Memory Map....995
22.5 Functional Description....995

22.5.1 Continuous BCLK....995
22.5.2 Bus Sizing Configuration....996
22.5.2.1 8 BIT PORT SUPPORT....997
22.5.2.1.1 MOTOROLA 68000....997
22.5.2.1.2 INTEL 386....997
22.5.3 EIM Operational Modes....997
22.5.4 Burst Mode (Synchronous) Memory Operation....998
22.5.5 Burst Clock Divisor (BCD)....999
22.5.6 Burst Clock Start (BCS)....999
22.5.7 Multiplexed Address/Data Mode Support....1000
22.5.8 Mixed Master/Memory Burst Modes Support....1000
22.5.9 AXI (Master) Bus Cycles Support....1000
22.5.10 WAIT_B Signal, RWSC and WWSC bit fields Usage....1003
22.5.11 IPS Register Interface....1003
22.5.12 MRS Set for PSRAM....1003
22.5.13 EIM Access Termination....1004
22.5.14 Error Conditions....1004
22.5.15 DTACK Mode....1005
22.5.16 RDY_INT Signal as Interrupt....1005
22.5.17 RDY_INT Signal as Ready After Reset Indication....1005
22.5.18 EIM_GRANT / EIM_BUSY Handshake Description....1006
22.5.19 LPMD / LPACK Handshake Description....1006
22.5.20 Endianness....1006
22.5.21 Strobe Signal Use....1007
22.6 Initialization Information....1008
22.6.1 Booting from EIM....1008
22.7 Typical Application....1008
22.7.1 Access to Intel Sibley Flash....1009
22.7.1.1 Intel Sibley Flash Asynchronous Mode Configuration....1009

22.7.1.2 Intel Sibley Flash Synchronous Mode Configuration....1009
22.7.1.3 Intel Sibley Flash Utility....1009
22.7.2 Access to MDOC Device....1010
22.7.2.1 MDOC Device Boot....1010
22.7.2.2 MDOC Device Asynchronous Mode Configuration....1010
22.7.2.3 MDOC Device Utility....1010
22.7.3 Access to Micron PSRAM....1011
22.7.3.1 Micron PSRAM Asynchronous Mode Configuration....1011
22.7.3.2 Micron PSRAM Synchronous Mode Configuration....1011
22.7.4 Access to Samsung OneNAND....1011
22.7.4.1 Samsung OneNAND Boot....1011
22.7.4.2 Samsung OneNAND Asynchronous Mode Configuration....1012
22.7.4.3 Samsung OneNAND Synchronous Mode Configuration....1012
22.7.4.4 Samsung OneNAND Utility....1012
22.7.5 Access to Samsung UtRAM....1013
22.7.5.1 Samsung UtRAM Asynchronous Mode Configuration....1013
22.7.5.2 Samsung UtRAM Synchronous Mode Configuration....1013
22.7.6 Access to Spansion Flash....1013
22.7.6.1 Spansion Flash Asynchronous Mode Configuration....1014
22.7.6.2 Spansion Flash Synchronous Mode Configuration....1014
22.7.6.3 Spansion Flash Utility....1014
22.7.7 8 bit support....1016
22.8 External Bus Timing Diagrams....1016
22.8.1 Asynchronous Read Memory Accesses Timing Diagram....1016
22.8.2 Asynchronous Write Memory Accesses Timing Diagram....1017
22.8.3 Asynchronous Read/Write Memory Accesses Timing Diagram....1018
22.8.4 Asynchronous Read/Write Using RAL, WAL and CSREC....1020
22.8.5 Consecutive Asynchronous Write Memory Accesses Timing Diagram....1021
22.8.6 Consecutive Asynchronous Read Memory Accesses Timing Diagram....1024

22.8.7 Burst (Synchronous Mode) Read Memory Accesses Timing Diagram - BCD=0 1026
22.8.8 Burst (Synchronous Mode) Read Memory Accesses Timing Diagram - BCD=1 1027
22.8.9 Burst (Synchronous Mode) Write Memory Access Timing - BCD=1 1028
22.8.10 Asynchronous Page Mode Access 1030
22.8.11 DTACK Mode - AXI Single Access 1030
22.8.12 DTACK Mode - AXI Single Write Access 1033
22.8.13 DTACK Mode - AXI Burst Access 1034
22.9 EIM Memory Map/Register Definition 1035
22.9.1 Chip Select n General Configuration Register 1 (EIM_CS*n*GCR1) 1038
22.9.2 Chip Select n General Configuration Register 2 (EIM_CS*n*GCR2) 1042
22.9.3 Chip Select n Read Configuration Register 1 (EIM_CS*n*RCR1) 1043
22.9.4 Chip Select n Read Configuration Register 2 (EIM_CS*n*RCR2) 1046
22.9.5 Chip Select n Write Configuration Register 1 (EIM_CS*n*WCR1) 1047
22.9.6 Chip Select n Write Configuration Register 2 (EIM_CS*n*WCR2) 1050
22.9.7 EIM Configuration Register (EIM_WCR) 1051
22.9.8 DLL Control Register (EIM_DCR) 1053
22.9.9 DLL Status Register (EIM_DSR) 1054
22.9.10 EIM IP Access Register (EIM_WIAR) 1055
22.9.11 Error Address Register (EIM_EAR) 1056

Chapter 23
10/100/1000-Mbps Ethernet MAC (ENET)

23.1 Introduction 1057
23.2 Overview 1057
23.2.1 Features 1058
23.2.1.1 Ethernet MAC features 1058
23.2.1.2 IP protocol performance optimization features 1059
23.2.1.3 IEEE 1588 features 1060
23.2.2 Block diagram 1060
23.3 External Signals 1061

23.4 Clocks ... 1066
23.5 Memory map/register definition ... 1067
23.5.1 Interrupt Event Register (ENET_EIR) ... 1072
23.5.2 Interrupt Mask Register (ENET_EIMR) ... 1075
23.5.3 Receive Descriptor Active Register (ENET_RDAR) ... 1078
23.5.4 Transmit Descriptor Active Register (ENET_TDAR) ... 1078
23.5.5 Ethernet Control Register (ENET_ECR) ... 1080
23.5.6 MII Management Frame Register (ENET_MMFR) ... 1082
23.5.7 MII Speed Control Register (ENET_MSCR) ... 1083
23.5.8 MIB Control Register (ENET_MIBC) ... 1085
23.5.9 Receive Control Register (ENET_RCR) ... 1087
23.5.10 Transmit Control Register (ENET_TCR) ... 1090
23.5.11 Physical Address Lower Register (ENET_PALR) ... 1092
23.5.12 Physical Address Upper Register (ENET_PAUR) ... 1092
23.5.13 Opcode/Pause Duration Register (ENET_OPD) ... 1093
23.5.14 Descriptor Individual Upper Address Register (ENET_IAUR) ... 1093
23.5.15 Descriptor Individual Lower Address Register (ENET_IALR) ... 1094
23.5.16 Descriptor Group Upper Address Register (ENET_GAUR) ... 1094
23.5.17 Descriptor Group Lower Address Register (ENET_GALR) ... 1095
23.5.18 Transmit FIFO Watermark Register (ENET_TFWR) ... 1095
23.5.19 Receive Descriptor Ring Start Register (ENET_RDSR) ... 1096
23.5.20 Transmit Buffer Descriptor Ring Start Register (ENET_TDSR) ... 1097
23.5.21 Maximum Receive Buffer Size Register (ENET_MRBR) ... 1098
23.5.22 Receive FIFO Section Full Threshold (ENET_RSFL) ... 1099
23.5.23 Receive FIFO Section Empty Threshold (ENET_RSEM) ... 1099
23.5.24 Receive FIFO Almost Empty Threshold (ENET_RAEM) ... 1100
23.5.25 Receive FIFO Almost Full Threshold (ENET_RAFL) ... 1100
23.5.26 Transmit FIFO Section Empty Threshold (ENET_TSEM) ... 1101
23.5.27 Transmit FIFO Almost Empty Threshold (ENET_TAEM) ... 1101

23.5.28 Transmit FIFO Almost Full Threshold (ENET_TAFL)..... 1102
23.5.29 Transmit Inter-Packet Gap (ENET_TIPG)..... 1102
23.5.30 Frame Truncation Length (ENET_FTRL)..... 1103
23.5.31 Transmit Accelerator Function Configuration (ENET_TACC)..... 1103
23.5.32 Receive Accelerator Function Configuration (ENET_RACC)..... 1104
23.5.33 Reserved Statistic Register (ENET_RMON_T_DROP)..... 1105
23.5.34 Tx Packet Count Statistic Register (ENET_RMON_T_PACKETS)..... 1106
23.5.35 Tx Broadcast Packets Statistic Register (ENET_RMON_T_BC_PKT)..... 1106
23.5.36 Tx Multicast Packets Statistic Register (ENET_RMON_T_MC_PKT)..... 1107
23.5.37 Tx Packets with CRC/Align Error Statistic Register (ENET_RMON_T_CRC_ALIGN)..... 1107
23.5.38 Tx Packets Less Than Bytes and Good CRC Statistic Register (ENET_RMON_T_UNDERSIZE)..... 1108
23.5.39 Tx Packets GT MAX_FL bytes and Good CRC Statistic Register (ENET_RMON_T_OVERSIZE)..... 1108
23.5.40 Tx Packets Less Than 64 Bytes and Bad CRC Statistic Register (ENET_RMON_T_FRAG)..... 1109
23.5.41 Tx Packets Greater Than MAX_FL bytes and Bad CRC Statistic Register (ENET_RMON_T_JAB).... 1109
23.5.42 Tx Collision Count Statistic Register (ENET_RMON_T_COL)..... 1110
23.5.43 Tx 64-Byte Packets Statistic Register (ENET_RMON_T_P64)..... 1110
23.5.44 Tx 65- to 127-byte Packets Statistic Register (ENET_RMON_T_P65TO127)..... 1111
23.5.45 Tx 128- to 255-byte Packets Statistic Register (ENET_RMON_T_P128TO255)..... 1111
23.5.46 Tx 256- to 511-byte Packets Statistic Register (ENET_RMON_T_P256TO511)..... 1112
23.5.47 Tx 512- to 1023-byte Packets Statistic Register (ENET_RMON_T_P512TO1023)..... 1112
23.5.48 Tx 1024- to 2047-byte Packets Statistic Register (ENET_RMON_T_P1024TO2047)..... 1113
23.5.49 Tx Packets Greater Than 2048 Bytes Statistic Register (ENET_RMON_T_P_GTE2048)..... 1113
23.5.50 Tx Octets Statistic Register (ENET_RMON_T_OCTETS)..... 1113
23.5.51 Reserved Statistic Register (ENET_IEEE_T_DROP)..... 1114
23.5.52 Frames Transmitted OK Statistic Register (ENET_IEEE_T_FRAME_OK)..... 1114
23.5.53 Frames Transmitted with Single Collision Statistic Register (ENET_IEEE_T_1COL)..... 1115
23.5.54 Frames Transmitted with Multiple Collisions Statistic Register (ENET_IEEE_T_MCOL)..... 1115
23.5.55 Frames Transmitted after Deferral Delay Statistic Register (ENET_IEEE_T_DEF)..... 1116
23.5.56 Frames Transmitted with Late Collision Statistic Register (ENET_IEEE_T_LCOL)..... 1116

23.5.57 Frames Transmitted with Excessive Collisions Statistic Register (ENET_IEEE_T_EXCOL)..... 1117
23.5.58 Frames Transmitted with Tx FIFO Underrun Statistic Register (ENET_IEEE_T_MACERR)..... 1117
23.5.59 Frames Transmitted with Carrier Sense Error Statistic Register (ENET_IEEE_T_CSERR)..... 1118
23.5.60 Reserved Statistic Register (ENET_IEEE_T_SQE)..... 1118
23.5.61 Flow Control Pause Frames Transmitted Statistic Register (ENET_IEEE_T_FDXFC)..... 1119
23.5.62 Octet Count for Frames Transmitted w/o Error Statistic Register (ENET_IEEE_T_OCTETS_OK)..... 1119
23.5.63 Rx Packet Count Statistic Register (ENET_RMON_R_PACKETS)..... 1120
23.5.64 Rx Broadcast Packets Statistic Register (ENET_RMON_R_BC_PKT)..... 1120
23.5.65 Rx Multicast Packets Statistic Register (ENET_RMON_R_MC_PKT)..... 1121
23.5.66 Rx Packets with CRC/Align Error Statistic Register (ENET_RMON_R_CRC_ALIGN)..... 1121
23.5.67 Rx Packets with Less Than 64 Bytes and Good CRC Statistic Register (ENET_RMON_R_UNDERSIZE)..... 1122
23.5.68 Rx Packets Greater Than MAX_FL and Good CRC Statistic Register (ENET_RMON_R_OVERSIZE)..... 1122
23.5.69 Rx Packets Less Than 64 Bytes and Bad CRC Statistic Register (ENET_RMON_R_FRAG)..... 1123
23.5.70 Rx Packets Greater Than MAX_FL Bytes and Bad CRC Statistic Register (ENET_RMON_R_JAB)... 1123
23.5.71 Reserved Statistic Register (ENET_RMON_R_RESVD_0)..... 1123
23.5.72 Rx 64-Byte Packets Statistic Register (ENET_RMON_R_P64)..... 1124
23.5.73 Rx 65- to 127-Byte Packets Statistic Register (ENET_RMON_R_P65TO127)..... 1124
23.5.74 Rx 128- to 255-Byte Packets Statistic Register (ENET_RMON_R_P128TO255)..... 1125
23.5.75 Rx 256- to 511-Byte Packets Statistic Register (ENET_RMON_R_P256TO511)..... 1125
23.5.76 Rx 512- to 1023-Byte Packets Statistic Register (ENET_RMON_R_P512TO1023)..... 1126
23.5.77 Rx 1024- to 2047-Byte Packets Statistic Register (ENET_RMON_R_P1024TO2047)..... 1126
23.5.78 Rx Packets Greater than 2048 Bytes Statistic Register (ENET_RMON_R_P_GTE2048)..... 1127
23.5.79 Rx Octets Statistic Register (ENET_RMON_R_OCTETS)..... 1127
23.5.80 Frames not Counted Correctly Statistic Register (ENET_IEEE_R_DROP)..... 1128
23.5.81 Frames Received OK Statistic Register (ENET_IEEE_R_FRAME_OK)..... 1128
23.5.82 Frames Received with CRC Error Statistic Register (ENET_IEEE_R_CRC)..... 1129
23.5.83 Frames Received with Alignment Error Statistic Register (ENET_IEEE_R_ALIGN)..... 1129

23.5.84 Receive FIFO Overflow Count Statistic Register (ENET_IEEE_R_MACERR)....1130
23.5.85 Flow Control Pause Frames Received Statistic Register (ENET_IEEE_R_FDXFC)....1130
23.5.86 Octet Count for Frames Received without Error Statistic Register (ENET_IEEE_R_OCTETS_OK)....1131
23.5.87 Adjustable Timer Control Register (ENET_ATCR)....1131
23.5.88 Timer Value Register (ENET_ATVR)....1133
23.5.89 Timer Offset Register (ENET_ATOFF)....1133
23.5.90 Timer Period Register (ENET_ATPER)....1134
23.5.91 Timer Correction Register (ENET_ATCOR)....1134
23.5.92 Time-Stamping Clock Period Register (ENET_ATINC)....1135
23.5.93 Timestamp of Last Transmitted Frame (ENET_ATSTMP)....1135
23.5.94 Timer Global Status Register (ENET_TGSR)....1136
23.5.95 Timer Control Status Register (ENET_TCSR*n*)....1137
23.5.96 Timer Compare Capture Register (ENET_TCCR*n*)....1138
23.6 Functional description....1138
23.6.1 Ethernet MAC frame formats....1139
23.6.1.1 Pause Frames....1140
23.6.1.2 Magic packets....1141
23.6.2 IP and higher layers frame format....1142
23.6.2.1 Ethernet types....1142
23.6.2.2 IPv4 datagram format....1142
23.6.2.3 IPv6 datagram format....1143
23.6.2.4 Internet Control Message Protocol (ICMP) datagram format....1144
23.6.2.5 User Datagram Protocol (UDP) datagram format....1145
23.6.2.6 TCP datagram format....1145
23.6.3 IEEE 1588 message formats....1146
23.6.3.1 Transport encapsulation....1146
23.6.3.1.1 UDP/IP....1146
23.6.3.1.2 Native Ethernet (PTPv2)....1147
23.6.3.2 PTP header....1147

23.6.3.2.1 PTPv1 header ... 1148
23.6.3.2.2 PTPv2 header ... 1148
23.6.4 MAC receive ... 1150
23.6.4.1 Collision detection in half-duplex mode ... 1151
23.6.4.2 Preamble processing ... 1151
23.6.4.3 MAC address check ... 1152
23.6.4.3.1 Unicast address check ... 1152
23.6.4.3.2 Multicast and unicast address resolution ... 1152
23.6.4.3.3 Broadcast address reject ... 1153
23.6.4.3.4 Miss-bit implementation ... 1153
23.6.4.4 Frame length/type verification: payload length check ... 1154
23.6.4.5 Frame length/type verification: frame length check ... 1154
23.6.4.6 VLAN frames processing ... 1154
23.6.4.7 Pause frame termination ... 1154
23.6.4.8 CRC check ... 1155
23.6.4.9 Frame padding removal ... 1155
23.6.5 MAC transmit ... 1156
23.6.5.1 Frame payload padding ... 1157
23.6.5.2 MAC address insertion ... 1157
23.6.5.3 CRC-32 generation ... 1157
23.6.5.4 Inter-packet gap (IPG) ... 1158
23.6.5.5 Collision detection and handling — half-duplex operation only ... 1158
23.6.6 Full-duplex flow control operation ... 1160
23.6.6.1 Remote device congestion ... 1160
23.6.6.2 Local device/FIFO congestion ... 1161
23.6.7 Magic packet detection ... 1162
23.6.7.1 Sleep mode ... 1162
23.6.7.2 Magic packet detection ... 1162
23.6.7.3 Wakeup ... 1162

23.6.8 IP accelerator functions....1163
23.6.8.1 Checksum calculation....1163
23.6.8.2 Additional padding processing....1164
23.6.8.3 32-bit Ethernet payload alignment....1164
23.6.8.3.1 Receive processing....1165
23.6.8.3.2 Transmit processing....1165
23.6.8.4 Received frame discard....1165
23.6.8.5 IPv4 fragments....1166
23.6.8.6 IPv6 support....1166
23.6.8.6.1 Receive processing....1166
23.6.8.6.2 Transmit processing....1167
23.6.9 Resets and stop controls....1167
23.6.9.1 Hardware reset....1167
23.6.9.2 Soft reset....1167
23.6.9.3 Hardware freeze....1168
23.6.9.4 Graceful stop....1168
23.6.9.4.1 Graceful transmit stop (GTS)....1169
23.6.9.4.2 Graceful receive stop (GRS)....1169
23.6.9.4.3 Graceful stop interrupt (GRA)....1170
23.6.10 IEEE 1588 functions....1170
23.6.10.1 Adjustable timer module....1171
23.6.10.1.1 Adjustable timer implementation....1171
23.6.10.2 Transmit timestamping....1173
23.6.10.3 Receive timestamping....1173
23.6.10.4 Time synchronization....1173
23.6.10.5 Input Capture and Output Compare....1174
23.6.10.5.1 Input capture....1174
23.6.10.5.2 Output compare....1174
23.6.10.5.3 DMA requests....1174

23.6.11 FIFO thresholds....1174
23.6.11.1 Receive FIFO....1175
23.6.11.2 Transmit FIFO....1176
23.6.12 Loopback options....1177
23.6.13 Legacy buffer descriptors....1178
23.6.13.1 Legacy receive buffer descriptor....1178
23.6.13.2 Legacy transmit buffer descriptor....1178
23.6.14 Enhanced buffer descriptors....1179
23.6.14.1 Enhanced receive buffer descriptor....1179
23.6.14.2 Enhanced transmit buffer descriptor....1183
23.6.15 Client FIFO application interface....1185
23.6.15.1 Data structure description....1185
23.6.15.2 Data structure examples....1187
23.6.15.3 Frame status....1188
23.6.16 FIFO protection....1188
23.6.16.1 Transmit FIFO underflow....1188
23.6.16.2 Transmit FIFO overflow....1189
23.6.16.3 Receive FIFO overflow....1190
23.6.17 PHY management interface....1190
23.6.17.1 MDIO clause 22 frame format....1191
23.6.17.2 MDIO clause 45 frame format....1191
23.6.17.3 MDIO clock generation....1193
23.6.17.4 MDIO operation....1193
23.6.18 Ethernet interfaces....1193
23.6.18.1 RMII interface....1194
23.6.18.2 RGMII interface....1195
23.6.18.3 MII Interface — transmit....1196
23.6.18.3.1 Transmit with collision — half-duplex....1197
23.6.18.4 MII interface — receive....1197

Chapter 24
Electrophoretic Display Controller (EPDC)

24.1 Overview 1199

24.1.1 EPDC Block Diagram 1200

24.2 External Signals 1201

24.3 Programming Model 1203

24.3.1 Assumptions 1203

24.3.2 Register Space (Write/Set/Clear/Toggle) 1204

24.3.3 Interrupts 1205

24.3.3.1 Interrupt Sources 1205

24.3.3.2 Enabling/Masking Interrupts 1205

24.3.3.3 Handling/Clearing Interrupts 1205

24.3.4 Controller Setup 1206

24.3.4.1 Memory Requirements 1206

24.3.4.2 Panel Architecture Configuration 1207

24.3.4.3 TFT Panel Timing Configuration 1210

24.3.4.4 Source Driver and Pixel Clock Configuration 1215

24.3.4.5 Initializing the Display 1216

24.3.4.5.1 Reset/Clocks and Buffer Preparation 1217

24.3.4.5.2 Performing an Initialization Display Update 1217

24.3.5 Dual-Scan Configuration 1218

24.3.6 Update Buffer Analysis Functions 1221

24.3.7 Waveform Mode Selection (AUTOWV) 1222

24.3.8 Panel Interface Generator (Pigeon Mode) 1222

24.3.9 Display Update Programming 1226

24.3.9.1 Initiating a Display Update 1226

24.3.9.2 Update Processing and Collisions 1227

24.3.9.3 Multiple Update Flow 1231

24.3.10 Architectural Clock Gating (Low Power Mode) 1234

24.3.11 Performance Tuning and Considerations....1236
24.3.11.1 Memory and Bus Bandwidth Requirements....1236
24.3.11.2 Pixel Latency FIFO....1237
24.3.11.3 Basic Watermarking Control....1237
24.3.11.4 Update/Refresh Tuning (VSCAN_HOLDOFF)....1238
24.3.11.5 System-Level Arbitration Control....1238
24.4 EPDC Memory Map/Register Definition....1240
24.4.1 EPDC Control Register (EPDC_CTRL*n*)....1249
24.4.2 EPDC Waveform Address Pointer (EPDC_WVADDR)....1250
24.4.3 EPDC Working Buffer Address (EPDC_WB_ADDR)....1251
24.4.4 EPDC Screen Resolution (EPDC_RES)....1252
24.4.5 EPDC Format Control Register (EPDC_FORMAT*n*)....1252
24.4.6 EPDC FIFO control register (EPDC_FIFOCTRL*n*)....1254
24.4.7 EPDC Update Region Address (EPDC_UPD_ADDR)....1255
24.4.8 EPDC Update Region Stride (EPDC_UPD_STRIDE)....1256
24.4.9 EPDC Update Command Co-ordinate (EPDC_UPD_CORD)....1257
24.4.10 EPDC Update Command Size (EPDC_UPD_SIZE)....1257
24.4.11 EPDC Update Command Control (EPDC_UPD_CTRL*n*)....1258
24.4.12 EPDC Update Fixed Pixel Control (EPDC_UPD_FIXED*n*)....1259
24.4.13 EPDC Temperature Register (EPDC_TEMP)....1260
24.4.14 Waveform Mode Lookup Table Control Register. (EPDC_AUTOWV_LUT)....1261
24.4.15 EPDC Timing Control Engine Control Register (EPDC_TCE_CTRL*n*)....1261
24.4.16 EPDC Timing Control Engine Source-Driver Config Register (EPDC_TCE_SDCFG*n*)....1263
24.4.17 EPDC Timing Control Engine Gate-Driver Config Register (EPDC_TCE_GDCFG*n*)....1265
24.4.18 EPDC Timing Control Engine Horizontal Timing Register 1 (EPDC_TCE_HSCAN1*n*)....1266
24.4.19 EPDC Timing Control Engine Horizontal Timing Register 2 (EPDC_TCE_HSCAN2*n*)....1266
24.4.20 EPDC Timing Control Engine Vertical Timing Register (EPDC_TCE_VSCAN*n*)....1267
24.4.21 EPDC Timing Control Engine OE timing control Register (EPDC_TCE_OE*n*)....1268
24.4.22 EPDC Timing Control Engine Driver Polarity Register (EPDC_TCE_POLARITY)....1268

24.4.23 EPDC Timing Control Engine Timing Register 1 (EPDC_TCE_TIMING1*n*)..... 1269
24.4.24 EPDC Timing Control Engine Timing Register 2 (EPDC_TCE_TIMING2*n*)..... 1271
24.4.25 EPDC Timing Control Engine Timing Register 3 (EPDC_TCE_TIMING3*n*)..... 1271
24.4.26 EPDC Pigeon Mode Control Register 0 (EPDC_PIGEON_CTRL0*n*).....1272
24.4.27 EPDC Pigeon Mode Control Register 1 (EPDC_PIGEON_CTRL1*n*).....1272
24.4.28 EPDC IRQ Mask Register for LUT 0~31 (EPDC_IRQ_MASK1*n*)..... 1273
24.4.29 EPDC IRQ Mask Register for LUT 32~63 (EPDC_IRQ_MASK2*n*)..... 1273
24.4.30 EPDC Interrupt Register for LUT 0~31 (EPDC_IRQ1*n*).....1274
24.4.31 EPDC Interrupt Registerr for LUT 32~63 (EPDC_IRQ2*n*)..... 1274
24.4.32 EPDC IRQ Mask Register (EPDC_IRQ_MASK*n*)..... 1275
24.4.33 EPDC Interrupt Register (EPDC_IRQ*n*)..... 1276
24.4.34 EPDC Status Register - LUTs (EPDC_STATUS_LUTS1*n*).....1278
24.4.35 EPDC Status Register - LUTs (EPDC_STATUS_LUTS2*n*).....1278
24.4.36 EPDC Status Register - Next Available LUT (EPDC_STATUS_NEXTLUT)..... 1278
24.4.37 EPDC LUT Collision Status (EPDC_STATUS_COL1*n*)..... 1280
24.4.38 EPDC LUT Collision Status (EPDC_STATUS_COL2*n*)..... 1280
24.4.39 EPDC General Status Register (EPDC_STATUS*n*).....1281
24.4.40 EPDC Collision Region Co-ordinate (EPDC_UPD_COL_CORD)..... 1283
24.4.41 EPDC Collision Region Size (EPDC_UPD_COL_SIZE)..... 1283
24.4.42 1-level Histogram Parameter Register. (EPDC_HIST1_PARAM)..... 1284
24.4.43 2-level Histogram Parameter Register. (EPDC_HIST2_PARAM)..... 1285
24.4.44 4-level Histogram Parameter Register. (EPDC_HIST4_PARAM)..... 1285
24.4.45 8-level Histogram Parameter 0 Register. (EPDC_HIST8_PARAM0)..... 1286
24.4.46 8-level Histogram Parameter 1 Register. (EPDC_HIST8_PARAM1)..... 1287
24.4.47 16-level Histogram Parameter 0 Register. (EPDC_HIST16_PARAM0)..... 1288
24.4.48 16-level Histogram Parameter Register. (EPDC_HIST16_PARAM1)..... 1288
24.4.49 16-level Histogram Parameter Register. (EPDC_HIST16_PARAM2)..... 1289
24.4.50 16-level Histogram Parameter Register. (EPDC_HIST16_PARAM3)..... 1290
24.4.51 EPDC General Purpose I/O Debug register (EPDC_GPIO*n*).....1291

24.4.52 EPDC Version Register (EPDC_VERSION)..... 1292
24.4.53 Panel Interface Signal Generator Register 0_0 (EPDC_PIGEON_0_0).....1292
24.4.54 Panel Interface Signal Generator Register 0_1 (EPDC_PIGEON_0_1).....1294
24.4.55 Panel Interface Signal Generator Register 0_1 (EPDC_PIGEON_0_2).....1294
24.4.56 Panel Interface Signal Generator Register 1_0 (EPDC_PIGEON_1_0).....1295
24.4.57 Panel Interface Signal Generator Register 1_1 (EPDC_PIGEON_1_1).....1296
24.4.58 Panel Interface Signal Generator Register 1_1 (EPDC_PIGEON_1_2).....1297
24.4.59 Panel Interface Signal Generator Register 2_0 (EPDC_PIGEON_2_0).....1297
24.4.60 Panel Interface Signal Generator Register 2_1 (EPDC_PIGEON_2_1).....1298
24.4.61 Panel Interface Signal Generator Register 2_1 (EPDC_PIGEON_2_2).....1299
24.4.62 Panel Interface Signal Generator Register 3_0 (EPDC_PIGEON_3_0).....1300
24.4.63 Panel Interface Signal Generator Register 3_1 (EPDC_PIGEON_3_1).....1301
24.4.64 Panel Interface Signal Generator Register 3_1 (EPDC_PIGEON_3_2).....1301
24.4.65 Panel Interface Signal Generator Register 4_0 (EPDC_PIGEON_4_0).....1302
24.4.66 Panel Interface Signal Generator Register 4_1 (EPDC_PIGEON_4_1).....1303
24.4.67 Panel Interface Signal Generator Register 4_1 (EPDC_PIGEON_4_2).....1304
24.4.68 Panel Interface Signal Generator Register 5_0 (EPDC_PIGEON_5_0).....1304
24.4.69 Panel Interface Signal Generator Register 5_1 (EPDC_PIGEON_5_1).....1306
24.4.70 Panel Interface Signal Generator Register 5_1 (EPDC_PIGEON_5_2).....1306
24.4.71 Panel Interface Signal Generator Register 6_0 (EPDC_PIGEON_6_0).....1307
24.4.72 Panel Interface Signal Generator Register 6_1 (EPDC_PIGEON_6_1).....1308
24.4.73 Panel Interface Signal Generator Register 6_1 (EPDC_PIGEON_6_2).....1309
24.4.74 Panel Interface Signal Generator Register 7_0 (EPDC_PIGEON_7_0).....1309
24.4.75 Panel Interface Signal Generator Register 7_1 (EPDC_PIGEON_7_1).....1310
24.4.76 Panel Interface Signal Generator Register 7_1 (EPDC_PIGEON_7_2).....1311
24.4.77 Panel Interface Signal Generator Register 8_0 (EPDC_PIGEON_8_0).....1312
24.4.78 Panel Interface Signal Generator Register 8_1 (EPDC_PIGEON_8_1).....1313
24.4.79 Panel Interface Signal Generator Register 8_1 (EPDC_PIGEON_8_2).....1313
24.4.80 Panel Interface Signal Generator Register 9_0 (EPDC_PIGEON_9_0).....1314

24.4.81 Panel Interface Signal Generator Register 9_1 (EPDC_PIGEON_9_1)....1315
24.4.82 Panel Interface Signal Generator Register 9_1 (EPDC_PIGEON_9_2)....1316
24.4.83 Panel Interface Signal Generator Register 10_0 (EPDC_PIGEON_10_0).... 1316
24.4.84 Panel Interface Signal Generator Register 10_1 (EPDC_PIGEON_10_1).... 1318
24.4.85 Panel Interface Signal Generator Register 10_1 (EPDC_PIGEON_10_2).... 1318
24.4.86 Panel Interface Signal Generator Register 11_0 (EPDC_PIGEON_11_0).... 1319
24.4.87 Panel Interface Signal Generator Register 11_1 (EPDC_PIGEON_11_1).... 1320
24.4.88 Panel Interface Signal Generator Register 11_1 (EPDC_PIGEON_11_2).... 1321
24.4.89 Panel Interface Signal Generator Register 12_0 (EPDC_PIGEON_12_0).... 1321
24.4.90 Panel Interface Signal Generator Register 12_1 (EPDC_PIGEON_12_1).... 1322
24.4.91 Panel Interface Signal Generator Register 12_1 (EPDC_PIGEON_12_2).... 1323
24.4.92 Panel Interface Signal Generator Register 13_0 (EPDC_PIGEON_13_0).... 1324
24.4.93 Panel Interface Signal Generator Register 13_1 (EPDC_PIGEON_13_1).... 1325
24.4.94 Panel Interface Signal Generator Register 13_1 (EPDC_PIGEON_13_2).... 1325
24.4.95 Panel Interface Signal Generator Register 14_0 (EPDC_PIGEON_14_0).... 1326
24.4.96 Panel Interface Signal Generator Register 14_1 (EPDC_PIGEON_14_1).... 1327
24.4.97 Panel Interface Signal Generator Register 14_1 (EPDC_PIGEON_14_2).... 1328
24.4.98 Panel Interface Signal Generator Register 15_0 (EPDC_PIGEON_15_0).... 1328
24.4.99 Panel Interface Signal Generator Register 15_1 (EPDC_PIGEON_15_1).... 1330
24.4.100 Panel Interface Signal Generator Register 15_1 (EPDC_PIGEON_15_2).... 1330
24.4.101 Panel Interface Signal Generator Register 16_0 (EPDC_PIGEON_16_0).... 1331
24.4.102 Panel Interface Signal Generator Register 16_1 (EPDC_PIGEON_16_1).... 1332
24.4.103 Panel Interface Signal Generator Register 16_1 (EPDC_PIGEON_16_2).... 1333

Chapter 25
Enhanced Periodic Interrupt Timer (EPIT)

25.1 Overview....1335
25.1.1 EPIT features....1335
25.1.2 EPIT modes and operations.... 1336
25.2 External signals....1336

25.3 Clocks ... 1336
25.4 Functional Description ... 1338
25.4.1 Operating modes ... 1338
25.4.1.1 Operating in set-and-forget mode ... 1338
25.4.1.2 Operating in free-running mode ... 1338
25.4.2 Operations ... 1339
25.4.3 Compare Event ... 1339
25.4.3.1 Counter Value Overwrite ... 1340
25.4.3.2 Low-Power Mode Behavior ... 1341
25.4.3.3 Debug Mode Behavior ... 1341
25.5 Initialization/ Application Information ... 1341
25.5.1 Change of Clock Source ... 1341
25.6 EPIT Memory Map/Register Definition ... 1342
25.6.1 Control register (EPIT*x*_CR) ... 1342
25.6.2 Status register (EPIT*x*_SR) ... 1345
25.6.3 Load register (EPIT*x*_LR) ... 1345
25.6.4 Compare register (EPIT*x*_CMPR) ... 1346
25.6.5 Counter register (EPIT*x*_CNR) ... 1346

Chapter 26
Enhanced Serial Audio Interface (ESAI)

26.1 Overview ... 1347
26.1.1 Features ... 1349
26.1.2 Modes of Operation ... 1349
26.1.2.1 Normal/Network/On-Demand Mode Selection ... 1349
26.1.2.2 Synchronous/Asynchronous Operating Modes ... 1350
26.1.2.3 Frame Sync Selection ... 1350
26.1.2.4 Shift Direction Selection ... 1351
26.2 External Signals ... 1352
26.2.1 Serial Transmit 0 Data Pin ... 1353

26.2.2 Serial Transmit 1 Data Pin ... 1353
26.2.3 Serial Transmit 2/Receive 3 Data Pin ... 1354
26.2.4 Serial Transmit 3/Receive 2 Data Pin ... 1354
26.2.5 Serial Transmit 4/Receive 1 Data Pin ... 1355
26.2.6 Serial Transmit 5/Receive 0 Data Pin ... 1355
26.2.7 Receiver Serial Clock ... 1356
26.2.8 Transmitter Serial Clock ... 1357
26.2.9 Frame Sync for Receiver ... 1358
26.2.10 Frame Sync for Transmitter ... 1359
26.2.11 High Frequency Clock for Transmitter ... 1359
26.2.12 High Frequency Clock for Receiver ... 1360
26.2.13 Serial I/O Flags ... 1360
26.3 Clocks ... 1361
26.4 Functional Description ... 1362
26.4.1 ESAI After Reset ... 1362
26.4.2 ESAI Interrupt Requests ... 1362
26.4.3 ESAI DMA Requests from the FIFOs ... 1363
26.4.4 ESAI Transmit and Receive Shift Registers ... 1364
26.4.4.1 ESAI Transmit Shift Registers ... 1364
26.4.4.2 ESAI Receive Shift Registers ... 1367
26.5 Initialization Information ... 1367
26.5.1 ESAI Initialization ... 1367
26.5.2 ESAI Initialization Examples ... 1368
26.5.2.1 Initializing the ESAI using Personal Reset ... 1368
26.5.2.2 Initializing the ESAI Transmitter Section ... 1369
26.5.2.3 Initializing the ESAI Receiver Section ... 1369
26.6 ESAI Memory Map/Register Definition ... 1370
26.6.1 ESAI Transmit Data Register (ESAI_ETDR) ... 1372
26.6.2 ESAI Receive Data Register (ESAI_ERDR) ... 1372

26.6.3 ESAI Control Register (ESAI_ECR)....1373
26.6.4 ESAI Status Register (ESAI_ESR)....1374
26.6.5 Transmit FIFO Configuration Register (ESAI_TFCR)....1376
26.6.6 Transmit FIFO Status Register (ESAI_TFSR)....1378
26.6.7 Receive FIFO Configuration Register (ESAI_RFCR)....1379
26.6.8 Receive FIFO Status Register (ESAI_RFSR)....1381
26.6.9 Transmit Data Register n (ESAI_TX*n*)....1382
26.6.10 ESAI Transmit Slot Register (ESAI_TSR)....1382
26.6.11 Receive Data Register n (ESAI_RX*n*)....1383
26.6.12 Serial Audio Interface Status Register (ESAI_SAISR)....1384
26.6.13 Serial Audio Interface Control Register (ESAI_SAICR)....1386
26.6.14 Transmit Control Register (ESAI_TCR)....1389
26.6.15 Transmit Clock Control Register (ESAI_TCCR)....1397
26.6.16 Receive Control Register (ESAI_RCR)....1401
26.6.17 Receive Clock Control Register (ESAI_RCCR)....1405
26.6.18 Transmit Slot Mask Register A (ESAI_TSMA)....1408
26.6.19 Transmit Slot Mask Register B (ESAI_TSMB)....1409
26.6.20 Receive Slot Mask Register A (ESAI_RSMA)....1410
26.6.21 Receive Slot Mask Register B (ESAI_RSMB)....1411
26.6.22 Port C Direction Register (ESAI_PRRC)....1412
26.6.23 Port C Control Register (ESAI_PCRC)....1412

Chapter 27
Flexible Controller Area Network (FLEXCAN)

27.1 Overview....1415
27.1.1 Block Diagram....1415
27.1.2 FLEXCAN Module Features....1417
27.1.3 Modes of Operation....1418
27.2 External Signals....1419
27.3 Clocks....1420

27.4 Message Buffer Structure 1420
27.5 Rx FIFO Structure 1424
27.6 Functional Description 1428
27.6.1 Functional Overview 1428
27.6.2 Transmit Process 1428
27.6.3 Arbitration process 1429
27.6.3.1 Lowest Mailbox number first 1430
27.6.3.2 Highest Mailbox priority first 1430
27.6.3.2.1 Local Priority disabled 1430
27.6.3.2.2 Local Priority enabled 1431
27.6.4 Receive Process 1432
27.6.5 Matching Process 1434
27.6.6 Move Process 1438
27.6.6.1 Move-in 1439
27.6.6.2 Move-out 1440
27.6.7 Data Coherence 1440
27.6.7.1 Transmission Abort Mechanism 1440
27.6.7.2 Message Buffer Inactivation 1441
27.6.7.3 Message Buffer Lock Mechanism 1442
27.6.8 Rx FIFO 1443
27.6.9 CAN Protocol Related Features 1445
27.6.9.1 Remote Frames 1445
27.6.9.2 Overload Frames 1446
27.6.9.3 Time Stamp 1446
27.6.9.4 Protocol Timing 1446
27.6.9.5 Arbitration and Matching Timing 1449
27.6.10 Modes of Operation Details 1451
27.6.10.1 Freeze Mode 1451
27.6.10.2 Module Disable Mode 1452

27.6.10.3 Stop Mode ... 1453
27.6.11 Interrupts ... 1454
27.7 Initialization/Application Information ... 1454
27.7.1 FLEXCAN Initialization Sequence ... 1455
27.8 FLEXCAN Memory Map/Register Definition ... 1456
27.8.1 Module Configuration Register (FLEXCAN*x*_MCR) ... 1463
27.8.2 Control 1 Register (FLEXCAN*x*_CTRL1) ... 1468
27.8.3 Free Running Timer Register (FLEXCAN*x*_TIMER) ... 1471
27.8.4 Rx Mailboxes Global Mask Register (FLEXCAN*x*_RXMGMASK) ... 1471
27.8.5 Rx Buffer 14 Mask Register (FLEXCAN*x*_RX14MASK) ... 1472
27.8.6 Rx Buffer 15 Mask Register (FLEXCAN*x*_RX15MASK) ... 1473
27.8.7 Error Counter Register (FLEXCAN*x*_ECR) ... 1474
27.8.8 Error and Status 1 Register (FLEXCAN*x*_ESR1) ... 1475
27.8.9 Interrupt Masks 2 Register (FLEXCAN*x*_IMASK2) ... 1479
27.8.10 Interrupt Masks 1 Register (FLEXCAN*x*_IMASK1) ... 1479
27.8.11 Interrupt Flags 2 Register (FLEXCAN*x*_IFLAG2) ... 1480
27.8.12 Interrupt Flags 1 Register (FLEXCAN*x*_IFLAG1) ... 1480
27.8.13 Control 2 Register (FLEXCAN*x*_CTRL2) ... 1482
27.8.14 Error and Status 2 Register (FLEXCAN*x*_ESR2) ... 1488
27.8.15 CRC Register (FLEXCAN*x*_CRCR) ... 1490
27.8.16 Rx FIFO Global Mask Register (FLEXCAN*x*_RXFGMASK) ... 1491
27.8.17 Rx FIFO Information Register (FLEXCAN*x*_RXFIR) ... 1492
27.8.18 Rx Individual Mask Registers (FLEXCAN*x*_RXIMR*n*) ... 1493
27.8.19 Glitch Filter Width Registers (FLEXCAN*x*_GFWR) ... 1493

Chapter 28
General Power Controller (GPC)

28.1 Overview ... 1495
28.2 Clocks ... 1496
28.3 Power Gating Control (PGC) ... 1496

28.3.1 Overview ... 1497
28.3.1.1 Features ... 1498
28.4 GPC Interrupt Controller (INTC) ... 1498
28.4.1 Interrupt Controller features ... 1498
28.5 GPC Memory Map/Register Definition ... 1499
28.5.1 GPC Interface control register (GPC_CNTR) ... 1499
28.5.2 IRQ masking register 1 (GPC_IMR1) ... 1501
28.5.3 IRQ masking register 2 (GPC_IMR2) ... 1501
28.5.4 IRQ masking register 3 (GPC_IMR3) ... 1501
28.5.5 IRQ masking register 4 (GPC_IMR4) ... 1502
28.5.6 IRQ status resister 1 (GPC_ISR1) ... 1502
28.5.7 IRQ status resister 2 (GPC_ISR2) ... 1503
28.5.8 IRQ status resister 3 (GPC_ISR3) ... 1503
28.5.9 IRQ status resister 4 (GPC_ISR4) ... 1504
28.6 PGC Memory Map/Register Definition ... 1504
28.6.1 PGC GPU PGC Control Register (PGC_GPU_CTRL) ... 1505
28.6.2 PGC GPU Power Up Sequence Control Register (PGC_GPU_PUPSCR) ... 1506
28.6.3 PGC GPU Pull Down Sequence Control Register (PGC_GPU_PDNSCR) ... 1506
28.6.4 PGC GPU Power Gating Controller Status Register (PGC_GPU_SR) ... 1507
28.6.5 PGC CPU Control Register (PGC_CPU_CTRL) ... 1508
28.6.6 PGC CPU Power Up Sequence Control Register (PGC_CPU_PUPSCR) ... 1508
28.6.7 PGC CPU Pull Down Sequence Control Register (PGC_CPU_PDNSCR) ... 1509
28.6.8 PGC CPU Power Gating Controller Status Register (PGC_CPU_SR) ... 1510

Chapter 29
General Purpose Input/Output (GPIO)

29.1 Overview ... 1511
29.1.1 Block Diagram ... 1513
29.1.2 Features ... 1514
29.2 External Signals ... 1514

29.3 Clocks ... 1520
29.4 GPIO Functional Description ... 1521
29.4.1 GPIO Function ... 1521
29.4.2 GPIO pad structure ... 1521
29.4.2.1 Input Driver ... 1522
29.4.2.1.1 Schmitt trigger ... 1522
29.4.2.1.2 Input keeper ... 1523
29.4.2.2 Output driver ... 1523
29.4.2.2.1 Drive strength ... 1524
29.4.2.2.2 Output keeper ... 1524
29.4.2.2.3 PU / PD / Keeper Logic ... 1525
29.4.2.2.4 Open drain ... 1525
29.4.2.3 Operating Frequency ... 1526
29.4.3 GPIO Programming ... 1528
29.4.3.1 GPIO Read Mode ... 1528
29.4.3.2 GPIO Write Mode ... 1528
29.4.4 Interrupt Control Unit ... 1529
29.5 GPIO Memory Map/Register Definition ... 1529
29.5.1 GPIO data register (GPIO*x*_DR) ... 1531
29.5.2 GPIO direction register (GPIO*x*_GDIR) ... 1532
29.5.3 GPIO pad status register (GPIO*x*_PSR) ... 1533
29.5.4 GPIO interrupt configuration register1 (GPIO*x*_ICR1) ... 1533
29.5.5 GPIO interrupt configuration register2 (GPIO*x*_ICR2) ... 1537
29.5.6 GPIO interrupt mask register (GPIO*x*_IMR) ... 1540
29.5.7 GPIO interrupt status register (GPIO*x*_ISR) ... 1541
29.5.8 GPIO edge select register (GPIO*x*_EDGE_SEL) ... 1542

Chapter 30
General Purpose Media Interface (GPMI)

30.1 Overview ... 1543

30.2 External Signals.... 1544
30.3 Clocks.... 1545
30.4 GPMI NAND Mode.... 1546
30.4.1 Multiple NAND Support.... 1546
30.4.2 GPMI NAND Timing and Clocking.... 1547
30.4.3 Basic NAND Timing.... 1547
30.4.3.1 NAND Asynchronous Timing.... 1547
30.4.3.2 NAND Asynchronous EDO Mode Timing.... 1549
30.4.3.3 NAND ONFI Source Synchronous Mode Timing.... 1552
30.4.3.4 NAND Toggle Mode Timing.... 1557
30.4.4 Hardware BCH Interface.... 1565
30.5 Behavior During Reset.... 1566
30.6 GPMI Memory Map/Register Definition.... 1566
30.6.1 GPMI Control Register 0 Description (GPMI_CTRL0*n*).... 1568
30.6.2 GPMI Compare Register Description (GPMI_COMPARE).... 1570
30.6.3 GPMI Integrated ECC Control Register Description (GPMI_ECCCTRL*n*).... 1571
30.6.4 GPMI Integrated ECC Transfer Count Register Description (GPMI_ECCCOUNT).... 1572
30.6.5 GPMI Payload Address Register Description (GPMI_PAYLOAD).... 1572
30.6.6 GPMI Auxiliary Address Register Description (GPMI_AUXILIARY).... 1573
30.6.7 GPMI Control Register 1 Description (GPMI_CTRL1*n*).... 1574
30.6.8 GPMI Timing Register 0 Description (GPMI_TIMING0).... 1577
30.6.9 GPMI Timing Register 1 Description (GPMI_TIMING1).... 1577
30.6.10 GPMI Timing Register 2 Description (GPMI_TIMING2).... 1578
30.6.11 GPMI DMA Data Transfer Register Description (GPMI_DATA).... 1579
30.6.12 GPMI Status Register Description (GPMI_STAT).... 1579
30.6.13 GPMI Debug Information Register Description (GPMI_DEBUG).... 1582
30.6.14 GPMI Version Register Description (GPMI_VERSION).... 1583
30.6.15 GPMI Debug2 Information Register Description (GPMI_DEBUG2).... 1583
30.6.16 GPMI Debug3 Information Register Description (GPMI_DEBUG3).... 1586

30.6.17 GPMI Double Rate Read DLL Control Register Description (GPMI_READ_DDR_DLL_CTRL)..... 1587
30.6.18 GPMI Double Rate Write DLL Control Register Description (GPMI_WRITE_DDR_DLL_CTRL).....1588
30.6.19 GPMI Double Rate Read DLL Status Register Description (GPMI_READ_DDR_DLL_STS)..... 1590
30.6.20 GPMI Double Rate Write DLL Status Register Description (GPMI_WRITE_DDR_DLL_STS)..... 1591

Chapter 31
General Purpose Timer (GPT)

31.1 Overview.....1595
31.1.1 Features..... 1597
31.1.2 Modes and Operation..... 1597
31.2 External Signals..... 1597
31.2.1 External Clock Input 1598
31.2.2 Input Capture Trigger Signals 1598
31.2.3 Output Compare Signals..... 1599
31.3 Clocks..... 1599
31.4 Functional Description.....1601
31.4.1 Operating Modes.....1601
31.4.1.1 Restart Mode..... 1601
31.4.1.2 Free-Run Mode..... 1601
31.4.2 Operation.....1602
31.4.2.1 Input Capture.....1602
31.4.2.2 Output Compare..... 1603
31.4.2.3 Interrupts..... 1604
31.4.2.4 Low Power Mode Behavior..... 1605
31.4.2.5 Debug Mode Behavior..... 1605
31.5 Initialization/ Application Information 1605
31.5.1 Selecting the Clock Source 1605
31.6 GPT Memory Map/Register Definition.....1606
31.6.1 GPT Control Register (GPT_CR)..... 1607
31.6.2 GPT Prescaler Register (GPT_PR)..... 1611

31.6.3 GPT Status Register (GPT_SR)....1612
31.6.4 GPT Interrupt Register (GPT_IR)....1613
31.6.5 GPT Output Compare Register 1 (GPT_OCR1)....1614
31.6.6 GPT Output Compare Register 2 (GPT_OCR2)....1615
31.6.7 GPT Output Compare Register 3 (GPT_OCR3)....1615
31.6.8 GPT Input Capture Register 1 (GPT_ICR1)....1616
31.6.9 GPT Input Capture Register 2 (GPT_ICR2)....1616
31.6.10 GPT Counter Register (GPT_CNT)....1617

Chapter 32
2D Graphics Processing Unit (GPU2D)

32.1 Overview....1619
32.2 GPU2D Block Diagram....1619
32.2.1 R2D GPU....1619
32.3 GPU2D Features....1620
32.3.1 Full Featured R2D GPU Pipeline....1620
32.4 GPU2D OPERATIONS....1621
32.4.1 R2D GPU Operations....1621
32.4.1.1 Line....1621
32.4.1.2 Rectangle Fill and Clear....1621
32.4.1.3 BitBLT....1622
32.4.1.4 Stretch BLT....1623
32.4.1.5 Monochrome Expansion and Mask BLT....1623
32.4.1.5.1 Monochrome expansion....1623
32.4.1.5.2 Mask BLT....1623
32.4.1.6 Filter BLT....1624
32.4.1.7 R2D Performance of different operations....1625
32.4.1.8 Rotation....1626
32.4.1.9 Transparency Mode....1626
32.4.1.10 Clipping....1626

32.4.1.11 R2D GPU Data Formats.... 1626
32.4.1.12 ARGB Data Conversion of R2D GPU.... 1627
32.4.1.13 YUV to RGB Conversion of R2D GPU.... 1627
32.4.1.14 Color Index Input Conversion Support of R2D GPU.... 1628
32.4.1.15 Source/Destination Pre-multiply and De-Multiply Support.... 1628
32.4.1.16 Alpha Blending.... 1628
32.4.1.17 GPU Cache Management.... 1630

Chapter 33
3D Graphics Processing Unit (GPU3D)

33.1 Overview.... 1631
33.2 GPU3D Block Diagram.... 1631
33.3 GPU3D Hardware Features.... 1633
33.3.1 Rasterization.... 1636
33.3.2 Fragment Processing.... 1636
33.3.3 Dest/Alpha Blending.... 1638
33.3.4 Z/Stencil Buffer.... 1638
33.3.5 Render Target.... 1638
33.4 Usage Mode.... 1639

Chapter 34
HDMI Transmitter (HDMI)

34.1 Overview.... 1641
34.1.1 HDMI Operational Model Overview.... 1641
34.1.1.1 Interfaces.... 1643
34.1.1.2 Features.... 1644
34.2 External Signals.... 1645
34.3 Clocks.... 1645
34.4 Functional Description.... 1646
34.4.1 HDMI TX Functional Overview.... 1646
34.4.2 Video Pixel Sampler.... 1647

34.4.2.1 HDMI Transmitter Controller Databook Functional Description....1648
34.4.3 Supported Video Mode....1650
34.4.4 Video Packetizer....1653
34.4.5 Color Space Conversion....1654
34.4.6 Audio Interfaces....1655
34.4.6.1 CTS Calculation....1656
34.4.6.2 Audio DMA Interface....1656
34.4.6.2.1 AHB Master....1657
34.4.6.2.2 DMA Engine....1659
34.4.6.2.2.1 Functional Behavior....1659
34.4.6.2.2.2 DMA Operation....1660
34.4.6.2.2.3 Transfer Data, Package, and Word....1662
34.4.6.2.3 Audio FIFO....1664
34.4.6.2.4 FIFO Occupancy/FIFO Almost Empty Flags....1664
34.4.7 Supported Audio Formats....1665
34.4.8 Frame Composer....1665
34.4.9 HDCP Encryption Engine....1667
34.4.10 EDID/HDCP I2C E-DDC Interface....1668
34.4.10.1 I2C Master Interface Normal Mode....1669
34.4.10.2 I2C Master Interface Extended Read Mode....1669
34.4.11 System Configuration Interfaces....1670
34.4.11.1 AMBA AHB Slave Interface....1670
34.4.12 CEC Hardware Engine....1670
34.5 HDMI Memory Map/Register Definition....1672
34.5.1 Design Identification Register (HDMI_DESIGN_ID)....1688
34.5.2 Revision Identification Register (HDMI_REVISION_ID)....1689
34.5.3 Product Identification Register 0 (HDMI_PRODUCT_ID0)....1690
34.5.4 Product Identification Register 1 (HDMI_PRODUCT_ID1)....1690
34.5.5 Configuration Identification Register 0 (HDMI_CONFIG0_ID)....1691

34.5.6 Configuration Identification Register 1 (HDMI_CONFIG1_ID)..... 1692
34.5.7 Configuration Identification Register 2 (HDMI_CONFIG2_ID)..... 1693
34.5.8 Configuration Identification Register 3 (HDMI_CONFIG3_ID)..... 1693
34.5.9 Frame Composer Interrupt Status Register 0 (HDMI_IH_FC_STAT0)..... 1694
34.5.10 Frame Composer Interrupt Status Register 1 (HDMI_IH_FC_STAT1)..... 1695
34.5.11 Frame Composer Interrupt Status Register 2 (HDMI_IH_FC_STAT2)..... 1696
34.5.12 Audio Sampler Interrupt Status Register (HDMI_IH_AS_STAT0)..... 1697
34.5.13 PHY Interface Interrupt Status Register (HDMI_IH_PHY_STAT0)..... 1698
34.5.14 E-DDC I2C Master Interrupt Status Register (HDMI_IH_I2CM_STAT0)..... 1699
34.5.15 CEC Interrupt Status Register (HDMI_IH_CEC_STAT0)..... 1700
34.5.16 Video Packetizer Interrupt Status Register (HDMI_IH_VP_STAT0)..... 1701
34.5.17 PHY GEN2 I2C Master Interrupt Status Register (HDMI_IH_I2CMPHY_STAT0)..... 1702
34.5.18 AHB Audio DMA Interrupt Status Register (HDMI_IH_AHBDMAAUD_STAT0)..... 1702
34.5.19 Frame Composer Interrupt Mute Control Register 0 (HDMI_IH_MUTE_FC_STAT0)..... 1704
34.5.20 Frame Composer Interrupt Mute Control Register 1 (HDMI_IH_MUTE_FC_STAT1)..... 1705
34.5.21 Frame Composer Interrupt Mute Control Register 2 (HDMI_IH_MUTE_FC_STAT2)..... 1706
34.5.22 Audio Sampler Interrupt Mute Control Register 0 (HDMI_IH_MUTE_AS_STAT0)..... 1706
34.5.23 PHY Interface Interrupt Mute Control Register (HDMI_IH_MUTE_PHY_STAT0)..... 1707
34.5.24 E-DDC I2C Master Interrupt Mute Control Register (HDMI_IH_MUTE_I2CM_STAT0)..... 1708
34.5.25 CEC Interrupt Mute Control Register (HDMI_IH_MUTE_CEC_STAT0)..... 1708
34.5.26 Video Packetizer Interrupt Mute Control Register (HDMI_IH_MUTE_VP_STAT0)..... 1709
34.5.27 PHY GEN 2 I2C Master Interrupt Mute Control Register (HDMI_IH_MUTE_I2CMPHY_STAT0)..... 1710
34.5.28 AHB Audio DMA Interrupt Mute Control Register (HDMI_IH_MUTE_AHBDMAAUD_STAT0)..... 1711
34.5.29 Global Interrupt Mute Control Register (HDMI_IH_MUTE)..... 1712
34.5.30 Video Input Mapping and Internal Data Enable Configuration Register (HDMI_TX_INVID0)..... 1712
34.5.31 Video Input Stuffing Enable Register (HDMI_TX_INSTUFFING)..... 1713
34.5.32 Video Input GY Data Channel Stuffing Register 0 (HDMI_TX_GYDATA0)..... 1714
34.5.33 Video Input GY Data Channel Stuffing Register 1 (HDMI_TX_GYDATA1)..... 1715
34.5.34 Video Input RCR Data Channel Stuffing Register 0 (HDMI_TX_RCRDATA0)..... 1715

34.5.35 Video Input RCR Data Channel Stuffing Register 1 (HDMI_TX_RCRDATA1)..... 1716
34.5.36 Video Input RCB Data Channel Stuffing Register 0 (HDMI_TX_BCBDATA0)..... 1716
34.5.37 Video Input RCB Data Channel Stuffing Register 1 (HDMI_TX_BCBDATA1)..... 1717
34.5.38 Video Packetizer Packing Phase Status Register (HDMI_VP_STATUS)..... 1717
34.5.39 Video Packetizer Pixel Repetition and Color Depth Register (HDMI_VP_PR_CD)..... 1718
34.5.40 Video Packetizer Stuffing and Default Packing Phase Register (HDMI_VP_STUFF)..... 1719
34.5.41 Video Packetizer YCC422 Remapping Register (HDMI_VP_REMAP)..... 1720
34.5.42 Video Packetizer Output, Bypass, and Enable Configuration Register (HDMI_VP_CONF)..... 1721
34.5.43 Video Packetizer Interrupt Mask Register (HDMI_VP_MASK)..... 1721
34.5.44 Frame Composer Input Video Configuration and HDCP Keepout Register (HDMI_FC_INVIDCONF) 1722
34.5.45 Frame Composer Input Video HActive Pixels Register 0 (HDMI_FC_INHACTIV0)..... 1724
34.5.46 Frame Composer Input Video HActive Pixels Register 1 (HDMI_FC_INHACTIV1)..... 1724
34.5.47 Frame Composer Input Video HBlank Pixels Register 0 (HDMI_FC_INHBLANK0)..... 1725
34.5.48 Frame Composer Input Video HBlank Pixels Register 1 (HDMI_FC_INHBLANK1)..... 1725
34.5.49 Frame Composer Input Video VActive Pixels Register 0 (HDMI_FC_INVACTIV0)..... 1726
34.5.50 Frame Composer Input Video VActive Pixels Register 1 (HDMI_FC_INVACTIV1)..... 1727
34.5.51 Frame Composer Input Video VBlank Pixels Register (HDMI_FC_INVBLANK)..... 1727
34.5.52 Frame Composer Input Video HSync Front Porch Register 0 (HDMI_FC_HSYNCINDELAY0)..... 1728
34.5.53 Frame Composer Input Video HSync Front Porch Register 1 (HDMI_FC_HSYNCINDELAY1)..... 1728
34.5.54 Frame Composer Input Video HSync Width Register 0 (HDMI_FC_HSYNCINWIDTH0)..... 1729
34.5.55 Frame Composer Input Video HSync Width Register 1 (HDMI_FC_HSYNCINWIDTH1)..... 1730
34.5.56 Frame Composer Input Video VSync Front Porch Register (HDMI_FC_VSYNCINDELAY)..... 1730
34.5.57 Frame Composer Input Video VSync Width Register (HDMI_FC_VSYNCINWIDTH)..... 1731
34.5.58 Frame Composer Input Video Refresh Rate Register 0 (HDMI_FC_INFREQ0)..... 1731
34.5.59 Frame Composer Input Video Refresh Rate Register 1 (HDMI_FC_INFREQ1)..... 1732
34.5.60 Frame Composer Input Video Refresh Rate Register 2 (HDMI_FC_INFREQ2)..... 1732
34.5.61 Frame Composer Control Period Duration Register (HDMI_FC_CTRLDUR)..... 1733
34.5.62 Frame Composer Extended Control Period Duration Register (HDMI_FC_EXCTRLDUR)..... 1734
34.5.63 Frame Composer Extended Control Period Maximum Spacing Register (HDMI_FC_EXCTRLSPAC).1734

34.5.64 Frame Composer Channel 0 Non-Preamble Data Register (HDMI_FC_CH0PREAM)..... 1735
34.5.65 Frame Composer Channel 1 Non-Preamble Data Register (HDMI_FC_CH1PREAM)..... 1735
34.5.66 Frame Composer Channel 2 Non-Preamble Data Register (HDMI_FC_CH2PREAM)..... 1736
34.5.67 Frame Composer AVI Configuration Register 3 (HDMI_FC_AVICONF3)..... 1736
34.5.68 Frame Composer GCP Packet Configuration Register (HDMI_FC_GCP)..... 1737
34.5.69 Frame Composer AVI Packet Configuration Register 0 (HDMI_FC_AVICONF0)..... 1737
34.5.70 Frame Composer AVI Packet Configuration Register 1 (HDMI_FC_AVICONF1)..... 1739
34.5.71 FC_AVICONFFrame Composer AVI Packet Configuration Register 2 (HDMI_FC_AVICONF2)..... 1739
34.5.72 Frame Composer AVI Packet VIC Register (HDMI_FC_AVIVID)..... 1740
34.5.73 Frame Composer AVI Packet End of Top Bar Register 0 (HDMI_FC_AVIETB0)..... 1740
34.5.74 Frame Composer AVI Packet End of Top Bar Register 1 (HDMI_FC_AVIETB1)..... 1741
34.5.75 Frame Composer AVI Packet Start of Bottom Bar Register 0 (HDMI_FC_AVISBB0)..... 1741
34.5.76 Frame Composer AVI Packet Start of Bottom Bar Register 1 (HDMI_FC_AVISBB1)..... 1742
34.5.77 Frame Composer AVI Packet End of Left Bar Register 0 (HDMI_FC_AVIELB0)..... 1742
34.5.78 Frame Composer AVI Packet End of Left Bar Register 1 (HDMI_FC_AVIELB1)..... 1743
34.5.79 Frame Composer AVI Packet Start of Right Bar Register 0 (HDMI_FC_AVISRB0)..... 1743
34.5.80 Frame Composer AVI Packet Start of Right Bar Register 1 (HDMI_FC_AVISRB1)..... 1744
34.5.81 Frame Composer AUD Packet Configuration Register 0 (HDMI_FC_AUDICONF0)..... 1744
34.5.82 Frame Composer AUD Packet Configuration Register 1 (HDMI_FC_AUDICONF1)..... 1745
34.5.83 Frame Composer AUD Packet Configuration Register 2 (HDMI_FC_AUDICONF2)..... 1746
34.5.84 Frame Composer AUD Packet Configuration Register 3 (HDMI_FC_AUDICONF3)..... 1746
34.5.85 Frame Composer VSI Packet Data IEEE Register 0 (HDMI_FC_VSDIEEEID0)..... 1746
34.5.86 Frame Composer VSI Packet Data Size Register (HDMI_FC_VSDSIZE)..... 1747
34.5.87 Frame Composer VSI Packet Data IEEE Register 1 (HDMI_FC_VSDIEEEID1)..... 1748
34.5.88 Frame Composer VSI Packet Data IEEE Register 2 (HDMI_FC_VSDIEEEID2)..... 1748
34.5.89 Frame Composer VSI Packet Data IEEE Register 0 (HDMI_FC_VSDPAYLOAD0)..... 1749
34.5.90 Frame Composer VSI Packet Data IEEE Register 1 (HDMI_FC_VSDPAYLOAD1)..... 1749
34.5.91 Frame Composer VSI Packet Data IEEE Register 2 (HDMI_FC_VSDPAYLOAD2)..... 1750
34.5.92 Frame Composer VSI Packet Data IEEE Register 3 (HDMI_FC_VSDPAYLOAD3)..... 1750

34.5.93 Frame Composer VSI Packet Data IEEE Register 4 (HDMI_FC_VSDPAYLOAD4)..... 1751
34.5.94 Frame Composer VSI Packet Data IEEE Register 5 (HDMI_FC_VSDPAYLOAD5)..... 1751
34.5.95 Frame Composer VSI Packet Data IEEE Register 6 (HDMI_FC_VSDPAYLOAD6)..... 1752
34.5.96 Frame Composer VSI Packet Data IEEE Register 7 (HDMI_FC_VSDPAYLOAD7)..... 1752
34.5.97 Frame Composer VSI Packet Data IEEE Register 8 (HDMI_FC_VSDPAYLOAD8)..... 1753
34.5.98 Frame Composer VSI Packet Data IEEE Register 9 (HDMI_FC_VSDPAYLOAD9)..... 1753
34.5.99 Frame Composer VSI Packet Data IEEE Register 10 (HDMI_FC_VSDPAYLOAD10)..... 1754
34.5.100 Frame Composer VSI Packet Data IEEE Register 11 (HDMI_FC_VSDPAYLOAD11)..... 1754
34.5.101 Frame Composer VSI Packet Data IEEE Register 12 (HDMI_FC_VSDPAYLOAD12)..... 1755
34.5.102 Frame Composer VSI Packet Data IEEE Register 13 (HDMI_FC_VSDPAYLOAD13)..... 1755
34.5.103 Frame Composer VSI Packet Data IEEE Register 14 (HDMI_FC_VSDPAYLOAD14)..... 1756
34.5.104 Frame Composer VSI Packet Data IEEE Register 15 (HDMI_FC_VSDPAYLOAD15)..... 1756
34.5.105 Frame Composer VSI Packet Data IEEE Register 16 (HDMI_FC_VSDPAYLOAD16)..... 1757
34.5.106 Frame Composer VSI Packet Data IEEE Register 17 (HDMI_FC_VSDPAYLOAD17)..... 1757
34.5.107 Frame Composer VSI Packet Data IEEE Register 18 (HDMI_FC_VSDPAYLOAD18)..... 1758
34.5.108 Frame Composer VSI Packet Data IEEE Register 19 (HDMI_FC_VSDPAYLOAD19)..... 1758
34.5.109 Frame Composer VSI Packet Data IEEE Register 20 (HDMI_FC_VSDPAYLOAD20)..... 1759
34.5.110 Frame Composer VSI Packet Data IEEE Register 21 (HDMI_FC_VSDPAYLOAD21)..... 1759
34.5.111 Frame Composer VSI Packet Data IEEE Register 22 (HDMI_FC_VSDPAYLOAD22)..... 1760
34.5.112 Frame Composer VSI Packet Data IEEE Register 23 (HDMI_FC_VSDPAYLOAD23)..... 1760
34.5.113 Frame Composer SPD Packet Data Vendor Name Register 0 (HDMI_FC_SPDVENDORNAME0).....1761
34.5.114 Frame Composer SPD Packet Data Product Name Register 0 (HDMI_FC_SPDPRODUCTNAME0)... 1761
34.5.115 Frame Composer SPD Packet Data Source Product Descriptor Register (HDMI_FC_SPDDEVICEINF)..... 1762
34.5.116 Frame Composer Audio Sample Flat and Layout Configuration Register (HDMI_FC_AUDSCONF)... 1762
34.5.117 Frame Composer Audio Packet Sample Present Status Register (HDMI_FC_AUDSSTAT)..... 1763
34.5.118 Frame Composer Number of High Priority Packets Attended Configuration Register (HDMI_FC_CTRLQHIGH)..... 1764
34.5.119 Frame Composer Number of Low Priority Packets Attended Configuration Register (HDMI_FC_CTRLQLOW)..... 1764

34.5.120 Frame Composer ACP Packet Type Configuration Register 0 (HDMI_FC_ACP0).....1765
34.5.121 Frame Composer ACP Packet Type Configuration Register 1 (HDMI_FC_ACP1).....1765
34.5.122 FC_ISCR1_Frame Composer Packet Status, Valid, and Continue Configuration Register (HDMI_FC_ISCR1_0).....1766
34.5.123 Frame Composer ISCR1 Packet Body Register 1 (HDMI_FC_ISCR1_1)..... 1766
34.5.124 Frame Composer ISCR2 Packet Body Register 0 (HDMI_FC_ISCR2_0)..... 1767
34.5.125 Frame Composer Data Island Auto Packet Scheduling Register 0 (HDMI_FC_DATAUTO0)..... 1768
34.5.126 Frame Composer Data Island Auto Packet Scheduling Register 1 (HDMI_FC_DATAUTO1)..... 1768
34.5.127 Frame Composer Data Island Auto Packet Scheduling Register 2 (HDMI_FC_DATAUTO2)..... 1769
34.5.128 Frame Composer Data Island Manual Packet Request Register (HDMI_FC_DATMAN).....1770
34.5.129 Frame Composer Data Island Auto Packet Scheduling Register 3 (HDMI_FC_DATAUTO3)..... 1770
34.5.130 Frame Composer Round Robin ACR Packet Insertion Register 0 (HDMI_FC_RDRB0)..... 1771
34.5.131 Frame Composer Round Robin ACR Packet Insertion Register 1 (HDMI_FC_RDRB1)..... 1772
34.5.132 Frame Composer Round Robin ACR Packet Insertion Register 2 (HDMI_FC_RDRB2)..... 1772
34.5.133 Frame Composer Round Robin ACR Packet Insertion Register 3 (HDMI_FC_RDRB3)..... 1773
34.5.134 Frame Composer Round Robin ACR Packet Insertion Register 4 (HDMI_FC_RDRB4)..... 1773
34.5.135 Frame Composer Round Robin ACR Packet Insertion Register 5 (HDMI_FC_RDRB5)..... 1774
34.5.136 Frame Composer Round Robin ACR Packet Insertion Register 6 (HDMI_FC_RDRB6)..... 1775
34.5.137 Frame Composer Round Robin ACR Packet Insertion Register 7 (HDMI_FC_RDRB7)..... 1775
34.5.138 Frame Composer Packet Interrupt Mask Register 0 (HDMI_FC_MASK0)..... 1776
34.5.139 Frame Composer Packet Interrupt Mask Register 1 (HDMI_FC_MASK1)..... 1777
34.5.140 Frame Composer Pixel Repetition Configuration Register (HDMI_FC_PRCONF).....1777
34.5.141 Frame Composer GMD Packet Status Register (HDMI_FC_GMD_STAT)..... 1779
34.5.142 Frame Composer GMD Packet Enable Register (HDMI_FC_GMD_EN).....1780
34.5.143 Frame Composer GMD Packet Update Register (HDMI_FC_GMD_UP)..... 1780
34.5.144 Frame Composer GMD Packet Schedule Configuration Register (HDMI_FC_GMD_CONF)..... 1781
34.5.145 Frame Composer GMD Packet Profile and Gamut Sequence Configuration Register (HDMI_FC_GMD_HB).....1782
34.5.146 Frame Composer GMD Packet Body Register 0 (HDMI_FC_GMD_PB0).....1782

34.5.147 Frame Composer GMD Packet Body Register 1 (HDMI_FC_GMD_PB1).....1783
34.5.148 Frame Composer GMD Packet Body Register 2 (HDMI_FC_GMD_PB2).....1783
34.5.149 Frame Composer GMD Packet Body Register 3 (HDMI_FC_GMD_PB3).....1784
34.5.150 Frame Composer GMD Packet Body Register 4 (HDMI_FC_GMD_PB4).....1784
34.5.151 Frame Composer GMD Packet Body Register 5 (HDMI_FC_GMD_PB5).....1785
34.5.152 Frame Composer GMD Packet Body Register 6 (HDMI_FC_GMD_PB6).....1785
34.5.153 Frame Composer GMD Packet Body Register 7 (HDMI_FC_GMD_PB7).....1786
34.5.154 Frame Composer GMD Packet Body Register 8 (HDMI_FC_GMD_PB8).....1786
34.5.155 Frame Composer GMD Packet Body Register 9 (HDMI_FC_GMD_PB9).....1787
34.5.156 Frame Composer GMD Packet Body Register 10 (HDMI_FC_GMD_PB10).....1787
34.5.157 Frame Composer GMD Packet Body Register 11 (HDMI_FC_GMD_PB11).....1788
34.5.158 Frame Composer GMD Packet Body Register 12 (HDMI_FC_GMD_PB12).....1788
34.5.159 Frame Composer GMD Packet Body Register 13 (HDMI_FC_GMD_PB13).....1789
34.5.160 Frame Composer GMD Packet Body Register 14 (HDMI_FC_GMD_PB14).....1789
34.5.161 Frame Composer GMD Packet Body Register 15 (HDMI_FC_GMD_PB15).....1790
34.5.162 Frame Composer GMD Packet Body Register 16 (HDMI_FC_GMD_PB16).....1790
34.5.163 Frame Composer GMD Packet Body Register 17 (HDMI_FC_GMD_PB17).....1791
34.5.164 Frame Composer GMD Packet Body Register 18 (HDMI_FC_GMD_PB18).....1791
34.5.165 Frame Composer GMD Packet Body Register 19 (HDMI_FC_GMD_PB19).....1792
34.5.166 Frame Composer GMD Packet Body Register 20 (HDMI_FC_GMD_PB20).....1792
34.5.167 Frame Composer GMD Packet Body Register 21 (HDMI_FC_GMD_PB21).....1793
34.5.168 Frame Composer GMD Packet Body Register 22 (HDMI_FC_GMD_PB22).....1793
34.5.169 Frame Composer GMD Packet Body Register 23 (HDMI_FC_GMD_PB23).....1794
34.5.170 Frame Composer GMD Packet Body Register 24 (HDMI_FC_GMD_PB24).....1794
34.5.171 Frame Composer GMD Packet Body Register 25 (HDMI_FC_GMD_PB25).....1795
34.5.172 Frame Composer GMD Packet Body Register 26 (HDMI_FC_GMD_PB26).....1795
34.5.173 Frame Composer GMD Packet Body Register 27 (HDMI_FC_GMD_PB27).....1796
34.5.174 Frame Composer Video/Audio Force Enable Register (HDMI_FC_DBGFORCE)..... 1796
34.5.175 Frame Composer Audio Channel 0 Register 0 (HDMI_FC_DBGAUD0CH0)..... 1797

34.5.176 Frame Composer Audio Channel 0 Register 1 (HDMI_FC_DBGAUD1CH0)..... 1798
34.5.177 Frame Composer Audio Channel 0 Register 2 (HDMI_FC_DBGAUD2CH0)..... 1798
34.5.178 Frame Composer Audio Channel 1 Register 0 (HDMI_FC_DBGAUD0CH1)..... 1799
34.5.179 Frame Composer Audio Channel 1 Register 1 (HDMI_FC_DBGAUD1CH1)..... 1799
34.5.180 Frame Composer Audio Channel 1 Register 2 (HDMI_FC_DBGAUD2CH1)..... 1800
34.5.181 Frame Composer Debug Audio Channel 2 Register 0 (HDMI_FC_DBGAUD0CH2).....1800
34.5.182 Frame Composer Debug Audio Channel 2 Register 1 (HDMI_FC_DBGAUD1CH2).....1801
34.5.183 Frame Composer Audio Channel 2 Register 2 (HDMI_FC_DBGAUD2CH2)..... 1801
34.5.184 Frame Composer Audio Channel 3 Register 0 (HDMI_FC_DBGAUD0CH3)..... 1802
34.5.185 Frame Composer Audio Channel 3 Register 1 (HDMI_FC_DBGAUD1CH3)..... 1802
34.5.186 Frame Composer Audio Channel 3 Register 2 (HDMI_FC_DBGAUD2CH3)..... 1803
34.5.187 Frame Composer Audio Channel 4 Register 0 (HDMI_FC_DBGAUD0CH4)..... 1803
34.5.188 Frame Composer Audio Channel 4 Register 1 (HDMI_FC_DBGAUD1CH4)..... 1804
34.5.189 Frame Composer Audio Channel 4 Register 2 (HDMI_FC_DBGAUD2CH4)..... 1804
34.5.190 Frame Composer Audio Channel 5 Register 0 (HDMI_FC_DBGAUD0CH5)..... 1805
34.5.191 Frame Composer Audio Channel 5 Register 1 (HDMI_FC_DBGAUD1CH5)..... 1805
34.5.192 Frame Composer Audio Channel 5 Register 2 (HDMI_FC_DBGAUD2CH5)..... 1806
34.5.193 Frame Composer Audio Channel 6 Register 0 (HDMI_FC_DBGAUD0CH6)..... 1806
34.5.194 Frame Composer Audio Channel 6 Register 1 (HDMI_FC_DBGAUD1CH6)..... 1807
34.5.195 Frame Composer Audio Channel 6 Register 2 (HDMI_FC_DBGAUD2CH6)..... 1807
34.5.196 Frame Composer Audio Channel 7 Register 1 (HDMI_FC_DBGAUD0CH7)..... 1808
34.5.197 Frame Composer Audio Channel 7 Register 0 (HDMI_FC_DBGAUD1CH7)..... 1808
34.5.198 Frame Composer Audio Channel 7 Register 2 (HDMI_FC_DBGAUD2CH7)..... 1809
34.5.199 Frame Composer TMDS Channel 0 Register (HDMI_FC_DBGTMDS0)..... 1809
34.5.200 Frame Composer TMDS Channel 1 Register (HDMI_FC_DBGTMDS1)..... 1810
34.5.201 Frame Composer TMDS Channel 2 Register (HDMI_FC_DBGTMDS2)..... 1810
34.5.202 PHY Configuration Register (HDMI_PHY_CONF0).....1811
34.5.203 PHY Test Interface Register 0 (HDMI_PHY_TST0).....1812
34.5.204 PHY Test Interface Register 1 (HDMI_PHY_TST1).....1812

34.5.205 PHY Test Interface Register 2 (HDMI_PHY_TST2)....1813
34.5.206 PHY RXSENSE, PLL lock, and HPD Status Register (HDMI_PHY_STAT0)....1813
34.5.207 PHY RXSENSE, PLL lock, and HPD Interrupt Register (HDMI_PHY_INT0)....1814
34.5.208 PHY RXSENSE, PLL lock, and HPD Mask Register (HDMI_PHY_MASK0)....1815
34.5.209 PHY RXSENSE, PLL lock and HPD Polarity Register (HDMI_PHY_POL0)....1816
34.5.210 PHY I2C Slave Address Configuration Register (HDMI_PHY_I2CM_SLAVE_ADDR)....1817
34.5.211 PHY I2C Address Configuration Register (HDMI_PHY_I2CM_ADDRESS_ADDR)....1817
34.5.212 PHY I2C Data Write Register 1 (HDMI_PHY_I2CM_DATAO_1_ADDR)....1818
34.5.213 PHY I2C Data Write Register 0 (HDMI_PHY_I2CM_DATAO_0_ADDR)....1819
34.5.214 PHY I2C Data Read Register 1 (HDMI_PHY_I2CM_DATAI_1_ADDR)....1819
34.5.215 PHY I2C Data Read Register 0 (HDMI_PHY_I2CM_DATAI_0_ADDR)....1820
34.5.216 PHY I2C Read/Write Operation (HDMI_PHY_I2CM_OPERATION_ADDR)....1820
34.5.217 PHY I2C Done Interrupt Register (HDMI_PHY_I2CM_INT_ADDR)....1821
34.5.218 PHY I2C Done Interrupt Register (HDMI_PHY_I2CM_CTLINT_ADDR)....1822
34.5.219 PHY I2C Speed Control Register (HDMI_PHY_I2CM_DIV_ADDR)....1823
34.5.220 PHY I2C Software Reset Register (HDMI_PHY_I2CM_SOFTRSTZ_ADDR)....1823
34.5.221 PHY I2C Slow Speed SCL High Level Control Register 1 (HDMI_PHY_I2CM_SS_SCL_HCNT_1_ADDR)....1824
34.5.222 PHY I2C Slow Speed SCL High Level Control Register 0 (HDMI_PHY_I2CM_SS_SCL_HCNT_0_ADDR)....1825
34.5.223 PHY I2C Slow Speed SCL Low Level Control Register 1 (HDMI_PHY_I2CM_SS_SCL_LCNT_1_ADDR)....1825
34.5.224 PHY I2C Slow Speed SCL Low Level Control Register 0 (HDMI_PHY_I2CM_SS_SCL_LCNT_0_ADDR)....1826
34.5.225 PHY I2C Fast Speed SCL High Level Control Register 1 (HDMI_PHY_I2CM_FS_SCL_HCNT_1_ADDR)....1826
34.5.226 PHY I2C Fast Speed SCL High Level Control Register 0 (HDMI_PHY_I2CM_FS_SCL_HCNT_0_ADDR)....1827
34.5.227 PHY I2C Fast Speed SCL Low Level Control Register 1 (HDMI_PHY_I2CM_FS_SCL_LCNT_1_ADDR)....1827

34.5.228 PHY I2C Fast Speed SCL Low Level Control Register 0 (HDMI_PHY_I2CM_FS_SCL_LCNT_0_ADDR)..... 1828
34.5.229 Audio Clock Regenerator N Value Register 1 (HDMI_AUD_N1)..... 1828
34.5.230 Audio Clock Regenerator N Value Register 2 (HDMI_AUD_N2)..... 1829
34.5.231 Audio Clock Regenerator N Value Register 3 (HDMI_AUD_N3)..... 1829
34.5.232 AUD_CTS1 (HDMI_AUD_CTS1)..... 1830
34.5.233 AUD_CTS2 (HDMI_AUD_CTS2)..... 1830
34.5.234 AUD_CTS3 (HDMI_AUD_CTS3)..... 1831
34.5.235 Audio DMA Start Register (HDMI_AHB_DMA_CONF0)..... 1831
34.5.236 AHB_DMA_START (HDMI_AHB_DMA_START)..... 1832
34.5.237 Audio DMA Stop Register (HDMI_AHB_DMA_STOP)..... 1833
34.5.238 Audio DMA FIFO Threshold Register (HDMI_AHB_DMA_THRSLD)..... 1834
34.5.239 Audio DMA Start Address Register 0 (HDMI_AHB_DMA_STRADDR0)..... 1834
34.5.240 Audio DMA Start Address Register 1 (HDMI_AHB_DMA_STRADDR1)..... 1835
34.5.241 Audio DMA Start Address Register 2 (HDMI_AHB_DMA_STRADDR2)..... 1835
34.5.242 Audio DMA Start Address Register 3 (HDMI_AHB_DMA_STRADDR3)..... 1836
34.5.243 Audio DMA Stop Address Register 0 (HDMI_AHB_DMA_STPADDR0)..... 1836
34.5.244 Audio DMA Stop Address Register 1 (HDMI_AHB_DMA_STPADDR1)..... 1837
34.5.245 Audio DMA Stop Address Register 2 (HDMI_AHB_DMA_STPADDR2)..... 1837
34.5.246 Audio DMA Stop Address Register 3 (HDMI_AHB_DMA_STPADDR3)..... 1838
34.5.247 Audio DMA Burst Start Address Register 0 (HDMI_AHB_DMA_BSTADDR0)..... 1838
34.5.248 Audio DMA Burst Start Address Register 1 (HDMI_AHB_DMA_BSTADDR1)..... 1839
34.5.249 Audio DMA Burst Start Address Register 2 (HDMI_AHB_DMA_BSTADDR2)..... 1839
34.5.250 Audio DMA Burst Start Address Register 3 (HDMI_AHB_DMA_BSTADDR3)..... 1839
34.5.251 Audio DMA Burst Length Register 0 (HDMI_AHB_DMA_MBLENGTH0)..... 1840
34.5.252 Audio DMA Burst Length Register 1 (HDMI_AHB_DMA_MBLENGTH1)..... 1841
34.5.253 Audio DMA Interrupt Status Register (HDMI_AHB_DMA_STAT)..... 1841
34.5.254 Audio DMA Interrupt Register (HDMI_AHB_DMA_INT)..... 1842
34.5.255 Audio DMA Mask Interrupt Register (HDMI_AHB_DMA_MASK)..... 1843

34.5.256 Audio DMA Polarity Interrupt Register (HDMI_AHB_DMA_POL)....1844
34.5.257 Audio DMA Channel Enable Configuration Register 1 (HDMI_AHB_DMA_CONF1)....1845
34.5.258 Audio DMA Buffer Interrupt Status Register (HDMI_AHB_DMA_BUFFSTAT)....1846
34.5.259 Audio DMA Buffer Interrupt Register (HDMI_AHB_DMA_BUFFINT)....1847
34.5.260 Audio DMA Buffer Mask Interrupt Register (HDMI_AHB_DMA_BUFFMASK)....1848
34.5.261 Audio DMA Buffer Polarity Interrupt Register (HDMI_AHB_DMA_BUFFPOL)....1848
34.5.262 Main Controller Synchronous Clock Domain Disable Register (HDMI_MC_CLKDIS)....1849
34.5.263 Main Controller Software Reset Register (HDMI_MC_SWRSTZREQ)....1850
34.5.264 Main Controller Feed Through Control Register (HDMI_MC_FLOWCTRL)....1851
34.5.265 Main Controller PHY Reset Register (HDMI_MC_PHYRSTZ)....1851
34.5.266 Main Controller Clock Present Register (HDMI_MC_LOCKONCLOCK)....1852
34.5.267 Main Controller HEAC PHY Reset Register (HDMI_MC_HEACPHY_RST)....1853
34.5.268 Color Space Converter Interpolation and Decimation Configuration Register (HDMI_CSC_CFG)....1853
34.5.269 Color Space Converter Scale and Deep Color Configuration Register (HDMI_CSC_SCALE)....1854
34.5.270 CSC_COEF_A1_MSB (HDMI_CSC_COEF_A1_MSB)....1855
34.5.271 CSC_COEF_A1_LSB (HDMI_CSC_COEF_A1_LSB)....1855
34.5.272 CSC_COEF_A2_MSB (HDMI_CSC_COEF_A2_MSB)....1856
34.5.273 CSC_COEF_A2_LSB (HDMI_CSC_COEF_A2_LSB)....1856
34.5.274 CSC_COEF_A3_MSB (HDMI_CSC_COEF_A3_MSB)....1857
34.5.275 CSC_COEF_A3_LSB (HDMI_CSC_COEF_A3_LSB)....1857
34.5.276 CSC_COEF_A4_MSB (HDMI_CSC_COEF_A4_MSB)....1858
34.5.277 CSC_COEF_A4_LSB (HDMI_CSC_COEF_A4_LSB)....1858
34.5.278 CSC_COEF_B1_MSB (HDMI_CSC_COEF_B1_MSB)....1859
34.5.279 CSC_COEF_B1_LSB (HDMI_CSC_COEF_B1_LSB)....1859
34.5.280 CSC_COEF_B2_MSB (HDMI_CSC_COEF_B2_MSB)....1860
34.5.281 CSC_COEF_B2_LSB (HDMI_CSC_COEF_B2_LSB)....1860
34.5.282 CSC_COEF_B3_MSB (HDMI_CSC_COEF_B3_MSB)....1861
34.5.283 CSC_COEF_B3_LSB (HDMI_CSC_COEF_B3_LSB)....1861
34.5.284 CSC_COEF_B4_MSB (HDMI_CSC_COEF_B4_MSB)....1862

34.5.285 CSC_COEF_B4_LSB (HDMI_CSC_COEF_B4_LSB)....1862
34.5.286 CSC_COEF_C1_MSB (HDMI_CSC_COEF_C1_MSB)....1863
34.5.287 CSC_COEF_C1_LSB (HDMI_CSC_COEF_C1_LSB)....1863
34.5.288 CSC_COEF_C2_MSB (HDMI_CSC_COEF_C2_MSB)....1864
34.5.289 CSC_COEF_C2_LSB (HDMI_CSC_COEF_C2_LSB)....1864
34.5.290 CSC_COEF_C3_MSB (HDMI_CSC_COEF_C3_MSB)....1865
34.5.291 CSC_COEF_C3_LSB (HDMI_CSC_COEF_C3_LSB)....1865
34.5.292 CSC_COEFC4_MSB (HDMI_CSC_COEFC4_MSB)....1866
34.5.293 CSC_COEFC4_LSB (HDMI_CSC_COEFC4_LSB)....1866
34.5.294 CEC_CTRL (HDMI_CEC_CTRL)....1867
34.5.295 CEC_MASK (HDMI_CEC_MASK)....1868
34.5.296 CEC_ADDR_L (HDMI_CEC_ADDR_L)....1869
34.5.297 CEC_ADDR_H (HDMI_CEC_ADDR_H)....1870
34.5.298 CEC_TX_CNT (HDMI_CEC_TX_CNT)....1871
34.5.299 CEC_RX_CNT (HDMI_CEC_RX_CNT)....1872
34.5.300 CEC_TX_DATA (HDMI_CEC_TX_DATA*n*)....1873
34.5.301 CEC_RX_DATA (HDMI_CEC_RX_DATA*n*)....1873
34.5.302 CEC_LOCK (HDMI_CEC_LOCK)....1874
34.5.303 CEC_WKUPCTRL (HDMI_CEC_WKUPCTRL)....1874
34.5.304 I2CM_SLAVE (HDMI_I2CM_SLAVE)....1875
34.5.305 I2CM_ADDRESS (HDMI_I2CM_ADDRESS)....1876
34.5.306 I2CM_DATAO (HDMI_I2CM_DATAO)....1876
34.5.307 I2CM_DATAI (HDMI_I2CM_DATAI)....1877
34.5.308 I2CM_OPERATION (HDMI_I2CM_OPERATION)....1877
34.5.309 I2CM_INT (HDMI_I2CM_INT)....1878
34.5.310 I2CM_CTLINT (HDMI_I2CM_CTLINT)....1879
34.5.311 I2CM_DIV (HDMI_I2CM_DIV)....1879
34.5.312 I2CM_SEGADDR (HDMI_I2CM_SEGADDR)....1880
34.5.313 I2CM_SOFTRSTZ (HDMI_I2CM_SOFTRSTZ)....1881

34.5.314 I2CM_SEGPTR (HDMI_I2CM_SEGPTR) 1881
34.5.315 I2CM_SS_SCL_HCNT_1_ADDR (HDMI_I2CM_SS_SCL_HCNT_1_ADDR) 1882
34.5.316 I2CM_SS_SCL_HCNT_0_ADDR (HDMI_I2CM_SS_SCL_HCNT_0_ADDR) 1882
34.5.317 I2CM_SS_SCL_LCNT_1_ADDR (HDMI_I2CM_SS_SCL_LCNT_1_ADDR) 1883
34.5.318 I2CM_SS_SCL_LCNT_0_ADDR (HDMI_I2CM_SS_SCL_LCNT_0_ADDR) 1883
34.5.319 I2CM_FS_SCL_HCNT_1_ADDR (HDMI_I2CM_FS_SCL_HCNT_1_ADDR) 1884
34.5.320 I2CM_FS_SCL_HCNT_0_ADDR (HDMI_I2CM_FS_SCL_HCNT_0_ADDR) 1884
34.5.321 I2CM_FS_SCL_LCNT_1_ADDR (HDMI_I2CM_FS_SCL_LCNT_1_ADDR) 1885
34.5.322 I2CM_FS_SCL_LCNT_0_ADDR (HDMI_I2CM_FS_SCL_LCNT_0_ADDR) 1885
34.5.323 BASE_POINTER_ADDR (HDMI_BASE_POINTER_ADDR) 1886

Chapter 35
HDMI 3D Tx PHY (HDMI_PHY)

35.1 Overview 1887
35.1.1 General Description 1887
35.1.2 Applications 1887
35.1.3 Standards Compliance 1888
35.1.4 Features 1888
35.1.5 HDMI 3D Tx PHY System-Level Overview 1889
35.1.5.1 System-Level Block Diagram 1889
35.1.5.2 HDMI 3D Tx PHY 1889
35.1.5.2.1 Interfaces 1890
35.1.5.3 HDMI 1.4 I/O Pads 1890
35.1.5.3.1 I/O Pads Description 1890
35.2 External Signals 1891
35.2.1 Top-Level I/O Diagram 1892
35.2.2 Top -Level Signal Descriptions 1893
35.2.2.1 TMDS Interface 1894
35.2.2.2 Reset Signals 1894
35.2.2.3 External Component 1895

35.3 Functional Description....1895
35.3.1 Functional Overview....1895
35.3.2 Operating Modes....1897
35.3.2.1 Power-Down Mode....1898
35.3.2.2 Active Mode....1899
35.3.2.2.1 Wide Interface Mode....1901
35.3.2.3 Power Sequence....1901
35.3.2.3.1 PLL/MPLL....1904
35.3.2.3.2 Resistor, ADC Calibration....1906
35.3.2.3.3 Clock Alignment....1908
35.3.2.4 Color Depth and Color Mode Selection....1909
35.3.2.4.1 Pixel Repetition Clock Generation Selection....1909
35.3.3 Configuration and Test Mode....1909
35.3.3.1 Power-Up Configuration....1910
35.4 System-Level Implementation....1913
35.4.1 System Operation....1913
35.4.1.1 Powering Up and Powering Down....1913
35.4.1.2 Active Mode Requirements....1915
35.4.1.3 Power-Up Requirements....1915
35.4.1.4 Power Supply Sequence When the HDMI 3D Tx PHY is Not Used....1915
35.4.1.5 Power-Down Requirements....1915
35.5 Reference Clock....1915
35.6 Control Registers....1916
35.6.1 Control Registers Module Design Architecture....1916
35.7 HDMI_PHY Memory Map/Register Definition....1916
35.7.1 Power Control (HDMI_PHY_PWRCTRL)....1918
35.7.2 Serializer Divider Control (HDMI_PHY_SERDIVCTRL)....1920
35.7.3 Serializer Clock Control (HDMI_PHY_SERCKCTRL)....1920
35.7.4 Serializer Clock Kill Control (HDMI_PHY_SERCKKILLCTRL)....1921

35.7.5 Transmitter and Resistance Calibration Control (HDMI_PHY_TXRESCTRL)..... 1922
35.7.6 Clock Calibration Control (HDMI_PHY_CKCALCTRL)..... 1923
35.7.7 Color Depth, Pixel Repetition, Clock Divider for PLL and MPLL, and Edge Rate Control (HDMI_PHY_CPCE_CTRL)..... 1924
35.7.8 Tx and Clock Measure Control (HDMI_PHY_TXCLKMEASCTRL)..... 1926
35.7.9 Tx Measure Control (HDMI_PHY_TXMEASCTRL)..... 1927
35.7.10 Clock Symbol and Transmitter Control (HDMI_PHY_CKSYMTXCTRL)..... 1929
35.7.11 Comparator Sequence Control (HDMI_PHY_CMPSEQCTRL)..... 1930
35.7.12 Comparator Power Control (HDMI_PHY_CMPPWRCTRL)..... 1931
35.7.13 Comparator Mode Control (HDMI_PHY_CMPMODECTRL)..... 1931
35.7.14 Measure Control (HDMI_PHY_MEASCTRL)..... 1932
35.7.15 Voltage Level Control (HDMI_PHY_VLEVCTRL)..... 1933
35.7.16 Digital-to-Analog Control (HDMI_PHY_D2ACTRL)..... 1934
35.7.17 Current Control (HDMI_PHY_CURRCTRL)..... 1935
35.7.18 Drive Analog Control (HDMI_PHY_DRVANACTRL)..... 1935
35.7.19 PLL Measure Control (HDMI_PHY_PLLMEASCTRL)..... 1936
35.7.20 PLL Phase and Bypass Control (HDMI_PHY_PLLPHBYCTRL)..... 1938
35.7.21 Gear Shift, Reset Mode, and Power State Control (HDMI_PHY_GRP_CTRL)..... 1939
35.7.22 Gmp Control (HDMI_PHY_GMPCTRL)..... 1940
35.7.23 MPLL Measure Control (HDMI_PHY_MPLLMEASCTRL)..... 1941
35.7.24 MPLL and PLL Phase, Scope Clock Select, and MUX Clock Control (HDMI_PHY_MSM_CTRL)..... 1943
35.7.25 Scope, Comparator Result and Power Bad Status (HDMI_PHY_SCRPB_STATUS)..... 1944
35.7.26 Transmission Termination (HDMI_PHY_TXTERM)..... 1946
35.7.27 Power Sequence, TX Clock Alignment, Resistance Calibration, Pattern Generator Skip Bit, and TMDS Encoder Enable (HDMI_PHY_PTRPT_ENBL)..... 1947
35.7.28 Pattern Generator Mode (HDMI_PHY_PATTERNGEN)..... 1949
35.7.29 The Soft-Reset and DAC Enable, Clock Alignment and PG Mode (HDMI_PHY_SDCAP_MODE)..... 1950
35.7.30 Scope Mode register (HDMI_PHY_SCOPEMODE)..... 1952
35.7.31 Digital Transmission Mode (HDMI_PHY_DIGTXMODE)..... 1953

35.7.32 Scope, Transmission Clock Alignment, and Resistance Calibration Set-on-Done Status (HDMI_PHY_STR_STATUS)..... 1957
35.7.33 Scope Counter on Channel 0 (HDMI_PHY_SCOPECNT0)..... 1959
35.7.34 Scope Counter on Channel 1 (HDMI_PHY_SCOPECNT1)..... 1959
35.7.35 Scope Counter on Channel 2 (HDMI_PHY_SCOPECNT2)..... 1960
35.7.36 Scope Counter on Clock Channel (HDMI_PHY_SCOPECNTCLK)..... 1960
35.7.37 Scope Sample Count MSB, Scope Sample Repetition (HDMI_PHY_SCOPESAMPLE)..... 1961
35.7.38 Scope Counter MSB Channel 0 and Channel 1 (HDMI_PHY_SCOPECNTMSB01)..... 1962
35.7.39 Scope Counter MSB Channel 2 and Clock Channel (HDMI_PHY_SCOPECNTMSB2CK)..... 1962
35.8 Appendix A: Driver Voltage Level Configuration..... 1963
35.9 Appendix B..... 1964
35.9.1 Single or Two-PLL in Coherent or Non-Coherent Mode of Operation..... 1965
35.9.2 PLL/MPLL Generic Configuration Settings..... 1965
35.10 Appendix C: 3D Video Formats..... 1971

Chapter 36
I2C Controller (I2C)

36.1 Overview..... 1975
36.1.1 Features..... 1977
36.1.2 Modes and operations..... 1978
36.2 External Signals..... 1978
36.3 Clocks..... 1979
36.4 Functional description..... 1979
36.4.1 I2C system configuration..... 1979
36.4.2 Arbitration procedure..... 1980
36.4.3 Clock synchronization..... 1980
36.4.4 Handshaking..... 1981
36.4.5 Clock stretching..... 1981
36.4.6 Peripheral bus accesses..... 1982
36.4.7 Generation of transfer error on IP bus..... 1982

36.4.8 Reset....1982
36.4.9 Interrupts....1982
36.4.10 Byte order....1982
36.5 Initialization....1983
36.5.1 Initialization sequence....1983
36.5.2 Generation of Start....1983
36.5.3 Post-transfer software response....1983
36.5.4 Generation of Stop....1984
36.5.5 Generation of Repeated Start....1984
36.5.6 Slave mode....1985
36.5.7 Arbitration lost....1985
36.6 Software restriction....1991
36.7 I2C Memory Map/Register Definition....1992
36.7.1 I2C Address Register (I2C*x*_IADR)....1993
36.7.2 I2C Frequency Divider Register (I2C*x*_IFDR)....1993
36.7.3 I2C Control Register (I2C*x*_I2CR)....1995
36.7.4 I2C Status Register (I2C*x*_I2SR)....1996
36.7.5 I2C Data I/O Register (I2C*x*_I2DR)....1998

Chapter 37
IOMUX Controller (IOMUXC)

37.1 Overview....1999
37.1.1 Features....2000
37.2 Clocks....2001
37.3 Functional description....2001
37.3.1 ALT6 and ALT7 extended muxing modes....2002
37.3.2 SW Loopback through SION bit....2003
37.3.3 Daisy chain - multi pads driving same module input pin....2003
37.4 IOMUXC Memory Map/Register Definition....2004
37.4.1 GPR0 (IOMUXC_GPR0)....2031

37.4.2 GPR1 (IOMUXC_GPR1)..... 2034
37.4.3 GPR2 (IOMUXC_GPR2)..... 2037
37.4.4 GPR3 (IOMUXC_GPR3)..... 2039
37.4.5 GPR4 (IOMUXC_GPR4)..... 2043
37.4.6 GPR5 (IOMUXC_GPR5)..... 2046
37.4.7 GPR6 (IOMUXC_GPR6)..... 2047
37.4.8 GPR7 (IOMUXC_GPR7)..... 2048
37.4.9 GPR8 (IOMUXC_GPR8)..... 2048
37.4.10 GPR9 (IOMUXC_GPR9)..... 2050
37.4.11 GPR10 (IOMUXC_GPR10)..... 2051
37.4.12 GPR11 (IOMUXC_GPR11)..... 2053
37.4.13 GPR12 (IOMUXC_GPR12)..... 2053
37.4.14 GPR13 (IOMUXC_GPR13)..... 2055
37.4.15 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA10)..... 2057
37.4.16 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA11)..... 2058
37.4.17 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA12)..... 2059
37.4.18 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA13)..... 2060
37.4.19 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA14)..... 2061
37.4.20 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA15)..... 2062
37.4.21 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA16)..... 2063
37.4.22 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA17)..... 2064
37.4.23 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA18)..... 2065
37.4.24 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA19)..... 2066
37.4.25 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA04)..... 2067
37.4.26 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA05)..... 2068
37.4.27 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA06)..... 2069
37.4.28 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA07)..... 2070
37.4.29 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA08)..... 2071
37.4.30 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA09)..... 2072

37.4.31 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA_EN)..... 2073
37.4.32 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_HSYNC)..... 2074
37.4.33 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_PIXCLK)..... 2075
37.4.34 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_VSYNC)..... 2076
37.4.35 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DI0_DISP_CLK)..... 2077
37.4.36 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DI0_PIN15)..... 2077
37.4.37 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DI0_PIN02)..... 2078
37.4.38 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DI0_PIN03)..... 2079
37.4.39 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DI0_PIN04)..... 2079
37.4.40 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA00)..... 2080
37.4.41 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA01)..... 2081
37.4.42 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA10)..... 2082
37.4.43 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA11)..... 2082
37.4.44 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA12)..... 2083
37.4.45 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA13)..... 2084
37.4.46 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA14)..... 2085
37.4.47 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA15)..... 2086
37.4.48 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA16)..... 2087
37.4.49 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA17)..... 2088
37.4.50 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA18)..... 2089
37.4.51 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA19)..... 2090
37.4.52 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA02)..... 2091
37.4.53 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA20)..... 2092
37.4.54 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA21)..... 2093
37.4.55 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA22)..... 2094
37.4.56 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA23)..... 2095
37.4.57 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA03)..... 2096
37.4.58 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA04)..... 2097
37.4.59 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA05)..... 2098

37.4.60 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA06)..... 2099
37.4.61 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA07)..... 2100
37.4.62 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA08)..... 2101
37.4.63 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA09)..... 2102
37.4.64 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR16)..... 2103
37.4.65 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR17)..... 2104
37.4.66 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR18)..... 2105
37.4.67 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR19)..... 2106
37.4.68 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR20)..... 2107
37.4.69 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR21)..... 2108
37.4.70 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR22)..... 2109
37.4.71 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR23)..... 2110
37.4.72 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR24)..... 2111
37.4.73 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR25)..... 2112
37.4.74 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_BCLK)..... 2113
37.4.75 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_CS0_B)..... 2113
37.4.76 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_CS1_B)..... 2114
37.4.77 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA16)..... 2115
37.4.78 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA17)..... 2116
37.4.79 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA18)..... 2117
37.4.80 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA19)..... 2118
37.4.81 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA20)..... 2119
37.4.82 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA21)..... 2120
37.4.83 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA22)..... 2121
37.4.84 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA23)..... 2122
37.4.85 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA24)..... 2123
37.4.86 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA25)..... 2124
37.4.87 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA26)..... 2125
37.4.88 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA27)..... 2126

37.4.89 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA28)....2127
37.4.90 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA29)....2128
37.4.91 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA30)....2129
37.4.92 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA31)....2130
37.4.93 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD00)....2131
37.4.94 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD01)....2132
37.4.95 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD10)....2133
37.4.96 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD11)....2134
37.4.97 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD12)....2135
37.4.98 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD13)....2136
37.4.99 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD14)....2137
37.4.100 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD15)....2138
37.4.101 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD02)....2139
37.4.102 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD03)....2140
37.4.103 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD04)....2141
37.4.104 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD05)....2142
37.4.105 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD06)....2143
37.4.106 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD07)....2144
37.4.107 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD08)....2145
37.4.108 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD09)....2146
37.4.109 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_EB0_B)....2147
37.4.110 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_EB1_B)....2148
37.4.111 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_EB2_B)....2149
37.4.112 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_EB3_B)....2150
37.4.113 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_LBA_B)....2151
37.4.114 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_OE_B)....2152
37.4.115 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_RW)....2153
37.4.116 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_WAIT_B)....2154
37.4.117 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_CRS_DV)....2155

37.4.118 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_MDC)....2156
37.4.119 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO)....2157
37.4.120 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_REF_CLK)....2158
37.4.121 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RX_ER)....2159
37.4.122 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA0)....2160
37.4.123 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA1)....2161
37.4.124 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TX_EN)....2162
37.4.125 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA0)....2163
37.4.126 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA1)....2164
37.4.127 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO00)....2165
37.4.128 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO01)....2166
37.4.129 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO16)....2167
37.4.130 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO17)....2168
37.4.131 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO18)....2169
37.4.132 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO19)....2170
37.4.133 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO02)....2171
37.4.134 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO03)....2172
37.4.135 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO04)....2173
37.4.136 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO05)....2174
37.4.137 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO06)....2175
37.4.138 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO07)....2176
37.4.139 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO08)....2177
37.4.140 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO09)....2178
37.4.141 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_COL0)....2179
37.4.142 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_COL1)....2180
37.4.143 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_COL2)....2181
37.4.144 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_COL3)....2182
37.4.145 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_COL4)....2183
37.4.146 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_ROW0)....2184

37.4.147 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_ROW1)..... 2185
37.4.148 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_ROW2)..... 2186
37.4.149 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_ROW3)..... 2187
37.4.150 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_ROW4)..... 2188
37.4.151 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_ALE).....2189
37.4.152 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CLE).....2189
37.4.153 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CS0_B).....2190
37.4.154 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CS1_B).....2191
37.4.155 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CS2_B).....2191
37.4.156 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CS3_B).....2192
37.4.157 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA00)..... 2193
37.4.158 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA01)..... 2194
37.4.159 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA02)..... 2195
37.4.160 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA03)..... 2196
37.4.161 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA04)..... 2197
37.4.162 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA05)..... 2198
37.4.163 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA06)..... 2199
37.4.164 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA07)..... 2200
37.4.165 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_READY_B).....2201
37.4.166 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_WP_B).....2201
37.4.167 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RD0).....2202
37.4.168 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RD1).....2203
37.4.169 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RD2).....2204
37.4.170 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RD3).....2205
37.4.171 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RX_CTL)..... 2206
37.4.172 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RXC)..... 2207
37.4.173 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TD0).....2208
37.4.174 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TD1).....2209
37.4.175 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TD2).....2210

37.4.176 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TD3)....2211
37.4.177 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TX_CTL)....2212
37.4.178 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TXC)....2213
37.4.179 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_CLK)....2214
37.4.180 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_CMD)....2215
37.4.181 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0)....2216
37.4.182 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1)....2217
37.4.183 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2)....2218
37.4.184 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3)....2219
37.4.185 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CLK)....2220
37.4.186 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CMD)....2221
37.4.187 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0)....2222
37.4.188 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1)....2223
37.4.189 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2)....2224
37.4.190 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3)....2225
37.4.191 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_CLK)....2226
37.4.192 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_CMD)....2227
37.4.193 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA0)....2228
37.4.194 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA1)....2229
37.4.195 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA2)....2230
37.4.196 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA3)....2230
37.4.197 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA4)....2231
37.4.198 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA5)....2232
37.4.199 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA6)....2233
37.4.200 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA7)....2234
37.4.201 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_RESET)....2235
37.4.202 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_CLK)....2236
37.4.203 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_CMD)....2237
37.4.204 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA0)....2238

37.4.205 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA1)....2239
37.4.206 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA2)....2240
37.4.207 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA3)....2241
37.4.208 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA4)....2241
37.4.209 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA5)....2242
37.4.210 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA6)....2243
37.4.211 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA7)....2244
37.4.212 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA10).... 2245
37.4.213 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA11).... 2246
37.4.214 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA12).... 2248
37.4.215 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA13).... 2250
37.4.216 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA14).... 2252
37.4.217 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA15).... 2253
37.4.218 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA16).... 2255
37.4.219 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA17).... 2257
37.4.220 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA18).... 2259
37.4.221 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA19).... 2260
37.4.222 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA04).... 2262
37.4.223 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA05).... 2264
37.4.224 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA06).... 2266
37.4.225 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA07).... 2267
37.4.226 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA08).... 2269
37.4.227 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA09).... 2271
37.4.228 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA_EN).... 2273
37.4.229 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_HSYNC).... 2274
37.4.230 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_PIXCLK).... 2276
37.4.231 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_VSYNC).... 2278
37.4.232 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DI0_DISP_CLK).... 2280
37.4.233 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN15).... 2281

37.4.234 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN02)..... 2283
37.4.235 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN03)..... 2285
37.4.236 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN04)..... 2287
37.4.237 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA00)..... 2288
37.4.238 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA01)..... 2290
37.4.239 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA10)..... 2292
37.4.240 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA11)..... 2294
37.4.241 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA12)..... 2295
37.4.242 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA13)..... 2297
37.4.243 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA14)..... 2299
37.4.244 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA15)..... 2301
37.4.245 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA16)..... 2302
37.4.246 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA17)..... 2304
37.4.247 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA18)..... 2306
37.4.248 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA19)..... 2308
37.4.249 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA02)..... 2309
37.4.250 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA20)..... 2311
37.4.251 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA21)..... 2313
37.4.252 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA22)..... 2315
37.4.253 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA23)..... 2316
37.4.254 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA03)..... 2318
37.4.255 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA04)..... 2320
37.4.256 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA05)..... 2322
37.4.257 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA06)..... 2323
37.4.258 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA07)..... 2325
37.4.259 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA08)..... 2327
37.4.260 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA09)..... 2329
37.4.261 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR00)..... 2331
37.4.262 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR01)..... 2333

37.4.263 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR10)..... 2335
37.4.264 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR11)..... 2337
37.4.265 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR12)..... 2339
37.4.266 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR13)..... 2341
37.4.267 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR14)..... 2343
37.4.268 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR15)..... 2345
37.4.269 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR02)..... 2347
37.4.270 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR03)..... 2349
37.4.271 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR04)..... 2351
37.4.272 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR05)..... 2353
37.4.273 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR06)..... 2355
37.4.274 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR07)..... 2357
37.4.275 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR08)..... 2359
37.4.276 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR09)..... 2361
37.4.277 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS_B)..... 2363
37.4.278 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_CS0_B)..... 2365
37.4.279 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_CS1_B)..... 2367
37.4.280 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0)..... 2369
37.4.281 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1)..... 2371
37.4.282 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2)..... 2373
37.4.283 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3)..... 2375
37.4.284 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM4)..... 2377
37.4.285 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM5)..... 2379
37.4.286 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM6)..... 2381
37.4.287 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM7)..... 2383
37.4.288 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS_B)..... 2385
37.4.289 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET)..... 2387
37.4.290 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA0)..... 2389
37.4.291 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA1)..... 2391

37.4.292 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA2)..... 2393
37.4.293 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE0).....2395
37.4.294 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE1).....2397
37.4.295 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK0_P)..... 2399
37.4.296 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK1_P)..... 2401
37.4.297 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ODT0)..... 2403
37.4.298 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ODT1)..... 2405
37.4.299 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0_P)..... 2407
37.4.300 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS1_P)..... 2409
37.4.301 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2_P)..... 2411
37.4.302 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3_P)..... 2413
37.4.303 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS4_P)..... 2415
37.4.304 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS5_P)..... 2417
37.4.305 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS6_P)..... 2419
37.4.306 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS7_P)..... 2421
37.4.307 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDWE_B).....2423
37.4.308 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR16)..... 2424
37.4.309 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR17)..... 2426
37.4.310 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR18)..... 2428
37.4.311 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR19)..... 2429
37.4.312 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR20)..... 2431
37.4.313 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR21)..... 2433
37.4.314 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR22)..... 2435
37.4.315 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR23)..... 2436
37.4.316 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR24)..... 2438
37.4.317 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR25)..... 2440
37.4.318 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_BCLK).....2442
37.4.319 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_CS0_B).....2443
37.4.320 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_CS1_B).....2445

37.4.321 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA16)..... 2447
37.4.322 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA17)..... 2449
37.4.323 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA18)..... 2450
37.4.324 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA19)..... 2452
37.4.325 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA20)..... 2454
37.4.326 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA21)..... 2456
37.4.327 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA22)..... 2457
37.4.328 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA23)..... 2459
37.4.329 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA24)..... 2461
37.4.330 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA25)..... 2463
37.4.331 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA26)..... 2464
37.4.332 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA27)..... 2466
37.4.333 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA28)..... 2468
37.4.334 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA29)..... 2470
37.4.335 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA30)..... 2471
37.4.336 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA31)..... 2473
37.4.337 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD00).....2475
37.4.338 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD01).....2477
37.4.339 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD10).....2478
37.4.340 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD11).....2480
37.4.341 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD12).....2482
37.4.342 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD13).....2484
37.4.343 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD14).....2485
37.4.344 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD15).....2487
37.4.345 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD02).....2489
37.4.346 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD03).....2491
37.4.347 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD04).....2492
37.4.348 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD05).....2494
37.4.349 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD06).....2496

37.4.350 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD07)....2498
37.4.351 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD08)....2499
37.4.352 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD09)....2501
37.4.353 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_EB0_B)....2503
37.4.354 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_EB1_B)....2505
37.4.355 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_EB2_B)....2506
37.4.356 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_EB3_B)....2508
37.4.357 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_LBA_B)....2510
37.4.358 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_OE_B).... 2512
37.4.359 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_RW).... 2513
37.4.360 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_WAIT_B).... 2515
37.4.361 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_CRS_DV).... 2517
37.4.362 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_MDC).... 2519
37.4.363 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_MDIO)....2520
37.4.364 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_REF_CLK).... 2522
37.4.365 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RX_ER)....2524
37.4.366 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RX_DATA0)....2526
37.4.367 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RX_DATA1)....2527
37.4.368 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TX_EN)....2529
37.4.369 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TX_DATA0).... 2531
37.4.370 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TX_DATA1).... 2533
37.4.371 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO00).... 2534
37.4.372 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO01).... 2536
37.4.373 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO16).... 2538
37.4.374 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO17).... 2540
37.4.375 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO18).... 2541
37.4.376 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO19).... 2543
37.4.377 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO02).... 2545
37.4.378 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO03).... 2547

37.4.379 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO04)..... 2548
37.4.380 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO05)..... 2550
37.4.381 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO06)..... 2552
37.4.382 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO07)..... 2554
37.4.383 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO08)..... 2555
37.4.384 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO09)..... 2557
37.4.385 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD).....2559
37.4.386 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK).....2561
37.4.387 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI)..... 2562
37.4.388 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO).....2564
37.4.389 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS).....2566
37.4.390 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TRSTB)..... 2567
37.4.391 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_COL0)..... 2569
37.4.392 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_COL1)..... 2571
37.4.393 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_COL2)..... 2572
37.4.394 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_COL3)..... 2574
37.4.395 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_COL4)..... 2576
37.4.396 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW0)..... 2578
37.4.397 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW1)..... 2579
37.4.398 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW2)..... 2581
37.4.399 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW3)..... 2583
37.4.400 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW4)..... 2585
37.4.401 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_ALE)..... 2586
37.4.402 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CLE).....2588
37.4.403 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CS0_B).....2590
37.4.404 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CS1_B).....2592
37.4.405 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CS2_B).....2593
37.4.406 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CS3_B).....2595
37.4.407 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA00)..... 2597

37.4.408 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA01)..... 2599
37.4.409 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA02)..... 2600
37.4.410 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA03)..... 2602
37.4.411 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA04)..... 2604
37.4.412 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA05)..... 2606
37.4.413 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA06)..... 2607
37.4.414 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA07)..... 2609
37.4.415 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_READY_B)..... 2611
37.4.416 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_WP_B)..... 2613
37.4.417 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RD0)..... 2615
37.4.418 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RD1)..... 2617
37.4.419 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RD2)..... 2619
37.4.420 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RD3)..... 2621
37.4.421 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RX_CTL)..... 2623
37.4.422 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RXC)..... 2625
37.4.423 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TD0)..... 2627
37.4.424 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TD1)..... 2629
37.4.425 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TD2)..... 2631
37.4.426 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TD3)..... 2633
37.4.427 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TX_CTL)..... 2635
37.4.428 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TXC)..... 2637
37.4.429 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_CLK)..... 2639
37.4.430 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_CMD)..... 2641
37.4.431 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0)..... 2642
37.4.432 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1)..... 2644
37.4.433 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2)..... 2646
37.4.434 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3)..... 2648
37.4.435 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CLK)..... 2649
37.4.436 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CMD)..... 2651

37.4.437 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0)..... 2653
37.4.438 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1)..... 2655
37.4.439 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2)..... 2656
37.4.440 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3)..... 2658
37.4.441 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_CLK).....2660
37.4.442 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_CMD)..... 2662
37.4.443 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA0)..... 2663
37.4.444 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA1)..... 2665
37.4.445 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA2)..... 2667
37.4.446 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA3)..... 2669
37.4.447 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA4)..... 2670
37.4.448 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA5)..... 2672
37.4.449 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA6)..... 2674
37.4.450 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA7)..... 2676
37.4.451 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_RESET)..... 2677
37.4.452 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_CLK).....2679
37.4.453 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_CMD)..... 2681
37.4.454 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA0)..... 2683
37.4.455 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA1)..... 2684
37.4.456 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA2)..... 2686
37.4.457 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA3)..... 2688
37.4.458 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA4)..... 2690
37.4.459 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA5)..... 2691
37.4.460 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA6)..... 2693
37.4.461 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA7)..... 2695
37.4.462 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B7DS).....2697
37.4.463 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_ADDDS).....2697
37.4.464 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL).....2698
37.4.465 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDRPKE).....2699

37.4.466 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDRPK)..... 2700
37.4.467 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDRHYS)..... 2701
37.4.468 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDRMODE)..... 2702
37.4.469 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B0DS)..... 2703
37.4.470 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII)..... 2703
37.4.471 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_CTLDS)..... 2704
37.4.472 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B1DS)..... 2705
37.4.473 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE)..... 2705
37.4.474 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B2DS)..... 2706
37.4.475 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B3DS)..... 2707
37.4.476 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B4DS)..... 2708
37.4.477 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B5DS)..... 2708
37.4.478 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_RGMII_TERM)..... 2709
37.4.479 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B6DS)..... 2710
37.4.480 Select Input Register (IOMUXC_ANALOG_USB_OTG_ID_SELECT_INPUT)..... 2711
37.4.481 Select Input Register (IOMUXC_ASRC_ASRCK_CLOCK_6_SELECT_INPUT)..... 2711
37.4.482 Select Input Register (IOMUXC_AUD4_INPUT_DA_AMX_SELECT_INPUT)..... 2712
37.4.483 Select Input Register (IOMUXC_AUD4_INPUT_DB_AMX_SELECT_INPUT)..... 2713
37.4.484 Select Input Register (IOMUXC_AUD4_INPUT_RXCLK_AMX_SELECT_INPUT)..... 2714
37.4.485 Select Input Register (IOMUXC_AUD4_INPUT_RXFS_AMX_SELECT_INPUT)..... 2715
37.4.486 Select Input Register (IOMUXC_AUD4_INPUT_TXCLK_AMX_SELECT_INPUT)..... 2716
37.4.487 Select Input Register (IOMUXC_AUD4_INPUT_TXFS_AMX_SELECT_INPUT)..... 2717
37.4.488 Select Input Register (IOMUXC_AUD5_INPUT_DA_AMX_SELECT_INPUT)..... 2718
37.4.489 Select Input Register (IOMUXC_AUD5_INPUT_DB_AMX_SELECT_INPUT)..... 2719
37.4.490 Select Input Register (IOMUXC_AUD5_INPUT_RXCLK_AMX_SELECT_INPUT)..... 2720
37.4.491 Select Input Register (IOMUXC_AUD5_INPUT_RXFS_AMX_SELECT_INPUT)..... 2721
37.4.492 Select Input Register (IOMUXC_AUD5_INPUT_TXCLK_AMX_SELECT_INPUT)..... 2722
37.4.493 Select Input Register (IOMUXC_AUD5_INPUT_TXFS_AMX_SELECT_INPUT)..... 2723
37.4.494 Select Input Register (IOMUXC_FLEXCAN1_RX_SELECT_INPUT)..... 2723

37.4.495 Select Input Register (IOMUXC_FLEXCAN2_RX_SELECT_INPUT)....2724
37.4.496 Select Input Register (IOMUXC_CCM_PMIC_READY_SELECT_INPUT).... 2725
37.4.497 Select Input Register (IOMUXC_ECSPI1_CSPI_CLK_IN_SELECT_INPUT).... 2725
37.4.498 Select Input Register (IOMUXC_ECSPI1_MISO_SELECT_INPUT)....2726
37.4.499 Select Input Register (IOMUXC_ECSPI1_MOSI_SELECT_INPUT)....2727
37.4.500 Select Input Register (IOMUXC_ECSPI1_SS0_SELECT_INPUT).... 2727
37.4.501 Select Input Register (IOMUXC_ECSPI1_SS1_SELECT_INPUT).... 2728
37.4.502 Select Input Register (IOMUXC_ECSPI1_SS2_SELECT_INPUT).... 2729
37.4.503 Select Input Register (IOMUXC_ECSPI1_SS3_SELECT_INPUT).... 2730
37.4.504 Select Input Register (IOMUXC_ECSPI2_CSPI_CLK_IN_SELECT_INPUT).... 2730
37.4.505 Select Input Register (IOMUXC_ECSPI2_MISO_SELECT_INPUT)....2731
37.4.506 Select Input Register (IOMUXC_ECSPI2_MOSI_SELECT_INPUT)....2732
37.4.507 Select Input Register (IOMUXC_ECSPI2_SS0_SELECT_INPUT).... 2732
37.4.508 Select Input Register (IOMUXC_ECSPI2_SS1_SELECT_INPUT).... 2733
37.4.509 Select Input Register (IOMUXC_ECSPI4_SS0_SELECT_INPUT).... 2734
37.4.510 Select Input Register (IOMUXC_ENET_REF_CLK_SELECT_INPUT).... 2735
37.4.511 Select Input Register (IOMUXC_ENET_MAC0_MDIO_SELECT_INPUT)....2736
37.4.512 Select Input Register (IOMUXC_ENET_MAC0_RX_CLK_SELECT_INPUT)....2737
37.4.513 Select Input Register (IOMUXC_ENET_MAC0_RX_DATA0_SELECT_INPUT)....2738
37.4.514 Select Input Register (IOMUXC_ENET_MAC0_RX_DATA1_SELECT_INPUT)....2739
37.4.515 Select Input Register (IOMUXC_ENET_MAC0_RX_DATA2_SELECT_INPUT)....2740
37.4.516 Select Input Register (IOMUXC_ENET_MAC0_RX_DATA3_SELECT_INPUT)....2741
37.4.517 Select Input Register (IOMUXC_ENET_MAC0_RX_EN_SELECT_INPUT)....2742
37.4.518 Select Input Register (IOMUXC_ESAI_RX_FS_SELECT_INPUT)....2743
37.4.519 Select Input Register (IOMUXC_ESAI_TX_FS_SELECT_INPUT).... 2744
37.4.520 Select Input Register (IOMUXC_ESAI_RX_HF_CLK_SELECT_INPUT).... 2745
37.4.521 Select Input Register (IOMUXC_ESAI_TX_HF_CLK_SELECT_INPUT).... 2746
37.4.522 Select Input Register (IOMUXC_ESAI_RX_CLK_SELECT_INPUT).... 2747
37.4.523 Select Input Register (IOMUXC_ESAI_TX_CLK_SELECT_INPUT)....2748

37.4.524 Select Input Register (IOMUXC_ESAI_SDO0_SELECT_INPUT)....2749
37.4.525 Select Input Register (IOMUXC_ESAI_SDO1_SELECT_INPUT)....2750
37.4.526 Select Input Register (IOMUXC_ESAI_SDO2_SDI3_SELECT_INPUT).... 2751
37.4.527 Select Input Register (IOMUXC_ESAI_SDO3_SDI2_SELECT_INPUT).... 2752
37.4.528 Select Input Register (IOMUXC_ESAI_SDO4_SDI1_SELECT_INPUT).... 2753
37.4.529 Select Input Register (IOMUXC_ESAI_SDO5_SDI0_SELECT_INPUT).... 2754
37.4.530 Select Input Register (IOMUXC_HDMI_ICECIN_SELECT_INPUT)....2755
37.4.531 Select Input Register (IOMUXC_HDMI_II2C_CLKIN_SELECT_INPUT)....2756
37.4.532 Select Input Register (IOMUXC_HDMI_II2C_DATAIN_SELECT_INPUT).... 2757
37.4.533 Select Input Register (IOMUXC_I2C1_SCL_IN_SELECT_INPUT).... 2758
37.4.534 Select Input Register (IOMUXC_I2C1_SDA_IN_SELECT_INPUT)....2759
37.4.535 Select Input Register (IOMUXC_I2C2_SCL_IN_SELECT_INPUT).... 2760
37.4.536 Select Input Register (IOMUXC_I2C2_SDA_IN_SELECT_INPUT)....2761
37.4.537 Select Input Register (IOMUXC_I2C3_SCL_IN_SELECT_INPUT).... 2761
37.4.538 Select Input Register (IOMUXC_I2C3_SDA_IN_SELECT_INPUT)....2762
37.4.539 Select Input Register (IOMUXC_I2C4_SCL_IN_SELECT_INPUT).... 2763
37.4.540 Select Input Register (IOMUXC_I2C4_SDA_IN_SELECT_INPUT)....2763
37.4.541 Select Input Register (IOMUXC_IPU1_SENS1_DATA10_SELECT_INPUT).... 2764
37.4.542 Select Input Register (IOMUXC_IPU1_SENS1_DATA11_SELECT_INPUT).... 2765
37.4.543 Select Input Register (IOMUXC_IPU1_SENS1_DATA12_SELECT_INPUT).... 2766
37.4.544 Select Input Register (IOMUXC_IPU1_SENS1_DATA13_SELECT_INPUT).... 2767
37.4.545 Select Input Register (IOMUXC_IPU1_SENS1_DATA14_SELECT_INPUT).... 2768
37.4.546 Select Input Register (IOMUXC_IPU1_SENS1_DATA15_SELECT_INPUT).... 2769
37.4.547 Select Input Register (IOMUXC_IPU1_SENS1_DATA16_SELECT_INPUT).... 2770
37.4.548 Select Input Register (IOMUXC_IPU1_SENS1_DATA17_SELECT_INPUT).... 2771
37.4.549 Select Input Register (IOMUXC_IPU1_SENS1_DATA18_SELECT_INPUT).... 2772
37.4.550 Select Input Register (IOMUXC_IPU1_SENS1_DATA19_SELECT_INPUT).... 2773
37.4.551 Select Input Register (IOMUXC_IPU1_SENS1_DATA_EN_SELECT_INPUT).... 2774
37.4.552 Select Input Register (IOMUXC_IPU1_SENS1_HSYNC_SELECT_INPUT).... 2775

37.4.553 Select Input Register (IOMUXC_IPU1_SENS1_PIX_CLK_SELECT_INPUT)..... 2776
37.4.554 Select Input Register (IOMUXC_IPU1_SENS1_VSYNC_SELECT_INPUT)..... 2777
37.4.555 Select Input Register (IOMUXC_KEY_COL5_SELECT_INPUT).....2777
37.4.556 Select Input Register (IOMUXC_KEY_COL6_SELECT_INPUT).....2778
37.4.557 Select Input Register (IOMUXC_KEY_COL7_SELECT_INPUT).....2779
37.4.558 Select Input Register (IOMUXC_KEY_ROW5_SELECT_INPUT)..... 2779
37.4.559 Select Input Register (IOMUXC_KEY_ROW6_SELECT_INPUT)..... 2780
37.4.560 Select Input Register (IOMUXC_KEY_ROW7_SELECT_INPUT)..... 2781
37.4.561 Select Input Register (IOMUXC_MLB_MLB_CLK_IN_SELECT_INPUT)..... 2781
37.4.562 Select Input Register (IOMUXC_MLB_MLB_DATA_IN_SELECT_INPUT)..... 2782
37.4.563 Select Input Register (IOMUXC_MLB_MLB_SIG_IN_SELECT_INPUT).....2783
37.4.564 Select Input Register (IOMUXC_SDMA_EVENTS14_SELECT_INPUT).....2784
37.4.565 Select Input Register (IOMUXC_SDMA_EVENTS15_SELECT_INPUT).....2785
37.4.566 Select Input Register (IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT).....2785
37.4.567 Select Input Register (IOMUXC_SPDIF_TX_CLK2_SELECT_INPUT).....2786
37.4.568 Select Input Register (IOMUXC_UART1_UART_RTS_B_SELECT_INPUT).....2787
37.4.569 Select Input Register (IOMUXC_UART1_UART_RX_DATA_SELECT_INPUT)..... 2787
37.4.570 Select Input Register (IOMUXC_UART2_UART_RTS_B_SELECT_INPUT).....2788
37.4.571 Select Input Register (IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT)..... 2789
37.4.572 Select Input Register (IOMUXC_UART3_UART_RTS_B_SELECT_INPUT).....2789
37.4.573 Select Input Register (IOMUXC_UART3_UART_RX_DATA_SELECT_INPUT)..... 2790
37.4.574 Select Input Register (IOMUXC_UART4_UART_RTS_B_SELECT_INPUT).....2791
37.4.575 Select Input Register (IOMUXC_UART4_UART_RX_DATA_SELECT_INPUT)..... 2791
37.4.576 Select Input Register (IOMUXC_UART5_UART_RTS_B_SELECT_INPUT).....2792
37.4.577 Select Input Register (IOMUXC_UART5_UART_RX_DATA_SELECT_INPUT)..... 2793
37.4.578 Select Input Register (IOMUXC_USB_OTG_OC_SELECT_INPUT)..... 2794
37.4.579 Select Input Register (IOMUXC_USB_H1_OC_SELECT_INPUT).....2795
37.4.580 Select Input Register (IOMUXC_USDHC1_CARD_CLK_IN_SELECT_INPUT).....2796
37.4.581 Select Input Register (IOMUXC_USDHC1_WP_ON_SELECT_INPUT)..... 2797

37.4.582 Select Input Register (IOMUXC_USDHC2_CARD_CLK_IN_SELECT_INPUT)....2798
37.4.583 Select Input Register (IOMUXC_USDHC3_CARD_CLK_IN_SELECT_INPUT)....2799
37.4.584 Select Input Register (IOMUXC_USDHC4_CARD_CLK_IN_SELECT_INPUT)....2800

Chapter 38
Image Processing Unit (IPU)

38.1 Overview....2801
38.1.1 Architecture....2802
38.1.2 Features And Functionality....2803
38.1.2.1 External Ports....2803
38.1.2.1.1 Camera Ports....2803
38.1.2.1.2 Display Ports....2807
38.1.2.1.2.1 Access Modes....2807
38.1.2.1.2.2 Display Interface....2809
38.1.2.1.3 Memory Port....2812
38.1.2.1.4 Processing....2814
38.1.2.1.4.1 Processing flows....2814
38.1.2.1.4.2 Display Processor (DP)....2815
38.1.2.1.5 Video De-Interlacer or Combiner (VDIC)....2816
38.1.2.1.5.1 De-interlacing in the VDIC....2816
38.1.2.1.5.2 Combining in the VDIC....2817
38.1.2.1.5.3 Image Converter (IC)....2817
38.1.2.1.5.4 Image Rotator (IRT)....2818
38.1.2.1.6 Automatic Procedures....2819
38.1.2.1.6.1 Screen Refresh....2819
38.1.2.1.6.2 Update Of The Display Buffer....2820
38.1.2.1.6.3 Camera Preview....2820
38.2 External Signals....2820
38.3 Clocks....2825
38.4 Functional Description....2825

38.4.1 IPU detailed block diagram....2825
38.4.2 Image DMA Controller (IDMAC)....2827
38.4.2.1 IDMAC's channels....2829
38.4.2.2 IBIW & IBIR - Internal bus interface for write and read....2831
38.4.2.3 FCW & FCR - Format converter write and read....2831
38.4.2.4 Buffering units....2834
38.4.2.4.1 Handling real time channels....2835
38.4.2.5 AXIW - AXI Write and AXIR - AXI Read....2835
38.4.2.6 CC_W & CC_R - Channel Control Write and Read....2835
38.4.2.6.1 Locking the arbitration and reordering the AXI bursts....2836
38.4.2.7 AAU_W & AAU_R- Address Arithmetic Unit for Write and Read....2837
38.4.2.7.1 Scrolling support....2839
38.4.2.8 ATC - Alpha Transparency Controller....2839
38.4.2.8.1 Conditional read....2841
38.4.2.9 LUT- Look Up Table....2841
38.4.2.10 CPMEM - Channel Parameter Memory....2842
38.4.2.10.1 CPMEM's words' structure for non interleaved mode....2843
38.4.2.10.2 CPMEM's words' structure for interleaved mode....2848
38.4.2.10.3 Accessing the CPMEM for programming....2856
38.4.2.10.4 Alternate IDMAC settings....2857
38.4.2.11 IDMAC's modes of operation....2857
38.4.2.11.1 Rotation modes....2857
38.4.2.11.2 Frame size....2858
38.4.2.12 IDMAC's restriction....2860
38.4.2.13 IDMAC's Endianness support....2860
38.4.2.14 IDMAC's internal events....2861
38.4.3 Camera Sensor Interface (CSI)....2861
38.4.3.1 CSI Block Diagram....2861
38.4.3.2 CSI Interface....2862

38.4.3.2.1 Parallel Interface 2862
38.4.3.2.2 High-speed serial interface - MIPI (Mobile Industry Processor Interface). .. 2863
38.4.3.3 Test Mode 2863
38.4.3.4 Sensor Image Frame Relations 2864
38.4.3.5 Companding 2865
38.4.3.6 Timing/Data mode protocols 2866
38.4.3.6.1 Gated Mode 2866
38.4.3.6.2 Non-Gated Mode 2866
38.4.3.6.3 BT.656 mode 2867
38.4.3.6.4 BT.1120 mode 2867
38.4.3.7 Packing to memory 2868
38.4.3.8 Skipping frames 2869
38.4.3.9 16 bit camera support 2869
38.4.3.10 CSI Restrictions 2870
38.4.4 Sensor Multi FIFO Controller (SMFC) 2870
38.4.4.1 SMFC's Features 2871
38.4.4.2 SMFC's Functional description 2871
38.4.4.2.1 SMFC Master interface 2874
38.4.4.2.2 Restrictions 2875
38.4.5 Image Converter 2875
38.4.5.1 IC Block Diagram 2875
38.4.5.2 Processing tasks 2876
38.4.5.3 Downsizing Section 2877
38.4.5.4 Main Processing Section 2879
38.4.5.5 Rotation Section 2882
38.4.5.6 IC Task Parameter Memory 2884
38.4.5.7 IC's DMA channels 2888
38.4.5.8 IC restrictions 2889
38.4.5.9 IC bridge 2889

38.4.6 Display port....2889
38.4.6.1 Display ports channels....2890
38.4.6.2 Supported display interfaces....2891
38.4.6.2.1 Synchronous Interfaces....2892
38.4.6.2.2 Asynchronous Parallel Interfaces....2892
38.4.6.3 Display port's bandwidth....2892
38.4.6.4 Display Dual Mode....2893
38.4.6.5 Display Errors....2893
38.4.6.5.1 Data starvation errors....2893
38.4.6.5.2 Anti tearing errors....2894
38.4.6.6 Display port's restrictions....2894
38.4.7 DC - Display Controller....2895
38.4.7.1 Channels flow control....2897
38.4.7.1.1 New Frame control....2897
38.4.7.1.2 Antitearing control....2897
38.4.7.1.3 User command mode control....2897
38.4.7.2 Arbitration Unit....2898
38.4.7.2.1 Access request generator....2898
38.4.7.2.2 DI arbiter....2898
38.4.7.2.3 Source arbiter....2898
38.4.7.3 Microcode processing unit....2899
38.4.7.3.1 Channels address control....2899
38.4.7.3.2 General purpose Data oriented events counters....2899
38.4.7.3.3 Microcode address generator....2899
38.4.7.3.4 Template's Memory Access Arbiter....2900
38.4.7.4 DC's Template structure....2900
38.4.7.4.1 DC template's memory map....2900
38.4.7.5 Display controls' generator....2908
38.4.7.5.1 Bus Mapping Unit....2909

38.4.8 DMFC - Display Multi FIFO Controller....2914
38.4.8.1 DP and DC read channels....2915
38.4.8.1.1 FIFO allocation to channels....2915
38.4.8.1.2 Arbitration between channels....2916
38.4.8.1.3 Watermark....2916
38.4.8.2 IC interface....2917
38.4.8.3 DC write channel and AHB accesses....2917
38.4.9 DP - Display Processor....2917
38.4.9.1 The DP programming model....2918
38.4.9.2 Displayed Planes....2918
38.4.9.3 Combining Unit....2919
38.4.9.4 Cursor Generator....2920
38.4.9.5 Color Space Conversion unit - CSC....2920
38.4.9.5.1 Gamut mapping....2922
38.4.9.6 Gamma correction....2923
38.4.9.7 DC interface....2924
38.4.9.8 DP's flows management....2924
38.4.9.9 DP debug unit....2926
38.4.9.10 Restriction....2927
38.4.10 Display Interface (DI)....2927
38.4.10.1 DC interface, data accumulator and clock domain synchronizer....2929
38.4.10.2 Parallel interface data synchronizer and data oriented interface....2929
38.4.10.3 Timing generator....2929
38.4.10.3.1 Waveform concatenation....2931
38.4.10.3.2 The basic counter....2932
38.4.10.3.3 Counter number 9....2935
38.4.10.3.4 DI's active window....2935
38.4.10.4 Waveform settings for asynchronous interface pins....2936
38.4.10.5 Low Level Access - LLA....2938

38.4.10.6 Using a mask channel 2938
38.4.11 Video De Interlacing or Combining Block (VDIC) 2938
38.4.11.1 VDIC Features 2941
38.4.11.2 De interlacer (DI) sub-block 2941
38.4.11.2.1 Vertical Filter Block (di_vfilt) 2941
38.4.11.2.2 Motion Calculator Block (di_mcalc) 2942
38.4.11.2.3 Spatial Motion Filter (di_sfilt) 2943
38.4.11.2.4 Interpolated Pixel Calculator Block (di_interp) 2943
38.4.11.2.5 Median Filter Block (di_med) 2943
38.4.11.2.6 Soft Switch Block (di_sswitch) 2943
38.4.11.3 DMA only Mode 2944
38.4.11.4 Real Time Mode 2944
38.4.11.5 CSI only Mode 2944
38.4.11.6 Using Combining in the VDIC 2944
38.4.11.7 VDIC Restrictions 2945
38.4.12 Control Module (CM) 2945
38.4.12.1 Block Diagram 2946
38.4.12.2 Frame Synchronization Unit 2947
38.4.12.2.1 General Description 2947
38.4.12.2.2 Frame Synchronization Flow 2947
38.4.12.2.3 FSU's fundamentals 2948
38.4.12.2.4 IPU main flows 2951
38.4.12.2.5 Sub-Frame Double-Buffering (Band Mode) 2959
38.4.12.2.6 Automatic Window Refresh 2960
38.4.12.2.7 VPU Sub Frame Synchronization 2960
38.4.12.2.8 IPU VDOA synchronization 2960
38.4.12.3 Interrupt Generator 2961
38.4.12.4 SDMA event generator 2967
38.4.12.5 General Configuration Registers 2968

38.4.12.6 Shadow Registers Module (SRM)..... 2968
38.4.12.6.1 Switching between 2 flows..... 2968
38.4.12.6.2 Updating parameters between frames..... 2969
38.4.12.6.3 Updating the memory..... 2969
38.4.12.6.4 SRM priority..... 2970
38.4.12.6.5 SRM entries mapping..... 2970
38.4.12.7 Memory Access Unit..... 2987
38.4.12.8 SISG - Still Image Synchronization Generator..... 2988
38.4.12.9 Clock Change procedure..... 2990
38.4.12.10 Low Power Modes..... 2991
38.4.12.10.1 STOP Mode..... 2992
38.4.12.10.2 Low Power Screen Refresh mode - LPSR..... 2993
38.5 IPU Memory Map/Register Definition..... 2995
38.5.1 Configuration Register (IPU_CONF)..... 3014
38.5.2 SISG Control 0 Register (IPU_SISG_CTRL0)..... 3017
38.5.3 SISG Control 1 Register (IPU_SISG_CTRL1)..... 3018
38.5.4 SISG Set<i> Register (IPU_SISG_SET_i)..... 3018
38.5.5 SISG Clear <i> Register (IPU_SISG_CLR_i)..... 3019
38.5.6 Interrupt Control Register 1 (IPU_INT_CTRL_1)..... 3020
38.5.7 Interrupt Control Register 2 (IPU_INT_CTRL_2)..... 3024
38.5.8 Interrupt Control Register 3 (IPU_INT_CTRL_3)..... 3026
38.5.9 Interrupt Control Register 4 (IPU_INT_CTRL_4)..... 3030
38.5.10 Interrupt Control Register 5 (IPU_INT_CTRL_5)..... 3033
38.5.11 Interrupt Control Register 6 (IPU_INT_CTRL_6)..... 3038
38.5.12 Interrupt Control Register 7 (IPU_INT_CTRL_7)..... 3041
38.5.13 Interrupt Control Register 8 (IPU_INT_CTRL_8)..... 3043
38.5.14 Interrupt Control Register 9 (IPU_INT_CTRL_9)..... 3045
38.5.15 Interrupt Control Register 10 (IPU_INT_CTRL_10)..... 3047
38.5.16 Interrupt Control Register 11 (IPU_INT_CTRL_11)..... 3049

38.5.17 Interrupt Control Register 12 (IPU_INT_CTRL_12)..... 3052
38.5.18 Interrupt Control Register 13 (IPU_INT_CTRL_13)..... 3054
38.5.19 Interrupt Control Register 14 (IPU_INT_CTRL_14)..... 3058
38.5.20 Interrupt Control Register15 (IPU_INT_CTRL_15)..... 3061
38.5.21 SDMA Event Control Register 1 (IPU_SDMA_EVENT_1)..... 3065
38.5.22 SDMA Event Control Register 2 (IPU_SDMA_EVENT_2)..... 3069
38.5.23 SDMA Event Control Register 3 (IPU_SDMA_EVENT_3)..... 3072
38.5.24 SDMA Event Control Register 4 (IPU_SDMA_EVENT_4)..... 3077
38.5.25 SDMA Event Control Register 7 (IPU_SDMA_EVENT_7)..... 3080
38.5.26 SDMA Event Control Register 8 (IPU_SDMA_EVENT_8)..... 3082
38.5.27 SDMA Event Control Register 11 (IPU_SDMA_EVENT_11)..... 3083
38.5.28 SDMA Event Control Register 12 (IPU_SDMA_EVENT_12)..... 3086
38.5.29 SDMA Event Control Register 13 (IPU_SDMA_EVENT_13)..... 3088
38.5.30 SDMA Event Control Register 14 (IPU_SDMA_EVENT_14)..... 3092
38.5.31 Shadow Registers Memory Priority 1 Register (IPU_SRM_PRI1)..... 3095
38.5.32 Shadow Registers Memory Priority 2 Register (IPU_SRM_PRI2)..... 3096
38.5.33 FSU Processing Flow 1 Register (IPU_FS_PROC_FLOW1)..... 3098
38.5.34 FSU Processing Flow 2 Register (IPU_FS_PROC_FLOW2)..... 3102
38.5.35 FSU Processing Flow 3 Register (IPU_FS_PROC_FLOW3)..... 3105
38.5.36 FSU Displaying Flow 1 Register (IPU_FS_DISP_FLOW1)..... 3108
38.5.37 FSU Displaying Flow 2 Register (IPU_FS_DISP_FLOW2)..... 3111
38.5.38 SKIP Register (IPU_SKIP)..... 3113
38.5.39 Display Alternate Configuration Register (IPU_DISP_ALT_CONF)..... 3115
38.5.39 Display General Control Register (IPU_DISP_GEN)..... 3116
38.5.40 Display Alternate Flow Control Register 1 (IPU_DISP_ALT1)..... 3119
38.5.41 Display Alternate Flow Control Register 2 (IPU_DISP_ALT2)..... 3120
38.5.42 Display Alternate Flow Control Register 3 (IPU_DISP_ALT3)..... 3121
38.5.43 Display Alternate Flow Control Register 4 (IPU_DISP_ALT4)..... 3123
38.5.44 Memory Reset Control Register (IPU_MEM_RST)..... 3124

38.5.45 Power Modes Control Register (IPU_PM)....3126
38.5.46 General Purpose Register (IPU_GPR)....3129
38.5.47 Channel Double Buffer Mode Select 0 Register (IPU_CH_DB_MODE_SEL0)....3131
38.5.48 Channel Double Buffer Mode Select 1 Register (IPU_CH_DB_MODE_SEL1)....3135
38.5.49 Alternate Channel Double Buffer Mode Select 0 Register (IPU_ALT_CH_DB_MODE_SEL0)....3138
38.5.50 Alternate Channel Double Buffer Mode Select1 Register (IPU_ALT_CH_DB_MODE_SEL1)....3140
38.5.51 Alternate Channel Triple Buffer Mode Select 0 Register (IPU_ALT_CH_TRB_MODE_SEL0)....3141
38.5.52 Alternate Channel Triple Buffer Mode Select 1 Register (IPU_ALT_CH_TRB_MODE_SEL1)....3143
38.5.52 Interrupt Status Register 1 (IPU_INT_STAT_1)....3144
38.5.53 Interrupt Status Register2 (IPU_INT_STAT_2)....3149
38.5.54 Interrupt Status Register 3 (IPU_INT_STAT_3)....3152
38.5.55 Interrupt Status Register 4 (IPU_INT_STAT_4)....3156
38.5.56 Interrupt Status Register 5 (IPU_INT_STAT_5)....3159
38.5.57 Interrupt Status Register 6 (IPU_INT_STAT_6)....3165
38.5.58 Interrupt Status Register7 1 (IPU_INT_STAT_7)....3169
38.5.59 Interrupt Status Register 8 (IPU_INT_STAT_8)....3172
38.5.60 Interrupt Status Register 9 (IPU_INT_STAT_9)....3175
38.5.61 Interrupt Status Register 10 (IPU_INT_STAT_10)....3177
38.5.62 Interrupt Status Register 11 (IPU_INT_STAT_11)....3180
38.5.63 Interrupt Status Register 12 (IPU_INT_STAT_12)....3184
38.5.64 Interrupt Status Register 13 (IPU_INT_STAT_13)....3186
38.5.65 Interrupt Status Register 14 (IPU_INT_STAT_14)....3191
38.5.66 Interrupt Status Register 15 (IPU_INT_STAT_15)....3194
38.5.67 Current Buffer Register 0 (IPU_CUR_BUF_0)....3198
38.5.68 Current Buffer Register 1 (IPU_CUR_BUF_1)....3203
38.5.69 Alternate Current Buffer Register 0 (IPU_ALT_CUR_0)....3207
38.5.70 Alternate Current Buffer Register 1 (IPU_ALT_CUR_1)....3209
38.5.71 Shadow Registers Memory Status Register (IPU_SRM_STAT)....3212
38.5.72 Processing Status Tasks Register (IPU_PROC_TASKS_STAT)....3214

38.5.73 Display Tasks Status Register (IPU_DISP_TASKS_STAT)..... 3216
38.5.74 Triple Current Buffer Register 0 (IPU_TRIPLE_CUR_BUF_0)..... 3218
38.5.75 Triple Current Buffer Register 1 (IPU_TRIPLE_CUR_BUF_1)..... 3220
38.5.76 Triple Current Buffer Register 2 (IPU_TRIPLE_CUR_BUF_2)..... 3221
38.5.76 Triple Current Buffer Register 3 (IPU_TRIPLE_CUR_BUF_3)..... 3222
38.5.76 IPU Channels Buffer 0 Ready 0 Register (IPU_CH_BUF0_RDY0)..... 3222
38.5.77 IPU Channels Buffer 0 Ready 1 Register (IPU_CH_BUF0_RDY1)..... 3226
38.5.78 IPU Channels Buffer 1 Ready 0 Register (IPU_CH_BUF1_RDY0)..... 3228
38.5.79 IPU Channels Buffer 1 Ready 1Register (IPU_CH_BUF1_RDY1)..... 3231
38.5.80 IPU Alternate Channels Buffer 0 Ready 0 Register (IPU_ALT_CH_BUF0_RDY0).....3234
38.5.81 IPU Alternate Channels Buffer 0 Ready 1 Register (IPU_ALT_CH_BUF0_RDY1).....3235
38.5.82 IPU Alternate Channels Buffer1 Ready 0 Register (IPU_ALT_CH_BUF1_RDY0).....3236
38.5.83 IPU Alternate Channels Buffer 1 Ready 1 Register (IPU_ALT_CH_BUF1_RDY1).....3237
38.5.84 IPU Channels Buffer 2 Ready 0 Register (IPU_CH_BUF2_RDY0)..... 3238
38.5.85 IPU Channels Buffer 2 Ready 1 Register (IPU_CH_BUF2_RDY1)..... 3240
38.5.86 IDMAC Configuration Register (IPU_IDMAC_CONF)..... 3241
38.5.87 IDMAC Channel Enable 1 Register (IPU_IDMAC_CH_EN_1)..... 3243
38.5.88 IDMAC Channel Enable 2 Register (IPU_IDMAC_CH_EN_2)..... 3246
38.5.89 IDMAC Separate Alpha Indication Register (IPU_IDMAC_SEP_ALPHA).....3248
38.5.90 IDMAC Alternate Separate Alpha Indication Register (IPU_IDMAC_ALT_SEP_ALPHA).....3250
38.5.91 IDMAC Channel Priority 1 Register (IPU_IDMAC_CH_PRI_1)..... 3252
38.5.92 IDMAC Channel Priority 2 Register (IPU_IDMAC_CH_PRI_2)..... 3255
38.5.93 IDMAC Channel Watermark Enable 1 Register (IPU_IDMAC_WM_EN_1).....3257
38.5.94 IDMAC Channel Watermark Enable 2 Register (IPU_IDMAC_WM_EN_2).....3259
38.5.95 IDMAC Channel Lock Enable 1Register (IPU_IDMAC_LOCK_EN_1).....3260
38.5.96 IDMAC Channel Lock Enable 2Register (IPU_IDMAC_LOCK_EN_2).....3262
38.5.97 IDMAC Channel Alternate Address 0 Register (IPU_IDMAC_SUB_ADDR_0)..... 3263
38.5.98 IDMAC Channel Alternate Address 1 Register (IPU_IDMAC_SUB_ADDR_1)..... 3264
38.5.99 IDMAC Channel Alternate Address 2 Register (IPU_IDMAC_SUB_ADDR_2)..... 3265

38.5.100 IDMAC Channel Alternate Address 3 Register (IPU_IDMAC_SUB_ADDR_3)..... 3266
38.5.101 IDMAC Channel Alternate Address 4 Register (IPU_IDMAC_SUB_ADDR_4)..... 3268
38.5.102 IDMAC Band Mode Enable 1 Register (IPU_IDMAC_BNDM_EN_1)..... 3269
38.5.103 IDMAC Band Mode Enable 2 Register (IPU_IDMAC_BNDM_EN_2)..... 3272
38.5.104 IDMAC Scroll Coordinations Register (IPU_IDMAC_SC_CORD)..... 3273
38.5.105 IDMAC Scroll Coordinations Register 1 (IPU_IDMAC_SC_CORD_1)..... 3274
38.5.106 IDMAC Channel Busy 1 Register (IPU_IDMAC_CH_BUSY_1).....3275
38.5.107 IDMAC Channel Busy 2 Register (IPU_IDMAC_CH_BUSY_2).....3281
38.5.108 DP Common Configuration Sync Flow Register (IPU_DP_COM_CONF_SYNC)..... 3285
38.5.109 DP Graphic Window Control Sync Flow Register (IPU_DP_Graph_Wind_CTRL_SYNC)..... 3287
38.5.110 DP Partial Plane Window Position Sync Flow Register (IPU_DP_FG_POS_SYNC)..... 3288
38.5.111 DP Cursor Position and Size Sync Flow Register (IPU_DP_CUR_POS_SYNC).....3288
38.5.112 DP Color Cursor Mapping Sync Flow Register (IPU_DP_CUR_MAP_SYNC).....3289
38.5.113 DP Gamma Constants Sync Flow Register i (IPU_DP_GAMMA_C_SYNC_i)..... 3290
38.5.114 DP Gamma Correction Slope Sync Flow Register i (IPU_DP_GAMMA_S_SYNC_i)..... 3291
38.5.115 DP Color Space Conversion Control Sync Flow Registers (IPU_DP_CSCA_SYNC_i).....3291
38.5.116 DP Color Conversion Control Sync Flow Register 0 (IPU_DP_SCS_SYNC_0)..... 3292
38.5.117 DP Color Conversion Control Sync Flow Register 1 (IPU_DP_SCS_SYNC_1)..... 3293
38.5.118 DP Cursor Position and Size Alternate Register (IPU_DP_CUR_POS_ALT)..... 3294
38.5.119 DP Common Configuration Async 0 Flow Register (IPU_DP_COM_CONF_ASYNC0).....3295
38.5.120 DP Graphic Window Control Async 0 Flow Register (IPU_DP_GRAPH_WIND_CTRL_ASYNC0)... 3297
38.5.121 DP Partial Plane Window Position Async 0 Flow Register (IPU_DP_FG_POS_ASYNC0)..... 3298
38.5.122 DP Cursor Position and Size Async 0 Flow Register (IPU_DP_CUR_POS_ASYNC0).....3298
38.5.123 DP Color Cursor Mapping Async 0 Flow Register (IPU_DP_CUR_MAP_ASYNC0).....3299
38.5.124 DP Gamma Constant Async 0 Flow Register i (IPU_DP_GAMMA_C_ASYNC0_i)..... 3300
38.5.125 DP Gamma Correction Slope Async 0 Flow Register i (IPU_DP_GAMMA_S_ASYNC0_i).....3301
38.5.126 DP Color Space Conversion Control Async 0 Flow Register i (IPU_DP_CSCA_ASYNC0_i)..... 3301
38.5.127 DP Color Conversion Control Async 0 Flow Register 0 (IPU_DP_CSC_ASYNC0_0).....3302
38.5.128 DP Color Conversion Control Async 1 Flow Register (IPU_DP_CSC_ASYNC_1).....3303

38.5.129 DP Common Configuration Async 1 Flow Register (IPU_DP_COM_CONF_ASYNC1)..... 3304
38.5.130 DP Debug Control Register (IPU_DP_DEBUG_CNT)..... 3306
38.5.131 DP Graphic Window Control Async 1 Flow Register (IPU_DP_GRAPH_WIND_CTRL_ASYNC1)... 3307
38.5.132 DP Debug Status Register (IPU_DP_DEBUG_STAT)..... 3308
38.5.133 DP Partial Plane Window Position Async 1 Flow Register (IPU_DP_FG_POS_ASYNC1)..... 3310
38.5.134 DP Cursor Postion and Size Async 1 Flow Register (IPU_DP_CUR_POS_ASYNC1)..... 3310
38.5.135 DP Color Cursor Mapping Async 1 Flow Register (IPU_DP_CUR_MAP_ASYNC1)..... 3311
38.5.136 DP Gamma Constants Async 1 Flow Register i (IPU_DP_GAMMA_C_ASYNC1_i)..... 3312
38.5.137 DP Gamma Correction Slope Async 1 Flow Register i (IPU_DP_GAMMA_S_ASYN1_i)..... 3313
38.5.138 DP Color Space Converstion Control Async 1 Flow Register i (IPU_DP_CSCA_ASYNC1_i)..... 3313
38.5.139 DP Color Conversion Control Async 1 Flow Register 0 (IPU_DP_CSC_ASYNC1_0)..... 3314
38.5.140 DP Color Conversion Control Async 1 Flow Register 1 (IPU_DP_CSC_ASYNC1_1)..... 3315
38.5.141 IC Configuration Register (IPU_IC_CONF)..... 3316
38.5.142 IC Preprocessing Encoder Resizing Coefficients Register (IPU_IC_PRP_ENC_RSC)..... 3318
38.5.143 IC Preprocessing View-Finder Resizing Coefficients Register (IPU_IC_PRP_VF_RSC)..... 3319
38.5.144 IC Postprocessing Encoder Resizing Coefficients Register (IPU_IC_PP_RSC)..... 3320
38.5.145 IC Combining Parameters Register 1 (IPU_IC_CMBP_1)..... 3321
38.5.146 IC Combining Parameters Register 2 (IPU_IC_CMBP_2)..... 3321
38.5.147 IC IDMAC Parameters 1 Register (IPU_IC_IDMAC_1)..... 3322
38.5.148 IC IDMAC Parameters 2 Register (IPU_IC_IDMAC_2)..... 3325
38.5.149 IC IDMAC Parameters 3Register (IPU_IC_IDMAC_3)..... 3326
38.5.150 IC IDMAC Parameters 4 Register (IPU_IC_IDMAC_4)..... 3326
38.5.151 CSI0 Sensor Configuration Register (IPU_CSI0_SENS_CONF)..... 3327
38.5.152 CSI0 Sense Frame Size Register (IPU_CSI0_SENS_FRM_SIZE)..... 3330
38.5.153 CSI0 Actual Frame Size Register (IPU_CSI0_ACT_FRM_SIZE)..... 3330
38.5.154 CSI0 Output Control Register (IPU_CSI0_OUT_FRM_CTRL)..... 3331
38.5.155 CSI0 Test Control Register (IPU_CSI0_TST_CTRL)..... 3332
38.5.156 CSI0 CCIR Code Register 1 (IPU_CSI0_CCIR_CODE_1)..... 3333
38.5.157 CSI0 CCIR Code Register 2 (IPU_CSI0_CCIR_CODE_2)..... 3334

38.5.158 CSI0 CCIR Code Register 3 (IPU_CSI0_CCIR_CODE_3)....3335
38.5.159 CSI0 Data Identifier Register (IPU_CSI0_DI)....3335
38.5.160 CSI0 SKIP Register (IPU_CSI0_SKIP)....3336
38.5.161 CSI0 Compander Control Register (IPU_CSI0_CPD_CTRL)....3337
38.5.162 CSI0 Red Component Compander Constants Register <i> (IPU_CSI0_CPD_RC_i)....3338
38.5.163 CSI0 Red Component Compander SLOPE Register <i> (IPU_CSI0_CPD_RS_i)....3339
38.5.164 CSI0 GR Component Compander Constants Register <i> (IPU_CSI0_CPD_GRC_i)....3339
38.5.165 CSI0 GR Component Compander SLOPE Register <i> (IPU_CSI0_CPD_GRS_i)....3340
38.5.166 CSI0 GB Component Compander Constants Register <i> (IPU_CSI0_CPD_GBC_i)....3341
38.5.167 CSI0 GB Component Compander SLOPE Register <i> (IPU_CSI0_CPD_GBS_i)....3341
38.5.168 CSI0 Blue Component Compander Constants Register <i> (IPU_CSI0_CPD_BC_i)....3342
38.5.169 CSI0 Blue Component Compander SLOPE Register <i> (IPU_CSI0_CPD_BS_i)....3343
38.5.170 CSI0 Compander Offset Register 1 (IPU_CSI0_CPD_OFFSET1)....3343
38.5.171 CSI0 Compander Offset Register 2 (IPU_CSI0_CPD_OFFSET2)....3344
38.5.172 CSI1 Sensor Configuration Register (IPU_CSI1_SENS_CONF)....3345
38.5.173 CSI1 Sense Frame Size Register (IPU_CSI1_SENS_FRM_SIZE)....3347
38.5.174 CSI1 Actual Frame Size Register (IPU_CSI1_ACT_FRM_SIZE)....3348
38.5.175 CSI1 Output Control Register (IPU_CSI1_OUT_FRM_CTRL)....3349
38.5.176 CSI1 Test Control Register (IPU_CSI1_TST_CTRL)....3350
38.5.177 CSI1 CCIR Code Register 1 (IPU_CSI1_CCIR_CODE_1)....3351
38.5.178 CSI1 CCIR Code Register 2 (IPU_CSI1_CCIR_CODE_2)....3352
38.5.179 CSI1 CCIR Code Register 3 (IPU_CSI1_CCIR_CODE_3)....3353
38.5.180 CSI1 Data Identifier Register (IPU_CSI1_DI)....3353
38.5.181 CSI1 SKIP Register (IPU_CSI1_SKIP)....3354
38.5.182 CSI1 Compander Control Register (IPU_CSI1_CPD_CTRL)....3355
38.5.183 CSI1 Red Component Compander Constants Register <i> (IPU_CSI1_CPD_RC_i)....3356
38.5.184 CSI1 Red Component Compander SLOPE Register <i> (IPU_CSI1_CPD_RS_i)....3356
38.5.185 CSI1 GR Component Compander Constants Register <i> (IPU_CSI1_CPD_GRC_i)....3357
38.5.186 CSI1 GR Component Compander SLOPE Register <i> (IPU_CSI1_CPD_GRS_i)....3358

38.5.187 CSI1 GB Component Compander Constants Register <i> (IPU_CSI1_CPD_GBC_i)..... 3358
38.5.188 CSI1 GB Component Compander SLOPE Register <i> (IPU_CSI1_CPD_GBS_i)..... 3359
38.5.189 CSI1 Blue Component Compander Constants Register <i> (IPU_CSI1_CPD_BC_i)..... 3360
38.5.190 CSI1 Blue Component Compander SLOPE Register <i> (IPU_CSI1_CPD_BS_i)..... 3360
38.5.191 CSI1 Compander Offset Register 1 (IPU_CSI1_CPD_OFFSET1)..... 3361
38.5.192 CSI1 Compander Offset Register 2 (IPU_CSI1_CPD_OFFSET2)..... 3362
38.5.193 DI0 General Register (IPU_DI0_GENERAL)..... 3363
38.5.194 DI0 Base Sync Clock Gen 0 Register (IPU_DI0_BS_CLKGEN0)..... 3365
38.5.195 DI0 Base Sync Clock Gen 1 Register (IPU_DI0_BS_CLKGEN1)..... 3366
38.5.196 DI0 Sync Wave Gen 1 Register 0 (IPU_DI0_SW_GEN0_1)..... 3367
38.5.197 DI0 Sync Wave Gen 2 Register 0 (IPU_DI0_SW_GEN0_2)..... 3368
38.5.198 DI0 Sync Wave Gen 3 Register 0 (IPU_DI0_SW_GEN0_3)..... 3369
38.5.199 DI0 Sync Wave Gen 4 Register 0 (IPU_DI0_SW_GEN0_4)..... 3370
38.5.200 DI0 Sync Wave Gen 5 Register 0 (IPU_DI0_SW_GEN0_5)..... 3372
38.5.201 DI0 Sync Wave Gen 6 Register 0 (IPU_DI0_SW_GEN0_6)..... 3373
38.5.202 DI0 Sync Wave Gen 7 Register 0 (IPU_DI0_SW_GEN0_7)..... 3374
38.5.203 DI0 Sync Wave Gen 8 Register 0 (IPU_DI0_SW_GEN0_8)..... 3375
38.5.204 DI0 Sync Wave Gen 9 Register 0 (IPU_DI0_SW_GEN0_9)..... 3376
38.5.205 DI0 Sync Wave Gen 1 Register 1 (IPU_DI0_SW_GEN1_1)..... 3378
38.5.206 DI0 Sync Wave Gen 2 Register 1 (IPU_DI0_SW_GEN1_2)..... 3380
38.5.207 DI0 Sync Wave Gen 3 Register 1 (IPU_DI0_SW_GEN1_3)..... 3382
38.5.208 DI0 Sync Wave Gen 4 Register 1 (IPU_DI0_SW_GEN1_4)..... 3384
38.5.209 DI0 Sync Wave Gen 5 Register 1 (IPU_DI0_SW_GEN1_5)..... 3386
38.5.210 DI0 Sync Wave Gen 6 Register 1 (IPU_DI0_SW_GEN1_6)..... 3388
38.5.211 DI0 Sync Wave Gen 7 Register 1 (IPU_DI0_SW_GEN1_7)..... 3390
38.5.212 DI0 Sync Wave Gen 8 Register 1 (IPU_DI0_SW_GEN1_8)..... 3392
38.5.213 DI0 Sync Wave Gen 9 Register 1 (IPU_DI0_SW_GEN1_9)..... 3394
38.5.214 DI0 Sync Assistance Gen Register (IPU_DI0_SYNC_AS_GEN)..... 3395
38.5.215 DI0 Data Wave Gen <i> Register (IPU_DI0_DW_GEN_i)..... 3396

38.5.216 DI0 Data Wave Set 0 <i> Register (IPU_DI0_DW_SET0_i)..... 3398
38.5.217 DI0 Data Wave Set 1 <i> Register (IPU_DI0_DW_SET1_i)..... 3398
38.5.218 DI0 Data Wave Set 2 <i> Register (IPU_DI0_DW_SET2_i)..... 3399
38.5.219 DI0 Data Wave Set 3 <i> Register (IPU_DI0_DW_SET3_i)..... 3399
38.5.220 DI0 Step Repeat <i> Registers (IPU_DI0_STP_REP_i)..... 3400
38.5.221 DI0 Step Repeat 9 Registers (IPU_DI0_STP_REP_9)..... 3400
38.5.222 DI0 Serial Display Control Register (IPU_DI0_SER_CONF)..... 3401
38.5.223 DI0 Special Signals Control Register (IPU_DI0_SSC)..... 3403
38.5.224 DI0 Polarity Register (IPU_DI0_POL)..... 3406
38.5.225 DI0 Active Window 0 Register (IPU_DI0_AW0)..... 3407
38.5.226 DI0 Active Window 1 Register (IPU_DI0_AW1)..... 3408
38.5.227 DI0 Screen Configuration Register (IPU_DI0_SCR_CONF)..... 3409
38.5.228 DI0 Status Register (IPU_DI0_STAT)..... 3410
38.5.229 DI1General Register (IPU_DI1_GENERAL)..... 3412
38.5.230 DI1 Base Sync Clock Gen 0 Register (IPU_DI1_BS_CLKGEN0)..... 3414
38.5.231 DI1 Base Sync Clock Gen 1 Register (IPU_DI1_BS_CLKGEN1)..... 3415
38.5.232 DI1 Sync Wave Gen 1 Register 0 (IPU_DI1_SW_GEN0_1)..... 3416
38.5.233 DI1 Sync Wave Gen 2 Register 0 (IPU_DI1_SW_GEN0_2)..... 3417
38.5.234 DI1 Sync Wave Gen 3 Register 0 (IPU_DI1_SW_GEN0_3)..... 3418
38.5.235 DI1 Sync Wave Gen 4 Register 0 (IPU_DI1_SW_GEN0_4)..... 3419
38.5.236 DI1 Sync Wave Gen 5 Register 0 (IPU_DI1_SW_GEN0_5)..... 3421
38.5.237 DI1 Sync Wave Gen 6 Register 0 (IPU_DI1_SW_GEN0_6)..... 3422
38.5.238 DI1 Sync Wave Gen 7 Register 0 (IPU_DI1_SW_GEN0_7)..... 3423
38.5.239 DI1 Sync Wave Gen 8 Register 0 (IPU_DI1_SW_GEN0_8)..... 3424
38.5.240 DI1Sync Wave Gen 9 Register 0 (IPU_DI1_SW_GEN0_9)..... 3425
38.5.241 DI1 Sync Wave Gen 1 Register 1 (IPU_DI1_SW_GEN1_1)..... 3427
38.5.242 DI1 Sync Wave Gen 2 Register 1 (IPU_DI1_SW_GEN1_2)..... 3429
38.5.243 DI1 Sync Wave Gen 3 Register 1 (IPU_DI1_SW_GEN1_3)..... 3431
38.5.244 DI1 Sync Wave Gen 4 Register 1 (IPU_DI1_SW_GEN1_4)..... 3433

38.5.245 DI1 Sync Wave Gen 5 Register 1 (IPU_DI1_SW_GEN1_5)....3435
38.5.246 DI1 Sync Wave Gen 6 Register 1 (IPU_DI1_SW_GEN1_6)....3437
38.5.247 DI1Sync Wave Gen 7 Register 1 (IPU_DI1_SW_GEN1_7)....3439
38.5.248 DI1 Sync Wave Gen 8 Register 1 (IPU_DI1_SW_GEN1_8)....3441
38.5.249 DI1 Sync Wave Gen 9 Register 1 (IPU_DI1_SW_GEN1_9)....3443
38.5.250 DI1 Sync Assistance Gen Register (IPU_DI1_SYNC_AS_GEN).... 3444
38.5.251 DI1 Data Wave Gen <i> Register (IPU_DI1_DW_GEN_i)....3445
38.5.252 DI1 Data Wave Set 0 <i> Register (IPU_DI1_DW_SET0_i).... 3447
38.5.253 DI1 Data Wave Set 1 <i> Register (IPU_DI1_DW_SET1_i).... 3447
38.5.254 DI1 Data Wave Set 2 <i> Register (IPU_DI1_DW_SET2_i).... 3448
38.5.255 DI1 Data Wave Set 3 <i> Register (IPU_DI1_DW_SET3_i).... 3448
38.5.256 DI1 Step Repeat <i> Registers (IPU_D1_STP_REP_i).... 3449
38.5.257 DI1Step Repeat 9 Registers (IPU_DI1_STP_REP_9)....3449
38.5.258 DI1 Serial Display Control Register (IPU_DI1_SER_CONF)....3450
38.5.259 DI1 Special Signals Control Register (IPU_DI1_SSC)....3452
38.5.260 DI1 Polarity Register (IPU_DI1_POL)....3455
38.5.261 DI1Active Window 0 Register (IPU_DI1_AW0)....3456
38.5.262 DI1 Active Window 1 Register (IPU_DI1_AW1)....3457
38.5.263 DI1 Screen Configuration Register (IPU_DI1_SCR_CONF).... 3458
38.5.264 DI1 Status Register (IPU_DI1_STAT).... 3459
38.5.265 SMFC Mapping Register (IPU_SMFC_MAP)....3460
38.5.266 SMFC Watermark Control Register (IPU_SMFC_WMC)....3461
38.5.267 SMFC Burst Size Register (IPU_SMFC_BS).... 3463
38.5.268 DC Read Channel Configuration Register (IPU_DC_READ_CH_CONF).... 3464
38.5.269 DC Read Channel Start Address Register (IPU_DC_READ_SH_ADDR)....3466
38.5.270 DC Routine Link Register 0 Channel 0 (IPU_DC_RL0_CH_0)....3466
38.5.271 DC Routine Link Register 1 Channel 0 (IPU_DC_RL1_CH_0)....3467
38.5.272 DC Routine Link Register2 Channel 0 (IPU_DC_RL2_CH_0)....3468
38.5.273 DC Routine Link Registe3 Channel 0 (IPU_DC_RL3_CH_0).... 3469

38.5.274 DC Routine Link Register 4 Channel 0 (IPU_DC_RL4_CH_0)....3470
38.5.275 DC Write Channel 1 Configuration Register (IPU_DC_WR_CH_CONF_1)....3471
38.5.276 DC Write Channel 1 Address Configuration Register (IPU_DC_WR_CH_ADDR_1)....3473
38.5.277 DC Routine Link Register 0 Channel 1 (IPU_DC_RL0_CH_1)....3473
38.5.278 DC Routine Link Register 1 Channel 1 (IPU_DC_RL1_CH_1)....3474
38.5.279 DC Routine Link Register 2 Channel 1 (IPU_DC_RL2_CH_1)....3475
38.5.280 DC Routine Link Register 3 Channel 1 (IPU_DC_RL3_CH_1)....3476
38.5.281 DC Routine Link Register 4 Channel 1 (IPU_DC_RL4_CH_1)....3477
38.5.282 DC Write Channel 2 Configuration Register (IPU_DC_WR_CH_CONF_2)....3478
38.5.283 DC Write Channel 2 Address Configuration Register (IPU_DC_WR_CH_ADDR_2)....3479
38.5.284 DC Routine Link Register 0 Channel 2 (IPU_DC_RL0_CH_2)....3480
38.5.285 DC Routine Link Register 1 Channel 2 (IPU_DC_RL1_CH_2)....3481
38.5.286 DC Routine Link Register 2 Channel 2 (IPU_DC_RL2_CH_2)....3482
38.5.287 DC Routine Link Register 3 Channel 2 (IPU_DC_RL3_CH_2)....3483
38.5.288 DC Routine Link Register 4 Channel 2 (IPU_DC_RL4_CH_2)....3484
38.5.289 DC Command Channel 3 Configuration Register (IPU_DC_CMD_CH_CONF_3)....3485
38.5.290 DC Command Channel 4 Configuration Register (IPU_DC_CMD_CH_CONF_4)....3486
38.5.291 DC Write Channel 5Configuration Register (IPU_DC_WR_CH_CONF_5)....3487
38.5.292 DC Write Channel 5Address Configuration Register (IPU_DC_WR_CH_ADDR_5)....3488
38.5.293 DC Routine Link Register 0 Channel 5 (IPU_DC_RL0_CH_5)....3489
38.5.294 DC Routine Link Register 1 Channel 5 (IPU_DC_RL1_CH_5)....3490
38.5.295 DC Routine Link Register 2 Channel 5 (IPU_DC_RL2_CH_5)....3491
38.5.296 DC Routine Link Register3 Channel 5 (IPU_DC_RL3_CH_5)....3492
38.5.297 DC Routine Link Register 4 Channel 5 (IPU_DC_RL4_CH_5)....3493
38.5.298 DC Write Channel 6 Configuration Register (IPU_DC_WR_CH_CONF_6)....3494
38.5.299 DC Write Channel 6 Address Configuration Register (IPU_DC_WR_CH_ADDR_6)....3495
38.5.300 DC Routine Link Register 0Channel 6 (IPU_DC_RL0_CH_6)....3496
38.5.301 DC Routine Link Register 1 Channel 6 (IPU_DC_RL1_CH_6)....3497
38.5.302 DC Routine Link Register 2 Channel 6 (IPU_DC_RL2_CH_6)....3498

38.5.303 DC Routine Link Register 3 Channel 6 (IPU_DC_RL3_CH_6)....3499
38.5.304 DC Routine Link Register 4 Channel 6 (IPU_DC_RL4_CH_6)....3500
38.5.305 DC Write Channel 8 Configuration 1Register (IPU_DC_WR_CH_CONF1_8)....3501
38.5.306 DC Write Channel 8 Configuration 2 Register (IPU_DC_WR_CH_CONF2_8)....3502
38.5.307 DC Routine Link Register 1 Channel 8 (IPU_DC_RL1_CH_8)....3502
38.5.308 DC Routine Link Register 2 Channel 8 (IPU_DC_RL2_CH_8)....3503
38.5.309 DC Routine Link Register 3 Channel 8 (IPU_DC_RL3_CH_8)....3504
38.5.310 DC Routine Link Register 4 Channel 8 (IPU_DC_RL4_CH_8)....3504
38.5.311 DC Routine Link Register 5 Channel 8 (IPU_DC_RL5_CH_8)....3505
38.5.312 DC Routine Link Register 6 Channel 8 (IPU_DC_RL6_CH_8)....3506
38.5.313 DC Write Channel 9 Configuration 1Register (IPU_DC_WR_CH_CONF1_9)....3506
38.5.314 DC Write Channel 9Configuration 2Register (IPU_DC_WR_CH_CONF2_9)....3507
38.5.315 DC Routine Link Register 1 Channel 9 (IPU_DC_RL1_CH_9)....3508
38.5.316 DC Routine Link Register 2 Channel 9 (IPU_DC_RL2_CH_9)....3509
38.5.317 DC Routine Link Register 3Channel 9 (IPU_DC_RL3_CH_9)....3510
38.5.318 DC Routine Link Register 4 Channel 9 (IPU_DC_RL4_CH_9)....3510
38.5.319 DC Routine Link Register 5 Channel 9 (IPU_DC_RL5_CH_9)....3511
38.5.320 DC Routine Link Register 6 Channel 9 (IPU_DC_RL6_CH_9)....3512
38.5.321 DC General Register (IPU_DC_GEN)....3512
38.5.322 DC Display Configuration 1 Register 0 (IPU_DC_DISP_CONF1_0)....3514
38.5.323 DC Display Configuration 1 Register 1 (IPU_DC_DISP_CONF1_1)....3515
38.5.324 DC Display Configuration 1 Register 2 (IPU_DC_DISP_CONF1_2)....3517
38.5.325 DC Display Configuration 1 Register 3 (IPU_DC_DISP_CONF1_3)....3518
38.5.326 DC Display Configuration 2 Register 0 (IPU_DC_DISP_CONF2_0)....3519
38.5.327 DC Display Configuration 2 Register 1 (IPU_DC_DISP_CONF2_1)....3520
38.5.328 DC Display Configuration 2 Register 2 (IPU_DC_DISP_CONF2_2)....3520
38.5.329 DC Display Configuration 2 Register 3 (IPU_DC_DISP_CONF2_3)....3520
38.5.330 DC DI0Configuration Register 1 (IPU_DC_DI0_CONF_1)....3521
38.5.331 DC DI0Configuration Register 2 (IPU_DC_DI0_CONF_2)....3521

38.5.332 DC DI1Configuration Register 1 (IPU_DC_DI1_CONF_1).....3521
38.5.333 DC DI1Configuration Register 2 (IPU_DC_DI1_CONF_2).....3522
38.5.334 DC Mapping Configuration Register 0 (IPU_DC_MAP_CONF_0).....3522
38.5.335 DC Mapping Configuration Register 1 (IPU_DC_MAP_CONF_1).....3523
38.5.336 DC Mapping Configuration Register 2 (IPU_DC_MAP_CONF_2).....3524
38.5.337 DC Mapping Configuration Register 3 (IPU_DC_MAP_CONF_3).....3525
38.5.338 DC Mapping Configuration Register 4 (IPU_DC_MAP_CONF_4).....3526
38.5.339 DC Mapping Configuration Register 5 (IPU_DC_MAP_CONF_5).....3527
38.5.340 DC Mapping Configuration Register 6 (IPU_DC_MAP_CONF_6).....3528
38.5.341 DC Mapping Configuration Register 7 (IPU_DC_MAP_CONF_7).....3529
38.5.342 DC Mapping Configuration Register 8 (IPU_DC_MAP_CONF_8).....3530
38.5.343 DC Mapping Configuration Register 9 (IPU_DC_MAP_CONF_9).....3531
38.5.344 DC Mapping Configuration Register 10 (IPU_DC_MAP_CONF_10).....3532
38.5.345 DC Mapping Configuration Register 11 (IPU_DC_MAP_CONF_11).....3533
38.5.346 DC Mapping Configuration Register 12 (IPU_DC_MAP_CONF_12).....3534
38.5.347 DC Mapping Configuration Register 13 (IPU_DC_MAP_CONF_13).....3535
38.5.348 DC Mapping Configuration Register 14 (IPU_DC_MAP_CONF_14).....3536
38.5.349 DC Mapping Configuration Register 15 (IPU_DC_MAP_CONF_15).....3537
38.5.350 DC Mapping Configuration Register 16 (IPU_DC_MAP_CONF_16).....3537
38.5.351 DC Mapping Configuration Register 17 (IPU_DC_MAP_CONF_17).....3538
38.5.352 DC Mapping Configuration Register 18 (IPU_DC_MAP_CONF_18).....3539
38.5.353 DC Mapping Configuration Register 19 (IPU_DC_MAP_CONF_19).....3539
38.5.354 DC Mapping Configuration Register 20 (IPU_DC_MAP_CONF_20).....3540
38.5.355 DC Mapping Configuration Register 21 (IPU_DC_MAP_CONF_21).....3541
38.5.356 DC Mapping Configuration Register 22 (IPU_DC_MAP_CONF_22).....3541
38.5.357 DC Mapping Configuration Register 23 (IPU_DC_MAP_CONF_23).....3542
38.5.358 DC Mapping Configuration Register 24 (IPU_DC_MAP_CONF_24).....3543
38.5.359 DC Mapping Configuration Register 25 (IPU_DC_MAP_CONF_25).....3543
38.5.360 DC Mapping Configuration Register 26 (IPU_DC_MAP_CONF_26).....3544

38.5.361 DC User General Data Event 0 Register 0 (IPU_DC_UGDE0_0)....3545
38.5.362 DC User General Data Event 0 Register 1 (IPU_DC_UGDE0_1)....3546
38.5.363 DC User General Data Event 0 Register2 (IPU_DC_UGDE0_2)....3547
38.5.364 DC User General Data Event 0 Register 3 (IPU_DC_UGDE0_3)....3547
38.5.365 DC User General Data Event 1Register0 (IPU_DC_UGDE1_0)....3548
38.5.366 DC User General Data Event 1 Register 1 (IPU_DC_UGDE1_1)....3549
38.5.367 DC User General Data Event 1Register 2 (IPU_DC_UGDE1_2)....3550
38.5.368 DC User General Data Event 1Register 3 (IPU_DC_UGDE1_3)....3550
38.5.369 DC User General Data Event 2 Register 0 (IPU_DC_UGDE2_0)....3551
38.5.370 DC User General Data Event 2 Register 1 (IPU_DC_UGDE2_1)....3552
38.5.371 DC User General Data Event 2Register 2 (IPU_DC_UGDE2_2)....3553
38.5.372 DC User General Data Event 2Register 3 (IPU_DC_UGDE2_3)....3553
38.5.373 DC User General Data Event 3Register 0 (IPU_DC_UGDE3_0)....3554
38.5.374 DC User General Data Event 3Register 1 (IPU_DC_UGDE3_1)....3555
38.5.375 DC User General Data Event 3Register 2 (IPU_DC_UGDE3_2)....3556
38.5.376 DC User General Data Event 3Register 2 (IPU_DC_UGDE3_3)....3556
38.5.377 DC Low Level Access Control Register 0 (IPU_DC_LLA0).... 3556
38.5.378 DC Low Level Access Control Register 1 (IPU_DC_LLA1).... 3557
38.5.379 DC Read Low Level Read Access Control Register 0 (IPU_DC_R_LLA0).... 3557
38.5.380 DC Read Low Level Read Access Control Register1 (IPU_DC_R_LLA1).... 3558
38.5.381 DC Write Channel 5 Configuration Register (IPU_DC_WR_CH_ADDR_5_ALT).... 3558
38.5.382 DC Status Register (IPU_DC_STAT).... 3560
38.5.383 DMFC Read Channel Register (IPU_DMFC_RD_CHAN).... 3562
38.5.384 DMFC Write Channel Register (IPU_DMFC_WR_CHAN).... 3564
38.5.385 DMFC Write Channel Definition Register (IPU_DMFC_WR_CHAN_DEF).... 3567
38.5.386 DMFC Display Processor Channel Register (IPU_DMFC_DP_CHAN)....3569
38.5.387 DMFC Display Processor Channel Definition Register (IPU_DMFC_DP_CHAN_DEF)....3572
38.5.388 DMFC General 1 Register (IPU_DMFC_GENERAL_1).... 3574
38.5.389 DMFC General 2 Register (IPU_DMFC_GENERAL_2).... 3576

38.5.390 DMFC IC Interface Control Register (IPU_DMFC_IC_CTRL)....3577
38.5.391 DMFC Write Channel Alternate Register (IPU_DMFC_WR_CHAN_ALT)....3578
38.5.392 DMFC Write Channel Definition Alternate Register (IPU_DMFC_WR_CHAN_DEF_ALT)....3579
38.5.393 DMFC MFC Display Processor Channel Alternate Register (IPU_DMFC_DP_CHAN_ALT)....3580
38.5.394 DMFC Display Channel Definition Alternate Register (IPU_DMFC_DP_CHAN_DEF_ALT)....3583
38.5.395 DMFC General 1 Alternate Register (IPU_DMFC_GENERAL1_ALT)....3585
38.5.396 DMFC Status Register (IPU_DMFC_STAT)....3587
38.5.397 VDI Field Size Register (IPU_VDI_FSIZE)....3588
38.5.398 VDI Control Register (IPU_VDI_C)....3589
38.5.399 VDI Control Register 2 (IPU_VDI_C2_)....3591
38.5.400 VDI Combining Parameters Register 1 (IPU_VDI_CMDP_1)....3592
38.5.401 VDI Combining Parameters Register 2 (IPU_VDI_CMDP_2)....3593
38.5.402 VDI Plane Size Register 1 (IPU_VDI_PS_1)....3593
38.5.403 VDI Plane Size Register 2 (IPU_VDI_PS_2)....3594
38.5.404 VDI Plane Size Register 3 (IPU_VDI_PS_3)....3595
38.5.405 VDI Plane Size Register 4 (IPU_VDI_PS_4)....3595

Chapter 39
Keypad Port (KPP)

39.1 Overview....3597
39.1.1 Features....3599
39.1.2 Modes and Operations....3599
39.2 Clocks....3599
39.3 External Signals....3599
39.3.1 Input Pins....3601
39.3.2 Output Pins....3601
39.3.3 Generation of Transfer Error Signal on Peripheral Bus....3601
39.4 Functional Description....3602
39.4.1 Keypad Matrix Construction....3602
39.4.2 Keypad Port Configuration....3602

39.4.3 Keypad Matrix Scanning ... 3602
39.4.4 Keypad Standby ... 3603
39.4.5 Glitch Suppression on Keypad Inputs ... 3603
39.4.6 Multiple Key Closures ... 3604
39.4.6.1 Ghost Key Problem and Correction ... 3606
39.4.7 3-Point Contact Keys Support ... 3608
39.5 Initialization/Application Information ... 3609
39.5.1 Typical Keypad Configuration and Scanning Sequence ... 3609
39.5.2 Key Press Interrupt Scanning Sequence ... 3610
39.5.3 Additional Comments ... 3610
39.6 KPP Memory Map/Register Definition ... 3610
39.6.1 Keypad Control Register (KPP_KPCR) ... 3611
39.6.2 Keypad Status Register (KPP_KPSR) ... 3612
39.6.3 Keypad Data Direction Register (KPP_KDDR) ... 3613
39.6.4 Keypad Data Register (KPP_KPDR) ... 3614

Chapter 40
LVDS Display Bridge (LDB)

40.1 Overview ... 3617
40.1.1 Relevant Standards ... 3619
40.2 External Signals ... 3619
40.3 Clocks ... 3619
40.4 Input and Output Ports ... 3620
40.4.1 Input Parallel Display Ports ... 3620
40.4.2 Output LVDS Ports ... 3621
40.5 Processing ... 3621
40.5.1 Mapping of Input Data Busses ... 3621
40.5.2 Bit Mapping ... 3622
40.6 LDB Memory Map/Register Definition ... 3622
40.6.1 LDB Control Register (LDB_CTRL) ... 3623

Chapter 41
MIPI - Camera Serial Interface Host Controller (MIPI_CSI)

41.1 Overview....3627

41.2 Features....3627

41.3 Architecture....3628

41.3.1 Startup Sequence....3629

41.3.2 Interrupt mechanism....3631

41.3.3 Signals....3632

41.4 Timing Interfaces....3633

41.4.1 Image Data Interface....3633

41.5 Payload Data Output Format....3638

41.5.1 General/Arbitrary Data Reception....3638

41.5.2 RGB888 Data Reception....3639

41.5.3 RGB666 Data Reception....3640

41.5.4 RGB565 Data Reception....3641

41.5.5 RGB555 Data Reception....3642

41.5.6 RGB444 Data Reception....3643

41.5.7 YUV422 8-bit Data Reception....3643

41.5.8 YUV422 10-bit Data Reception....3644

41.5.9 YUV420 8-bit (Legacy) Data Reception....3645

41.5.10 YUV420 8-bit Data Reception....3646

41.5.11 YUV420 10-bit Data Reception....3647

41.5.12 RAW6 Data Reception....3649

41.5.13 RAW7 Data Reception....3649

41.5.14 RAW8 Data Reception....3650

41.5.15 RAW10 Data Reception....3650

41.5.16 RAW12 Data Reception....3651

41.5.17 RAW14 Data Reception....3652

41.6 MIPI_CSI Memory Map/Register Definition....3653

41.6.1 Controller Version Identification Register (MIPI_CSI_VERSION)....3654
41.6.2 Number of Active Data Lanes (MIPI_CSI_N_LANES)....3655
41.6.3 Phy shutdown control (MIPI_CSI_PHY_SHUTDOWNZ)....3656
41.6.4 Phy reset control (MIPI_CSI_DPHY_RSTZ)....3657
41.6.5 CSI2 controller reset (MIPI_CSI_CSI2_RESETN)....3657
41.6.6 General settings for all blocks (MIPI_CSI_PHY_STATE)....3659
41.6.7 Data IDs for which IDI reports line boundary matching errors (MIPI_CSI_DATA_IDS_1)....3660
41.6.8 Error state register 1 (MIPI_CSI_ERR1)....3662
41.6.9 Error state register 2 (MIPI_CSI_ERR2)....3666
41.6.10 Masks for errors 2 (MIPI_CSI_MASK2)....3668
41.6.11 D-PHY Test interface control 0 (MIPI_CSI_PHY_TST_CRTL0)....3670
41.6.12 D-PHY Test interface control 1 (MIPI_CSI_PHY_TST_CTRL1)....3671

Chapter 42
MIPI DSI Host Controller (MIPI_DSI)

42.1 Overview....3673
42.2 Features....3673
42.2.1 System Overview....3674
42.3 Clocks....3674
42.4 Architecture....3675
42.4.1 Architecture Overview....3675
42.4.2 DBI-2 interface....3676
42.4.3 DPI-2 Interface....3693
42.4.4 Error control and timeout timers....3695
42.4.5 Timeout timers....3695
42.4.6 Error control....3695
42.4.7 Generic packets....3697
42.4.8 Signals....3698
42.5 Programming....3699
42.5.1 DSI and D-PHY initialization sequence....3699

42.6 MIPI_DSI Memory Map/Register Definition....3700
42.6.1 Version of the DSI host ctrl (MIPI_DSI_VERSION).... 3702
42.6.2 Core power up (MIPI_DSI_PWR_UP)....3702
42.6.3 Number of active data lanes (MIPI_DSI_CLKMGR_CFG).... 3703
42.6.4 DPI interface configuration (MIPI_DSI_DPI_CFG)....3703
42.6.5 DBI interface configuration (MIPI_DSI_DBI_CFG).... 3705
42.6.6 DBI command size configuration (MIPI_DSI_DBIS_CMDSIZE).... 3706
42.6.7 Packet handler configuration (MIPI_DSI_PCKHDL_CFG).... 3707
42.6.8 Video Mode Configuration (MIPI_DSI_VID_MODE_CFG).... 3708
42.6.9 Video packet configuration (MIPI_DSI_VID_PKT_CFG).... 3709
42.6.10 Command mode configuration (MIPI_DSI_CMD_MODE_CFG).... 3710
42.6.11 Line timer configuration (MIPI_DSI_TMR_LINE_CFG).... 3712
42.6.12 Vertical timing configuration (MIPI_DSI_VTIMING_CFG).... 3712
42.6.13 D-PHY timing configuration (MIPI_DSI_PHY_TMR_CFG).... 3713
42.6.14 Generic packet Header configuration (MIPI_DSI_GEN_HDR).... 3713
42.6.15 Generic payload data in/out (MIPI_DSI_GEN_PLD_DATA)....3714
42.6.16 Command packet status (MIPI_DSI_CMD_PKT_STATUS).... 3714
42.6.17 Time Out timers configuration (MIPI_DSI_TO_CNT_CFG0)....3716
42.6.18 Interrupt status register 0 (MIPI_DSI_ERROR_ST0)....3716
42.6.19 Interrupt status register 1 (MIPI_DSI_ERROR_ST1)....3718
42.6.20 Masks Interrupt generation trigged by ERROR_ST0 register (MIPI_DSI_ERROR_MSK0).... 3719
42.6.21 Masks Interrupt generation trigged by ERROR_ST1 register (MIPI_DSI_ERROR_MSK1).... 3721
42.6.22 D-PHY reset control (MIPI_DSI_PHY_RSTZ).... 3723
42.6.23 D-PHY interface configuration (MIPI_DSI_PHY_IF_CFG_)....3724
42.6.24 D-PHY PPI interface control (MIPI_DSI_PHY_IF_CTRL)....3724
42.6.25 D-PHY PPI status interface (MIPI_DSI_PHY_STATUS)....3725
42.6.26 D-PHY Test interface control 0 (MIPI_DSI_PHY_TST_CTRL0).... 3727
42.6.27 D-PHY Test interface control 1 (MIPI_DSI_PHY_TST_CTRL1).... 3727

Chapter 43

MIPI HSI Host Controller (MIPI_HSI)

43.1 Overview....3729

43.1.1 Features....3729

43.2 External Signals....3730

43.2.1 External Signals Overview....3730

43.2.2 Ports Table....3731

43.3 Clocks....3731

43.4 Functional Description....3731

43.4.1 DMA AHB /PIO AHB slave and Data Channel Buffer....3732

43.4.1.1 AHB Slave Interface....3733

43.4.1.2 AHB Master Interface....3734

43.4.1.3 Control Registers....3734

43.4.1.4 DMA Engine....3734

43.4.1.5 Rx/Tx FIFO controller....3734

43.4.1.6 Arbiter....3734

43.4.1.7 4K Bytes Dual Port RAM....3734

43.4.1.8 Typical Software Operation Flow for DMA....3734

43.4.1.9 Typical Software Operation Flow 1 for Data Port....3735

43.4.1.10 Typical Software Operation Flow 2 for Data Port....3735

43.4.2 HSI Reset....3735

43.5 HSI Memory Map/Register Definition....3735

43.5.1 HSI Control Register (MIPI_HSI_CTRL)....3742

43.5.2 HSI Tx Config Register (MIPI_HSI_TX_CONF)....3744

43.5.3 HSI Rx Config Register (MIPI_HSI_RX_CONF)....3746

43.5.4 HSI Capability Register (MIPI_HSI_CAP)....3749

43.5.5 HSI Tx Water Mark Level 0 Register (MIPI_HSI_TX_WML0)....3750

43.5.6 HSI Tx Water Mark Level 1 Register (MIPI_HSI_TX_TML1)....3752

43.5.7 HSI Tx Arbiter Priority 0 Register (MIPI_HSI_TX_ARB_PRI0)....3754

43.5.8 HSI Tx Arbiter Priority 1 Register (MIPI_HSI_TX_ARB_PRI1)....3756

43.5.9 HSI Line Status Register (MIPI_HSI_LINE_ST)..... 3759
43.5.10 HSI ID Bits Register (MIPI_HSI_ID_BIT)..... 3761
43.5.11 Tx and Rx Fif0 Threshold Configuration Register (MIPI_HSI_FIFO_THR_CONF)..... 3762
43.5.12 Tx and Rx Channel Soft Reset Register (MIPI_HSI_CH_SFTRST)..... 3765
43.5.13 HSI Interrupt Status Register (MIPI_HSI_IRQSTAT)..... 3767
43.5.14 HSI Interrupt Status Enable Register (MIPI_HSI_IRQSTAT_EN)..... 3769
43.5.15 HSI Interrupt Signal Enable Register (MIPI_HSI_IRQSIG_EN)..... 3771
43.5.16 HSI FIFO Threshold Interrupt Status Register (MIPI_HSI_FIFO_THR_IRQSTAT)..... 3773
43.5.17 HSI FIFO Threshold Interrupt Status Enable Register (MIPI_HSI_FIFO_THR_IRQSTAT_EN)..... 3776
43.5.18 HSI FIFO Threshold Interrupt Signal Enable Register (MIPI_HSI_FIFO_THR_IRQSIG_EN)..... 3779
43.5.19 Tx Channel n Data Port Register (MIPI_HSI_TX_CH*n*_DP)..... 3782
43.5.20 Rx Channel n Data Port Register (MIPI_HSI_RX_CH*n*_DP)..... 3783
43.5.21 HSI Error Interrupt Status Register (MIPI_HSI_ERR_IRQSTAT)..... 3784
43.5.22 HSI Error Interrupt Status Enable Register (MIPI_HSI_ERR_IRQSTAT_EN)..... 3787
43.5.23 HSI Error Interrupt Signal Enable Register (MIPI_HSI_ERR_IRQSIG_EN)..... 3789
43.5.24 Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA*n*_CONF)..... 3791
43.5.25 Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA*n*_CONF)..... 3792
43.5.26 Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA*n*_STA_ADDR)..... 3793
43.5.27 Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA*n*_STA_ADDR)..... 3793
43.5.28 DMA Interrupt Status Register (MIPI_HSI_DMA_IRQSTAT)..... 3794
43.5.29 DMA Interrupt Enable Register (MIPI_HSI_DMA_IRQSTAT_EN)..... 3796
43.5.30 DMA Interrupt Status Signal Enable Register (MIPI_HSI_DMA_IRQSIG_EN)..... 3798
43.5.31 DMA Error Interrupt Status Register (MIPI_HSI_DMA_ERR_IRQSTAT)..... 3801
43.5.32 DMA Error Interrupt Enable Register (MIPI_HSI_DMA_ERR_IRQSTAT_EN)..... 3803
43.5.33 DMA Error Interrupt Signal Enable Register (MIPI_HSI_DMA_ERR_IRQSIG_EN)..... 3805
43.5.34 DMA Single Request Enable Register (MIPI_HSI_DMA_SINGLE_REQ_EN)..... 3808
43.5.35 Tx Fifo Size Configuration Register 0 (MIPI_HSI_TX_FIFO_SIZE_CONF0)..... 3810
43.5.36 Tx Fifo Size Configuration Register 1 (MIPI_HSI_TX_FIFO_SIZE_CONF1)..... 3813
43.5.37 Rx Fifo Size Configuration Register 0 (MIPI_HSI_RX_FIFO_SIZE_CONF0)..... 3816

43.5.38 Rx Fifo Size Configuration Register 1 (MIPI_HSI_RX_FIFO_SIZE_CONF1)....3819
43.5.39 Tx Fifo Status Register (MIPI_HSI_TX_FIFO_STAT)....3822
43.5.40 Rx Fifo Status Register (MIPI_HSI_RX_FIFO_STAT)....3824
43.5.41 AHB Master Config Register (MIPI_HSI_AHB_MASTER_CONF)....3826
43.5.42 TX Break Length Register (MIPI_HSI_TX_BREAK_LEN)....3827

Chapter 44
MediaLB (MLB)

44.1 Overview....3829
44.1.1 Block Diagram....3829
44.1.1.1 Bus Interfaces....3830
44.2 External Signals....3831
44.3 Clocks....3831
44.4 Functional Description....3832
44.4.1 MediaLB Block....3832
44.4.1.1 MediaLB Channel Address to Logical Channel Mapping....3833
44.4.2 Host Bus Interface Block....3833
44.4.2.1 HBI Physical Addresses....3833
44.4.3 Routing Fabric Block....3834
44.4.3.1 Data Buffer RAM....3834
44.4.3.1.1 Synchronous Channels....3834
44.4.3.1.2 Isochronous Channels....3835
44.4.3.1.3 Asynchronous and Control Channels....3835
44.4.3.2 Channel Table RAM....3835
44.4.3.2.1 Address Mapping....3835
44.4.3.2.2 Channel Allocation Table....3836
44.4.3.2.2.1 Channel Setup....3837
44.4.3.2.3 Channel Descriptor Table....3838
44.4.3.2.3.1 Synchronous Channel Operation....3838
44.4.3.2.3.2 Synchronous Channel Descriptors....3839

44.4.3.2.3.3 Isochronous Channel Descriptors....3841
44.4.3.2.3.4 Asynchronous and Control Channel Descriptors....3842
44.4.4 Memory Interface Block....3844
44.4.4.1 CTR Access....3844
44.4.4.1.1 Direct CTR Writes....3845
44.4.4.1.2 Direct CTR Reads....3845
44.4.4.1.3 CTR Addressing....3846
44.4.4.2 DBR Access....3846
44.4.4.2.1 Direct DBR Writes....3847
44.4.4.2.2 Direct DBR Reads....3847
44.4.5 Interrupt Interface Block....3847
44.4.6 AMBA AHB Block....3848
44.4.6.1 AHB Descriptor Table....3848
44.4.6.2 AHB Synchronous Channel Descriptors....3850
44.4.6.3 AHB Isochronous Channel Descriptors....3850
44.4.6.4 AHB Asynchronous and Control Channel Descriptors....3851
44.4.6.4.1 Single-packet Mode....3851
44.4.6.4.2 Multiple-packet Mode....3853
44.5 Software Flow....3854
44.5.1 Channel Initialization....3855
44.5.1.1 Configure the Hardware....3855
44.5.1.2 Program the Routing Fabric Block....3855
44.5.1.3 Program the AMBA AHB Block DMAs....3856
44.5.1.4 Synchronize and Unmute Synchronous Channel....3858
44.5.2 Channel Servicing....3858
44.5.2.1 Servicing the AMBA AHB Block (DMA) Interrupts....3858
44.5.2.2 Servicing the MediaLB Interrupts....3859
44.5.2.3 Polling for MediaLB System Commands....3859
44.5.3 Low Power Mode....3860

44.6 MLB150 Memory Map/Register Definition....3860
44.6.1 MediaLB Control 0 Register (MLB_MLBC0)....3863
44.6.2 MediaLB 6-pin Control 0 Register (MLB_MLBPC0)....3866
44.6.3 MediaLB Channel Status 0 Register (MLB_MS0)....3866
44.6.4 MediaLB 6-pin Control 2 Register (MLB_MLBPC2)....3867
44.6.5 MediaLB Channel Status1 Register (MLB_MS1)....3868
44.6.6 MediaLB System Status Register (MLB_MSS)....3869
44.6.7 MediaLB System Data Register (MLB_MSD)....3870
44.6.8 MediaLB Interrupt Enable Register (MLB_MIEN)....3871
44.6.9 MediaLB 6-pin Control 1 Register (MLB_MLBPC1)....3872
44.6.10 MediaLB Control 1 Register (MLB_MLBC1)....3873
44.6.11 HBI Control Register (MLB_HCTL)....3874
44.6.12 HBI Channel Mask 0 Register (MLB_HCMR0)....3875
44.6.13 HBI Channel Mask 1 Register (MLB_HCMR1)....3875
44.6.14 HBI Channel Error 0 Register (MLB_HCER0)....3876
44.6.15 HBI Channel Error 1 Register (MLB_HCER1)....3876
44.6.16 HBI Channel Busy 0 Register (MLB_HCBR0)....3877
44.6.17 HBI Channel Busy 1 Register (MLB_HCBR1)....3877
44.6.18 MIF Data 0 Register (MLB_MDAT0)....3878
44.6.19 MIF Data 1 Register (MLB_MDAT1)....3878
44.6.20 MIF Data 2 Register (MLB_MDAT2)....3878
44.6.21 MIF Data 3 Register (MLB_MDAT3)....3879
44.6.22 MIF Data Write Enable 0 Register (MLB_MDWE0)....3879
44.6.23 MIF Data Write Enable 1 Register (MLB_MDWE1)....3879
44.6.24 MIF Data Write Enable 2 Register (MLB_MDWE2)....3880
44.6.25 MIF Data Write Enable 3 Register (MLB_MDWE3)....3880
44.6.26 MIF Control Register (MLB_MCTL)....3881
44.6.27 MIF Address Register (MLB_MADR)....3881
44.6.28 AHB Control Register (MLB_ACTL)....3882

44.6.29 AHB Channel Status 0 Register (MLB_ACSR0)....3883
44.6.30 AHB Channel Status 1 Register (MLB_ACSR1)....3884
44.6.31 AHB Channel Mask 0 Register (MLB_ACMR0)....3884
44.6.32 AHB Channel Mask 1 Register (MLB_ACMR1)....3885

Chapter 45
Multi Mode DDR Controller (MMDC)

45.1 Overview....3887
45.1.1 MMDC feature summary....3889
45.2 External Signals....3891
45.3 Clocks....3891
45.4 Functional Description....3892
45.4.1 Write/Read data flow....3892
45.4.1.1 Write data flow....3892
45.4.1.2 Read data flow....3893
45.4.2 MMDC initialization....3893
45.4.3 Configuring the MMDC registers....3895
45.4.4 MMDC Address Space....3895
45.4.4.1 Address decoding....3895
45.4.4.2 Chip select settings....3898
45.4.4.2.1 Creating 4 Gbyte address space with 2 Gbyte CS density....3898
45.4.4.2.2 Creating 2 Gbyte address spaces with 1 Gbyte CS density....3899
45.4.4.3 Translation of AXI accesses to DDR accessess....3900
45.4.4.3.1 Example 1....3900
45.4.4.3.2 Example 2....3900
45.4.4.3.3 Example 3....3901
45.4.4.3.4 Example 4....3902
45.4.4.3.5 Example 5....3902
45.4.4.4 Address mirroring....3903
45.4.5 LPDDR2 and DDR3 pin mux mapping....3904

45.4.6 Power Saving and Clock Frequency Change modes ... 3905
45.4.6.1 Power saving general ... 3905
45.4.6.2 Self refresh and Frequency change entry/exit ... 3906
45.4.7 Reset ... 3908
45.4.7.1 Hard reset ... 3908
45.4.7.2 Warm reset ... 3908
45.4.7.3 Software reset ... 3909
45.4.8 Refresh Scheme ... 3909
45.4.9 Burst Length options towards DDR ... 3910
45.4.10 Exclusive accesses handling ... 3910
45.4.11 AXI Error Handling ... 3912
45.5 Performance ... 3912
45.5.1 Arbitration and reordering mechanism ... 3912
45.5.1.1 Arbitration General ... 3912
45.5.1.2 Real time channel mode ... 3913
45.5.1.3 Dynamic scoring mode (Arbitration Winning Conditions) ... 3913
45.5.1.4 Guarding (aging) mechanism ... 3914
45.5.2 Prediction mechanism ... 3914
45.5.3 Special Optimization for accesses towards DDR3 ... 3915
45.6 MMDC Debug ... 3916
45.6.1 Hardware debug monitor ... 3916
45.6.2 Step By Step (SBS) software monitor ... 3916
45.7 MMDC Profiling ... 3917
45.8 LPDDR2 Refresh Rate Update and Timing Derating ... 3919
45.9 DLL Switching ... 3920
45.9.1 DLL Off mode ... 3920
45.10 ODT Configuration ... 3921
45.11 Calibration Process ... 3923
45.11.1 Delay-line ... 3924

45.11.2 ZQ calibration 3924
45.11.2.1 ZQ automatic (hardware) calibration process..... 3925
45.11.2.1.1 ZQ automatic Pull-up calibration..... 3926
45.11.2.1.2 ZQ automatic Pull-down calibration..... 3926
45.11.2.2 ZQ software calibration process..... 3926
45.11.2.3 ZQ calibration commands 3927
45.11.3 Read DQS Gating Calibration..... 3927
45.11.3.1 Hardware DQS Gating Calibration..... 3927
45.11.3.1.1 Hardware DQS Calibration with MPR..... 3928
45.11.3.1.2 Hardware DQS Calibration with pre-defined value..... 3928
45.11.3.2 SW read DQS gating Calibration..... 3931
45.11.3.2.1 SW read Calibration with MPR..... 3931
45.11.3.2.2 SW read Calibration with pre-defined value..... 3931
45.11.4 Read Calibration..... 3933
45.11.4.1 Hardware (automatic) Read Calibration..... 3934
45.11.4.1.1 Hardware (automatic) Calibration with MPR/DQ Calibration..... 3934
45.11.4.1.2 Hardware (automatic) Calibration with pre-defined value..... 3934
45.11.4.2 SW Read Calibration..... 3936
45.11.4.2.1 Calibration with MPR/DQ calibration..... 3936
45.11.4.2.2 Calibration with pre-defined value..... 3936
45.11.5 Write Calibration..... 3938
45.11.5.1 HW (automatic) Write Calibration..... 3938
45.11.5.2 SW Write Calibration..... 3940
45.11.6 Write leveling Calibration..... 3941
45.11.6.1 Hardware Write Leveling Calibration..... 3942
45.11.6.2 SW Write Leveling Calibration..... 3944
45.11.7 Write fine tuning..... 3945
45.11.8 Read fine tuning..... 3945
45.11.9 ZQ Fine Tuning..... 3945

45.11.10 Duty cycle adjustment....3946
45.12 MMDC Memory Map/Register Definition....3946
45.12.1 MMDC Core Control Register (MMDC*x*_MDCTL)....3957
45.12.2 MMDC Core Power Down Control Register (MMDC*x*_MDPDC)....3959
45.12.3 MMDC Core ODT Timing Control Register (MMDC*x*_MDOTC)....3961
45.12.4 MMDC Core Timing Configuration Register 0 (MMDC*x*_MDCFG0)....3963
45.12.5 MMDC Core Timing Configuration Register 1 (MMDC*x*_MDCFG1)....3965
45.12.6 MMDC Core Timing Configuration Register 2 (MMDC*x*_MDCFG2)....3967
45.12.7 MMDC Core Miscellaneous Register (MMDC*x*_MDMISC)....3969
45.12.8 MMDC Core Special Command Register (MMDC*x*_MDSCR)....3972
45.12.9 MMDC Core Refresh Control Register (MMDC*x*_MDREF)....3975
45.12.10 MMDC Core Read/Write Command Delay Register (MMDC*x*_MDRWD)....3978
45.12.11 MMDC Core Out of Reset Delays Register (MMDC*x*_MDOR)....3980
45.12.12 MMDC Core MRR Data Register (MMDC*x*_MDMRR)....3981
45.12.13 MMDC Core Timing Configuration Register 3 (MMDC*x*_MDCFG3LP)....3982
45.12.14 MMDC Core MR4 Derating Register (MMDC*x*_MDMR4)....3983
45.12.15 MMDC Core Address Space Partition Register (MMDC*x*_MDASP)....3985
45.12.16 MMDC Core AXI Reordering Control Register (MMDC*x*_MAARCR)....3986
45.12.17 MMDC Core Power Saving Control and Status Register (MMDC*x*_MAPSR)....3988
45.12.18 MMDC Core Exclusive ID Monitor Register0 (MMDC*x*_MAEXIDR0)....3991
45.12.19 MMDC Core Exclusive ID Monitor Register1 (MMDC*x*_MAEXIDR1)....3991
45.12.20 MMDC Core Debug and Profiling Control Register 0 (MMDC*x*_MADPCR0)....3992
45.12.21 MMDC Core Debug and Profiling Control Register 1 (MMDC*x*_MADPCR1)....3994
45.12.22 MMDC Core Debug and Profiling Status Register 0 (MMDC*x*_MADPSR0)....3994
45.12.23 MMDC Core Debug and Profiling Status Register 1 (MMDC*x*_MADPSR1)....3995
45.12.24 MMDC Core Debug and Profiling Status Register 2 (MMDC*x*_MADPSR2)....3995
45.12.25 MMDC Core Debug and Profiling Status Register 3 (MMDC*x*_MADPSR3)....3996
45.12.26 MMDC Core Debug and Profiling Status Register 4 (MMDC*x*_MADPSR4)....3996
45.12.27 MMDC Core Debug and Profiling Status Register 5 (MMDC*x*_MADPSR5)....3997

45.12.28 MMDC Core Step By Step Address Register (MMDC*x*_MASBS0)..... 3997
45.12.29 MMDC Core Step By Step Address Attributes Register (MMDC*x*_MASBS1)..... 3998
45.12.30 MMDC Core General Purpose Register (MMDC*x*_MAGENP)..... 3999
45.12.31 MMDC PHY ZQ HW control register (MMDC*x*_MPZQHWCTRL).....4000
45.12.32 MMDC PHY ZQ SW control register (MMDC*x*_MPZQSWCTRL).....4002
45.12.33 MMDC PHY Write Leveling Configuration and Error Status Register (MMDC*x*_MPWLGCR).....4004
45.12.34 MMDC PHY Write Leveling Delay Control Register 0 (MMDC*x*_MPWLDECTRL0).....4007
45.12.35 MMDC PHY Write Leveling Delay Control Register 1 (MMDC*x*_MPWLDECTRL1).....4009
45.12.36 MMDC PHY Write Leveling delay-line Status Register (MMDC*x*_MPWLDLST)..... 4012
45.12.37 MMDC PHY ODT control register (MMDC*x*_MPODTCTRL)..... 4014
45.12.38 MMDC PHY Read DQ Byte0 Delay Register (MMDC*x*_MPRDDQBY0DL)..... 4016
45.12.39 MMDC PHY Read DQ Byte1 Delay Register (MMDC*x*_MPRDDQBY1DL)..... 4019
45.12.40 MMDC PHY Read DQ Byte2 Delay Register (MMDC*x*_MPRDDQBY2DL)..... 4021
45.12.41 MMDC PHY Read DQ Byte3 Delay Register (MMDC*x*_MPRDDQBY3DL)..... 4024
45.12.42 MMDC PHY Write DQ Byte0 Delay Register (MMDC*x*_MPWRDQBY0DL).....4027
45.12.43 MMDC PHY Write DQ Byte1 Delay Register (MMDC*x*_MPWRDQBY1DL).....4029
45.12.44 MMDC PHY Write DQ Byte2 Delay Register (MMDC*x*_MPWRDQBY2DL).....4031
45.12.45 MMDC PHY Write DQ Byte3 Delay Register (MMDC*x*_MPWRDQBY3DL).....4033
45.12.46 MMDC PHY Read DQS Gating Control Register 0 (MMDC*x*_MPDGCTRL0)..... 4036
45.12.47 MMDC PHY Read DQS Gating Control Register 1 (MMDC*x*_MPDGCTRL1)..... 4038
45.12.48 MMDC PHY Read DQS Gating delay-line Status Register (MMDC*x*_MPDGDLST0)..... 4040
45.12.49 MMDC PHY Read delay-lines Configuration Register (MMDC*x*_MPRDDLCTL)..... 4042
45.12.50 MMDC PHY Read delay-lines Status Register (MMDC*x*_MPRDDLST).....4043
45.12.51 MMDC PHY Write delay-lines Configuration Register (MMDC*x*_MPWRDLCTL)..... 4044
45.12.52 MMDC PHY Write delay-lines Status Register (MMDC*x*_MPWRDLST).....4046
45.12.53 MMDC PHY CK Control Register (MMDC*x*_MPSDCTRL).....4047
45.12.54 MMDC ZQ LPDDR2 HW Control Register (MMDC*x*_MPZQLP2CTL).....4048
45.12.55 MMDC PHY Read Delay HW Calibration Control Register (MMDC*x*_MPRDDLHWCTL)..... 4049
45.12.56 MMDC PHY Write Delay HW Calibration Control Register (MMDC*x*_MPWRDLHWCTL)..... 4051

45.12.57 MMDC PHY Read Delay HW Calibration Status Register 0 (MMDC*x*_MPRDDLHWST0)..... 4053
45.12.58 MMDC PHY Read Delay HW Calibration Status Register 1 (MMDC*x*_MPRDDLHWST1)..... 4054
45.12.59 MMDC PHY Write Delay HW Calibration Status Register 0 (MMDC*x*_MPWRDLHWST0).....4055
45.12.60 MMDC PHY Write Delay HW Calibration Status Register 1 (MMDC*x*_MPWRDLHWST1).....4056
45.12.61 MMDC PHY Write Leveling HW Error Register (MMDC*x*_MPWLHWERR).....4057
45.12.62 MMDC PHY Read DQS Gating HW Status Register 0 (MMDC*x*_MPDGHWST0)..... 4057
45.12.63 MMDC PHY Read DQS Gating HW Status Register 1 (MMDC*x*_MPDGHWST1)..... 4058
45.12.64 MMDC PHY Read DQS Gating HW Status Register 2 (MMDC*x*_MPDGHWST2)..... 4059
45.12.65 MMDC PHY Read DQS Gating HW Status Register 3 (MMDC*x*_MPDGHWST3)..... 4059
45.12.66 MMDC PHY Pre-defined Compare Register 1 (MMDC*x*_MPPDCMPR1)..... 4060
45.12.67 MMDC PHY Pre-defined Compare and CA delay-line Configuration Register (MMDC*x*_MPPDCMPR2).....4061
45.12.68 MMDC PHY SW Dummy Access Register (MMDC*x*_MPSWDAR0).....4062
45.12.69 MMDC PHY SW Dummy Read Data Register 0 (MMDC*x*_MPSWDRDR0).....4064
45.12.70 MMDC PHY SW Dummy Read Data Register 1 (MMDC*x*_MPSWDRDR1).....4065
45.12.71 MMDC PHY SW Dummy Read Data Register 2 (MMDC*x*_MPSWDRDR2).....4065
45.12.72 MMDC PHY SW Dummy Read Data Register 3 (MMDC*x*_MPSWDRDR3).....4066
45.12.73 MMDC PHY SW Dummy Read Data Register 4 (MMDC*x*_MPSWDRDR4).....4066
45.12.74 MMDC PHY SW Dummy Read Data Register 5 (MMDC*x*_MPSWDRDR5).....4067
45.12.75 MMDC PHY SW Dummy Read Data Register 6 (MMDC*x*_MPSWDRDR6).....4067
45.12.76 MMDC PHY SW Dummy Read Data Register 7 (MMDC*x*_MPSWDRDR7).....4068
45.12.77 MMDC PHY Measure Unit Register (MMDC*x*_MPMUR0).....4068
45.12.78 MMDC Write CA delay-line controller (MMDC*x*_MPWRCADL).....4069
45.12.79 MMDC Duty Cycle Control Register (MMDC*x*_MPDCCR)..... 4071

Chapter 46
Network Interconnect Bus System (NIC-301)

46.1 Overview 4075
46.1.1 Block diagram..... 4075
46.1.2 NIC-301 Main Features..... 4076

46.1.3 Modes and Operations 4077
46.2 External Signals 4077
46.3 Memory Map and Register Definition 4077
46.3.1 Memory Map 4077
46.3.2 Configuration programmers model 4078
46.3.2.1 Address control and ID registers 4078
46.3.2.2 AMBA master interface block (AMIB) configuration registers 4079
46.3.2.3 ASIB (AMBA slave interface block) configuration registers 4079
46.3.3 Register Descriptions 4079
46.3.3.1 QoS registers' address look-up example 4080
46.3.4 NIC-specific parameters 4080

Chapter 47
On-Chip OTP Controller (OCOTP_CTRL)

47.1 Overview 4083
47.1.1 Features 4083
47.2 Clocks 4083
47.3 Top-Level Symbol and Functional Overview 4084
47.3.1 Operation 4084
47.3.1.1 Shadow Register Reload 4085
47.3.1.2 Fuse and Shadow Register Read 4085
47.3.1.3 Fuse and Shadow Register Writes 4086
47.3.1.4 Write Postamble 4087
47.3.2 Fuse Shadow Memory Footprint 4088
47.3.3 OTP Read/Write Timing Parameters 4089
47.3.4 Hardware Visible Fuses 4090
47.3.5 Behavior During Reset 4091
47.4 Fuse Map 4091
47.5 OCOTP Memory Map/Register Definition 4091
47.5.1 OTP Controller Control Register (OCOTP_CTRL*n*) 4094

47.5.2 OTP Controller Timing Register (OCOTP_TIMING)..... 4096
47.5.3 OTP Controller Write Data Register (OCOTP_DATA)..... 4096
47.5.4 OTP Controller Write Data Register (OCOTP_READ_CTRL)..... 4097
47.5.5 OTP Controller Read Data Register (OCOTP_READ_FUSE_DATA)..... 4098
47.5.6 Sticky bit Register (OCOTP_SW_STICKY)..... 4098
47.5.7 Software Controllable Signals Register (OCOTP_SCS*n*)..... 4099
47.5.8 OTP Controller Version Register (OCOTP_VERSION)..... 4100
47.5.9 Value of OTP Bank0 Word0 (Lock controls) (OCOTP_LOCK)..... 4101
47.5.10 Value of OTP Bank0 Word1 (Configuration and Manufacturing Info.) (OCOTP_CFG0)..... 4104
47.5.11 Value of OTP Bank0 Word2 (Configuration and Manufacturing Info.) (OCOTP_CFG1)..... 4105
47.5.12 Value of OTP Bank0 Word3 (Configuration and Manufacturing Info.) (OCOTP_CFG2)..... 4105
47.5.13 Value of OTP Bank0 Word4 (Configuration and Manufacturing Info.) (OCOTP_CFG3)..... 4106
47.5.14 Value of OTP Bank0 Word5 (Configuration and Manufacturing Info.) (OCOTP_CFG4)..... 4106
47.5.15 Value of OTP Bank0 Word6 (Configuration and Manufacturing Info.) (OCOTP_CFG5)..... 4107
47.5.16 Value of OTP Bank0 Word7 (Configuration and Manufacturing Info.) (OCOTP_CFG6)..... 4107
47.5.17 Value of OTP Bank1 Word0 (Memory Related Info.) (OCOTP_MEM0)..... 4108
47.5.18 Value of OTP Bank1 Word1 (Memory Related Info.) (OCOTP_MEM1)..... 4108
47.5.19 Value of OTP Bank1 Word2 (Memory Related Info.) (OCOTP_MEM2)..... 4109
47.5.20 Value of OTP Bank1 Word3 (Memory Related Info.) (OCOTP_MEM3)..... 4109
47.5.21 Value of OTP Bank1 Word4 (Memory Related Info.) (OCOTP_MEM4)..... 4110
47.5.22 Value of OTP Bank1 Word5 (General Purpose Customer Defined Info.) (OCOTP_ANA0)..... 4110
47.5.23 Value of OTP Bank1 Word6 (General Purpose Customer Defined Info.) (OCOTP_ANA1)..... 4111
47.5.24 Value of OTP Bank1 Word7 (General Purpose Customer Defined Info.) (OCOTP_ANA2)..... 4111
47.5.25 Shadow Register for OTP Bank3 Word0 (SRK Hash) (OCOTP_SRK0)..... 4112
47.5.26 Shadow Register for OTP Bank3 Word1 (SRK Hash) (OCOTP_SRK1)..... 4112
47.5.27 Shadow Register for OTP Bank3 Word2 (SRK Hash) (OCOTP_SRK2)..... 4113
47.5.28 Shadow Register for OTP Bank3 Word3 (SRK Hash) (OCOTP_SRK3)..... 4113
47.5.29 Shadow Register for OTP Bank3 Word4 (SRK Hash) (OCOTP_SRK4)..... 4114
47.5.30 Shadow Register for OTP Bank3 Word5 (SRK Hash) (OCOTP_SRK5)..... 4114

47.5.31 Shadow Register for OTP Bank3 Word6 (SRK Hash) (OCOTP_SRK6)..... 4115
47.5.32 Shadow Register for OTP Bank3 Word7 (SRK Hash) (OCOTP_SRK7)..... 4115
47.5.33 Value of OTP Bank4 Word0 (Secure JTAG Response Field) (OCOTP_SJC_RESP0)..... 4116
47.5.34 Value of OTP Bank4 Word1 (Secure JTAG Response Field) (OCOTP_SJC_RESP1)..... 4116
47.5.35 Value of OTP Bank4 Word2 (MAC Address) (OCOTP_MAC0)..... 4117
47.5.36 Value of OTP Bank4 Word3 (MAC Address) (OCOTP_MAC1)..... 4117
47.5.37 Value of OTP Bank4 Word6 (HW Capabilities) (OCOTP_GP1)..... 4118
47.5.38 Value of OTP Bank4 Word7 (HW Capabilities) (OCOTP_GP2)..... 4118
47.5.39 Value of OTP Bank5 Word5 (HW Capabilities) (OCOTP_MISC_CONF)..... 4119
47.5.40 Value of OTP Bank5 Word6 (HW Capabilities) (OCOTP_FIELD_RETURN)..... 4119
47.5.41 Value of OTP Bank5 Word7 (HW Capabilities) (OCOTP_SRK_REVOKE)..... 4120

Chapter 48
On-Chip RAM Memory Controller (OCRAM)

48.1 Overview..... 4121
48.2 Basic Functions..... 4122
48.2.1 Memory Map..... 4122
48.2.2 Read/Write Arbitration..... 4123
48.3 Advanced Features..... 4123
48.3.1 Read Data Wait State..... 4123
48.3.2 Read Address Pipeline..... 4124
48.3.3 Write Data Pipeline..... 4124
48.3.4 Write Address Pipeline..... 4124
48.4 Programmable Registers..... 4125

Chapter 49
PCI Express (PCIe)

49.1 Overview..... 4127
49.1.1 Terms and Abbreviations..... 4127
49.2 Architecture..... 4129
49.2.1 Common Xpress Port Logic (CXPL)..... 4131

49.2.2 Transmit Application-Dependent Module (XADM)..... 4132
49.2.2.1 Arbitration..... 4133
49.2.2.2 Credit Checking..... 4134
49.2.3 Receive Application-Dependent Module (RADM)..... 4134
49.2.3.1 Posted and Non-Posted Request and Completion TLP Processing..... 4135
49.2.3.1.1 TLP Routing..... 4136
49.2.3.2 Received Completion TLP Processing..... 4136
49.2.3.3 Message Processing..... 4137
49.2.4 Configuration-Dependent Module (CDM)..... 4137
49.2.5 Local Bus Controller (LBC) and Data bus Interface (DBI)..... 4138
49.2.5.1 Overview (LBC)..... 4138
49.2.5.1.1 Simultaneous Transactions..... 4140
49.2.5.2 ELBI..... 4140
49.2.5.3 CDM Register Space Layout..... 4140
49.2.5.3.1 PCI Configuration Header and Capability Registers (in CDM)..... 4142
49.2.5.3.2 Port Logic (PL) Registers (in CDM)..... 4142
49.2.5.3.3 Memory Mapping PL Registers..... 4142
49.2.5.4 PCIe Wire Access (EP mode)..... 4143
49.2.5.5 PCIe Wire Access (RC mode)..... 4144
49.3 Core Operations..... 4145
49.3.1 Initialization..... 4145
49.3.2 Link Establishment..... 4146
49.3.3 Transmit TLP Processing..... 4147
49.3.3.1 Transmit Overview..... 4148
49.3.3.2 Transmit TLP Arbitration..... 4148
49.3.3.2.1 Client-Based Arbitration..... 4149
49.3.3.3 Transmit Retry..... 4149
49.3.3.4 Transmit DLLP Priorities..... 4150
49.3.4 Receive TLP Processing..... 4150

49.3.4.1 Receive Overview....4150
49.3.4.2 Receive Filtering....4151
49.3.4.2.1 Filtering Rules Applicable for all TLPs Received....4152
49.3.4.2.2 Filtering Rules Based on TLP Type Defined in PCIe Specification....4153
49.3.4.2.2.1 EP Mode Filtering Rules....4154
49.3.4.2.2.2 RC Mode Filtering Rules....4155
49.3.4.2.3 Filtering Rules Not Defined in PCIe Specification....4156
49.3.4.3 Receive Routing....4157
49.3.4.3.1 EP Mode....4157
49.3.4.3.2 RC mode....4159
49.3.4.3.3 ECRC Handling....4160
49.3.4.3.4 Request TLP Routing Rules....4160
49.3.4.3.5 Completion TLP Routing Rules....4162
49.3.4.4 Receive Queuing....4163
49.3.4.4.1 Queuing Architecture....4164
49.3.4.4.1.1 SEGMENTED-BUFFER RECEIVE QUEUE CONFIGURATION (CX_RADMQ_MODE=2)....4164
49.3.4.4.2 Queue Modes....4165
49.3.4.4.3 Order Enforcement....4166
49.3.4.4.4 Queue to Port Mapping....4166
49.3.5 Error Handling....4167
49.3.5.1 Error Handling Overview....4167
49.3.5.2 PCIe Baseline Capability....4168
49.3.5.3 Advanced Error Reporting (AER)....4168
49.3.5.3.1 Advisory Non-Fatal Error Messages....4169
49.3.5.4 Error Source Classification....4170
49.3.5.5 Error Detection....4171
49.3.5.6 Application Error Reporting Interface....4173
49.3.5.7 Handling of General Errors with the AXI Bridge....4174

49.3.5.8 AXI....4174
49.3.5.9 Handling of ECRC/LCRC Errors for IO/MEM with the AXI Bridge....4174
49.3.5.9.1 Link CRC (LCRC) - a correctable Error....4174
49.3.5.9.2 End-to-end CRC (ECRC) - an Uncorrectable Non-Fatal Error....4175
49.3.6 Messages....4175
49.3.6.1 Message Generation....4176
49.3.6.1.1 Vendor Defined Message (VDM) Generation....4179
49.3.6.1.1.1 Application Msg/MsgD Programming Examples....4179
49.3.6.1.2 AHB/AXI Message Address and Size Limitations....4180
49.3.6.2 Message Reception....4181
49.3.6.2.1 Message Reception IO Interfaces....4183
49.3.6.2.2 Routing of Received Messages to SII and optionally to Application....4184
49.3.6.2.3 Accessing Header and Payload Fields of Received Messages....4185
49.3.7 Interrupts....4186
49.3.7.1 Interrupts Overview....4186
49.3.7.1.1 PCI Legacy Interrupt....4187
49.3.7.1.2 MSI....4188
49.3.7.1.3 MSI-X....4190
49.3.7.2 Interrupts (EP Mode)....4192
49.3.7.3 Interrupts (RC Mode)....4193
49.3.7.4 MSI Generation in the AXI Bridge....4194
49.3.7.5 MSI Reception in the AHB/AXI Bridge....4194
49.3.7.5.1 AHB/AXI MSI Controller (Optional in RC mode)....4194
49.3.7.5.2 Programming and Usage Model....4196
49.3.8 Flow Control....4199
49.3.9 Address Translation....4200
49.3.9.1 Internal Address Translation (iATU)....4202
49.3.9.1.1 Outbound (TX) Features....4202
49.3.9.1.2 Inbound (RX) Features....4203

49.3.9.1.3 Programming (iATU)....4203
49.3.9.2 Outbound iATU Operation....4204
49.3.9.2.1 Inbound iATU Operation....4209
49.3.9.3 Overview (iATU)....4209
49.3.10 Gen2 5.0 GT/s Operation....4216
49.3.10.1 Overview (Gen2 5.0 GT/s)....4216
49.3.10.2 Speed Changing....4216
49.3.11 Power Management....4217
49.3.11.1 L0s Power Down....4219
49.3.11.2 L1 Power Down....4220
49.3.12 Completion Timeout Ranges....4222
49.4 AXI Bridge Module....4222
49.4.1 Product Overview....4223
49.4.1.1 Overview....4223
49.4.1.2 Interfaces....4224
49.4.1.3 Features List....4225
49.4.1.4 Limitations....4226
49.4.2 Bridge Architecture....4227
49.4.2.1 Bridge Architecture Overview....4227
49.4.2.1.1 Inbound Processing Module Chain....4228
49.4.2.1.2 Outbound Processing Module Chain....4229
49.4.2.2 Decomposition....4230
49.4.2.3 Bridge Buffering....4230
49.4.2.4 Outbound Bridge Tag Management....4232
49.4.2.5 Inbound Bridge Tag Management....4234
49.4.2.6 Inbound Order Enforcement for AXI Bridge....4236
49.4.2.6.1 Ordering Enforcement Hardware Lock Feature....4237
49.4.2.6.2 Re-Ordering Effects of AXI Fabric....4238
49.4.2.6.2.1 Inbound Ordering Limitation....4239

49.4.2.6.3 Additional Information (ordering)....4239
49.4.2.6.4 PCIe Completion Reordering....4240
49.4.2.6.5 Outbound Ordering Limitation #1....4240
49.4.2.6.6 Outbound Ordering Limitation #2....4241
49.4.2.6.7 Additional Information (AXI bridge bandwidth)....4241
49.4.3 PCIe AXI Core Operations....4241
49.4.3.1 AXI Sideband (Misc. Bus) Signals....4241
49.4.3.2 Supported AXI Transfer Type....4242
49.4.3.3 Supported AXI Burst Operations....4242
49.4.3.4 I/O and CFG Transaction Handling over AXI Bridge....4242
49.4.3.4.1 Outbound I/O and CFG Transaction Handling....4243
49.4.3.4.1.1 Method I: Address Translation Method of Sending an Outbound IO or CFG Transfer....4243
49.4.3.4.2 Inbound I/O and CFG Transaction Handling....4243
49.4.4 Additional AXI Reference Material....4244
49.4.4.1 AXI Decomposition Rules....4244
49.4.4.1.1 Outbound Decomposition....4244
49.4.4.1.1.1 Decomposition Side-Effects....4246
49.4.4.1.1.2 Reducing Outbound Decomposition....4246
49.4.4.1.2 Inbound Decomposition....4246
49.4.4.1.2.1 Reducing Inbound Decomposition....4247
49.5 App Note: Order Enforcement Using the PCIe Core....4248
49.5.1 PCIe Ordering Rule Overview....4248
49.5.2 PCIe Core Inbound Order Enforcement....4249
49.5.2.1 Single queue....4249
49.5.2.2 Multiple queue....4250
49.5.2.3 Segmented buffer queue....4251
49.5.3 PCIe Core Outbound Order Enforcement....4252
49.5.4 PCIe AHB/AXI Bridge Order Enforcement....4254

49.5.5 Additional Information.... 4254
49.6 App Note: Calculating Gen1 PCI Express and AXI Bridge Throughput.... 4254
49.6.1 PCI Express Throughput.... 4255
49.6.2 Effective Throughput.... 4255
49.6.2.1 Effective Throughput Calculation.... 4255
49.6.2.1.1 Packet Level: (Start and End, Link CRC, Header).... 4255
49.6.2.1.2 Link Layer: (Flow Control and ACK/NAK DLLPs).... 4256
49.6.2.2 Other Factors Impacting Throughput.... 4257
49.7 PCIe Registers (EP mode).... 4258
49.7.1 Register Space Layout (EP mode).... 4258
49.7.2 PF Register Maps.... 4259
49.7.2.1 PF PCI Configuration Space Header - Type 0.... 4260
49.7.2.2 PF PCI Standard Capability Structures Register Maps.... 4260
49.7.2.3 PF PCI Express Extended Capability Register Maps.... 4262
49.7.3 VF Register Maps.... 4264
49.7.3.1 VF PCI Configuration Space Header - Type 0.... 4264
49.7.3.2 VF PCI Standard Capability Structures Register Maps.... 4264
49.7.4 Accessing Configuration Registers.... 4265
49.8 PCIe Registers (RC mode).... 4267
49.8.1 Register Space Layout.... 4267
49.8.2 Register Maps.... 4268
49.8.2.1 PCI Standard Capability Structures Register Maps.... 4269
49.8.2.2 PCI Express Extended Capability Register Maps.... 4271
49.8.3 Accessing Configuration Registers (Configuration).... 4272
49.9 PCIe Registers: Port Logic.... 4274
49.9.1 Overview (Port Logic).... 4274
49.9.2 Non-Standard Addressing of the iATU Port Logic Registers.... 4274
49.9.3 Accessing Configuration Registers (Port Logic Registers).... 4275
49.10 PCIe CTRL EP Mode Memory Map/Register Definition.... 4276

49.10.1 Device ID and Vendor ID Register (PCIE_EP_DeviceID)..... 4278
49.10.2 Command and Status Register (PCIE_EP_Command)..... 4278
49.10.3 BIST Register (PCIE_EP_BIST)..... 4281
49.10.4 Base Address 0 (PCIE_EP_BAR0)..... 4282
49.10.5 BAR 0 Mask Register (PCIE_EP_MASK0)..... 4284
49.10.6 BAR 1 Mask Register (PCIE_EP_MASK1)..... 4287
49.10.7 BAR 2 Mask Register (PCIE_EP_MASK2)..... 4288
49.10.8 BAR 3 Mask Register (PCIE_EP_MASK3)..... 4289
49.10.9 CardBus CIS Pointer Register (PCIE_EP_CISP)..... 4290
49.10.10 Subsystem ID and Subsystem Vendor ID Register (PCIE_EP_SSID)..... 4290
49.10.11 Expansion ROM Base Address Register (PCIE_EP_EROMBAR)..... 4291
49.10.12 Expansion ROM BAR Mask Register (PCIE_EP_EROMMASK)..... 4292
49.10.13 Capability Pointer Register (PCIE_EP_CAPPR)..... 4293
49.10.14 Interrupt Line and Pin Register (PCIE_EP_ILR)..... 4293
49.10.15 AER Capability Header (PCIE_EP_AER)..... 4294
49.10.16 Uncorrectable Error Status Register (PCIE_EP_UESR)..... 4295
49.10.17 Uncorrectable Error Mask Register (PCIE_EP_UEMR)..... 4297
49.10.18 Uncorrectable Error Severity Register (PCIE_EP_UESevR)..... 4299
49.10.19 Correctable Error Status Register (PCIE_EP_CESR)..... 4301
49.10.20 Correctable Error Mask Register (PCIE_EP_CEMR)..... 4302
49.10.21 Advanced Capabilities and Control Register (PCIE_EP_ACCR)..... 4304
49.10.22 Header Log Register (PCIE_EP_HLR)..... 4305
49.10.23 VC Extended Capability Header (PCIE_EP_VCECHR)..... 4305
49.10.24 Port VC Capability Register 1 (PCIE_EP_PVCCR1)..... 4306
49.10.25 Port VC Capability Register 2 (PCIE_EP_PVCCR2)..... 4307
49.10.26 Port VC Control and Status Register (PCIE_EP_PVCCSR)..... 4308
49.10.27 VC Resource Capability Register n (PCIE_EP_VCRCR)..... 4310
49.10.28 VC Resource Control Register n (PCIE_EP_VCRConR)..... 4312
49.10.29 VC Resource Status Register n (PCIE_EP_VCRSR)..... 4314

49.11 PCIe CTRL RC Mode Memory Map/Register Definition....4315
49.11.1 Device ID and Vendor ID Register (PCIE_RC_DeviceID)....4318
49.11.2 Command and Status Register (PCIE_RC_Command)....4318
49.11.3 Revision ID and Class Code Register (PCIE_RC_RevID)....4321
49.11.4 BIST Register (PCIE_RC_BIST)....4321
49.11.5 Base Address 0 (PCIE_RC_BAR0)....4322
49.11.6 Base Address 1 (PCIE_RC_BAR1)....4325
49.11.7 Bus Number Registers (PCIE_RC_BNR)....4325
49.11.8 I/O Base Limit Secondary Status Register (PCIE_RC_IOBLSSR)....4327
49.11.9 Memory Base and Memory Limit Register (PCIE_RC_MEM_BLR)....4329
49.11.10 Prefetchable Memory Base and Limit Register (PCIE_RC_PREF_MEM_BLR)....4330
49.11.11 Prefetchable Base Upper 32 Bits Register (PCIE_RC_PREF_BASE_U32)....4330
49.11.12 Prefetchable Limit Upper 32 Bits Register (PCIE_RC_PREF_LIM_U32)....4331
49.11.13 I/O Base and Limit Upper 16 Bits Register (PCIE_RC_IO_BASE_LIM_U16)....4331
49.11.14 Capability Pointer Register (PCIE_RC_CAPPR)....4332
49.11.15 Expansion ROM Base Address Register (PCIE_RC_EROMBAR)....4332
49.11.16 Expansion ROM BAR Mask Register (PCIE_RC_EROMMASK)....4333
49.11.17 Power Management Capability Register (PCIE_RC_PMCR)....4334
49.11.18 Power Management Control and Status Register (PCIE_RC_PMCSR)....4337
49.11.19 PCI Express Capability ID Register (PCIE_RC_CIDR)....4338
49.11.20 Device Capabilities Register (PCIE_RC_DCR)....4341
49.11.21 Device Control Register (PCIE_RC_DConR)....4343
49.11.22 Link Capabilities Register (PCIE_RC_LCR)....4345
49.11.23 Link Control and Status Register (PCIE_RC_LCSR)....4348
49.11.24 Slot Capabilities Register (PCIE_RC_SCR)....4350
49.11.25 Slot Control and Status Register (PCIE_RC_SCSR)....4353
49.11.26 Root Control and Capabilities Register (PCIE_RC_RCCR)....4356
49.11.27 Root Status Register (PCIE_RC_RSR)....4358
49.11.28 Device Capabilities 2 Register (PCIE_RC_DCR2)....4359

49.11.29 Device Control and Status 2 Register (PCIE_RC_DCSR2)..... 4361
49.11.30 Link Capabilities 2 Register (PCIE_RC_LCR2)..... 4363
49.11.31 Link Control and Status 2 Register (PCIE_RC_LCSR2)..... 4365
49.11.32 AER Capability Header (PCIE_RC_AER)..... 4367
49.11.33 Uncorrectable Error Status Register (PCIE_RC_UESR)..... 4368
49.11.34 Uncorrectable Error Mask Register (PCIE_RC_UEMR)..... 4370
49.11.35 Uncorrectable Error Severity Register (PCIE_RC_UESevR)..... 4372
49.11.36 Correctable Error Status Register (PCIE_RC_CESR)..... 4374
49.11.37 Correctable Error Mask Register (PCIE_RC_CEMR)..... 4375
49.11.38 Advanced Capabilities and Control Register (PCIE_RC_ACCR)..... 4377
49.11.39 Header Log Register (PCIE_RC_HLR)..... 4378
49.11.40 Root Error Command Register (PCIE_RC_RECR)..... 4379
49.11.41 Root Error Status Register (PCIE_RC_RESR)..... 4380
49.11.42 Error Source Identification Register (PCIE_RC_ESIR)..... 4381
49.11.43 VC Extended Capability Header (PCIE_RC_VCECHR)..... 4382
49.11.44 Port VC Capability Register 1 (PCIE_RC_PVCCR1)..... 4383
49.11.45 Port VC Capability Register 2 (PCIE_RC_PVCCR2)..... 4384
49.11.46 Port VC Control and Status Register (PCIE_RC_PVCCSR)..... 4385
49.11.47 VC Resource Capability Register n (PCIE_RC_VCRCR)..... 4387
49.11.48 VC Resource Control Register n (PCIE_RC_VCRConR)..... 4389
49.11.49 VC Resource Status Register n (PCIE_RC_VCRSR)..... 4391
49.12 PCIe CTRL Port Logic Memory Map/Register Definition..... 4392
49.12.1 Ack Latency Timer and Replay Timer Register (PCIE_PL_ALTRTR)..... 4398
49.12.2 Vendor Specific DLLP Register (PCIE_PL_VSDR)..... 4398
49.12.3 Port Force Link Register (PCIE_PL_PFLR)..... 4399
49.12.4 Ack Frequency and L0-L1 ASPM Control Register (PCIE_PL_AFLACR)..... 4400
49.12.5 Port Link Control Register (PCIE_PL_PLCR)..... 4402
49.12.6 Lane Skew Register (PCIE_PL_LSR)..... 4404
49.12.7 Symbol Number Register (PCIE_PL_SNR)..... 4405

49.12.8 Symbol Timer Register and Filter Mask Register 1 (PCIE_PL_STRFM1)..... 4406
49.12.9 Filter Mask Register 2 (PCIE_PL_STRFM2).....4410
49.12.10 AMBA Multiple Outbound Decomposed NP Sub-Requests Control Register (PCIE_PL_AMODNPSR)..... 4411
49.12.11 Debug Register 0 (PCIE_PL_DEBUG0).....4412
49.12.12 Debug Register 1 (PCIE_PL_DEBUG1).....4412
49.12.13 Transmit Posted FC Credit Status Register (PCIE_PL_TPFCSR)..... 4413
49.12.14 Transmit Non-Posted FC Credit Status Register (PCIE_PL_TNFCSR)..... 4414
49.12.15 Transmit Completion FC Credit Status Register (PCIE_PL_TCFCSR)..... 4415
49.12.16 Queue Status Register (PCIE_PL_QSR)..... 4416
49.12.17 VC Transmit Arbitration Register 1 (PCIE_PL_VCTAR1).....4418
49.12.18 VC Transmit Arbitration Register 2 (PCIE_PL_VCTAR2).....4419
49.12.19 VC0 Posted Receive Queue Control (PCIE_PL_VC0PRQC).....4420
49.12.20 VC0 Non-Posted Receive Queue Control (PCIE_PL_VC0NRQC).....4422
49.12.21 VC0 Completion Receive Queue Control (PCIE_PL_VC0CRQC)..... 4423
49.12.22 VCn Posted Receive Queue Control (PCIE_PL_VC*n*PRQC).....4424
49.12.23 VCn Non-Posted Receive Queue Control (PCIE_PL_VC*n*NRQC).....4426
49.12.24 VCn Completion Receive Queue Control (PCIE_PL_VC*n*CRQC)..... 4427
49.12.25 VC0 Posted Buffer Depth (PCIE_PL_VC0PBD).....4428
49.12.26 VC0 Non-Posted Buffer Depth (PCIE_PL_VC0NPBD).....4429
49.12.27 VC0 Completion Buffer Depth (PCIE_PL_VC0CBD)..... 4430
49.12.28 VCn Posted Buffer Depth (PCIE_PL_VC*n*PBD).....4431
49.12.29 VCn Non-Posted Buffer Depth (PCIE_PL_VC*n*NPBD).....4432
49.12.30 VCn Completion Buffer Depth (PCIE_PL_VC*n*CBD)..... 4433
49.12.31 Gen2 Control Register (PCIE_PL_G2CR)..... 4433
49.12.32 PHY Status (PCIE_PL_PHY_STATUS).....4435
49.12.33 PHY Control (PCIE_PL_PHY_CTRL)..... 4435
49.12.34 Master Response Composer Control Register 0 (PCIE_PL_MRCCR0).....4436
49.12.35 Master Response Composer Control Register 1 (PCIE_PL_MRCCR1).....4437

49.12.36 MSI Controller Address (PCIE_PL_MSICA)..... 4438
49.12.37 MSI Controller Upper Address (PCIE_PL_MSICUA)..... 4438
49.12.38 MSI Controller Interrupt n Enable (PCIE_PL_MSICI*n*_ENB)..... 4439
49.12.39 MSI Controller Interrupt n Mask (PCIE_PL_MSICI*n*_MASK)..... 4439
49.12.40 MSI Controller Interrupt n Status (PCIE_PL_MSICI*n*_STATUS)..... 4440
49.12.41 MSI Controller General Purpose IO Register (PCIE_PL_MSICGPIO)..... 4440
49.12.42 iATU Viewport Register (PCIE_PL_iATUVR)..... 4441
49.12.43 iATU Region Control 1 Register (PCIE_PL_iATURC1)..... 4442
49.12.44 iATU Region Control 2 Register (PCIE_PL_iATURC2)..... 4444
49.12.45 iATU Region Lower Base Address Register (PCIE_PL_iATURLBA)..... 4447
49.12.46 iATU Region Upper Base Address Register (PCIE_PL_iATURUBA)..... 4448
49.12.47 iATU Region Limit Address Register (PCIE_PL_iATURLA)..... 4449
49.12.48 iATU Region Lower Target Address Register (PCIE_PL_iATURLTA)..... 4449
49.12.49 iATU Region Upper Target Address Register (PCIE_PL_iATURUTA)..... 4450

Chapter 50
PCI Express PHY (PCIe_PHY)

50.1 Overview..... 4451
50.2 Applications..... 4451
50.3 PCIe2 PHY Features..... 4451
50.3.1 Standards Compliance..... 4451
50.3.2 PHY Features..... 4452
50.4 External Signals..... 4452
50.5 Functional Description..... 4452
50.5.1 Clocks and Resets..... 4452
50.5.1.1 Reference Clock Enables..... 4453
50.5.1.2 Reference Clock Frequency Selection..... 4453
50.5.1.3 Spread Spectrum Clocking..... 4453
50.5.2 Termination Resistance Tuning..... 4453
50.5.3 Control Register Access..... 4454

50.6 System Operation....4455
50.6.1 Powering Up and Powering Down....4455
50.6.1.1 Power Up Requirements....4455
50.6.1.2 Power-On Reset....4456
50.6.1.3 Power Supply Sequencing when the PCIe 2.0 PHY is Not Used....4457
50.6.1.4 Power Down Requirements....4457
50.7 Control Memory Map/Register Definition....4457
50.7.1 Register ID Low 16 bits (PCIE_PHY_IDCODE_LO)....4461
50.7.2 Register ID High 16 bits (PCIE_PHY_IDCODE_HI)....4461
50.7.3 Debug Register (PCIE_PHY_DEBUG)....4462
50.7.4 Debug Register (PCIE_PHY_RTUNE_DEBUG)....4462
50.7.5 PCIE_PHY_RTUNE_STAT....4463
50.7.6 PCIE_PHY_SS_PHASE....4463
50.7.7 PCIE_PHY_SS_FREQ....4464
50.7.8 PCIE_PHY_ATEOVRD....4464
50.7.9 PCIE_PHY_MPLL_OVRD_IN_LO....4465
50.7.10 PCIE_PHY_MPLL_OVRD_IN_HI....4466
50.7.11 PCIE_PHY_SSC_OVRD_IN....4467
50.7.12 PCIE_PHY_BS_OVRD_IN....4468
50.7.13 PCIE_PHY_LEVEL_OVRD_IN....4469
50.7.14 PCIE_PHY_SUP_OVRD_OUT....4469
50.7.15 PCIE_PHY_MPLL_ASIC_IN....4470
50.7.16 PCIE_PHY_BS_ASIC_IN....4471
50.7.17 PCIE_PHY_LEVEL_ASIC_IN....4472
50.7.18 PCIE_PHY_SSC_ASIC_IN....4473
50.7.19 PCIE_PHY_SUP_ASIC_OUT....4473
50.7.20 PCIE_PHY_ATEOVRD_STATUS....4474
50.7.21 PCIE_PHY_SCOPE_ENABLES....4475
50.7.22 PCIE_PHY_SCOPE_SAMPLES....4476

50.7.23 PCIE_PHY_SCOPE_COUNT....4476
50.7.24 PCIE_PHY_SCOPE_CTL....4477
50.7.25 PCIE_PHY_SCOPE_MASK*n*....4477
50.7.26 PCIE_PHY_MPLL_LOOP_CTL....4478
50.7.27 PCIE_PHY_MPLL_ATB_MEAS2....4478
50.7.28 PCIE_PHY_MPLL_OVR....4479
50.7.29 PCIE_PHY_RTUNE_RTUNE_CTRL....4480
50.7.30 PCIE_PHY_TX_OVRD_IN_LO....4481
50.7.31 PCIE_PHY_TX_OVRD_IN_HI....4483
50.7.32 PCIE_PHY_TX_OVRD_DRV_LO....4484
50.7.33 PCIE_PHY_TX_OVRD_OUT....4484
50.7.34 PCIE_PHY_RX_OVRD_IN_LO....4485
50.7.35 PCIE_PHY_RX_OVRD_IN_HI....4486
50.7.36 PCIE_PHY_RX_OVRD_OUT....4487
50.7.37 PCIE_PHY_TX_ASIC_IN....4488
50.7.38 PCIE_PHY_TX_ASIC_DRV_LO....4489
50.7.39 PCIE_PHY_TX_ASIC_DRV_HI....4490
50.7.40 PCIE_PHY_TX_ASIC_OUT....4490
50.7.41 PCIE_PHY_RX_ASIC_IN....4491
50.7.42 PCIE_PHY_RX_ASIC_OUT....4492
50.7.43 PCIE_PHY_TX_VMD_FSM_TX_VCM_0....4493
50.7.44 PCIE_PHY_TX_VMD_FSM_TX_VCM_1....4493
50.7.45 PCIE_PHY_TX_VMD_FSM_TX_VCM_DEBUG_IN....4494
50.7.46 PCIE_PHY_TX_VMD_FSM_TX_VCM_DEBUG_OUT....4495
50.7.47 PCIE_PHY_TX_LBERT_CTL....4495
50.7.48 PCIE_PHY_RX_LBERT_CTL....4496
50.7.49 PCIE_PHY_RX_LBERT_ERR....4497
50.7.50 PCIE_PHY_RX_SCOPE_CTL....4497
50.7.51 PCIE_PHY_RX_SCOPE_PHASE....4498

50.7.52 PCIE_PHY_RX_DPLL_FREQ ... 4498
50.7.53 PCIE_PHY_RX_CDR_CTL ... 4499
50.7.54 PCIE_PHY_RX_CDR_CDR_FSM_DEBUG ... 4500
50.7.55 PCIE_PHY_RX_CDR_LOCK_VEC_OVRD ... 4501
50.7.56 PCIE_PHY_RX_CDR_LOCK_VEC ... 4502
50.7.57 PCIE_PHY_RX_CDR_ADAP_FSM ... 4502
50.7.58 PCIE_PHY_RX_ATB0 ... 4503
50.7.59 PCIE_PHY_RX_ATB1 ... 4504
50.7.60 PCIE_PHY_RX_ENPWR0 ... 4504
50.7.61 PCIE_PHY_RX_PMIX_PHASE ... 4505
50.7.62 PCIE_PHY_RX_ENPWR1 ... 4506
50.7.63 PCIE_PHY_RX_ENPWR2 ... 4507
50.7.64 PCIE_PHY_RX_SCOPE ... 4508
50.7.65 PCIE_PHY_TX_TXDRV_CNTRL ... 4509
50.7.66 PCIE_PHY_TX_POWER_CTL ... 4510
50.7.67 PCIE_PHY_TX_ALT_BLOCK ... 4511
50.7.68 PCIE_PHY_TX_ALT_AND_LOOPBACK ... 4512
50.7.69 PCIE_PHY_TX_TX_ATB_REG ... 4513

Chapter 51
Power Management Unit (PMU)

51.1 Overview ... 4515
51.2 Digital LDO Regulators ... 4517
51.3 Analog LDO Regulators ... 4518
51.3.1 LDO 1P1 ... 4518
51.3.2 LDO 2P5 ... 4519
51.3.3 Low Power Operation ... 4519
51.4 USB LDO Regulator ... 4520
51.5 SNVS Regulator ... 4520
51.6 Power Modes ... 4520

51.6.1 Reverse Well Biasing....4520
51.7 PMU Memory Map/Register Definition....4521
51.7.1 Regulator 1P1 Register (PMU_REG_1P1*n*)....4523
51.7.2 Regulator 3P0 Register (PMU_REG_3P0*n*)....4526
51.7.3 Regulator 2P5 Register (PMU_REG_2P5*n*)....4528
51.7.4 Digital Regulator Core Register (PMU_REG_CORE*n*)....4530
51.7.5 Miscellaneous Register 0 (PMU_MISC0*n*)....4533
51.7.6 Miscellaneous Register 1 (PMU_MISC1*n*)....4536
51.7.7 Miscellaneous Control Register (PMU_MISC2*n*)....4538

Chapter 52
Pulse Width Modulation (PWM)

52.1 Overview....4543
52.2 External Signals....4544
52.3 Clocks....4545
52.4 Functional Description....4546
52.4.1 Operation....4547
52.4.1.1 FIFO....4547
52.4.1.2 Rollover and Compare Event....4548
52.4.1.3 Low Power Mode Behavior....4548
52.4.1.4 Debug Mode Behavior....4548
52.5 Enable Sequence for the PWM....4548
52.6 Disable Sequence for the PWM....4549
52.7 PWM Memory Map/Register Definition....4549
52.7.1 PWM Control Register (PWM*x*_PWMCR)....4551
52.7.2 PWM Status Register (PWM*x*_PWMSR)....4553
52.7.3 PWM Interrupt Register (PWM*x*_PWMIR)....4554
52.7.4 PWM Sample Register (PWM*x*_PWMSAR)....4555
52.7.5 PWM Period Register (PWM*x*_PWMPR)....4556
52.7.6 PWM Counter Register (PWM*x*_PWMCNR)....4557

Chapter 53
Pixel Pipeline (PXP)

53.1 Overview....4559

53.2 Clocks....4560

53.3 Top-level architecture....4560

53.3.1 Processing Details....4562

53.3.2 Scaling Operation....4563

53.3.3 Decimation Image Scaling....4564

53.3.4 Bilinear Image Scaling Filter....4566

53.3.5 YUV 4:2:2 Image Scaling....4568

53.3.6 YUV 4:2:0 Image Scaling....4569

53.3.7 RGB/YUV444 Image Scaling....4571

53.3.8 Color Space Conversion (CSC)....4571

53.3.9 CSC1 Operation....4572

53.3.10 YUV versus YCbCr Support....4573

53.3.11 CSC2 operation....4573

53.3.12 Alpha Blending/Color Key....4574

53.3.13 Alpha Blend....4574

53.3.14 Color Key....4575

53.3.15 LUT....4575

53.3.16 Lookup Modes....4575

53.3.17 DIRECT_Y8....4576

53.3.18 DIRECT_RGB444....4576

53.3.19 DIRECT_RGB454....4576

53.3.20 CACHE_RGB565....4577

53.3.21 Output Modes....4578

53.3.22 Y8....4578

53.3.23 RGBW4444CFA....4579

53.3.23.1 CFA Correction....4579

53.3.24 RGB888 4580
53.3.25 Rotation 4580
53.3.26 Output Buffer 4583
53.3.27 Address calculator 4583
53.3.28 Block size selection 4583
53.3.29 Interlaced Video Support 4583
53.3.30 Theory of Operation 4584
53.3.31 Pixel Handling 4584
53.3.32 Output Buffer Composition 4585
53.3.33 PS Image Processing 4586
53.3.34 Letterboxing 4587
53.3.35 Clipping source images 4587
53.3.36 Color Key Processing 4589
53.3.37 In Place Processing (PS buffer is destination buffer) 4591
53.3.38 Alpha Surface (AS) Processing 4591
53.3.39 Alpha Handling 4591
53.3.40 Color Key Processing (AS_CTRL) 4591
53.4 Output Image Processing 4592
53.4.1 Output Image Size 4592
53.4.2 Output Format 4592
53.4.3 Rotation/Flip operations 4592
53.5 Queuing PXP transactions 4593
53.6 Error Handling 4593
53.6.1 Known PXP Limitations/Issues 4594
53.7 PXP Memory Map/Register Definition 4594
53.7.1 Control Register 0 (PXP_CTRL*n*) 4597
53.7.2 Status Register (PXP_STAT*n*) 4600
53.7.3 Output Buffer Control Register (PXP_OUT_CTRL*n*) 4601
53.7.4 Output Frame Buffer Pointer (PXP_OUT_BUF) 4603

53.7.5 Output Frame Buffer Pointer #2 (PXP_OUT_BUF2)..... 4604
53.7.6 Output Buffer Pitch (PXP_OUT_PITCH)..... 4605
53.7.7 Output Surface Lower Right Coordinate (PXP_OUT_LRC)..... 4605
53.7.8 Processed Surface Upper Left Coordinate (PXP_OUT_PS_ULC)..... 4606
53.7.9 Processed Surface Lower Right Coordinate (PXP_OUT_PS_LRC)..... 4607
53.7.10 Alpha Surface Upper Left Coordinate (PXP_OUT_AS_ULC)..... 4608
53.7.11 Alpha Surface Lower Right Coordinate (PXP_OUT_AS_LRC)..... 4609
53.7.12 Processed Surface (PS) Control Register (PXP_PS_CTRL*n*)..... 4610
53.7.13 PS Input Buffer Address (PXP_PS_BUF)..... 4612
53.7.14 PS U/Cb or 2 Plane UV Input Buffer Address (PXP_PS_UBUF)..... 4612
53.7.15 PS V/Cr Input Buffer Address (PXP_PS_VBUF)..... 4613
53.7.16 Processed Surface Pitch (PXP_PS_PITCH)..... 4614
53.7.17 PS Background Color (PXP_PS_BACKGROUND)..... 4614
53.7.18 PS Scale Factor Register (PXP_PS_SCALE)..... 4615
53.7.19 PS Scale Offset Register (PXP_PS_OFFSET)..... 4616
53.7.20 PS Color Key Low (PXP_PS_CLRKEYLOW)..... 4617
53.7.21 PS Color Key High (PXP_PS_CLRKEYHIGH)..... 4618
53.7.22 Alpha Surface Control (PXP_AS_CTRL)..... 4619
53.7.23 Alpha Surface Buffer Pointer (PXP_AS_BUF)..... 4621
53.7.24 Alpha Surface Pitch (PXP_AS_PITCH)..... 4622
53.7.25 Overlay Color Key Low (PXP_AS_CLRKEYLOW)..... 4622
53.7.26 Overlay Color Key High (PXP_AS_CLRKEYHIGH)..... 4623
53.7.27 Color Space Conversion Coefficient Register 0 (PXP_CSC1_COEF0)..... 4624
53.7.28 Color Space Conversion Coefficient Register 1 (PXP_CSC1_COEF1)..... 4625
53.7.29 Color Space Conversion Coefficient Register 2 (PXP_CSC1_COEF2)..... 4626
53.7.30 Color Space Conversion Control Register (PXP_CSC2_CTRL)..... 4627
53.7.31 Color Space Conversion Coefficient Register 0 (PXP_CSC2_COEF0)..... 4628
53.7.32 Color Space Conversion Coefficient Register 1 (PXP_CSC2_COEF1)..... 4629
53.7.33 Color Space Conversion Coefficient Register 2 (PXP_CSC2_COEF2)..... 4629

53.7.34 Color Space Conversion Coefficient Register 3 (PXP_CSC2_COEF3)..... 4630
53.7.35 Color Space Conversion Coefficient Register 4 (PXP_CSC2_COEF4)..... 4631
53.7.36 Color Space Conversion Coefficient Register 5 (PXP_CSC2_COEF5)..... 4631
53.7.37 Lookup Table Control Register (PXP_LUT_CTRL).....4632
53.7.38 Lookup Table Control Register (PXP_LUT_ADDR)..... 4634
53.7.39 Lookup Table Data Register (PXP_LUT_DATA)..... 4635
53.7.40 Lookup Table External Memory Address Register (PXP_LUT_EXTMEM)..... 4636
53.7.41 Color Filter Array Register (PXP_CFA)..... 4637
53.7.42 Histogram Control Register (PXP_HIST_CTRL)..... 4637
53.7.43 2-level Histogram Parameter Register (PXP_HIST2_PARAM)..... 4638
53.7.44 4-level Histogram Parameter Register (PXP_HIST4_PARAM)..... 4639
53.7.45 8-level Histogram Parameter 0 Register (PXP_HIST8_PARAM0)..... 4640
53.7.46 8-level Histogram Parameter 1 Register (PXP_HIST8_PARAM1)..... 4641
53.7.47 16-level Histogram Parameter 0 Register (PXP_HIST16_PARAM0)..... 4642
53.7.48 16-level Histogram Parameter 1 Register (PXP_HIST16_PARAM1)..... 4643
53.7.49 16-level Histogram Parameter 2 Register (PXP_HIST16_PARAM2)..... 4644
53.7.50 16-level Histogram Parameter 3 Register (PXP_HIST16_PARAM3)..... 4645
53.7.51 PXP Power Control Register (PXP_POWER).....4646
53.7.52 Next Frame Pointer (PXP_NEXT)..... 4646

Chapter 54
ROM Controller with Patch (ROMC)

54.1 Overview.....4649
54.1.1 Features..... 4650
54.1.2 Modes of Operation..... 4650
54.1.2.1 Low Power Mode.....4651
54.2 Clocks..... 4651
54.3 Memory Map..... 4651
54.3.1 ROM Memory Map in detail..... 4651
54.4 Functional Description.....4652

54.4.1 ROM Controller (ROMC) Functional Description....4652
54.4.1.1 Functionality overview....4652
54.4.2 ROMC Functional Description....4652
54.4.2.1 ROMC Disabling....4653
54.4.2.2 ROMC Event Priority....4653
54.4.2.3 Data Fixing....4653
54.4.2.4 Opcode Patching....4654
54.4.2.4.1 Typical Software Response to Opcode Patch....4655
54.4.2.5 External Boot Feature....4656
54.4.2.6 Alternate Masters and ROMC....4657
54.5 ROMCP Memory Map/Register Definition....4657
54.5.1 ROMC Data Registers (ROMC_ROMPATCH*n*D)....4658
54.5.2 ROMC Control Register (ROMC_ROMPATCHCNTL)....4659
54.5.3 ROMC Enable Register High (ROMC_ROMPATCHENH)....4660
54.5.4 ROMC Enable Register Low (ROMC_ROMPATCHENL)....4660
54.5.5 ROMC Address Registers (ROMC_ROMPATCH*n*A)....4661
54.5.6 ROMC Status Register (ROMC_ROMPATCHSR)....4662

Chapter 55
Smart Direct Memory Access Controller (SDMA)

55.1 Overview....4665
55.1.1 Block Diagram....4665
55.1.2 Features....4667
55.2 External Signals....4669
55.3 Clocks....4669
55.4 Functional Description....4669
55.4.1 SDMA Core....4670
55.4.1.1 SDMA Core Structure....4671
55.4.1.2 Program Control Unit (PCU)....4674
55.4.1.2.1 Instruction Types....4674

55.4.1.2.2 PCU States ... 4675
55.4.1.3 SDMA Core Memory ... 4678
55.4.2 Scheduler ... 4678
55.4.2.1 Primary Functions ... 4678
55.4.2.2 Channels and DMA Requests ... 4679
55.4.2.2.1 Channels ... 4679
55.4.2.2.2 DMA Requests ... 4679
55.4.2.2.3 Mapping from DMA Requests to Channels and Priorities ... 4679
55.4.2.3 Scheduler Functional Description ... 4679
55.4.2.3.1 Scheduler Overview ... 4679
55.4.2.3.2 DMA Requests Scanning ... 4680
55.4.2.3.3 Mapping DMA Requests to Pending Channels ... 4681
55.4.2.3.4 Channel Overflow ... 4684
55.4.2.3.5 Runnable Channels Evaluation ... 4684
55.4.2.3.6 Next Channel Decision Tree ... 4686
55.4.2.3.7 Scheduler State Diagram ... 4688
55.4.2.3.8 Scheduler Pipeline Timing Diagram ... 4690
55.4.2.3.9 Channel-DMA Request Mapping ... 4690
55.4.2.3.10 Examples: How to Start a Channel ... 4690
55.4.2.4 Context Switching ... 4691
55.4.2.4.1 Context Switch Modes ... 4692
55.4.2.4.2 Context Switch Procedure ... 4692
55.4.2.4.3 Context Map in Memory ... 4694
55.4.3 Functional Units ... 4694
55.4.3.1 Burst DMA Unit ... 4694
55.4.3.1.1 Burst DMA Structure ... 4695
55.4.3.1.2 Burst DMA Registers ... 4696
55.4.3.1.3 Burst DMA Data Transfers ... 4697
55.4.3.1.3.1 Data Retrieval from the Arm platform Memory ... 4697

55.4.3.1.3.2 Storing Data Into the Arm platform Memory....4697
55.4.3.1.3.3 Transferring Data Between Two Arm platform Memory Locations-Burst DMA Unit....4698
55.4.3.2 Peripheral DMA Unit....4698
55.4.3.2.1 Peripheral DMA Structure....4699
55.4.3.2.2 Peripheral DMA Registers....4700
55.4.3.2.3 Peripheral DMA Data Transfers....4701
55.4.3.2.3.1 Data Retrieval from the Arm platform Memory or Peripheral....4701
55.4.3.2.3.2 Storing Data into the Arm platform Memory or Peripheral....4701
55.4.3.2.3.3 Transferring Data Between Two Arm platform Memory Locations-Peripheral DMA Unit....4702
55.4.4 SDMA Security Support....4702
55.4.4.1 Locked Mode....4702
55.4.5 OnCE and PCU Debug States....4703
55.4.6 SDMA Clocks and Low Power Modes....4705
55.4.6.1 Clock Gating and Low Power Modes....4706
55.4.6.1.1 Coarse Clock Gating....4706
55.4.6.1.2 Refined Clock Gating....4707
55.4.6.1.3 Low Power Modes and User Control....4707
55.4.6.1.3.1 SLEEP Mode....4708
55.4.6.1.3.2 RUN Mode....4708
55.4.6.1.3.3 DEBUG Mode....4709
55.4.6.1.4 Stop Mode Response....4709
55.4.6.2 Reset....4709
55.4.7 Software Interface....4709
55.4.8 Initialization Information....4710
55.4.8.1 Hardware Reset....4710
55.4.8.2 Channel Script Execution....4711
55.4.8.3 Initialization and Script Execution Setup Sequence....4711

55.4.9 SDMA Programming Model....4712
55.4.9.1 State and Registers Per Channel....4712
55.4.9.2 General Purpose Registers....4713
55.4.9.3 Functional Unit State....4713
55.4.9.3.1 Program Counter Register (PC)....4713
55.4.9.3.2 Flags....4713
55.4.9.3.3 Return Program Counter (RPC)....4714
55.4.9.3.4 Loop Mode Start Program Counter (SPC)....4714
55.4.9.3.5 Loop Mode End Program Counter (EPC)....4714
55.4.9.4 Context Switching-Programming....4715
55.4.9.5 Address Space....4716
55.4.9.5.1 Instruction Memory Map....4717
55.4.9.5.2 Data Memory Map....4717
55.4.10 SDMA Initialization....4719
55.4.10.1 Hardware Reset-SDMA....4719
55.4.10.2 Standard Boot Sequence....4719
55.4.10.3 User-Defined Boot Sequence....4720
55.4.10.4 Script Loading and Context Initialization....4720
55.4.11 Instruction Description....4721
55.4.11.1 Scheduling Instructions....4721
55.4.11.2 Conditional Branch Instructions....4721
55.4.11.3 Unconditional Jump Instructions....4722
55.4.11.4 Subroutine Return Instructions....4722
55.4.11.5 Loop Instruction....4722
55.4.11.6 Miscellaneous Instructions....4723
55.4.11.7 Logic Instructions....4723
55.4.11.8 Arithmetic Instructions....4723
55.4.11.9 Compare Instructions....4724
55.4.11.10 Test Instructions....4724

55.4.11.11 Byte Permutation Instructions....4724
55.4.11.12 Bit Shift Instructions....4725
55.4.11.13 Bit Manipulation Instructions....4725
55.4.11.14 SDMA Memory Access Instructions....4725
55.4.11.15 Functional Unit Instructions....4726
55.4.11.16 Illegal Instructions....4726
55.4.11.17 Debug Instructions....4726
55.4.12 Functional Units Programming Model....4727
55.4.12.1 Burst DMA Unit Programming....4728
55.4.12.1.1 Memory Source Address Register (MSA)....4728
55.4.12.1.2 Memory Destination Address Register (MDA)....4729
55.4.12.1.3 Memory Data Buffer Register (MD)....4729
55.4.12.1.4 State Register (MS)....4730
55.4.12.1.5 Burst DMA Write (stf)....4731
55.4.12.1.6 Burst DMA Read (ldf)....4734
55.4.12.1.7 Prefetch/Flush and Auto-Flush Management-Burst DMA Unit....4735
55.4.12.1.8 Data Alignment and Endianness-Burst DMA Unit....4737
55.4.12.1.8.1 Burst DMA in Read Mode....4737
55.4.12.1.8.2 Burst DMA in Write Mode....4738
55.4.12.1.8.3 Endianness-Burst DMA Unit....4739
55.4.12.1.9 Burst DMA Unit Copy Mode....4740
55.4.12.1.10 Burst DMA Unit Error Management....4741
55.4.12.1.11 Conditional Yielding-Burst DMA Unit....4742
55.4.12.2 Peripheral DMA Unit Programming....4743
55.4.12.2.1 Peripheral Source Address Register (PSA)....4744
55.4.12.2.2 Peripheral Destination Address Register (PDA)....4745
55.4.12.2.3 Peripheral Data Register (PD)....4745
55.4.12.2.4 Peripheral State Register (PS)....4746
55.4.12.2.5 Peripheral DMA Write (stf)-Write Mode....4747

55.4.12.2.6 Peripheral DMA Read (ldf)-Read Mode....4750
55.4.12.2.7 Peripheral DMA Unit Copy Mode....4751
55.4.12.2.8 Error Management....4752
55.4.12.2.8.1 Immediate Errors....4752
55.4.12.2.8.2 Data Transfer Errors....4752
55.4.12.2.8.3 Read Error (First Phase)....4753
55.4.12.2.8.4 Write Error and Read Error (Second Phase)....4753
55.4.12.2.8.5 Copy Mode Errors....4754
55.4.12.2.8.6 Error Check Example....4754
55.4.12.2.9 Peripheral DMA Unit Prefetch/Flush Management....4755
55.4.12.3 OnCE and Real-Time Debug....4755
55.4.12.3.1 Memory and Register Access....4756
55.4.12.3.2 Hardware Breakpoints....4756
55.4.12.3.3 Watchpoints....4756
55.4.12.3.4 Software Breakpoints....4756
55.4.12.3.5 Core Control....4756
55.4.13 The OnCE Controller....4756
55.4.13.1 OnCE Commands....4757
55.4.13.2 Sending Commands to the OnCE Controller....4758
55.4.13.2.1 Using the JTAG Interface....4758
55.4.13.2.2 Using the Arm platform....4758
55.4.13.2.3 Conflicts Between the JTAG and the Arm platform Accesses....4760
55.4.13.3 Executing a Command from the OnCE....4760
55.4.13.3.1 Nature of the Commands....4760
55.4.13.3.2 Execution Request....4761
55.4.13.3.3 Command Execution....4761
55.4.13.4 Registers Descriptions....4763
55.4.13.4.1 Event Cell Counter Register (ECOUNT)....4763
55.4.13.4.2 Event Cell Address Registers (EAA or EAB)....4763

55.4.13.4.3 Event Cell Address Mask Register (EAM)....4764
55.4.13.4.4 Event Cell Data Register (ED)....4764
55.4.13.4.5 Event Cell Data Mask Register (EDM)....4764
55.4.13.4.6 Real Time Buffer Register (RTB)....4764
55.4.13.4.7 Event Control Register (ECTL)....4765
55.4.13.4.8 Trace Buffer (TB)....4765
55.4.13.4.9 OnCE Status Register (OSTAT)....4765
55.4.13.5 JTAG Interface Requirements....4766
55.4.13.5.1 TCK Speed Limitation....4766
55.4.13.5.2 Synchronization Implementation....4766
55.4.13.5.3 JTAG Controller Start-Up Recommended Procedure....4768
55.4.14 Using the OnCE....4768
55.4.14.1 Activating Clocks in Debug Mode....4768
55.4.14.2 Getting the Current Status....4768
55.4.14.3 Methods of Entering Debug Mode....4769
55.4.14.3.1 External Debug Request During Reset....4769
55.4.14.3.2 Debug Request During Normal Activity....4769
55.4.14.3.3 Software Breakpoint Instruction....4769
55.4.14.3.4 Event Detection Unit Matching Condition....4769
55.4.14.4 Executing Instructions in Debug Mode....4770
55.4.14.5 Command Sequences Examples....4770
55.4.14.5.1 Getting the SDMA Status....4770
55.4.14.5.2 Saving the Context....4771
55.4.14.5.3 Restoring the Context....4772
55.4.14.5.4 Accessing the Memory....4773
55.4.14.5.5 Resuming Program Execution....4774
55.4.14.5.6 Single Stepping in RAM....4774
55.4.14.5.7 Single Stepping in ROM....4775
55.4.14.6 OnCE Event Detection Unit....4775

55.4.14.7 Clock Gating and Reset....4776
55.4.14.7.1 Clocks....4776
55.4.14.7.2 Resets....4777
55.4.14.8 Real Time Features....4777
55.4.14.8.1 Trace Buffer....4777
55.4.14.8.2 Real Time Buffer....4779
55.4.14.8.3 Emulation Pin....4779
55.4.14.8.4 Real-Time Debug Outputs....4779
55.5 Instruction Set....4783
55.5.1 Instruction Encoding....4783
55.5.2 SDMA Instruction Set....4785
55.5.2.1 ADD (Addition)....4786
55.5.2.2 ADDI (Add with Immediate Value)....4787
55.5.2.3 AND (Logical AND)....4788
55.5.2.4 ANDI (Logical AND with Immediate Value)....4789
55.5.2.5 ANDN (Logical AND NOT)....4790
55.5.2.6 ANDNI (Logical AND with Negated Immediate Value)....4791
55.5.2.7 ASR1 (Arithmetic Shift Right by 1 Bit)....4792
55.5.2.8 BCLRI1 (Bit Clear Immediate)....4793
55.5.2.9 BDF (Conditional Branch if Destination Fault)....4794
55.5.2.10 BF (Conditional Branch if False)....4795
55.5.2.11 BSETI (Bit Set Immediate)....4796
55.5.2.12 BSF (Conditional Branch if Source Fault)....4797
55.5.2.13 BT (Conditional Branch if True)....4798
55.5.2.14 BTSTI (Bit Test immediate)....4799
55.5.2.15 CLRF (Clear Arm platform flags)....4800
55.5.2.16 CMPEQ (Compare for Equal)....4801
55.5.2.17 CMPEQI (Compare with Immediate for Equal)....4802
55.5.2.18 CMPHS (Compare for Higher or Same)....4803

55.5.2.19 CMPLT (Compare for Less Than)....4804
55.5.2.20 cpShReg (Update Context of PCU Registers and Flag)....4805
55.5.2.21 DONE (DONE, Yield)....4805
55.5.2.22 ILLEGAL (ILLEGAL Instruction)....4807
55.5.2.23 JMP (Unconditional Jump Immediate)....4808
55.5.2.24 JMPR (Unconditional Jump)....4808
55.5.2.25 JSR (Unconditional Jump to Subroutine Immediate)....4809
55.5.2.26 JSRR (Unconditional Jump to Subroutine)....4810
55.5.2.27 LD (Load Register)....4811
55.5.2.28 LDF (Load Register from Functional Unit)....4812
55.5.2.29 LDI (Load Register with Immediate Value)....4814
55.5.2.30 LDRPC (Load from RPC to Register)....4815
55.5.2.31 LOOP (Hardware Loop)....4816
55.5.2.32 LSL1 (Logical Shift Left by 1 Bit)....4818
55.5.2.33 LSR1 (Logical Shift Right by 1 Bit)....4819
55.5.2.34 MOV (Logical Move)....4820
55.5.2.35 NOTIFY (Notify to Arm platform)....4821
55.5.2.36 OR (Logical OR)....4822
55.5.2.37 ORI (Logical OR with Immediate Value)....4823
55.5.2.38 RET (Return from Subroutine)....4824
55.5.2.39 REVB (Reverse Byte Order)....4825
55.5.2.40 Reverse Low Order Bytes(REVBLO)....4825
55.5.2.41 ROR1 (Rotate Right by 1 Bit)....4826
55.5.2.42 RORB (Rotate Right by 1 Byte)....4827
55.5.2.43 SOFTBKPT (Software Breakpoint)....4828
55.5.2.44 ST (Store Register)....4828
55.5.2.45 STF (Store Register in Functional Unit)....4830
55.5.2.46 SUB (Subtract)....4833
55.5.2.47 SUBI (Subtract with Immediate)....4834

55.5.2.48 TST (Test with Zero)..... 4835
55.5.2.49 TSTI (Test Immediate).....4836
55.5.2.50 XOR (Logical Exclusive OR)..... 4837
55.5.2.51 XORI (Exclusive OR with Immediate).....4838
55.5.2.52 YIELD, YIELDGE (DONE, Yield).....4839
55.6 Software Restrictions.....4839
55.6.1 Unsupported Burst DMA Access Sequence..... 4839
55.7 Application Notes..... 4840
55.7.1 Data Structures for Boot Code and Channel Scripts.....4840
55.7.1.1 Buffer Descriptor Format.....4841
55.7.1.2 Buffer Descriptor Commands for Bootload scripts..... 4844
55.7.1.3 Example of Buffer Descriptors for Channel 0..... 4845
55.7.1.4 Channel Context.....4848
55.7.2 Typical Data Transfer Supported by SDMA DMA Units..... 4848
55.7.2.1 External Memory to External Memory.....4849
55.7.2.2 Peripheral to Peripheral Transfer..... 4850
55.7.2.2.1 Source and Destination Target Have the Same Data Path Width.....4850
55.7.2.2.2 Source and Destination Target Have a Different Data Path Width..... 4851
55.7.2.3 Transfer Between Peripheral and External Memory.....4852
55.7.2.3.1 Peripheral to External Memory Transfer..... 4852
55.7.2.3.2 External Memory to Peripheral Transfer..... 4854
55.7.2.4 Transfer Between External Memory and Internal Memory.....4855
55.7.2.4.1 Internal Memory to Internal Memory.....4855
55.7.2.4.2 Transfer Between Peripheral and Internal Memory.....4855
55.8 Arm Platform Memory Map and Control Register Definitions.....4855
55.8.1 Arm platform Channel 0 Pointer (SDMAARM_MC0PTR).....4861
55.8.2 Channel Interrupts (SDMAARM_INTR)..... 4861
55.8.3 Channel Stop/Channel Status (SDMAARM_STOP_STAT).....4861
55.8.4 Channel Start (SDMAARM_HSTART).....4862

55.8.5 Channel Event Override (SDMAARM_EVTOVR)..... 4862
55.8.6 Channel BP Override (SDMAARM_DSPOVR)..... 4863
55.8.7 Channel Arm platform Override (SDMAARM_HOSTOVR)..... 4863
55.8.8 Channel Event Pending (SDMAARM_EVTPEND)..... 4863
55.8.9 Reset Register (SDMAARM_RESET)..... 4864
55.8.10 DMA Request Error Register (SDMAARM_EVTERR)..... 4865
55.8.11 Channel Arm platform Interrupt Mask (SDMAARM_INTRMASK)..... 4865
55.8.12 Schedule Status (SDMAARM_PSW)..... 4866
55.8.13 DMA Request Error Register (SDMAARM_EVTERRDBG)..... 4866
55.8.14 Configuration Register (SDMAARM_CONFIG)..... 4867
55.8.15 SDMA LOCK (SDMAARM_SDMA_LOCK)..... 4868
55.8.16 OnCE Enable (SDMAARM_ONCE_ENB)..... 4869
55.8.17 OnCE Data Register (SDMAARM_ONCE_DATA)..... 4870
55.8.18 OnCE Instruction Register (SDMAARM_ONCE_INSTR)..... 4870
55.8.19 OnCE Status Register (SDMAARM_ONCE_STAT)..... 4870
55.8.20 OnCE Command Register (SDMAARM_ONCE_CMD)..... 4872
55.8.21 Illegal Instruction Trap Address (SDMAARM_ILLINSTADDR)..... 4873
55.8.22 Channel 0 Boot Address (SDMAARM_CHN0ADDR)..... 4873
55.8.23 DMA Requests (SDMAARM_EVT_MIRROR)..... 4874
55.8.24 DMA Requests 2 (SDMAARM_EVT_MIRROR2)..... 4874
55.8.25 Cross-Trigger Events Configuration Register 1 (SDMAARM_XTRIG_CONF1)..... 4875
55.8.26 Cross-Trigger Events Configuration Register 2 (SDMAARM_XTRIG_CONF2)..... 4877
55.8.27 Channel Priority Registers (SDMAARM_SDMA_CHNPRI*n*)..... 4878
55.8.28 Channel Enable RAM (SDMAARM_CHNENBL*n*)..... 4878
55.9 BP Memory Map and Control Register Definitions..... 4879
55.9.1 Channel 0 Pointer (SDMABP_DC0PTR)..... 4879
55.9.2 Channel Interrupts (SDMABP_INTR)..... 4880
55.9.3 Channel Stop/Channel Status (SDMABP_STOP_STAT)..... 4880
55.9.4 Channel Start (SDMABP_DSTART)..... 4881

55.9.5 DMA Request Error Register (SDMABP_EVTERR)....4881
55.9.6 Channel DSP Interrupt Mask (SDMABP_INTRMASK)....4882
55.9.7 DMA Request Error Register (SDMABP_EVTERRDBG)....4882
55.10 SDMA Internal (Core) Memory Map and Internal Register Definitions....4882
55.10.1 Arm platform Channel 0 Pointer (SDMACORE_MC0PTR)....4884
55.10.2 Current Channel Pointer (SDMACORE_CCPTR)....4884
55.10.3 Current Channel Register (SDMACORE_CCR)....4884
55.10.4 Highest Pending Channel Register (SDMACORE_NCR)....4885
55.10.5 External DMA Requests Mirror (SDMACORE_EVENTS)....4886
55.10.6 Current Channel Priority (SDMACORE_CCPRI)....4887
55.10.7 Next Channel Priority (SDMACORE_NCPRI)....4887
55.10.8 OnCE Event Cell Counter (SDMACORE_ECOUNT)....4888
55.10.9 OnCE Event Cell Control Register (SDMACORE_ECTL)....4888
55.10.10 OnCE Event Address Register A (SDMACORE_EAA)....4890
55.10.11 OnCE Event Cell Address Register B (SDMACORE_EAB)....4890
55.10.12 OnCE Event Cell Address Mask (SDMACORE_EAM)....4890
55.10.13 OnCE Event Cell Data Register (SDMACORE_ED)....4891
55.10.14 OnCE Event Cell Data Mask (SDMACORE_EDM)....4891
55.10.15 OnCE Real-Time Buffer (SDMACORE_RTB)....4892
55.10.16 OnCE Trace Buffer (SDMACORE_TB)....4892
55.10.17 OnCE Status (SDMACORE_OSTAT)....4893
55.10.18 Channel 0 Boot Address (SDMACORE_MCHN0ADDR)....4895
55.10.19 ENDIAN Status Register (SDMACORE_ENDIANNESS)....4896
55.10.20 Lock Status Register (SDMACORE_SDMA_LOCK)....4897
55.10.21 External DMA Requests Mirror #2 (SDMACORE_EVENTS2)....4898
55.11 SDMA Peripheral Registers....4898

Chapter 56
System JTAG Controller (SJC)

56.1 Overview....4899

56.1.1 Features ... 4900
56.1.2 Modes of Operation ... 4901
56.2 External Signals ... 4903
56.2.1 External Signal Overview ... 4904
56.2.2 TAP Controller ... 4905
56.2.3 Accessing ExtraDebug Registers ... 4907
56.3 TAP Selection Block (TSB) ... 4909
56.3.1 Select Mode Using Software ... 4910
56.4 Boundary Scan Register (BSR) ... 4911
56.5 SJC Instruction Register (SJIR) ... 4911
56.5.1 SJC ID_CODE Instruction (SJC IDCODE) ... 4912
56.5.2 SAMPLE/PRELOAD Instruction ... 4913
56.5.3 EXTEST Instruction ... 4914
56.5.4 HIGHZ Instruction ... 4914
56.5.5 BYPASS Instruction ... 4914
56.5.6 ENABLE_ExtraDebug Instruction ... 4915
56.5.7 ENTER_DEBUG instruction ... 4915
56.5.8 TAP Select Instruction ... 4916
56.5.9 EXTEST_PULSE instruction ... 4916
56.5.10 EXTEST_TRAIN instruction ... 4916
56.6 Security ... 4917
56.6.1 JTAG Security Modes ... 4917
56.6.1.1 Mode 1: No Debug - Maximum Security ... 4917
56.6.1.2 Mode 2: Secure JTAG - High Security ... 4918
56.6.1.2.1 Challenge/Response Mechanism in System JTAG Mode ... 4918
56.6.1.3 Mode 3: JTAG Enabled - Low Security ... 4919
56.6.2 Software Enabled JTAG ... 4919
56.6.3 Kill Trace ... 4920
56.6.4 SJC Disable Fuse ... 4921

56.7 Functional Description....4922
56.7.1 Static Core Debug....4922
56.7.2 Reset Mechanism....4922
56.8 Initialization/Application Information....4923
56.9 SJC Memory Map/Register Definition....4924
56.9.1 General Purpose Unsecured Status Register 1 (SJC_GPUSR1)....4925
56.9.2 General Purpose Unsecured Status Register 2 (SJC_GPUSR2)....4927
56.9.3 General Purpose Unsecured Status Register 3 (SJC_GPUSR3)....4927
56.9.4 General Purpose Secured Status Register (SJC_GPSSR)....4928
56.9.5 Debug Control Register (SJC_DCR)....4929
56.9.6 Security Status Register (SJC_SSR)....4931
56.9.7 General Purpose Clocks Control Register (SJC_GPCCR)....4934

Chapter 57
Secure Non-Volatile Storage (SNVS)

57.1 SNVS overview....4935
57.1.1 SNVS features....4935
57.1.2 Modes of operation....4936
57.2 External signals....4936
57.3 Clocks....4936
57.4 SNVS structure....4937
57.4.1 SNVS_HP (high-power domain)....4938
57.4.2 Non-secure real-time counter....4938
57.4.2.1 Calibrating the time counter....4938
57.4.2.2 Time counter alarm....4939
57.4.2.3 Periodic interrupt....4939
57.5 SNVS_LP (low-power domain)....4940
57.5.1 Behavior during system power down....4940
57.5.2 Monotonic Counter (MC)....4940
57.6 SNVS reset and system powerup....4941

57.6.1 PMIC interface....4942
57.7 SNVS interrupts and alarms....4943
57.8 Programming guidelines....4944
57.8.1 RTC control bits setting....4944
57.8.2 RTC value read....4945
57.8.3 General initialization guidelines....4945
57.9 SNVS Memory Map/Register Definition....4946
57.9.1 SNVS_HP Lock register (SNVS_HPLR)....4948
57.9.2 SNVS_HP Command register (SNVS_HPCOMR)....4950
57.9.3 SNVS_HP Control register (SNVS_HPCR)....4952
57.9.4 SNVS_HP Status register (SNVS_HPSR)....4955
57.9.5 SNVS_HP Real-Time Counter MSB Register (SNVS_HPRTCMR)....4957
57.9.6 SNVS_HP Real-Time Counter LSB Register (SNVS_HPRTCLR)....4958
57.9.7 SNVS_HP Time Alarm MSB Register (SNVS_HPTAMR)....4958
57.9.8 SNVS_HP Time Alarm LSB Register (SNVS_HPTALR)....4959
57.9.9 SNVS_LP Lock Register (SNVS_LPLR)....4960
57.9.10 SNVS_LP Control Register (SNVS_LPCR)....4962
57.9.11 SNVS_LP Status Register (SNVS_LPSR)....4964
57.9.12 SNVS_LP Secure Monotonic Counter MSB Register (SNVS_LPSMCMR)....4966
57.9.13 SNVS_LP Secure Monotonic Counter LSB Register (SNVS_LPSMCLR)....4967
57.9.14 SNVS_LP General-Purpose Register (SNVS_LPGPR)....4967
57.9.15 SNVS_HP Version ID Register 1 (SNVS_HPVIDR1)....4968
57.9.16 SNVS_HP Version ID Register 2 (SNVS_HPVIDR2)....4968

Chapter 58
Shared Peripheral Bus Arbiter (SPBA)

58.1 Overview....4971
58.1.1 Features....4972
58.1.2 Modes of operation....4973
58.2 Clocks....4973

58.3 Functional description....4974
58.3.1 Masters arbitration....4974
58.4 Resource ownership control....4977
58.4.1 Access control....4977
58.4.1.1 Peripheral access....4977
58.4.1.2 Peripheral Right Register access....4978
58.4.2 Owner election....4979
58.4.3 Ending ownership....4979
58.4.3.1 Software Controlled Ownership Ending....4979
58.4.4 The Un-owned State....4980
58.5 SPBA Memory Map/Register Definition....4980
58.5.1 Peripheral Rights Register (SPBA_PRR*n*)....4982

Chapter 59
Sony/Philips Digital Interface (SPDIF)

59.1 Overview....4985
59.2 External Signals....4987
59.3 Clocks....4987
59.4 Functional Description....4987
59.4.1 SPDIF Receiver....4989
59.4.1.1 Audio Data Reception....4989
59.4.1.1.1 Application Note....4991
59.4.1.2 Channel Status Reception....4992
59.4.1.2.1 Channel Status Interrupt....4992
59.4.1.3 User Bit Reception....4992
59.4.1.4 Validity Flag Reception....4994
59.4.1.5 SPDIF Receiver Interrupt Exception Definition....4995
59.4.1.6 Standards Compliance....4995
59.4.1.7 SPDIF PLOCK Detection and Rxclk Output....4996
59.4.1.8 Measuring Frequency of SPDIF_RxClk....4996

59.4.2 SPDIF Transmitter ... 4997
59.4.2.1 Audio Data Transmission ... 4997
59.4.2.2 Channel Status Transmission ... 4998
59.4.2.3 Validity Flag Transmission ... 4998
59.5 SPDIF Memory Map/Register Definition ... 4998
59.5.1 SPDIF Configuration Register (SPDIF_SCR) ... 5000
59.5.2 CDText Control Register (SPDIF_SRCD) ... 5002
59.5.3 PhaseConfig Register (SPDIF_SRPC) ... 5003
59.5.4 InterruptEn Register (SPDIF_SIE) ... 5004
59.5.5 InterruptStat Register (SPDIF_SIS) ... 5006
59.5.6 InterruptClear Register (SPDIF_SIC) ... 5008
59.5.7 SPDIFRxLeft Register (SPDIF_SRL) ... 5009
59.5.8 SPDIFRxRight Register (SPDIF_SRR) ... 5010
59.5.9 SPDIFRxCChannel_h Register (SPDIF_SRCSH) ... 5010
59.5.10 SPDIFRxCChannel_l Register (SPDIF_SRCSL) ... 5011
59.5.11 UchannelRx Register (SPDIF_SRU) ... 5011
59.5.12 QchannelRx Register (SPDIF_SRQ) ... 5012
59.5.13 SPDIFTxLeft Register (SPDIF_STL) ... 5012
59.5.14 SPDIFTxRight Register (SPDIF_STR) ... 5013
59.5.15 SPDIFTxCChannelCons_h Register (SPDIF_STCSCH) ... 5013
59.5.16 SPDIFTxCChannelCons_l Register (SPDIF_STCSCL) ... 5014
59.5.17 FreqMeas Register (SPDIF_SRFM) ... 5014
59.5.18 SPDIFTxClk Register (SPDIF_STC) ... 5015

Chapter 60
System Reset Controller (SRC)

60.1 SRC Overview ... 5017
60.1.1 Features ... 5017
60.2 External Signals ... 5017
60.3 Clocks ... 5019

60.4 Top-level resets, power-up sequence and external supply integration..... 5019
60.4.1 Reset and Power-up Flow..... 5019
60.4.2 Finite-State Machine (FSM)..... 5022
60.4.3 Power mode transitions..... 5023
60.5 Power-On Reset and power sequencing..... 5024
60.5.1 External POR using SRC_POR_B..... 5024
60.5.2 Internal POR..... 5024
60.6 Functional Description..... 5025
60.6.1 Reset Control..... 5025
60.6.1.1 Reset inputs and outputs..... 5025
60.6.1.2 Reset Handling..... 5027
60.6.1.2.1 Reset Qualification..... 5027
60.6.1.2.2 Reset Sequence and De-Assertion..... 5028
60.6.1.2.3 POR (SRC_POR_B)..... 5028
60.6.1.2.4 COLD RESET..... 5029
60.6.1.2.5 WARM RESET..... 5030
60.6.2 Parallel Reset Requests..... 5031
60.6.3 Boot Mode Control..... 5032
60.6.3.1 BOOT_MODE Pin Latching..... 5032
60.7 SRC Memory Map/Register Definition..... 5033
60.7.1 SRC Control Register (SRC_SCR)..... 5035
60.7.2 SRC Boot Mode Register 1 (SRC_SBMR1)..... 5038
60.7.3 SRC Reset Status Register (SRC_SRSR)..... 5039
60.7.4 SRC Interrupt Status Register (SRC_SISR)..... 5041
60.7.5 SRC Interrupt Mask Register (SRC_SIMR)..... 5043
60.7.6 SRC Boot Mode Register 2 (SRC_SBMR2)..... 5045
60.7.7 SRC General Purpose Register 1 (SRC_GPR1)..... 5046
60.7.8 SRC General Purpose Register 2 (SRC_GPR2)..... 5047
60.7.9 SRC General Purpose Register 3 (SRC_GPR3)..... 5047

60.7.10 SRC General Purpose Register 4 (SRC_GPR4)..... 5048
60.7.11 SRC General Purpose Register 5 (SRC_GPR5)..... 5048
60.7.12 SRC General Purpose Register 6 (SRC_GPR6)..... 5049
60.7.13 SRC General Purpose Register 7 (SRC_GPR7)..... 5049
60.7.14 SRC General Purpose Register 8 (SRC_GPR8)..... 5050
60.7.15 SRC General Purpose Register 9 (SRC_GPR9)..... 5050
60.7.16 SRC General Purpose Register 10 (SRC_GPR10)..... 5051

Chapter 61
Synchronous Serial Interface (SSI)

61.1 Overview.....5053
61.1.1 Features..... 5054
61.1.2 Modes of Operation..... 5055
61.2 External Signal Description..... 5055
61.2.1 Signals Overview..... 5055
61.3 Clocks..... 5059
61.4 SSI Transmit FIFO 0 & 1 Registers.....5059
61.5 SSI Transmit Shift Register (TXSR)..... 5060
61.6 SSI Receive FIFO 0 and 1 Registers.....5062
61.7 SSI Receive Shift Register (RXSR).....5063
61.8 Functional Description.....5065
61.8.1 Operating Modes.....5065
61.8.1.1 Normal Mode..... 5067
61.8.1.1.1 Normal Mode Transmit.....5067
61.8.1.1.2 Normal Mode Receive..... 5068
61.8.1.2 Network Mode..... 5070
61.8.1.2.1 Network Mode Transmit.....5071
61.8.1.2.2 Network Mode Receive..... 5072
61.8.1.3 Gated Clock Mode..... 5074
61.8.1.4 I2S Mode.....5077

61.8.1.5 AC97 Mode....5079
61.8.1.5.1 AC97 Fixed Mode (SSI.SACNT[1]=0)....5081
61.8.1.5.2 AC97 Variable Mode (SSI.SACNT[1]=1)....5081
61.8.2 External Frame and Clock Operation....5082
61.8.2.1 Data Alignment Formats Supported....5082
61.8.3 SSI Architecture....5083
61.8.4 SSI Clocking....5084
61.8.4.1 SSI Clock and Frame Sync Generation....5085
61.8.4.2 DIV2, PSR and PM Bit Description....5086
61.8.5 Receive Interrupt Enable Bit Description....5088
61.8.6 Transmit Interrupt Enable Bit Description....5089
61.8.7 Internal Frame and Clock Shutdown....5090
61.8.8 Peripheral Bus Interface....5092
61.8.8.1 Transfer Lengths Supported....5092
61.8.8.2 Transfer Bus Errors....5092
61.8.8.3 Clock Rate....5093
61.8.9 Reset....5093
61.9 SSI Memory Map/Register Definition....5093
61.9.1 SSI Transmit Data Register n (SSI*x*_STX*n*)....5096
61.9.2 SSI Receive Data Register n (SSI*x*_SRX*n*)....5096
61.9.3 SSI Control Register (SSI*x*_SCR)....5097
61.9.4 SSI Interrupt Status Register (SSI*x*_SISR)....5099
61.9.5 SSI Interrupt Enable Register (SSI*x*_SIER)....5105
61.9.6 SSI Transmit Configuration Register (SSI*x*_STCR)....5109
61.9.7 SSI Receive Configuration Register (SSI*x*_SRCR)....5111
61.9.8 SSI Transmit Clock Control Register (SSI*x*_STCCR)....5113
61.9.9 SSI Receive Clock Control Register (SSI*x*_SRCCR)....5115
61.9.10 SSI FIFO Control/Status Register (SSI*x*_SFCSR)....5116
61.9.11 SSI AC97 Control Register (SSI*x*_SACNT)....5120

61.9.12 SSI AC97 Command Address Register (SSI*x*_SACADD)....5121
61.9.13 SSI AC97 Command Data Register (SSI*x*_SACDAT)....5121
61.9.14 SSI AC97 Tag Register (SSI*x*_SATAG)....5122
61.9.15 SSI Transmit Time Slot Mask Register (SSI*x*_STMSK)....5122
61.9.16 SSI Receive Time Slot Mask Register (SSI*x*_SRMSK)....5123
61.9.17 SSI AC97 Channel Status Register (SSI*x*_SACCST)....5123
61.9.18 SSI AC97 Channel Enable Register (SSI*x*_SACCEN)....5124
61.9.19 SSI AC97 Channel Disable Register (SSI*x*_SACCDIS)....5124

Chapter 62
Temperature Monitor (TEMPMON)

62.1 Overview....5125
62.2 Software Usage Guidelines....5126
62.3 TEMPMON Memory Map/Register Definition....5127
62.3.1 Tempsensor Control Register 0 (TEMPMON_TEMPSENSE0*n*)....5128
62.3.2 Tempsensor Control Register 1 (TEMPMON_TEMPSENSE1*n*)....5130

Chapter 63
TrustZone Address Space Controller (TZASC)

63.1 Overview....5131
63.2 Clocks....5132
63.3 i.MX 6Solo/6DualLite Specific Configuration....5132
63.4 Address Mapping in various memory mapping modes....5133

Chapter 64
Universal Asynchronous Receiver/Transmitter (UART)

64.1 Overview....5135
64.1.1 Features....5136
64.1.2 Modes of operation....5137
64.2 External Signals....5137
64.2.1 Detailed Signal Descriptions....5142
64.2.1.1 Interrupt Signals....5142
64.2.1.1.1 interrupt_uart - UART Interrupt....5142

64.2.1.2 DMA Request Signals....5142
64.2.1.2.1 dma_req_rx - Receiver DMA Request....5142
64.2.1.2.2 dma_req_tx - Transmitter DMA Request....5142
64.2.1.3 Special Signals....5142
64.2.1.3.1 stop_req - Stop Mode....5142
64.2.1.3.2 doze_req - Doze Mode....5143
64.2.1.3.3 debug_req - Debug Mode....5143
64.3 Clocks....5143
64.4 Functional Description....5143
64.4.1 Interrupts and DMA Requests....5143
64.4.2 Clocks....5145
64.4.2.1 Clock requirements....5145
64.4.2.2 Maximum Baud Rate....5145
64.4.2.3 Clocking in Low-Power Modes....5146
64.4.3 General UART Definitions....5146
64.4.3.1 RTS_B - UART Request To Send....5147
64.4.3.2 RTS Edge Triggered Interrupt....5148
64.4.3.3 DTR_B - Data Terminal Ready....5148
64.4.3.4 DSR_B - Data Set Ready....5149
64.4.3.5 DTR_B/DSR_B Edge Triggered Interrupt....5149
64.4.3.6 DCD_B - Data Carrier Detect....5150
64.4.3.7 RI_B - Ring Indicator....5150
64.4.3.8 CTS_B - Clear To Send....5150
64.4.3.9 Programmable CTS_B Deassertion....5150
64.4.3.10 TX_DATA - UART Transmit....5151
64.4.3.11 RX_DATA - UART Receive....5151
64.4.4 Transmitter....5153
64.4.4.1 Transmitter FIFO Empty Interrupt Suppression....5153
64.4.4.2 Transmitting a Break Condition....5155

64.4.5 Receiver ... 5155
64.4.5.1 Idle Line Detect ... 5156
64.4.5.2 Aging Character Detect ... 5157
64.4.5.3 Receiver Wake ... 5158
64.4.5.4 Receiving a BREAK Condition ... 5159
64.4.5.5 Vote Logic ... 5159
64.4.5.6 Baud Rate Automatic Detection Logic ... 5161
64.4.5.6.1 Baud Rate Automatic Detection Protocol ... 5162
64.4.5.6.2 New Baud Rate Determination ... 5162
64.4.5.6.2.1 New Autobaud Counter Stopped bit and Interrupt ... 5163
64.4.6 Escape Sequence Detection ... 5163
64.5 Binary Rate Multiplier (BRM) ... 5165
64.6 Infrared Interface ... 5167
64.6.1 Generalities-Infrared ... 5167
64.6.2 Inverted Transmission and Reception bits (INVT & INVR) ... 5168
64.6.3 InfraRed Special Case (IRSC) Bit ... 5168
64.6.4 IrDA interrupt ... 5169
64.6.5 Conclusion about IrDA ... 5170
64.6.6 Programming IrDA Interface ... 5171
64.6.6.1 High Speed ... 5171
64.6.6.2 Low Speed ... 5171
64.7 9-bit RS-485 Mode ... 5172
64.7.1 Generalities ... 5172
64.7.2 Transmit 9-bit RS-485 frames ... 5173
64.7.3 Receive 9-bit RS-485 frames ... 5173
64.7.3.1 RS-485 Slave Address Normal Detect Mode ... 5173
64.7.3.2 RS-485 Slave Address Automatic Detect Mode ... 5174
64.8 Low Power Modes ... 5174
64.8.1 UART Operation in System Doze Mode ... 5175

64.8.2 UART Operation in System Stop Mode 5175
64.8.3 Power Saving Method in UART 5176
64.9 UART Operation in System Debug State 5176
64.10 Reset 5177
64.10.1 Hardware reset 5177
64.10.2 Software reset 5177
64.11 Transfer Error 5177
64.12 Functional Timing 5178
64.12.1 IrDA Mode 5178
64.13 Initialization 5178
64.13.1 Programming the UART in RS-232 mode 5178
64.13.2 Programming the UART in 9-bit RS-485 mode 5180
64.14 References 5181
64.15 UART Memory Map/Register Definition 5182
64.15.1 UART Receiver Register (UART*x*_URXD) 5187
64.15.2 UART Transmitter Register (UART*x*_UTXD) 5189
64.15.3 UART Control Register 1 (UART*x*_UCR1) 5190
64.15.4 UART Control Register 2 (UART*x*_UCR2) 5192
64.15.5 UART Control Register 3 (UART*x*_UCR3) 5195
64.15.6 UART Control Register 4 (UART*x*_UCR4) 5197
64.15.7 UART FIFO Control Register (UART*x*_UFCR) 5199
64.15.8 UART Status Register 1 (UART*x*_USR1) 5201
64.15.9 UART Status Register 2 (UART*x*_USR2) 5204
64.15.10 UART Escape Character Register (UART*x*_UESC) 5206
64.15.11 UART Escape Timer Register (UART*x*_UTIM) 5207
64.15.12 UART BRM Incremental Register (UART*x*_UBIR) 5207
64.15.13 UART BRM Modulator Register (UART*x*_UBMR) 5208
64.15.14 UART Baud Rate Count Register (UART*x*_UBRC) 5208
64.15.15 UART One Millisecond Register (UART*x*_ONEMS) 5209

64.15.16 UART Test Register (UART*x*_UTS)....5210
64.15.17 UART RS-485 Mode Control Register (UART*x*_UMCR)....5211

Chapter 65
Universal Serial Bus Controller (USB)

65.1 Overview....5213
65.1.1 Features....5214
65.1.2 Modes of Operation....5216
65.1.2.1 Normal Mode....5216
65.1.2.2 Low-Power Mode....5217
65.2 External Signals....5217
65.3 Functional Description....5218
65.3.1 USB 2.0 Controller Core 0....5218
65.3.1.1 Host Mode....5218
65.3.1.2 Peripheral (Device) Mode....5218
65.3.1.3 Pins Used for OTG Controller....5218
65.3.2 USB 2.0 Controller Core 1....5219
65.3.2.1 Pins Used for Host Controller 1....5219
65.3.3 USB 2.0 Controller Core 2....5219
65.3.4 USB 2.0 Controller Core 3....5220
65.3.5 USB Power Control....5220
65.3.5.1 Entering Low Power Suspend Mode....5220
65.3.5.2 Wake-Up Events....5221
65.3.5.2.1 Host Mode Events....5221
65.3.6 Interrupts....5222
65.3.6.1 USB Core Interrupts....5222
65.3.6.2 USB Wake-Up Interrupts....5222
65.4 USB Operation Model....5222
65.4.1 Register Interface....5223
65.4.1.1 Configuration, Control and Status Register Set....5223

65.4.1.2 Identification Registers 5225
65.4.1.3 OTG Operations 5225
65.4.1.3.1 Register Bits 5225
65.4.2 Host Data Structures 5227
65.4.2.1 Periodic Frame List 5227
65.4.2.2 Asynchronous List Queue Head Pointer 5229
65.4.2.3 Isochronous (High-Speed) Transfer Descriptor (iTD) 5230
65.4.2.3.1 Next Link Pointer 5231
65.4.2.3.2 iTD Transaction Status and Control List 5232
65.4.2.3.3 iTD Buffer Page Pointer List (Plus) 5233
65.4.2.4 Split Transaction Isochronous Transfer Descriptor (siTD) 5235
65.4.2.4.1 Next Link Pointer 5235
65.4.2.4.2 siTD Endpoint Capabilities/Characteristics 5236
65.4.2.4.3 siTD Transfer State 5237
65.4.2.4.4 siTD Buffer Pointer List (plus) 5238
65.4.2.4.5 siTD Back Link Pointer 5239
65.4.2.5 Queue element transfer descriptor (qTD) 5239
65.4.2.5.1 Next qTD Pointer 5240
65.4.2.5.2 Alternate Next qTD Pointer 5241
65.4.2.5.3 qTD Token 5241
65.4.2.5.4 qTD Buffer Page Pointer List 5244
65.4.2.6 Queue Head 5245
65.4.2.6.1 Queue Head Horizontal Link Pointer 5245
65.4.2.6.2 Queue Head Endpoint Capabilities/Characteristics 5246
65.4.2.6.3 Transfer Overlay-Queue Head 5248
65.4.2.7 Periodic Frame Span Traversal Node (FSTN) 5249
65.4.2.7.1 FSTN Normal Path Pointer 5250
65.4.2.7.2 FSTN Back Path Link Pointer 5250
65.4.3 Host Operational Model 5251

65.4.3.1 Host Controller Initialization 5251
65.4.3.2 Port Routing and Control..... 5253
65.4.3.2.1 Port Routing Control through EHCI Configured (CF) Bit 5255
65.4.3.2.2 Port Routing Control through PortOwner and Disconnect Event 5256
65.4.3.2.3 Example Port Routing State Machine 5258
65.4.3.2.3.1 EHCI HC Owner 5258
65.4.3.2.3.2 Companion HC Owner 5259
65.4.3.2.4 Port Power 5259
65.4.3.2.5 Port Reporting Over-Current 5260
65.4.3.3 Suspend/Resume-Host Operational Model 5261
65.4.3.3.1 Port Suspend/Resume 5261
65.4.3.4 Schedule Traversal Rules.....5264
65.4.3.4.1 Example - Preserving Micro-Frame Integrity 5266
65.4.3.4.1.1 Transaction Fit - A Best-Fit Approximation Algorithm 5266
65.4.3.5 Periodic Schedule Frame Boundaries vs Bus Frame Boundaries 5268
65.4.3.6 Periodic Schedule 5271
65.4.3.7 Managing Isochronous Transfers Using iTDs 5273
65.4.3.7.1 Host Controller Operational Model for iTDs 5273
65.4.3.7.2 Software Operational Model for iTDs 5275
65.4.3.7.2.1 Periodic scheduling threshold.....5277
65.4.3.8 Asynchronous Schedule 5278
65.4.3.8.1 Adding Queue Heads to Asynchronous Schedule..... 5279
65.4.3.8.2 Removing Queue Heads from Asynchronous Schedule 5280
65.4.3.8.3 Empty Asynchronous Schedule Detection 5282
65.4.3.8.4 Restarting Asynchronous Schedule Before EOF 5283
65.4.3.8.4.1 Example Method for Restarting Asynchronous Schedule Traversal 5284
65.4.3.8.4.2 Async Sched Not Active 5285
65.4.3.8.4.3 Async Sched Active 5285

65.4.3.8.4.4 Async Sched Sleeping 5286
65.4.3.8.4.5 Example Derivation for AsyncSchedSleepTime..... 5286
65.4.3.8.5 Asynchronous schedule traversal: Start Event..... 5286
65.4.3.8.6 Reclamation Status Bit (USBSTS Register)..... 5287
65.4.3.9 Operational Model for Nak Counter..... 5287
65.4.3.9.1 Nak Count Reload Control 5289
65.4.3.9.1.1 Wait for List Head 5290
65.4.3.9.1.2 Do Reload 5290
65.4.3.9.1.3 Wait for Start Event 5290
65.4.3.10 Managing Control/Bulk/Interrupt Transfers through Queue Heads..... 5291
65.4.3.10.1 Fetch Queue Head 5293
65.4.3.10.2 Advance Queue 5293
65.4.3.10.3 Execute Transaction 5294
65.4.3.10.3.1 Interrupt Transfer Pre-condition Criteria 5295
65.4.3.10.3.2 Asynchronous Transfer Pre-operations and Pre-condition Criteria 5295
65.4.3.10.3.3 Transfer Type Independent Pre-operations..... 5295
65.4.3.10.3.4 Halting a Queue Head 5298
65.4.3.10.3.5 Asynchronous Schedule Park Mode 5299
65.4.3.10.4 Write Back qTD 5301
65.4.3.10.5 Follow Queue Head Horizontal Pointer 5301
65.4.3.10.6 Buffer Pointer List Use for Data Streaming with qTDs 5302
65.4.3.10.7 Adding Interrupt Queue Heads to the Periodic Schedule 5304
65.4.3.10.8 Managing Transfer Complete Interrupts from Queue Heads 5304
65.4.3.11 Ping Control..... 5305
65.4.3.12 Split Transactions..... 5306
65.4.3.12.1 Split Transactions for Asynchronous Transfers 5307
65.4.3.12.1.1 Asynchronous - Do Start Split..... 5308
65.4.3.12.1.2 Asynchronous - Do Complete Split 5308

65.4.3.12.2 Split Transaction Interrupt 5309
65.4.3.12.2.1 Split Transaction Scheduling Mechanisms for Interrupt 5310
65.4.3.12.2.2 Host Controller Operational Model for FSTNs.....5313
65.4.3.12.2.3 Software Operational Model for FSTNs..... 5316
65.4.3.12.2.4 Tracking Split Transaction Progress for Interrupt Transfers .5317
65.4.3.12.2.5 Split Transaction Execution State Machine for Interrupt.....5318
65.4.3.12.2.6 Rebalancing the periodic schedule 5324
65.4.3.12.3 Split Transaction Isochronous 5325
65.4.3.12.3.1 Split Transaction Scheduling Mechanisms for Isochronous . 5325
65.4.3.12.3.2 Tracking Split Transaction Progress for Isochronous Transfers.....5330
65.4.3.12.3.3 Split Transaction Execution State Machine for Isochronous 5332
65.4.3.12.3.4 Periodic Isochronous - Do Start Split 5333
65.4.3.12.3.5 Periodic Isochronous - Do Complete Split 5335
65.4.3.12.3.6 Complete-Split for Scheduling Boundary Cases 2a, 2b 5338
65.4.3.12.3.7 Split Transaction for Isochronous - Processing Examples..... 5340
65.4.3.13 Host Controller Pause..... 5342
65.4.3.14 Port Test Modes -Host Operational Model..... 5343
65.4.3.15 Interrupts-Host Operational Model..... 5343
65.4.3.15.1 Transfer/Transaction Based Interrupts 5345
65.4.3.15.1.1 Transaction Error 5345
65.4.3.15.1.2 Serial Bus Babble..... 5345
65.4.3.15.1.3 Data Buffer Error 5346
65.4.3.15.1.4 USB Interrupt (Interrupt on Completion (IOC)) 5347
65.4.3.15.1.5 Short Packet.....5347
65.4.3.15.2 Host Controller Event Interrupts 5347
65.4.3.15.2.1 Port Change Events 5348
65.4.3.15.2.2 Frame List Rollover 5348
65.4.3.15.2.3 Interrupt on Async Advance 5348

65.4.3.15.2.4 Host System Error 5348
65.4.4 EHCI Deviation.....5349
65.4.4.1 Embedded Transaction Translator Function.....5350
65.4.4.1.1 Capability Registers.....5350
65.4.4.1.2 Operational Registers.....5351
65.4.4.1.3 Discovery-EHCI Deviation.....5351
65.4.4.1.4 Data Structures.....5351
65.4.4.1.5 Operational Model.....5352
65.4.4.1.5.1 Micro- frame Pipeline.....5352
65.4.4.1.5.2 Split State Machines.....5353
65.4.4.1.5.3 Asynchronous Transaction Scheduling and Buffer Management.....5354
65.4.4.1.5.4 Periodic Transaction Scheduling and Buffer Management....5354
65.4.4.1.5.5 Multiple Transaction Translators.....5355
65.4.4.2 Device Operation.....5355
65.4.4.2.1 USB_USBMODE Register.....5355
65.4.4.2.2 Non-Zero Fields the Register File.....5355
65.4.4.2.3 SOF Interrupt.....5356
65.4.4.3 Embedded Design Interface.....5356
65.4.4.3.1 Frame Adjust Register.....5356
65.4.4.4 Miscellaneous variations from EHCI.....5356
65.4.4.4.1 Programmable Physical Interface Behaviour.....5356
65.4.4.4.2 Discovery.....5356
65.4.4.4.2.1 Port Reset.....5356
65.4.4.4.2.2 Port Speed Detection.....5357
65.4.4.4.3 Port Test Mode.....5357
65.4.5 Device Data Structures.....5357
65.4.5.1 Endpoint Queue Head (dQH).....5359
65.4.5.1.1 Endpoint Capabilities/Characteristics.....5360

65.4.5.1.2 Transfer Overlay-Endpoint Queue Head 5360
65.4.5.1.3 Current dTD Pointer 5361
65.4.5.1.4 Set-up Buffer 5361
65.4.5.2 Endpoint Transfer Descriptor (dTD) 5361
65.4.6 Device Operational Model 5363
65.4.6.1 Device Controller Initialization 5364
65.4.6.2 Port State and Control 5365
65.4.6.2.1 Bus Reset 5367
65.4.6.2.2 Suspend/Resume 5368
65.4.6.2.2.1 Suspend 5368
65.4.6.2.2.2 Resume 5369
65.4.6.3 Managing Endpoints 5369
65.4.6.3.1 Endpoint Initialization 5370
65.4.6.3.2 Stalling 5371
65.4.6.3.3 Data Toggle 5372
65.4.6.3.3.1 Data Toggle Reset 5372
65.4.6.3.3.2 Data Toggle Inhibit 5372
65.4.6.3.3.3 Priming Transmit Endpoints 5372
65.4.6.3.3.4 Priming Receive Endpoints 5373
65.4.6.4 Operational Model For Packet Transfers 5373
65.4.6.4.1 Interrupt/Bulk Endpoint Operational Model 5374
65.4.6.4.1.1 Interrupt/Bulk Endpoint Bus Response Matrix 5376
65.4.6.4.2 Control Endpoint Operation Model 5376
65.4.6.4.2.1 Setup Phase 5376
65.4.6.4.2.2 Data Phase 5377
65.4.6.4.2.3 Status Phase 5378
65.4.6.4.2.4 Control Endpoint Bus Response Matrix 5378
65.4.6.4.3 Isochronous Endpoint Operational Model 5379
65.4.6.4.3.1 Isochronous Pipe Synchronization 5381

65.4.6.4.3.2 Isochronous Endpoint Bus Response Matrix 5381
65.4.6.5 Managing Queue Heads 5381
65.4.6.5.1 Queue Head Initialization 5382
65.4.6.5.2 Operational Model For Setup Transfers 5383
65.4.6.6 Managing Transfers with Transfer Descriptors 5384
65.4.6.6.1 Software Link Pointers 5384
65.4.6.6.2 Building a Transfer Descriptor 5384
65.4.6.6.3 Executing A Transfer Descriptor 5385
65.4.6.6.4 Transfer Completion 5386
65.4.6.6.5 Flushing/De-priming an Endpoint 5386
65.4.6.6.6 Device Error Matrix 5387
65.4.6.7 Servicing Interrupts 5388
65.4.6.7.1 High-Frequency Interrupts 5388
65.4.6.7.2 Low-Frequency Interrupts 5388
65.4.6.7.3 Error Interrupts 5388
65.5 USB Non-Core Memory Map/Register Definition 5389
65.5.1 USB OTG Control Register (USBNC_USB_OTG_CTRL) 5391
65.5.2 USB Host1 Control Register (USBNC_USB_UH1_CTRL) 5394
65.5.3 USB Host2 Control Register (USBNC_USB_UH2_CTRL) 5397
65.5.4 USB Host3 Control Register (USBNC_USB_UH3_CTRL) 5399
65.5.5 USB Host2 HSIC Control Register (USBNC_USB_UH2_HSIC_CTRL) 5401
65.5.6 USB Host3 HSIC Control Register (USBNC_USB_UH3_HSIC_CTRL) 5403
65.5.7 OTG UTMI PHY Control 0 Register (USBNC_USB_OTG_PHY_CTRL_0) 5404
65.5.8 Host1 UTMI PHY Control 0 Register (USBNC_USB_UH1_PHY_CTRL_0) 5405
65.6 USB Core Memory Map/Register Definition 5406
65.6.1 Identification register (USB_*n*ID) 5411
65.6.2 Hardware General (USB_*n*HWGENERAL) 5412
65.6.3 Host Hardware Parameters (USB_*n*HWHOST) 5413
65.6.4 Device Hardware Parameters (USB_*n*HWDEVICE) 5414

65.6.5 TX Buffer Hardware Parameters (USB_*n*HWTXBUF)..... 5414
65.6.6 RX Buffer Hardware Parameters (USB_*n*HWRXBUF).....5415
65.6.7 General Purpose Timer #0 Load (USB_*n*GPTIMER0LD).....5416
65.6.8 General Purpose Timer #0 Controller (USB_*n*GPTIMER0CTRL).....5416
65.6.9 General Purpose Timer #1 Load (USB_*n*GPTIMER1LD).....5418
65.6.10 General Purpose Timer #1 Controller (USB_*n*GPTIMER1CTRL).....5418
65.6.11 System Bus Config (USB_*n*SBUSCFG)..... 5419
65.6.12 Capability Registers Length (USB_*n*CAPLENGTH).....5420
65.6.13 Host Controller Interface Version (USB_*n*HCIVERSION)..... 5421
65.6.14 Host Controller Structural Parameters (USB_*n*HCSPARAMS)..... 5421
65.6.15 Host Controller Capability Parameters (USB_*n*HCCPARAMS)..... 5423
65.6.16 Device Controller Interface Version (USB_*n*DCIVERSION)..... 5425
65.6.17 Device Controller Capability Parameters (USB_*n*DCCPARAMS)..... 5425
65.6.18 USB Command Register (USB_*n*USBCMD).....5427
65.6.19 USB Status Register (USB_*n*USBSTS).....5431
65.6.20 Interrupt Enable Register (USB_*n*USBINTR).....5435
65.6.21 USB Frame Index (USB_*n*FRINDEX).....5437
65.6.22 Frame List Base Address (USB_*n*PERIODICLISTBASE).....5438
65.6.23 Device Address (USB_*n*DEVICEADDR).....5438
65.6.24 Next Asynch. Address (USB_*n*ASYNCLISTADDR).....5439
65.6.25 Endpoint List Address (USB_*n*ENDPTLISTADDR).....5440
65.6.26 Programmable Burst Size (USB_*n*BURSTSIZE).....5440
65.6.27 TX FIFO Fill Tuning (USB_*n*TXFILLTUNING)..... 5441
65.6.28 Endpoint NAK (USB_*n*ENDPTNAK)..... 5443
65.6.29 Endpoint NAK Enable (USB_*n*ENDPTNAKEN).....5443
65.6.30 Configure Flag Register (USB_*n*CONFIGFLAG)..... 5444
65.6.31 Port Status & Control (USB_*n*PORTSC1)..... 5444
65.6.32 On-The-Go Status & control (USB_*n*OTGSC)..... 5451
65.6.33 USB Device Mode (USB_*n*USBMODE)..... 5455

65.6.34 Endpoint Setup Status (USB_*n*ENDPTSETUPSTAT)....5456
65.6.35 Endpoint Prime (USB_*n*ENDPTPRIME)....5457
65.6.36 Endpoint Flush (USB_*n*ENDPTFLUSH)....5458
65.6.37 Endpoint Status (USB_*n*ENDPTSTAT)....5458
65.6.38 Endpoint Complete (USB_*n*ENDPTCOMPLETE)....5459
65.6.39 Endpoint Control0 (USB_*n*ENDPTCTRL0)....5460
65.6.40 Endpoint Control 1 (USB_*n*ENDPTCTRL1)....5462
65.6.41 Endpoint Control 2 (USB_*n*ENDPTCTRL2)....5465
65.6.42 Endpoint Control 3 (USB_*n*ENDPTCTRL3)....5467
65.6.43 Endpoint Control 4 (USB_*n*ENDPTCTRL4)....5470
65.6.44 Endpoint Control 5 (USB_*n*ENDPTCTRL5)....5473
65.6.45 Endpoint Control 6 (USB_*n*ENDPTCTRL6)....5476
65.6.46 Endpoint Control 7 (USB_*n*ENDPTCTRL7)....5479

Chapter 66
Universal Serial Bus 2.0 Integrated PHY (USB-PHY)

66.1 USB PHY Overview....5483
66.2 Operation....5484
66.2.1 UTMI....5485
66.2.2 Digital Transmitter....5485
66.2.3 Digital Receiver....5485
66.2.4 Analog Receiver....5485
66.2.4.1 HS Differential Receiver....5486
66.2.4.2 Squelch Detector....5487
66.2.4.3 LS/FS Differential Receiver....5487
66.2.4.4 HS Disconnect Detector....5487
66.2.4.5 USB Plugged-In Detector....5487
66.2.4.6 Single-Ended USB_DP Receiver....5488
66.2.4.7 Single-Ended USB_DN Receiver....5488
66.2.4.8 9X Oversample Module....5488

66.2.5 Analog Transmitter ... 5488
66.2.5.1 Switchable High-Speed 45Ω Termination Resistors ... 5488
66.2.5.2 Low-Speed/Full-Speed Differential Driver ... 5489
66.2.5.3 High-Speed Differential Driver ... 5489
66.2.5.4 Switchable 1.5KΩ USB_DP Pullup Resistor ... 5489
66.2.5.5 Switchable 15KΩ USB_DP Pulldown Resistor ... 5489
66.2.6 Recommended Register Configuration for USB Certification ... 5491
66.2.7 Charger detection ... 5491
66.2.7.1 Charger detect control table ... 5491
66.2.7.2 Data pin contact detector ... 5491
66.2.7.3 Charger detector ... 5492
66.2.7.4 Charger detection software flow ... 5492
66.2.7.5 Dead Battery Protect ... 5494
66.3 USB PHY Memory Map/Register Definition ... 5494
66.3.1 USB PHY Power-Down Register (USBPHY*x*_PWD*n*) ... 5497
66.3.2 USB PHY Transmitter Control Register (USBPHY*x*_TX*n*) ... 5499
66.3.3 USB PHY Receiver Control Register (USBPHY*x*_RX*n*) ... 5500
66.3.4 USB PHY General Control Register (USBPHY*x*_CTRL*n*) ... 5502
66.3.5 USB PHY Status Register (USBPHY*x*_STATUS) ... 5505
66.3.6 USB PHY Debug Register (USBPHY*x*_DEBUG*n*) ... 5507
66.3.7 UTMI Debug Status Register 0 (USBPHY*x*_DEBUG0_STATUS) ... 5509
66.3.8 UTMI Debug Status Register 1 (USBPHY*x*_DEBUG1*n*) ... 5510
66.3.9 UTMI RTL Version (USBPHY*x*_VERSION) ... 5511
66.4 USB Analog Memory Map/Register Definition ... 5511
66.4.1 USB VBUS Detect Register (USB_ANALOG_USB1_VBUS_DETECT*n*) ... 5513
66.4.2 USB Charger Detect Register (USB_ANALOG_USB1_CHRG_DETECT*n*) ... 5514
66.4.3 USB VBUS Detect Status Register (USB_ANALOG_USB1_VBUS_DETECT_STAT) ... 5516
66.4.4 USB Charger Detect Status Register (USB_ANALOG_USB1_CHRG_DETECT_STAT) ... 5518
66.4.5 USB Misc Register (USB_ANALOG_USB1_MISC*n*) ... 5519

66.4.6 USB VBUS Detect Register (USB_ANALOG_USB2_VBUS_DETECT*n*)..... 5520
66.4.7 USB Charger Detect Register (USB_ANALOG_USB2_CHRG_DETECT*n*).....5522
66.4.8 USB VBUS Detect Status Register (USB_ANALOG_USB2_VBUS_DETECT_STAT).....5524
66.4.9 USB Charger Detect Status Register (USB_ANALOG_USB2_CHRG_DETECT_STAT)..... 5526
66.4.10 USB Misc Register (USB_ANALOG_USB2_MISC*n*).....5527
66.4.11 Chip Silicon Version (USB_ANALOG_DIGPROG).....5528

Chapter 67
Ultra Secured Digital Host Controller (uSDHC)

67.1 Overview.....5529
67.1.1 Features..... 5532
67.1.2 Modes and Operations..... 5533
67.1.2.1 Data Transfer Modes.....5533
67.2 External Signals..... 5533
67.2.1 Signals Overview..... 5536
67.3 Clocks..... 5537
67.4 Functional Description.....5537
67.4.1 Data Buffer.....5537
67.4.1.1 Write Operation Sequence..... 5540
67.4.1.2 Read Operation Sequence..... 5540
67.4.1.3 Data Buffer and Block Size..... 5541
67.4.1.4 Dividing Large Data Transfer..... 5542
67.4.1.5 External DMA Request.....5543
67.4.2 DMA AHB Interface.....5544
67.4.2.1 Internal DMA Request..... 5545
67.4.2.2 DMA Burst Length..... 5546
67.4.2.3 AHB Master Interface.....5546
67.4.2.4 ADMA Engine..... 5546
67.4.2.4.1 ADMA Concept and Descriptor Format.....5547
67.4.2.4.2 ADMA Interrupt..... 5551

67.4.2.4.3 ADMA Error....5552
67.4.3 Register Bank with IP Bus Interface....5552
67.4.3.1 SD Protocol Unit....5553
67.4.3.2 SD control misc....5554
67.4.3.3 SD Clock control....5554
67.4.3.4 Command control....5554
67.4.3.5 Data control....5555
67.4.4 Clock & Reset Manager....5555
67.4.5 Clock Generator....5556
67.4.6 SDIO Card Interrupt....5556
67.4.6.1 Interrupts in 1-bit Mode....5556
67.4.6.2 Interrupt in 4-bit Mode....5556
67.4.6.3 Card Interrupt Handling....5557
67.4.7 Card Insertion and Removal Detection....5558
67.4.8 Power Management and Wake Up Events....5559
67.4.8.1 Setting Wake Up Events....5560
67.4.9 MMC fast boot....5560
67.4.9.1 Boot operation....5560
67.4.9.2 Alternative boot operation....5561
67.5 Initialization/Application of uSDHC....5562
67.5.1 Command Send & Response Receive Basic Operation....5562
67.5.2 Card Identification Mode....5563
67.5.2.1 Card Detect....5563
67.5.2.2 Reset....5564
67.5.2.3 Voltage Validation....5565
67.5.2.4 Card Registry....5567
67.5.3 Card Access....5568
67.5.3.1 Block Write....5568
67.5.3.1.1 Normal Write....5568

67.5.3.1.2 DDR Write....5570
67.5.3.1.3 Write with Pause....5570
67.5.3.2 Block Read....5572
67.5.3.2.1 Normal Read....5572
67.5.3.2.2 DDR Read....5573
67.5.3.2.3 Read with Pause....5573
67.5.3.2.4 DLL (Delay Line) in Read Path....5574
67.5.3.3 Suspend Resume....5576
67.5.3.3.1 Suspend....5576
67.5.3.3.2 Resume....5577
67.5.3.4 ADMA Usage....5577
67.5.3.5 Transfer Error....5578
67.5.3.5.1 CRC Error....5578
67.5.3.5.2 Internal DMA Error....5578
67.5.3.5.3 Transfer ADMA Error....5579
67.5.3.5.4 Auto CMD12 Error....5579
67.5.3.6 Card Interrupt....5580
67.5.4 Switch Function....5580
67.5.4.1 Query, Enable and Disable SDIO High Speed Mode....5581
67.5.4.2 Query, Enable and Disable SD High Speed Mode/SDR50/SDR104/DDR50....5581
67.5.4.3 Query, Enable and Disable MMC High Speed Mode....5582
67.5.4.4 Set MMC Bus Width....5582
67.5.5 ADMA Operation....5582
67.5.5.1 ADMA1 Operation....5583
67.5.5.2 ADMA2 Operation....5583
67.5.6 Fast Boot Operation....5584
67.5.6.1 Normal fast boot flow....5584
67.5.6.2 Alternative fast boot flow....5585
67.5.6.3 Fast boot application case (in DMA mode)....5585

67.6 Commands for MMC/SD/SDIO 5587
67.7 Software Restrictions 5592
67.7.1 Initialization Active 5592
67.7.2 Software Polling Procedure 5593
67.7.3 Suspend Operation 5593
67.7.4 Data Length Setting 5593
67.7.5 (A)DMA Address Setting 5593
67.7.6 Data Port Access 5594
67.7.7 Change Clock Frequency 5594
67.7.8 Multi-block Read 5594
67.8 uSDHC Memory Map/Register Definition 5594
67.8.1 DMA System Address (uSDHC*x*_DS_ADDR) 5600
67.8.2 Block Attributes (uSDHC*x*_BLK_ATT) 5600
67.8.3 Command Argument (uSDHC*x*_CMD_ARG) 5602
67.8.4 Command Transfer Type (uSDHC*x*_CMD_XFR_TYP) 5602
67.8.5 Command Response0 (uSDHC*x*_CMD_RSP0) 5606
67.8.6 Command Response1 (uSDHC*x*_CMD_RSP1) 5606
67.8.7 Command Response2 (uSDHC*x*_CMD_RSP2) 5607
67.8.8 Command Response3 (uSDHC*x*_CMD_RSP3) 5607
67.8.9 Data Buffer Access Port (uSDHC*x*_DATA_BUFF_ACC_PORT) 5609
67.8.10 Present State (uSDHC*x*_PRES_STATE) 5609
67.8.11 Protocol Control (uSDHC*x*_PROT_CTRL) 5615
67.8.12 System Control (uSDHC*x*_SYS_CTRL) 5620
67.8.13 Interrupt Status (uSDHC*x*_INT_STATUS) 5623
67.8.14 Interrupt Status Enable (uSDHC*x*_INT_STATUS_EN) 5629
67.8.15 Interrupt Signal Enable (uSDHC*x*_INT_SIGNAL_EN) 5632
67.8.16 Auto CMD12 Error Status (uSDHC*x*_AUTOCMD12_ERR_STATUS) 5634
67.8.17 Host Controller Capabilities (uSDHC*x*_HOST_CTRL_CAP) 5638
67.8.18 Watermark Level (uSDHC*x*_WTMK_LVL) 5640

67.8.19 Mixer Control (uSDHCx_MIX_CTRL)....5641
67.8.20 Force Event (uSDHCx_FORCE_EVENT)....5643
67.8.21 ADMA Error Status Register (uSDHCx_ADMA_ERR_STATUS)....5646
67.8.22 ADMA System Address (uSDHCx_ADMA_SYS_ADDR)....5648
67.8.23 DLL (Delay Line) Control (uSDHCx_DLL_CTRL)....5649
67.8.24 DLL Status (uSDHCx_DLL_STATUS)....5651
67.8.25 CLK Tuning Control and Status (uSDHCx_CLK_TUNE_CTRL_STATUS)....5652
67.8.26 Vendor Specific Register (uSDHCx_VEND_SPEC)....5654
67.8.27 MMC Boot Register (uSDHCx_MMC_BOOT)....5657
67.8.28 Vendor Specific 2 Register (uSDHCx_VEND_SPEC2)....5658

Chapter 68
Video Data Order Adapter (VDOA)

68.1 Overview....5661
68.1.1 Block Diagram....5661
68.1.2 Features....5662
68.2 Clocks....5662
68.3 Functional Description....5663
68.3.1 Memory organization....5663
68.3.1.1 Frame organization - VPU....5663
68.3.1.2 Frame organization - IPU....5664
68.3.2 Conversion types....5664
68.3.3 IPU synchronization....5664
68.3.4 Double buffering....5664
68.4 VDOA Memory Map/Register Definition....5665
68.4.1 VDOA Control Register (VDOA_VDOAC)....5666
68.4.2 VDOA Start and Reset (VDOA_VDOASRR)....5667
68.4.3 VDOA Interrupt Enable Register (VDOA_VDOAIE)....5668
68.4.4 VDOA Interrupt Status Register (VDOA_VDOAIST)....5668
68.4.5 VDOA Frame Parameters Register (VDOA_VDOAFP)....5669

68.4.6 VDOA IPU External Buffer 0 Frame 0 Address Register (VDOA_VDOAIEBA00)..... 5669
68.4.7 VDOA IPU External Buffer 0 Frame 1 Address Register (VDOA_VDOAIEBA01)..... 5670
68.4.8 VDOA IPU External Buffer 0 Frame 2 Address Register (VDOA_VDOAIEBA02)..... 5670
68.4.9 VDOA IPU External Buffer 1 Frame 0 Address Register (VDOA_VDOAIEBA10)..... 5670
68.4.10 VDOA IPU External Buffer 1 Frame 1 Address Register (VDOA_VDOAIEBA11)..... 5671
68.4.11 VDOA IPU External Buffer 1 Frame 2 Address Register (VDOA_VDOAIEBA12)..... 5671
68.4.12 VDOA IPU Stride Line Register (VDOA_VDOASL)..... 5672
68.4.13 VDOA IPU U (Chroma) Buffer Offset Register (VDOA_VDOAIUBO)..... 5672
68.4.14 VDOA VPU External Buffer 0 Address Register (VDOA_VDOAVEBA0)..... 5673
68.4.15 VDOA VPU External Buffer 1 Address Register (VDOA_VDOAVEBA1)..... 5673
68.4.16 VDOA VPU External Buffer 2 Address Register (VDOA_VDOAVEBA2)..... 5673
68.4.17 VDOA VPU U (Chroma) Buffer Offset Register (VDOA_VDOAVUBO)..... 5674
68.4.18 VDOA Status Register (VDOA_VDOASR)..... 5675

Chapter 69
Video Processing Unit (VPU)

69.1 Overview..... 5677
69.1.1 Features..... 5678
69.2 Modes of Operation..... 5680
69.2.1 Normal Operating Mode..... 5680
69.2.2 Low Power Mode..... 5680
69.3 Clocks..... 5680
69.4 Functional Description..... 5681
69.4.1 VPU Architecture..... 5681
69.4.1.1 Embedded BIT processor..... 5682
69.4.1.2 Video CODEC Hardware..... 5682
69.4.1.2.1 Inter-Predictor..... 5682
69.4.1.2.2 AC/DC and Intra-Predictor..... 5683
69.4.1.2.3 Inverse transform/Inverse quantization..... 5683
69.4.1.2.4 De-blocking/Overlap-smoothing filter..... 5683

69.4.1.2.5 Coefficient buffer interface....5683
69.4.1.2.6 Macroblock controller....5683
69.4.1.2.7 Rotation/Mirroring....5684
69.4.2 Reset....5685
69.4.3 Interrupts....5686
69.4.4 Endianness....5686
69.5 Initialization Information....5687
69.6 Application Information....5688
69.6.1 Video Decoding Processing Control....5688
69.6.1.1 Video Decoding Process Flow....5688
69.6.1.2 Video Decoding Process Command....5690
69.6.1.3 Video Decoding Process Finish Detection....5691
69.6.1.4 Video Decoding Process Flow Example....5692
69.6.2 Video Encoding Processing Control....5694
69.6.2.1 The Pipeline for Encoding....5694
69.6.3 Video Codec Processing Buffer Requirement....5698
69.6.3.1 Memory Map Types of Frame Buffer....5699
69.6.3.2 Frame Buffer....5700
69.6.3.3 BIT Processor Program Buffer....5702
69.6.3.4 Working Buffer....5703
69.6.3.5 Bitstream Buffer....5703
69.6.3.6 Parameter Buffer....5704
69.6.3.7 Search RAM....5704
69.6.3.8 Buffer Requirement Summary....5704
69.7 VPU Memory Map/Register Definition....5704
69.7.1 BIT Processor run start (VPU_CodeRun)....5706
69.7.2 BIT Boot Code Download Data register (VPU_CodeDown)....5706
69.7.3 Host Interrupt Request to BIT (VPU_HostIntReq)....5707
69.7.4 BIT Interrupt Clear (VPU_BitIntClear)....5708

69.7.5 BIT Interrupt Status (VPU_BitIntSts)....5709
69.7.6 BIT Current PC (VPU_BitCurPc)....5710
69.7.7 BIT CODEC Busy (VPU_BitCodecBusy)....5711

Chapter 70
Watchdog Timer (WDOG)

70.1 Overview....5713
70.1.1 Features....5714
70.2 External signals....5715
70.3 Clocks....5715
70.4 Watchdog mechanism and system integration....5715
70.5 Functional description....5717
70.5.1 Timeout event....5717
70.5.1.1 Servicing WDOG to reload the counter....5718
70.5.2 Interrupt event....5718
70.5.3 Power-down counter event....5718
70.5.4 Low power modes....5719
70.5.4.1 STOP and DOZE mode....5719
70.5.4.2 WAIT mode....5719
70.5.5 Debug mode....5720
70.5.6 Operations....5720
70.5.6.1 Watchdog reset generation....5720
70.5.6.2 WDOG_B generation....5720
70.5.7 Reset....5722
70.5.8 Interrupt....5723
70.5.9 Flow Diagrams....5723
70.6 Initialization....5725
70.7 WDOG Memory Map/Register Definition....5726
70.7.1 Watchdog Control Register (WDOG*x*_WCR)....5726
70.7.2 Watchdog Service Register (WDOG*x*_WSR)....5728

70.7.3 Watchdog Reset Status Register (WDOG*x*_WRSR)..... 5729
70.7.4 Watchdog Interrupt Control Register (WDOG*x*_WICR)..... 5730
70.7.5 Watchdog Miscellaneous Control Register (WDOG*x*_WMCR)..... 5731

Chapter 71
Crystal Oscillator (XTALOSC)

71.1 Overview..... 5733
71.2 External Signals..... 5733
71.3 Crystal Oscillator 24 MHz..... 5734
71.3.1 Oscillator Configuration (24 MHz)..... 5734
71.3.2 Crystal Frequency Detection(24 MHz)..... 5735
71.4 Crystal Oscillator 32 kHz..... 5736
71.4.1 Oscillator Configuration (32 kHz)..... 5736
71.4.2 Bypass Configuration (32 kHz)..... 5737
71.5 XTALOSC 24MHz Memory Map/Register Definition..... 5737
71.5.1 Miscellaneous Register 0 (XTALOSC24M_MISC0*n*)..... 5739

Chapter 1
Introduction

1.1 About This Document

The i.MX 6Solo/6DualLite application processors are NXP's latest additions to a growing family of real-time processing products offering high-performance processing optimized for lowest power consumption.

The i.MX 6Solo/6DualLite processors feature NXP's advanced implementation of the ARM®Cortex®-A9 core.

The processors can be interfaced with DDR3, DDR3L LPDDR2 (single and dual channel) DRAM memory devices.

These products are suitable for applications such as:

- Automotive navigation and entertainment
- High-end Mobile Internet Devices and high-end PDAs
- Netbooks
- Nettops
- High-end portable media players with HD video capability
- Portable navigation devices
- Gaming Consoles

1.1.1 Audience

The reference manual is intended for the board-level product designers and product software developers. This manual assumes that the reader has a background in computer engineering and/or software engineering and understands the concepts of the digital system design, microprocessor architecture, input/output (I/O) devices, industry standard communication, and device interface protocols.

1.1.2 Organization

The reference manual describes the chip at a system level and provides an architectural overview. It also describes the system memory map, system-level interrupt events, external pins and pin multiplexing, external memory, system debug, system boot, multimedia subsystem, power management, and system security.

1.1.3 Suggested Reading

This section lists the additional resources that provide background for the information in this manual, as well as general information about the architecture.

1.1.3.1 General Information

The following documentation provides useful background information about the ARM Cortex processor.

For information about the ARM Cortex processor see:

- **http://infocenter.arm.com**

1.1.3.2 Related Documentation

For a current list of documentation, refer to **http://www.nxp.com**.

1.1.4 Conventions

The reference manual uses the following notational conventions:

cleared / set
When a bit has a value of zero, it is said to be cleared; when it has a value of one, it is said to be set.

mnemonics
Instruction mnemonics are shown in lowercase bold.

italics
Italics indicate variable command parameters, for example, **bcctr***x*.

The book titles in the text are set in italics.

15
An integer in decimal.

0x
the prefix to denote a hexadecimal number.

0b
The prefix to denote a binary number. Binary values of 0 and 1 are written without a prefix.

n'H4000CA00
The n-bit hexadecimal number.

BLK_REG_NAME
The register names are all uppercase. The block mnemonic is prepended with an underscore delimiter (_).

BLK_REG[FIELD]
The fields within registers appear in brackets. For example, ESR[RLS] refers to the Receive Last Slot field of the ESAI Status Register.

BLK_REG[*n*]
The bit number *n* within the BLK.REG register.

BLK_REG[*l:r*]
The register bit ranges. The ranges are indicated by the left-most bit number *l* and the right-most bit number *r*, separated by a colon (:). For example, ESR[15:0] refers to the lower half word in the ESAI Status Register.

x, U
In some contexts, such as signal encodings, an unitalicized x indicates a "don't care" or "uninitialized". The binary value can be 1 or 0.

x
An italicized *x* indicates an alphanumeric variable.

n, m
Italicized *n or m* represent integer variables.

!
Binary logic operator NOT.

&&
Binary logic operator AND.

||
Binary logic operator OR.

^ or <O+>
Binary logic operator XOR. For example, A <O+> B.

|
Bit-wise OR. For example, 0b0001 | 0b1000 yields the value of 0b1001.

&
Bit-wise AND. For example, 0b0001 and 0b1000 yields the value of 0b0000.

{A,B}

Concatenation, where the *n*-bit value A is prepended to the *m*-bit value B to form an (*n* +*m*)-bit value. For example, {0, REG*m* [14:0]} yeilds a 16-bit value with 0 in the most significant bit.

- or grey fill

Indicates a reserved bit field in a register. Although these bits can be written to with ones or zeros, they always read zeros.

>>

Shift right logical one position.

<<

Shift left logical one position.

<=

Assignment.

==

Compare equal.

!=

Compare not equal.

>

Greater than.

<

Less than.

1.1.5 Register Access

1.1.5.1 Register Diagram Field Access Type Legend

This figure provides the interpretation of the notation used in the register diagrams for a number of common field access types:

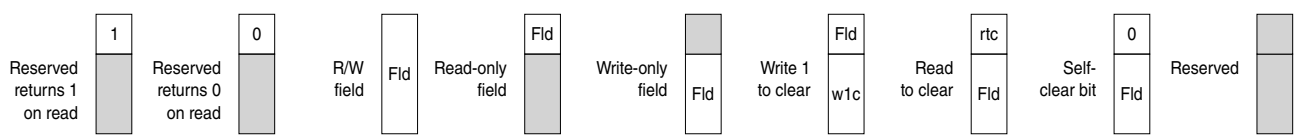


Figure 1-1. Register Field Conventions

NOTE

For reserved register fields, the software should mask off the data in the field after a read (the software can't rely on the contents of data read from a reserved field) and always write all zeros.

1.1.5.2 Register Macro Usage

A common operation is to update one field without disturbing the contents of the remaining fields in the register. Normally, this requires a read-modify-write (RMW) operation, where the CPU reads the register, modifies the target field, then writes the results back to the register. This is an expensive operation in terms of CPU cycles, because of the initial register read.

To address this issue, some hardware registers are implemented as a group, including registers that can be used to either set, clear, or toggle (SCT) individual bits of the primary register. When writing to an SCT register, all the bits set to 1 perform the associated operation on the primary register, while the bits set to 0 are not affected. The SCT registers always read back 0, and should be considered write-only. The SCT registers are not implemented if the primary register is read-only.

With this architecture, it is possible to update one or more fields using only register writes. First, all bits of the target fields are cleared by a write to the associated clear register, then the desired value of the target fields is written to the set register. This sequence of two writes is referred to as a clear-set (CS) operation.

A CS operation does have one potential drawback. Whenever a field is modified, the hardware sees a value of 0 before the final value is written. For most fields, passing through the 0 state is not a problem. Nonetheless, this behavior is something to consider when using a CS operation.

Also, a CS operation is not required for fields that are one-bit wide. While the CS operation works in this case, it is more efficient to simply set or clear the target bit (that is, one write instead of two). A simple set or clear operation is also atomic, while a CS operation is not.

Note that not all macros for set, clear, or toggle (SCT) are atomic. For registers that do not provide hardware support for this functionality, these macros are implemented as a sequence of read-modify-write operations. When an atomic operation is required, the developer should pay attention to this detail, because unexpected behavior might result if an interrupt occurs in the middle of the critical section comprising the update sequence.

A set of SCT registers is offered for registers in many modules on this device, as described in this manual. In a module memory map table, the suffix _SET, _CLR, or _TOG is added to the base name of the register. For example, the CCM_ANALOG_PLL_ARM register has three other registers called CCM_ANALOG_PLL_ARM_SET, CCM_ANALOG_PLL_ARM_CLR, and CCM_ANALOG_PLL_ARM_TOG.

In the sub-section that describes one of these sets of registers, a short-hand convention is used to denote that a register has the SCT register set. There is an italicized *n* appended to the end of the short register name. Using the above example, the name used for this register is CCM_ANALOG_PLL_ARM*n*. When you see this designation, there is a SCT register set associated with the register, and you can verify this by checking it in the memory map table. The address offset for each of these registers is given in the form of the following example:

Address: 20C_8000h base + 0h offset + (4d × i), where i=0d to 3d

In this example, the address for each of the base registers and their three SCT registers can be calculated as:

Register	Address
CCM_ANALOG_PLL_ARM	20C_8000h
CCM_ANALOG_PLL_ARM_SET	20C_8004h
CCM_ANALOG_PLL_ARM_CLR	20c_8008h
CCM_ANALOG_PLL_ARM_TOG	20C_800Ch

1.1.6 Signal Conventions

_b, _B

When appended to a signal name, this indicates that a signal is active-low.

$\overline{\text{NEG_ACTIVE}}$

Overbar also denotes a negative active signal.

UPPERCASE

Package pin names, Block I/O signals.

lowercase

Lowercase is used to indicate internal signals.

1.1.7 Acronyms and Abbreviations

The table below contains acronyms and abbreviations used in this document.

Acronyms and Abbreviated Terms

Term	Meaning
ADC	Analog-to-Digital Converter
AHB	Advanced High-performance Bus
AIPS	Arm IP Bus
ALU	Arithmetic Logic Unit

Table continues on the next page...

Term	Meaning
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
ASRC	Asynchronous Sample Rate Converter
AXI	Advanced eXtensible Interface
BIST	Built-In Self Test
CA/CM	Arm Cortex-A/Cortex-M
CAAM	Cryptographic Acceleration and Assurance Module
CAN	Controller Area Network
CCM	Clock Controller Module
CPU	Central Processing Unit
CSI	CMOS Sensor Interface
CSU	Central Security Unit
CTI	Cross Trigger Interface
DAP	Debug Access Port
DDR	Double data rate
DMA	Direct memory access
DPLL	Digital phase-locked loop
DRAM	Dynamic random access memory
ECC	Error correcting codes
ECSPI	Enhanced Configurable SPI
EDMA	Enhanced Direct Memory Access
EIM	External Interface Module
ENET	Ethernet
EPDC	Electrophoretic Display Controller
EPIT	Enhanced Periodic Interrupt Timer
EPROM	Erasable Programmable Read-Only Memory
ETF	Embedded Trace FIFO
ETM	Embedded Trace Macrocell
FIFO	First-In-First-Out
GIC	General Interrupt Controller
GPC	General Power Controller
GPIO	General-Purpose I/O
GPR	General-Purpose Register
GPS	Global Positioning System
GPT	General-Purpose Timer
GPU	Graphics Processing Unit
GPV	Global Programmers View
HAB	High-Assurance Boot
I2C or I^2C	Inter-Integrated Circuit
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers

Table continues on the next page...

Term	Meaning
IOMUX	Input-Output Multiplexer
IP	Intellectual Property
IrDA	Infrared Data Association
JTAG	Joint Test Action Group (a serial bus protocol usually used for test purposes)
LDO	Low-Dropout
LIFO	Last-In-First-Out
LRU	Least-Recently Used
LSB	Least-Significant Byte
LUT	Look-Up Table
LVDS	Low Voltage Differential Signaling
MAC	Medium Access Control
MCM	Miscellaneous control Module
MMC	Multimedia Card
MMDC	Multi Mode DDR Controller
MSB	Most-Significant Byte
MT/s	Mega Transfers per second
OCRAM	On-Chip Random-Access Memory
OCOTP	On-Chip One-Time Programmable Controller
PCI	Peripheral Component Interconnect
PCIe	PCI express
PCMCIA	Personal Computer Memory Card International Association
PGC	Power Gating Controller
PIC	Programmable Interrupt Controller
PMU	Power Management Unit
POR	Power-On Reset
PSRAM	Pseudo-Static Random Access Memory
PWM	Pulse Width Modulation
PXP	Pixel Pipeline
QoS	Quality of Service
R2D	Radians to Degrees
RISC	Reduced Instruction Set Computing
ROM	Read-Only Memory
ROMCP	ROM Controller with Patch
RTOS	Real-Time Operating System
Rx	Receive
SAI	Synchronous Audio Interface
SCU	Snoop Control Unit
SD	Secure Digital
SDIO	Secure Digital Input/Output
SDLC	Synchronous Data Link Control
SDMA	Smart DMA

Table continues on the next page...

Term	Meaning
SIM	Subscriber Identification Module
SNVS	Secure Non-Volatile Storage
SoC	System-on-Chip
SPBA	Shared Peripheral Bus Arbiter
SPDIF	Sony Phillips Digital Interface
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SRC	System Reset Controller
TFT	Thin-Film Transistor
TPIU	Trace Port Interface
TSGEN	Time Stamp Generator
Tx	Transmit
TZASC	TrustZone Address Space Controller
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
USDHC	Ultra Secured Digital Host Controller
WDOG	Watchdog
WLAN	Wireless Local Area Network
WXGA	Wide Extended Graphics Array

1.2 Introduction

This chapter introduces the architecture of the i.MX 6Solo/6DualLite Multimedia Applications Processors. The i.MX 6Solo/6DualLite processors represent NXP's latest achievement in integrated multimedia applications processors.

It is part of a growing i.MX family of multimedia-focused products, offering high-performance processing optimized for the lowest power consumption.

1.3 Target Applications

The i.MX 6Solo/6DualLite applications processor is tailored for use in multimedia-centric smart mobile devices, driver information systems including infotainment and graphical instrument clusters, and portable medical devices.

The architecture's flexibility enables it to be used in a wide variety of general embedded applications. The i.MX 6Solo/6DualLite processor provides all interfaces necessary to connect peripherals such as WLAN, Bluetooth™, GPS, camera sensors, and multiple displays.

1.4 Features

The i.MX 6Solo/6DualLite processors are based on ARM Cortex-A9 MPCore™ Platform, which has the following features:

- The i.MX 6Solo supports single ARM Cortex-A9 MPCore (with TrustZone)
- The i.MX 6DualLite supports dual ARM Cortex-A9 MPCore (with TrustZone)

The core configuration is symmetric, where each core includes:

- 32 KByte L1 Instruction Cache
- 32 KByte L1 Data Cache
- Private Timer and Watchdog
- Cortex-A9 NEON MPE (Media Processing Engine) Co-processor

The ARM Cortex-A9 MPCore complex includes:

- General Interrupt Controller (GIC) with 128 interrupt support
- Global Timer
- Snoop Control Unit (SCU)
- 512 KB unified I/D L2 cache:
  - Used by one core in i.MX 6Solo
  - Shared by two cores in i.MX 6DualLite
- Two Master AXI (32-bit) bus interfaces output of L2 cache

Target frequency of the core (including Neon and L1 cache) is 800 MHz non-overdrive over the specified temperature range:

- 1 GHz overdrive over the specified temperature range
- 800 MHz non-overdrive over the specified temperature range
- NEON MPE coprocessor
- SIMD Media Processing Architecture
- NEON register file with 32x64-bit general-purpose registers
- NEON Integer execute pipeline (ALU, Shift, MAC)
- NEON dual, single-precision floating point execute pipeline (FADD, FMUL)
- NEON load/store and permute pipeline
- Supports single and double-precision add, subtract, multiply, divide, multiply and accumulate, and square root operations as described in the ARM VFPv3 architecture.
- Provides conversions between 16-bit, 32-bit, and 64-bit floating-point formats and ARM integer word formats

The memory system consists of the following components:
- Level 1 Cache—32 KB Instruction, 32 KB Data cache per core
- Level 2 Cache—Unified instruction and data (256 KB)

On-Chip Memory:
- Boot ROM, including HAB (96 KB)
- Internal multimedia / shared, fast access RAM (OCRAM, 128 KB)
- Secure/non-secure RAM (16 KB)

External memory interfaces:

The i.MX 6Solo/6DualLite processors support latest, high volume, cost effective handheld DRAM, NOR, and NAND Flash memory standards.

- In i.MX 6Solo, 16/32-bit LP-DDR2-800, 16/32-bit DDR3-800, and LV-DDR3-800
- In i.MX 6DualLite, 16/32/64-bit LP-DDR2-800, 16/32/64-bit DDR3-800, and LV-DDR3-800
- 8-bit NAND-Flash, including support for Raw MLC/SLC, 2 KB, 4 KB, and 8 KB page size, BA-NAND, PBA-NAND, LBA-NAND, OneNAND™ and others.
- BCH ECC up to 40 bit.
- 16/32-bit NOR Flash. All EIMv2 pins are muxed on other interfaces.
- 16/32-bit PSRAM, Cellular RAM

Each i.MX 6Solo/6DualLite processor enables the following interfaces to external devices (some of them are muxed and not available simultaneously):
- Displays—Total four interfaces available. Total raw pixel rate of all interfaces is up to 266 Mpixels/sec, 24 bpp. Up to three interfaces may be active in parallel (excluding EPDC).
- One Parallel 24-bit display port, up to 225 Mpixels/sec (for example, WUXGA at 60 Hz or dual HD1080 and WXGA at 60 Hz)
- LVDS serial ports—One port up to 170 Mpixels/sec or two ports up to 85 MP/sec (for example, WUXGA at 60 Hz) each
- HDMI 1.4 port
- MIPI/DSI, two lanes at 1 Gbps
- EPDC, Color, and monochrome E-INK, up to 1650x2332 resolution and 5-bit grayscale

Camera sensors:
- Up to two interfaces may be active at once
- Parallel Camera port (up to 20 bit and up to 240 MHz peak)
- MIPI CSI-2 / Parallel port, supporting from 80 Mbps to 800 Mbps speed per data lane. The CSI-2 Receiver core can manage one clock lane and up to two data lanes. Each i.MX 6Solo/6DualLite processor has two lanes.

Expansion cards:

Four MMC/SD/SDIO card ports all supporting:

- 1-bit or 4-bit transfer mode specifications for SD and SDIO cards up to UHS-I SDR-104 mode (104 MB/s max)
- 1-bit, 4-bit, or 8-bit transfer mode specifications for MMC cards up to 52 MHz in both SDR and DDR modes (104 MB/s max)
- Compatible with SD Memory, miniSD Memory, SDIO, miniSDIO, SD Combo, MMC and MMC RS, embedded MMC and embedded SD cards
- Host clock frequency variable between 32kHz to 52 MHz
- Up to 200 Mbps data transfer for SD/SDIO cards using 4 parallel data lines
- Up to 416 Mbps data transfer for MMC cards using 8 parallel data lines
- Up to 832 Mbps data transfer for MMC/SD cards using 8 parallel data lines in DDR mode
- Supports Single Block, Multi Block and Byte
- Dual internal 32x32-bit FIFOs

USB:

- One High Speed (HS) USB 2.0 OTG (Up to 480 Mbps), with integrated HS USB Phy
- Three USB 2.0 (480 Mbps) hosts:
- One HS host with integrated High Speed Phy
- Two HS hosts with integrated HS-IC USB (High Speed Inter-Chip USB) Phy

Expansion PCI Express port (PCIe) v2.0 one lane:

- PCI Express (Gen 2.0) dual mode complex, supporting Root complex operations and Endpoint operations. Uses x1 PHY configuration.

Miscellaneous IPs and interfaces:

- Three I2S/SSI/AC97, up to 1.4 Mbps each
- Enhanced Serial Audio Interface (ESAI), up to 1.4 Mbps per channel
- Five UARTs, up to 4.0 Mbps each:
  - Providing RS232 interface
  - Supporting 9-bit RS485 multidrop mode
  - One of the five UARTs (UART1) supports 8-wire while others four supports 4-wire. This is due to the SoC IOMUX limitation, since all UART IPs are identical.
- Four eCSPI (Enhanced CSPI). Using chip selects to support multiple peripherals
- Four I2C, supporting 400 kbps
- Gigabit Ethernet Controller (IEEE1588 compliant), 10/100/1000 Mbps
- Four Pulse Width Modulators (PWM)

- System JTAG Controller (SJC)
- GPIO with interrupt capabilities
- 8x8 Key Pad Port (KPP)
- Sony Philips Digital Interface (SPDIF), Rx and Tx
- Two Controller Area Network (FlexCAN), 1 Mbps each
- Two Watchdog timers (WDOG)
- Audio MUX (AUDMUX)
- MLB (MediaLB) provides interface to MOST Networks (MOST25, MOST50, MOST150) with the option of DTCP cipher accelerator

The i.MX 6Solo/6DualLite processors integrate advanced power management unit and controllers:

- Provide PMU, including DCDC and LDO supplies, for on-chip resources
- Use Temperature Sensor for monitoring the die temperature
- Support DVFS techniques for low power modes
- Use SW State Retention and Power Gating for ARM and MPE
- Support various levels of system power modes
- Use flexible clock gating control scheme

The i.MX 6Solo/6DualLite processors use dedicated HW accelerators to meet the targeted multimedia performance. The use of HW accelerators is a key factor in obtaining high performance at low power consumption numbers, while having the CPU core relatively free for performing other tasks.

The i.MX 6Solo/6DualLite processors incorporate the following hardware accelerators:

- VPU—Video Processing Unit
- IPUv3H—Image Processing Unit version 3H
- GPU3Dv5—3D Graphics Processing Unit (OpenGL ES 2.0) version 5
- GPU2Dv2—2D Graphics Processing Unit (BitBlt)
- PXP—PiXel Processing Pipeline. Off loading key pixel processing operations are required to support the EPD display applications.
- ASRC—Asynchronous Sample Rate Converter

Security functions are enabled and accelerated by the following hardware:

- ARM TrustZone including the TZ architecture (separation of interrupts, memory mapping, etc.)
- SJC—System JTAG Controller. Protecting JTAG from debug port attacks by regulating or blocking the access to the system debug features.
- CAAM—Cryptographic Acceleration and Assurance Module, containing cryptographic and hash engines, 16 KB secure RAM, and True and Pseudo Random Number Generator (NIST certified).
- SNVS—Secure Non-Volatile Storage, including Secure Real Time Clock

- CSU—Central Security Unit. Enhancement for the IC Identification Module (IIM). Will be configured during boot and by eFUSEs and will determine the security level operation mode as well as the TZ policy.
- A-HAB—Advanced High Assurance Boot—HABv4 with the new embedded enhancements: SHA-256, 2048-bit RSA key, version control mechanism, warm boot, CSU, and TZ initialization.

NOTE

The actual feature set depends on the part number. Functions, such as video hardware acceleration, and 2D and 3D hardware graphics acceleration may not be enabled for specific part numbers.

1.5 Architectural Overview

This section contains details about the chip architecture.

1.5.1 Block Diagram

A simplified block diagram is provided in the following section.

1.5.1.1 Simplified Block Diagram

A high-level block diagram is shown in the following figure. This diagram provides a view of the chip's major sub-systems (processor domains, shared peripherals domain, and memories) and logical connectivity.

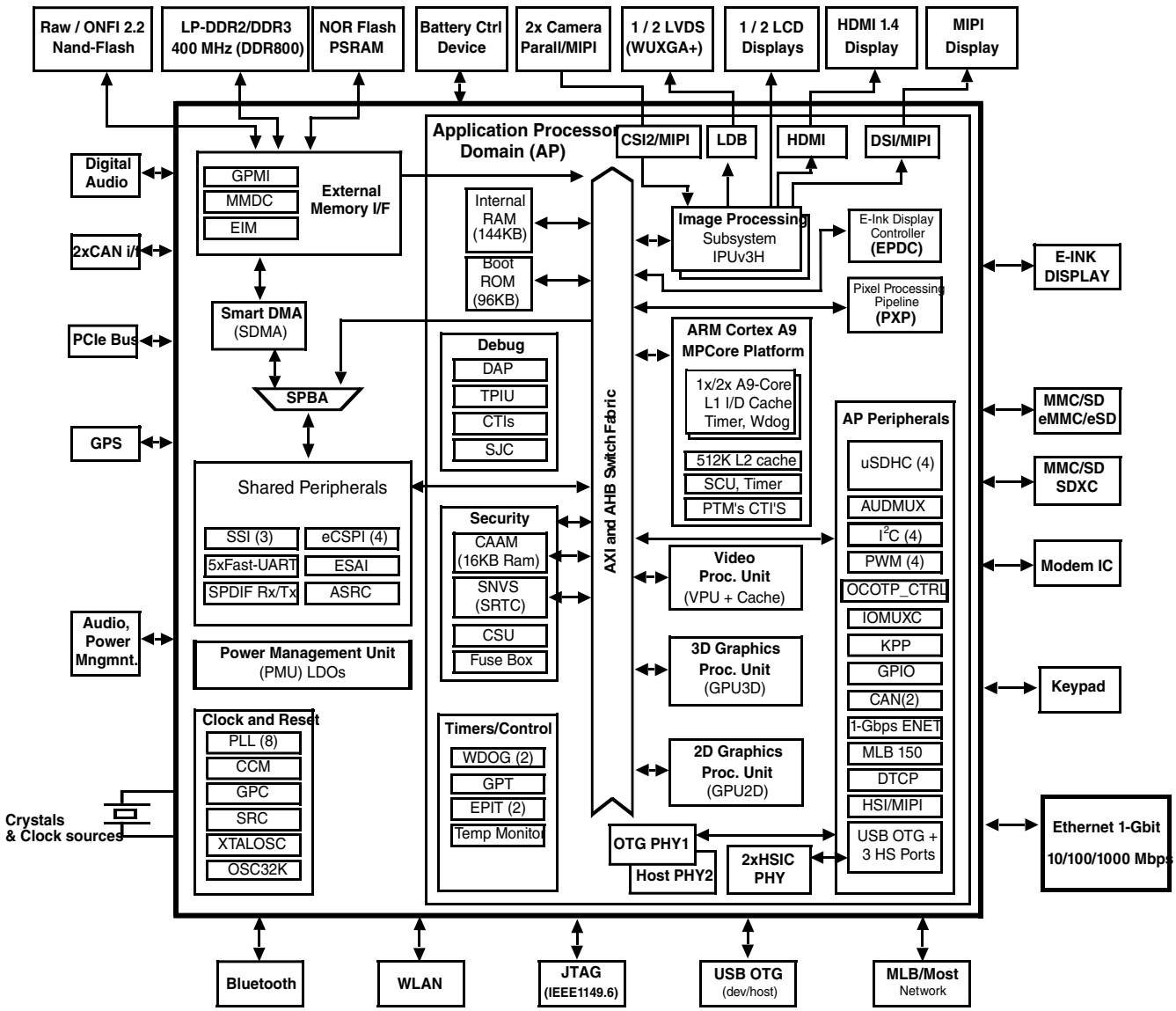


Figure 1-2. Simplified block diagram

1.5.2 Architectural Partitioning

Architecture supports processing-intensive tasks in the following ways:

- ARM Cortex A9 MPCore™ Platform provides hardware features for:
  - Operating System
  - User applications (including control over hardware accelerators and non-accelerated functions)
  - TrustZone applications

- Smart DMA enables data transfer between non-mastering peripherals and external or internal memories
- System Control is supported via:
  - Clock Control Module (CCM)
  - Eight PLLs and two PFDs
  - XTALOSC- 24 MHz Crystal oscillator source support
  - OSC32KHz - 32.768 Hz Crystal oscillator source support
  - System Reset Controller (SRC)
  - General Power Controller (GPC)
  - Temperature Sensor for monitoring and alarming on high temperature situations.
- Multimedia is supported with:
  - One Image Processing Unit-IPUv3H
    - Connectivity to displays and controllers (Parallel, LVDS, MIPI, HDMI), cameras (parallel, MIPI).
    - Display Processing: video/graphics combining, image enhancement
    - Image conversions: resizing, rotation/inversion, color conversion, de-interlacing
    - Synchronization and control capabilities, allowing autonomous operation
  - Video Processing Unit (VPU), supports various decoding/encoding formats in HW, up to 1080p plus SD 30 fps decoding (H.264, VC1, RV10, DivX, etc.), and up to 1080p 30 fps encoding (H.264, etc.).
  - Graphics Processing Unit (GPU3Dv5) graphics processing compliant with the following:
    - OpenGL ES 1.1 and 2.0 including extensions.
    - Windows Direct3D
  - Graphics Processing Unit (GPU2Dv2)
  - Audio
    - Audio codecs are provided by SW, which runs on ARM core, supporting (but not limited to) MP3, WMA, AAC, HE-AAC and Pro10
    - 3x SSIs
    - ESAI
    - SPDIF Tx/Rx
    - Audmux
    - Audio sample rate conversion accelerator (ASRC)
- Security is supported by:
  - High Assurance Boot (HAB4) System
  - ARM TrustZone (TZ) Trusted Execution environment
  - DDR Memory secure region protection by TrustZone Address Space Controller
  - On-chip RAM (OCRAM) single region TrustZone protection
  - Peripheral access policy control, using Central Security Unit (CSU)

    - One-Time Programmable (OTP) electrical fuse array (Total of 3840-bit e-fuses) via the On-chip electrical fuse controller (OCOTP_CTRL)
    - CAAM and SNVS security architecture, providing:
        - 16 KB Secure RAM
        - Secure Real Time Counters
        - Security State Controller
        - Encryption and Hashing functions, Random Number Generator (RNG)
        - Security Violation/Tamper Detection & Reporting
    - System JTAG controller (SJC)
    - Secure Real Time Clock (SRTC)
    - TrustZone Watchdog (TZ WDOG)
- Connectivity peripherals, timers and External Memory Interfaces:
    - Embedded DMAs
    - 3.3V IO voltage for seamless integration
    - Four USB 2.0 ports, including four PHYs: 2x HS-USB (OTG) and 2x HS-IC USB integrated PHYs
    - Memory, 800 MT/s per Line, supports LP-DDR2 (2x16, 2x32, 2x32 interleaved mode - x64), DDR3 (x16/x32/x64) and DDR3L (x16/x32/x64)
    - Nand-Flash (MLC up to 40-bit ECC) and NOR Flash memory interface via GPMI Nand-Flash controller
    - Timers: 2xEPIT, GPT and two Watch Dog timers (one of which is used for TZ), in addition to the timers and watchdog timers integrated within the ARM Cortex A9 MPCore™ platform.
    - Miscellaneous connectivity support - PCIe, FLEXCAN, MLB, MMC/SD, I2C, SPI, UART, PWM and Keypad interface

1.5.3 Endianness Support

The chip supports only the Little Endian mode.

1.5.4 Memory Interfaces

The chip DDR controller supports these memory interfaces:

- LP-DDR2-800, 1x32 channel for i.MX 6Solo and 2x32 channels for i.MX 6DualLite, 400-MHz clock
- DDR3/DDR3L 32-bit in i.MX 6Solo, DDR3/LV-DDR3 32/64-bit in i.MX 6DualLite, both 400-MHz clocks.

NAND flash controller supports the following:

- Eight NAND flash buses
- Configurable page size of 2 KB, 4 KB, or 8 KB
- Configurable spare area per page of up to 512 B
- Supports interleaved accesses to up to eight NAND devices
- Legacy raw SLC-type and MLC-type devices
- Supports up to 40-bit ECC by the hardware BCH ECC accelerator, capable of 200 MB/s encode/decode rate
- ONFI2.2-compliant
  - Timing modes 0-5 for both the asynchronous and synchronous I/F. A synchronous clock rate of up to 100 MHz with a data rate of up to 200 MB/s.
- Vendor-specific devices: BA-NAND (Micron), PBA-NAND and LBA-NAND (Toshiba), E2-NAND (Hynix), EF-NAND (Samsung), Samsung's "Toggle-mode" NAND (clock rate of up to 66 MHz and 80 MHz, with a data rate of up to 133 MB/s and 160 MB/s respectively)

The EIM block provides an interface for SRAM and PSRAM, and a 16/8-bit NOR flash. All EIM pins are muxed on other interfaces.

Chapter 2
Memory Maps

2.1 Memory system overview

This section introduces the memory architecture of the chip.

NOTE

Accessing the reserved memory regions can result in unpredictable behavior.

2.2 Arm Platform Memory Map

The chip memory map has been provided in the following tables. The mapping of the DDR memory address space can be configured by software or hardware fuses, depending on the memory device type (LPDDR2/DDR2/DDR3) and the selected interleaving/non-interleaving scheme.

A combined memory map (of both the 2x32 and x64) is shown in this table:

Table 2-1. System memory map

Start address	End address	Size	Description
1000_0000	FFFF_FFFF	3840 MB	MMDC—DDR Controller. See DDR mapping to MMDC controller ports.
0800_0000	0FFF_FFFF	128 MB	EIM—(NOR/SRAM)
02A0_0000	02DF_FFFF	4 MB	Reserved
0260_0000	029F_FFFF	4 MB	IPU
0220_C000	023F_FFFF	2 MB	Reserved
0220_8000	0220_BFFF	16 KB	MIPI_HSI
0220_4000	0220_7FFF	16 KB	Reserved
0220_0000	0220_3FFF	16 KB	Reserved
0210_0000	021F_FFFF	1 MB	Peripheral IPs via AIPS-2. See Table 2-3.

Table continues on the next page...

Table 2-1. System memory map (continued)

Start address	End address	Size	Description
0200_0000	020F_FFFF	1 MB	Peripheral IPs via AIPS-1. See Table 2-2.
01FF_C000	01FF_FFFF	16 KB	PCIe registers
0100_0000	01FF_BFFF	16368 KB	PCIe
00D0_0000	00FF_FFFF	3072 KB	Reserved
00C0_0000	00CF_FFFF	1 MB	GPV_1 PL301 (fast1)
00B0_0000	00BF_FFFF	1 MB	GPV_0 PL301 (fast2) configuration port
00A0_3000	00AF_FFFF	1012 KB	Reserved
00A0_2000	00A0_2FFF	4 KB	PL310 (L2 cache controller)
00A0_0000	00A0_1FFF	8 KB	ARM MP 0000—00FCh SCU registers 0100—01FFh Interrupt controller interfaces 0200—02FFh Global timer 0300—05FFh Reserved 0600—06FFh Private timers and watchdogs 0700—0FFFh Reserved 1000—1FFFh Interrupt distributor
0092_0000	009F_FFFF	0.875 MB	OCRAM aliased
0090_0000	0091_FFFF	0.125 MB	OCRAM 128 KB
0080_0000	008F_FFFF	1 MB	GPV_4 PL301 (fast3) configuration port
0040_0000	007F_FFFF	4 MB	Reserved
0030_0000	003F_FFFF	1 MB	GPV_3 PL301 (per2) configuration port
0020_0000	002F_FFFF	1 MB	GPV_2 PL301 (per1) configuration port
0013_C000	001F_FFFF	784 KB	Reserved
0013_8000	0013_BFFF	16 KB	DTCP
0013_4000	0013_7FFF	16 KB	GPU 2D
0013_0000	0013_3FFF	16 KB	GPU 3D
0012_9000	0012_FFFF	28 KB	Reserved
0012_0000	0012_8FFF	36 KB	HDMI
0011_8000	0011_FFFF	32 KB	Reserved
0011_4000	0011_7FFF	16 KB	BCH
0011_2000	0011_3FFF	8 KB	GPMI
0011_0000	0011_1FFF	8 KB	APBH DMA
0010_4000	0010_FFFF	48 KB	Reserved
0010_0000	0010_3FFF	16 KB	CAAM (16 KB secure RAM)
0001_8000	000F_FFFF	928 KB	Reserved
0000_0000	0001_7FFF	96 KB	Boot ROM (ROMCP)

The table below shows the Arm IP Bus (AIPS) detailed memory map.

Table 2-2. AIPS-1 memory map

Start Address	End Address	Region	NIC Port	Size
020F_C000	020F_FFFF	AIPS-1	Reserved	16 KB
020F_8000	020F_BFFF		Reserved	16 KB
020F_4000	020F_7FFF		EPDC	16 KB
020F_0000	020F_3FFF		PXP	16 KB
020E_C000	020E_FFFF		SDMA	16 KB
020E_8000	020E_BFFF		DCIC2	16 KB
020E_4000	020E_7FFF		DCIC1	16 KB
020E_0000	020E_3FFF		IOMUXC	16 KB
020D_C2C0	020D_FFFF		Reserved	15680 B
020D_C2A0	020D_C2BF		PGC_ARM	32 B
020D_C280	020D_C29F		Reserved	32 B
020D_C260	020D_C27F		PGC_PU	32 B
020D_C000	020D_C25F		GPC	608 B
020D_8000	020D_BFFF		SRC	16 KB
020D_4000	020D_7FFF		EPIT2	16 KB
020D_0000	020D_3FFF		EPIT1	16 KB
020C_C000	020C_FFFF		SNVS_HP	16 KB
020C_B000	020C_BFFF		Reserved	4 KB
020C_A000	020C_AFFF		USBPHY2	4 KB
020C_9000	020C_9FFF		USBPHY1	4 KB
020C_8000	020C_8FFF		ANALOG: (PLLs, PFDs, Regulators, LDOs, Temp Sensor)	4 KB
020C_4000	020C_7FFF		CCM	16 KB
020C_0000	020C_3FFF		WDOG2	16 KB
020B_C000	020B_FFFF		WDOG1	16 KB
020B_8000	020B_BFFF		KPP	16 KB
020B_4000	020B_7FFF		GPIO7	16 KB
020B_0000	020B_3FFF		GPIO6	16 KB
020A_C000	020A_FFFF		GPIO5	16 KB
020A_8000	020A_BFFF		GPIO4	16 KB
020A_4000	020A_7FFF		GPIO3	16 KB
020A_0000	020A_3FFF		GPIO2	16 KB
0209_C000	0209_FFFF		GPIO1	16 KB
0209_8000	0209_BFFF	AIPS-1	GPT	16 KB
0209_4000	0209_7FFF		CAN2	16 KB
0209_0000	0209_3FFF		CAN1	16 KB
0208_C000	0208_FFFF		PWM4	16 KB
0208_8000	0208_BFFF		PWM3	16 KB
0208_4000	0208_7FFF		PWM2	16 KB

Table continues on the next page...

Table 2-2. AIPS-1 memory map (continued)

Start Address	End Address	Region	NIC Port	Size
0208_0000	0208_3FFF		PWM1	16 KB
0207_C000	0207_FFFF		AIPS-1 Configuration	16 KB
0204_0000	0207_BFFF	AIPS-1	VPU	240 KB
0203_C000	0203_FFFF	AIPS-1 (via SPBA) Glob,Module ENABLE	SPBA	16 KB
0203_8000	0203_BFFF		Reserved for SDMA internal registers	16 KB
0203_4000	0203_7FFF		ASRC	16 KB
0203_0000	0203_3FFF		SSI3	16 KB
0202_C000	0202_FFFF		SSI2	16 KB
0202_8000	0202_BFFF		SSI1	16 KB
0202_4000	0202_7FFF		ESAI	16 KB
0202_0000	0202_3FFF		UART1	16 KB
0201_C000	0201_FFFF		Reserved for SDMA internal registers	16 KB
0201_8000	0201_BFFF		Reserved	16 KB
0201_4000	0201_7FFF		eCSPI4	16 KB
0201_0000	0201_3FFF		eCSPI3	16 KB
0200_C000	0200_FFFF		eCSPI2	16 KB
0200_8000	0200_BFFF		eCSPI1	16 KB
0200_4000	0200_7FFF		SPDIF	16 KB
0200_0000	0200_3FFF		Reserved for SDMA internal registers	16 KB

The table below shows the AIPS-2 detailed memory map.

Table 2-3. AIPS-2 memory map

Start Address	End Address	Region	Allocation	Size
021F_C000	021F_FFFF	AIPS-2	Reserved	16 KB
021F_8000	021F_BFFF		I2C4	16 KB
021F_4000	021F_7FFF		UART5	16 KB
021F_0000	021F_3FFF		UART4	16 KB
021E_C000	021E_FFFF		UART3	16 KB
021E_8000	021E_BFFF		UART2	16 KB
021E_4000	021E_7FFF		VDOA	16 KB
021E_0000	021E_3FFF		MIPI (DSI port)	16 KB
021D_C000	021D_FFFF		MIPI (CSI port)	16 KB
021D_8000	021D_BFFF		AUDMUX	16 KB
021D_4000	021D_7FFF		TZASC2	16 KB
021D_0000	021D_3FFF		TZASC1	16 KB
021C_C000	021C_FFFF		Reserved	16 KB

Table continues on the next page...

Table 2-3. AIPS-2 memory map (continued)

Start Address	End Address	Region	Allocation	Size
021C_8000	021C_BFFF		Reserved	16 KB
021C_4000	021C_7FFF		Reserved	16 KB
021C_0000	021C_3FFF		CSU	16 KB
021B_C000	021B_FFFF		OCOTP_CTRL	16 KB
021B_8000	021B_BFFF		EIM	16 KB
021B_4000	021B_7FFF		MMDC (port 1)	16 KB
021B_0000	021B_3FFF		MMDC	16 KB
021A_C000	021A_FFFF		ROMCP	16 KB
021A_8000	021A_BFFF		I2C3	16 KB
021A_4000	021A_7FFF		I2C2	16 KB
021A_0000	021A_3FFF		I2C1	16 KB
0219_C000	0219_FFFF		uSDHC4	16 KB
0219_8000	0219_BFFF		uSDHC3	16 KB
0219_4000	0219_7FFF		uSDHC2	16 KB
0219_0000	0219_3FFF		uSDHC1	16 KB
0218_C000	0218_FFFF	AIPS-2	MLB150	16 KB
0218_8000	0218_BFFF		ENET	16 KB
0218_4000	0218_7FFF		USBOH3 (USB)	16 KB
0218_0000	0218_3FFF		USBOH3 (pl301)	16 KB
0217_C000	0217_FFFF		AIPS-2 configuration	16 KB
0216_1000	0217_BFFF		Arm Cortex A9 MPCore Platform—Reserved	108 KB
0214_0000	0216_0FFF		Arm Cortex A9 MPCore platform/DAP	132 KB (See Table 2-4)
0211_0000	0213_FFFF		Reserved	192 KB
0210_0000	0210_FFFF		CAAM	64 KB

The table below shows the Debug Access Port (DAP) detailed memory map.

Table 2-4. DAP memory map

Start Address	End Address	Region	Allocation	Size
0216_0000	0216_0FFF	Arm Cortex A9 MPCore Platform, DAP	Platform Control	4 KB
0215_F000	0215_FFFF		Reserved	4 KB
0215_E000	0215_EFFF		Reserved	4 KB
0215_D000	0215_DFFF		PTM1	4 KB
0215_C000	0215_CFFF		PTM0	4 KB
0215_B000	0215_BFFF		Reserved	4 KB
0215_A000	0215_AFFF		Reserved	4 KB
0215_9000	0215_9FFF		CTI1	4 KB

Table continues on the next page...

Table 2-4. DAP memory map (continued)

Start Address	End Address	Region	Allocation	Size
0215_8000	0215_8FFF		CTI0	4 KB
0215_7000	0215_7FFF		Reserved	4 KB
0215_6000	0215_6FFF		Reserved	4 KB
0215_5000	0215_5FFF		Reserved	4 KB
0215_4000	0215_4FFF		Reserved	4 KB
0215_3000	0215_3FFF		CPU1 PMU	4 KB
0215_2000	0215_2FFF		CPU1 Debug i/f	4 KB
0215_1000	0215_1FFF		CPU0 PMU	4 KB
0215_0000	0215_0FFF		CPU0 Debug i/f	4 KB
0214_F000	0214_FFFF		CA9-INTEG	4 KB
0214_5000	0214_EFFF		Reserved	40 KB
0214_4000	0214_4FFF		FUNNEL	4 KB
0214_3000	0214_3FFF		TPIU	4 KB
0214_2000	0214_2FFF		ext. CTI	4 KB
0214_1000	0214_1FFF		ETB	4 KB
0214_0000	0214_0FFF		DAP ROM Table	4 KB

NOTE

Accessing the reserved memory regions can result in unpredictable behavior.

2.3 DDR mapping to MMDC controller ports

The following table lists the various DDR configuration modes and how each mode is mapped to the DDR channels.

Memory Mapping Mode	DDR Memory Map Config[1:0] fuse value	Start Address	End Address	Total Size	MMDC channel/ports
X32/X64-bit, fixed mapping	‘00’	1000_0000	FFFF_FFFF	3840 MB	MMDC #0
Dual-channel (2x 32-bit), Fixed mapping (LPDDR2)	‘01’	8000_0000	FFFF_FFFF	2048 MB	MMDC #0
		1000_0000	7FFF_FFFF	1792 MB	MMDC #1
Dual channel (2x 32-bit), Interleaved mapping (LPDDR2) [1]	‘10’	1000_0000	FFFF_FFFF	3840 MB	MMDC #0 , MMDC #1 (interleaved data)
Illegal	'11'				

1. In the 4KB interleaving mode the system bus maps each consecutive 4KB region to a 4KB region in the other MMDC port, such that "odd" 4KB spaces are mapped to MMDC0, and "even" 4KB regions mapped to MMDC1.

2.4 DMA memory map

The Smart DMA memory map is shown in the following table.

Table 2-5. SDMA peripheral memory map

Peripheral	Base address	Size
Reserved for SDMA internal memory	0x0000	4 KB
SPDIF	0x1000	4 KB
eCSPI1	0x2000	4 KB
eCSPI2	0x3000	4 KB
eCSPI3	0x4000	4 KB
eCSPI4	0x5000	4 KB
Reserved	0x6000	4 KB
Reserved for SDMA internal registers	0x7000	4 KB
UART1	0x8000	4 KB
ESAI	0x9000	4 KB
SSI1	0xA000	4 KB
SSI2	0xB000	4 KB
SSI3	0xC000	4 KB
ASRC	0xD000	4 KB
Reserved	0xE000	4 KB
SPBA Registers	0xF000	4 KB

NOTE

Accessing the reserved memory regions can result in an unpredictable behavior.

Chapter 3
Interrupts and DMA Events

3.1 Overview

This section describes the assignments of interrupts from the Arm domain in Cortex A9 interruptsand from the DMA events in SDMA event mapping.

3.2 Cortex A9 interrupts

The Global Interrupt Controller (GIC) collects up to 128 interrupt requests from all chip sources and provides an interface to the Cortex A9 CPU cores. The first 32 interrupts are private to the CPUs' interface. These interrupts are not included in the table below. All interrupts besides those private to the CPU are hooked up to the GPC.

Each interrupt can be configured as a normal or a secure interrupt. Software force registers and software priority masking are also supported. This table describes the Arm Cortex A9 interrupt sources:

Table 3-1. Arm Cortex A9 domain interrupt summary

IRQ	Interrupt Source	Interrupt Description
32	IOMUXC	General-Purpose Register 1 from IOMUXC. Used to notify the cores on the exception condition while booting.
33	DAP	Debug Access Port interrupt request
34	SDMA	SDMA interrupt request from all channels
35	VPU	JPEG codec interrupt request
36	SNVS	PMIC power off request
37	IPU	IPU error interrupt request
38	IPU1	IPU1 sync interrupt request
39	Reserved	Reserved
40	Reserved	Reserved

Table continues on the next page...

Table 3-1. Arm Cortex A9 domain interrupt summary (continued)

IRQ	Interrupt Source	Interrupt Description
41	GPU3D	GPU3D interrupt request
42	R2D GPU2D	R2D GPU2D general interrupt request
43	Reserved	Reserved
44	VPU	VPU interrupt request
45	APBH DMA	Logical OR of APBH DMA channels 0-3 completion and error interrupts
46	EIM	EIM interrupt request
47	BCH	BCH operation complete interrupt
48	GPMI	GPMI operation timeout error interrupt
49	DTCP	DTCP interrupt request
50	VDOA	VDOA interrupt request
51	SNVS	SRTC consolidated interrupt
52	SNVS	SRTC security interrupt
53	CSU	CSU interrupt request 1. Indicates to the processor that one or more alarm inputs were asserted.
54	uSDHC1	uSDHC1 (Enhanced SDHC) interrupt request
55	uSDHC2	uSDHC2 (Enhanced SDHC) interrupt request
56	uSDHC3	uSDHC3 (Enhanced SDHC) interrupt request
57	uSDHC4	uSDHC4 (Enhanced SDHC) interrupt request
58	UART1	UART1 interrupt request
59	UART2	UART2 interrupt request
60	UART3	UART3 interrupt request
61	UART4	UART4 interrupt request
62	UART5	UART5 interrupt request
63	eCSPI1	eCSPI1 interrupt request
64	eCSPI2	eCSPI2 interrupt request
65	eCSPI3	eCSPI3 interrupt request
66	eCSPI4	eCSPI4 interrupt request
67	I2C4	I2C4 interrupt request
68	I2C1	I2C1 interrupt request
69	I2C2	I2C2 interrupt request
70	I2C3	I2C3 interrupt request
71	Reserved	Reserved
72	USB	USB Host 1 interrupt request
73	USB	USB Host 2 interrupt request
74	USB	USB Host 3 interrupt request
75	USB	USB OTG interrupt request
76	USB_PHY	UTMI0 interrupt request
77	USB_PHY	UTMI1 interrupt request
78	SSI1	SSI1 interrupt request

Table continues on the next page...

Table 3-1. Arm Cortex A9 domain interrupt summary (continued)

IRQ	Interrupt Source	Interrupt Description
79	SSI2	SSI2 interrupt request
80	SSI3	SSI3 interrupt request
81	Temperature Monitor	Temperature Sensor (temperature greater than threshold) interrupt request
82	ASRC	ASRC interrupt request
83	ESAI	ESAI interrupt request
84	SPDIF	SPDIF interrupt
85	MLB150	MLB error interrupt request
86	PMU	PMU analog regulator brownout occurred.
87	GPT	Logical OR of GPT rollover interrupt line, input capture 1 and 2 lines, output compare 1, 2, and 3 interrupt lines
88	EPIT1	EPIT1 output compare interrupt
89	EPIT2	EPIT2 output compare interrupt
90	GPIO1	INT7 interrupt request
91	GPIO1	INT6 interrupt request
92	GPIO1	INT5 interrupt request
93	GPIO1	INT4 interrupt request
94	GPIO1	INT3 interrupt request
95	GPIO1	INT2 interrupt request
96	GPIO1	INT1 interrupt request
97	GPIO1	INT0 interrupt request
98	GPIO1	Combined interrupt indication for GPIO1 signals 0 - 15
99	GPIO1	Combined interrupt indication for GPIO1 signals 16 - 31
100	GPIO2	Combined interrupt indication for GPIO2 signals 0 - 15
101	GPIO2	Combined interrupt indication for GPIO2 signals 16 - 31
102	GPIO3	Combined interrupt indication for GPIO3 signals 0 - 15
103	GPIO3	Combined interrupt indication for GPIO3 signals 16 - 31
104	GPIO4	Combined interrupt indication for GPIO4 signals 0 - 15
105	GPIO4	Combined interrupt indication for GPIO4 signals 16 - 31
106	GPIO5	Combined interrupt indication for GPIO5 signals 0 - 15
107	GPIO5	Combined interrupt indication for GPIO5 signals 16 - 31
108	GPIO6	Combined interrupt indication for GPIO6 signals 0 - 15
109	GPIO6	Combined interrupt indication for GPIO6 signals 16 - 31
110	GPIO7	Combined interrupt indication for GPIO7 signals 0 - 15
111	GPIO7	Combined interrupt indication for GPIO7 signals 16 - 31
112	WDOG1	WDOG1 timer reset interrupt request
113	WDOG2	WDOG2 timer reset interrupt request
114	KPP	Key Pad interrupt request
115	PWM1	Cumulative interrupt line for PWM1. Logical OR of rollover, compare, and FIFO waterlevel crossing interrupts.

Table continues on the next page...

Table 3-1. Arm Cortex A9 domain interrupt summary (continued)

IRQ	Interrupt Source	Interrupt Description
116	PWM2	Cumulative interrupt line for PWM2. Logical OR of rollover, compare, and FIFO waterlevel crossing interrupts.
117	PWM3	Cumulative interrupt line for PWM3. Logical OR of rollover, compare, and FIFO waterlevel crossing interrupts.
118	PWM4	Cumulative interrupt line for PWM4. Logical OR of rollover, compare, and FIFO waterlevel crossing interrupts.
119	CCM	CCM interrupt request 1
120	CCM	CCM interrupt request 2
121	GPC	GPC interrupt request 1
122	Reserved	Reserved
123	SRC	SRC interrupt request
124	CPU	L2 interrupt request
125	CPU	Parity Check error interrupt request
126	CPU	Performance Unit interrupt
127	CPU	CTI trigger outputs interrupt
128	SRC	Combined CPU WDOG interrupts out of SRC
129	EPDC	EPDC interrupt request
130	PXP	PXP interrupt request
131	Reserved	Reserved
132	MIPI_CSI	CSI interrupt request 1
133	MIPI_CSI	CSI interrupt request 2
134	MIPI_DSI	DSI interrupt request
135	MIPI_HSI	HSI interrupt request
136	SJC	SJC interrupt from the General-Purpose register
137	CAAM	CAAM job ring 0 interrupt
138	CAAM	CAAM job ring 1 interrupt
139	Reserved	Reserved
140	TZASC1	TZASC1 interrupt request
141	TZASC2	TZASC2 interrupt request
142	FLEXCAN1	FLEXCAN1 combined interrupt. Logical OR of ini_int_busoff, ini_int_error, ipi_int_mbor, ipi_int_rxwarning, ipi_int_txwarning and ipi_int_wakein. Combined interrupt of ini_int_busoff,ini_int_error,ipi_int_mbor,ipi_int_txwarning and ipi_int_waken
143	FLEXCAN2	FLEXCAN2 combined interrupt. Logical OR of ini_int_busoff, ini_int_error, ipi_int_mbor, ipi_int_rxwarning, ipi_int_txwarning and ipi_int_wakein.
144	Reserved	Reserved
145	Reserved	Reserved
146	Reserved	Reserved
147	HDMI	HDMI video and CEC control interrupt request.
148	HDMI	HDMI CEC dedicated wake-up interrupt.

Table continues on the next page...

Table 3-1. Arm Cortex A9 domain interrupt summary (continued)

IRQ	Interrupt Source	Interrupt Description
149	MLB150	Interrupt request for channels [31:0]. Interrupt request for channels [63:32] available on IRQ #149 if SMX bit is set in MLB150 AHB control register (ACTL), otherwise interrupt for channels [63:32] interrupt is available on IRQ #158.
150	ENET	ENET Interrupt Request
151	ENET	ENET 1588 Timer interrupt [synchronous] request
152	PCIe	PCIe interrupt request 1
153	PCIe	PCIe interrupt request 2
154	PCIe	PCIe interrupt request 3
155	PCIe	PCIe interrupt request 4
156	DCIC1	DCIC1 interrupt request
157	DCIC2	DCIC2 interrupt request
158	MLB150	Interrupt request for channel[63:32]. If SMX bit is not set in MLB150 AHB control register (ACTL), it is the default working mode.
159	PMU	PMU digital regulator brownout occurred.

3.3 SDMA event mapping

This table shows the DMA request signals for the peripherals in the chip:

Table 3-2. SDMA event mapping

Event Number	DMA Source	Description
0	VPU	VPU DMA request
1	I2C4	I2C4 DMA event
2	IPU / IOMUX	IPU-1 DMA Event; Muxed with external DMA pad #1, controlled by GPR0[0]
3	eCSPI1 / I2C3	eCSPI1 Rx request; Muxed with I2C3 controlled by IOMUXC register GPR0[1]
4	eCSPI1 / I2C2	eCSPI1 Tx request; Muxed with I2C2 controlled by IOMUXC register GPR0[2]
5	eCSPI2 / I2C1	eCSPI2 Rx request; Muxed with I2C1 controlled by IOMUXC register GPR0[3]
6	eCSPI2	eCSPI2 Tx request
7	eCSPI3	eCSPI3 Rx request
8	eCSPI3	eCSPI3 Tx request
9	eCSPI4 / EPIT2	eCSPI4 Rx request; Muxed with EPIT2 DMA request controlled by IOMUXC register GPR0[5]
10	eCSPI4/ I2C1	eCSPI4 Tx request; Muxed with I2C1 controlled by IOMUXC register GPR0[4]
11	Reserved	Reserved
12	Reserved	Reserved
13	GPT	GPT counter event

Table continues on the next page...

Table 3-2. SDMA event mapping (continued)

Event Number	DMA Source	Description
14	SPDIF / IOMUX	SPDIF RX DMA request; Muxed with external DMA pad #2 controlled by IOMUXC register GPR0[7]
15	SPDIF	SPDIF TX DMA request
16	EPIT1	EPIT1 request
17	ASRC	ASRC DMA1 request (Pair A input Request)
18	ASRC	ASRC DMA2 request (Pair B input Request)
19	ASRC	ASRC DMA3 request (Pair C input Request)
20	ASRC	ASRC DMA4 request (Pair A output Request)
21	ASRC	ASRC DMA5 request (Pair B output Request)
22	ASRC	ASRC DMA6 request (Pair C output Request)
23	ESAI / I2C3	ESAI Rx FIFO DMA request; Muxed with I2C3 controlled by IOMUXC register GPR0[6]
24	ESAI	ESAI Tx FIFO DMA request
25	UART1	UART1 Rx FIFO
26	UART1	UART1 Tx FIFO
27	UART2	UART2 Rx FIFO
28	UART2	UART2 Tx FIFO
29	UART3	UART3 Rx FIFO
30	UART3	UART3 Tx FIFO
31	UART4	UART4 Rx FIFO
32	UART4	UART4 Tx FIFO
33	UART5	UART5 Rx FIFO
34	UART5	UART5 Tx FIFO
35	SSI1	SSI1 receive 1 DMA request
36	SSI1	SSI1 transmit 1 DMA request
37	SSI1	SSI1 receive 0 DMA request
38	SSI1	SSI1 transmit 0 DMA request
39	SSI2	SSI2 receive 1 DMA request
40	SSI2	SSI2 transmit 1 DMA request
41	SSI2	SSI2 receive 0 DMA request
42	SSI2	SSI2 transmit 0 DMA request
43	SSI3	SSI3 receive 1 DMA request
44	SSI3	SSI3 transmit 1 DMA request
45	SSI3	SSI3 receive 0 DMA request
46	SSI3	SSI3 transmit 0 DMA request
47	DTCP	DTCP DMA request

As shown in the table, some of the events are the output of a mux of two signals or triggers. The selection of this mux is controlled by the general-purpose registers in IOMUXC.

Chapter 4
External Signals and Pin Multiplexing

4.1 Overview

The chip contains a limited number of pins, most of which have multiple signal options. These signal-to-pin and pin-to-signal options are selected by the input-output multiplexer called IOMUX. The IOMUX is also used to configure other pin characteristics, such as voltage level, drive strength, and hysteresis.

Pin Assignments lists the pad names of the chip, the various signals that can be assigned to each of the pads, and the default settings for each pad. The muxing options table lists the external signals grouped by the module instance, the muxing options for each signal, and the registers used to route the signal to the chosen pad.

4.1.1 Pin Assignments

Table 4-1. Pin Assignments

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
BOOT_MODE0		SRC_BOOT_MODE0		
BOOT_MODE1		SRC_BOOT_MODE1		
CLK1_N		XTALOSC_CLK1_N		
CLK1_P		XTALOSC_CLK1_P		
CLK2_N		XTALOSC_CLK2_N		
CLK2_P		XTALOSC_CLK2_P		
CSI0_DAT4	ALT0	IPU1_CSI0_DATA04	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA04
	ALT1	EIM_DATA02	PUS - 100K_OHM_PU	
	ALT2	ECSPI1_SCLK	PUE - PULL	
	ALT3	KEY_COL5	PKE - ENABLED	
	ALT4	AUD3_TXC	ODE - DISABLED	
	ALT5	GPIO5_IO22	SPEED - 100MHZ	
	ALT7	ARM_TRACE01	DSE - 40_OHM	

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			SRE - SLOW	
CSI0_DAT5	ALT0	IPU1_CSI0_DATA05	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA05
	ALT1	EIM_DATA03	PUS - 100K_OHM_PU	
	ALT2	ECSPI1_MOSI	PUE - PULL	
	ALT3	KEY_ROW5	PKE - ENABLED	
	ALT4	AUD3_TXD	ODE - DISABLED	
	ALT5	GPIO5_IO23	SPEED - 100MHZ	
	ALT7	ARM_TRACE02	DSE - 40_OHM	
			SRE - SLOW	
CSI0_DAT6	ALT0	IPU1_CSI0_DATA06	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA06
	ALT1	EIM_DATA04	PUS - 100K_OHM_PU	
	ALT2	ECSPI1_MISO	PUE - PULL	
	ALT3	KEY_COL6	PKE - ENABLED	
	ALT4	AUD3_TXFS	ODE - DISABLED	
	ALT5	GPIO5_IO24	SPEED - 100MHZ	
	ALT7	ARM_TRACE03	DSE - 40_OHM	
			SRE - SLOW	
CSI0_DAT7	ALT0	IPU1_CSI0_DATA07	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA07
	ALT1	EIM_DATA05	PUS - 100K_OHM_PU	
	ALT2	ECSPI1_SS0	PUE - PULL	
	ALT3	KEY_ROW6	PKE - ENABLED	
	ALT4	AUD3_RXD	ODE - DISABLED	
	ALT5	GPIO5_IO25	SPEED - 100MHZ	
	ALT7	ARM_TRACE04	DSE - 40_OHM	
			SRE - SLOW	
CSI0_DAT8	ALT0	IPU1_CSI0_DATA08	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA08
	ALT1	EIM_DATA06	PUS - 100K_OHM_PU	
	ALT2	ECSPI2_SCLK	PUE - PULL	
	ALT3	KEY_COL7	PKE - ENABLED	
	ALT4	I2C1_SDA	ODE - DISABLED	
	ALT5	GPIO5_IO26	SPEED - 100MHZ	
	ALT7	ARM_TRACE05	DSE - 40_OHM	
			SRE - SLOW	
CSI0_DAT9	ALT0	IPU1_CSI0_DATA09	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA09
	ALT1	EIM_DATA07	PUS - 100K_OHM_PU	
	ALT2	ECSPI2_MOSI	PUE - PULL	
	ALT3	KEY_ROW7	PKE - ENABLED	
	ALT4	I2C1_SCL	ODE - DISABLED	

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT5	GPIO5_IO27	SPEED - 100MHZ	
	ALT7	ARM_TRACE06	DSE - 40_OHM SRE - SLOW	
CSI0_DAT10	ALT0	IPU1_CSI0_DATA10	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA10
	ALT1	AUD3_RXC	PUS - 100K_OHM_PU	
	ALT2	ECSPI2_MISO	PUE - PULL	
	ALT3	UART1_TX_DATA	PKE - ENABLED	
	ALT5	GPIO5_IO28	ODE - DISABLED	
	ALT7	ARM_TRACE07	SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
CSI0_DAT11	ALT0	IPU1_CSI0_DATA11	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA11
	ALT1	AUD3_RXFS	PUS - 100K_OHM_PU	
	ALT2	ECSPI2_SS0	PUE - PULL	
	ALT3	UART1_RX_DATA	PKE - ENABLED	
	ALT5	GPIO5_IO29	ODE - DISABLED	
	ALT7	ARM_TRACE08	SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
CSI0_DAT12	ALT0	IPU1_CSI0_DATA12	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA12
	ALT1	EIM_DATA08	PUS - 100K_OHM_PU	
	ALT3	UART4_TX_DATA	PUE - PULL	
	ALT5	GPIO5_IO30	PKE - ENABLED	
	ALT7	ARM_TRACE09	ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
CSI0_DAT13	ALT0	IPU1_CSI0_DATA13	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA13
	ALT1	EIM_DATA09	PUS - 100K_OHM_PU	
	ALT3	UART4_RX_DATA	PUE - PULL	
	ALT5	GPIO5_IO31	PKE - ENABLED	
	ALT7	ARM_TRACE10	ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
CSI0_DAT14	ALT0	IPU1_CSI0_DATA14	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA14
	ALT1	EIM_DATA10	PUS - 100K_OHM_PU	
	ALT3	UART5_TX_DATA	PUE - PULL	

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT5	GPIO6_IO00	PKE - ENABLED	
	ALT7	ARM_TRACE11	ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
CSI0_DAT15	ALT0	IPU1_CSI0_DATA15	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA15
	ALT1	EIM_DATA11	PUS - 100K_OHM_PU	
	ALT3	UART5_RX_DATA	PUE - PULL	
	ALT5	GPIO6_IO01	PKE - ENABLED	
	ALT7	ARM_TRACE12	ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
CSI0_DAT16	ALT0	IPU1_CSI0_DATA16	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA16
	ALT1	EIM_DATA12	PUS - 100K_OHM_PU	
	ALT3	UART4_RTS_B	PUE - PULL	
	ALT5	GPIO6_IO02	PKE - ENABLED	
	ALT7	ARM_TRACE13	ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
CSI0_DAT17	ALT0	IPU1_CSI0_DATA17	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA17
	ALT1	EIM_DATA13	PUS - 100K_OHM_PU	
	ALT3	UART4_CTS_B	PUE - PULL	
	ALT5	GPIO6_IO03	PKE - ENABLED	
	ALT7	ARM_TRACE14	ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
CSI0_DAT18	ALT0	IPU1_CSI0_DATA18	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA18
	ALT1	EIM_DATA14	PUS - 100K_OHM_PU	
	ALT3	UART5_RTS_B	PUE - PULL	
	ALT5	GPIO6_IO04	PKE - ENABLED	
	ALT7	ARM_TRACE15	ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
CSI0_DAT19	ALT0	IPU1_CSI0_DATA19	HYS - ENABLED	SW_PAD_CTL_PAD_CSI0_DATA19

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT1	EIM_DATA15	PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
	ALT3	UART5_CTS_B		
	ALT5	GPIO6_IO05		
CSI0_DATA_EN	ALT0	IPU1_CSI0_DATA_EN	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_CSI0_DATA_EN
	ALT1	EIM_DATA00		
	ALT5	GPIO5_IO20		
	ALT7	ARM_TRACE_CLK		
CSI0_MCLK	ALT0	IPU1_CSI0_HSYNC	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_CSI0_HSYNC
	ALT3	CCM_CLKO1		
	ALT5	GPIO5_IO19		
	ALT7	ARM_TRACE_CTL		
CSI0_PIXCLK	ALT0	IPU1_CSI0_PIXCLK	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_CSI0_PIXCLK
	ALT5	GPIO5_IO18		
	ALT7	ARM_EVENTO		
CSI0_VSYNC	ALT0	IPU1_CSI0_VSYNC	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM	SW_PAD_CTL_PAD_CSI0_VSYNC
	ALT1	EIM_DATA01		
	ALT5	GPIO5_IO21		
	ALT7	ARM_TRACE00		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			SRE - SLOW	
CSI_CLK0M		CSI_CLK0_N		
CSI_CLK0P		CSI_CLK0_P		
CSI_D0M		CSI_DATA0_N		
CSI_D0P		CSI_DATA0_P		
CSI_D1M		CSI_DATA1_N		
CSI_D1P		CSI_DATA1_P		
DI0_DISP_CLK	ALT0	IPU1_DI0_DISP_CLK	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DI0_DISP_CLK
	ALT5	GPIO4_IO16		
DI0_PIN2	ALT0	IPU1_DI0_PIN02	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DI0_PIN02
	ALT2	AUD6_TXD		
	ALT5	GPIO4_IO18		
DI0_PIN3	ALT0	IPU1_DI0_PIN03	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DI0_PIN03
	ALT2	AUD6_TXFS		
	ALT5	GPIO4_IO19		
DI0_PIN4	ALT0	IPU1_DI0_PIN04	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM	SW_PAD_CTL_PAD_DI0_PIN04
	ALT2	AUD6_RXD		
	ALT3	SD1_WP		
	ALT5	GPIO4_IO20		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			SRE - SLOW	
DI0_PIN15	ALT0	IPU1_DI0_PIN15	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DI0_PIN15
	ALT2	AUD6_TXC		
	ALT5	GPIO4_IO17		
DISP0_DAT0	ALT0	IPU1_DISP0_DATA00	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA00
	ALT2	ECSPI3_SCLK		
	ALT5	GPIO4_IO21		
DISP0_DAT1	ALT0	IPU1_DISP0_DATA01	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA01
	ALT2	ECSPI3_MOSI		
	ALT5	GPIO4_IO22		
DISP0_DAT2	ALT0	IPU1_DISP0_DATA02	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA02
	ALT2	ECSPI3_MISO		
	ALT5	GPIO4_IO23		
DISP0_DAT3	ALT0	IPU1_DISP0_DATA03	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED	SW_PAD_CTL_PAD_DISP0_DATA03
	ALT2	ECSPI3_SS0		
	ALT5	GPIO4_IO24		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
DISP0_DAT4	ALT0	IPU1_DISP0_DATA04	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA04
	ALT2	ECSPI3_SS1		
	ALT5	GPIO4_IO25		
DISP0_DAT5	ALT0	IPU1_DISP0_DATA05	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA05
	ALT2	ECSPI3_SS2		
	ALT3	AUD6_RXFS		
	ALT5	GPIO4_IO26		
DISP0_DAT6	ALT0	IPU1_DISP0_DATA06	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA06
	ALT2	ECSPI3_SS3		
	ALT3	AUD6_RXC		
	ALT5	GPIO4_IO27		
DISP0_DAT7	ALT0	IPU1_DISP0_DATA07	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA07
	ALT2	ECSPI3_RDY		
	ALT5	GPIO4_IO28		
DISP0_DAT8	ALT0	IPU1_DISP0_DATA08	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL	SW_PAD_CTL_PAD_DISP0_DATA08
	ALT2	PWM1_OUT		
	ALT3	WDOG1_B		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT5	GPIO4_IO29	PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
DISP0_DAT9	ALT0	IPU1_DISP0_DATA09	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA09
	ALT2	PWM2_OUT		
	ALT3	WDOG2_B		
	ALT5	GPIO4_IO30		
DISP0_DAT10	ALT0	IPU1_DISP0_DATA10	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA10
	ALT5	GPIO4_IO31		
DISP0_DAT11	ALT0	IPU1_DISP0_DATA11	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA11
	ALT5	GPIO5_IO05		
DISP0_DAT12	ALT0	IPU1_DISP0_DATA12	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA12
	ALT5	GPIO5_IO06		
DISP0_DAT13	ALT0	IPU1_DISP0_DATA13	HYS - ENABLED	SW_PAD_CTL_PAD_DISP0_DATA13

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT3	AUD5_RXFS	PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
	ALT5	GPIO5_IO07		
DISP0_DAT14	ALT0	IPU1_DISP0_DATA14	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA14
	ALT3	AUD5_RXC		
	ALT5	GPIO5_IO08		
DISP0_DAT15	ALT0	IPU1_DISP0_DATA15	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA15
	ALT2	ECSPI1_SS1		
	ALT3	ECSPI2_SS1		
	ALT5	GPIO5_IO09		
DISP0_DAT16	ALT0	IPU1_DISP0_DATA16	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA16
	ALT2	ECSPI2_MOSI		
	ALT3	AUD5_TXC		
	ALT4	SDMA_EXT_EVENT0		
	ALT5	GPIO5_IO10		
DISP0_DAT17	ALT0	IPU1_DISP0_DATA17	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM	SW_PAD_CTL_PAD_DISP0_DATA17
	ALT2	ECSPI2_MISO		
	ALT3	AUD5_TXD		
	ALT4	SDMA_EXT_EVENT1		
	ALT5	GPIO5_IO11		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			SRE - SLOW	
DISP0_DAT18	ALT0	IPU1_DISP0_DATA18	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA18
	ALT2	ECSPI2_SS0		
	ALT3	AUD5_TXFS		
	ALT4	AUD4_RXFS		
	ALT5	GPIO5_IO12		
	ALT7	EIM_CS2_B		
DISP0_DAT19	ALT0	IPU1_DISP0_DATA19	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA19
	ALT2	ECSPI2_SCLK		
	ALT3	AUD5_RXD		
	ALT4	AUD4_RXC		
	ALT5	GPIO5_IO13		
	ALT7	EIM_CS3_B		
DISP0_DAT20	ALT0	IPU1_DISP0_DATA20	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA20
	ALT2	ECSPI1_SCLK		
	ALT3	AUD4_TXC		
	ALT5	GPIO5_IO14		
DISP0_DAT21	ALT0	IPU1_DISP0_DATA21	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA21
	ALT2	ECSPI1_MOSI		
	ALT3	AUD4_TXD		
	ALT5	GPIO5_IO15		
DISP0_DAT22	ALT0	IPU1_DISP0_DATA22	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED	SW_PAD_CTL_PAD_DISP0_DATA22
	ALT2	ECSPI1_MISO		
	ALT3	AUD4_TXFS		
	ALT5	GPIO5_IO16		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
DISP0_DAT23	ALT0	IPU1_DISP0_DATA23	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_DISP0_DATA23
	ALT2	ECSPI1_SS0		
	ALT3	AUD4_RXD		
	ALT5	GPIO5_IO17		
DRAM_A0		DRAM_ADDR00	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_ADDR00 SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_ADDDS
DRAM_A1		DRAM_ADDR01	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_ADDR01 SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_ADDDS
DRAM_A2		DRAM_ADDR02	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_ADDR02 SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_ADDDS
DRAM_A3		DRAM_ADDR03	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_ADDR03 SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_ADDDS
DRAM_A4		DRAM_ADDR04	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_ADDR04 SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_ADDDS
DRAM_A5		DRAM_ADDR05	DDR_SEL - LPDDR2 DDR_INPUT - CMOS	SW_PAD_CTL_PAD_DRAM_ADDR05 SW_PAD_CTL_GRP_DDR_TYPE

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			HYS - DISABLED	SW_PAD_CTL_GRP_ADDDS
			ODT - DISABLED	
			DSE - 40_OHM	
DRAM_A6		DRAM_ADDR06	DDR_SEL - LPDDR2	SW_PAD_CTL_PAD_DRAM_ADDR06
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE
			HYS - DISABLED	SW_PAD_CTL_GRP_ADDDS
			ODT - DISABLED	
			DSE - 40_OHM	
DRAM_A7		DRAM_ADDR07	DDR_SEL - LPDDR2	SW_PAD_CTL_PAD_DRAM_ADDR07
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE
			HYS - DISABLED	SW_PAD_CTL_GRP_ADDDS
			ODT - DISABLED	
			DSE - 40_OHM	
DRAM_A8		DRAM_ADDR08	DDR_SEL - LPDDR2	SW_PAD_CTL_PAD_DRAM_ADDR08
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE
			HYS - DISABLED	SW_PAD_CTL_GRP_ADDDS
			ODT - DISABLED	
			DSE - 40_OHM	
DRAM_A9		DRAM_ADDR09	DDR_SEL - LPDDR2	SW_PAD_CTL_PAD_DRAM_ADDR09
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE
			HYS - DISABLED	SW_PAD_CTL_GRP_ADDDS
			ODT - DISABLED	
			DSE - 40_OHM	
DRAM_A10		DRAM_ADDR10	DDR_SEL - LPDDR2	SW_PAD_CTL_PAD_DRAM_ADDR10
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE
			HYS - DISABLED	SW_PAD_CTL_GRP_ADDDS
			ODT - DISABLED	
			DSE - 40_OHM	
DRAM_A11		DRAM_ADDR11	DDR_SEL - LPDDR2	SW_PAD_CTL_PAD_DRAM_ADDR11
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE
			HYS - DISABLED	SW_PAD_CTL_GRP_ADDDS
			ODT - DISABLED	
			DSE - 40_OHM	
DRAM_A12		DRAM_ADDR12	DDR_SEL - LPDDR2	SW_PAD_CTL_PAD_DRAM_ADDR12
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE
			HYS - DISABLED	SW_PAD_CTL_GRP_ADDDS
			ODT - DISABLED	
			DSE - 40_OHM	

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
DRAM_A13		DRAM_ADDR13	DDR_SEL - LPDDR2	SW_PAD_CTL_PAD_DRAM_ADDR13
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE
			HYS - DISABLED	SW_PAD_CTL_GRP_ADDDS
			ODT - DISABLED	
			DSE - 40_OHM	
DRAM_A14		DRAM_ADDR14	DDR_SEL - LPDDR2	SW_PAD_CTL_PAD_DRAM_ADDR14
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE
			HYS - DISABLED	SW_PAD_CTL_GRP_ADDDS
			ODT - DISABLED	
			DSE - 40_OHM	
DRAM_A15		DRAM_ADDR15	DDR_SEL - LPDDR2	SW_PAD_CTL_PAD_DRAM_ADDR15
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE
			HYS - DISABLED	SW_PAD_CTL_GRP_ADDDS
			ODT - DISABLED	
			DSE - 40_OHM	
DRAM_CAS		DRAM_CAS_B	DDR_SEL - LPDDR2	SW_PAD_CTL_PAD_DRAM_CAS_B
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE
			HYS - DISABLED	
			ODT - DISABLED	
			DSE - 40_OHM	
DRAM_CS0		DRAM_CS0_B	DDR_SEL - LPDDR2	SW_PAD_CTL_PAD_DRAM_CS0_B
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE
			HYS - DISABLED	SW_PAD_CTL_GRP_CTLDS
			ODT - DISABLED	
			DSE - 40_OHM	
DRAM_CS1		DRAM_CS1_B	DDR_SEL - LPDDR2	SW_PAD_CTL_PAD_DRAM_CS1_B
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE
			HYS - DISABLED	SW_PAD_CTL_GRP_CTLDS
			ODT - DISABLED	
			DSE - 40_OHM	
DRAM_D0		DRAM_DATA00	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B0DS
DRAM_D1		DRAM_DATA01	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B0DS
DRAM_D2		DRAM_DATA02	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B0DS
DRAM_D3		DRAM_DATA03	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B0DS
DRAM_D4		DRAM_DATA04	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B0DS
DRAM_D5		DRAM_DATA05	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B0DS
DRAM_D6		DRAM_DATA06	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B0DS
DRAM_D7		DRAM_DATA07	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B0DS
DRAM_D8		DRAM_DATA08	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B1DS
DRAM_D9		DRAM_DATA09	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B1DS
DRAM_D10		DRAM_DATA10	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B1DS
DRAM_D11		DRAM_DATA11	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B1DS
DRAM_D12		DRAM_DATA12	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B1DS
DRAM_D13		DRAM_DATA13	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B1DS

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
DRAM_D14		DRAM_DATA14	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B1DS
DRAM_D15		DRAM_DATA15	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B1DS
DRAM_D16		DRAM_DATA16	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B2DS
DRAM_D17		DRAM_DATA17	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B2DS
DRAM_D18		DRAM_DATA18	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B2DS
DRAM_D19		DRAM_DATA19	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B2DS
DRAM_D20		DRAM_DATA20	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B2DS
DRAM_D21		DRAM_DATA21	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B2DS
DRAM_D22		DRAM_DATA22	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B2DS
DRAM_D23		DRAM_DATA23	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B2DS
DRAM_D24		DRAM_DATA24	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B3DS
DRAM_D25		DRAM_DATA25	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B3DS
DRAM_D26		DRAM_DATA26	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B3DS
DRAM_D27		DRAM_DATA27	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B3DS
DRAM_D28		DRAM_DATA28	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B3DS
DRAM_D29		DRAM_DATA29	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B3DS
DRAM_D30		DRAM_DATA30	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B3DS
DRAM_D31		DRAM_DATA31	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B3DS
DRAM_D32		DRAM_DATA32	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B4DS

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
DRAM_D33		DRAM_DATA33	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B4DS
DRAM_D34		DRAM_DATA34	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B4DS
DRAM_D35		DRAM_DATA35	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B4DS
DRAM_D36		DRAM_DATA36	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B4DS
DRAM_D37		DRAM_DATA37	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B4DS
DRAM_D38		DRAM_DATA38	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B4DS
DRAM_D39		DRAM_DATA39	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B4DS
DRAM_D40		DRAM_DATA40	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B5DS
DRAM_D41		DRAM_DATA41	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B5DS
DRAM_D42		DRAM_DATA42	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B5DS
DRAM_D43		DRAM_DATA43	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B5DS
DRAM_D44		DRAM_DATA44	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B5DS
DRAM_D45		DRAM_DATA45	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B5DS
DRAM_D46		DRAM_DATA46	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B5DS
DRAM_D47		DRAM_DATA47	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B5DS
DRAM_D48		DRAM_DATA48	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B6DS
DRAM_D49		DRAM_DATA49	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B6DS
DRAM_D50		DRAM_DATA50	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B6DS
DRAM_D51		DRAM_DATA51	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B6DS

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
DRAM_D52		DRAM_DATA52	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B6DS
DRAM_D53		DRAM_DATA53	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B6DS
DRAM_D54		DRAM_DATA54	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B6DS
DRAM_D55		DRAM_DATA55	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B6DS
DRAM_D56		DRAM_DATA56	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B7DS
DRAM_D57		DRAM_DATA57	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B7DS
DRAM_D58		DRAM_DATA58	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B7DS
DRAM_D59		DRAM_DATA59	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B7DS
DRAM_D60		DRAM_DATA60	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B7DS
DRAM_D61		DRAM_DATA61	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B7DS
DRAM_D62		DRAM_DATA62	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B7DS
DRAM_D63		DRAM_DATA63	DDR_SEL - LPDDR2	SW_PAD_CTL_GRP_DDR_TYPE
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDRMODE
			HYS - DISABLED	SW_PAD_CTL_GRP_DDRHYS
			PUE - PULL	SW_PAD_CTL_GRP_DDRPK
			PKE - ENABLED	SW_PAD_CTL_GRP_DDRPKE
			DSE - 40_OHM	SW_PAD_CTL_GRP_B7DS
DRAM_DQM0		DRAM_DQM0	DDR_SEL - LPDDR2	SW_PAD_CTL_PAD_DRAM_DQM0
			DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE
			HYS - DISABLED	
			ODT - DISABLED	

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			DSE - 40_OHM	
DRAM_DQM1		DRAM_DQM1	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_DQM1 SW_PAD_CTL_GRP_DDR_TYPE
DRAM_DQM2		DRAM_DQM2	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_DQM2 SW_PAD_CTL_GRP_DDR_TYPE
DRAM_DQM3		DRAM_DQM3	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_DQM3 SW_PAD_CTL_GRP_DDR_TYPE
DRAM_DQM4		DRAM_DQM4	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_DQM4 SW_PAD_CTL_GRP_DDR_TYPE
DRAM_DQM5		DRAM_DQM5	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_DQM5 SW_PAD_CTL_GRP_DDR_TYPE
DRAM_DQM6		DRAM_DQM6	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_DQM6 SW_PAD_CTL_GRP_DDR_TYPE
DRAM_DQM7		DRAM_DQM7	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_DQM7 SW_PAD_CTL_GRP_DDR_TYPE
DRAM_SDODT0		DRAM_ODT0	DDR_SEL - LPDDR2 DDR_INPUT - CMOS	SW_PAD_CTL_PAD_DRAM_ODT0 SW_PAD_CTL_GRP_DDR_TYPE

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			HYS - DISABLED ODT - DISABLED DSE - 40_OHM	
DRAM_SDODT1		DRAM_ODT1	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_ODT1 SW_PAD_CTL_GRP_DDR_TYPE
DRAM_RAS		DRAM_RAS_B	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_RAS_B SW_PAD_CTL_GRP_DDR_TYPE
DRAM_RESET		DRAM_RESET	DDR_SEL - RESERVED2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_RESET
DRAM_SDBA0		DRAM_SDBA0	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_SDBA0 SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_ADDDS
DRAM_SDBA1		DRAM_SDBA1	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_SDBA1 SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_ADDDS
DRAM_SDBA2		DRAM_SDBA2	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_SDBA2 SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_CTLDS
DRAM_SDCKE0		DRAM_SDCKE0	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED	SW_PAD_CTL_PAD_DRAM_SDCKE0 SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_CTLDS

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			DSE - 40_OHM	
DRAM_SDCKE1		DRAM_SDCKE1	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_SDCKE1 SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_CTLDS
DRAM_SDCLK_0_B		DRAM_SDCLK0_N		
DRAM_SDCLK_0		DRAM_SDCLK0_P	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_SDCLK0_P SW_PAD_CTL_GRP_DDR_TYPE
DRAM_SDCLK_1_B		DRAM_SDCLK1_N		
DRAM_SDCLK_1		DRAM_SDCLK1_P	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_SDCLK1_P SW_PAD_CTL_GRP_DDR_TYPE
DRAM_SDQS0_B		DRAM_SDQS0_N		
DRAM_SDQS0		DRAM_SDQS0_P	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED PUS - 100K_OHM_PD PUE - PULL PKE - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_SDQS0_P SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_DDRMODE_CTL SW_PAD_CTL_GRP_DDRHYS
DRAM_SDQS1_B		DRAM_SDQS1_N		
DRAM_SDQS1		DRAM_SDQS1_P	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED PUS - 100K_OHM_PD PUE - PULL PKE - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_SDQS1_P SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_DDRMODE_CTL SW_PAD_CTL_GRP_DDRHYS
DRAM_SDQS2_B		DRAM_SDQS2_N		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
DRAM_SDQS2		DRAM_SDQS2_P	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED PUS - 100K_OHM_PD PUE - PULL PKE - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_SDQS2_P SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_DDRMODE_CTL SW_PAD_CTL_GRP_DDRHYS
DRAM_SDQS3_B		DRAM_SDQS3_N		
DRAM_SDQS3		DRAM_SDQS3_P	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED PUS - 100K_OHM_PD PUE - PULL PKE - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_SDQS3_P SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_DDRMODE_CTL SW_PAD_CTL_GRP_DDRHYS
DRAM_SDQS4_B		DRAM_SDQS4_N		
DRAM_SDQS4		DRAM_SDQS4_P	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED PUS - 100K_OHM_PD PUE - PULL PKE - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_SDQS4_P SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_DDRMODE_CTL SW_PAD_CTL_GRP_DDRHYS
DRAM_SDQS5_B		DRAM_SDQS5_N		
DRAM_SDQS5		DRAM_SDQS5_P	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED PUS - 100K_OHM_PD PUE - PULL PKE - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_SDQS5_P SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_DDRMODE_CTL SW_PAD_CTL_GRP_DDRHYS
DRAM_SDQS6_B		DRAM_SDQS6_N		
DRAM_SDQS6		DRAM_SDQS6_P	DDR_SEL - LPDDR2 DDR_INPUT - CMOS	SW_PAD_CTL_PAD_DRAM_SDQS6_P SW_PAD_CTL_GRP_DDR_TYPE

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			HYS - DISABLED PUS - 100K_OHM_PD PUE - PULL PKE - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_GRP_DDRMODE_CTL SW_PAD_CTL_GRP_DDRHYS
DRAM_SDQS7_B		DRAM_SDQS7_N		
DRAM_SDQS7		DRAM_SDQS7_P	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED PUS - 100K_OHM_PD PUE - PULL PKE - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_SDQS7_P SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_DDRMODE_CTL SW_PAD_CTL_GRP_DDRHYS
DRAM_SDWE		DRAM_SDWE_B	DDR_SEL - LPDDR2 DDR_INPUT - CMOS HYS - DISABLED ODT - DISABLED DSE - 40_OHM	SW_PAD_CTL_PAD_DRAM_SDWE_B SW_PAD_CTL_GRP_DDR_TYPE SW_PAD_CTL_GRP_CTLDS
DSI_CLK0M		DSI_CLK0_N		
DSI_CLK0P		DSI_CLK0_P		
DSI_D0M		DSI_DATA0_N		
DSI_D0P		DSI_DATA0_P		
DSI_D1M		DSI_DATA1_N		
DSI_D1P		DSI_DATA1_P		
EIM_DA0	ALT0	EIM_AD00	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_AD00
	ALT1	IPU1_DISP1_DATA09		
	ALT2	IPU1_CSI1_DATA09		
	ALT5	GPIO3_IO00		
	ALT7	SRC_BOOT_CFG00		
	ALT8	EPDC_SDCLK_N		
EIM_DA1	ALT0	EIM_AD01	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED	SW_PAD_CTL_PAD_EIM_AD01
	ALT1	IPU1_DISP1_DATA08		
	ALT2	IPU1_CSI1_DATA08		
	ALT5	GPIO3_IO01		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT7	SRC_BOOT_CFG01	ODE - DISABLED	
	ALT8	EPDC_SDLE	SPEED - 100MHZ DSE - 40_OHM SRE - FAST	
EIM_DA2	ALT0	EIM_AD02	HYS - DISABLED	SW_PAD_CTL_PAD_EIM_AD02
	ALT1	IPU1_DISP1_DATA07	PUS - 100K_OHM_PU	
	ALT2	IPU1_CSI1_DATA07	PUE - PULL	
	ALT5	GPIO3_IO02	PKE - ENABLED	
	ALT7	SRC_BOOT_CFG02	ODE - DISABLED	
	ALT8	EPDC_BDR0	SPEED - 100MHZ DSE - 40_OHM SRE - FAST	
EIM_DA3	ALT0	EIM_AD03	HYS - DISABLED	SW_PAD_CTL_PAD_EIM_AD03
	ALT1	IPU1_DISP1_DATA06	PUS - 100K_OHM_PU	
	ALT2	IPU1_CSI1_DATA06	PUE - PULL	
	ALT5	GPIO3_IO03	PKE - ENABLED	
	ALT7	SRC_BOOT_CFG03	ODE - DISABLED	
	ALT8	EPDC_BDR1	SPEED - 100MHZ DSE - 40_OHM SRE - FAST	
EIM_DA4	ALT0	EIM_AD04	HYS - DISABLED	SW_PAD_CTL_PAD_EIM_AD04
	ALT1	IPU1_DISP1_DATA05	PUS - 100K_OHM_PU	
	ALT2	IPU1_CSI1_DATA05	PUE - PULL	
	ALT5	GPIO3_IO04	PKE - ENABLED	
	ALT7	SRC_BOOT_CFG04	ODE - DISABLED	
	ALT8	EPDC_SDCE0	SPEED - 100MHZ DSE - 40_OHM SRE - FAST	
EIM_DA5	ALT0	EIM_AD05	HYS - DISABLED	SW_PAD_CTL_PAD_EIM_AD05
	ALT1	IPU1_DISP1_DATA04	PUS - 100K_OHM_PU	
	ALT2	IPU1_CSI1_DATA04	PUE - PULL	
	ALT5	GPIO3_IO05	PKE - ENABLED	
	ALT7	SRC_BOOT_CFG05	ODE - DISABLED	
	ALT8	EPDC_SDCE1	SPEED - 100MHZ DSE - 40_OHM SRE - FAST	
EIM_DA6	ALT0	EIM_AD06	HYS - DISABLED	SW_PAD_CTL_PAD_EIM_AD06
	ALT1	IPU1_DISP1_DATA03	PUS - 100K_OHM_PU	

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT2	IPU1_CSI1_DATA03	PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	
	ALT5	GPIO3_IO06		
	ALT7	SRC_BOOT_CFG06		
	ALT8	EPDC_SDCE2		
EIM_DA7	ALT0	EIM_AD07	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_AD07
	ALT1	IPU1_DISP1_DATA02		
	ALT2	IPU1_CSI1_DATA02		
	ALT5	GPIO3_IO07		
	ALT7	SRC_BOOT_CFG07		
	ALT8	EPDC_SDCE3		
EIM_DA8	ALT0	EIM_AD08	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_AD08
	ALT1	IPU1_DISP1_DATA01		
	ALT2	IPU1_CSI1_DATA01		
	ALT5	GPIO3_IO08		
	ALT7	SRC_BOOT_CFG08		
	ALT8	EPDC_SDCE4		
EIM_DA9	ALT0	EIM_AD09	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_AD09
	ALT1	IPU1_DISP1_DATA00		
	ALT2	IPU1_CSI1_DATA00		
	ALT5	GPIO3_IO09		
	ALT7	SRC_BOOT_CFG09		
	ALT8	EPDC_SDCE5		
EIM_DA10	ALT0	EIM_AD10	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_AD10
	ALT1	IPU1_DI1_PIN15		
	ALT2	IPU1_CSI1_DATA_EN		
	ALT5	GPIO3_IO10		
	ALT7	SRC_BOOT_CFG10		
	ALT8	EPDC_DATA01		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
EIM_DA11	ALT0	EIM_AD11	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_AD11
	ALT1	IPU1_DI1_PIN02		
	ALT2	IPU1_CSI1_HSYNC		
	ALT5	GPIO3_IO11		
	ALT7	SRC_BOOT_CFG11		
	ALT8	EPDC_DATA03		
EIM_DA12	ALT0	EIM_AD12	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_AD12
	ALT1	IPU1_DI1_PIN03		
	ALT2	IPU1_CSI1_VSYNC		
	ALT5	GPIO3_IO12		
	ALT7	SRC_BOOT_CFG12		
	ALT8	EPDC_DATA02		
EIM_DA13	ALT0	EIM_AD13	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_AD13
	ALT1	IPU1_DI1_D0_CS		
	ALT5	GPIO3_IO13		
	ALT7	SRC_BOOT_CFG13		
	ALT8	EPDC_DATA13		
EIM_DA14	ALT0	EIM_AD14	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_AD14
	ALT1	IPU1_DI1_D1_CS		
	ALT5	GPIO3_IO14		
	ALT7	SRC_BOOT_CFG14		
	ALT8	EPDC_DATA14		
EIM_DA15	ALT0	EIM_AD15	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED	SW_PAD_CTL_PAD_EIM_AD15
	ALT1	IPU1_DI1_PIN01		
	ALT2	IPU1_DI1_PIN04		
	ALT5	GPIO3_IO15		
	ALT7	SRC_BOOT_CFG15		
	ALT8	EPDC_DATA09		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			SPEED - 100MHZ DSE - 40_OHM SRE - FAST	
EIM_A16	ALT0	EIM_ADDR16	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_ADDR16
	ALT1	IPU1_DI1_DISP_CLK		
	ALT2	IPU1_CSI1_PIXCLK		
	ALT5	GPIO2_IO22		
	ALT7	SRC_BOOT_CFG16		
	ALT8	EPDC_DATA00		
EIM_A17	ALT0	EIM_ADDR17	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_ADDR17
	ALT1	IPU1_DISP1_DATA12		
	ALT2	IPU1_CSI1_DATA12		
	ALT5	GPIO2_IO21		
	ALT7	SRC_BOOT_CFG17		
	ALT8	EPDC_PWR_STAT		
EIM_A18	ALT0	EIM_ADDR18	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_ADDR18
	ALT1	IPU1_DISP1_DATA13		
	ALT2	IPU1_CSI1_DATA13		
	ALT5	GPIO2_IO20		
	ALT7	SRC_BOOT_CFG18		
	ALT8	EPDC_PWR_CTRL0		
EIM_A19	ALT0	EIM_ADDR19	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_ADDR19
	ALT1	IPU1_DISP1_DATA14		
	ALT2	IPU1_CSI1_DATA14		
	ALT5	GPIO2_IO19		
	ALT7	SRC_BOOT_CFG19		
	ALT8	EPDC_PWR_CTRL1		
EIM_A20	ALT0	EIM_ADDR20	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL	SW_PAD_CTL_PAD_EIM_ADDR20
	ALT1	IPU1_DISP1_DATA15		
	ALT2	IPU1_CSI1_DATA15		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT5	GPIO2_IO18	PKE - ENABLED	
	ALT7	SRC_BOOT_CFG20	ODE - DISABLED	
	ALT8	EPDC_PWR_CTRL2	SPEED - 100MHZ	
			DSE - 40_OHM	
			SRE - FAST	
EIM_A21	ALT0	EIM_ADDR21	HYS - DISABLED	SW_PAD_CTL_PAD_EIM_ADDR21
	ALT1	IPU1_DISP1_DATA16	PUS - 100K_OHM_PU	
	ALT2	IPU1_CSI1_DATA16	PUE - PULL	
	ALT5	GPIO2_IO17	PKE - ENABLED	
	ALT7	SRC_BOOT_CFG21	ODE - DISABLED	
	ALT8	EPDC_GDCLK	SPEED - 100MHZ	
			DSE - 40_OHM	
			SRE - FAST	
EIM_A22	ALT0	EIM_ADDR22	HYS - DISABLED	SW_PAD_CTL_PAD_EIM_ADDR22
	ALT1	IPU1_DISP1_DATA17	PUS - 100K_OHM_PU	
	ALT2	IPU1_CSI1_DATA17	PUE - PULL	
	ALT5	GPIO2_IO16	PKE - ENABLED	
	ALT7	SRC_BOOT_CFG22	ODE - DISABLED	
	ALT8	EPDC_GDSP	SPEED - 100MHZ	
			DSE - 40_OHM	
			SRE - FAST	
EIM_A23	ALT0	EIM_ADDR23	HYS - DISABLED	SW_PAD_CTL_PAD_EIM_ADDR23
	ALT1	IPU1_DISP1_DATA18	PUS - 100K_OHM_PU	
	ALT2	IPU1_CSI1_DATA18	PUE - PULL	
	ALT4	IPU1_SISG3	PKE - ENABLED	
	ALT5	GPIO6_IO06	ODE - DISABLED	
	ALT7	SRC_BOOT_CFG23	SPEED - 100MHZ	
	ALT8	EPDC_GDOE	DSE - 40_OHM	
			SRE - FAST	
EIM_A24	ALT0	EIM_ADDR24	HYS - DISABLED	SW_PAD_CTL_PAD_EIM_ADDR24
	ALT1	IPU1_DISP1_DATA19	PUS - 100K_OHM_PU	
	ALT2	IPU1_CSI1_DATA19	PUE - PULL	
	ALT4	IPU1_SISG2	PKE - ENABLED	
	ALT5	GPIO5_IO04	ODE - DISABLED	
	ALT7	SRC_BOOT_CFG24	SPEED - 100MHZ	
	ALT8	EPDC_GDRL	DSE - 40_OHM	
			SRE - FAST	
EIM_A25	ALT0	EIM_ADDR25	HYS - DISABLED	SW_PAD_CTL_PAD_EIM_ADDR25

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT1	ECSPI4_SS1	PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	
	ALT2	ECSPI2_RDY		
	ALT3	IPU1_DI1_PIN12		
	ALT4	IPU1_DI0_D1_CS		
	ALT5	GPIO5_IO02		
	ALT6	HDMI_TX_CEC_LINE		
	ALT8	EPDC_DATA15		
	ALT9	EIM_ACLK_FREERUN		
EIM_BCLK	ALT0	EIM_BCLK	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_BCLK
	ALT1	IPU1_DI1_PIN16		
	ALT5	GPIO6_IO31		
	ALT8	EPDC_SDCE9		
EIM_CS0	ALT0	EIM_CS0_B	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_CS0_B
	ALT1	IPU1_DI1_PIN05		
	ALT2	ECSPI2_SCLK		
	ALT5	GPIO2_IO23		
	ALT8	EPDC_DATA06		
EIM_CS1	ALT0	EIM_CS1_B	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_CS1_B
	ALT1	IPU1_DI1_PIN06		
	ALT2	ECSPI2_MOSI		
	ALT5	GPIO2_IO24		
	ALT8	EPDC_DATA08		
EIM_D16	ALT0	EIM_DATA16	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ	SW_PAD_CTL_PAD_EIM_DATA16
	ALT1	ECSPI1_SCLK		
	ALT2	IPU1_DI0_PIN05		
	ALT3	IPU1_CSI1_DATA18		
	ALT4	HDMI_TX_DDC_SDA		
	ALT5	GPIO3_IO16		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT6	I2C2_SDA	DSE - 40_OHM	
	ALT8	EPDC_DATA10	SRE - SLOW	
EIM_D17	ALT0	EIM_DATA17	HYS - ENABLED	SW_PAD_CTL_PAD_EIM_DATA17
	ALT1	ECSPI1_MISO	PUS - 100K_OHM_PU	
	ALT2	IPU1_DI0_PIN06	PUE - PULL	
	ALT3	IPU1_CSI1_PIXCLK	PKE - ENABLED	
	ALT4	DCIC1_OUT	ODE - DISABLED	
	ALT5	GPIO3_IO17	SPEED - 100MHZ	
	ALT6	I2C3_SCL	DSE - 40_OHM	
	ALT8	EPDC_VCOM0	SRE - SLOW	
EIM_D18	ALT0	EIM_DATA18	HYS - ENABLED	SW_PAD_CTL_PAD_EIM_DATA18
	ALT1	ECSPI1_MOSI	PUS - 100K_OHM_PU	
	ALT2	IPU1_DI0_PIN07	PUE - PULL	
	ALT3	IPU1_CSI1_DATA17	PKE - ENABLED	
	ALT4	IPU1_DI1_D0_CS	ODE - DISABLED	
	ALT5	GPIO3_IO18	SPEED - 100MHZ	
	ALT6	I2C3_SDA	DSE - 40_OHM	
	ALT8	EPDC_VCOM1	SRE - SLOW	
EIM_D19	ALT0	EIM_DATA19	HYS - ENABLED	SW_PAD_CTL_PAD_EIM_DATA19
	ALT1	ECSPI1_SS1	PUS - 100K_OHM_PU	
	ALT2	IPU1_DI0_PIN08	PUE - PULL	
	ALT3	IPU1_CSI1_DATA16	PKE - ENABLED	
	ALT4	UART1_CTS_B	ODE - DISABLED	
	ALT5	GPIO3_IO19	SPEED - 100MHZ	
	ALT6	EPIT1_OUT	DSE - 40_OHM	
	ALT8	EPDC_DATA12	SRE - SLOW	
EIM_D20	ALT0	EIM_DATA20	HYS - ENABLED	SW_PAD_CTL_PAD_EIM_DATA20
	ALT1	ECSPI4_SS0	PUS - 100K_OHM_PU	
	ALT2	IPU1_DI0_PIN16	PUE - PULL	
	ALT3	IPU1_CSI1_DATA15	PKE - ENABLED	
	ALT4	UART1_RTS_B	ODE - DISABLED	
	ALT5	GPIO3_IO20	SPEED - 100MHZ	
	ALT6	EPIT2_OUT	DSE - 40_OHM	
			SRE - SLOW	
EIM_D21	ALT0	EIM_DATA21	HYS - ENABLED	SW_PAD_CTL_PAD_EIM_DATA21
	ALT1	ECSPI4_SCLK	PUS - 100K_OHM_PU	
	ALT2	IPU1_DI0_PIN17	PUE - PULL	
	ALT3	IPU1_CSI1_DATA11	PKE - ENABLED	

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT4	USB_OTG_OC	ODE - DISABLED	
	ALT5	GPIO3_IO21	SPEED - 100MHZ	
	ALT6	I2C1_SCL	DSE - 40_OHM	
	ALT7	SPDIF_IN	SRE - SLOW	
EIM_D22	ALT0	EIM_DATA22	HYS - ENABLED	SW_PAD_CTL_PAD_EIM_DATA22
	ALT1	ECSPI4_MISO	PUS - 100K_OHM_PD	
	ALT2	IPU1_DI0_PIN01	PUE - PULL	
	ALT3	IPU1_CSI1_DATA10	PKE - ENABLED	
	ALT4	USB_OTG_PWR	ODE - DISABLED	
	ALT5	GPIO3_IO22	SPEED - 100MHZ	
	ALT6	SPDIF_OUT	DSE - 40_OHM	
	ALT8	EPDC_SDCE6	SRE - SLOW	
EIM_D23	ALT0	EIM_DATA23	HYS - ENABLED	SW_PAD_CTL_PAD_EIM_DATA23
	ALT1	IPU1_DI0_D0_CS	PUS - 100K_OHM_PU	
	ALT2	UART3_CTS_B	PUE - PULL	
	ALT3	UART1_DCD_B	PKE - ENABLED	
	ALT4	IPU1_CSI1_DATA_EN	ODE - DISABLED	
	ALT5	GPIO3_IO23	SPEED - 100MHZ	
	ALT6	IPU1_DI1_PIN02	DSE - 40_OHM	
	ALT7	IPU1_DI1_PIN14	SRE - SLOW	
	ALT8	EPDC_DATA11		
EIM_D24	ALT0	EIM_DATA24	HYS - ENABLED	SW_PAD_CTL_PAD_EIM_DATA24
	ALT1	ECSPI4_SS2	PUS - 100K_OHM_PU	
	ALT2	UART3_TX_DATA	PUE - PULL	
	ALT3	ECSPI1_SS2	PKE - ENABLED	
	ALT4	ECSPI2_SS2	ODE - DISABLED	
	ALT5	GPIO3_IO24	SPEED - 100MHZ	
	ALT6	AUD5_RXFS	DSE - 40_OHM	
	ALT7	UART1_DTR_B	SRE - SLOW	
	ALT8	EPDC_SDCE7		
EIM_D25	ALT0	EIM_DATA25	HYS - ENABLED	SW_PAD_CTL_PAD_EIM_DATA25
	ALT1	ECSPI4_SS3	PUS - 100K_OHM_PU	
	ALT2	UART3_RX_DATA	PUE - PULL	
	ALT3	ECSPI1_SS3	PKE - ENABLED	
	ALT4	ECSPI2_SS3	ODE - DISABLED	
	ALT5	GPIO3_IO25	SPEED - 100MHZ	
	ALT6	AUD5_RXC	DSE - 40_OHM	
	ALT7	UART1_DSR_B	SRE - SLOW	
	ALT8	EPDC_SDCE8		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
EIM_D26	ALT0	EIM_DATA26	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_EIM_DATA26
	ALT1	IPU1_DI1_PIN11		
	ALT2	IPU1_CSI0_DATA01		
	ALT3	IPU1_CSI1_DATA14		
	ALT4	UART2_TX_DATA		
	ALT5	GPIO3_IO26		
	ALT6	IPU1_SISG2		
	ALT7	IPU1_DISP1_DATA22		
	ALT8	EPDC_SDOED		
EIM_D27	ALT0	EIM_DATA27	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_EIM_DATA27
	ALT1	IPU1_DI1_PIN13		
	ALT2	IPU1_CSI0_DATA00		
	ALT3	IPU1_CSI1_DATA13		
	ALT4	UART2_RX_DATA		
	ALT5	GPIO3_IO27		
	ALT6	IPU1_SISG3		
	ALT7	IPU1_DISP1_DATA23		
	ALT8	EPDC_SDOE		
EIM_D28	ALT0	EIM_DATA28	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_EIM_DATA28
	ALT1	I2C1_SDA		
	ALT2	ECSPI4_MOSI		
	ALT3	IPU1_CSI1_DATA12		
	ALT4	UART2_CTS_B		
	ALT5	GPIO3_IO28		
	ALT6	IPU1_EXT_TRIG		
	ALT7	IPU1_DI0_PIN13		
	ALT8	EPDC_PWR_CTRL3		
EIM_D29	ALT0	EIM_DATA29	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_EIM_DATA29
	ALT1	IPU1_DI1_PIN15		
	ALT2	ECSPI4_SS0		
	ALT4	UART2_RTS_B		
	ALT5	GPIO3_IO29		
	ALT6	IPU1_CSI1_VSYNC		
	ALT7	IPU1_DI0_PIN14		
	ALT8	EPDC_PWR_WAKE		
EIM_D30	ALT0	EIM_DATA30	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED	SW_PAD_CTL_PAD_EIM_DATA30
	ALT1	IPU1_DISP1_DATA21		
	ALT2	IPU1_DI0_PIN11		
	ALT3	IPU1_CSI0_DATA03		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT4	UART3_CTS_B	ODE - DISABLED	
	ALT5	GPIO3_IO30	SPEED - 100MHZ	
	ALT6	USB_H1_OC	DSE - 40_OHM	
	ALT8	EPDC_SDOEZ	SRE - SLOW	
EIM_D31	ALT0	EIM_DATA31	HYS - ENABLED	SW_PAD_CTL_PAD_EIM_DATA31
	ALT1	IPU1_DISP1_DATA20	PUS - 100K_OHM_PD	
	ALT2	IPU1_DI0_PIN12	PUE - PULL	
	ALT3	IPU1_CSI0_DATA02	PKE - ENABLED	
	ALT4	UART3_RTS_B	ODE - DISABLED	
	ALT5	GPIO3_IO31	SPEED - 100MHZ	
	ALT6	USB_H1_PWR	DSE - 40_OHM	
	ALT8	EPDC_SDCLK_P	SRE - SLOW	
	ALT9	EIM_ACLK_FREERUN		
EIM_EB0	ALT0	EIM_EB0_B	HYS - DISABLED	SW_PAD_CTL_PAD_EIM_EB0_B
	ALT1	IPU1_DISP1_DATA11	PUS - 100K_OHM_PU	
	ALT2	IPU1_CSI1_DATA11	PUE - PULL	
	ALT4	CCM_PMIC_READY	PKE - ENABLED	
	ALT5	GPIO2_IO28	ODE - DISABLED	
	ALT7	SRC_BOOT_CFG27	SPEED - 100MHZ	
	ALT8	EPDC_PWR_COM	DSE - 40_OHM	
			SRE - FAST	
EIM_EB1	ALT0	EIM_EB1_B	HYS - DISABLED	SW_PAD_CTL_PAD_EIM_EB1_B
	ALT1	IPU1_DISP1_DATA10	PUS - 100K_OHM_PU	
	ALT2	IPU1_CSI1_DATA10	PUE - PULL	
	ALT5	GPIO2_IO29	PKE - ENABLED	
	ALT7	SRC_BOOT_CFG28	ODE - DISABLED	
	ALT8	EPDC_SDSHR	SPEED - 100MHZ	
			DSE - 40_OHM	
			SRE - FAST	
EIM_EB2	ALT0	EIM_EB2_B	HYS - ENABLED	SW_PAD_CTL_PAD_EIM_EB2_B
	ALT1	ECSPI1_SS0	PUS - 100K_OHM_PU	
	ALT3	IPU1_CSI1_DATA19	PUE - PULL	
	ALT4	HDMI_TX_DDC_SCL	PKE - ENABLED	
	ALT5	GPIO2_IO30	ODE - DISABLED	
	ALT6	I2C2_SCL	SPEED - 100MHZ	
	ALT7	SRC_BOOT_CFG30	DSE - 40_OHM	
	ALT8	EPDC_DATA05	SRE - SLOW	
EIM_EB3	ALT0	EIM_EB3_B	HYS - ENABLED	SW_PAD_CTL_PAD_EIM_EB3_B

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT1	ECSPI4_RDY	PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
	ALT2	UART3_RTS_B		
	ALT3	UART1_RI_B		
	ALT4	IPU1_CSI1_HSYNC		
	ALT5	GPIO2_IO31		
	ALT6	IPU1_DI1_PIN03		
	ALT7	SRC_BOOT_CFG31		
	ALT8	EPDC_SDCE0		
	ALT9	EIM_ACLK_FREERUN		
EIM_LBA	ALT0	EIM_LBA_B	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_LBA_B
	ALT1	IPU1_DI1_PIN17		
	ALT2	ECSPI2_SS1		
	ALT5	GPIO2_IO27		
	ALT7	SRC_BOOT_CFG26		
	ALT8	EPDC_DATA04		
EIM_OE	ALT0	EIM_OE_B	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_OE_B
	ALT1	IPU1_DI1_PIN07		
	ALT2	ECSPI2_MISO		
	ALT5	GPIO2_IO25		
	ALT8	EPDC_PWR_IRQ		
EIM_RW	ALT0	EIM_RW	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_EIM_RW
	ALT1	IPU1_DI1_PIN08		
	ALT2	ECSPI2_SS0		
	ALT5	GPIO2_IO26		
	ALT7	SRC_BOOT_CFG29		
	ALT8	EPDC_DATA07		
EIM_WAIT	ALT0	EIM_WAIT_B	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED	SW_PAD_CTL_PAD_EIM_WAIT_B
	ALT1	EIM_DTACK_B		
	ALT5	GPIO5_IO00		
	ALT7	SRC_BOOT_CFG25		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			SPEED - 50MHZ DSE - 60_OHM SRE - SLOW	
ENET_CRS_DV	ALT1	ENET_RX_EN	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_ENET_CRS_DV
	ALT2	ESAI_TX_CLK		
	ALT3	SPDIF_EXT_CLK		
	ALT5	GPIO1_IO25		
ENET_MDC	ALT0	MLB_DATA	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_ENET_MDC
	ALT1	ENET_MDC		
	ALT2	ESAI_TX5_RX0		
	ALT4	ENET_1588_EVENT1_IN		
	ALT5	GPIO1_IO31		
ENET_MDIO	ALT1	ENET_MDIO	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_ENET_MDIO
	ALT2	ESAI_RX_CLK		
	ALT4	ENET_1588_EVENT1_OUT		
	ALT5	GPIO1_IO22		
	ALT6	SPDIF_LOCK		
ENET_REF_CLK	ALT1	ENET_TX_CLK	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_ENET_REF_CLK
	ALT2	ESAI_RX_FS		
	ALT5	GPIO1_IO23		
	ALT6	SPDIF_SR_CLK		
ENET_RXD0	ALT1	ENET_RX_DATA0	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL	SW_PAD_CTL_PAD_ENET_RX_DATA0
	ALT2	ESAI_TX_HF_CLK		
	ALT3	SPDIF_OUT		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT5	GPIO1_IO27	PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
ENET_RXD1	ALT0	MLB_SIG	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_ENET_RX_DATA1
	ALT1	ENET_RX_DATA1		
	ALT2	ESAI_TX_FS		
	ALT4	ENET_1588_EVENT3_OUT		
	ALT5	GPIO1_IO26		
ENET_RX_ER	ALT0	USB_OTG_ID	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_ENET_RX_ER
	ALT1	ENET_RX_ER		
	ALT2	ESAI_RX_HF_CLK		
	ALT3	SPDIF_IN		
	ALT4	ENET_1588_EVENT2_OUT		
	ALT5	GPIO1_IO24		
ENET_TXD0	ALT1	ENET_TX_DATA0	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_ENET_TX_DATA0
	ALT2	ESAI_TX4_RX1		
	ALT5	GPIO1_IO30		
ENET_TXD1	ALT0	MLB_CLK	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_ENET_TX_DATA1
	ALT1	ENET_TX_DATA1		
	ALT2	ESAI_TX2_RX3		
	ALT4	ENET_1588_EVENT0_IN		
	ALT5	GPIO1_IO29		
	ALT9	I2C4_SDA		
ENET_TX_EN	ALT1	ENET_TX_EN	HYS - ENABLED	SW_PAD_CTL_PAD_ENET_TX_EN

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT2	ESAI_TX3_RX2	PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
	ALT5	GPIO1_IO28		
	ALT9	I2C4_SCL		
GPIO_0	ALT0	CCM_CLKO1	HYS - ENABLED PUS - 100K_OHM_PD PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_GPIO00
	ALT2	KEY_COL5		
	ALT3	ASRC_EXT_CLK		
	ALT4	EPIT1_OUT		
	ALT5	GPIO1_IO00		
	ALT6	USB_H1_PWR		
	ALT7	SNVS_VIO_5		
GPIO_1	ALT0	ESAI_RX_CLK	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_GPIO01
	ALT1	WDOG2_B		
	ALT2	KEY_ROW5		
	ALT3	USB_OTG_ID		
	ALT4	PWM2_OUT		
	ALT5	GPIO1_IO01		
	ALT6	SD1_CD_B		
GPIO_2	ALT0	ESAI_TX_FS	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_GPIO02
	ALT2	KEY_ROW6		
	ALT5	GPIO1_IO02		
	ALT6	SD2_WP		
	ALT7	MLB_DATA		
GPIO_3	ALT0	ESAI_RX_HF_CLK	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM	SW_PAD_CTL_PAD_GPIO03
	ALT2	I2C3_SCL		
	ALT3	XTALOSC_REF_CLK_24M		
	ALT4	CCM_CLKO2		
	ALT5	GPIO1_IO03		
	ALT6	USB_H1_OC		
	ALT7	MLB_CLK		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			SRE - SLOW	
GPIO_4	ALT0	ESAI_TX_HF_CLK	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_GPIO04
	ALT2	KEY_COL7		
	ALT5	GPIO1_IO04		
	ALT6	SD2_CD_B		
GPIO_5	ALT0	ESAI_TX2_RX3	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_GPIO05
	ALT2	KEY_ROW7		
	ALT3	CCM_CLKO1		
	ALT5	GPIO1_IO05		
	ALT6	I2C3_SCL		
	ALT7	ARM_EVENTI		
GPIO_6	ALT0	ESAI_TX_CLK	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_GPIO06
	ALT2	I2C3_SDA		
	ALT5	GPIO1_IO06		
	ALT6	SD2_LCTL		
	ALT7	MLB_SIG		
GPIO_7	ALT0	ESAI_TX4_RX1	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_GPIO07
	ALT2	EPIT1_OUT		
	ALT3	FLEXCAN1_TX		
	ALT4	UART2_TX_DATA		
	ALT5	GPIO1_IO07		
	ALT6	SPDIF_LOCK		
	ALT7	USB_OTG_HOST_MODE		
	ALT8	I2C4_SCL		
GPIO_8	ALT0	ESAI_TX5_RX0	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED	SW_PAD_CTL_PAD_GPIO08
	ALT1	XTALOSC_REF_CLK_32K		
	ALT2	EPIT2_OUT		
	ALT3	FLEXCAN1_RX		
	ALT4	UART2_RX_DATA		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT5	GPIO1_IO08	SPEED - 100MHZ	
	ALT6	SPDIF_SR_CLK	DSE - 40_OHM	
	ALT7	USB_OTG_PWR_CTL_WAKE	SRE - SLOW	
	ALT8	I2C4_SDA		
GPIO_9	ALT0	ESAI_RX_FS	HYS - ENABLED	SW_PAD_CTL_PAD_GPIO09
	ALT1	WDOG1_B	PUS - 100K_OHM_PU	
	ALT2	KEY_COL6	PUE - PULL	
	ALT3	CCM_REF_EN_B	PKE - ENABLED	
	ALT4	PWM1_OUT	ODE - DISABLED	
	ALT5	GPIO1_IO09	SPEED - 100MHZ	
	ALT6	SD1_WP	DSE - 40_OHM	
			SRE - SLOW	
GPIO_16	ALT0	ESAI_TX3_RX2	HYS - ENABLED	SW_PAD_CTL_PAD_GPIO16
	ALT1	ENET_1588_EVENT2_IN	PUS - 100K_OHM_PU	
	ALT2	ENET_REF_CLK	PUE - PULL	
	ALT3	SD1_LCTL	PKE - ENABLED	
	ALT4	SPDIF_IN	ODE - DISABLED	
	ALT5	GPIO7_IO11	SPEED - 100MHZ	
	ALT6	I2C3_SDA	DSE - 40_OHM	
	ALT7	JTAG_DE_B	SRE - SLOW	
GPIO_17	ALT0	ESAI_TX0	HYS - ENABLED	SW_PAD_CTL_PAD_GPIO17
	ALT1	ENET_1588_EVENT3_IN	PUS - 100K_OHM_PU	
	ALT2	CCM_PMIC_READY	PUE - PULL	
	ALT3	SDMA_EXT_EVENT0	PKE - ENABLED	
	ALT4	SPDIF_OUT	ODE - DISABLED	
	ALT5	GPIO7_IO12	SPEED - 100MHZ	
			DSE - 40_OHM	
			SRE - SLOW	
GPIO_18	ALT0	ESAI_TX1	HYS - ENABLED	SW_PAD_CTL_PAD_GPIO18
	ALT1	ENET_RX_CLK	PUS - 100K_OHM_PU	
	ALT2	SD3_VSELECT	PUE - PULL	
	ALT3	SDMA_EXT_EVENT1	PKE - ENABLED	
	ALT4	ASRC_EXT_CLK	ODE - DISABLED	
	ALT5	GPIO7_IO13	SPEED - 100MHZ	
	ALT6	SNVS_VIO_5_CTL	DSE - 40_OHM	
			SRE - SLOW	
GPIO_19	ALT0	KEY_COL5	HYS - ENABLED	SW_PAD_CTL_PAD_GPIO19

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT1	ENET_1588_EVENT0_OUT	PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
	ALT2	SPDIF_OUT		
	ALT3	CCM_CLKO1		
	ALT4	ECSPI1_RDY		
	ALT5	GPIO4_IO05		
	ALT6	ENET_TX_ER		
HDMI_CLKM		HDMI_TX_CLK_N		
HDMI_CLKP		HDMI_TX_CLK_P		
HDMI_D0M		HDMI_TX_DATA0_N		
HDMI_D0P		HDMI_TX_DATA0_P		
HDMI_D1M		HDMI_TX_DATA1_N		
HDMI_D1P		HDMI_TX_DATA1_P		
HDMI_D2M		HDMI_TX_DATA2_N		
HDMI_D2P		HDMI_TX_DATA2_P		
HDMI_DDCCEC		HDMI_TX_DDC_CEC		
HDMI_HPD		HDMI_TX_HPD		
JTAG_MOD		JTAG_MOD	HYS - DISABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 50MHZ DSE - 60_OHM SRE - SLOW	SW_PAD_CTL_PAD_JTAG_MOD
JTAG_TCK		JTAG_TCK	HYS - DISABLED PUS - 47K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 50MHZ DSE - 60_OHM SRE - SLOW	SW_PAD_CTL_PAD_JTAG_TCK
JTAG_TDI		JTAG_TDI	HYS - DISABLED PUS - 47K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED	SW_PAD_CTL_PAD_JTAG_TDI

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			SPEED - 50MHZ DSE - 60_OHM SRE - SLOW	
JTAG_TDO		JTAG_TDO	HYS - DISABLED PUS - 100K_OHM_PU PUE - KEEP PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - FAST	SW_PAD_CTL_PAD_JTAG_TDO
JTAG_TMS		JTAG_TMS	HYS - DISABLED PUS - 47K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 50MHZ DSE - 60_OHM SRE - SLOW	SW_PAD_CTL_PAD_JTAG_TMS
JTAG_TRSTB		JTAG_TRSTB	HYS - DISABLED PUS - 47K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 50MHZ DSE - 60_OHM SRE - SLOW	SW_PAD_CTL_PAD_JTAG_TRSTB
KEY_COL0	ALT0	ECSPI1_SCLK	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_KEY_COL0
	ALT1	ENET_RX_DATA3		
	ALT2	AUD5_TXC		
	ALT3	KEY_COL0		
	ALT4	UART4_TX_DATA		
	ALT5	GPIO4_IO06		
	ALT6	DCIC1_OUT		
KEY_COL1	ALT0	ECSPI1_MISO	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL	SW_PAD_CTL_PAD_KEY_COL1
	ALT1	ENET_MDIO		
	ALT2	AUD5_TXFS		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT3	KEY_COL1	PKE - ENABLED	
	ALT4	UART5_TX_DATA	ODE - DISABLED	
	ALT5	GPIO4_IO08	SPEED - 100MHZ	
	ALT6	SD1_VSELECT	DSE - 40_OHM	
			SRE - SLOW	
KEY_COL2	ALT0	ECSPI1_SS1	HYS - ENABLED	SW_PAD_CTL_PAD_KEY_COL2
	ALT1	ENET_RX_DATA2	PUS - 100K_OHM_PU	
	ALT2	FLEXCAN1_TX	PUE - PULL	
	ALT3	KEY_COL2	PKE - ENABLED	
	ALT4	ENET_MDC	ODE - DISABLED	
	ALT5	GPIO4_IO10	SPEED - 100MHZ	
	ALT6	USB_H1_PWR_CTL_WAKE	DSE - 40_OHM	
			SRE - SLOW	
KEY_COL3	ALT0	ECSPI1_SS3	HYS - ENABLED	SW_PAD_CTL_PAD_KEY_COL3
	ALT1	ENET_CRS	PUS - 100K_OHM_PU	
	ALT2	HDMI_TX_DDC_SCL	PUE - PULL	
	ALT3	KEY_COL3	PKE - ENABLED	
	ALT4	I2C2_SCL	ODE - DISABLED	
	ALT5	GPIO4_IO12	SPEED - 100MHZ	
	ALT6	SPDIF_IN	DSE - 40_OHM	
			SRE - SLOW	
KEY_COL4	ALT0	FLEXCAN2_TX	HYS - ENABLED	SW_PAD_CTL_PAD_KEY_COL4
	ALT1	IPU1_SISG4	PUS - 100K_OHM_PU	
	ALT2	USB_OTG_OC	PUE - PULL	
	ALT3	KEY_COL4	PKE - ENABLED	
	ALT4	UART5_RTS_B	ODE - DISABLED	
	ALT5	GPIO4_IO14	SPEED - 100MHZ	
			DSE - 40_OHM	
			SRE - SLOW	
KEY_ROW0	ALT0	ECSPI1_MOSI	HYS - ENABLED	SW_PAD_CTL_PAD_KEY_ROW0
	ALT1	ENET_TX_DATA3	PUS - 100K_OHM_PU	
	ALT2	AUD5_TXD	PUE - PULL	
	ALT3	KEY_ROW0	PKE - ENABLED	
	ALT4	UART4_RX_DATA	ODE - DISABLED	
	ALT5	GPIO4_IO07	SPEED - 100MHZ	
	ALT6	DCIC2_OUT	DSE - 40_OHM	
			SRE - SLOW	
KEY_ROW1	ALT0	ECSPI1_SS0	HYS - ENABLED	SW_PAD_CTL_PAD_KEY_ROW1

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT1	ENET_COL	PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
	ALT2	AUD5_RXD		
	ALT3	KEY_ROW1		
	ALT4	UART5_RX_DATA		
	ALT5	GPIO4_IO09		
	ALT6	SD2_VSELECT		
KEY_ROW2	ALT0	ECSPI1_SS2	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_KEY_ROW2
	ALT1	ENET_TX_DATA2		
	ALT2	FLEXCAN1_RX		
	ALT3	KEY_ROW2		
	ALT4	SD2_VSELECT		
	ALT5	GPIO4_IO11		
	ALT6	HDMI_TX_CEC_LINE		
KEY_ROW3	ALT1	ASRC_EXT_CLK	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_KEY_ROW3
	ALT2	HDMI_TX_DDC_SDA		
	ALT3	KEY_ROW3		
	ALT4	I2C2_SDA		
	ALT5	GPIO4_IO13		
	ALT6	SD1_VSELECT		
KEY_ROW4	ALT0	FLEXCAN2_RX	HYS - ENABLED PUS - 100K_OHM_PD PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_KEY_ROW4
	ALT1	IPU1_SISG5		
	ALT2	USB_OTG_PWR		
	ALT3	KEY_ROW4		
	ALT4	UART5_CTS_B		
	ALT5	GPIO4_IO15		
LVDS0_CLK_N		LVDS0_CLK_N		
LVDS0_CLK_P		LVDS0_CLK_P		
LVDS0_TX0_N		LVDS0_DATA0_N		
LVDS0_TX0_P		LVDS0_DATA0_P		
LVDS0_TX1_N		LVDS0_DATA1_N		
LVDS0_TX1_P		LVDS0_DATA1_P		
LVDS0_TX2_N		LVDS0_DATA2_N		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
LVDS0_TX2_P		LVDS0_DATA2_P		
LVDS0_TX3_N		LVDS0_DATA3_N		
LVDS0_TX3_P		LVDS0_DATA3_P		
LVDS1_CLK_N		LVDS1_CLK_N		
LVDS1_CLK_P		LVDS1_CLK_P		
LVDS1_TX0_N		LVDS1_DATA0_N		
LVDS1_TX0_P		LVDS1_DATA0_P		
LVDS1_TX1_N		LVDS1_DATA1_N		
LVDS1_TX1_P		LVDS1_DATA1_P		
LVDS1_TX2_N		LVDS1_DATA2_N		
LVDS1_TX2_P		LVDS1_DATA2_P		
LVDS1_TX3_N		LVDS1_DATA3_N		
LVDS1_TX3_P		LVDS1_DATA3_P		
MLB_CN		MLB_CLK_N		
MLB_CP		MLB_CLK_P		
MLB_DN		MLB_DATA_N		
MLB_DP		MLB_DATA_P		
MLB_SN		MLB_SIG_N		
MLB_SP		MLB_SIG_P		
NANDF_ALE	ALT0	NAND_ALE	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_NAND_ALE
	ALT1	SD4_RESET		
	ALT5	GPIO6_IO08		
NANDF_CLE	ALT0	NAND_CLE	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_NAND_CLE
	ALT5	GPIO6_IO07		
NANDF_CS0	ALT0	NAND_CE0_B	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL	SW_PAD_CTL_PAD_NAND_CS0_B
	ALT5	GPIO6_IO11		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
NANDF_CS1	ALT0	NAND_CE1_B	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_NAND_CS1_B
	ALT1	SD4_VSELECT		
	ALT2	SD3_VSELECT		
	ALT5	GPIO6_IO14		
NANDF_CS2	ALT0	NAND_CE2_B	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_NAND_CS2_B
	ALT1	IPU1_SISG0		
	ALT2	ESAI_TX0		
	ALT3	EIM_CRE		
	ALT4	CCM_CLKO2		
	ALT5	GPIO6_IO15		
NANDF_CS3	ALT0	NAND_CE3_B	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_NAND_CS3_B
	ALT1	IPU1_SISG1		
	ALT2	ESAI_TX1		
	ALT3	EIM_ADDR26		
	ALT5	GPIO6_IO16		
	ALT9	I2C4_SDA		
NANDF_D0	ALT0	NAND_DATA00	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_NAND_DATA00
	ALT1	SD1_DATA4		
	ALT5	GPIO2_IO00		
NANDF_D1	ALT0	NAND_DATA01	HYS - ENABLED	SW_PAD_CTL_PAD_NAND_DATA01

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT1	SD1_DATA5	PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
	ALT5	GPIO2_IO01		
NANDF_D2	ALT0	NAND_DATA02	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_NAND_DATA02
	ALT1	SD1_DATA6		
	ALT5	GPIO2_IO02		
NANDF_D3	ALT0	NAND_DATA03	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_NAND_DATA03
	ALT1	SD1_DATA7		
	ALT5	GPIO2_IO03		
NANDF_D4	ALT0	NAND_DATA04	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_NAND_DATA04
	ALT1	SD2_DATA4		
	ALT5	GPIO2_IO04		
NANDF_D5	ALT0	NAND_DATA05	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM	SW_PAD_CTL_PAD_NAND_DATA05
	ALT1	SD2_DATA5		
	ALT5	GPIO2_IO05		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			SRE - SLOW	
NANDF_D6	ALT0	NAND_DATA06	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_NAND_DATA06
	ALT1	SD2_DATA6		
	ALT5	GPIO2_IO06		
NANDF_D7	ALT0	NAND_DATA07	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_NAND_DATA07
	ALT1	SD2_DATA7		
	ALT5	GPIO2_IO07		
NANDF_RB0	ALT0	NAND_READY_B	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_NAND_READY_B
	ALT5	GPIO6_IO10		
NANDF_WP_B	ALT0	NAND_WP_B	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_NAND_WP_B
	ALT5	GPIO6_IO09		
	ALT9	I2C4_SCL		
ONOFF		SRC_ONOFF		
PCIE_RXM		PCIE_RX_N		
PCIE_RXP		PCIE_RX_P		
PCIE_TXM		PCIE_TX_N		
PCIE_TXP		PCIE_TX_P		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
PMIC_ON_REQ		SNVS_PMIC_ON_REQ		
PMIC_STBY_REQ		CCM_PMIC_STBY_REQ		
POR_B		SRC_POR_B		
RGMII_RD0	ALT0	HSI_RX_READY	DDR_SEL - 1P2V_IO	SW_PAD_CTL_PAD_RGMII_RD0
	ALT1	RGMII_RD0	DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE_RGMII
	ALT5	GPIO6_IO25	HYS - ENABLED	SW_PAD_CTL_GRP_RGMII_TERM
			PUS - 100K_OHM_PU	
			PUE - PULL	
			PKE - ENABLED	
			ODT - DISABLED	
			DSE - 37_OHM	
RGMII_RD1	ALT0	HSI_TX_FLAG	DDR_SEL - 1P2V_IO	SW_PAD_CTL_PAD_RGMII_RD1
	ALT1	RGMII_RD1	DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE_RGMII
	ALT5	GPIO6_IO27	HYS - ENABLED	SW_PAD_CTL_GRP_RGMII_TERM
			PUS - 100K_OHM_PU	
			PUE - PULL	
			PKE - ENABLED	
			ODT - DISABLED	
			DSE - 37_OHM	
RGMII_RD2	ALT0	HSI_TX_DATA	DDR_SEL - 1P2V_IO	SW_PAD_CTL_PAD_RGMII_RD2
	ALT1	RGMII_RD2	DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE_RGMII
	ALT5	GPIO6_IO28	HYS - ENABLED	SW_PAD_CTL_GRP_RGMII_TERM
			PUS - 100K_OHM_PU	
			PUE - PULL	
			PKE - ENABLED	
			ODT - DISABLED	
			DSE - 37_OHM	
RGMII_RD3	ALT0	HSI_TX_WAKE	DDR_SEL - 1P2V_IO	SW_PAD_CTL_PAD_RGMII_RD3
	ALT1	RGMII_RD3	DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE_RGMII
	ALT5	GPIO6_IO29	HYS - ENABLED	SW_PAD_CTL_GRP_RGMII_TERM
			PUS - 100K_OHM_PU	
			PUE - PULL	
			PKE - ENABLED	
			ODT - DISABLED	
			DSE - 37_OHM	
RGMII_RXC	ALT0	USB_H3_STROBE	DDR_SEL - 1P2V_IO	SW_PAD_CTL_PAD_RGMII_RXC
	ALT1	RGMII_RXC	DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE_RGMII
	ALT5	GPIO6_IO30	HYS - ENABLED	SW_PAD_CTL_GRP_RGMII_TERM

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			PUS - 100K_OHM_PD PUE - PULL PKE - ENABLED ODT - DISABLED DSE - 37_OHM	
RGMII_RX_CTL	ALT0	USB_H3_DATA	DDR_SEL - 1P2V_IO DDR_INPUT - CMOS HYS - ENABLED PUS - 100K_OHM_PD PUE - PULL PKE - ENABLED ODT - DISABLED DSE - 37_OHM	SW_PAD_CTL_PAD_RGMII_RX_CTL SW_PAD_CTL_GRP_DDR_TYPE_RGMII SW_PAD_CTL_GRP_RGMII_TERM
	ALT1	RGMII_RX_CTL		
	ALT5	GPIO6_IO24		
RGMII_TD0	ALT0	HSI_TX_READY	DDR_SEL - 1P2V_IO DDR_INPUT - CMOS HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODT - DISABLED DSE - 37_OHM	SW_PAD_CTL_PAD_RGMII_TD0 SW_PAD_CTL_GRP_DDR_TYPE_RGMII
	ALT1	RGMII_TD0		
	ALT5	GPIO6_IO20		
RGMII_TD1	ALT0	HSI_RX_FLAG	DDR_SEL - 1P2V_IO DDR_INPUT - CMOS HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODT - DISABLED DSE - 37_OHM	SW_PAD_CTL_PAD_RGMII_TD1 SW_PAD_CTL_GRP_DDR_TYPE_RGMII
	ALT1	RGMII_TD1		
	ALT5	GPIO6_IO21		
RGMII_TD2	ALT0	HSI_RX_DATA	DDR_SEL - 1P2V_IO DDR_INPUT - CMOS HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODT - DISABLED DSE - 37_OHM	SW_PAD_CTL_PAD_RGMII_TD2 SW_PAD_CTL_GRP_DDR_TYPE_RGMII
	ALT1	RGMII_TD2		
	ALT5	GPIO6_IO22		
RGMII_TD3	ALT0	HSI_RX_WAKE	DDR_SEL - 1P2V_IO	SW_PAD_CTL_PAD_RGMII_TD3

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT1	RGMII_TD3	DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE_RGMII
	ALT5	GPIO6_IO23	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODT - DISABLED DSE - 37_OHM	
RGMII_TXC	ALT0	USB_H2_DATA	DDR_SEL - 1P2V_IO	SW_PAD_CTL_PAD_RGMII_TXC
	ALT1	RGMII_TXC	DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE_RGMII
	ALT2	SPDIF_EXT_CLK	HYS - ENABLED	
	ALT5	GPIO6_IO19	PUS - 100K_OHM_PD	
	ALT7	XTALOSC_REF_CLK_24M	PUE - PULL PKE - ENABLED ODT - DISABLED DSE - 37_OHM	
RGMII_TX_CTL	ALT0	USB_H2_STROBE	DDR_SEL - 1P2V_IO	SW_PAD_CTL_PAD_RGMII_TX_CTL
	ALT1	RGMII_TX_CTL	DDR_INPUT - CMOS	SW_PAD_CTL_GRP_DDR_TYPE_RGMII
	ALT5	GPIO6_IO26	HYS - ENABLED	
	ALT7	ENET_REF_CLK	PUS - 100K_OHM_PD PUE - PULL PKE - ENABLED ODT - DISABLED DSE - 37_OHM	
RTC_XTALI		XTALOSC_RTC_XTALI		
RTC_XTALO		XTALOSC_RTC_XTALO		
SD1_CLK	ALT0	SD1_CLK	HYS - ENABLED	SW_PAD_CTL_PAD_SD1_CLK
	ALT3	GPT_CLKIN	PUS - 100K_OHM_PU	
	ALT5	GPIO1_IO20	PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
SD1_CMD	ALT0	SD1_CMD	HYS - ENABLED	SW_PAD_CTL_PAD_SD1_CMD
	ALT2	PWM4_OUT	PUS - 100K_OHM_PU	
	ALT3	GPT_COMPARE1	PUE - PULL	
	ALT5	GPIO1_IO18	PKE - ENABLED ODE - DISABLED	

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
SD1_DAT0	ALT0	SD1_DATA0	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD1_DATA0
	ALT3	GPT_CAPTURE1		
	ALT5	GPIO1_IO16		
SD1_DAT1	ALT0	SD1_DATA1	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD1_DATA1
	ALT2	PWM3_OUT		
	ALT3	GPT_CAPTURE2		
	ALT5	GPIO1_IO17		
SD1_DAT2	ALT0	SD1_DATA2	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD1_DATA2
	ALT2	GPT_COMPARE2		
	ALT3	PWM2_OUT		
	ALT4	WDOG1_B		
	ALT5	GPIO1_IO19		
	ALT6	WDOG1_RESET_B_DEB		
SD1_DAT3	ALT0	SD1_DATA3	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD1_DATA3
	ALT2	GPT_COMPARE3		
	ALT3	PWM1_OUT		
	ALT4	WDOG2_B		
	ALT5	GPIO1_IO21		
	ALT6	WDOG2_RESET_B_DEB		
SD2_CLK	ALT0	SD2_CLK	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL	SW_PAD_CTL_PAD_SD2_CLK
	ALT2	KEY_COL5		
	ALT3	AUD4_RXFS		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT5	GPIO1_IO10	PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
SD2_CMD	ALT0	SD2_CMD	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD2_CMD
	ALT2	KEY_ROW5		
	ALT3	AUD4_RXC		
	ALT5	GPIO1_IO11		
SD2_DAT0	ALT0	SD2_DATA0	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD2_DATA0
	ALT3	AUD4_RXD		
	ALT4	KEY_ROW7		
	ALT5	GPIO1_IO15		
	ALT6	DCIC2_OUT		
SD2_DAT1	ALT0	SD2_DATA1	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD2_DATA1
	ALT2	EIM_CS2_B		
	ALT3	AUD4_TXFS		
	ALT4	KEY_COL7		
	ALT5	GPIO1_IO14		
SD2_DAT2	ALT0	SD2_DATA2	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD2_DATA2
	ALT2	EIM_CS3_B		
	ALT3	AUD4_TXD		
	ALT4	KEY_ROW6		
	ALT5	GPIO1_IO13		
SD2_DAT3	ALT0	SD2_DATA3	HYS - ENABLED	SW_PAD_CTL_PAD_SD2_DATA3

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT2	KEY_COL6	PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
	ALT3	AUD4_TXC		
	ALT5	GPIO1_IO12		
SD3_CLK	ALT0	SD3_CLK	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD3_CLK
	ALT1	UART2_RTS_B		
	ALT2	FLEXCAN1_RX		
	ALT5	GPIO7_IO03		
SD3_CMD	ALT0	SD3_CMD	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD3_CMD
	ALT1	UART2_CTS_B		
	ALT2	FLEXCAN1_TX		
	ALT5	GPIO7_IO02		
SD3_DAT0	ALT0	SD3_DATA0	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD3_DATA0
	ALT1	UART1_CTS_B		
	ALT2	FLEXCAN2_TX		
	ALT5	GPIO7_IO04		
SD3_DAT1	ALT0	SD3_DATA1	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM	SW_PAD_CTL_PAD_SD3_DATA1
	ALT1	UART1_RTS_B		
	ALT2	FLEXCAN2_RX		
	ALT5	GPIO7_IO05		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			SRE - SLOW	
SD3_DAT2	ALT0	SD3_DATA2	HYS - ENABLED	SW_PAD_CTL_PAD_SD3_DATA2
	ALT5	GPIO7_IO06	PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
SD3_DAT3	ALT0	SD3_DATA3	HYS - ENABLED	SW_PAD_CTL_PAD_SD3_DATA3
	ALT1	UART3_CTS_B	PUS - 100K_OHM_PU	
	ALT5	GPIO7_IO07	PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
SD3_DAT4	ALT0	SD3_DATA4	HYS - ENABLED	SW_PAD_CTL_PAD_SD3_DATA4
	ALT1	UART2_RX_DATA	PUS - 100K_OHM_PU	
	ALT5	GPIO7_IO01	PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
SD3_DAT5	ALT0	SD3_DATA5	HYS - ENABLED	SW_PAD_CTL_PAD_SD3_DATA5
	ALT1	UART2_TX_DATA	PUS - 100K_OHM_PU	
	ALT5	GPIO7_IO00	PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
SD3_DAT6	ALT0	SD3_DATA6	HYS - ENABLED	SW_PAD_CTL_PAD_SD3_DATA6
	ALT1	UART1_RX_DATA	PUS - 100K_OHM_PU	
	ALT5	GPIO6_IO18	PUE - PULL PKE - ENABLED ODE - DISABLED	

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
SD3_DAT7	ALT0	SD3_DATA7	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD3_DATA7
	ALT1	UART1_TX_DATA		
	ALT5	GPIO6_IO17		
SD3_RST	ALT0	SD3_RESET	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD3_RESET
	ALT1	UART3_RTS_B		
	ALT5	GPIO7_IO08		
SD4_CLK	ALT0	SD4_CLK	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD4_CLK
	ALT1	NAND_WE_B		
	ALT2	UART3_RX_DATA		
	ALT5	GPIO7_IO10		
SD4_CMD	ALT0	SD4_CMD	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD4_CMD
	ALT1	NAND_RE_B		
	ALT2	UART3_TX_DATA		
	ALT5	GPIO7_IO09		
SD4_DAT0	ALT1	SD4_DATA0	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL	SW_PAD_CTL_PAD_SD4_DATA0
	ALT2	NAND_DQS		
	ALT5	GPIO2_IO08		

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
			PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
SD4_DAT1	ALT1	SD4_DATA1	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD4_DATA1
	ALT2	PWM3_OUT		
	ALT5	GPIO2_IO09		
SD4_DAT2	ALT1	SD4_DATA2	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD4_DATA2
	ALT2	PWM4_OUT		
	ALT5	GPIO2_IO10		
SD4_DAT3	ALT1	SD4_DATA3	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD4_DATA3
	ALT5	GPIO2_IO11		
SD4_DAT4	ALT1	SD4_DATA4	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD4_DATA4
	ALT2	UART2_RX_DATA		
	ALT5	GPIO2_IO12		
SD4_DAT5	ALT1	SD4_DATA5	HYS - ENABLED	SW_PAD_CTL_PAD_SD4_DATA5

Table continues on the next page...

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT2	UART2_RTS_B	PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	
	ALT5	GPIO2_IO13		
SD4_DAT6	ALT1	SD4_DATA6	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD4_DATA6
	ALT2	UART2_CTS_B		
	ALT5	GPIO2_IO14		
SD4_DAT7	ALT1	SD4_DATA7	HYS - ENABLED PUS - 100K_OHM_PU PUE - PULL PKE - ENABLED ODE - DISABLED SPEED - 100MHZ DSE - 40_OHM SRE - SLOW	SW_PAD_CTL_PAD_SD4_DATA7
	ALT2	UART2_TX_DATA		
	ALT5	GPIO2_IO15		
TAMPER		SNVS_TAMPER		
TEST_MODE		TCU_TEST_MODE		
USB_H1_DN		USB_H1_DN		
USB_H1_DP		USB_H1_DP		
USB_H1_VBUS		USB_H1_VBUS		
USB_OTG_CHD_B		USB_OTG_CHD_B		
USB_OTG_DN		USB_OTG_DN		
USB_OTG_DP		USB_OTG_DP		
USB_OTG_VBUS		USB_OTG_VBUS		
XTALI		XTALOSC_XTALI		
XTALO		XTALOSC_XTALO		

4.1.2 Muxing Options

An additional view of external signals muxing is shown by the presentation of the muxing options per block/instance.

Table 4-2. Muxing Options

Signal	Pad (Mode)	Mux/Input Select Registers
ARM - ARM Platform		
ARM_EVENTI	GPIO_5 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_GPIO05
ARM_EVENTO	CSI0_PIXCLK (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_PIXCLK
ARM_TRACE00	CSI0_VSYNC (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_VSYNC
ARM_TRACE01	CSI0_DAT4 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA04
ARM_TRACE02	CSI0_DAT5 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA05
ARM_TRACE03	CSI0_DAT6 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA06
ARM_TRACE04	CSI0_DAT7 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA07
ARM_TRACE05	CSI0_DAT8 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA08
ARM_TRACE06	CSI0_DAT9 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA09
ARM_TRACE07	CSI0_DAT10 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA10
ARM_TRACE08	CSI0_DAT11 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA11
ARM_TRACE09	CSI0_DAT12 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA12
ARM_TRACE10	CSI0_DAT13 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA13
ARM_TRACE11	CSI0_DAT14 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA14
ARM_TRACE12	CSI0_DAT15 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA15
ARM_TRACE13	CSI0_DAT16 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA16
ARM_TRACE14	CSI0_DAT17 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA17
ARM_TRACE15	CSI0_DAT18 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA18
ARM_TRACE_CLK	CSI0_DATA_EN (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA_EN
ARM_TRACE_CTL	CSI0_MCLK (ALT7)	IOMUXC_SW_MUX_CTL_PAD_CSI0_HSYNC
ASRC - Asynchronous Sample Rate Converter		
ASRC_EXT_CLK	GPIO_0 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_GPIO00 IOMUXC_ASRC_ASRCK_CLOCK_6_SELECT_INPUT
	GPIO_18 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_GPIO18 IOMUXC_ASRC_ASRCK_CLOCK_6_SELECT_INPUT
	KEY_ROW3 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW3 IOMUXC_ASRC_ASRCK_CLOCK_6_SELECT_INPUT
AUDMUX - Digital Audio Multiplexer		
AUD3_RXC	CSI0_DAT10 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA10
AUD3_RXD	CSI0_DAT7 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA07
AUD3_RXFS	CSI0_DAT11 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA11
AUD3_TXC	CSI0_DAT4 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA04
AUD3_TXD	CSI0_DAT5 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA05
AUD3_TXFS	CSI0_DAT6 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA06

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
AUD4_RXC	DISP0_DAT19 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA19
		IOMUXC_AUD4_INPUT_RXCLK_AMX_SELECT_INPUT
	SD2_CMD (ALT3)	IOMUXC_SW_MUX_CTL_PAD_SD2_CMD
		IOMUXC_AUD4_INPUT_RXCLK_AMX_SELECT_INPUT
AUD4_RXD	DISP0_DAT23 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA23
		IOMUXC_AUD4_INPUT_DA_AMX_SELECT_INPUT
	SD2_DAT0 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0
		IOMUXC_AUD4_INPUT_DA_AMX_SELECT_INPUT
AUD4_RXFS	DISP0_DAT18 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA18
		IOMUXC_AUD4_INPUT_RXFS_AMX_SELECT_INPUT
	SD2_CLK (ALT3)	IOMUXC_SW_MUX_CTL_PAD_SD2_CLK
		IOMUXC_AUD4_INPUT_RXFS_AMX_SELECT_INPUT
AUD4_TXC	DISP0_DAT20 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA20
		IOMUXC_AUD4_INPUT_TXCLK_AMX_SELECT_INPUT
	SD2_DAT3 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3
		IOMUXC_AUD4_INPUT_TXCLK_AMX_SELECT_INPUT
AUD4_TXD	DISP0_DAT21 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA21
		IOMUXC_AUD4_INPUT_DB_AMX_SELECT_INPUT
	SD2_DAT2 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2
		IOMUXC_AUD4_INPUT_DB_AMX_SELECT_INPUT
AUD4_TXFS	DISP0_DAT22 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA22
		IOMUXC_AUD4_INPUT_TXFS_AMX_SELECT_INPUT
	SD2_DAT1 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1
		IOMUXC_AUD4_INPUT_TXFS_AMX_SELECT_INPUT
AUD5_RXC	DISP0_DAT14 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA14
		IOMUXC_AUD5_INPUT_RXCLK_AMX_SELECT_INPUT
	EIM_D25 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA25
		IOMUXC_AUD5_INPUT_RXCLK_AMX_SELECT_INPUT
AUD5_RXD	DISP0_DAT19 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA19
		IOMUXC_AUD5_INPUT_DA_AMX_SELECT_INPUT
	KEY_ROW1 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW1
		IOMUXC_AUD5_INPUT_DA_AMX_SELECT_INPUT
AUD5_RXFS	DISP0_DAT13 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA13
		IOMUXC_AUD5_INPUT_RXFS_AMX_SELECT_INPUT
	EIM_D24 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA24
		IOMUXC_AUD5_INPUT_RXFS_AMX_SELECT_INPUT
AUD5_TXC	DISP0_DAT16 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA16
		IOMUXC_AUD5_INPUT_TXCLK_AMX_SELECT_INPUT

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
	KEY_COL0 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL0
		IOMUXC_AUD5_INPUT_TXCLK_AMX_SELECT_INPUT
AUD5_TXD	DISP0_DAT17 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA17
		IOMUXC_AUD5_INPUT_DB_AMX_SELECT_INPUT
	KEY_ROW0 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW0
		IOMUXC_AUD5_INPUT_DB_AMX_SELECT_INPUT
AUD5_TXFS	DISP0_DAT18 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA18
		IOMUXC_AUD5_INPUT_TXFS_AMX_SELECT_INPUT
	KEY_COL1 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL1
		IOMUXC_AUD5_INPUT_TXFS_AMX_SELECT_INPUT
AUD6_RXC	DISP0_DAT6 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA06
AUD6_RXD	DI0_PIN4 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN04
AUD6_RXFS	DISP0_DAT5 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA05
AUD6_TXC	DI0_PIN15 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN15
AUD6_TXD	DI0_PIN2 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN02
AUD6_TXFS	DI0_PIN3 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN03
CCM - Clock Controller Module		
CCM_CLKO1	CSI0_MCLK (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_HSYNC
	GPIO_0 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_GPIO00
	GPIO_5 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_GPIO05
	GPIO_19 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_GPIO19
CCM_CLKO2	GPIO_3 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_GPIO03
	NANDF_CS2 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS2_B
CCM_PMIC_READY	EIM_EB0 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB0_B
		IOMUXC_CCM_PMIC_READY_SELECT_INPUT
	GPIO_17 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_GPIO17
		IOMUXC_CCM_PMIC_READY_SELECT_INPUT
CCM_PMIC_STBY_REQ	PMIC_STBY_REQ	Not multiplexed.
CCM_REF_EN_B	GPIO_9 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_GPIO09
DCIC1 - Display Content Integrity Checker		
DCIC1_OUT	EIM_D17 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA17
	KEY_COL0 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL0
DCIC2 - Display Content Integrity Checker		
DCIC2_OUT	KEY_ROW0 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW0
	SD2_DAT0 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0
ECSPI1 - Enhanced Configurable SPI		
ECSPI1_MISO	CSI0_DAT6 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA06
		IOMUXC_ECSPI1_MISO_SELECT_INPUT
	DISP0_DAT22 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA22

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
		IOMUXC_ECSPI1_MISO_SELECT_INPUT
	EIM_D17 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA17
		IOMUXC_ECSPI1_MISO_SELECT_INPUT
	KEY_COL1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL1
		IOMUXC_ECSPI1_MISO_SELECT_INPUT
ECSPI1_MOSI	CSI0_DAT5 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA05
		IOMUXC_ECSPI1_MOSI_SELECT_INPUT
	DISP0_DAT21 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA21
		IOMUXC_ECSPI1_MOSI_SELECT_INPUT
	EIM_D18 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA18
		IOMUXC_ECSPI1_MOSI_SELECT_INPUT
	KEY_ROW0 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW0
		IOMUXC_ECSPI1_MOSI_SELECT_INPUT
ECSPI1_RDY	GPIO_19 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_GPIO19
ECSPI1_SCLK	CSI0_DAT4 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA04
		IOMUXC_ECSPI1_CSPI_CLK_IN_SELECT_INPUT
	DISP0_DAT20 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA20
		IOMUXC_ECSPI1_CSPI_CLK_IN_SELECT_INPUT
	EIM_D16 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA16
		IOMUXC_ECSPI1_CSPI_CLK_IN_SELECT_INPUT
	KEY_COL0 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL0
		IOMUXC_ECSPI1_CSPI_CLK_IN_SELECT_INPUT
ECSPI1_SS0	CSI0_DAT7 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA07
		IOMUXC_ECSPI1_SS0_SELECT_INPUT
	DISP0_DAT23 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA23
		IOMUXC_ECSPI1_SS0_SELECT_INPUT
	EIM_EB2 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB2_B
		IOMUXC_ECSPI1_SS0_SELECT_INPUT
	KEY_ROW1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW1
		IOMUXC_ECSPI1_SS0_SELECT_INPUT
ECSPI1_SS1	DISP0_DAT15 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA15
		IOMUXC_ECSPI1_SS1_SELECT_INPUT
	EIM_D19 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA19
		IOMUXC_ECSPI1_SS1_SELECT_INPUT
	KEY_COL2 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL2
		IOMUXC_ECSPI1_SS1_SELECT_INPUT
ECSPI1_SS2	EIM_D24 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA24
		IOMUXC_ECSPI1_SS2_SELECT_INPUT

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
	KEY_ROW2 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW2
		IOMUXC_ECSPI1_SS2_SELECT_INPUT
ECSPI1_SS3	EIM_D25 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA25
		IOMUXC_ECSPI1_SS3_SELECT_INPUT
	KEY_COL3 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL3
		IOMUXC_ECSPI1_SS3_SELECT_INPUT
ECSPI2 - Enhanced Configurable SPI		
ECSPI2_MISO	CSI0_DAT10 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA10
		IOMUXC_ECSPI2_MISO_SELECT_INPUT
	DISP0_DAT17 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA17
		IOMUXC_ECSPI2_MISO_SELECT_INPUT
	EIM_OE (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_OE_B
		IOMUXC_ECSPI2_MISO_SELECT_INPUT
ECSPI2_MOSI	CSI0_DAT9 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA09
		IOMUXC_ECSPI2_MOSI_SELECT_INPUT
	DISP0_DAT16 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA16
		IOMUXC_ECSPI2_MOSI_SELECT_INPUT
	EIM_CS1 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_CS1_B
		IOMUXC_ECSPI2_MOSI_SELECT_INPUT
ECSPI2_RDY	EIM_A25 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR25
ECSPI2_SCLK	CSI0_DAT8 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA08
		IOMUXC_ECSPI2_CSPI_CLK_IN_SELECT_INPUT
	DISP0_DAT19 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA19
		IOMUXC_ECSPI2_CSPI_CLK_IN_SELECT_INPUT
	EIM_CS0 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_CS0_B
		IOMUXC_ECSPI2_CSPI_CLK_IN_SELECT_INPUT
ECSPI2_SS0	CSI0_DAT11 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA11
		IOMUXC_ECSPI2_SS0_SELECT_INPUT
	DISP0_DAT18 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA18
		IOMUXC_ECSPI2_SS0_SELECT_INPUT
	EIM_RW (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_RW
		IOMUXC_ECSPI2_SS0_SELECT_INPUT
ECSPI2_SS1	DISP0_DAT15 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA15
		IOMUXC_ECSPI2_SS1_SELECT_INPUT
	EIM_LBA (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_LBA_B
		IOMUXC_ECSPI2_SS1_SELECT_INPUT
ECSPI2_SS2	EIM_D24 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA24
ECSPI2_SS3	EIM_D25 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA25

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
ECSPI3 - Enhanced Configurable SPI		
ECSPI3_MISO	DISP0_DAT2 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA02
ECSPI3_MOSI	DISP0_DAT1 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA01
ECSPI3_RDY	DISP0_DAT7 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA07
ECSPI3_SCLK	DISP0_DAT0 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA00
ECSPI3_SS0	DISP0_DAT3 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA03
ECSPI3_SS1	DISP0_DAT4 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA04
ECSPI3_SS2	DISP0_DAT5 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA05
ECSPI3_SS3	DISP0_DAT6 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA06
ECSPI4 - Enhanced Configurable SPI		
ECSPI4_MISO	EIM_D22 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA22
ECSPI4_MOSI	EIM_D28 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA28
ECSPI4_RDY	EIM_EB3 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB3_B
ECSPI4_SCLK	EIM_D21 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA21
ECSPI4_SS0	EIM_D20 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA20 IOMUXC_ECSPI4_SS0_SELECT_INPUT
	EIM_D29 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA29 IOMUXC_ECSPI4_SS0_SELECT_INPUT
ECSPI4_SS1	EIM_A25 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR25
ECSPI4_SS2	EIM_D24 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA24
ECSPI4_SS3	EIM_D25 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA25
EIM - External Interface Module		
EIM_ACLK_FREERUN	EIM_A25 (ALT9)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR25
	EIM_D31 (ALT9)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA31
	EIM_EB3 (ALT9)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB3_B
EIM_AD00	EIM_DA0 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD00
EIM_AD01	EIM_DA1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD01
EIM_AD02	EIM_DA2 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD02
EIM_AD03	EIM_DA3 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD03
EIM_AD04	EIM_DA4 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD04
EIM_AD05	EIM_DA5 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD05
EIM_AD06	EIM_DA6 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD06
EIM_AD07	EIM_DA7 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD07
EIM_AD08	EIM_DA8 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD08
EIM_AD09	EIM_DA9 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD09
EIM_AD10	EIM_DA10 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD10
EIM_AD11	EIM_DA11 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD11
EIM_AD12	EIM_DA12 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD12
EIM_AD13	EIM_DA13 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD13

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
EIM_AD14	EIM_DA14 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD14
EIM_AD15	EIM_DA15 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD15
EIM_ADDR16	EIM_A16 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR16
EIM_ADDR17	EIM_A17 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR17
EIM_ADDR18	EIM_A18 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR18
EIM_ADDR19	EIM_A19 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR19
EIM_ADDR20	EIM_A20 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR20
EIM_ADDR21	EIM_A21 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR21
EIM_ADDR22	EIM_A22 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR22
EIM_ADDR23	EIM_A23 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR23
EIM_ADDR24	EIM_A24 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR24
EIM_ADDR25	EIM_A25 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR25
EIM_ADDR26	NANDF_CS3 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS3_B
EIM_BCLK	EIM_BCLK (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_BCLK
EIM_CRE	NANDF_CS2 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS2_B
EIM_CS0_B	EIM_CS0 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_CS0_B
EIM_CS1_B	EIM_CS1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_CS1_B
EIM_CS2_B	DISP0_DAT18 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA18
	SD2_DAT1 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1
EIM_CS3_B	DISP0_DAT19 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA19
	SD2_DAT2 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2
EIM_DATA00	CSI0_DATA_EN (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA_EN
EIM_DATA01	CSI0_VSYNC (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_VSYNC
EIM_DATA02	CSI0_DAT4 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA04
EIM_DATA03	CSI0_DAT5 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA05
EIM_DATA04	CSI0_DAT6 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA06
EIM_DATA05	CSI0_DAT7 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA07
EIM_DATA06	CSI0_DAT8 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA08
EIM_DATA07	CSI0_DAT9 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA09
EIM_DATA08	CSI0_DAT12 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA12
EIM_DATA09	CSI0_DAT13 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA13
EIM_DATA10	CSI0_DAT14 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA14
EIM_DATA11	CSI0_DAT15 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA15
EIM_DATA12	CSI0_DAT16 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA16
EIM_DATA13	CSI0_DAT17 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA17
EIM_DATA14	CSI0_DAT18 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA18
EIM_DATA15	CSI0_DAT19 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA19
EIM_DATA16	EIM_D16 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA16
EIM_DATA17	EIM_D17 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA17

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
EIM_DATA18	EIM_D18 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA18
EIM_DATA19	EIM_D19 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA19
EIM_DATA20	EIM_D20 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA20
EIM_DATA21	EIM_D21 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA21
EIM_DATA22	EIM_D22 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA22
EIM_DATA23	EIM_D23 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA23
EIM_DATA24	EIM_D24 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA24
EIM_DATA25	EIM_D25 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA25
EIM_DATA26	EIM_D26 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA26
EIM_DATA27	EIM_D27 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA27
EIM_DATA28	EIM_D28 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA28
EIM_DATA29	EIM_D29 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA29
EIM_DATA30	EIM_D30 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA30
EIM_DATA31	EIM_D31 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA31
EIM_DTACK_B	EIM_WAIT (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_WAIT_B
EIM_EB0_B	EIM_EB0 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB0_B
EIM_EB1_B	EIM_EB1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB1_B
EIM_EB2_B	EIM_EB2 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB2_B
EIM_EB3_B	EIM_EB3 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB3_B
EIM_LBA_B	EIM_LBA (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_LBA_B
EIM_OE_B	EIM_OE (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_OE_B
EIM_RW	EIM_RW (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_RW
EIM_WAIT_B	EIM_WAIT (ALT0)	IOMUXC_SW_MUX_CTL_PAD_EIM_WAIT_B
ENET - 10/100/1000-Mbps Ethernet MAC		
ENET_1588_EVENT0_IN	ENET_TXD1 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA1
ENET_1588_EVENT0_OUT	GPIO_19 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_GPIO19
ENET_1588_EVENT1_IN	ENET_MDC (ALT4)	IOMUXC_SW_MUX_CTL_PAD_ENET_MDC
ENET_1588_EVENT1_OUT	ENET_MDIO (ALT4)	IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO
ENET_1588_EVENT2_IN	GPIO_16 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_GPIO16
ENET_1588_EVENT2_OUT	ENET_RX_ER (ALT4)	IOMUXC_SW_MUX_CTL_PAD_ENET_RX_ER
ENET_1588_EVENT3_IN	GPIO_17 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_GPIO17
ENET_1588_EVENT3_OUT	ENET_RXD1 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA1
ENET_COL	KEY_ROW1 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW1
ENET_CRS	KEY_COL3 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL3
ENET_MDC	ENET_MDC (ALT1)	IOMUXC_SW_MUX_CTL_PAD_ENET_MDC
	KEY_COL2 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL2
ENET_MDIO	ENET_MDIO (ALT1)	IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO IOMUXC_ENET_MAC0_MDIO_SELECT_INPUT
	KEY_COL1 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL1

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
		IOMUXC_ENET_MAC0_MDIO_SELECT_INPUT
ENET_REF_CLK	GPIO_16 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_GPIO16
		IOMUXC_ENET_REF_CLK_SELECT_INPUT
	RGMII_TX_CTL (ALT7)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TX_CTL
		IOMUXC_ENET_REF_CLK_SELECT_INPUT
ENET_RX_CLK	GPIO_18 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_GPIO18
		IOMUXC_ENET_MAC0_RX_CLK_SELECT_INPUT
ENET_RX_DATA0	ENET_RXD0 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA0
		IOMUXC_ENET_MAC0_RX_DATA0_SELECT_INPUT
ENET_RX_DATA1	ENET_RXD1 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA1
		IOMUXC_ENET_MAC0_RX_DATA1_SELECT_INPUT
ENET_RX_DATA2	KEY_COL2 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL2
		IOMUXC_ENET_MAC0_RX_DATA2_SELECT_INPUT
ENET_RX_DATA3	KEY_COL0 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL0
		IOMUXC_ENET_MAC0_RX_DATA3_SELECT_INPUT
ENET_RX_EN	ENET_CRS_DV (ALT1)	IOMUXC_SW_MUX_CTL_PAD_ENET_CRS_DV
		IOMUXC_ENET_MAC0_RX_EN_SELECT_INPUT
ENET_RX_ER	ENET_RX_ER (ALT1)	IOMUXC_SW_MUX_CTL_PAD_ENET_RX_ER
ENET_TX_CLK	ENET_REF_CLK (ALT1)	IOMUXC_SW_MUX_CTL_PAD_ENET_REF_CLK
ENET_TX_DATA0	ENET_TXD0 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA0
ENET_TX_DATA1	ENET_TXD1 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA1
ENET_TX_DATA2	KEY_ROW2 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW2
ENET_TX_DATA3	KEY_ROW0 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW0
ENET_TX_EN	ENET_TX_EN (ALT1)	IOMUXC_SW_MUX_CTL_PAD_ENET_TX_EN
ENET_TX_ER	GPIO_19 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_GPIO19
RGMII_RD0	RGMII_RD0 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RD0
		IOMUXC_ENET_MAC0_RX_DATA0_SELECT_INPUT
RGMII_RD1	RGMII_RD1 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RD1
		IOMUXC_ENET_MAC0_RX_DATA1_SELECT_INPUT
RGMII_RD2	RGMII_RD2 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RD2
		IOMUXC_ENET_MAC0_RX_DATA2_SELECT_INPUT
RGMII_RD3	RGMII_RD3 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RD3
		IOMUXC_ENET_MAC0_RX_DATA3_SELECT_INPUT
RGMII_RXC	RGMII_RXC (ALT1)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RXC
		IOMUXC_ENET_MAC0_RX_CLK_SELECT_INPUT
RGMII_RX_CTL	RGMII_RX_CTL (ALT1)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RX_CTL
		IOMUXC_ENET_MAC0_RX_EN_SELECT_INPUT
RGMII_TD0	RGMII_TD0 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TD0
RGMII_TD1	RGMII_TD1 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TD1

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
RGMII_TD2	RGMII_TD2 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TD2
RGMII_TD3	RGMII_TD3 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TD3
RGMII_TXC	RGMII_TXC (ALT1)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TXC
RGMII_TX_CTL	RGMII_TX_CTL (ALT1)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TX_CTL
EPDC - Electrophoretic Display Controller		
EPDC_BDR0	EIM_DA2 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD02
EPDC_BDR1	EIM_DA3 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD03
EPDC_DATA00	EIM_A16 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR16
EPDC_DATA01	EIM_DA10 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD10
EPDC_DATA02	EIM_DA12 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD12
EPDC_DATA03	EIM_DA11 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD11
EPDC_DATA04	EIM_LBA (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_LBA_B
EPDC_DATA05	EIM_EB2 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB2_B
EPDC_DATA06	EIM_CS0 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_CS0_B
EPDC_DATA07	EIM_RW (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_RW
EPDC_DATA08	EIM_CS1 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_CS1_B
EPDC_DATA09	EIM_DA15 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD15
EPDC_DATA10	EIM_D16 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA16
EPDC_DATA11	EIM_D23 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA23
EPDC_DATA12	EIM_D19 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA19
EPDC_DATA13	EIM_DA13 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD13
EPDC_DATA14	EIM_DA14 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD14
EPDC_DATA15	EIM_A25 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR25
EPDC_GDCLK	EIM_A21 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR21
EPDC_GDOE	EIM_A23 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR23
EPDC_GDRL	EIM_A24 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR24
EPDC_GDSP	EIM_A22 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR22
EPDC_PWR_COM	EIM_EB0 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB0_B
EPDC_PWR_CTRL0	EIM_A18 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR18
EPDC_PWR_CTRL1	EIM_A19 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR19
EPDC_PWR_CTRL2	EIM_A20 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR20
EPDC_PWR_CTRL3	EIM_D28 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA28
EPDC_PWR_IRQ	EIM_OE (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_OE_B
EPDC_PWR_STAT	EIM_A17 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR17
EPDC_PWR_WAKE	EIM_D29 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA29
EPDC_SDCE0	EIM_DA4 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD04
	EIM_EB3 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB3_B
EPDC_SDCE1	EIM_DA5 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD05
EPDC_SDCE2	EIM_DA6 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD06

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
EPDC_SDCE3	EIM_DA7 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD07
EPDC_SDCE4	EIM_DA8 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD08
EPDC_SDCE5	EIM_DA9 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD09
EPDC_SDCE6	EIM_D22 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA22
EPDC_SDCE7	EIM_D24 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA24
EPDC_SDCE8	EIM_D25 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA25
EPDC_SDCE9	EIM_BCLK (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_BCLK
EPDC_SDCLK_N	EIM_DA0 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD00
EPDC_SDCLK_P	EIM_D31 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA31
EPDC_SDLE	EIM_DA1 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD01
EPDC_SDOE	EIM_D27 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA27
EPDC_SDOED	EIM_D26 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA26
EPDC_SDOEZ	EIM_D30 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA30
EPDC_SDSHR	EIM_EB1 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB1_B
EPDC_VCOM0	EIM_D17 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA17
EPDC_VCOM1	EIM_D18 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA18
EPIT1 - Enhanced Periodic Interrupt Timer		
EPIT1_OUT	EIM_D19 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA19
	GPIO_0 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_GPIO00
	GPIO_7 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_GPIO07
EPIT2 - Enhanced Periodic Interrupt Timer		
EPIT2_OUT	EIM_D20 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA20
	GPIO_8 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_GPIO08
ESAI - Enhanced Serial Audio Interface		
ESAI_RX_CLK	ENET_MDIO (ALT2)	IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO IOMUXC_ESAI_RX_CLK_SELECT_INPUT
	GPIO_1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_GPIO01 IOMUXC_ESAI_RX_CLK_SELECT_INPUT
ESAI_RX_FS	ENET_REF_CLK (ALT2)	IOMUXC_SW_MUX_CTL_PAD_ENET_REF_CLK IOMUXC_ESAI_RX_FS_SELECT_INPUT
	GPIO_9 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_GPIO09 IOMUXC_ESAI_RX_FS_SELECT_INPUT
ESAI_RX_HF_CLK	ENET_RX_ER (ALT2)	IOMUXC_SW_MUX_CTL_PAD_ENET_RX_ER IOMUXC_ESAI_RX_HF_CLK_SELECT_INPUT
	GPIO_3 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_GPIO03 IOMUXC_ESAI_RX_HF_CLK_SELECT_INPUT
ESAI_TX0	GPIO_17 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_GPIO17 IOMUXC_ESAI_SDO0_SELECT_INPUT
	NANDF_CS2 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS2_B

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
		IOMUXC_ESAI_SDO0_SELECT_INPUT
ESAI_TX1	GPIO_18 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_GPIO18
		IOMUXC_ESAI_SDO1_SELECT_INPUT
	NANDF_CS3 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS3_B
		IOMUXC_ESAI_SDO1_SELECT_INPUT
ESAI_TX2_RX3	ENET_TXD1 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA1
		IOMUXC_ESAI_SDO2_SDI3_SELECT_INPUT
	GPIO_5 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_GPIO05
		IOMUXC_ESAI_SDO2_SDI3_SELECT_INPUT
ESAI_TX3_RX2	ENET_TX_EN (ALT2)	IOMUXC_SW_MUX_CTL_PAD_ENET_TX_EN
		IOMUXC_ESAI_SDO3_SDI2_SELECT_INPUT
	GPIO_16 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_GPIO16
		IOMUXC_ESAI_SDO3_SDI2_SELECT_INPUT
ESAI_TX4_RX1	ENET_TXD0 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA0
		IOMUXC_ESAI_SDO4_SDI1_SELECT_INPUT
	GPIO_7 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_GPIO07
		IOMUXC_ESAI_SDO4_SDI1_SELECT_INPUT
ESAI_TX5_RX0	ENET_MDC (ALT2)	IOMUXC_SW_MUX_CTL_PAD_ENET_MDC
		IOMUXC_ESAI_SDO5_SDI0_SELECT_INPUT
	GPIO_8 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_GPIO08
		IOMUXC_ESAI_SDO5_SDI0_SELECT_INPUT
ESAI_TX_CLK	ENET_CRS_DV (ALT2)	IOMUXC_SW_MUX_CTL_PAD_ENET_CRS_DV
		IOMUXC_ESAI_TX_CLK_SELECT_INPUT
	GPIO_6 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_GPIO06
		IOMUXC_ESAI_TX_CLK_SELECT_INPUT
ESAI_TX_FS	ENET_RXD1 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA1
		IOMUXC_ESAI_TX_FS_SELECT_INPUT
	GPIO_2 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_GPIO02
		IOMUXC_ESAI_TX_FS_SELECT_INPUT
ESAI_TX_HF_CLK	ENET_RXD0 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA0
		IOMUXC_ESAI_TX_HF_CLK_SELECT_INPUT
	GPIO_4 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_GPIO04
		IOMUXC_ESAI_TX_HF_CLK_SELECT_INPUT
FLEXCAN1 - Flexible Controller Area Network		
FLEXCAN1_RX	GPIO_8 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_GPIO08
		IOMUXC_FLEXCAN1_RX_SELECT_INPUT
	KEY_ROW2 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW2
		IOMUXC_FLEXCAN1_RX_SELECT_INPUT

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
	SD3_CLK (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD3_CLK
		IOMUXC_FLEXCAN1_RX_SELECT_INPUT
FLEXCAN1_TX	GPIO_7 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_GPIO07
	KEY_COL2 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL2
	SD3_CMD (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD3_CMD
FLEXCAN2 - Flexible Controller Area Network		
FLEXCAN2_RX	KEY_ROW4 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW4
		IOMUXC_FLEXCAN2_RX_SELECT_INPUT
	SD3_DAT1 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA1
		IOMUXC_FLEXCAN2_RX_SELECT_INPUT
FLEXCAN2_TX	KEY_COL4 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL4
	SD3_DAT0 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA0
GPIO1 - General Purpose Input/Output		
GPIO1_IO00	GPIO_0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_GPIO00
GPIO1_IO01	GPIO_1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_GPIO01
GPIO1_IO02	GPIO_2 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_GPIO02
GPIO1_IO03	GPIO_3 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_GPIO03
GPIO1_IO04	GPIO_4 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_GPIO04
GPIO1_IO05	GPIO_5 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_GPIO05
GPIO1_IO06	GPIO_6 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_GPIO06
GPIO1_IO07	GPIO_7 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_GPIO07
GPIO1_IO08	GPIO_8 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_GPIO08
GPIO1_IO09	GPIO_9 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_GPIO09
GPIO1_IO10	SD2_CLK (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD2_CLK
GPIO1_IO11	SD2_CMD (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD2_CMD
GPIO1_IO12	SD2_DAT3 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3
GPIO1_IO13	SD2_DAT2 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2
GPIO1_IO14	SD2_DAT1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1
GPIO1_IO15	SD2_DAT0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0
GPIO1_IO16	SD1_DAT0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0
GPIO1_IO17	SD1_DAT1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1
GPIO1_IO18	SD1_CMD (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD1_CMD
GPIO1_IO19	SD1_DAT2 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2
GPIO1_IO20	SD1_CLK (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD1_CLK
GPIO1_IO21	SD1_DAT3 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3
GPIO1_IO22	ENET_MDIO (ALT5)	IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO
GPIO1_IO23	ENET_REF_CLK (ALT5)	IOMUXC_SW_MUX_CTL_PAD_ENET_REF_CLK
GPIO1_IO24	ENET_RX_ER (ALT5)	IOMUXC_SW_MUX_CTL_PAD_ENET_RX_ER
GPIO1_IO25	ENET_CRS_DV (ALT5)	IOMUXC_SW_MUX_CTL_PAD_ENET_CRS_DV

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
GPIO1_IO26	ENET_RXD1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA1
GPIO1_IO27	ENET_RXD0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA0
GPIO1_IO28	ENET_TX_EN (ALT5)	IOMUXC_SW_MUX_CTL_PAD_ENET_TX_EN
GPIO1_IO29	ENET_TXD1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA1
GPIO1_IO30	ENET_TXD0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA0
GPIO1_IO31	ENET_MDC (ALT5)	IOMUXC_SW_MUX_CTL_PAD_ENET_MDC
GPIO2 - General Purpose Input/Output		
GPIO2_IO00	NANDF_D0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA00
GPIO2_IO01	NANDF_D1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA01
GPIO2_IO02	NANDF_D2 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA02
GPIO2_IO03	NANDF_D3 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA03
GPIO2_IO04	NANDF_D4 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA04
GPIO2_IO05	NANDF_D5 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA05
GPIO2_IO06	NANDF_D6 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA06
GPIO2_IO07	NANDF_D7 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA07
GPIO2_IO08	SD4_DAT0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA0
GPIO2_IO09	SD4_DAT1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA1
GPIO2_IO10	SD4_DAT2 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA2
GPIO2_IO11	SD4_DAT3 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA3
GPIO2_IO12	SD4_DAT4 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA4
GPIO2_IO13	SD4_DAT5 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA5
GPIO2_IO14	SD4_DAT6 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA6
GPIO2_IO15	SD4_DAT7 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA7
GPIO2_IO16	EIM_A22 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR22
GPIO2_IO17	EIM_A21 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR21
GPIO2_IO18	EIM_A20 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR20
GPIO2_IO19	EIM_A19 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR19
GPIO2_IO20	EIM_A18 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR18
GPIO2_IO21	EIM_A17 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR17
GPIO2_IO22	EIM_A16 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR16
GPIO2_IO23	EIM_CS0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_CS0_B
GPIO2_IO24	EIM_CS1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_CS1_B
GPIO2_IO25	EIM_OE (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_OE_B
GPIO2_IO26	EIM_RW (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_RW
GPIO2_IO27	EIM_LBA (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_LBA_B
GPIO2_IO28	EIM_EB0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB0_B
GPIO2_IO29	EIM_EB1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB1_B
GPIO2_IO30	EIM_EB2 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB2_B
GPIO2_IO31	EIM_EB3 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB3_B

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
GPIO3 - General Purpose Input/Output		
GPIO3_IO00	EIM_DA0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD00
GPIO3_IO01	EIM_DA1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD01
GPIO3_IO02	EIM_DA2 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD02
GPIO3_IO03	EIM_DA3 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD03
GPIO3_IO04	EIM_DA4 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD04
GPIO3_IO05	EIM_DA5 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD05
GPIO3_IO06	EIM_DA6 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD06
GPIO3_IO07	EIM_DA7 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD07
GPIO3_IO08	EIM_DA8 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD08
GPIO3_IO09	EIM_DA9 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD09
GPIO3_IO10	EIM_DA10 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD10
GPIO3_IO11	EIM_DA11 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD11
GPIO3_IO12	EIM_DA12 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD12
GPIO3_IO13	EIM_DA13 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD13
GPIO3_IO14	EIM_DA14 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD14
GPIO3_IO15	EIM_DA15 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD15
GPIO3_IO16	EIM_D16 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA16
GPIO3_IO17	EIM_D17 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA17
GPIO3_IO18	EIM_D18 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA18
GPIO3_IO19	EIM_D19 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA19
GPIO3_IO20	EIM_D20 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA20
GPIO3_IO21	EIM_D21 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA21
GPIO3_IO22	EIM_D22 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA22
GPIO3_IO23	EIM_D23 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA23
GPIO3_IO24	EIM_D24 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA24
GPIO3_IO25	EIM_D25 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA25
GPIO3_IO26	EIM_D26 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA26
GPIO3_IO27	EIM_D27 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA27
GPIO3_IO28	EIM_D28 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA28
GPIO3_IO29	EIM_D29 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA29
GPIO3_IO30	EIM_D30 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA30
GPIO3_IO31	EIM_D31 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA31
GPIO4 - General Purpose Input/Output		
GPIO4_IO05	GPIO_19 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_GPIO19
GPIO4_IO06	KEY_COL0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL0
GPIO4_IO07	KEY_ROW0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW0
GPIO4_IO08	KEY_COL1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL1
GPIO4_IO09	KEY_ROW1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW1

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
GPIO4_IO10	KEY_COL2 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL2
GPIO4_IO11	KEY_ROW2 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW2
GPIO4_IO12	KEY_COL3 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL3
GPIO4_IO13	KEY_ROW3 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW3
GPIO4_IO14	KEY_COL4 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL4
GPIO4_IO15	KEY_ROW4 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW4
GPIO4_IO16	DI0_DISP_CLK (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DI0_DISP_CLK
GPIO4_IO17	DI0_PIN15 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN15
GPIO4_IO18	DI0_PIN2 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN02
GPIO4_IO19	DI0_PIN3 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN03
GPIO4_IO20	DI0_PIN4 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN04
GPIO4_IO21	DISP0_DAT0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA00
GPIO4_IO22	DISP0_DAT1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA01
GPIO4_IO23	DISP0_DAT2 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA02
GPIO4_IO24	DISP0_DAT3 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA03
GPIO4_IO25	DISP0_DAT4 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA04
GPIO4_IO26	DISP0_DAT5 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA05
GPIO4_IO27	DISP0_DAT6 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA06
GPIO4_IO28	DISP0_DAT7 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA07
GPIO4_IO29	DISP0_DAT8 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA08
GPIO4_IO30	DISP0_DAT9 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA09
GPIO4_IO31	DISP0_DAT10 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA10
GPIO5 - General Purpose Input/Output		
GPIO5_IO00	EIM_WAIT (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_WAIT_B
GPIO5_IO02	EIM_A25 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR25
GPIO5_IO04	EIM_A24 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR24
GPIO5_IO05	DISP0_DAT11 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA11
GPIO5_IO06	DISP0_DAT12 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA12
GPIO5_IO07	DISP0_DAT13 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA13
GPIO5_IO08	DISP0_DAT14 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA14
GPIO5_IO09	DISP0_DAT15 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA15
GPIO5_IO10	DISP0_DAT16 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA16
GPIO5_IO11	DISP0_DAT17 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA17
GPIO5_IO12	DISP0_DAT18 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA18
GPIO5_IO13	DISP0_DAT19 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA19
GPIO5_IO14	DISP0_DAT20 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA20
GPIO5_IO15	DISP0_DAT21 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA21
GPIO5_IO16	DISP0_DAT22 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA22
GPIO5_IO17	DISP0_DAT23 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA23

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
GPIO5_IO18	CSI0_PIXCLK (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_PIXCLK
GPIO5_IO19	CSI0_MCLK (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_HSYNC
GPIO5_IO20	CSI0_DATA_EN (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA_EN
GPIO5_IO21	CSI0_VSYNC (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_VSYNC
GPIO5_IO22	CSI0_DAT4 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA04
GPIO5_IO23	CSI0_DAT5 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA05
GPIO5_IO24	CSI0_DAT6 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA06
GPIO5_IO25	CSI0_DAT7 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA07
GPIO5_IO26	CSI0_DAT8 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA08
GPIO5_IO27	CSI0_DAT9 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA09
GPIO5_IO28	CSI0_DAT10 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA10
GPIO5_IO29	CSI0_DAT11 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA11
GPIO5_IO30	CSI0_DAT12 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA12
GPIO5_IO31	CSI0_DAT13 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA13
GPIO6 - General Purpose Input/Output		
GPIO6_IO00	CSI0_DAT14 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA14
GPIO6_IO01	CSI0_DAT15 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA15
GPIO6_IO02	CSI0_DAT16 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA16
GPIO6_IO03	CSI0_DAT17 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA17
GPIO6_IO04	CSI0_DAT18 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA18
GPIO6_IO05	CSI0_DAT19 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA19
GPIO6_IO06	EIM_A23 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR23
GPIO6_IO07	NANDF_CLE (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_CLE
GPIO6_IO08	NANDF_ALE (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_ALE
GPIO6_IO09	NANDF_WP_B (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_WP_B
GPIO6_IO10	NANDF_RB0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_READY_B
GPIO6_IO11	NANDF_CS0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS0_B
GPIO6_IO14	NANDF_CS1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS1_B
GPIO6_IO15	NANDF_CS2 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS2_B
GPIO6_IO16	NANDF_CS3 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS3_B
GPIO6_IO17	SD3_DAT7 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA7
GPIO6_IO18	SD3_DAT6 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA6
GPIO6_IO19	RGMII_TXC (ALT5)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TXC
GPIO6_IO20	RGMII_TD0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TD0
GPIO6_IO21	RGMII_TD1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TD1
GPIO6_IO22	RGMII_TD2 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TD2
GPIO6_IO23	RGMII_TD3 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TD3
GPIO6_IO24	RGMII_RX_CTL (ALT5)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RX_CTL
GPIO6_IO25	RGMII_RD0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RD0

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
GPIO6_IO26	RGMII_TX_CTL (ALT5)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TX_CTL
GPIO6_IO27	RGMII_RD1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RD1
GPIO6_IO28	RGMII_RD2 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RD2
GPIO6_IO29	RGMII_RD3 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RD3
GPIO6_IO30	RGMII_RXC (ALT5)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RXC
GPIO6_IO31	EIM_BCLK (ALT5)	IOMUXC_SW_MUX_CTL_PAD_EIM_BCLK
GPIO7 - General Purpose Input/Output		
GPIO7_IO00	SD3_DAT5 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA5
GPIO7_IO01	SD3_DAT4 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA4
GPIO7_IO02	SD3_CMD (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD3_CMD
GPIO7_IO03	SD3_CLK (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD3_CLK
GPIO7_IO04	SD3_DAT0 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA0
GPIO7_IO05	SD3_DAT1 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA1
GPIO7_IO06	SD3_DAT2 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA2
GPIO7_IO07	SD3_DAT3 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA3
GPIO7_IO08	SD3_RST (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD3_RESET
GPIO7_IO09	SD4_CMD (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD4_CMD
GPIO7_IO10	SD4_CLK (ALT5)	IOMUXC_SW_MUX_CTL_PAD_SD4_CLK
GPIO7_IO11	GPIO_16 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_GPIO16
GPIO7_IO12	GPIO_17 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_GPIO17
GPIO7_IO13	GPIO_18 (ALT5)	IOMUXC_SW_MUX_CTL_PAD_GPIO18
GPMI - General Purpose Media Interface		
NAND_ALE	NANDF_ALE (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_ALE
NAND_CE0_B	NANDF_CS0 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS0_B
NAND_CE1_B	NANDF_CS1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS1_B
NAND_CE2_B	NANDF_CS2 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS2_B
NAND_CE3_B	NANDF_CS3 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS3_B
NAND_CLE	NANDF_CLE (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_CLE
NAND_DATA00	NANDF_D0 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA00
NAND_DATA01	NANDF_D1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA01
NAND_DATA02	NANDF_D2 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA02
NAND_DATA03	NANDF_D3 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA03
NAND_DATA04	NANDF_D4 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA04
NAND_DATA05	NANDF_D5 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA05
NAND_DATA06	NANDF_D6 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA06
NAND_DATA07	NANDF_D7 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA07
NAND_DQS	SD4_DAT0 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA0
NAND_READY_B	NANDF_RB0 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_READY_B
NAND_RE_B	SD4_CMD (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD4_CMD

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
NAND_WE_B	SD4_CLK (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD4_CLK
NAND_WP_B	NANDF_WP_B (ALT0)	IOMUXC_SW_MUX_CTL_PAD_NAND_WP_B
GPT - General Purpose Timer		
GPT_CAPTURE1	SD1_DAT0 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0
GPT_CAPTURE2	SD1_DAT1 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1
GPT_CLKIN	SD1_CLK (ALT3)	IOMUXC_SW_MUX_CTL_PAD_SD1_CLK
GPT_COMPARE1	SD1_CMD (ALT3)	IOMUXC_SW_MUX_CTL_PAD_SD1_CMD
GPT_COMPARE2	SD1_DAT2 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2
GPT_COMPARE3	SD1_DAT3 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3
HDMI - High Definition Multimedia Interface		
HDMI_TX_CEC_LINE	EIM_A25 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR25 IOMUXC_HDMI_ICECIN_SELECT_INPUT
	KEY_ROW2 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW2 IOMUXC_HDMI_ICECIN_SELECT_INPUT
HDMI_TX_CLK_N	HDMI_CLKM	Not multiplexed.
HDMI_TX_CLK_P	HDMI_CLKP	Not multiplexed.
HDMI_TX_DATA0_N	HDMI_D0M	Not multiplexed.
HDMI_TX_DATA0_P	HDMI_D0P	Not multiplexed.
HDMI_TX_DATA1_N	HDMI_D1M	Not multiplexed.
HDMI_TX_DATA1_P	HDMI_D1P	Not multiplexed.
HDMI_TX_DATA2_N	HDMI_D2M	Not multiplexed.
HDMI_TX_DATA2_P	HDMI_D2P	Not multiplexed.
HDMI_TX_DDC_CEC	HDMI_DDCCEC	Not multiplexed.
HDMI_TX_DDC_SCL	EIM_EB2 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB2_B IOMUXC_HDMI_II2C_CLKIN_SELECT_INPUT
	KEY_COL3 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL3 IOMUXC_HDMI_II2C_CLKIN_SELECT_INPUT
HDMI_TX_DDC_SDA	EIM_D16 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA16 IOMUXC_HDMI_II2C_DATAIN_SELECT_INPUT
	KEY_ROW3 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW3 IOMUXC_HDMI_II2C_DATAIN_SELECT_INPUT
HDMI_TX_HPD	HDMI_HPD	Not multiplexed.
I2C1 - I2C Controller		
I2C1_SCL	CSI0_DAT9 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA09 IOMUXC_I2C1_SCL_IN_SELECT_INPUT
	EIM_D21 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA21 IOMUXC_I2C1_SCL_IN_SELECT_INPUT
I2C1_SDA	CSI0_DAT8 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA08 IOMUXC_I2C1_SDA_IN_SELECT_INPUT

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
	EIM_D28 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA28
		IOMUXC_I2C1_SDA_IN_SELECT_INPUT
I2C2 - I2C Controller		
I2C2_SCL	EIM_EB2 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB2_B
		IOMUXC_I2C2_SCL_IN_SELECT_INPUT
	KEY_COL3 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL3
		IOMUXC_I2C2_SCL_IN_SELECT_INPUT
I2C2_SDA	EIM_D16 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA16
		IOMUXC_I2C2_SDA_IN_SELECT_INPUT
	KEY_ROW3 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW3
		IOMUXC_I2C2_SDA_IN_SELECT_INPUT
I2C3 - I2C Controller		
I2C3_SCL	EIM_D17 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA17
		IOMUXC_I2C3_SCL_IN_SELECT_INPUT
	GPIO_3 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_GPIO03
		IOMUXC_I2C3_SCL_IN_SELECT_INPUT
	GPIO_5 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_GPIO05
		IOMUXC_I2C3_SCL_IN_SELECT_INPUT
I2C3_SDA	EIM_D18 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA18
		IOMUXC_I2C3_SDA_IN_SELECT_INPUT
	GPIO_6 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_GPIO06
		IOMUXC_I2C3_SDA_IN_SELECT_INPUT
	GPIO_16 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_GPIO16
		IOMUXC_I2C3_SDA_IN_SELECT_INPUT
I2C4 - I2C Controller		
I2C4_SCL	ENET_TX_EN (ALT9)	IOMUXC_SW_MUX_CTL_PAD_ENET_TX_EN
		IOMUXC_I2C4_SCL_IN_SELECT_INPUT
	GPIO_7 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_GPIO07
		IOMUXC_I2C4_SCL_IN_SELECT_INPUT
	NANDF_WP_B (ALT9)	IOMUXC_SW_MUX_CTL_PAD_NAND_WP_B
		IOMUXC_I2C4_SCL_IN_SELECT_INPUT
I2C4_SDA	ENET_TXD1 (ALT9)	IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA1
		IOMUXC_I2C4_SDA_IN_SELECT_INPUT
	GPIO_8 (ALT8)	IOMUXC_SW_MUX_CTL_PAD_GPIO08
		IOMUXC_I2C4_SDA_IN_SELECT_INPUT
	NANDF_CS3 (ALT9)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS3_B
		IOMUXC_I2C4_SDA_IN_SELECT_INPUT
IPU1 - Image Processing Unit		

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
IPU1_CSI0_DATA00	EIM_D27 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA27
IPU1_CSI0_DATA01	EIM_D26 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA26
IPU1_CSI0_DATA02	EIM_D31 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA31
IPU1_CSI0_DATA03	EIM_D30 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA30
IPU1_CSI0_DATA04	CSI0_DAT4 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA04
IPU1_CSI0_DATA05	CSI0_DAT5 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA05
IPU1_CSI0_DATA06	CSI0_DAT6 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA06
IPU1_CSI0_DATA07	CSI0_DAT7 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA07
IPU1_CSI0_DATA08	CSI0_DAT8 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA08
IPU1_CSI0_DATA09	CSI0_DAT9 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA09
IPU1_CSI0_DATA10	CSI0_DAT10 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA10
IPU1_CSI0_DATA11	CSI0_DAT11 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA11
IPU1_CSI0_DATA12	CSI0_DAT12 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA12
IPU1_CSI0_DATA13	CSI0_DAT13 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA13
IPU1_CSI0_DATA14	CSI0_DAT14 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA14
IPU1_CSI0_DATA15	CSI0_DAT15 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA15
IPU1_CSI0_DATA16	CSI0_DAT16 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA16
IPU1_CSI0_DATA17	CSI0_DAT17 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA17
IPU1_CSI0_DATA18	CSI0_DAT18 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA18
IPU1_CSI0_DATA19	CSI0_DAT19 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA19
IPU1_CSI0_DATA_EN	CSI0_DATA_EN (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA_EN
IPU1_CSI0_HSYNC	CSI0_MCLK (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_HSYNC
IPU1_CSI0_PIXCLK	CSI0_PIXCLK (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_PIXCLK
IPU1_CSI0_VSYNC	CSI0_VSYNC (ALT0)	IOMUXC_SW_MUX_CTL_PAD_CSI0_VSYNC
IPU1_CSI1_DATA00	EIM_DA9 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD09
IPU1_CSI1_DATA01	EIM_DA8 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD08
IPU1_CSI1_DATA02	EIM_DA7 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD07
IPU1_CSI1_DATA03	EIM_DA6 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD06
IPU1_CSI1_DATA04	EIM_DA5 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD05
IPU1_CSI1_DATA05	EIM_DA4 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD04
IPU1_CSI1_DATA06	EIM_DA3 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD03
IPU1_CSI1_DATA07	EIM_DA2 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD02
IPU1_CSI1_DATA08	EIM_DA1 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD01
IPU1_CSI1_DATA09	EIM_DA0 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD00
IPU1_CSI1_DATA10	EIM_D22 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA22 IOMUXC_IPU1_SENS1_DATA10_SELECT_INPUT
	EIM_EB1 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB1_B IOMUXC_IPU1_SENS1_DATA10_SELECT_INPUT
IPU1_CSI1_DATA11	EIM_D21 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA21

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
		IOMUXC_IPU1_SENS1_DATA11_SELECT_INPUT
	EIM_EB0 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB0_B
		IOMUXC_IPU1_SENS1_DATA11_SELECT_INPUT
IPU1_CSI1_DATA12	EIM_A17 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR17
		IOMUXC_IPU1_SENS1_DATA12_SELECT_INPUT
	EIM_D28 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA28
		IOMUXC_IPU1_SENS1_DATA12_SELECT_INPUT
IPU1_CSI1_DATA13	EIM_A18 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR18
		IOMUXC_IPU1_SENS1_DATA13_SELECT_INPUT
	EIM_D27 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA27
		IOMUXC_IPU1_SENS1_DATA13_SELECT_INPUT
IPU1_CSI1_DATA14	EIM_A19 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR19
		IOMUXC_IPU1_SENS1_DATA14_SELECT_INPUT
	EIM_D26 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA26
		IOMUXC_IPU1_SENS1_DATA14_SELECT_INPUT
IPU1_CSI1_DATA15	EIM_A20 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR20
		IOMUXC_IPU1_SENS1_DATA15_SELECT_INPUT
	EIM_D20 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA20
		IOMUXC_IPU1_SENS1_DATA15_SELECT_INPUT
IPU1_CSI1_DATA16	EIM_A21 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR21
		IOMUXC_IPU1_SENS1_DATA16_SELECT_INPUT
	EIM_D19 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA19
		IOMUXC_IPU1_SENS1_DATA16_SELECT_INPUT
IPU1_CSI1_DATA17	EIM_A22 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR22
		IOMUXC_IPU1_SENS1_DATA17_SELECT_INPUT
	EIM_D18 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA18
		IOMUXC_IPU1_SENS1_DATA17_SELECT_INPUT
IPU1_CSI1_DATA18	EIM_A23 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR23
		IOMUXC_IPU1_SENS1_DATA18_SELECT_INPUT
	EIM_D16 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA16
		IOMUXC_IPU1_SENS1_DATA18_SELECT_INPUT
IPU1_CSI1_DATA19	EIM_A24 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR24
		IOMUXC_IPU1_SENS1_DATA19_SELECT_INPUT
	EIM_EB2 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB2_B
		IOMUXC_IPU1_SENS1_DATA19_SELECT_INPUT
IPU1_CSI1_DATA_EN	EIM_DA10 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD10
		IOMUXC_IPU1_SENS1_DATA_EN_SELECT_INPUT
	EIM_D23 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA23

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
		IOMUXC_IPU1_SENS1_DATA_EN_SELECT_INPUT
IPU1_CSI1_HSYNC	EIM_DA11 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD11
		IOMUXC_IPU1_SENS1_HSYNC_SELECT_INPUT
	EIM_EB3 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB3_B
		IOMUXC_IPU1_SENS1_HSYNC_SELECT_INPUT
IPU1_CSI1_PIXCLK	EIM_A16 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR16
		IOMUXC_IPU1_SENS1_PIX_CLK_SELECT_INPUT
	EIM_D17 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA17
		IOMUXC_IPU1_SENS1_PIX_CLK_SELECT_INPUT
IPU1_CSI1_VSYNC	EIM_DA12 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD12
		IOMUXC_IPU1_SENS1_VSYNC_SELECT_INPUT
	EIM_D29 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA29
		IOMUXC_IPU1_SENS1_VSYNC_SELECT_INPUT
IPU1_DI0_D0_CS	EIM_D23 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA23
IPU1_DI0_D1_CS	EIM_A25 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR25
IPU1_DI0_DISP_CLK	DI0_DISP_CLK (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DI0_DISP_CLK
IPU1_DI0_PIN01	EIM_D22 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA22
IPU1_DI0_PIN02	DI0_PIN2 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN02
IPU1_DI0_PIN03	DI0_PIN3 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN03
IPU1_DI0_PIN04	DI0_PIN4 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN04
IPU1_DI0_PIN05	EIM_D16 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA16
IPU1_DI0_PIN06	EIM_D17 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA17
IPU1_DI0_PIN07	EIM_D18 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA18
IPU1_DI0_PIN08	EIM_D19 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA19
IPU1_DI0_PIN11	EIM_D30 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA30
IPU1_DI0_PIN12	EIM_D31 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA31
IPU1_DI0_PIN13	EIM_D28 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA28
IPU1_DI0_PIN14	EIM_D29 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA29
IPU1_DI0_PIN15	DI0_PIN15 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN15
IPU1_DI0_PIN16	EIM_D20 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA20
IPU1_DI0_PIN17	EIM_D21 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA21
IPU1_DI1_D0_CS	EIM_DA13 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD13
	EIM_D18 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA18
IPU1_DI1_D1_CS	EIM_DA14 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD14
IPU1_DI1_DISP_CLK	EIM_A16 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR16
IPU1_DI1_PIN01	EIM_DA15 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD15
IPU1_DI1_PIN02	EIM_DA11 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD11
	EIM_D23 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA23
IPU1_DI1_PIN03	EIM_DA12 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD12

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
	EIM_EB3 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB3_B
IPU1_DI1_PIN04	EIM_DA15 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD15
IPU1_DI1_PIN05	EIM_CS0 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_CS0_B
IPU1_DI1_PIN06	EIM_CS1 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_CS1_B
IPU1_DI1_PIN07	EIM_OE (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_OE_B
IPU1_DI1_PIN08	EIM_RW (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_RW
IPU1_DI1_PIN11	EIM_D26 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA26
IPU1_DI1_PIN12	EIM_A25 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR25
IPU1_DI1_PIN13	EIM_D27 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA27
IPU1_DI1_PIN14	EIM_D23 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA23
IPU1_DI1_PIN15	EIM_DA10 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD10
	EIM_D29 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA29
IPU1_DI1_PIN16	EIM_BCLK (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_BCLK
IPU1_DI1_PIN17	EIM_LBA (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_LBA_B
IPU1_DISP0_DATA00	DISP0_DAT0 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA00
IPU1_DISP0_DATA01	DISP0_DAT1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA01
IPU1_DISP0_DATA02	DISP0_DAT2 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA02
IPU1_DISP0_DATA03	DISP0_DAT3 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA03
IPU1_DISP0_DATA04	DISP0_DAT4 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA04
IPU1_DISP0_DATA05	DISP0_DAT5 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA05
IPU1_DISP0_DATA06	DISP0_DAT6 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA06
IPU1_DISP0_DATA07	DISP0_DAT7 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA07
IPU1_DISP0_DATA08	DISP0_DAT8 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA08
IPU1_DISP0_DATA09	DISP0_DAT9 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA09
IPU1_DISP0_DATA10	DISP0_DAT10 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA10
IPU1_DISP0_DATA11	DISP0_DAT11 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA11
IPU1_DISP0_DATA12	DISP0_DAT12 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA12
IPU1_DISP0_DATA13	DISP0_DAT13 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA13
IPU1_DISP0_DATA14	DISP0_DAT14 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA14
IPU1_DISP0_DATA15	DISP0_DAT15 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA15
IPU1_DISP0_DATA16	DISP0_DAT16 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA16
IPU1_DISP0_DATA17	DISP0_DAT17 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA17
IPU1_DISP0_DATA18	DISP0_DAT18 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA18
IPU1_DISP0_DATA19	DISP0_DAT19 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA19
IPU1_DISP0_DATA20	DISP0_DAT20 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA20
IPU1_DISP0_DATA21	DISP0_DAT21 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA21
IPU1_DISP0_DATA22	DISP0_DAT22 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA22
IPU1_DISP0_DATA23	DISP0_DAT23 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA23
IPU1_DISP1_DATA00	EIM_DA9 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD09

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
IPU1_DISP1_DATA01	EIM_DA8 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD08
IPU1_DISP1_DATA02	EIM_DA7 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD07
IPU1_DISP1_DATA03	EIM_DA6 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD06
IPU1_DISP1_DATA04	EIM_DA5 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD05
IPU1_DISP1_DATA05	EIM_DA4 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD04
IPU1_DISP1_DATA06	EIM_DA3 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD03
IPU1_DISP1_DATA07	EIM_DA2 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD02
IPU1_DISP1_DATA08	EIM_DA1 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD01
IPU1_DISP1_DATA09	EIM_DA0 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD00
IPU1_DISP1_DATA10	EIM_EB1 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB1_B
IPU1_DISP1_DATA11	EIM_EB0 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB0_B
IPU1_DISP1_DATA12	EIM_A17 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR17
IPU1_DISP1_DATA13	EIM_A18 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR18
IPU1_DISP1_DATA14	EIM_A19 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR19
IPU1_DISP1_DATA15	EIM_A20 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR20
IPU1_DISP1_DATA16	EIM_A21 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR21
IPU1_DISP1_DATA17	EIM_A22 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR22
IPU1_DISP1_DATA18	EIM_A23 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR23
IPU1_DISP1_DATA19	EIM_A24 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR24
IPU1_DISP1_DATA20	EIM_D31 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA31
IPU1_DISP1_DATA21	EIM_D30 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA30
IPU1_DISP1_DATA22	EIM_D26 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA26
IPU1_DISP1_DATA23	EIM_D27 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA27
IPU1_EXT_TRIG	EIM_D28 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA28
IPU1_SISG0	NANDF_CS2 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS2_B
IPU1_SISG1	NANDF_CS3 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS3_B
IPU1_SISG2	EIM_A24 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR24
	EIM_D26 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA26
IPU1_SISG3	EIM_A23 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR23
	EIM_D27 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA27
IPU1_SISG4	KEY_COL4 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL4
IPU1_SISG5	KEY_ROW4 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW4
KPP - Keypad Port		
KEY_COL0	KEY_COL0 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL0
KEY_COL1	KEY_COL1 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL1
KEY_COL2	KEY_COL2 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL2
KEY_COL3	KEY_COL3 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL3
KEY_COL4	KEY_COL4 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL4
KEY_COL5	CSI0_DAT4 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA04

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
		IOMUXC_KEY_COL5_SELECT_INPUT
	GPIO_0 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_GPIO00
		IOMUXC_KEY_COL5_SELECT_INPUT
	GPIO_19 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_GPIO19
		IOMUXC_KEY_COL5_SELECT_INPUT
	SD2_CLK (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD2_CLK
		IOMUXC_KEY_COL5_SELECT_INPUT
KEY_COL6	CSI0_DAT6 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA06
		IOMUXC_KEY_COL6_SELECT_INPUT
	GPIO_9 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_GPIO09
		IOMUXC_KEY_COL6_SELECT_INPUT
	SD2_DAT3 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3
		IOMUXC_KEY_COL6_SELECT_INPUT
KEY_COL7	CSI0_DAT8 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA08
		IOMUXC_KEY_COL7_SELECT_INPUT
	GPIO_4 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_GPIO04
		IOMUXC_KEY_COL7_SELECT_INPUT
	SD2_DAT1 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1
		IOMUXC_KEY_COL7_SELECT_INPUT
KEY_ROW0	KEY_ROW0 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW0
KEY_ROW1	KEY_ROW1 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW1
KEY_ROW2	KEY_ROW2 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW2
KEY_ROW3	KEY_ROW3 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW3
KEY_ROW4	KEY_ROW4 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW4
KEY_ROW5	CSI0_DAT5 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA05
		IOMUXC_KEY_ROW5_SELECT_INPUT
	GPIO_1 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_GPIO01
		IOMUXC_KEY_ROW5_SELECT_INPUT
	SD2_CMD (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD2_CMD
		IOMUXC_KEY_ROW5_SELECT_INPUT
KEY_ROW6	CSI0_DAT7 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA07
		IOMUXC_KEY_ROW6_SELECT_INPUT
	GPIO_2 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_GPIO02
		IOMUXC_KEY_ROW6_SELECT_INPUT
	SD2_DAT2 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2
		IOMUXC_KEY_ROW6_SELECT_INPUT
KEY_ROW7	CSI0_DAT9 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA09
		IOMUXC_KEY_ROW7_SELECT_INPUT
	GPIO_5 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_GPIO05

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
		IOMUXC_KEY_ROW7_SELECT_INPUT
	SD2_DAT0 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0 IOMUXC_KEY_ROW7_SELECT_INPUT
LDB - LVDS Display Bridge		
LVDS0_CLK_N	LVDS0_CLK_N	Not multiplexed.
LVDS0_CLK_P	LVDS0_CLK_P	Not multiplexed.
LVDS0_DATA0_N	LVDS0_TX0_N	Not multiplexed.
LVDS0_DATA0_P	LVDS0_TX0_P	Not multiplexed.
LVDS0_DATA1_N	LVDS0_TX1_N	Not multiplexed.
LVDS0_DATA1_P	LVDS0_TX1_P	Not multiplexed.
LVDS0_DATA2_N	LVDS0_TX2_N	Not multiplexed.
LVDS0_DATA2_P	LVDS0_TX2_P	Not multiplexed.
LVDS0_DATA3_N	LVDS0_TX3_N	Not multiplexed.
LVDS0_DATA3_P	LVDS0_TX3_P	Not multiplexed.
LVDS1_CLK_N	LVDS1_CLK_N	Not multiplexed.
LVDS1_CLK_P	LVDS1_CLK_P	Not multiplexed.
LVDS1_DATA0_N	LVDS1_TX0_N	Not multiplexed.
LVDS1_DATA0_P	LVDS1_TX0_P	Not multiplexed.
LVDS1_DATA1_N	LVDS1_TX1_N	Not multiplexed.
LVDS1_DATA1_P	LVDS1_TX1_P	Not multiplexed.
LVDS1_DATA2_N	LVDS1_TX2_N	Not multiplexed.
LVDS1_DATA2_P	LVDS1_TX2_P	Not multiplexed.
LVDS1_DATA3_N	LVDS1_TX3_N	Not multiplexed.
LVDS1_DATA3_P	LVDS1_TX3_P	Not multiplexed.
MIPI_CSI - MIPI Camera Serial Interface		
CSI_CLK0_N	CSI_CLK0M	Not multiplexed.
CSI_CLK0_P	CSI_CLK0P	Not multiplexed.
CSI_DATA0_N	CSI_D0M	Not multiplexed.
CSI_DATA0_P	CSI_D0P	Not multiplexed.
CSI_DATA1_N	CSI_D1M	Not multiplexed.
CSI_DATA1_P	CSI_D1P	Not multiplexed.
MIPI_DSI - MIPI Display Serial Interface		
DSI_CLK0_N	DSI_CLK0M	Not multiplexed.
DSI_CLK0_P	DSI_CLK0P	Not multiplexed.
DSI_DATA0_N	DSI_D0M	Not multiplexed.
DSI_DATA0_P	DSI_D0P	Not multiplexed.
DSI_DATA1_N	DSI_D1M	Not multiplexed.
DSI_DATA1_P	DSI_D1P	Not multiplexed.
MIPI_HSI - MIPI High Speed Synchronous Interface		

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
HSI_RX_DATA	RGMII_TD2 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TD2
HSI_RX_FLAG	RGMII_TD1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TD1
HSI_RX_READY	RGMII_RD0 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RD0
HSI_RX_WAKE	RGMII_TD3 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TD3
HSI_TX_DATA	RGMII_RD2 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RD2
HSI_TX_FLAG	RGMII_RD1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RD1
HSI_TX_READY	RGMII_TD0 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TD0
HSI_TX_WAKE	RGMII_RD3 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RD3
MLB - Media Local Bus		
MLB_CLK	ENET_TXD1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA1 IOMUXC_MLB_MLB_CLK_IN_SELECT_INPUT
	GPIO_3 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_GPIO03 IOMUXC_MLB_MLB_CLK_IN_SELECT_INPUT
MLB_CLK_N	MLB_CN	Not multiplexed.
MLB_CLK_P	MLB_CP	Not multiplexed.
MLB_DATA	ENET_MDC (ALT0)	IOMUXC_SW_MUX_CTL_PAD_ENET_MDC IOMUXC_MLB_MLB_DATA_IN_SELECT_INPUT
	GPIO_2 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_GPIO02 IOMUXC_MLB_MLB_DATA_IN_SELECT_INPUT
MLB_DATA_N	MLB_DN	Not multiplexed.
MLB_DATA_P	MLB_DP	Not multiplexed.
MLB_SIG	ENET_RXD1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA1 IOMUXC_MLB_MLB_SIG_IN_SELECT_INPUT
	GPIO_6 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_GPIO06 IOMUXC_MLB_MLB_SIG_IN_SELECT_INPUT
MLB_SIG_N	MLB_SN	Not multiplexed.
MLB_SIG_P	MLB_SP	Not multiplexed.
MMDC - Multi Mode DDR Controller		
DRAM_ADDR00	DRAM_A0	Not multiplexed.
DRAM_ADDR01	DRAM_A1	Not multiplexed.
DRAM_ADDR02	DRAM_A2	Not multiplexed.
DRAM_ADDR03	DRAM_A3	Not multiplexed.
DRAM_ADDR04	DRAM_A4	Not multiplexed.
DRAM_ADDR05	DRAM_A5	Not multiplexed.
DRAM_ADDR06	DRAM_A6	Not multiplexed.
DRAM_ADDR07	DRAM_A7	Not multiplexed.
DRAM_ADDR08	DRAM_A8	Not multiplexed.
DRAM_ADDR09	DRAM_A9	Not multiplexed.
DRAM_ADDR10	DRAM_A10	Not multiplexed.

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
DRAM_ADDR11	DRAM_A11	Not multiplexed.
DRAM_ADDR12	DRAM_A12	Not multiplexed.
DRAM_ADDR13	DRAM_A13	Not multiplexed.
DRAM_ADDR14	DRAM_A14	Not multiplexed.
DRAM_ADDR15	DRAM_A15	Not multiplexed.
DRAM_CAS_B	DRAM_CAS	Not multiplexed.
DRAM_CS0_B	DRAM_CS0	Not multiplexed.
DRAM_CS1_B	DRAM_CS1	Not multiplexed.
DRAM_DATA00	DRAM_D0	Not multiplexed.
DRAM_DATA01	DRAM_D1	Not multiplexed.
DRAM_DATA02	DRAM_D2	Not multiplexed.
DRAM_DATA03	DRAM_D3	Not multiplexed.
DRAM_DATA04	DRAM_D4	Not multiplexed.
DRAM_DATA05	DRAM_D5	Not multiplexed.
DRAM_DATA06	DRAM_D6	Not multiplexed.
DRAM_DATA07	DRAM_D7	Not multiplexed.
DRAM_DATA08	DRAM_D8	Not multiplexed.
DRAM_DATA09	DRAM_D9	Not multiplexed.
DRAM_DATA10	DRAM_D10	Not multiplexed.
DRAM_DATA11	DRAM_D11	Not multiplexed.
DRAM_DATA12	DRAM_D12	Not multiplexed.
DRAM_DATA13	DRAM_D13	Not multiplexed.
DRAM_DATA14	DRAM_D14	Not multiplexed.
DRAM_DATA15	DRAM_D15	Not multiplexed.
DRAM_DATA16	DRAM_D16	Not multiplexed.
DRAM_DATA17	DRAM_D17	Not multiplexed.
DRAM_DATA18	DRAM_D18	Not multiplexed.
DRAM_DATA19	DRAM_D19	Not multiplexed.
DRAM_DATA20	DRAM_D20	Not multiplexed.
DRAM_DATA21	DRAM_D21	Not multiplexed.
DRAM_DATA22	DRAM_D22	Not multiplexed.
DRAM_DATA23	DRAM_D23	Not multiplexed.
DRAM_DATA24	DRAM_D24	Not multiplexed.
DRAM_DATA25	DRAM_D25	Not multiplexed.
DRAM_DATA26	DRAM_D26	Not multiplexed.
DRAM_DATA27	DRAM_D27	Not multiplexed.
DRAM_DATA28	DRAM_D28	Not multiplexed.
DRAM_DATA29	DRAM_D29	Not multiplexed.
DRAM_DATA30	DRAM_D30	Not multiplexed.

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
DRAM_DATA31	DRAM_D31	Not multiplexed.
DRAM_DATA32	DRAM_D32	Not multiplexed.
DRAM_DATA33	DRAM_D33	Not multiplexed.
DRAM_DATA34	DRAM_D34	Not multiplexed.
DRAM_DATA35	DRAM_D35	Not multiplexed.
DRAM_DATA36	DRAM_D36	Not multiplexed.
DRAM_DATA37	DRAM_D37	Not multiplexed.
DRAM_DATA38	DRAM_D38	Not multiplexed.
DRAM_DATA39	DRAM_D39	Not multiplexed.
DRAM_DATA40	DRAM_D40	Not multiplexed.
DRAM_DATA41	DRAM_D41	Not multiplexed.
DRAM_DATA42	DRAM_D42	Not multiplexed.
DRAM_DATA43	DRAM_D43	Not multiplexed.
DRAM_DATA44	DRAM_D44	Not multiplexed.
DRAM_DATA45	DRAM_D45	Not multiplexed.
DRAM_DATA46	DRAM_D46	Not multiplexed.
DRAM_DATA47	DRAM_D47	Not multiplexed.
DRAM_DATA48	DRAM_D48	Not multiplexed.
DRAM_DATA49	DRAM_D49	Not multiplexed.
DRAM_DATA50	DRAM_D50	Not multiplexed.
DRAM_DATA51	DRAM_D51	Not multiplexed.
DRAM_DATA52	DRAM_D52	Not multiplexed.
DRAM_DATA53	DRAM_D53	Not multiplexed.
DRAM_DATA54	DRAM_D54	Not multiplexed.
DRAM_DATA55	DRAM_D55	Not multiplexed.
DRAM_DATA56	DRAM_D56	Not multiplexed.
DRAM_DATA57	DRAM_D57	Not multiplexed.
DRAM_DATA58	DRAM_D58	Not multiplexed.
DRAM_DATA59	DRAM_D59	Not multiplexed.
DRAM_DATA60	DRAM_D60	Not multiplexed.
DRAM_DATA61	DRAM_D61	Not multiplexed.
DRAM_DATA62	DRAM_D62	Not multiplexed.
DRAM_DATA63	DRAM_D63	Not multiplexed.
DRAM_DQM0	DRAM_DQM0	Not multiplexed.
DRAM_DQM1	DRAM_DQM1	Not multiplexed.
DRAM_DQM2	DRAM_DQM2	Not multiplexed.
DRAM_DQM3	DRAM_DQM3	Not multiplexed.
DRAM_DQM4	DRAM_DQM4	Not multiplexed.
DRAM_DQM5	DRAM_DQM5	Not multiplexed.

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
DRAM_DQM6	DRAM_DQM6	Not multiplexed.
DRAM_DQM7	DRAM_DQM7	Not multiplexed.
DRAM_ODT0	DRAM_SDODT0	Not multiplexed.
DRAM_ODT1	DRAM_SDODT1	Not multiplexed.
DRAM_RAS_B	DRAM_RAS	Not multiplexed.
DRAM_RESET	DRAM_RESET	Not multiplexed.
DRAM_SDBA0	DRAM_SDBA0	Not multiplexed.
DRAM_SDBA1	DRAM_SDBA1	Not multiplexed.
DRAM_SDBA2	DRAM_SDBA2	Not multiplexed.
DRAM_SDCKE0	DRAM_SDCKE0	Not multiplexed.
DRAM_SDCKE1	DRAM_SDCKE1	Not multiplexed.
DRAM_SDCLK0_N	DRAM_SDCLK_0_B	Not multiplexed.
DRAM_SDCLK0_P	DRAM_SDCLK_0	Not multiplexed.
DRAM_SDCLK1_N	DRAM_SDCLK_1_B	Not multiplexed.
DRAM_SDCLK1_P	DRAM_SDCLK_1	Not multiplexed.
DRAM_SDQS0_N	DRAM_SDQS0_B	Not multiplexed.
DRAM_SDQS0_P	DRAM_SDQS0	Not multiplexed.
DRAM_SDQS1_N	DRAM_SDQS1_B	Not multiplexed.
DRAM_SDQS1_P	DRAM_SDQS1	Not multiplexed.
DRAM_SDQS2_N	DRAM_SDQS2_B	Not multiplexed.
DRAM_SDQS2_P	DRAM_SDQS2	Not multiplexed.
DRAM_SDQS3_N	DRAM_SDQS3_B	Not multiplexed.
DRAM_SDQS3_P	DRAM_SDQS3	Not multiplexed.
DRAM_SDQS4_N	DRAM_SDQS4_B	Not multiplexed.
DRAM_SDQS4_P	DRAM_SDQS4	Not multiplexed.
DRAM_SDQS5_N	DRAM_SDQS5_B	Not multiplexed.
DRAM_SDQS5_P	DRAM_SDQS5	Not multiplexed.
DRAM_SDQS6_N	DRAM_SDQS6_B	Not multiplexed.
DRAM_SDQS6_P	DRAM_SDQS6	Not multiplexed.
DRAM_SDQS7_N	DRAM_SDQS7_B	Not multiplexed.
DRAM_SDQS7_P	DRAM_SDQS7	Not multiplexed.
DRAM_SDWE_B	DRAM_SDWE	Not multiplexed.
PCIE - PCI Express		
PCIE_RX_N	PCIE_RXM	Not multiplexed.
PCIE_RX_P	PCIE_RXP	Not multiplexed.
PCIE_TX_N	PCIE_TXM	Not multiplexed.
PCIE_TX_P	PCIE_TXP	Not multiplexed.
PMU - Power Management Unit		
CSI_REXT	CSI_REXT	Not multiplexed.

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
DRAM_VREF	DRAM_VREF	Not multiplexed.
DRAM_ZQPAD	ZQPAD	Not multiplexed.
DSI_REXT	DSI_REXT	Not multiplexed.
GND	GND	Not multiplexed.
HDMI_REF	HDMI_REF	Not multiplexed.
HDMI_VP	HDMI_VP	Not multiplexed.
HDMI_VPH	HDMI_VPH	Not multiplexed.
NC	NC	Not multiplexed.
NVCC_CSI	NVCC_CSI	Not multiplexed.
NVCC_DRAM	NVCC_DRAM	Not multiplexed.
NVCC_EIM0	NVCC_EIM0	Not multiplexed.
NVCC_EIM1	NVCC_EIM1	Not multiplexed.
NVCC_EIM2	NVCC_EIM2	Not multiplexed.
NVCC_ENET	NVCC_ENET	Not multiplexed.
NVCC_GPIO	NVCC_GPIO	Not multiplexed.
NVCC_JTAG	NVCC_JTAG	Not multiplexed.
NVCC_LCD	NVCC_LCD	Not multiplexed.
NVCC_LVDS_2P5	NVCC_LVDS2P5	Not multiplexed.
NVCC_MIPI	NVCC_MIPI	Not multiplexed.
NVCC_NAND	NVCC_NANDF	Not multiplexed.
NVCC_PLL	NVCC_PLL_OUT	Not multiplexed.
NVCC_RGMII	NVCC_RGMII	Not multiplexed.
NVCC_SD1	NVCC_SD1	Not multiplexed.
NVCC_SD2	NVCC_SD2	Not multiplexed.
NVCC_SD3	NVCC_SD3	Not multiplexed.
PCIE_REXT	PCIE_REXT	Not multiplexed.
PCIE_VP	PCIE_VP	Not multiplexed.
PCIE_VPH	PCIE_VPH	Not multiplexed.
PCIE_VPTX	PCIE_VPTX	Not multiplexed.
USB_H1_VBUS	USB_H1_VBUS	Not multiplexed.
USB_OTG_VBUS	USB_OTG_VBUS	Not multiplexed.
VDD_ARM_CAP	VDDARM_CAP	Not multiplexed.
VDD_ARM_IN	VDDARM_IN	Not multiplexed.
VDD_HIGH_CAP	VDDHIGH_CAP	Not multiplexed.
VDD_HIGH_IN	VDDHIGH_IN	Not multiplexed.
VDD_PU_CAP	VDDPU_CAP	Not multiplexed.
VDD_SNVS_CAP	VDD_SNVS_CAP	Not multiplexed.
VDD_SNVS_IN	VDD_SNVS_IN	Not multiplexed.
VDD_SOC_CAP	VDDSOC_CAP	Not multiplexed.

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
VDD_SOC_IN	VDDSOC_IN	Not multiplexed.
VDD_USB_CAP	VDDUSB_CAP	Not multiplexed.
PWM1 - Pulse Width Modulation		
PWM1_OUT	DISP0_DAT8 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA08
	GPIO_9 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_GPIO09
	SD1_DAT3 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3
PWM2 - Pulse Width Modulation		
PWM2_OUT	DISP0_DAT9 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA09
	GPIO_1 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_GPIO01
	SD1_DAT2 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2
PWM3 - Pulse Width Modulation		
PWM3_OUT	SD1_DAT1 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1
	SD4_DAT1 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA1
PWM4 - Pulse Width Modulation		
PWM4_OUT	SD1_CMD (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD1_CMD
	SD4_DAT2 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA2
SDMA - Smart Direct Memory Access Controller		
SDMA_EXT_EVENT0	DISP0_DAT16 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA16 IOMUXC_SDMA_EVENTS14_SELECT_INPUT
	GPIO_17 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_GPIO17 IOMUXC_SDMA_EVENTS14_SELECT_INPUT
SDMA_EXT_EVENT1	DISP0_DAT17 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA17 IOMUXC_SDMA_EVENTS15_SELECT_INPUT
	GPIO_18 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_GPIO18 IOMUXC_SDMA_EVENTS15_SELECT_INPUT
SJC - System JTAG Controller		
JTAG_DE_B	GPIO_16 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_GPIO16
JTAG_MOD	JTAG_MOD	Not multiplexed.
JTAG_TCK	JTAG_TCK	Not multiplexed.
JTAG_TDI	JTAG_TDI	Not multiplexed.
JTAG_TDO	JTAG_TDO	Not multiplexed.
JTAG_TMS	JTAG_TMS	Not multiplexed.
JTAG_TRSTB	JTAG_TRSTB	Not multiplexed.
SNVS - Secure Non-Volatile Storage		
SNVS_PMIC_ON_REQ	PMIC_ON_REQ	Not multiplexed.
SNVS_TAMPER	TAMPER	Not multiplexed.
SNVS_VIO_5	GPIO_0 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_GPIO00
SNVS_VIO_5_CTL	GPIO_18 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_GPIO18
SPDIF - Sony/Philips Digital Interface		

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
SPDIF_EXT_CLK	ENET_CRS_DV (ALT3)	IOMUXC_SW_MUX_CTL_PAD_ENET_CRS_DV
		IOMUXC_SPDIF_TX_CLK2_SELECT_INPUT
	RGMII_TXC (ALT2)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TXC
		IOMUXC_SPDIF_TX_CLK2_SELECT_INPUT
SPDIF_IN	EIM_D21 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA21
		IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT
	ENET_RX_ER (ALT3)	IOMUXC_SW_MUX_CTL_PAD_ENET_RX_ER
		IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT
	GPIO_16 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_GPIO16
		IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT
	KEY_COL3 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL3
		IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT
SPDIF_LOCK	ENET_MDIO (ALT6)	IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO
	GPIO_7 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_GPIO07
SPDIF_OUT	EIM_D22 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA22
	ENET_RXD0 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA0
	GPIO_17 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_GPIO17
	GPIO_19 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_GPIO19
SPDIF_SR_CLK	ENET_REF_CLK (ALT6)	IOMUXC_SW_MUX_CTL_PAD_ENET_REF_CLK
	GPIO_8 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_GPIO08
SRC - System Reset Controller		
SRC_BOOT_CFG00	EIM_DA0 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD00
SRC_BOOT_CFG01	EIM_DA1 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD01
SRC_BOOT_CFG02	EIM_DA2 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD02
SRC_BOOT_CFG03	EIM_DA3 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD03
SRC_BOOT_CFG04	EIM_DA4 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD04
SRC_BOOT_CFG05	EIM_DA5 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD05
SRC_BOOT_CFG06	EIM_DA6 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD06
SRC_BOOT_CFG07	EIM_DA7 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD07
SRC_BOOT_CFG08	EIM_DA8 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD08
SRC_BOOT_CFG09	EIM_DA9 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD09
SRC_BOOT_CFG10	EIM_DA10 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD10
SRC_BOOT_CFG11	EIM_DA11 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD11
SRC_BOOT_CFG12	EIM_DA12 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD12
SRC_BOOT_CFG13	EIM_DA13 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD13
SRC_BOOT_CFG14	EIM_DA14 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD14
SRC_BOOT_CFG15	EIM_DA15 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_AD15
SRC_BOOT_CFG16	EIM_A16 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR16
SRC_BOOT_CFG17	EIM_A17 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR17

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
SRC_BOOT_CFG18	EIM_A18 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR18
SRC_BOOT_CFG19	EIM_A19 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR19
SRC_BOOT_CFG20	EIM_A20 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR20
SRC_BOOT_CFG21	EIM_A21 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR21
SRC_BOOT_CFG22	EIM_A22 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR22
SRC_BOOT_CFG23	EIM_A23 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR23
SRC_BOOT_CFG24	EIM_A24 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR24
SRC_BOOT_CFG25	EIM_WAIT (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_WAIT_B
SRC_BOOT_CFG26	EIM_LBA (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_LBA_B
SRC_BOOT_CFG27	EIM_EB0 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB0_B
SRC_BOOT_CFG28	EIM_EB1 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB1_B
SRC_BOOT_CFG29	EIM_RW (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_RW
SRC_BOOT_CFG30	EIM_EB2 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB2_B
SRC_BOOT_CFG31	EIM_EB3 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB3_B
SRC_BOOT_MODE0	BOOT_MODE0	Not multiplexed.
SRC_BOOT_MODE1	BOOT_MODE1	Not multiplexed.
SRC_ONOFF	ONOFF	Not multiplexed.
SRC_POR_B	POR_B	Not multiplexed.
UART1 - Universal Asynchronous Receiver/Transmitter		
UART1_CTS_B	EIM_D19 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA19 IOMUXC_UART1_UART_RTS_B_SELECT_INPUT
	SD3_DAT0 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA0 IOMUXC_UART1_UART_RTS_B_SELECT_INPUT
UART1_DCD_B	EIM_D23 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA23
UART1_DSR_B	EIM_D25 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA25
UART1_DTR_B	EIM_D24 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA24
UART1_RI_B	EIM_EB3 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB3_B
UART1_RTS_B	EIM_D20 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA20 IOMUXC_UART1_UART_RTS_B_SELECT_INPUT
	SD3_DAT1 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA1 IOMUXC_UART1_UART_RTS_B_SELECT_INPUT
UART1_RX_DATA	CSI0_DAT11 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA11 IOMUXC_UART1_UART_RX_DATA_SELECT_INPUT
	SD3_DAT6 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA6 IOMUXC_UART1_UART_RX_DATA_SELECT_INPUT
UART1_TX_DATA	CSI0_DAT10 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA10 IOMUXC_UART1_UART_RX_DATA_SELECT_INPUT
	SD3_DAT7 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA7 IOMUXC_UART1_UART_RX_DATA_SELECT_INPUT

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
UART2 - Universal Asynchronous Receiver/Transmitter		
UART2_CTS_B	EIM_D28 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA28 IOMUXC_UART2_UART_RTS_B_SELECT_INPUT
	SD3_CMD (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD3_CMD IOMUXC_UART2_UART_RTS_B_SELECT_INPUT
	SD4_DAT6 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA6 IOMUXC_UART2_UART_RTS_B_SELECT_INPUT
UART2_RTS_B	EIM_D29 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA29 IOMUXC_UART2_UART_RTS_B_SELECT_INPUT
	SD3_CLK (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD3_CLK IOMUXC_UART2_UART_RTS_B_SELECT_INPUT
	SD4_DAT5 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA5 IOMUXC_UART2_UART_RTS_B_SELECT_INPUT
UART2_RX_DATA	EIM_D27 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA27 IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT
	GPIO_8 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_GPIO08 IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT
	SD3_DAT4 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA4 IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT
	SD4_DAT4 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA4 IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT
UART2_TX_DATA	EIM_D26 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA26 IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT
	GPIO_7 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_GPIO07 IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT
	SD3_DAT5 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA5 IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT
	SD4_DAT7 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA7 IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT
UART3 - Universal Asynchronous Receiver/Transmitter		
UART3_CTS_B	EIM_D23 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA23 IOMUXC_UART3_UART_RTS_B_SELECT_INPUT
	EIM_D30 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA30 IOMUXC_UART3_UART_RTS_B_SELECT_INPUT
	SD3_DAT3 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA3 IOMUXC_UART3_UART_RTS_B_SELECT_INPUT
UART3_RTS_B	EIM_D31 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA31 IOMUXC_UART3_UART_RTS_B_SELECT_INPUT

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
	EIM_EB3 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_EB3_B
		IOMUXC_UART3_UART_RTS_B_SELECT_INPUT
	SD3_RST (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD3_RESET
		IOMUXC_UART3_UART_RTS_B_SELECT_INPUT
UART3_RX_DATA	EIM_D25 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA25
		IOMUXC_UART3_UART_RX_DATA_SELECT_INPUT
	SD4_CLK (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD4_CLK
		IOMUXC_UART3_UART_RX_DATA_SELECT_INPUT
UART3_TX_DATA	EIM_D24 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA24
		IOMUXC_UART3_UART_RX_DATA_SELECT_INPUT
	SD4_CMD (ALT2)	IOMUXC_SW_MUX_CTL_PAD_SD4_CMD
		IOMUXC_UART3_UART_RX_DATA_SELECT_INPUT
UART4 - Universal Asynchronous Receiver/Transmitter		
UART4_CTS_B	CSI0_DAT17 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA17
		IOMUXC_UART4_UART_RTS_B_SELECT_INPUT
UART4_RTS_B	CSI0_DAT16 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA16
		IOMUXC_UART4_UART_RTS_B_SELECT_INPUT
UART4_RX_DATA	CSI0_DAT13 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA13
		IOMUXC_UART4_UART_RX_DATA_SELECT_INPUT
	KEY_ROW0 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW0
		IOMUXC_UART4_UART_RX_DATA_SELECT_INPUT
UART4_TX_DATA	CSI0_DAT12 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA12
		IOMUXC_UART4_UART_RX_DATA_SELECT_INPUT
	KEY_COL0 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL0
		IOMUXC_UART4_UART_RX_DATA_SELECT_INPUT
UART5 - Universal Asynchronous Receiver/Transmitter		
UART5_CTS_B	CSI0_DAT19 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA19
		IOMUXC_UART5_UART_RTS_B_SELECT_INPUT
	KEY_ROW4 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW4
		IOMUXC_UART5_UART_RTS_B_SELECT_INPUT
UART5_RTS_B	CSI0_DAT18 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA18
		IOMUXC_UART5_UART_RTS_B_SELECT_INPUT
	KEY_COL4 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL4
		IOMUXC_UART5_UART_RTS_B_SELECT_INPUT
UART5_RX_DATA	CSI0_DAT15 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA15
		IOMUXC_UART5_UART_RX_DATA_SELECT_INPUT
	KEY_ROW1 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW1
		IOMUXC_UART5_UART_RX_DATA_SELECT_INPUT

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
UART5_TX_DATA	CSI0_DAT14 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA14
		IOMUXC_UART5_UART_RX_DATA_SELECT_INPUT
	KEY_COL1 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL1
		IOMUXC_UART5_UART_RX_DATA_SELECT_INPUT
USB - Universal Serial Bus Controller		
USB_H1_DN	USB_H1_DN	Not multiplexed.
USB_H1_DP	USB_H1_DP	Not multiplexed.
USB_H1_OC	EIM_D30 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA30
		IOMUXC_USB_H1_OC_SELECT_INPUT
	GPIO_3 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_GPIO03
		IOMUXC_USB_H1_OC_SELECT_INPUT
USB_H1_PWR	EIM_D31 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA31
	GPIO_0 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_GPIO00
USB_H1_PWR_CTL_WAKE	KEY_COL2 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL2
USB_H2_DATA	RGMII_TXC (ALT0)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TXC
USB_H2_STROBE	RGMII_TX_CTL (ALT0)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TX_CTL
USB_H3_DATA	RGMII_RX_CTL (ALT0)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RX_CTL
USB_H3_STROBE	RGMII_RXC (ALT0)	IOMUXC_SW_MUX_CTL_PAD_RGMII_RXC
USB_OTG_CHD_B	USB_OTG_CHD_B	Not multiplexed.
USB_OTG_DN	USB_OTG_DN	Not multiplexed.
USB_OTG_DP	USB_OTG_DP	Not multiplexed.
USB_OTG_HOST_MODE	GPIO_7 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_GPIO07
USB_OTG_ID	ENET_RX_ER (ALT0)	IOMUXC_SW_MUX_CTL_PAD_ENET_RX_ER
		IOMUXC_ANALOG_USB_OTG_ID_SELECT_INPUT
	GPIO_1 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_GPIO01
		IOMUXC_ANALOG_USB_OTG_ID_SELECT_INPUT
USB_OTG_OC	EIM_D21 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA21
		IOMUXC_USB_OTG_OC_SELECT_INPUT
	KEY_COL4 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL4
		IOMUXC_USB_OTG_OC_SELECT_INPUT
USB_OTG_PWR	EIM_D22 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_EIM_DATA22
	KEY_ROW4 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW4
USB_OTG_PWR_CTL_WAKE	GPIO_8 (ALT7)	IOMUXC_SW_MUX_CTL_PAD_GPIO08
USDHC1 - Ultra Secured Digital Host Controller		
SD1_CD_B	GPIO_1 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_GPIO01
SD1_CLK	SD1_CLK (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD1_CLK
		IOMUXC_USDHC1_CARD_CLK_IN_SELECT_INPUT
SD1_CMD	SD1_CMD (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD1_CMD
SD1_DATA0	SD1_DAT0 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
SD1_DATA1	SD1_DAT1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1
SD1_DATA2	SD1_DAT2 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2
SD1_DATA3	SD1_DAT3 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3
SD1_DATA4	NANDF_D0 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA00
SD1_DATA5	NANDF_D1 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA01
SD1_DATA6	NANDF_D2 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA02
SD1_DATA7	NANDF_D3 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA03
SD1_LCTL	GPIO_16 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_GPIO16
SD1_VSELECT	KEY_COL1 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_KEY_COL1
	KEY_ROW3 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW3
SD1_WP	DI0_PIN4 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN04 IOMUXC_USDHC1_WP_ON_SELECT_INPUT
	GPIO_9 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_GPIO09 IOMUXC_USDHC1_WP_ON_SELECT_INPUT
USDHC2 - Ultra Secured Digital Host Controller		
SD2_CD_B	GPIO_4 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_GPIO04
SD2_CLK	SD2_CLK (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD2_CLK IOMUXC_USDHC2_CARD_CLK_IN_SELECT_INPUT
SD2_CMD	SD2_CMD (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD2_CMD
SD2_DATA0	SD2_DAT0 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0
SD2_DATA1	SD2_DAT1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1
SD2_DATA2	SD2_DAT2 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2
SD2_DATA3	SD2_DAT3 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3
SD2_DATA4	NANDF_D4 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA04
SD2_DATA5	NANDF_D5 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA05
SD2_DATA6	NANDF_D6 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA06
SD2_DATA7	NANDF_D7 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_NAND_DATA07
SD2_LCTL	GPIO_6 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_GPIO06
SD2_VSELECT	KEY_ROW1 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW1
	KEY_ROW2 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW2
SD2_WP	GPIO_2 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_GPIO02
USDHC3 - Ultra Secured Digital Host Controller		
SD3_CLK	SD3_CLK (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD3_CLK IOMUXC_USDHC3_CARD_CLK_IN_SELECT_INPUT
SD3_CMD	SD3_CMD (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD3_CMD
SD3_DATA0	SD3_DAT0 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA0
SD3_DATA1	SD3_DAT1 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA1
SD3_DATA2	SD3_DAT2 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA2
SD3_DATA3	SD3_DAT3 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA3

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
SD3_DATA4	SD3_DAT4 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA4
SD3_DATA5	SD3_DAT5 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA5
SD3_DATA6	SD3_DAT6 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA6
SD3_DATA7	SD3_DAT7 (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD3_DATA7
SD3_RESET	SD3_RST (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD3_RESET
SD3_VSELECT	GPIO_18 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_GPIO18
	NANDF_CS1 (ALT2)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS1_B
USDHC4 - Ultra Secured Digital Host Controller		
SD4_CLK	SD4_CLK (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD4_CLK IOMUXC_USDHC4_CARD_CLK_IN_SELECT_INPUT
SD4_CMD	SD4_CMD (ALT0)	IOMUXC_SW_MUX_CTL_PAD_SD4_CMD
SD4_DATA0	SD4_DAT0 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA0
SD4_DATA1	SD4_DAT1 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA1
SD4_DATA2	SD4_DAT2 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA2
SD4_DATA3	SD4_DAT3 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA3
SD4_DATA4	SD4_DAT4 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA4
SD4_DATA5	SD4_DAT5 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA5
SD4_DATA6	SD4_DAT6 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA6
SD4_DATA7	SD4_DAT7 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_SD4_DATA7
SD4_RESET	NANDF_ALE (ALT1)	IOMUXC_SW_MUX_CTL_PAD_NAND_ALE
SD4_VSELECT	NANDF_CS1 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_NAND_CS1_B
WDOG1 - Watchdog Timer		
WDOG1_B	DISP0_DAT8 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA08
	GPIO_9 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_GPIO09
	SD1_DAT2 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2
WDOG1_RESET_B_DEB	SD1_DAT2 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2
WDOG2 - Watchdog Timer		
WDOG2_B	DISP0_DAT9 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA09
	GPIO_1 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_GPIO01
	SD1_DAT3 (ALT4)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3
WDOG2_RESET_B_DEB	SD1_DAT3 (ALT6)	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3
XTALOSC - Crystal Oscillator		
XTALOSC_CLK1_N	CLK1_N	Not multiplexed.
XTALOSC_CLK1_P	CLK1_P	Not multiplexed.
XTALOSC_CLK2_N	CLK2_N	Not multiplexed.
XTALOSC_CLK2_P	CLK2_P	Not multiplexed.
XTALOSC_REF_CLK_24M	GPIO_3 (ALT3)	IOMUXC_SW_MUX_CTL_PAD_GPIO03
	RGMII_TXC (ALT7)	IOMUXC_SW_MUX_CTL_PAD_RGMII_TXC
XTALOSC_REF_CLK_32K	GPIO_8 (ALT1)	IOMUXC_SW_MUX_CTL_PAD_GPIO08

Table continues on the next page...

Table 4-2. Muxing Options (continued)

Signal	Pad (Mode)	Mux/Input Select Registers
XTALOSC_RTC_XTALI	RTC_XTALI	Not multiplexed.
XTALOSC_RTC_XTALO	RTC_XTALO	Not multiplexed.
XTALOSC_XTALI	XTALI	Not multiplexed.
XTALOSC_XTALO	XTALO	Not multiplexed.

Chapter 5
Fusemap

5.1 Boot Fusemap

The following section details the various modes and selection of the required boot devices. A separate map is given for each and every boot device. The device select is specified by BOOT_CFG1[7:3] fuses listed in Table 5-1. In Table 5-2, the "Address Offset" means the address offset compared with the OCOTP_CTRL base address, which is 0x21B_C000.

Table 5-1. Boot Device Select

Boot Device	BOOT _CFG1[7]	BOOT_CFG1[6]	BOOT_CFG1[5]	BOOT_CFG1[4]	BOOT_CFG1[3]
EIM (Table 5-3)	0	0	0	0	Memory Type: 0 - NOR Flash 1 - OneNAND
Serial ROM (Table 5-4)	0	0	1	1	X
SD/eSD (Table 5-5)	0	1	0	X	X
MMC/eMMC (Table 5-6)	0	1	1	X	X
NAND Flash (Table 5-7)	1	X	X	X	X

NOTE

Fuses marked as “Reserved” are reserved for NXP internal (and future) use only. Customers should not attempt to burn these, as the IC behavior may be unpredictable. The reserved fuses can be read as either ‘0’ or ‘1’.

Table 5-2. Lock Fuses

Address Offset	7	6	5	4	3	2	1	0
0x400[7:0]	Reserved	SJC_RESP_LOCK WRP,OP,RDP	MEM_TRIM_LOCK 1x - OP x1 - WP		BOOT_CFG_LOCK (Locking rows: 0x450-0x470) 1x - OP x1 - WP		TESTER_LOCK 1x - OP x1 - WP	
00x400[15:8]	Reserved	SRK_LOCK RD,WR,OP	GP2_LOCK 1x - OP x1 - WP		GP1_LOCK 1x - OP x1 - WP		MAC_ADDR_LOCK 1x - OP x1 - WP	
0x400[23:16]	Reserved	MISC_CONF_LOCK 1 - WP + OP of MISC_CONF	Reserved		ANALOG_LOCK 1x - OP x1 - WP		Reserved	
0x400[31:24]	Reserved							

Table 5-3. EIM Boot Fusemap

Address Offset	7	6	5	4	3	2	1	0
0x450[7:0] (BOOT_CFG1)	0	0	0	0	Memory Type: 0 - NOR Flash 1 - OneNAND	Reserved	Reserved	Reserved
0x450[15:8] (BOOT_CFG2)	Muxing Scheme: 00 - A/D16 (HW Default in external boot) 01 - A+DH 10 - A+DL 11 - Reserved		OneNand Page Size: 00 - 1KB 01 - 2KB 10 - 4KB 11 - Reserved		Reserved	Reserved	Reserved	Reserved
0x450[23:16] (BOOT_CFG3)	L1 I-Cache DISABLE	BT_MMU_DISABLE	DDR Memory Map default config 00 - Single DDR channel 01 - Fixed 2x32 map 10 - 4KB Interleaving Enabled 11 - Illegal		Reserved	Boot Frequencies (ARM/ DDR) 0 - 792 / 396 MHz 1 - 396 / 352 MHz	Reserved	Disable SDMMC Manufacture mode 0 - Enable 1 - Disable
0x450[31:24] (BOOT_CFG4)	Infinite Loop (for debug) 0 - Disabled	EEPROM Recovery Enable	eCSPI chip select: 00 - ECSPIx_SS0 (default)		eCSPI Addressing:	Port Select: 000 - eCSPI1 001 - eCSPI2		

Table continues on the next page...

Table 5-3. EIM Boot Fusemap (continued)

Address Offset	7	6	5	4	3	2	1	0
	1 - Enabled	0 - Disabled 1 - Enabled Fuse is Reserved for 'Serial-ROM' boot mode	01 - ECSPIx_SS1 10 - ECSPIx_SS2 11 - ECSPIx_SS3		0 - 2-bytes (16-bit) 1 - 3-bytes (24-bit)	010 - eCSPI3 011 - eCSPI4 100 - Reserved 101 - I2C1 110 - I2C2 111 - I2C3		
0x460[7:0]	Reserved			BT_FUSE_SEL	DIR_BT_DIS	SDP_READ_DISABLE (Silicon Revision 1.4 or later only)	SEC_CONFIG[1]	SDP_DISABLE (Silicon Revision 1.4 or later only)
0x460[15:8]	Reserved							
0x460[23:16]	JTAG_SMODE[1:0]		WDOG_ENABLE 0 - Disabled 1 - Enabled	SJC_DISABLE	Reserved			
0x460[31:24]	FORCE_INTERNAL_BOOT (Silicon Revision 1.4 or later only)	eMMC_RESET_EN	SDMMC_HYS_EN	TZASC_ENABLE	JTAG_HEO	KTE	Reserved	
0x470[7:0]	NAND_READ_CMD_CODE1[7:0]							
0x470[15:8]	NAND_READ_CMD_CODE2[7:0]							
0x470[23:16]	Reserved	LPB_BOOT (Core / DDR-Bus) 00 - LPB Disable 01 - 1 GPIO (def freq) 10 - Div by2 11 - Div by 4		BT_LPB_POLARITY (GPIO polarity) 0 - Active High 1 -Active Low	Reserved			
0x470[31:24]	Reserved	MMC_DLL_DLY[6:0]						
0x6D0[7:0]	Reserved (locked by MISC_CONF_LOCK)		PAD_SETTINGS[5:0]					

Table 5-4. Serial-ROM Boot Fusemap

Address Offset	7	6	5	4	3	2	1	0
0x450[7:0] (BOOT_CFG1)	0	0	1	1	Reserved	Reserved	Reserved	Reserved

Table continues on the next page...

Table 5-4. Serial-ROM Boot Fusemap (continued)

Address Offset	7	6	5	4	3	2	1	0
0x450[15:8] (BOOT_CFG2)	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	
0x450[23:16] (BOOT_CFG3)	L1 I-Cache DISABLE	BT_MMU_DISABLE	DDR Memory Map default config 00 - Single DDR channel 01 - Fixed 2x32 map 10 - 4KB Interleaving Enabled 11 - Illegal		Reserved	Boot Frequencies (ARM/DDR) 0 - 792 / 396 MHz 1 - 396 / 352 MHz	Reserved	Reserved
0x450[31:24] (BOOT_CFG4)	Infinite Loop (for debug) 0 - Disabled 1 - Enabled	EEPROM Recovery Enable 0 - Disabled 1 - Enabled Fuse is Reserved for 'Serial-ROM' Boot mode	eCSPI chip select: 00 - ECSPIx_SS0 (default) 01 - ECSPIx_SS1 10 - ECSPIx_SS2 11 - ECSPIx_SS3		eCSPI Addressing: 0 - 2-bytes (16-bit) 1 - 3-bytes (24-bit)	Port Select: 000 - eCSPI1 001 - eCSPI2 010 - eCSPI3 011 - eCSPI4 100 - Reserved 101 - I2C1 110 - I2C2 111 - I2C3		
0x460[7:0]	Reserved			BT_FUSE_SEL	DIR_BT_DIS	SDP_READ_DISABLE (Silicon Revision 1.4 or later only)	SEC_CONFIG[1]	SDP_DISABLE (Silicon Revision 1.4 or later only)
0x460[15:8]	Reserved							
0x460[23:16]	JTAG_SMODE[1:0]		WDOG_ENABLE 0 - Disabled 1 - Enabled	SJC_DISABLE	Reserved			
0x460[31:24]	FORCE_INTERNAL_BOOT (Silicon Revision 1.4 or later only)	Reserved		TZASC_ENABLE	JTAG_HEO	KTE	Reserved	
0x470[7:0]	NAND_READ_CMD_CODE1[7:0]							
0x470[15:8]	NAND_READ_CMD_CODE2[7:0]							
0x470[23:16]	Reserved	LPB_BOOT (Core / DDR-Bus) 00 - LPB Disable 01 - 1 GPIO (def freq) 10 - Div by2		BT_LPB_POLARITY (GPIO polarity)	Reserved			

Table continues on the next page...

Table 5-4. Serial-ROM Boot Fusemap (continued)

Address Offset	7	6	5	4	3	2	1	0
		11 - Div by 4		0 - Active High 1 -Active Low				
0x470[31:24]	Reserved	MMC_DLL_DLY[6:0]						
0x6D0[7:0]	Reserved (locked by MISC_CONF_LOCK)	PAD_SETTINGS[5:0]						

Table 5-5. SD/eSD Boot Fusemap

Address Offset	7	6	5	4	3	2	1	0
0x450[7:0] (BOOT_CFG1)	0	1	0	Fast Boot: 0 - Regular 1 - Fast Boot	SD/SDXC Speed 00 - Normal/SDR12 01 - High/SDR25 10 - SDR50 11 - SDR104		SD Power Cycle Enable 0 - No power cycle 1 - Enabled via USDHC_RST pad (uSDHC3 & 4 only)	Reserved (Feedback Clock Source Sel 0 - Internal loopback 1 - External pad
0x450[15:8] (BOOT_CFG2)	SD Calibration Step 00 - 1 delay cell 01 - 1 delay cell 10 - 2 delay cell 11 - 3 delay cell		Bus Width: 0 - 1-bit 1 - 4-bit	Port Select: 00 - uSDHC1 01 - uSDHC2 10 - uSDHC3 11 - uSDHC4		DLL Override: 0 - Boot ROM default 1 - Apply value per fuse field MMC_DLL_DLY[3:0]	Boot Acknowledge Disable/ Pull-Down 0 - Use default SD pad settings during power cycle Set pull-down on SD pads during power cycle (used only when "SD Power Cycle Enable" enabled)	Override Pad Settings (using PAD_SETTINGS value)
0x450[23:16] (BOOT_CFG3)	L1 I-Cache DISABLE	BT_MMU_DISABLE	DDR Memory Map default config 00 - Single DDR channel 01 - Fixed 2x32 map	Reserved		Boot Frequencies (ARM/DDR) 0 - 792 / 396 MHz	Reserved	Reserved

Table continues on the next page...

Table 5-5. SD/eSD Boot Fusemap (continued)

Address Offset	7	6	5	4	3	2	1	0
			10 - 4KB Interleaving Enabled 11 - Illegal			1 - 396 / 352 MHz		
0x450[31:24] (BOOT_CFG4)	Infinite Loop (for debug) 0 - Disabled 1 - Enabled	EEPROM Recovery Enable 0 - Disabled 1 - Enabled Fuse is Reserved for 'Serial-ROM' Boot mode	eCSPI chip select: 00 - ECSPIx_SS0 (default) 01 - ECSPIx_SS1 10 - ECSPIx_SS2 11 - ECSPIx_SS3		eCSPI Addressing: 0 - 2-bytes (16-bit) 1 - 3-bytes (24-bit)	Port Select: 000 - eCSPI1 001 - eCSPI2 010 - eCSPI3 011 - eCSPI4 100 - Reserved 101 - I2C1 110 - I2C2 111 - I2C3		
0x460[7:0]	Reserved			BT_FUSE_SEL	DIR_BT_DIS	SDP_READ_DISABLE (Silicon Revision 1.4 or later only)	SEC_CONFIG[1]	SDP_DISABLE (Silicon Revision 1.4 or later only)
0x460[15:8]	Reserved							
0x460[23:16]	JTAG_SMODE[1:0]		WDOG_ENABLE 0 - Disabled 1 - Enabled	SJC_DISABLE	Reserved			
0x460[31:24]	FORCE_INTERNAL_BOOT (Silicon Revision 1.4 or later only)	eMMC_RESET_EN	SDMMC_HYS_EN	TZASC_ENABLE	JTAG_HEO	KTE	Reserved	
0x470[7:0]	NAND_READ_CMD_CODE1[7:0]							
0x470[15:8]	NAND_READ_CMD_CODE2[7:0]							
0x470[23:16]	Reserved	LPB_BOOT (Core / DDR-Bus) 00 - LPB Disable 01 - 1 GPIO (def freq) 10 - Div by2 11 - Div by 4		BT_LPB_POLARITY (GPIO polarity) 0 - Active High 1 -Active Low	Reserved			
0x470[31:24]	Reserved	MMC_DLL_DLY[6:0]						
0x6D0[7:0]	Reserved (locked by MISC_CONF_LOCK)		PAD_SETTINGS[5:0]					

Table 5-6. MMC/eMMC Boot Fusemap

Address Offset	7	6	5	4	3	2	1	0
0x450[7:0] (BOOT_CFG1)	0	1	1	Fast Boot: 0 - Regular 1 - Fast Boot	SD/MMC Speed 0 - High 1- Normal	Reserved	Reserved	Reserved
0x450[15:8] (BOOT_CFG2)	Bus Width: 000 - 1-bit 001 - 4-bit 010 - 8-bit 101 - 4-bit DDR (MMC 4.4) 110 - 8-bit DDR (MMC 4.4) Else - Reserved			Port Select: 00 - uSDHC1 01 - uSDHC2 10 - uSDHC3 11 - uSDHC4		DLL Override: 0 - Boot ROM default 1 - Apply value per fuse field MMC_DLL_DLY[3:0]	Fast Boot Acknowledge Disable: 0 - Boot Ack Enabled 1 - Boot Ack Disabled	Override Pad Settings (using PAD_SETTINGS value)
0x450[23:16] (BOOT_CFG3)	L1 I-Cache DISABLE	BT_MMU_DISABLE	DDR Memory Map default config 00 - Single DDR channel 01 - Fixed 2x32 map 10 - 4KB Interleaving Enabled 11 - Illegal		Reserved	Boot Frequencies (ARM/DDR) 0 - 792 / 396 MHz 1 - 396 / 352 MHz	Reserved	Reserved
0x450[31:24] (BOOT_CFG4)	Infinite Loop (for debug) 0 - Disabled 1 - Enabled	EEPROM Recovery Enable 0 - Disabled 1 - Enabled Fuse is Reserved for 'Serial-ROM' Boot mode	eCSPI chip select: 00 - ECSPIx_SS0 (default) 01 - ECSPIx_SS1 10 - ECSPIx_SS2 11 - ECSPIx_SS3		eCSPI Addressing: 0 - 2-bytes (16-bit) 1 - 3-bytes (24-bit)	Port Select: 000 - eCSPI1 001 - eCSPI2 010 - eCSPI3 011 - eCSPI4 100 - Reserved 101 - I2C1 110 - I2C2 111 - I2C3		
0x460[7:0]	Reserved			BT_FUSE_SEL	DIR_BT_DIS	SDP_READ_DISABLE (Silicon Revision 1.4 or later only)	SEC_CONFIG[1]	SDP_DISABLE (Silicon Revision 1.4 or later only)
0x460[15:8]	Reserved							
0x460[23:16]	JTAG_SMODE[1:0]		WDOG_ENABLE 0 - Disabled 1 - Enabled	SJC_DISABLE	Reserved			
0x460[31:24]	FORCE_INTERNAL_B	eMMC_RESET_EN	SDMMC_HYS_EN	TZASC_ENABLE	JTAG_HEO	KTE	Reserved	

Table continues on the next page...

Table 5-6. MMC/eMMC Boot Fusemap (continued)

Address Offset	7	6	5	4	3	2	1	0
	OOT (Silicon Revision 1.4 or later only)							
0x470[7:0]	NAND_READ_CMD_CODE1[7:0]							
0x470[15:8]	NAND_READ_CMD_CODE2[7:0]							
0x470[23:16]	Reserved	LPB_BOOT (Core / DDR-Bus) 00 - LPB Disable 01 - 1 GPIO (def freq) 10 - Div by2 11 - Div by 4		BT_LPB_POLARITY (GPIO polarity) 0 - Active High 1 -Active Low	Reserved			
0x470[31:24]	Reserved	MMC_DLL_DLY[6:0]						
0x6D0[7:0]	Reserved (locked by MISC_CONF_LOCK)		PAD_SETTINGS[5:0]					

Table 5-7. NAND Boot Fusemap

Address Pffset	7	6	5	4	3	2	1	0
0x450[7:0] (BOOT_CFG1)	1	Reserved	BT_TOGGLEMODE 0 - Raw Nand 1 - Toggle-Mode NAND	Override Pad Settings (using PAD_SETTINGS value)	Nand Number Of Devices: 00 - 1 01 - 2 10 - 4 11 - Reserved		Nand_Row_address_bytes: 00 - 3 01 - 2 10 - 4 11 - 5	
0x450[15:8] (BOOT_CFG2)	Toggle Mode 33MHz Preamble Delay, Read Latency: 000 - 16 GPMICLK cycles 001 - 1 GPMICLK cycles 010 - 2 GPMICLK cycles 011 - 3 GPMICLK cycles 100 - 4 GPMICLK cycles 101 - 5 GPMICLK cycles 110 - 6 GPMICLK cycles 111 - 7 GPMICLK cycles			BOOT_SEARCH_COUNT: 00 - 2 01 - 2 10 - 4 11 - 8		Pages In Block: 00 - 128 01 - 64 10 - 32 11 - 256		Reset time 0 - 12ms 1 - 22ms (LBA Nand)
0x450[23:16] (BOOT_CFG3)	L1 I-Cache DISABLE	BT_MMU_DISABLE	DDR Memory Map default config 00 - Single DDR channel 01 - Fixed 2x32 map	Reserved	Boot Frequencies (ARM/DDR) 0 - 792 / 396 MHz		Reserved	Reserved

Table continues on the next page...

Table 5-7. NAND Boot Fusemap (continued)

Address Pffset	7	6	5	4	3	2	1	0
			10 - 4KB Interleaving Enabled 11 - Illegal			1 - 396 / 352 MHz		
0x450[31:24] (BOOT_CFG4)	Infinite Loop (for debug) 0 - Disabled 1 - Enabled	EEPROM Recovery Enable 0 - Disabled 1 - Enabled Fuse is Reserved for 'Serial-ROM' Boot mode	eCSPI chip select: 00 - ECSPIx_SS0 (default) 01 - ECSPIx_SS1 10 - ECSPIx_SS2 11 - ECSPIx_SS3		eCSPI Addressing: 0 - 2-bytes (16-bit) 1 - 3-bytes (24-bit)	Port Select: 000 - eCSPI1 001 - eCSPI2 010 - eCSPI3 011 - eCSPI4 100 - Reserved 101 - I2C1 110 - I2C2 111 - I2C3		
0x460[7:0]	Reserved			BT_FUSE_SEL	DIR_BT_DIS	SDP_READ_DISABLE (Silicon Revision 1.4 or later only)	SEC_CONFIG[1]	SDP_DISABLE (Silicon Revision 1.4 or later only)
0x460[15:8]	Reserved							
0x460[23:16]	JTAG_SMODE[1:0]		WDOG_ENABLE 0 - Disabled 1 - Enabled	SJC_DISABLE	Reserved			
0x460[31:24]	FORCE_INTERNAL_BOOT (Silicon Revision 1.4 or later only)	eMMC_RESET_EN	SDMMC_HYS_EN	TZASC_ENABLE	JTAG_HEO	KTE	Reserved	
0x470[7:0]	NAND_READ_CMD_CODE1[7:0]							
0x470[15:8]	NAND_READ_CMD_CODE2[7:0]							
0x470[23:16]	Reserved	LPB_BOOT (Core / DDR-Bus) 00 - LPB Disable 01 - 1 GPIO (def freq) 10 - Div by2 11 - Div by 4		BT_LPB_POLARITY (GPIO polarity) 0 - Active High 1 -Active Low	Reserved			
0x470[31:24]	Reserved	MMC_DLL_DLY[6:0]						
0x6D0[7:0]	Reserved (locked by MISC_CONF_LOCK)		PAD_SETTINGS[5:0]					

5.2 Fusemap Description Table

This section covers the fusemap descriptions of the chip.

Table 5-8. Fusemap Descriptions

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Settings	Locked by
0x400[1:0]	TESTER_LOCK	2	Provides Locking of various fuse bits.	00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded 1x - Override Protect (OP) x1 - Write Protect (WP) 11 - Both OP and WP	N/A
0x400[3:2]	BOOT_CFG_LOCK	2	Perform lock on BOOT related fuses.	00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded 1x - Override Protect (OP) x1 - Write Protect (WP) 11 - Both OP and WP	N/A
0x400[5:4]	MEM_TRIM_LOCK	2	Trimming fuses. Burned on the tester or by customer before the final product shipment.	00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded. 1x - Override Protect (OP) x1 - Write Protect (WP) 11 - Both OP and WP	N/A
0x400[6]	SJC_RESP_LOCK	1	Lock bit for JTAG response fuses. (SJC_RESP)	0 - Unlock	N/A

Table continues on the next page...

Table 5-8. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Settings	Locked by
				1 - Read Protect + Write Protect + Override Protect and Program protect lock	
0x400[9:8]]	MAC_ADDR_LOCK	2	Lock MAC_ADDR fuses.	00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded. 1x - Override Protect (OP) x1 - Write Protect (WP) 11 - Both OP and WP	N/A
0x400[11:10]	GP1_LOCK	2	Lock for General Purpose fuse register #1 (GP1)	00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded. 1x - Override Protect (OP) x1 - Write Protect (WP) 11 - Both OP and WP	N/A
0x400[13:12]	GP2_LOCK	2	Lock for General Purpose fuse register #2 (GP2)	00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded. 1x - Override Protect (OP) x1 - Write Protect (WP) 11 - Both OP and WP	N/A
0x400[14]	SRK_LOCK	1	Locking SRK_HASH[255:0]	0 - Unlock 1 - Write Protect + Override Protect	N/A
0x400[19:18]	ANALOG_LOCK	2	Lock bit for various fuses (Addresses: 0x4D0-0x4F0)	00 - Unlock (The controlled field can be read, sensed,	N/A

Table continues on the next page...

Table 5-8. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Settings	Locked by
				burned or overridden in the corresponded. 1x - Override Protect (OP) x1 - Write Protect (WP) 11 - Both OP and WP	
0x400[22]	MISC_CONF_LOCK	1	Locking of MISC_CONFIG fuses (address 0x6D0)	0 - Unlock 1 - WP + OP of MISC_CONF	N/A
0x410[31:0]	SJC_CHALL[31:0] / UNIQUE_ID[31:0]	32	Device Unique ID, also be used for SJC Challenge phrase. (bits 31-0)	Random, by test program.	TESTER_LOCK
0x420[31:0]	SJC_CHALL/ UNIQUE_ID[42:32]	32	Device Unique ID, also be used for SJC Challenge phrase. (bits 63-32)	Random, by test program.	TESTER_LOCK
0x430[21:20]	NUM_CORES	2	Indicates the type of device: single or dual cores.	00 - Dual Cores. 01 - Single core 10 - Reserved 11 - Reserved	TESTER_LOCK
0x430[24]	Reserved	1	Reserved	Reserved	TESTER_LOCK
0x440[17:16]	SoC core speed	2	SPEED_GRADING[1:0]	00 - 800MHz (Industrial grade) 01 - 852MHz (Automotive grade) 10 - 1000 MHz 11 - 1200 MHZ	TESTER_LOCK
0x450[7:0]	BOOT_CFG1	8	BOOT configuration register #1, Usage varies, depending on selected boot device.	See Table 5-1 for details.	BOOT_CFG_LOCK
0x450[15:8]	BOOT_CFG2	8	BOOT configuration register #2, Usage varies, depending on selected boot device.	Refer to the respective boot fusemap tables for details.	BOOT_CFG_LOCK

Table continues on the next page...

Table 5-8. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Settings	Locked by
0x450[23:16]	BOOT_CFG3	8	BOOT configuration register #3	Refer to the respective boot fusemap tables for details.	BOOT_CFG_LOCK
0x450[31:24]	BOOT_CFG4	8	BOOT configuration register #4	Refer to the respective boot fusemap tables for details.	BOOT_CFG_LOCK
0x460[0]	SDP_DISABLE	1	Enable/disable serial download support This fuse is available only on silicon revision 1.4 or later	0 – Serial download supported (default) 1 – No serial download support. Recommended for security-enabled applications.	BOOT_CFG_LOCK
0x460[1]	SEC_CONFIG[1]	1	Select polarity of GPIO pin used for LPB indication.	Out of FAB state: 0 - Open - allows any code to be flashed and executed, even if it has no valid signature. 1 - Closed (Security On)	BOOT_CFG_LOCK
0x460[2]	SDP_READ_DISABLE	1	Enable/disable reads using serial download support This fuse is available only on silicon revision 1.4 or later	0 – Reads and writes allowed using serial download (default) 1 – Reads using SDR READ_REGISTER are disabled. Writes using SDP are still possible.	BOOT_CFG_LOCK
0x460[3]	DIR_BT_DIS	1	Direct External Memory Boot Disable	0 - The reserved NXP modes are enabled 1 - The reserved NXP modes are disabled. This fuse must be blown to 1 for normal operation.	BOOT_CFG_LOCK
0x460[4]	BT_FUSE_SEL	1	Determines, whether using fuses for boot configuration, or GPIO /Serial loader.	If BOOT_MODE[1:0] = 0b10 0 - Bits of SBMR are overridden by GPIO pins.	BOOT_CFG_LOCK

Table continues on the next page...

Table 5-8. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Settings	Locked by
				1 - Specific bits of SBMR are controlled by eFUSE settings. If BOOT_MODE[1:0] = 0b00 0 - BOOT configuration eFuses are not yet programmed. Boot flow jumps to serial downloader. 1 - BOOT configuration eFuses have been programmed. Regular boot flow is performed.	
0x460[15:8]	Reserved	8	Reserved		BOOT_CFG_LOCK
0x460[20]	SJC_DISABLE	1	Disable/Enable the Secure JTAG Controller module. This fuse is used to create highest JTAG security level, where JTAG is totally blocked.	0 - Secure JTAG Controller is enabled 1 - Secure JTAG Controller is disabled	BOOT_CFG_LOCK
0x460[21]	WDOG_ENABLE	1	Watchdog Enable	Used to specify whether to enable / not watchdog at boot. 0 - Watch-Dog is disabled 1 - Watch-Dog is enabled	BOOT_CFG_LOCK
0x460[23:22]	JTAG_SMODE[1:0]	2	JTAG Security Mode. Controls the security mode of the JTAG debug interface	00 - JTAG enable mode 01 - Secure JTAG mode 11 - No debug mode	BOOT_CFG_LOCK
0x460[26]	KTE	1	Kill Trace Enable. Enables tracing capability on ETM, and other modules.	0 - Bus tracing is allowed 1 - Bus tracing is allowed in case security state as defined by Secure JTAG allows it (for	BOOT_CFG_LOCK

Table continues on the next page...

Table 5-8. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Settings	Locked by
				example, JTAG_ENABLE or NO_DEBUG)	
0x460[27]	JTAG_HEO	1	JTAG HAB Enable Override. Disallows HAB JTAG enabling. The HAB may normally enable JTAG debugging by means of the HAB_JDE-bit in the OCOTP SCS register. The JTAG_HEO-bit can override this behavior.	0 - HAB may enable JTAG debug access 1 - HAB JTAG enable is overridden (HAB may not enable JTAG debug access)	BOOT_CFG_LOCK
0x460[28]	TZASC_ENABLE	1	TZASC1,2 enable fuse.	0 - TZASC1,2 modules left in disable and bypass state. 1 - TZASC1,2 modules and associated clocks and muxing are enabled by Boot ROM code.	BOOT_CFG_LOCK
0x460[29]	SDMMC_HYS_EN	1	Override HYS bit for SD/MMC pads	0 - Override HYS bit for SD/MMC pads disabled. 1 - Once the fuse blown, the [HYS] bit of IOMUXC_SW_PAD_CTL_PAD_SDx_CLK, IOMUXC_SW_PAD_CTL_PAD_SDx_CMD, IOMUXC_SW_PAD_CTL_PAD_SDx_DAT0-n will be set.	BOOT_CFG_LOCK
0x460[30]	eMMC_RESET_EN	1	eMMC 4.4 - RESET TO PRE-IDLE STATE	0 -the CMD0 with argument 0xF0F0F0F0 will not be sent to put the eMMC card into pre-IDLE state. 1 - the CMD0 with argument 0xF0F0F0F0 will be sent to put the eMMC card into	BOOT_CFG_LOCK

Table continues on the next page...

Table 5-8. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Settings	Locked by
				pre-IDLE state so that eMMC card's fast boot can work properly.	
0x460[31]	FORCE_INTERNAL_BOOT	1	Enable/disable selection of boot mode from the BOOT_MODE[1:0] pins. This fuse is available only on silicon revision 1.4 or later	0 – Boot mode selected from the BOOT_MODE[1:0] pins (default). 1 – BOOT_MODE[1:0] pins are ignored. The boot mode is forced to "Internal Boot". This prevents impact from physical tampering with the BOOT_MODE pins. Recommended for security-enabled applications.	BOOT_CFG_LOCK
0x470[7:0]	NAND_READ_CMD_CODE1[7:0]	8	NAND_READ_CMD_CODE1	First command word to be used for Nand read.	BOOT_CFG_LOCK
0x470[15:8]	NAND_READ_CMD_CODE2[7:0]	8	NAND_READ_CMD_CODE2	Second command word to be used for Nand read.	BOOT_CFG_LOCK
0x470[20]	BT_LPB_POLARITY	1	Define GPIO3 polarity, for determining LPB boot mode.	0 - Active High 1 -Active Low	BOOT_CFG_LOCK
0x470[22:21]	LPB_BOOT	2	Defined the LowPower Boot options	(Core / DDR- Bus) 00 - LPB Disable 01 - 1 GPIO (def freq) 01 - Div by2 11 - Div by 4	BOOT_CFG_LOCK
0x470[30:24]	MMC_DLL_DLY[6:0]	7	eMMC 4.4/4.41 delay line default value (set by boot ROM), used in conjunction with "DLL Override" = 1 (BOOT_CFG3[3])	See RM for details.	BOOT_CFG_LOCK
0x480[7:6]	Temperature Grade	2	Indicates the device temperature grade	00 - Commercial (0 to 95C) 01 – Extended Commercial (-20 to 105C)	MEM_TRIM_LOCK

Table continues on the next page...

Table 5-8. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Settings	Locked by
				10 - Industrial (-40 to 105C) 11 - Automotive (-40 to 125C)	
0x4D0[31]	Reserved	1	Reserved	Reserved	ANALOG_LOCK
0x4F0[15:0]	USB_VID[31:0]	16	USB VID value,(to be used by USB software driver).		ANALOG_LOCK
0x4F0[31:16]	USB_PID[31:0]	16	USB PID value (to be used by USB software driver).		ANALOG_LOCK
0x580-0x5F0	SRK_HASH[255:0]	256	SRK key		SRK_LOCK
0x600[31:0]	SJC_RESP[31:0]	32	Response reference value for the secure JTAG controller	Customer / OEM use.	SJC_RESP_LOCK (locks also for read and explicit sense)
0x610[23:0]	SJC_RESP[55:32]	24	Response reference value for the secure JTAG controller	Customer / OEM use.	SJC_RESP_LOCK (locks also for read and explicit sense)
0x620[31:0]	MAC_ADDR[31:0]	32	Reserved for customers/software	Customer / OEM use.	MAC_ADDR_LOCK
0x630[15:0]	MAC_ADDR[47:32]	16	Reserved for customers/software	Customer / OEM use.	MAC_ADDR_LOCK
0x660[31:0]	GP1[31:0]	32	General Purpose fuse register #1		GP1_LOCK
0x670[31:0]	GP2[31:0]	32	General Purpose fuse register #2		GP2_LOCK
0x6D0[5:0]	PAD_SETTINGS	6	Used with conjunction of MMC/SD/Nand "Override Pad Settings" fuse value, as follows: 0 - Use IO default settings for boot device IO pads 1 - Use "Override" value, as set by this register	IO pads settings of selected boot interface, are override with this fuses, as follows: [0] - Slew Rate [3:1] Drive Strength [5:4] - Speed Settings. Refer to IO PAD chapter for "Settings" fields value	MISC_CONF_LOCK
0x6E0[0]	FIELD_RETURN	1	Configure device for field return testing. Fuse burning requires CSF command.	0 - Device is in functional / secure mode. 1 - Device is open for 'field-return' testing	FIELD_RETURN_LOCK

Chapter 6
External Memory Controllers

6.1 Overview

This chip has these external memory interfaces and controllers:

- Multi-Mode DDR Controller (MMDC)
- Raw NAND flash controller, consisting of GPMI2, BCH40 and APBH_DMA components including the Randomizer module.
- EIM-PSRAM/NOR flash controller

6.2 Multi-mode DDR controller (MMDC) overview and feature summary

The MMDC module is a DDR controller that supports several types of DDR memories and two channel memory widths (x32 and x64).

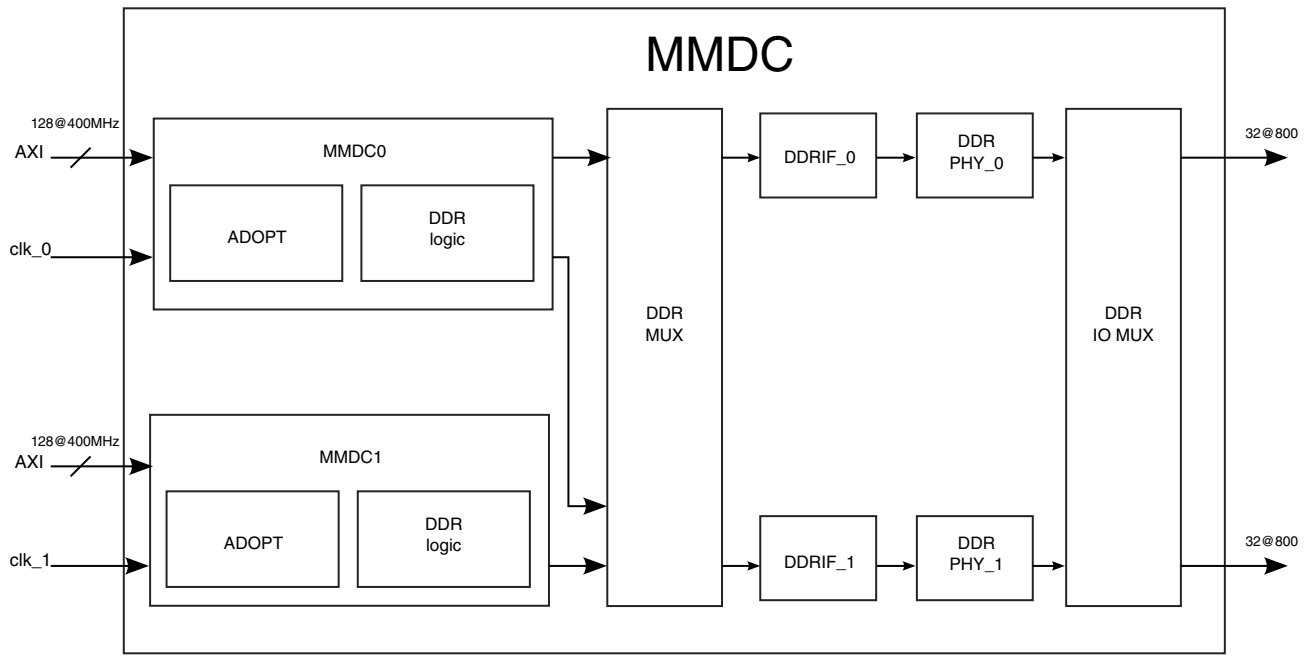


Figure 6-1. MMDC block diagram

Table 6-1. MMDC feature summary

Feature	Description
Supported standards	• LV-DDR3, DDR3 x16, x32, x64 (includes SODIMM) • LPDDR2 2ch x32, in either the split map or the interleaving mode • LPDDR2 1ch x32

DDR interface	• x16, x32, x64 data bus width
	• Density of 256 MB-4 GB
	  • Column size of 8-12 bits
	  • Row size of 11-16 bits
	• 2 CS per channel, with a separate CS allocation for the LPDDR2-DRAM)
	• Up to 4 GB address space and configurable address space per CS. For LPDDR2 2ch x32 up to 2 GB per channel
	• Interleaved accesses of LPDDR2-DRAM towards the same DDR channel. This is supported only when using the same clock frequency for LPDDR2-DRAM
	• Supports burst length of 8 (aligned) for DDR3 and burst lengths of 4 for LPDDR2

DDR performance	• DDR3 and LPDDR2 running at 400 MHz (800 MT/s) • Supports Real-Time priority by means of QoS the sideband priority signals from the chip to enable different priority levels in the re-ordering mechanism • Page hit/page miss optimizations • Consecutive read/write access optimizations • Supports deep read and write requests queues to enable bank prediction • Drives back the critical word in a read transaction as soon as it is received by the DDR device (doesn't wait until the whole data phase has been completed)

Table continues on the next page...

Table 6-1. MMDC feature summary (continued)

Feature	Description
	• Can track open memory pages • Supports bank interleaving • Special optimization for non-aligned wrap accesses in burst length 8
AXI interface	• AXI bus compliant with glueless interface to PL301 AXI network interconnect • Supports bus transfers of 8,16, 32, 64, and 128 bits (single accesses and bursts)

DDR calibration and delay-lines	• All calibrations can be done automatically by hardware or manually by software.
	• ZQ calibration for an external DDR device (in DDR3 through the ZQ calibration command and in LPDDR2 through the MRW command).
	  • Can be handled automatically for ZQ Short (periodically) and ZQ Long (at exit from self-refresh).
	  • Can be handled manually at ZQ INIT.

DDR general	• Configurable timing parameters • Configurable refresh scheme • Supports dynamic voltage, frequency change, and low-power mode entry through hardware negotiation with the system (req/ack handshake) • Suppors automatic self-refresh and power down entry and exit • Supports the fast and slow precharge power down in DDR3 • Supports various ODT control schemes. • Assertion/Deassertion of ODT control per read or write accesses and for active or passive CS • Supports MRW and MRR commands for LPDDR2 • Software control for moving to derated timing parameters and derated refresh rate according to the temperature variation. • Supports various debug and profiling modes

6.3 Raw NAND flash controller overview

The Raw NAND flash controller consists of three components: BCH, GPMI, and APBH_DMA.

- BCH is a 40-bit error correction hardware engine with an AXI bus master and a private connection to the GPMI.
- GPMI is the NAND controller pin interface.
- APBH_DMA is the DMA engine that drives the GPMI module.

This figure shows the modules' connectivity:

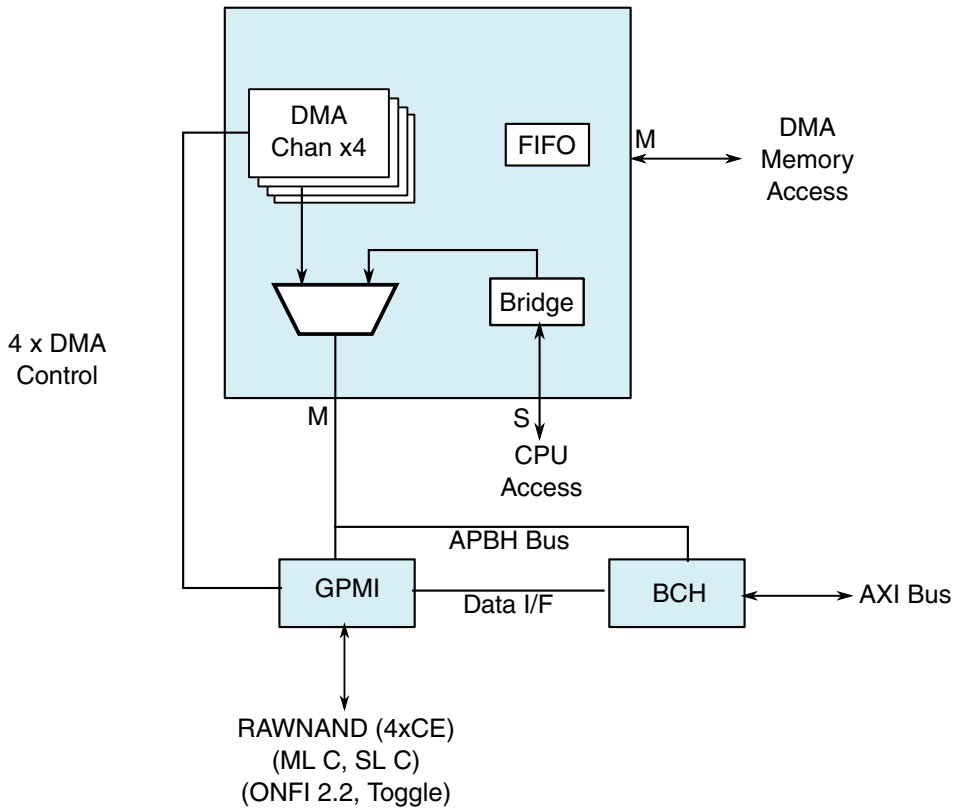


Figure 6-2. Raw NAND flash controller sub-system

6.3.1 NAND interface features

- ONFI2.2-compliant
  - Timing modes 0-5 for both the asynchronous and synchronous interfaces
  - Synchronous clock rate of up to 100 MHz with data rate of up to 200 MB/s
- Support for ganged ready/busy inputs
  - Support allows the use of a single package pin for all ready busy/busy input signals
  - In the ganged mode (selectable by the programmable mode bit), the GPMI expects that all NAND r/b pins are hardwired to the GPMI RDY_BUSY0
- Up to four NAND devices, supported by four chip-selects and one ganged ready/busy
- Legacy raw SLC, MLC type device
- BA-NAND (Micron)
- PBA-NAND and LBA-NAND (Toshiba)
- E2-NAND (Hynix)

- EF-NAND (Samsung)
- Samsung's "Toggle-mode" NAND (clock rate of up to 66 Mhz and 80 Mhz, with a data rate of up to 133 MB/s and 160 MB/s respectively)
- Configurable page size of 2 KB, 4 KB, or 8 KB
- Configurable spare area per page of up to 512 B
- Support for non-identical NAND devices is not required

6.3.2 NAND control features

The NAND control supports interleaved accesses to as many as four NAND devices while hiding the busy period of each device. However, a fully-independent interleaving can't be supported in the ganged ready/busy mode because that suggests a fully independent DMA channel operation on multiple paralleled-accessed chip-enables.

It also features:

- Delay line for adjusting the latching edge in case the frequency is relatively high while the delay is long
- Support for monitoring the external ready/busy signal and read status command polling to compare the R/B bit.

6.3.3 Internal interface features

- The AXI Master interface can access the system memory space.
- The APBH slave port for register configuration through the APBH bridge DMA

6.3.4 APBH-DMA overview

APBH-DMA provides the APB peripheral bus system for GPMI and BCH. APBH-DMA uses a single GPMI interface to support as many as four DMA channels for four NAND devices (four chip-enables) in the interleaved mode.

6.3.5 ECC-BCH features

BCH supports:

- Data rate of up to 200 MB/s
- 2 KB codeword (1 KB of data + parity)

    - Galois Field polynomial GF(2^14)
    - Up to 40-bit ECC for 1 KB of data (2,4,6,8,10,12,14,16, 18,20,22,24,26,28,30,32,34,36,38,40 configurable)
- 1 KB codeword (1/2 KB of data + parity)
    - Galois Field polynomial GF(2^13)
    - Up to 40-bit ECC for 1/2 KB of data (2,4,6,8,10,12,14,16, 18,20,22,24,26,28,30,32,34,36,38,40 configurable)

6.4 EIM-PSRAM/NOR flash controller overview

EIM is an external interface module that manages the interface to external chip devices, including the generation of chip selects, clocks, and control for external peripherals and memory.

It provides asynchronous and synchronous access to devices with an SRAM-like interface.

6.4.1 EIM features

- Up to four (software configurable) chip selects for external devices
    - Flexible address decoding; each chip-select memory space is determined separately according to the GPR bits in IOMUXC.
    - Maximum supported density is 128 MB by default (AUS bit is cleared). When the AUS bit is set, the maximum supported density is 32 MB.
- Selectable write protection for each chip select
- Support for multiplexed address/data bus operation (x16 and x32 port size)
- Programmable data port size for each chip select (x8, x16, and x32)
- Programmable wait-state generator for each chip select, for write and read accesses separately
- Asynchronous accesses with programmable setup and hold times for control signals
- Supports asynchronous page mode accesses (x16 and x32 port size)
- Supports a continuous burst clock that can be used as a reference clock for FPGA
- Independent synchronous memory burst read mode support for NOR-flash and PSRAM memories (x16 and x32 port size)
- Independent synchronous memory burst write mode support for PSRAM and NOR-flash memories (CellularRAM™ from Micron, Infineon, and Cypress, OneNAND™ and utRAM™ from Samsung, and COSMORAM™from Toshiba)
- Supports NAND-flash devices with a NOR-flash interfaces - OneNAND™ (Samsung)

- Independent programmable variable/fix latency support for read and write synchronous (burst) mode
- Support for little-endian operation
- ARM AXI slave interface accesses are only handled in parallel for single AXI ID transactions
- External interrupt support using the RDY_INT signal function as an external interrupt pin
- Boot from external device support according to boot signals, using the RDY_INT signal
  - RDY signal support assertion after reset
  - INT signal support assertion after reset for OneNAND™ (Samsung) device
  - Supports little-endian mode only

6.4.2 EIM boot scenarios

EIM allows booting from NOR-flash devices. To select the NOR flash as the boot source, use either the boot mode and configuration GPIO pins or the internal boot-related fuses.

See Sytem Boot for more information.

6.4.3 EIM boot configuration

This table shows the EIM boot configuration:

Table 6-2. EIM boot configuration

EIM_BOOT_CFG bus	EIM affected bits	EIM register
12	NUM16_BYP_GRANT	CS0GCR2
11	DSZ[2]	CS0GCR1
10	AUS	CS0GCR1
[9:8]	CSREC[2:1]	CS0GCR1
[7:5]	RWSC[4:2]	CS0RCR1
	WWSC[4:2]	CS0WCR
4	ERRST	WCR
3	RAL	CS0RCR1
	WAL	CS0WCR
2	MUM	CS0GCR1
	OEA[1]	CS0RCR1
[1:0]	DSZ[1:0]	CS0GCR1

6.4.4 OneNAND requirements

By default, the Ready/Busy pin is not in use, perform these steps to configure the OneNAND:

- Poll the device to see whether it is ready; software performs a read from the device.
- Connect the Ready/Busy signal of the OneNAND device to any GPIO pin and use it as an interrupt that indicates the OneNAND device ready state.

Chapter 7
System Debug

7.1 Overview

This section describes the hardware and software debug and application development features and resources of the chip. It describes the following:

- Core/platform-specific resources
- Resources associated with complex IP blocks
- Chip-wide resources
- Interface to the external debug and development tools

The debug and trace architecture is designed around the following:

- Arm CoreSight architecture, adapted to i.MX 6Solo/6DualLite SoC (for 1x and 2x core debug), including a cross-trigger subsystem for cross-domain triggering of debug resources
- JTAG port used to interact with cores under the debug by means of SJC, the system JTAG controller port
- DAP, the debug access port that supports the interface to the Arm RealView Debugging tools and other third-party tools
- TPIU, a trace port interface unit that efficiently accesses the program trace information from the system
- Various chip-wide resources, such as debug features built into the IP blocks and critical signal visibility available through alternate pin functions or observability muxes

7.2 Chip and Arm Platform Debug Architecture

The Arm Debug architecture is based on the CoreSight architecture by Arm. The CoreSight architecture provides a system-wide solution to real-time debug and trace.

The CoreSight architecture is embodied in a set of CoreSight components and compliant processors that form the CoreSight systems. Its architecture maintains the traditional requirements of debug and trace:

- To access the debug functionality without software interaction
- To connect to a running system without performing a reset

Full access to the processor debug capability is available by the Arm debug register map through the Advanced Peripheral Bus (APB) slave port. The core includes a Processor Debug Unit which stops program execution, examines and alters the processor and coprocessor state, examines and alters the memory and input/output peripheral state, and restarts the processor core.

7.2.1 Debug Features

- CoreSight Program Trace Macrocell (PTM): trace generator for the Arm Cortex-A9 core
- Support for a TrustZone-related 3-level debug scheme:
  - Debug in Non-Secure privileged and user, and Secure user
  - Debug in Non-Secure only
- EmbeddedICE-RT logic
  - Support for both the monitor-mode and halt-mode debugging:
  - Core run/halt control, debug status/control
  - Breakpoint/watchpoint control
  - Core-mapped and memory-mapped resource examination/modification
- Data communication channel between the Arm core and the host debugger via JTAG and the Debug Access Port (DAP) module
- PMU (Performance Metrics Unit) used for system profiling and debug
- CP15 register for debugging the MMU, I and D L1 cache, and TLB
- PL310 L2 cache provides an event monitoring signal routed to pins for visibility to help in system debug

The chip includes Arm CoreSight components for multicore debug and trace solutions.

See the following figure for the Arm debug architecture scheme.

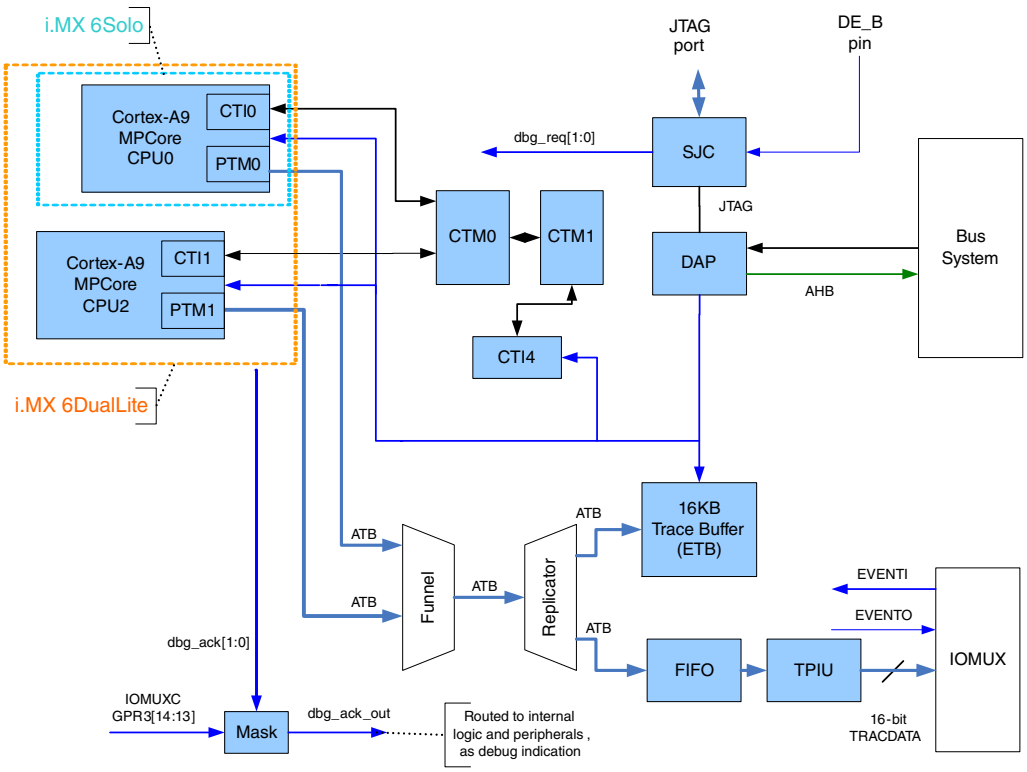


Figure 7-1. i.MX 6Solo/6DualLite Cortex-A9 platform debug architecture

7.2.2 Debug system components

The CoreSight components include:

- Embedded Trace Buffer (ETB): RAM array to be used for on-chip capture of trace data output from the PTM
- ATB Replicator to connect the trace data to TPIU (Trace Port Interface) and ETB (Embedded Trace Buffer)
- Cross Triggering logic for event routing, including CTIs and CTMs

Other related IPs and functionality:

- ROMPATCH module to support modification of program/data information in the MCU ROM
- Debug Visibility which selects critical signals routed to the I/O pads as alternate outputs for external visibility

7.2.2.1 AMBA Trace Bus (ATB)

ATB transfers trace data though the chip CoreSight infrastructure. The trace sources are ATB masters and the sinks are ATB slaves. The Arm (via PTM) cores are the data generators. Link components such as the Trace Funnel and Replicator provide both the master and slave interfaces.

The ATB protocol supports:

- Stalling of trace sources to enable the CoreSight components to funnel and combine the sources into a single trace stream
- Association of the trace data with the generating source using trace source IDs. The CoreSight system can trace up to 111 different items at any time
- Capture and transfer of multiple byte bus widths, currently to 32 bits
- A flushing mechanism to force the historic trace to drain from any sources, links, or sinks up to the point that the request is initiated

7.2.2.2 ATB replicator

The ATB replicator enables two trace sinks to be wired together and to operate from the same incoming trace stream.

There are no programmable registers. This component is invisible to the user on a particular trace path, from source to sink.

- Incoming ATB Interface—the ATB replicator accepts the trace data from the trace source, either directly or through a trace funnel.
- Outgoing ATB Interfaces—the ATB replicator sends identical trace data on the outgoing master port interfaces.

7.2.2.3 Embedded Cross Triggering

The ECT is a modular component from Arm Limited that supports the interaction and synchronization of multiple triggering events within a chip. The main function of the ECT (CTI and CTM) is to pass debug events from one core to another. For example, the ECT can communicate the debug state information from one core to another, so that the program execution can be stopped on both processors at the same time (if required).

The ECT consists of these types of modules:

- Cross Trigger Interfaces (CTI)
- Cross Trigger Matrix (CTM)

The cross trigger interfaces provide the interface between the component or subsystem and the cross trigger matrix. The system requires a CTI for each subsystem that supports cross triggering. The CTI combines and maps the trigger requests, and broadcasts them to all other interfaces on the ECT as channel events. When the CTI receives a channel event, it maps it onto a trigger output. This enables the subsystems to cross trigger with each other. The reception and transmission of triggers is performed through the trigger interface.

The Cross Trigger Matrix (CTM) combines the trigger requests generated from the CTIs and broadcasts them to all CTIs as channel triggers. The CTM controls the distribution of channel events. It provides the Channel Interfaces (CIs) to connect to either the CTIs or CTMs. This enables multiple CTIs to be linked together. The ECT is composed of three/ five CTIs (Cross Trigger Interface) and two CTMs (Cross Trigger Matrix). The ECT is key in the multi-core and multi-IP debug strategy. The outcome is a software-controlled debug signal matrix that receives signals from various sources (cores and peripherals) and propagates/ routes them to the different debug resources of the SoC. These debug resources include the time-stamping capability, profiling capabilities, real-time trace (trace enabled or disabled), triggers, SOC level multiplexing, and debug interrupts.

NOTE

Because the ECT should only be used during the debug sessions, it is disabled by default.

The chip's cross-triggering architecture consists of 2xCTMs and 2 / 3[1] CTIs cover these three basic groups of functionality:

- CTI 0,1 - for Arm core 0/1 respectively

1. Depending on specific part: i.MX 6Solo, i.MX 6DualLite.

7.2.2.3.1 Cross-Trigger Matrix (CTM)

The CTM (Cross-Trigger Matrix) is provided by Arm. A brief description is provided below. See the Arm documentation for more details.

The CTM is a relatively simple block with no configuration options. There are two CTM instances in i.MX6Solo/6DualLite in the Arm platform.

One of them is used to route 2x Core's CTI's, while the second one is used for the additional CoreSight CTIs. Each CTI has four channel lines, to which the CTI events are mapped. The exact mapping is configured in the CTI logic.

7.2.2.3.2 Cross-Trigger Interface (CTI)

The Cross-Trigger Interface (CTI) component is provided by Arm. A brief description of the CTI is provided below.

There are 3 CTIs in the chip's Arm platform. Two are of which, dedicated to each Core, while the 3rd is used for various other signals routing.

Each of these CTIs has 8 trigger inputs and 8 trigger outputs that connect to logic in the domain to be debugged or profiled. Each CTI also includes a 4 channel interface to the CTM (4 inputs and 4 outputs).

For more information, see Arm platform chapter.

7.2.2.4 Debug Access Port (DAP)

The DAP enables debug access to the chip modules through APB-AP (the APB access port) and APB-Mux (the APB multiplexer).

AHB-AP provides system access. Debug tools can use JTAG to connect to the chip.

DAP has the following features:

- AMBA 3 Peripheral Bus Multiplexor access though AMBA 3 APB Access Port, providing debug peripheral access through the APB interface.
- External JTAG access using the JTAG Debug Port (JTAG-DP).
- Internal chip module access using:
  - AHB Access Port (AHB-AP)
  - APB Access Port (APB-AP)
  - JTAG Access Port (JTAG-AP)

APB-Mux enables system access to CoreSight components connected to the Debug APB.

The ROM table provides a list of memory locations of CoreSight components connected to the Debug APB. This is visible from both tools and system access and one configures it during system implementation.

External read/write access to the internal interface is provided by JTAG-DP. JTAG-DP provides a standard interface for debug access to the chip through DAP. It interfaces to the DAP internal bus.

Internal access to on-chip buses and other interfaces are provided by the access ports (APs). The available APs are:

- AHB-AP which provides an AHB-Lite master for access to a system AHB bus.
- APB-AP which provides an AMBA 3 APB master for access to the Debug APB that configures all CoreSight components.
- JTAG-AP which provides JTAG access to on-chip components and operates as a JTAG master port to drive JTAG chains throughout the chip.

7.2.2.5 Coresight Embedded Trace Buffer (ETB)

The ETB provides on-chip storage of trace data using 32-bit RAM. The ETB accepts trace data from any CoreSight-compliant component trace source with an ATB master port, such as a trace source or a trace funnel. It is included in this device to remove dependencies from the trace pin pad speed, and enable low cost trace solutions. The TraceRAM size is 2 KB.

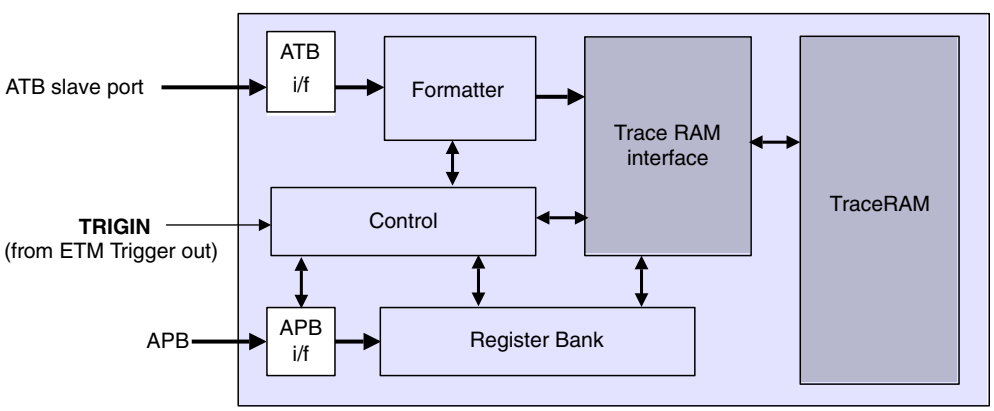


Figure 7-2. ETB Block Diagram

The ETB contains the following blocks:

- Formatter—Inserts source ID signals into the data packet stream so that trace data can be re-associated with its trace source after the data is read back out of the ETB.
- Control—Control registers for trace capture and flushing.

- APB interface—Read, write, and data pointers provide access to ETB registers. In addition, the APB interface supports wait states through the use of a PREADYDBG signal output by the ETB. The APB interface is synchronous to the ATB domain.
- Register bank—Contains the management, control, and status registers for triggers, flushing behavior, and external control.
- Trace RAM interface—Controls reads and writes to the Trace RAM.

7.2.2.5.1 Performance Profiling with the ETB

To create a performance profile (for example, gprof) for the target application, a means to collect trace over a long period of time is needed. The ETB buffer is too small to capture a meaningful profile in just one take. What is needed is to collect and concatenate data from the ETB buffer for multiple sequential runs. Using the ETB packet counter, the trace analysis tool can capture multiple sequential runs by executing code until the ETB is almost full, and halting or executing an interrupt handler to allow the buffer to be emptied, and then continuing executing code. The target halts or executes an interrupt handler when the buffer is almost full to empty the data and then the debugger runs the target again.

7.2.2.5.2 ETB Counter Control

The ETB packet counter is controlled by the ETB counter control register, ETB reload register, and ETB counter value register implemented in the Miscellaneous Control Module (MCM) accessible via the Private Peripheral Bus. Via the ETB counter control register the ETB control logic can be configured to cause an MCM Alert Interrupt, an NMI Interrupt, or cause a Debug halt when the down counter reaches 0. Other features of the ETB control logic include:

- Down counter to count as many as 512 x 32-bit packets.
- Reload request transfers reload value to counter.
- ATB valid and ready signals used to form counter decrement.
- The counter disarms itself when the count reaches 0.

The counter decrement enable will allow the counter to decrement at the rising edge of ATCLK when,

- The counter enable bit is set.
- The counter is armed.
- ATVALID and ATREADY are asserted.

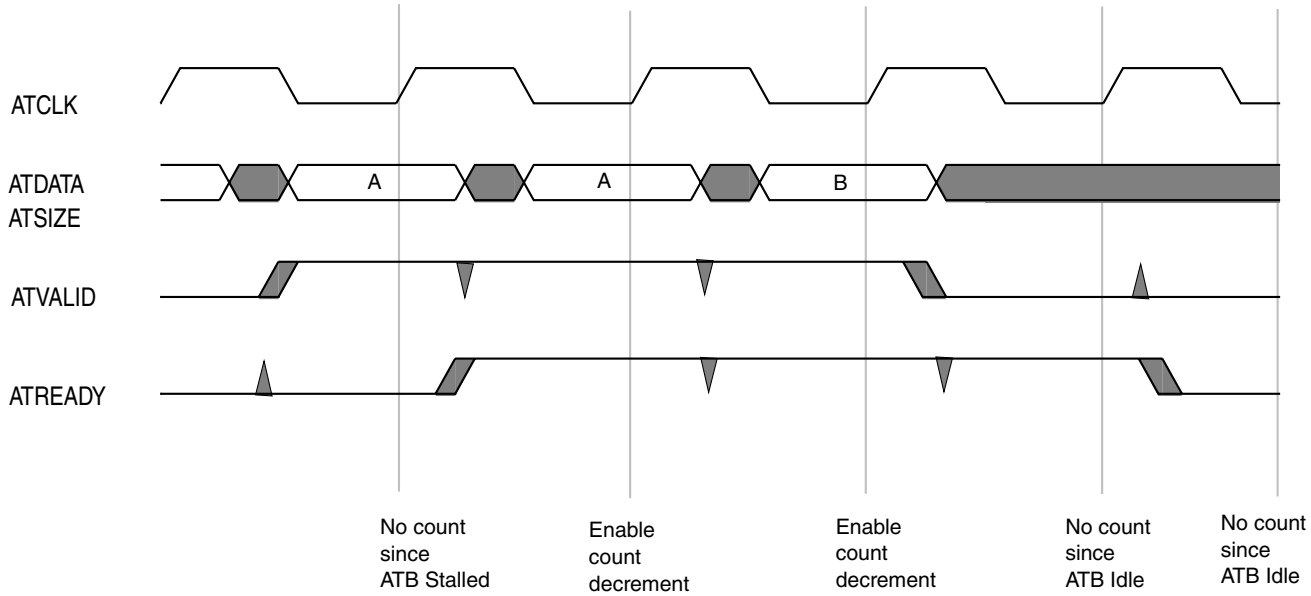


Figure 7-3. Count Decrement Enable

7.2.3 Chip-Specific SJC Features

7.2.3.1 JTAG Disable Mode

In addition to four different JTAG security modes that are implemented internally in the JTAG Controller, there is an option to disable the SJC functionality by e-fuse configuration.

This creates additional JTAG mode "JTAG Disabled" with highest level of JTAG protection. In this mode all JTAG features are disabled. Specifically, the following debug features are disabled in addition to the features that were already disabled in "No Debug" JTAG mode:

- Non-Secure JTAG control registers (PLL configuration, Deterministic Reset, PLL bypass)
- Non-Secure JTAG status registers (Core status)
- Chip Identification Code (IDCODE)

7.2.3.2 JTAG ID

Table 7-1. i.MX JTAG ID

Device	Silicon revision	JTAG ID
i.MX 6Solo	Rev 1.0	0891_B01Dh[1]
i.MX 6DualLite	Rev 1.0	0891_A01Dh[2]

1. In follow-on silicon revisions, the ID value is subject to change by incrementing the first nibble as follows: 1891_B01Dh for Rev 1.1, 2891_B01Dh for Rev 1.2 , etc.
2. In follow-on silicon revisions, the ID value is subject to change by incrementing the first nibble as follows: 1891_A01Dh for Rev 1.1, 2891_A01Dh for Rev 1.2 , etc.

7.2.4 System JTAG Controller - SJC

The SJC module is the bridge between external development and test instrumentation and the internal JTAG-accessible debug and test resources.

It implements and manages the daisy-chained topology consisting of its own TAP and those of the SDMA, and the Arm Debug Access Port (DAP).

The DAP itself has a JTAG serial links, which can drive the PCIe PHY JTAG logic, used for testing of the PHYs.

NOTE

Single Wire Debug (SWD) protocol is not supported.

7.2.5 System JTAG controller main features

- IEEE P1149.1, 1149.6 (standard JTAG) interface to off-chip test and development equipment
  - Includes an SJC-only mode for true IEEE P1149.1 compliance, used primarily for board-level implementation of boundary scan.
  - Supports IEEE P1149.6 extensions to the JTAG standard for AC testing of selected I/O signals.
- Debug-related control and status; puting selected cores into reset and/or debug mode and monitoring individual core status signals by means of JTAG
- System status, such as the state of the PLLs (locked or not locked)
- Four levels of security, ranging from no security to no JTAG accessibility to the chip

7.2.6 TAP Port

The SJC supports the following standard JTAG pins:

- TRSTB
- TDI
- TDO
- TCK
- TMS

7.2.7 SJC main blocks

- Interface to the outside world via the standard JTAG pins
- Interface to the external Debug_Event pin
- A master TAP controller which implements the standard JTAG state machine
- Implementation of the mandatory and optional IEEE P1149.1 (JTAG) instructions
  - Mandatory: "EXTEST", "SAMPLE/PRELOAD", and "BYPASS"
  - Optional: "ID_CODE" (SOC JTAG ID register), "HIGHZ"
- Supports the SDMA's DR-path-only JTAG architecture by implementing the controller portion of its TAP (including "BYPASS" as the default state) within the SJC
- The ExtraDebug registers, which implement a variety of control and status features
  - Three 32-bit insecure general purpose status registers
  - Two 32-bit secure status registers - one predefined, one general purpose.
  - Control and status registers for debug, core, charge pump, and PLL.
- Four levels of fuse-defined security, ranging from no security to no access.

Both predefined and user-defined control and status functions are supported by the SJC.

7.3 Smart DMA (SDMA) core

SDMA is a dedicated, programmable DMA engine. It is an integration of a 32-bit RISC core and DMA-specific hardware. It includes ports for the AP domain and a peripheral domain, along with a burst-capable port for direct external memory access.

NOTE

The SDMA and its integration in the chip is unchanged from previous i.MX chips.

The main SDMA debug features are:

- OnCE - On Chip Emulator, provides the following capabilities:
  - SDMA core control - run/halt/single-step
  - SDMA core register/memory-map access

    - Event detection, watchpoints, and hardware breakpoints
    - Real time buffer and PC trace buffer capability
- Trace buffer
    - Contains information to identify the 32 last changes of flow detected during a program execution
- Context dump
    - Includes information about all the channel dump activity
    - Current contents of SDMA RAM
- ROMPATCH

7.3.1 SDMA On Chip Emulation Module (OnCE) Feature Summary

The SDMA debug features are primarily defined by the OnCE portion of its design.

They are summarized as follows:

- Memory And Register Access - dedicated logic enables user-access to SDMA memory and register locations. These accesses are supported only when the processor is in debug mode.
- Event Detection Unit - watches signals from the data memory bus (DMBus) which is used by the RISC core to access its RAM, ROM, and memory-mapped registers
- Watchpoints - one output signal is available to watch event matching conditions at the chip level. Match conditions are defined by programming memory-mapped registers.
- Hardware Breakpoint - a counter is decremented after an event detection. A debug request is sent to the SDMA core only when the counter reaches the value of zero. It is possible to program the initial value of the counter or to disable the use of the counter if a debug request must be generated after each event detection.
- Real Time Buffer - The Real Time Buffer Register (RTB) is a single 32-bit memory-mapped register which can be accessed as a regular memory location during program execution. It is used to store and retrieve run time information without putting the SDMA in debug mode. Each write to this register causes an event. This register is, in fact, located in the OnCE. Executing through JTAG, a buffer command exports the content of this register through the JTAG port.
- Core Control (Core Status / Single Stepping) - Commands are provided to monitor and control processor activity. The commands can halt the core, rerun the core from another address location, and get processor status.
- Trace Buffer - a 32x32 buffer which records the last 32 changes of flow during program execution. The buffer stores data in a modulo fashion (i.e. the 33rd

instruction change replaces the 1st). Captured trace information is retrieved via reads to the Trace Buffer Register.

7.3.1.1 Other SDMA Debug Functionality

- Core Trace - basic core trace capability is available through debug visibility functionality only. PTM trace capability does not exist.
- ROM Patch - can be accomplished by manipulating the CHN0ADDR register through JTAG or via the MCU's ability to write to SDMA OnCE registers. This must be done right after reset and before the SDMA core is enabled to begin processing events.
- Additional debug control/status interaction with the SJC module
  - SJC-controlled Debug Request
  - SJC-readable Debug Acknowledge (in debug mode)
  - Debug clock control - allows SJC to force clocks on for debug purposes
  - Debug core state (SDMA RISC Core State) - 4 bits accessible from the SJC via JTAG

7.3.1.2 SDMA ROM Patching

After reset, the SDMA is in its IDLE_AFTER_RESET mode. A debug request also puts the SDMA in its DEBUG_IN_IDLE_AFTER_RESET mode. The new address boot must be stored in CHN0ADDR register (e.g., through the SDMA OnCE via debugger).

The user must then issue the exec_core <instruction> SDMA OnCE instruction to return to the IDLE_AFTER_RESET mode. The very first instructions of the boot code fetches the contents of this register (which is also mapped in the SDMA memory space) and jumps to the given address.

7.4 Miscellaneous

The Miscellaneous function described in this section provide useful general capabilities.

7.4.1 Clock/Reset/Power

CDBGPWRUPREQ and CDBGPWRUPACK are the handshake signals between the DAP and the clock control module to ensure debug power and clocks are turned on. If the debug components are always powered on, the handshake becomes a mechanism to turn

debug clocks on. Similarly, there is a register bit in the CCM which allows internal software to turn debug clocks on as well because the CDBGPWRUPREQ is in the TCLK domain and is inaccessible to software.

The Cortex-A9 and VSP cores can receive resets from the following sources:

- Debug Reset (CDBGRSTREQ bit within the SWJ-DP CTRL/STAT register of the DAP) in the TCLK domain. This allows the debug tools to reset the debug logic.
- System POR reset

7.5 Supported tools

DS-5 Arm Debugger is supported.

The debugger is connected to the chip from the host by the DS-5 ICE protocol converter. Other third party tools can be used via the standard JTAG interface, but may need to be adapted for individual IC. It is important to check with tool vendors for specific tool requirements, especially for on-chip IC.

Chapter 8
System Boot

8.1 Overview

The boot process begins at the Power-On Reset (POR) where the hardware reset logic forces the Arm core to begin the execution starting from the on-chip boot ROM.

The boot ROM code uses the state of the internal register BOOT_MODE[1:0] as well as the state of various eFUSEs and/or GPIO settings to determine the boot flow behavior of the device.

The main features of the ROM include:

- Support for booting from various boot devices
- Serial downloader support (USB OTG)
- Device Configuration Data (DCD) and plugin
- Digital signature and encryption based High-Assurance Boot (HAB)
- Wake-up from the low-power modes
- Wake-up secondary core

The boot ROM supports these boot devices:

- NOR flash
- NAND flash
- OneNAND flash
- SD/MMC
- Serial (I2C/SPI) NOR flash and EEPROM

The boot ROM uses the state of the BOOT_MODE and eFUSEs to determine the boot device. For development purposes, the eFUSEs used to determine the boot device may be overridden using the GPIO pin inputs.

The boot ROM code also allows to download the programs to be run on the device. The example is a provisioning program that can make further use of the serial connection to provide a boot device with a new image. Typically, the provisioning program is

downloaded to the internal RAM and allows to program the boot devices, such as the SD/MMC flash. The ROM serial downloader uses a high-speed USB in a non-stream mode connection.

The boot ROM allows waking up from the low-power modes and waking up the secondary core of Cortex-A9. On reset, the ROM checks the Arm core ID and the power gating status register. When waking from the low-power mode, the primary core (ID=0) skips loading an image from the boot device and jumps to the address saved in PERSISTENT_ENTRY0. The secondary cores (ID!=0) jump to the address in PERSISTENT_ENTRY<X> where 0<X<4 in all power modes.

The Device Configuration Data (DCD) feature allows the boot ROM code to obtain the SOC configuration data from an external program image residing on the boot device. As an example, the DCD can be used to program the DDR controller for optimal settings, improving the boot performance. The DCD is restricted to the memory areas and peripheral addresses that are considered essential for the boot purposes (see Write data command).

A key feature of the boot ROM is the ability to perform a secure boot, also known as a High-Assurance Boot (HAB). This is supported by the HAB security library which is a subcomponent of the ROM code. The HAB uses a combination of hardware and software together with the Public Key Infrastructure (PKI) protocol to protect the system from executing unauthorized programs. Before the HAB allows the user image to execute, the image must be signed. The signing process is done during the image build process by the private key holder and the signatures are then included as a part of the final program image. If configured to do so, the ROM verifies the signatures using the public keys included in the program image. In addition to supporting the digital signature verification to authenticate the program images, the encrypted boot is also supported. The encrypted boot can be used to prevent the cloning of the program image directly off the boot device. A secure boot with HAB can be performed on all boot devices supported on the chip in addition to the serial downloader. The HAB library in the boot ROM also provides the API functions, allowing the additional boot chain components (bootloaders) to extend the secure boot chain. The out-of-fab setting for the SEC_CONFIG is the open configuration, in which the ROM/HAB performs the image authentication, but all authentication errors are ignored and the image is still allowed to execute.

NOTE

Silicon Revision 1.6 and later supports additional fuses related to Serial Download mode. Fuse SDP_DISABLE enables/disables Serial Download mode support. Fuse SDP_READ_ENABLE enables/disables reads using Serial Download mode. For detailed information on these fuses, see Fusemap.

8.2 Boot modes

During reset, the chip checks the ARM core ID and power gating controller status register.

During boot, the core's behavior is defined by the boot mode pin settings, as described in Boot mode pin settings. When waking up from the low-power boot mode, the core skips the clock settings. The boot ROM checks that the PERSISTENT_ENTRY0 (see Persistent bits) is a pointer to a valid address space (OCRAM, DDR, or EIM). If the PERSISTENT_ENTRY0 is a pointer to a valid range, it starts the execution using the entry point from the PERSISTENT_ENTRY0 register. If the PERSISTENT_ENTRY0 is a pointer to an invalid range, the core performs the system reset.

For the secondary cores (with ID!=0), the boot ROM checks that the PERSISTENT_ENTRY is a pointer to a valid address space (OCRAM, DDR, or EIM). If the PERSISTENT_ENTRY is a pointer to a valid range, it starts the execution using the entry point from the PERSISTENT_ENTRY register. If the PERSISTENT_ENTRY is a pointer to an invalid range, it sets the error status registers (see Persistent bits), sends the wake-up error interrupt, and performs the Wait For Interrupt instruction. The interrupt service routine of the other core must reconfigure the system and reset the secondary core that failed to boot.

NOTE

The code that enables the secondary cores is not a part of the ROM, but it must be a part of the upper-level software.

8.2.1 Boot mode pin settings

The device has four boot modes (one is reserved for NXP use). The boot mode is selected based on the binary value stored in the internal BOOT_MODE register.

The BOOT_MODE is initialized by sampling the BOOT_MODE0 and BOOT_MODE1 inputs on the rising edge of the POR_B. After these inputs are sampled, their subsequent state does not affect the contents of the BOOT_MODE internal register. The state of the

internal BOOT_MODE register may be read from the BMOD[1:0] field of the SRC Boot Mode Register (SRC_SBMR2). The available boot modes are: Boot From Fuses, serial boot via USB, and Internal Boot. See this table for settings:

Table 8-1. Boot MODE pin settings

BOOT_MODE[1:0]	Boot Type
00	Boot From Fuses
01	Serial Downloader
10	Internal Boot
11	Reserved

NOTE

Silicon Revision 1.6 and later, the FORCE_INTERNAL_BOOT has been implemented. This fuse causes the BOOT_MODE inputs to be ignored and the boot ROM will force the boot mode to “Internal Boot”. This prevents impact from physically tampering with the BOOT_MODE pins and is recommended for security-enabled application. For detailed information on this fuses, see Fusemap.

8.2.2 High-level boot sequence

The figure found here show the high-level boot ROM code flow.

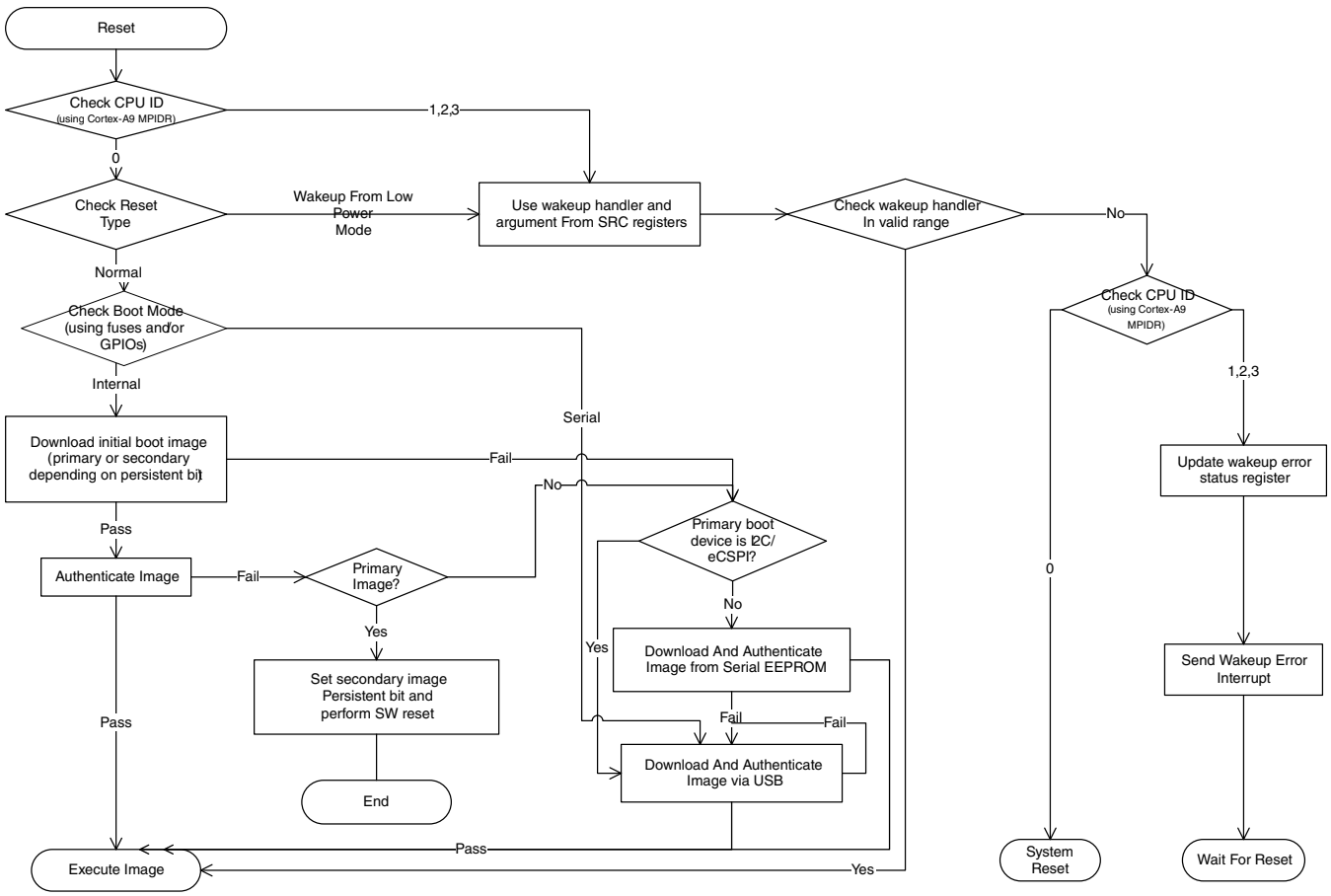


Figure 8-1. Boot flow

8.2.3 Boot From Fuses mode (BOOT_MODE[1:0] = 00b)

A value of 00b in the BOOT_MODE[1:0] register selects the Boot From Fuses mode.

This mode is similar to the Internal Boot mode described in Internal Boot mode (BOOT_MODE[1:0] = 0b10) with one difference. In this mode, the GPIO boot override pins are ignored. The boot ROM code uses the boot eFUSE settings only. This mode also supports a secure boot using HAB.

If set to Boot From Fuses, the boot flow is controlled by the BT_FUSE_SEL eFUSE value. If BT_FUSE_SEL = 0, indicating that the boot device (for example, flash, SD/MMC) was not programmed yet, the boot flow jumps directly to the Serial Downloader. If BT_FUSE_SEL = 1, the normal boot flow is followed, where the ROM attempts to boot from the selected boot device.

The first time a board is used, the default eFUSEs may be configured incorrectly for the hardware on the platform. In such case, the Boot ROM code may try to boot from a device that does not exist. This may cause an electrical/logic violation on some pads. Using the Boot From Fuses mode addresses this problem.

Setting the BT_FUSE_SEL=0 forces the ROM code to jump directly to the Serial Downloader. This allows a bootloader to be downloaded which can then provision the boot device with a program image and blow the BT_FUSE_SEL and the other boot configuration eFUSEs. After the reset, the boot ROM code determines that the BT_FUSE_SEL is blown (BT_FUSE_SEL = 1) and the ROM code performs an internal boot according to the new eFUSE settings. This allows the user to set BOOT_MODE[1:0]=00b on a production device and burn the fuses on the same device (by forcing the entry to the Serial Downloader), without changing the value of the BOOT_MODE[1:0] or the pullups/pulldowns on the BOOT_MODE pins.

8.2.4 Serial Downloader

The Serial Downloader provides a means to download a program image to the chip over the USB serial connection.

In this mode, the ROM programs the WDOG1 for a 90-second time-out if the WDOG_ENABLE eFuse is 1 and continuously polls for the USB connection. If no activity is found on the USB OTG1 and the watchdog timer expires, the Arm core is reset.

NOTE

After the downloaded image is loaded, it is responsible for managing the watchdog resets properly.

This figure shows the USB boot flow:

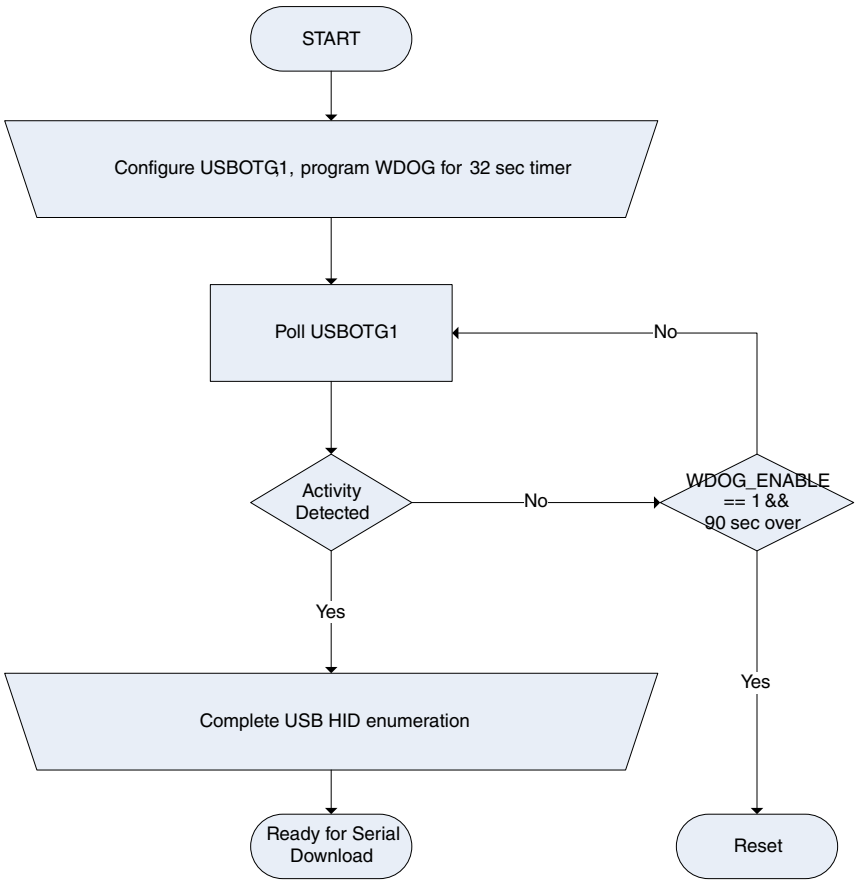


Figure 8-2. Serial Downloader boot flow

8.2.5 Internal Boot mode (BOOT_MODE[1:0] = 0b10)

A value of 0b10 in the BOOT_MODE[1:0] register selects the Internal Boot mode. In this mode, the processor continues to execute the boot code from the internal boot ROM.

The boot code performs the hardware initialization, loads the program image from the chosen boot device, performs the image validation using the HAB library (see Boot security settings), and then jumps to an address derived from the program image. If an error occurs during the internal boot, the boot code jumps to the Serial Downloader (see Serial Downloader). A secure boot using the HAB is possible in all the three boot modes.

When set to the Internal Boot, the boot flow may be controlled by a combination of eFUSE settings with an option of overriding the fuse settings using the General Purpose I/O (GPIO) pins. The GPIO Boot Select FUSE (BT_FUSE_SEL) determines whether the ROM uses the GPIO pins for a selected number of configuration parameters or eFUSEs in this mode.

- If BT_FUSE_SEL = 1, all boot options are controlled by the eFUSEs described in Table 8-2.
- If BT_FUSE_SEL = 0, the specific boot configuration parameters may be set using the GPIO pins rather than eFUSEs. The fuses that can be overridden when in this mode are indicated in the GPIO column of Table 8-2. Table 8-3 provides the details of the GPIO pins.

The use of the GPIO overrides is intended for development since these pads are used for other purposes in the deployed products. NXP recommends controlling the boot configuration by the eFUSEs in the deployed products and reserving the use of the GPIO mode for the development and testing purposes only.

8.2.6 Boot security settings

The internal boot modes use one of three security configurations.

- Closed: This level is intended for use with shipping-secure products. All HAB functions are executed and the security hardware is initialized (the Security Controller or SNVS enters the Secure state), the DCD is processed if present, and the program image is authenticated by the HAB before its execution. All detected errors are logged, and the boot flow is aborted with the control being passed to the serial downloader. At this level, the execution does not leave the internal ROM unless the target executable image is authenticated.
- Open: This level is intended for use in non-secure products or during the development phases of a secure product. All HAB functions are executed as for a closed device. The security hardware is initialized (except for the SNVS which is left in the Non-Secure state), the DCD is processed if present, and the program image is authenticated by the HAB before its execution. All detected errors are logged, but have no influence on the boot flow which continues as if the errors did not occur. This configuration is useful for a secure product development because the program image runs even if the authentication data is missing or incorrect, and the error log can be examined to determine the cause of the authentication failure.
- Field Return: This level is intended for the parts returned from the shipped products.

8.3 Device configuration

This section describes the external inputs that control the behavior of the Boot ROM code.

This includes the boot device selection (SPI, EIM, NOR, SD, MMC, and so on), boot device configuration (SD bus width, speed, and so on), and other. In general, the source for this configuration comes from the eFUSEs embedded inside the chip. However, certain configuration parameters can be sourced from the GPIO pins, allowing further flexibility during the development process.

8.3.1 Boot eFUSE descriptions

This table is a comprehensive list of the configuration parameters that the ROM uses.

Table 8-2. Boot eFUSE descriptions

Fuse	Configuration	Definition	GPIO[1]	Shipped value	Settings[2]
DIR_BT_DIS	OEM	Disables the NXP reserved modes.	NA	0	0—The reserved NXP modes are enabled. 1—The reserved NXP modes are disabled. This fuse must be blown to 1 for normal operation.
BT_FUSE_SEL	OEM	In the Internal Boot mode BOOT_MODE[1:0] = 10, the BT_FUSE_SEL fuse determines whether the boot settings indicated by a Yes in the GPIO column are controlled by the GPIO pins or the eFUSE settings in the On-Chip OTP Controller (OCOTP). In the Boot From Fuse mode BOOT_MODE[1:0] = 00, the BT_FUSE_SEL fuse indicates whether the bit configuration eFuses are programmed.	NA	0	If BOOT_MODE[1:0] = 0b10: • 0—The bits of the SBMR are overridden by the GPIO pins. • 1—The specific bits of the SBMR are controlled by the eFUSE settings. If BOOT_MODE[1:0] = 0b00 • 0—The BOOT configuration eFuses are not programmed yet. The boot flow jumps to the serial downloader. • 1—The BOOT configuration eFuses are programmed. The regular boot flow is performed.
SEC_CONFIG[1:0]	SEC_CONFIG[0] - NXP SEC_CONFIG[1] - OEM	Security Configuration, as defined in Boot security settings	NA	01	00—Reserved 01—Open (allows any program image, even if the authentication fails) 1x—Closed (The program image executes only if authenticated)
FIELD_RETURN	OEM	Enables the NXP reserved modes.			0—The NXP reserved modes are enabled/ disabled based on the DIR_BT_DIS value. 1—The NXP reserved modes are enabled.
SRK_HASH[255:0]	OEM	256-bit hash value of the super root key (SRK_HASH)	NA	0	Settings vary—used by HAB

Table continues on the next page...

Table 8-2. Boot eFUSE descriptions (continued)

Fuse	Configuration	Definition	GPIO[1]	Shipped value	Settings[2]
UNIQUE_ID[63:0]	NXP	Device Unique ID, 64-bit UID	NA	Unique ID	Settings vary—used by HAB
BT_MMU_DISABLE (BOOT_CFG3[6])	OEM	The MMU/L1 D Cache/ PL310 disable bit used by the boot ROM for fast HAB processing.	Yes	0	0—MMU/L1 D Cache/PL310 is enabled by the ROM during the boot. 1—MMU/L1 D Cache/PL310 is disabled by the ROM during the boot.
L1 I-Cache DISABLE (BOOT_CFG3[7])	OEM	L1 I Cache disable bit used by the boot during the entire execution.	Yes	0	0—L1 I Cache is enabled by the ROM during the boot. 1—L1 I Cache is disabled by the ROM during the boot.
BT_FREQ (BOOT_CFG3[2])	OEM	Boot frequency selection	Yes	0	0—Arm—792 MHz, DDR—396 MHz, AXI—264 MHz 1—Arm—396 MHz, DDR—352 MHz, AXI—176 MHz
BOOT_CFG1[7:0]	OEM	Boot configuration 1	Yes	0	Specific to the selected boot mode
BOOT_CFG2[7:0]	OEM	Boot configuration 2	Yes	0	Specific to the selected boot mode
BOOT_CFG4[6:0]	OEM	Boot configuration 4		0	Specific to the selected boot mode
BOOT_CFG4[7]	OEM	Infinite Loop Enable at the start of the boot ROM. Used for debugging purposes. Ignored if the DIR_BT_DIS is 1 and FIELD_RETURN is 0.	Yes	0	0—Disabled 1—Enabled
LPB_BOOT	OEM	USB Low-Power Boot	No	0	00—LPB Disable 01—1 GPIO (default frequencies) 10—Divide by 2 11—Divide by 4
BT_LPB_POLARITY	OEM	USB Low-Power Boot GPIO polarity	No	0	0—Low on the GPIO pad indicates the low-power condition. 1—High on the GPIO pad indicates the low-power condition.
WDOG_ENABLE	OEM	Watchdog reset counter enable	No	0	0—The watchdog reset counter is disabled during the serial downloader. 1—The watchdog reset counter is enabled during the serial downloader.
MMC_DLL_DLY[6:0]	OEM	uSDHC Delay Line settings	No	0	uSDHC Delay Line settings
SRK_REVOKE[2:0]	OEM	SRK revocation mask	No	0	SRK revocation mask
DISABLE_SDMMC_MFG	OEM	Disable the SDMMC manufacture mode	Yes	0	0—enable the SD/MMC MFG mode 1—disable the SD/MMC MFG mode
PAD_SETTINGS	OEM	Override values for the SD/MMC and NAND boot modes	No	0	Override these IO PAD settings: • PAD_SETTINGS[0]—Slew Rate

Table 8-2. Boot eFUSE descriptions

Fuse	Configuration	Definition	GPIO[1]	Shipped value	Settings[2]
					• PAD_SETTINGS[3:1]—Drive Strength • PAD_SETTINGS[5:4]—Speed Settings .

1. This setting can be overridden by the GPIO settings when the BT_FUSE_SEL fuse is intact. See GPIO Boot Overrides for the corresponding GPIO pin.
2. 0 = intact fuse and 1= blown fuse

8.3.2 GPIO boot overrides

This table provides a list of the GPIO boot overrides:

Table 8-3. GPIO override contact assignments

Package pin	Direction on reset	eFuse
BOOT_MODE1	Input	Boot mode selection
BOOT_MODE0	Input	
EIM_DA0	Input	BOOT_CFG1[0]
EIM_DA1	Input	BOOT_CFG1[1]
EIM_DA2	Input	BOOT_CFG1[2]
EIM_DA3	Input	BOOT_CFG1[3]
EIM_DA4	Input	BOOT_CFG1[4]
EIM_DA5	Input	BOOT_CFG1[5]
EIM_DA6	Input	BOOT_CFG1[6]
EIM_DA7	Input	BOOT_CFG1[7]
EIM_DA8	Input	BOOT_CFG2[0]
EIM_DA9	Input	BOOT_CFG2[1]
EIM_DA10	Input	BOOT_CFG2[2]
EIM_DA11	Input	BOOT_CFG2[3]
EIM_DA12	Input	BOOT_CFG2[4]
EIM_DA13	Input	BOOT_CFG2[5]
EIM_DA14	Input	BOOT_CFG2[6]
EIM_DA15	Input	BOOT_CFG2[7]
EIM_A16	Input	BOOT_CFG3[0]
EIM_A17	Input	BOOT_CFG3[1]
EIM_A18	Input	BOOT_CFG3[2]

Table continues on the next page...

Table 8-3. GPIO override contact assignments (continued)

Package pin	Direction on reset	eFuse
EIM_A19	Input	BOOT_CFG3[3]
EIM_A20	Input	BOOT_CFG3[4]
EIM_A21	Input	BOOT_CFG3[5]
EIM_A22	Input	BOOT_CFG3[6]
EIM_A23	Input	BOOT_CFG3[7]
EIM_A24	Input	BOOT_CFG4[0]
EIM_WAIT	Input	BOOT_CFG4[1]
EIM_LBA	Input	BOOT_CFG4[2]
EIM_EB0	Input	BOOT_CFG4[3]
EIM_EB1	Input	BOOT_CFG4[4]
EIM_RW	Input	BOOT_CFG4[5]
EIM_EB2	Input	BOOT_CFG4[6]
EIM_EB3	Input	BOOT_CFG4[7]

The input pins provided are sampled at boot, and can be used to override the corresponding eFUSE values, depending on the setting of the BT_FUSE_SEL fuse.

8.3.3 Device Configuration Data (DCD)

The DCD is the configuration information contained in the program image (external to the ROM) that the ROM interprets to configure various on-chip peripherals. See Device Configuration Data (DCD) for more details on DCD.

8.4 Device initialization

This section describes the details of the ROM and provides the initialization details.

This includes details on:

- The ROM memory map
- The RAM memory map
- On-chip blocks that the ROM must use or change the POR register default values
- Clock initialization
- Enabling the MMU/L2 cache
- Exception handling and interrupt handling

8.4.1 Internal ROM/RAM memory map

These figures show the iROM memory map:

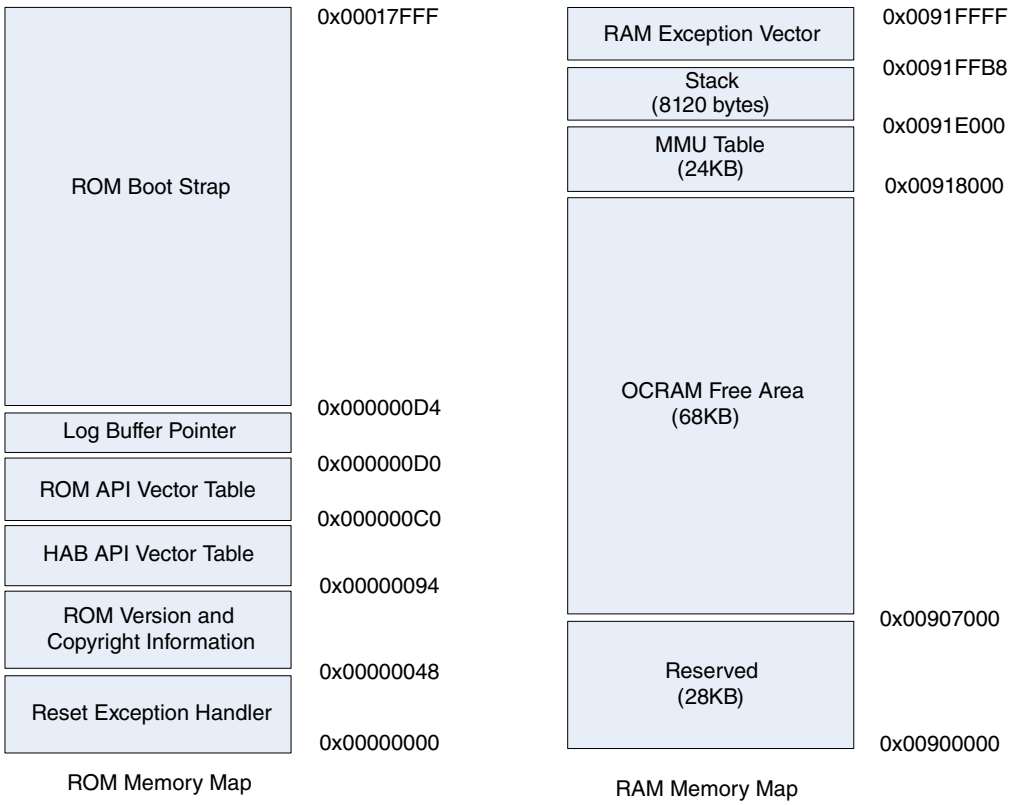


Figure 8-3. Internal ROM and RAM memory map

NOTE

If no ROM/HAB APIs are being used, the entire OCRAM region can be used freely after the boot.

8.4.2 Boot block activation

The boot ROM affects a number of different hardware blocks which are activated and play a vital role in the boot flow.

The ROM configures and uses the following blocks (listed in an alphabetical order) during the boot process. Note that the blocks actually used depend on the boot mode and the boot device selection:

- APBH—the DMA engine to drive the GPMI module

- BCH—40-bit error correction hardware engine with the AXI bus master and a private connection to the GPMI
- CCM—Clock Control Module
- ECSPI—Enhanced Configurable Serial Peripheral Interface
- EIM—External Interface Module used for the NOR and OneNAND devices
- I2C—I^2C controller
- GPMI—NAND controller pin interface
- OCOTP_CTRL—On-Chip OTP Controller; the OCOTP contains the eFUSEs
- IOMUXC—I/O Multiplexer Control which allows the GPIO use to override the eFUSE boot settings;
- IOMUXC GPR—I/O Multiplexer Control General-Purpose Registers
- CAAM—Cryptographic Acceleration and Assurance Module
- SNVS—Secure Non-Volatile Storage
- SRC—System Reset Controller
- USB—used for the serial download of a boot device provisioning program
- USDHC—Ultra-Secure Digital Host Controller
- WDOG-1—Watchdog timer
- DTCP—Digital Transmission Content Protection
- HDCP—High-bandwidth Digital Content Protection

8.4.3 Clocks at boot time

The table below show the various clocks and their sources used by the ROM.

Table 8-4. Normal frequency clocks configuration

Clock	CCM signal	Source	Frequency (MHz) BT_FREQ=0	Frequency (MHz) BT_FREQ=1
System PLL	pll1_sw_clk		792	792
528 MHz PLL	pll2_sw_clk		528	528
480 MHz PLL	pll3_sw_clk		480	480
Arm core clock	arm_clk_root	System PLL	792	396
EIM	aclk_eim_slow_clk_root	AXI	132	132
AHB	ahb_clk_root	528 MHz PLL/PFD307	132	76.6
IPG	ipg_clk_root	528 MHz PLL/PFD307	66	38.3
AXI_A	axi_a	528 MHz PLL/PFD307	396	306
AXI_B	axi_b	528 MHz PLL/PFD352	264	153.3
USB	usboh3_clk_root	480 MHz PLL	60	60
USDHC	usdhc1_clk_root usdhc2_clk_root	PFD400	198	198

Table continues on the next page...

Table 8-4. Normal frequency clocks configuration (continued)

Clock	CCM signal	Source	Frequency (MHz) BT_FREQ=0	Frequency (MHz) BT_FREQ=1
	usdhc3_clk_root usdhc4_clk_root			
ECSPI	ecspi_clk_root	480 MHz PLL	60	60
I^2C	per_clk_root	528 MHz PLL	66	38.3

After the reset, each Arm core has access to all peripherals. The ROM code disables the clocks listed in the following table, except for the boot devices listed in the second column.

Table 8-5. List of disabled clocks

Clock name	Enabled for boot device
CCGR0_APBHDMA	
CCGR1_ECSPI1	ECSPI1
CCGR1_ECSPI2	ECSPI2
CCGR1_ECSPI3	ECSPI3
CCGR1_ECSPI4	ECSPI4
CCGR1_FEC	
CCGR1_EPIT1	
CCGR1_EPIT2	
CCGR2_I2C1_SERIAL	I2C1
CCGR2_I2C2_SERIAL	I2C2
CCGR2_I2C3_SERIAL	I2C3
CCGR3_OPENVGAXICLK	
CCGR4_PWM1	
CCGR4_PWM2	
CCGR4_PWM3	
CCGR4_PWM4	
CCGR5_SDMA	
CCGR5_SPDIF	
CCGR5_SSI1	
CCGR5_SSI2	
CCGR5_SSI3	
CCGR5_UART	
CCGR5_UART_SERIAL	
CCGR6_USBOH3	USB
CCGR6_USDHC1	USDHC1
CCGR6_USDHC2	USDHC2
CCGR6_USDHC3	USDHC3

Table continues on the next page...

Table 8-5. List of disabled clocks (continued)

Clock name	Enabled for boot device
CCGR6_USDHC4	USDHC4
CCGR6_EIM_SLOW	NOR, OneNAND

8.4.4 Enabling MMU and caches

The boot ROM includes a feature that enables the Memory Management Unit (MMU) and the caches to improve the boot speed.

The L1 instruction cache is enabled at the start of the image download. The L1 data cache, L2 cache, and MMU are enabled during the image authentication. When the HAB authentication completes, the ROM disables the L1 data cache, L2 cache, and MMU.

The L1 Instruction cache, L1 data cache, L2 cache, and MMU is controlled by eFuse. By default, these features are enabled.

Enabling the MMU when booting non-securely with SEC_CONFIG=Open and setting the CSF pointer in the Image Vector Table to NULL has no impact on the boot performance. With this configuration, it is recommended to blow the BT_MMU_DISABLE fuse.

8.4.5 Exception handling

The exception vectors located at the start of the ROM are used to map all the Arm exceptions (except the reset exception) to a duplicate exception vector table in the internal RAM.

During the boot phase of CPU0, the RAM vectors point to the serial downloader in the ROM.

During the boot phase of a secondary CPU, the internal RAM vectors point to a function that sets the error status registers (see Persistent bits), sends a wakeup error interrupt, and performs the Wait For Interrupt instruction. The interrupt service routine of the primary CPU must reconfigure the system and reset the secondary CPU.

After the boot, the program image can overwrite the vectors as required. The code shown below is used to map the ROM exception vector table to the duplicate exception vector table in the RAM.

Mapping ROM Exception Vector Table

```
;; Define linker area for ROM exception vector table
AREA IROM_VECTORS, CODE, READONLY
LDR     PC, Reset_Addr
LDR     PC, Undefined_Addr
LDR     PC, SWI_Addr
LDR     PC, Prefetch_Addr
LDR     PC, Abort_Addr
NOP                         ; Reserved vector
LDR     PC, IRQ_Addr
LDR     PC, FIQ_Addr

;; Define exception vector table
Reset_Addr      DCD     start_address
Undefined_Addr  DCD     iRAM_Undefined_Handler
SWI_Addr        DCD     iRAM_SWI_Handler
Prefetch_Addr   DCD     iRAM_Prefetch_Handler
Abort_Addr      DCD     iRAM_Abort_Handler
                DCD     0               ; Reserved vector
IRQ_Addr        DCD     iRAM_IRQ_Handler
FIQ_Addr        DCD     iRAM_FIQ_Handler

start_address DCD start ;reset handler vector
```

8.4.6 Interrupt handling during boot

No special interrupt-handling routines are required during the boot process. The interrupts are disabled during the boot ROM execution and may be enabled in a later boot stage.

8.4.7 Persistent bits

Some modes of the boot ROM require the registers that keep their values after a warm reset. The SRC General-Purpose registers are used for this purpose.

See this table for persistent bits list and description:

Table 8-6. Persistent bits

Bit name	Bit location	Description
PERSIST_SECONDARY_BOOT	SRC_GPR10[30]	This bit identifies which image must be used—primary and secondary. Used only for the boot modes that support redundant boot.
PERSIST_BLOCK_REWRITE	SRC_GPR10[29]	This bit is used as a warning. It identifies that there are errors in the NAND blocks that hold the application image. See NAND flash for more details.
PERSISTENT_ENTRY0[31:0]	SRC_GPR1[31:0]	Holds the entry function for the CPU0 to wake up from the low-power mode.
PERSISTENT_ARG0[31:0]	SRC_GPR2[31:0]	Holds the argument of entry function for the CPU0 to wake up from the low-power mode.

Table continues on the next page...

Table 8-6. Persistent bits (continued)

Bit name	Bit location	Description
PERSIST_OVERRIDE_SBMR1	SRC_GPR10[28]	This bit instructs the ROM code to use the SRC_GPR9 register as if it is the SBMR1. This allows the software to override the fuse bits and boot from an alternative boot source. This feature is vaild for a WDOG time-out reset only, and can be disabled if the DIR_BT_DIS fuse is blown.
PERSIST_SBMR1_VALUE	SRC_GPR9	The value to override the SBMR1.
PERSISTENT_ENTRY1[31:0]	SRC_GPR3[31:0]	Holds the entry function for the CPU1 to wake up from the low-power mode.
PERSISTENT_ARG1[31:0]	SRC_GPR4[31:0]	Holds the argument of the entry function for the CPU1 to wake up from the low-power mode.
CPU1_ERROR_STATUS	SRC_GPR10[25]	CPU1 error status bit

8.5 Boot devices (internal boot)

The chip supports these boot flash devices:

- NOR flash with the External Interface Module (EIM), located on CS0, 16-bit bus width.
- OneNAND flash with the EIM interface, located on CS0, 16-bits bus width.
- Raw NAND (MLC and SLC), and Toggle-mode NAND flash through GPMI-2 interface. Page sizes of 2 KB, 4 KB, and 8 KB. The bus widths of 8-bit with 2 through 40-bit BCH hardware ECC (Error Correction) are supported.
- SD/MMC/eSD/SDXC/eMMC4.4 via USDHC interface, supporting high capacity cards.
- EEPROM boot via SPI (serial flash) and I^2C (via ECSPI and I^2C blocks respectively).

The selection of the external boot device type is controlled by the BOOT_CFG1[7:4] eFUSEs. See this table for more details:

Table 8-7. Boot device selection

BOOT_CFG1[7:4]	Boot device
0000	NOR/OneNAND (EIM)
0001	Reserved
0011	Serial ROM (I^2C/SPI)
010x	SD/eSD/SDXC
011x	MMC/eMMC
1xxx	Raw NAND

8.5.1 NOR flash/OneNAND using EIM interface

The External Interface Module (EIM) works in the asynchronous mode, and supports either muxed, Address/Data, or non-muxed schemes, based on the fuse settings.

Table 8-8. EIM boot eFUSE descriptions

Fuse	Config	Definition	GPIO[1]	Shipped value	Settings
BOOT_CFG1[7:4]	OEM	Boot device selection	Yes	0000	0000—boot from the EIM interface
BOOT_CFG1[3]	OEM	NOR/OneNAND selection	Yes	0	0—NOR 1—OneNAND
BOOT_CFG2[7:6]	OEM	Muxing scheme	Yes	00	00—muxed, 16-bit data (low half) interface 01—not muxed, 16-bit data (high half) interface 10—not muxed, 16-bit data (low half) interface 11—reserved
BOOT_CFG2[5:4]	OEM	OneNAND page size	Yes	00	00—1 KB 01—2 KB 10—4 KB 11—reserved

1. This setting can be overridden by the GPIO settings when the BT_FUSE_SEL fuse is intact. See GPIO Boot Overrides for the corresponding GPIO pin.

8.5.1.1 NOR flash boot operation

Booting from the NOR flash is supported via the EIM interface. The ROM reads the Image Vector Table and Boot Data structures to determine if the image can be executed directly from the EIM address space or copied to another memory.

The start field of the Boot Data Structure specifies the final location of the image (see Image Vector Table and Boot Data).

8.5.1.2 OneNAND flash boot operation

At system power-up, the OneNAND device automatically copies the Initial Load Region of 1 KB from the start of the flash array (sector 0 and sector 1, page 0, block 0) to its Boot RAM (OneNAND's internal RAM).

NOTE

The OneNAND boot RAM memory containing the Initial 1-KB Load Region must contain the IVT, DCD, and the Boot Data structures.

Next, the ROM processes the DCD and then proceeds to copy the program image contents to the application destination pointer (located in the start entry of the Boot Data (see Image Vector Table and Boot Data). The ROM determines the size of the program image by the length specified by the size entry in the Boot Data structure (see Image Vector Table and Boot Data). A failure in loading data from the OneNAND device for any reason forces the chip to enter the Serial Downloader. Otherwise, the booting from the OneNAND device continues.

This figure illustrates the layout of the program image on the OneNAND boot device:

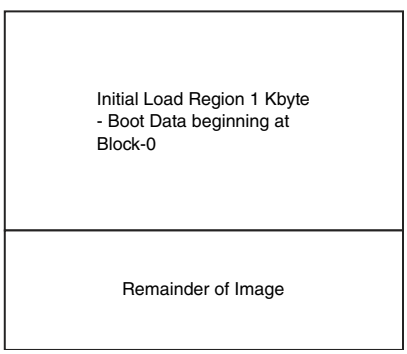


Figure 8-4. Program image layout on the OneNAND flash device

Before accessing the OneNAND device, the chip waits approximately 500 µs after the Power-On Reset. This delay is required for the OneNAND device to become ready. After this initial 500 µs delay, it can take additional 70 µs for the OneNAND device to load the Initial Load Region of 1 KB into its boot RAM. The chip polls the OneNAND device Interrupt Status Register to confirm that the first 1 KB was loaded to the OneNAND boot RAM before continuing with the boot flow.

8.5.1.3 IOMUX configuration for EIM devices

The EIM interface uses the dedicated contacts on the IC.

The contacts assigned to the data signals used by the EIM are shown in this table:

Table 8-9. EIM IOMUX pin configuration

Signal	A/D16 (Muxed, 16-bit data low half interface)	A+DH (Not muxed, 16-bit data high half interface)	A+DL (Not muxed, 16-bit data low half interface)
DATA0	EIM_DA0.alt0	EIM_D16.alt0	CSI0_DATA_EN.alt1
DATA1	EIM_DA1.alt0	EIM_D17.alt0	CSI0_VSYNC.alt1
DATA2	EIM_DA2.alt0	EIM_D18.alt0	CSI0_DAT4.alt1
DATA3	EIM_DA3.alt0	EIM_D19.alt0	CSI0_DAT5.alt1
DATA4	EIM_DA4.alt0	EIM_D20.alt0	CSI0_DAT6.alt1
DATA5	EIM_DA5.alt0	EIM_D21.alt0	CSI0_DAT7.alt1
DATA6	EIM_DA6.alt0	EIM_D22.alt0	CSI0_DAT8.alt1
DATA7	EIM_DA7.alt0	EIM_D23.alt0	CSI0_DAT9.alt1
DATA8	EIM_DA8.alt0	EIM_D24.alt0	CSI0_DAT12.alt1
DATA9	EIM_DA9.alt0	EIM_D25.alt0	CSI0_DAT13.alt1
DATA10	EIM_DA10.alt0	EIM_D26.alt0	CSI0_DAT14.alt1
DATA11	EIM_DA11.alt0	EIM_D27.alt0	CSI0_DAT15.alt1
DATA12	EIM_DA12.alt0	EIM_D28.alt0	CSI0_DAT16.alt1
DATA13	EIM_DA13.alt0	EIM_D29.alt0	CSI0_DAT17.alt1
DATA14	EIM_DA14.alt0	EIM_D30.alt0	CSI0_DAT18.alt1
DATA15	EIM_DA15.alt0	EIM_D31.alt0	CSI0_DAT19.alt1
ADDR0	EIM_DA0.alt0		
ADDR1	EIM_DA1.alt0		
ADDR2	EIM_DA2.alt0		
ADDR3	EIM_DA3.alt0		
ADDR4	EIM_DA4.alt0		
ADDR5	EIM_DA5.alt0		
ADDR6	EIM_DA6.alt0		
ADDR7	EIM_DA7.alt0		
ADDR8	EIM_DA8.alt0		
ADDR9	EIM_DA9.alt0		
ADDR10	EIM_DA10.alt0		
ADDR11	EIM_DA11.alt0		
ADDR12	EIM_DA12.alt0		
ADDR13	EIM_DA13.alt0		
ADDR14	EIM_DA14.alt0		
ADDR15	EIM_DA15.alt0		

8.5.2 NAND flash

The boot ROM supports a number of MLC/SLC NAND flash devices from different vendors and LBA NAND flash devices. The Error Correction and Control (ECC) subblock (BCH) is used to detect the errors.

8.5.2.1 NAND eFUSE configuration

The boot ROM determines the configuration of the external NAND flash by parameters, either provided by the eFUSE, or sampled on the GPIO pins during boot. See Table 8-10 for parameters details.

NOTE

BOOT_CFGx sampled on the GPIO pins depends on the BT_FUSE_SEL setting. See Boot Fusemap for details.

NOTE

For BOOT_CFG[3:2], although ROM always boots from CS0_B, for multiple chip selects, such as some NAND chips which consist of multi CS. This fuse must be burned correctly. The number of devices means the number of chip selects.

Table 8-10. NAND boot eFUSE descriptions

Fuse	Config	Definition	GPIO[1]	Shipped value	Settings
BOOT_CFG1[7]	OEM	Boot device selection	Yes	0	1—boot from the NAND Interface
BOOT_CFG1[5]	OEM	BT_TOGGLEMODE	Yes	0	0—raw NAND 1—toggle mode NAND
BOOT_CFG1[4]	OEM	Override pad settings	Yes	0	0—use the default values 1—use the PAD_SETTINGS values
BOOT_CFG1[3:2]	OEM	Number of devices	Yes	00	00—1 device 01—2 device 10—4 device 11—Reserved
BOOT_CFG1[1:0]	OEM	Row address cycles	Yes	00	00—3 01—2 10—4 11—5
BOOT_CFG2[7:5]	OEM	Toggle mode 33 MHz preamble delay, read latency	Yes	000	000—16 GPMICLK cycles 001—1 GPMICLK cycles

Table continues on the next page...

Table 8-10. NAND boot eFUSE descriptions (continued)

Fuse	Config	Definition	GPIO[1]	Shipped value	Settings
					010—2 GPMICLK cycles 011—3 GPMICLK cycles 100—4 GPMICLK cycles 101—5 GPMICLK cycles 110—6 GPMICLK cycles 111—7 GPMICLK cycles
BOOT_CFG2[4:3]	OEM	Boot search count	Yes	00	00—2 01—2 10—4 11—8
BOOT_CFG2[2:1]	OEM	Pages in block	Yes	00	00—128 01—64 10—32 11—256
BOOT_CFG2[0]	OEM	Reset time	Yes	0	0—12 ms 1—22 ms (LBA NAND)

1. The setting can be overridden by the GPIO settings when the BT_FUSE_SEL fuse is intact. See Table 3 for the corresponding GPIO pin.

8.5.2.2 NAND flash boot flow and Boot Control Blocks (BCB)

There are two BCB data structures:

- FCB
- DBBT

As a part of the NAND media initialization, the ROM driver uses safe NAND timings to search for the Firmware Configuration Block (FCB) that contains the optimum NAND timings, the page address of the Discovered Bad Block Table (DBBT) Search Area, and the start page address of the primary and secondary firmware.

The hardware ECC level to use is embedded inside the FCB block. The FCB data structure is also protected using the ECC. The driver reads raw 2112 bytes of the first sector and runs through the software ECC engine that determines whether the FCB data is valid or not.

If the FCB is found, the optimum NAND timings are loaded for further reads. If the ECC fails, or the fingerprints do not match, the Block Search state machine increments the page number to the Search Stride number of pages to read for the next BCB until the SearchCount pages have been read.

If the search fails to find a valid FCB, the NAND driver responds with an error and the boot ROM enters the serial download mode.

The FCB contains the page address of the DBBT Search Area, and the page address for primary and secondary boot images. The DBBT is searched in the DBBT Search Area, just like the FCB is searched. After the FCB is read, the DBBT is loaded, and the primary or secondary boot image is loaded using the starting page address from the FCB.

This figure shows the state diagram of the FCB search:

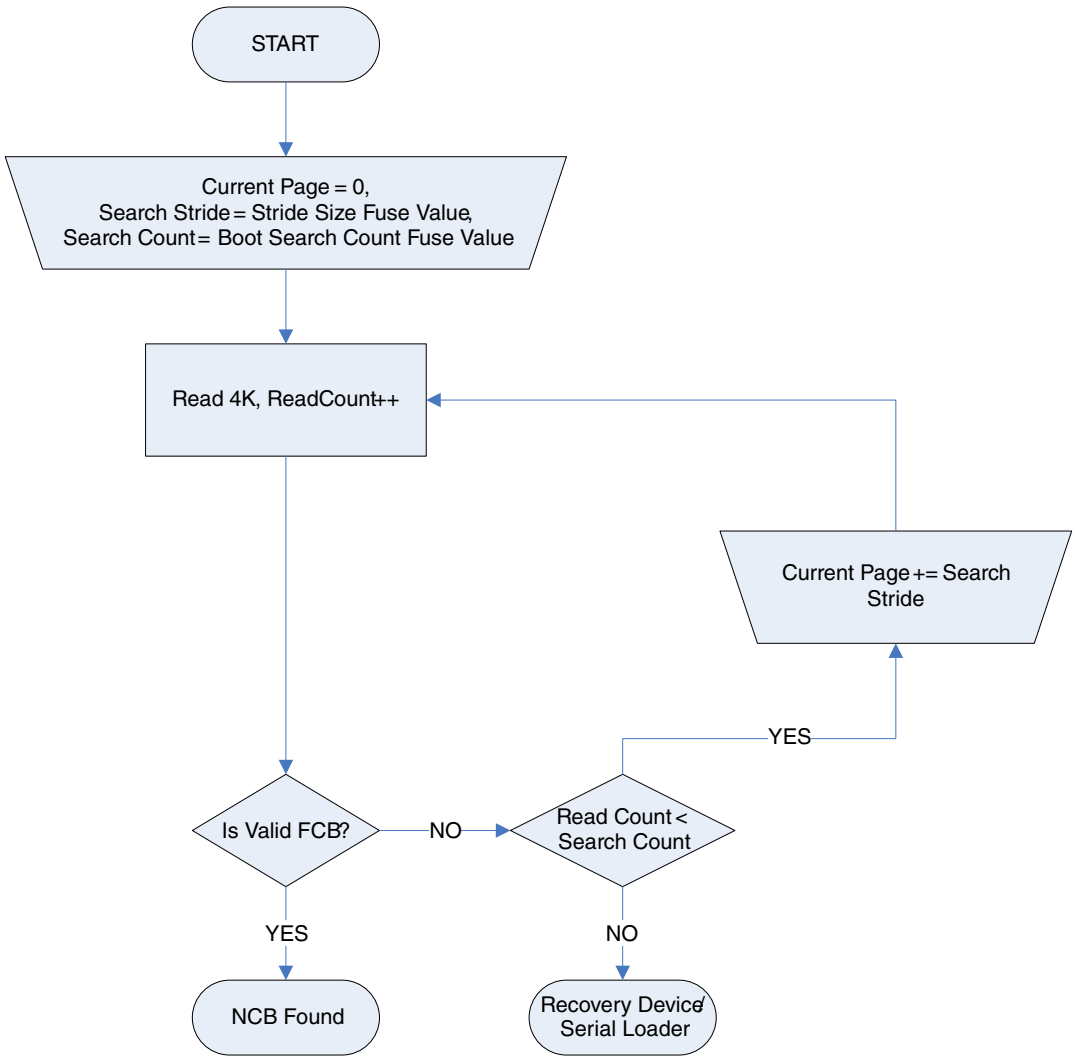


Figure 8-5. FCB search flow

When the FCB is found, the boot ROM searches for the Discovered Bad Blocks Table (DBBT). If the DBBT Search Area is 0 in the FCB, the ROM assumes that there are no bad blocks on the NAND device boot area. See this figure for the DBBT search flow:

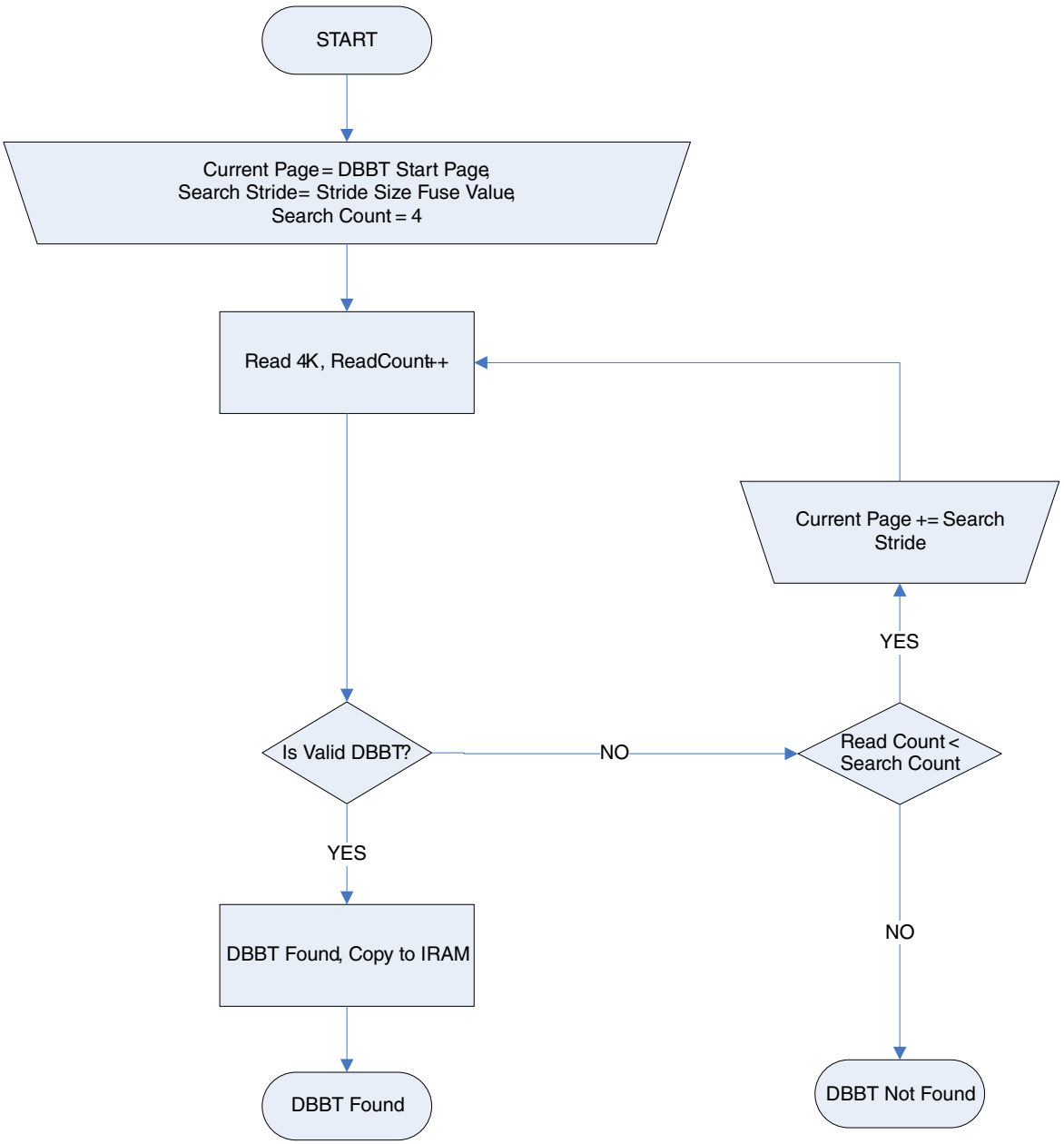


Figure 8-6. DBBT search flow

The BCB search and load function also monitors the ECC correction threshold and sets the PERSIST_BLOCK_REWRITE persistent bit if the threshold exceeds the maximum ECC correction ability.

If there is a page with a number of errors higher than ECC can correct during the primary image read, the boot ROM turns on the PERSIST_SECONDARY_BOOT bit and performs the software reset (After the software reset, the secondary image is used).

If there is a page with number of errors higher than ECC can correct during secondary image read, the boot ROM goes to the serial loader.

8.5.2.3 Firmware configuration block

The FCB is the first sector in the first good block. The FCB must be present at each search stride of the search area.

The search area contains copies of the FCB at each stride distance, so, in case the first NAND block becomes corrupted, the ROM finds its copy in the next NAND block. The search area must span over at least two NAND blocks. The location information for the DBBT search area, FW1, and FW2 are all specified in the FCB. This table shows the flash control block structure:

Table 8-11. Flash control block structure

Name	Start byte	Size in bytes	Description
Reserved	0	4	Reserved for Fingerprint #1(Checksum)
FingerPrint	4	4	32-bit word with a value of 0x20424346, in ascii "FCB"
Version	8	4	32-bit version number; this version of FCB is 0x00000001
m_NANDTiming	12	8	8 B of data for eight NAND timing parameters from the NAND datasheet. The eight parameters are: m_NandTiming[0]=data_setup, m_NandTiming[1]=data_hold, m_NandTiming[2]=address_setup, m_NandTiming[3]=dsample_time, m_NandTiming[4]=nand_timing_state, m_NandTiming[5]=REA, m_NandTiming[6]=RLOH, m_NandTiming[7]=RHOH. The ROM only uses the first four parameters, but the FCB provides space for other four parameters to be used by the bootloader or other applications.
PageDataSize	20	4	The number of bytes of data in a page. Typically, this is 2048 bytes for 2112 bytes page size or 4096 bytes for 4314/4224 bytes page size or 8192 for 8568 bytes page size.
TotalPageSize	24	4	The total number of bytes in a page. Typically, 2112 for 2-KB page or 4224 or 4314 for 4-KB page or 8568 for 8-KB page.

Table continues on the next page...

Table 8-11. Flash control block structure (continued)

Name	Start byte	Size in bytes	Description
SectorsPerBlock	28	4	The number of pages per block. Typically 64 or 128 or depending on the NAND device type.
NumberOfNANDs	32	4	Not used by ROM
TotalInternalDie	36	4	Not used by ROM
CellType	40	4	Not used by ROM
EccBlockNEccType	44	4	Value from 0 to 16 is used to set the BCH Error Corrrection level 0, 2, 4, .. or 40 for Block BN of ECC page, used in configuring the BCH40 page layout registers.
EccBlock0Size	48	4	Size of block B0 used in configuring the BCH40 page-layout registers.
EccBlockNSize	52	4	Size of block BN used in configuring the BCH40 page-layout registers.
EccBlock0EccType	56	4	Value from 0 to 16 used to set the BCH Error Corrrection level 0, 2, 4, .. or 40 for Block BN of ECC page, used in configuring the BCH40 page layout registers.
MetadataBytes	60	4	Size of metadata bytes used in configuring the BCH40 page-layout registers.
NumEccBlocksPerPage	64	4	Number of the ECC blocks BN not including B0. This value is used in configuring the BCH40 page-layout registers.
EccBlockNEccLevelSDK	68	4	Not used by ROM
EccBlock0SizeSDK	72	4	Not used by ROM
EccBlockNSizeSDK	76	4	Not used by ROM
EccBlock0EccLevelSDK	80	4	Not used by ROM
NumEccBlocksPerPageSDK	84	4	Not used by ROM
MetadataBytesSDK	88	4	Not used by ROM
EraseThreshold	92	4	Not used by ROM
Firmware1_startingPage	104	4	Page number address where the first copy of bootable firmware is located.
Firmware2_startingPage	108	4	Page number address where the second copy of bootable firmware is located.
PagesInFirmware1	112	4	Size of the first copy of firmware in pages.
PagesInFirmware2	116	4	Size of the second copy of firmware in pages.
DBBTSearchAreaStartAddress	120	4	Page address for the bad block table search area.
BadBlockMarkerByte	124	4	This is an input offset in the BCH page for the ROM to swap with the first byte of metadata after reading a page using the BCH40. The ROM supports the restoration of manufacturer-marked bad block markers in the page and this offset is the bad block marker offset location.

Table continues on the next page...

Table 8-11. Flash control block structure (continued)

Name	Start byte	Size in bytes	Description
BadBlockMarkerStartBit	128	4	This is an input bit offset in the BadBlockMarkerByte for the ROM to use when swapping eight bits with the first byte of metadata.
BBMarkerPhysicalOffset	132	4	This is the offset where the manufacturer leaves the bad block marker on a page.
BCHType	136	4	0 for BCH20 and 1 for BCH40. The chip is backwards compatible to BCH20 and this field tells the ROM to use the BCH20 or BCH40 block.
TMTiming2_ReadLatency	140	4	Toggle mode NAND timing parameter read latency, the ROM uses this value to configure the timing2 register of the GPMI.
TMTiming2_PreambleDelay	144	4	Toggle mode NAND timing parameter Preamble Delay. The ROM uses this value to configure the timing2 register of the GPMI.
TMTiming2_CEDelay	148	4	Toggle mode NAND timing parameter CE Delay. The ROM uses this value to configure the timing2 register of the GPMI.
TMTiming2_PostambleDelay	152	4	Toggle mode NAND timing parameter Postamble Delay. The ROM uses this value to configure the timing2 register of the GPMI.
TMTiming2_CmdAddPause	156	4	Toggle mode NAND timing parameter Cmd Add Pause. The ROM uses this value to configure the timing2 register of the GPMI.
TMTiming2_DataPause	160	4	Toggle mode NAND timing parameter Data Pause. The ROM uses this value to configure the timing2 register of the GPMI.
TMSpeed	164	4	This is the toggle mode speed for the ROM to configure the gpmi clock. 0 for 33 MHz, 1 for 40 MHz, and 2 for 66 MHz.
TMTiming1_BusyTimeout	168	4	Toggle mode NAND timing parameter Busy Timeout. The ROM uses this value to configure the timing1 register of the GPMI.
DISBBM	172	4	If 0, the ROM swaps the BadBlockMarkerByte with metadata[0] after reading a page using the BCH40. If the value is 1, the ROM does not swap.
BBMark_spare_offset	176	4	The offset in the metadata place which stores the data in the bad block marker place.
Onfi_sync_enable	180	4	Enable the Onfi nand sync mode support.
Onfi_sync_speed	184	4	Speed for the Onfi nand sync mode: 0 - 24 MHz, 1 - 33 MHz, 2 - 40 MHz, 3 - 50 MHz, 4 - 66 MHz, 5 - 80 MHz, 6 - 100 MHz, 7 - 133 MHz, 8 - 160 MHz, 9 - 200 MHz

Table continues on the next page...

Table 8-11. Flash control block structure (continued)

Name	Start byte	Size in bytes	Description
Onfi_syncNANDData	188	28	The parameters for the Onfi nand sync mode timing. They are read latency, ce_delay, preamble_delay, postamble_delay, cmdadd_pause, data_pause, and busy_timeout.
DISBB_Search	216	4	Disable the bad block search function when reading the firmware, only using DBBT.

8.5.2.4 Discovered Bad Block Table (DBBT)

See this table for the DBBT format:

Table 8-12. DBBT structure

Name	Start byte	Size in bytes	Description
reserved	0	4	-
FingerPrint	4	4	32-bit word with a value of 0x44424254,in ascii "DBBT"
Version	8	4	32-bit version number; this version of DBBT is 0x00000001
reserved	12	4	-
DBBT_NUM_OF_PAGES	16	4	Size of the DBBT in pages
reserved	20	4*PageSize-20	-
reserved	4*PageSize	4	-
Number of Entries	4*PageSize + 4	4	Number of bad blocks
Bad Block Number	4*PageSize + 8	4	First bad block number
Bad Block Number	4*PageSize + 12	4	Second bad block number
-	-	-	Next bad block number
-	-	-	-
Last bad block number	-	-	Last bad block number

8.5.2.5 Bad block handling in ROM

During the firmware boot, at the block boundary, the Bad Block table is searched for a match to the next block.

If no match is found, the next block can be loaded. If a match is found, the block must be skipped and the next block checked.

If the Bad Block table start page is null, check the manufactory made Bad Block marker. The location of the Bad Block maker is at the first three or last three pages in every block of the NAND flash. The NAND manufacturers normally use one byte in the spare area of certain pages within a block to mark that a block is bad or not. A value of 0xFF means good block, non-FF means bad block.

To preserve the BI (bad block information), the flash updater or gang programmer applications must swap the Bad Block Information (BI) data to byte 0 of the metadata area for every page before programming the NAND flash. When the ROM loads the firmware, it copies back the value at metadata[0] to the BI offset in the page data. This figure shows how the factory bad block marker is preserved:

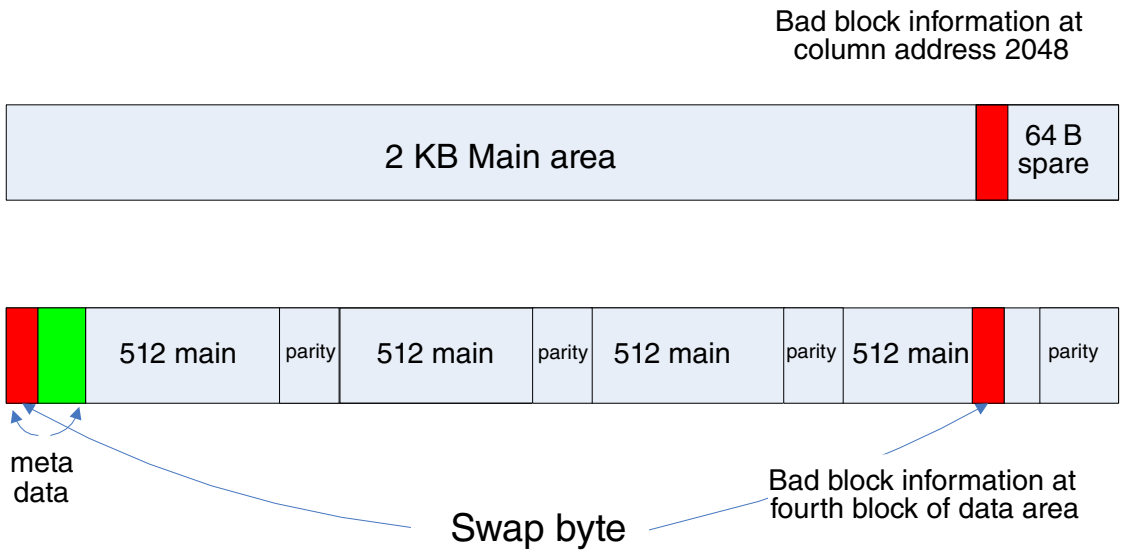


Figure 8-7. Factory bad block marker preservation

In the FCB structure, there are two elements (m_u32BadBlockMarkerByte and m_u32BadBlockMarkerStartBit) to indicate the byte and bit place in the page data that the manufacturer marked the bad block marker.

8.5.2.6 Toggle mode DDR NAND boot

If the BT_TOGGLEMODE efuse is blown, the ROM does the following to boot from the Samsung's toggle mode DDR NAND.

8.5.2.6.1 GPMI and BCH clocks configuration

The ROM sets the clock source and the dividers in the CCM registers.

If the BOOT_CFG1[5] is set (toggle mode), the GPMI/BCH CLK source is PLL2PFD4, and running at 66 MHz, otherwise the GPMI/ BCH CLK souce is PLL3, running at 24 MHz. The ROM sets the default values to timing0, timing1, and timing2 gpmi registers for 24 MHz clock speed. It uses the BOOT_CFG2[7:5] fuse to configure the GPMI timing2 register parameters preamble delay and read latency. The default value for these parameters is 2 when the fuses are not blown.

The default timing parameter values used by the ROM for the toggle-mode device are:

- Timing0.ADDRESS_SETUP = 5
- Timing0.DATA_SETUP = 10
- Timing0.DATA_HOLD = 10
- Timing1.DEVICE_BUSY_TIMEOUT = 0 x 500
- Timing2.READ_LATENCY = BOOT_CFG2[7:5] if blown, otherwise 2
- Timing2.CE_DELAY = 2
- Timing2.PREAMBLE_DELAY = BOOT_CFG2[7:5] if blown, otherwise 2
- Timing2.POSTAMBLE_DELAY = 3
- Timing2.CMDADD_PAUSE = 4
- Timing2.DATA_PAUSE = 6

The default timing parameters can be overriden by the TMTiming2_ReadLatency, TMTiming2_PreambleDelay, TMTiming2_CEDelay, TMTiming2_PostambleDelay, TMTiming2_CmdAddPause, and TMTiming2_DataPause parameters of the FCB.

8.5.2.6.2 Setup DMA for DDR transfers

In the DMA descriptors, the GPMI is configured to read the page data at a double data rate, the word length is set to 16, and the transfer count to a half of the page size.

8.5.2.6.3 Reconfigure timing and speed using values in FCB

After reading the FCB page with the GPMI set to default timings and a speed of 33 MHz, the ROM reconfigures the CCM dividers to run the gpmi/bch clks to a desired speed specified in the FCB for the rest of the boot process. The GPMI timing registers are also reconfigured to the values specified in the FCB.

The GPMI speed can be configured using the FCB parameter TMSpeed:

- 0—24 MHz
- 1—33 MHz
- 2—40 MHz
- 3—50 MHz
- 4—66 MHz
- 5—80 MHz

- 6—100 MHz
- 7—133 MHz
- 8—160 MHz
- 9—200 MHz

The GPMI timing0 register fields data_setup, data_hold, and address_setup are set to the values specified for the data_setup and data_hold and address_setup in the FCB member m_NANDTiming.

The GPMI timing1.DEVICE_BUSY_TIMEOUT is set to the value specified in the FCB member TMTiming1_BusyTimeout.

The GPMI timing2 register values are set using the FCB members TMTiming2.READ_LATENCY, CE_DELAY, PREAMBLE_DELAY, POSTAMBLE_DELAY, CMDADD_PAUSE, and DATA_PAUSE.

8.5.2.7 Typical NAND page organization

8.5.2.7.1 BCH ECC page organization

The first data block is called block 0 and the rest of the blocks are called block N. A separate ECC level scan is used for block 0 and block N.

The metadata bytes must be located at the beginning of a page, starting at byte 0, followed by the data block 0, the ECC bytes for data block 0, the block 1 and its ECC bytes, and so on, up until the N data blocks. The ECC level for the block 0 can be different from the ECC level for the rest of the blocks.

For the NAND boot with page-size restrictions and the data block size restricted to 512 B, only few combinations of the ECC for block 0 and block N are possible.

This figure shows the valid layout for 2112-byte sized page.

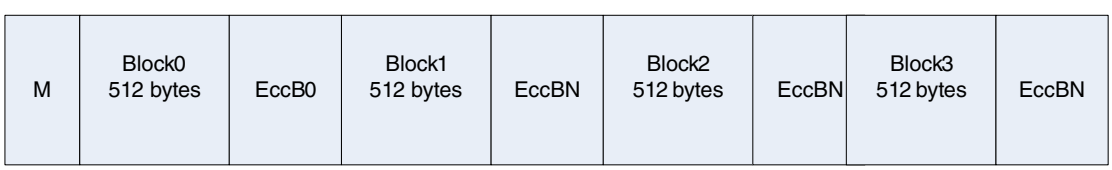


Figure 8-8. Valid layout for 2112-byte sized page

The example below is for 13 bits of parity (GF13). The number of ECC bits required for a data block is calculated using the (ECC_Correction_Level * 13) bits.

In the above layout, the ECC size for EccB0 and EccBN must be selected to not exceed a total page size of 2112 bytes. The EccB0 and EccBN can be one of the 2, 4, 6, 8, 10, 12, 14, 16, 18, and 20 bits on the ECC correction level. The total bytes are:

[M + (data_block_size x 4) + ([EccB0 + (EccBN x 3)] x 13) / 8] <= 2112;

M = metadata bytes and data_block_size is 512.

There are four data blocks of 512 bytes each in a page of 2-KB page sized NAND. The values of EccB0 and EccBN must be such that the above calculation does not result in a value greater than 2112 bytes.

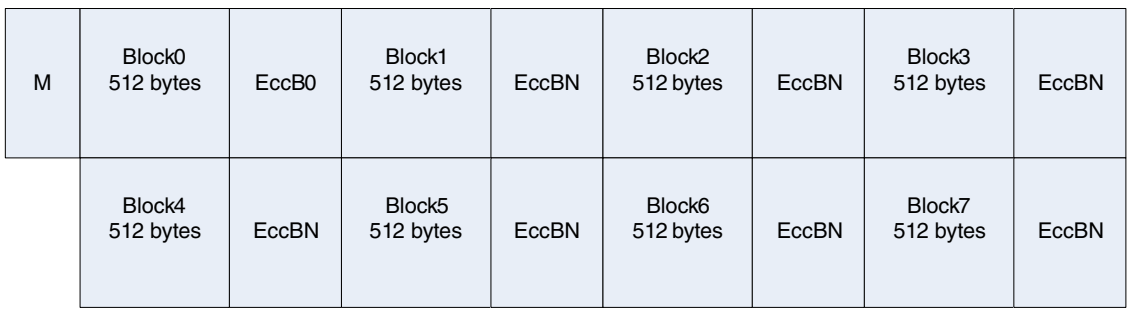


Figure 8-9. Valid layout for 4-KB sized page

Different NAND manufacturers have different sizes for a 4-KB page; 4314 bytes is typical.

[M + (data_block_size x 8) + ([EccB0 + (EccBN x 7)] x 13) / 8] <= 4314;

M= metadata bytes and data_block_size is 512.

There are eight data blocks of 512 bytes each in a page of a 4-KB page sized NAND. The values of the EccB0 and EccBN must be such that the above calculation does not result in a value greater than the size of a page in a 4-KB page NAND.

8.5.2.7.2 Metadata

The number of bytes used for the metadata is specified in the FCB. The metadata for the BCH encoded pages is placed at the beginning of a page. The ROM only cares about the first byte of metadata to swap it with a bad block marker byte in the page data after each page read; it is important to have at least one byte for the metadata bytes field in the FCB data structure.

8.5.2.8 IOMUX configuration for NAND

The following table shows the RawNAND IOMUX pin configuration. The NAND boot is only supported on chip select 0.

Table 8-13. NAND IOMUX pin configuration

Signal	Pad name
CLE	NANDF_CLE.alt0
ALE	NANDF_ALE.alt0
WPN	NANDF_WP_B.alt0
RD_N	SD4_CMD.alt1
WRN	SD4_CLK.alt1
READY0	NANDF_RB0.alt0
DQS	SD4_DAT0.alt2
CE0N	NANDF_CS0.alt0
CE1N	NANDF_CS1.alt0
CE2N	NANDF_CS2.alt0
CE3N	NANDF_CS3.alt0
D0	NANDF_D0.alt0
D1	NANDF_D1.alt0
D2	NANDF_D2.alt0
D3	NANDF_D3.alt0
D4	NANDF_D4.alt0
D5	NANDF_D5.alt0
D6	NANDF_D6.alt0
D7	NANDF_D7.alt0

8.5.3 Expansion device

The ROM supports booting from the MMC/eMMC and SD/eSD compliant devices.

8.5.3.1 Expansion device eFUSE configuration

The SD/MMC/eSD/eMMC/SDXC boot can be performed using either the USDHC ports, based on the setting of the BOOT_CFG2[4:3] (Port Select) fuse or it is associated to the GPIO input value at the boot.

All USDHC ports support the eMMC4.3 and eMMC4.4 fast boot. See this table for details:

Table 8-14. USDHC boot eFUSE descriptions

Fuse	Config	Definition	GPIO[1]	Shipped value	Settings
BOOT_CFG1[7:6]	OEM	Boot device selection	Yes	00	01 - Boot from the USDHC interfaces
BOOT_CFG1[5]	OEM	SD/MMC selection	Yes	0	0 - SD/eSD/SDXC 1 - MMC/eMMC
BOOT_CFG1[4]	OEM	Fast boot support	Yes	0	0 - Normal boot 1 - Fast boot
BOOT_CFG1[3:2]	OEM	SD/MMC speed mode	Yes	00	MMC 0 - High-speed mode 1 - Normal-speed mode SD 00 - Normal (SDR12) 01 - High (SDR25) 10 - SDR50 11 - SDR104
BOOT_CFG1[1]	OEM	SD power cycle enable/ eMMC reset enable	Yes	0	MMC 0 - eMMC reset disabled 1 - eMMC reset enabled via the SD_RST pad (on USDHC3 and USDHC4 only) SD 0 - No power cycle 1 - Power cycle enabled via the SD_RST pad (on USDHC3 and USDHC4 only)
BOOT_CFG1[0]	OEM	SD loopback clock source sel (for SDR50 and SDR104 only)	Yes	0	0 - Through the SD pad 1 - Direct
BOOT_CFG2[7:5]	OEM	Bus width/SD calibration step	Yes	000	SD/eSD/SDXC (BOOT_CFG1[5]=0) Bus width xx0 - 1-bit xx1 - 4-bit SD calibration step 00x - 1 delay cells 01x - 1 delay cells 10x - 2 delay cells 11x - 3 delay cells MMC/eMMC (BOOT_CFG1[5]=1) 000 - 1-bit 001 - 4-bit 010 - 8-bit

Table continues on the next page...

Table 8-14. USDHC boot eFUSE descriptions (continued)

Fuse	Config	Definition	GPIO[1]	Shipped value	Settings
					101 - 4-bit DDR (MMC 4.4) 110 - 8-bit DDR (MMC 4.4) Else - reserved
BOOT_CFG2[4:3]	OEM	Port select	Yes	00	00 - USDHC-1 01 - USDHC-2 10 - USDHC-3 11 - USDHC-4
BOOT_CFG2[2]	OEM	Boot frequencies	Yes	0	0 - Boot ROM default 1 - Apply value per fuse field MMC_DLL_DLY[6:0]
BOOT_CFG2[1]	OEM	Boot acknowledge disable/ pull down during power cycle enable	Yes	0	MMC 0 - Boot acknowledge enabled 1 - Boot acknowledge disabled SD 0 - Use default SD pad settings during the power cycle 1 - Set pull-down on SD pads during the power cycle (used only if "SD Power Cycle Enable" is enabled)
BOOT_CFG2[0]	OEM	Override pad settings	Yes	0	0 - Use default values 1 - Use PAD_SETTINGS values
MMC_DLL_DLY[6:0]	OEM	MMC DLL value/UHSI calibaration start value	No	0000000	MMC DLL value/UHSI calibration start value

1. The setting can be overridden by the GPIO settings when the BT_FUSE_SEL fuse is intact. See GPIO boot overrides for the corresponding GPIO pin.

The boot code supports these standards:

- MMCv4.4 or less
- eMMCv4.4 or less
- SDv2.0 or less
- eSDv2.10 rev-0.9, with or without FAST_BOOT
- SDXCv3.0

The MMC/SD/eSD/SDXC/eMMC can be connected to any of the USDHC blocks and can be booted by copying 4 KB of data from the MMC/SD/eSD/eMMC device to the internal RAM. After checking the Image Vector Table header value (0xD1) from program image, the ROM code performs a DCD check. After a successful DCD

extraction, the ROM code extracts from the Boot Data Structure the destination pointer and length of image to be copied to the RAM device from where the code execution occurs.

The maximum image size to load into the SD/MMC boot is 32 MB. This is due to a limited number of uSDHC ADMA Buffer Descriptors allocated by the ROM.

NOTE

The initial 4 KB of the program image must contain the IVT, DCD, and the Boot Data structures.

Table 8-15. SD/MMC frequencies

	SD	MMC	MMC (DDR mode)
Identification (KHz)	347.22		
Normal-speed mode (MHz)	25	20	25
High-speed mode (MHz)	50	40	50
UHSI SDR50 (MHz)	100		
UHSI SDR104 (MHz)	200		

NOTE

The boot ROM code reads the application image length and the application destination pointer from the image.

8.5.3.2 MMC and eMMC boot

This table provides the MMC and eMMC boot details.

Table 8-16. MMC and eMMC boot details

Normal boot mode	During the initialization (normal boot mode), the MMC frequency is set to 347.22 KHz. When the MMC card enters the identification portion of the initialization, the voltage validation is performed, and the ROM boot code checks the high-voltage settings and the card capacity. The ROM boot code supports both the high-capacity and low-capacity MMC/eMMC cards. After the initialization phase is complete, the ROM boot code switches to a higher frequency (20 MHz in the normal boot mode or 40 MHz in the high-speed mode). The eMMC is also interfaced via the USDHC and follows the same flow as the MMC. The boot partition can be selected for an MMC4.x card after the card initialization is complete. The ROM code reads the BOOT_PARTITION_ENABLE field in the Ext_CSD[179] to get the boot partition to be set. If there is no boot partition mentioned in the BOOT_PARTITION_ENABLE field or the user partition was mentioned, the ROM boots from the user partition.

Table continues on the next page...

Table 8-16. MMC and eMMC boot details (continued)

eMMC4.3 or eMMC4.4 device supporting special boot mode	If using an eMMC4.3 or eMMC4.4 device that supports the special boot mode, it can be initiated by pulling the CMD line low. If the BOOT ACK is enabled, the eMMC4.3/eMMC4.4 device sends the BOOT ACK via the DATA lines and the ROM can read the BOOT ACK [S010E] to identify the eMMC4.3/eMMC4.4 device. If the BOOT ACK is enabled, the ROM waits 50 ms to get the BOOT ACK. If BOOT ACK is disabled ROM waits 1 second for data. If the BOOT ACK or data was received, the eMMC4.3/eMMC4.4 is booted in the "boot mode", otherwise the eMMC4.3/eMMC4.4 boots as a normal MMC card from the selected boot partition. This boot mode can be selected by the BOOT_CFG1[4] (fast boot) fuse. The BOOT ACK is selected by the BOOT_CFG2[1] fuse.
eMMC4.4 device	If using the eMMC4.4 device, the Double Data Rate (DDR) mode can be used. This mode can be selected by the BOOT_CFG2[7:5] (bus width) fuse.

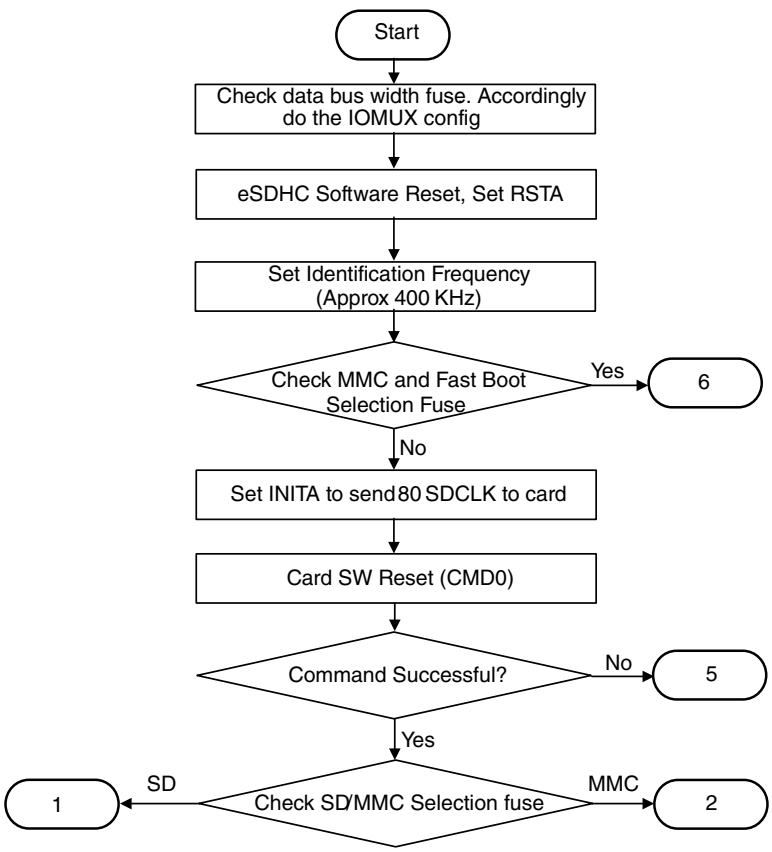


Figure 8-10. Expansion device boot flow (1 of 6)

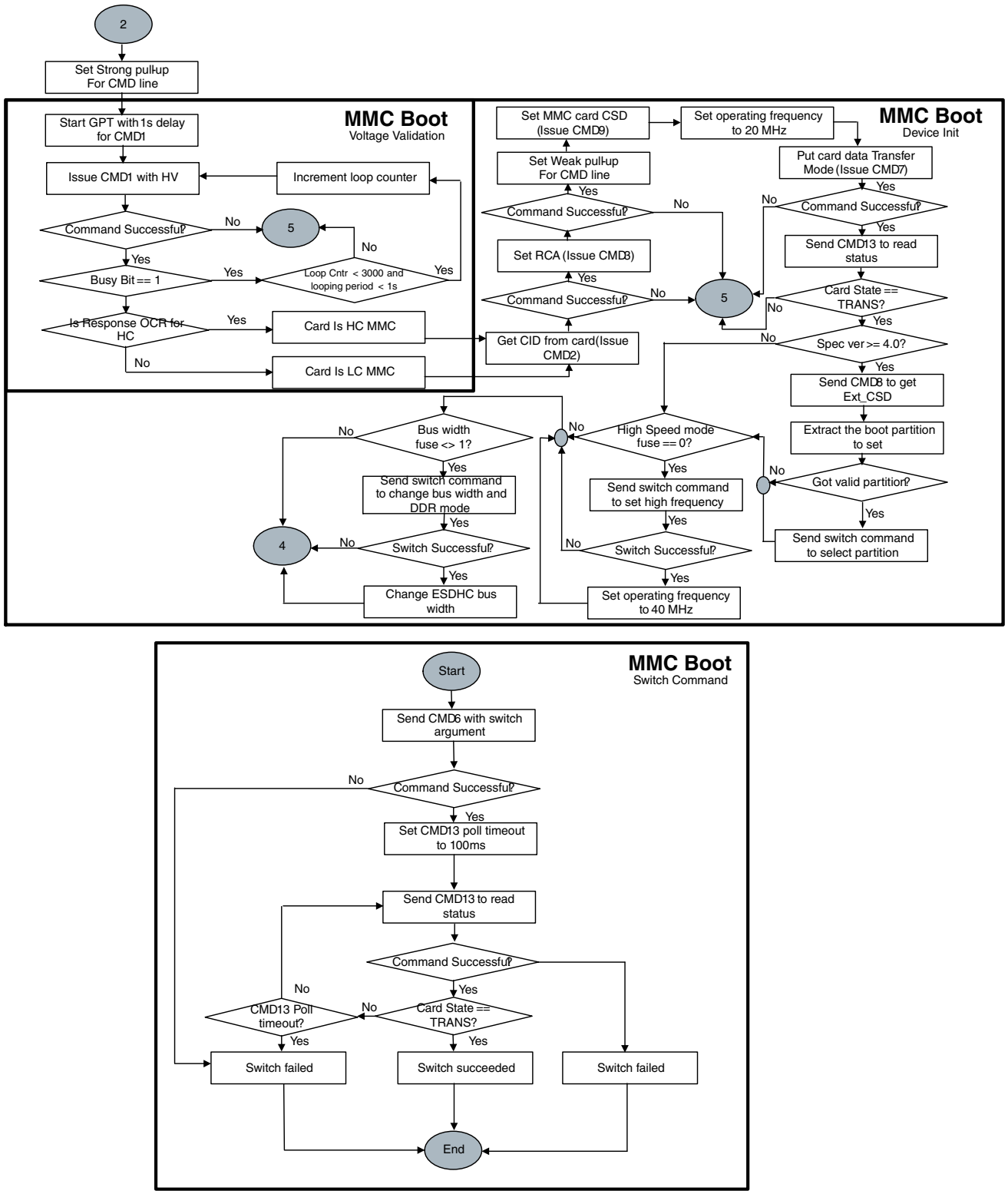


Figure 8-11. Expansion device (MMC) boot flow (2 of 6)

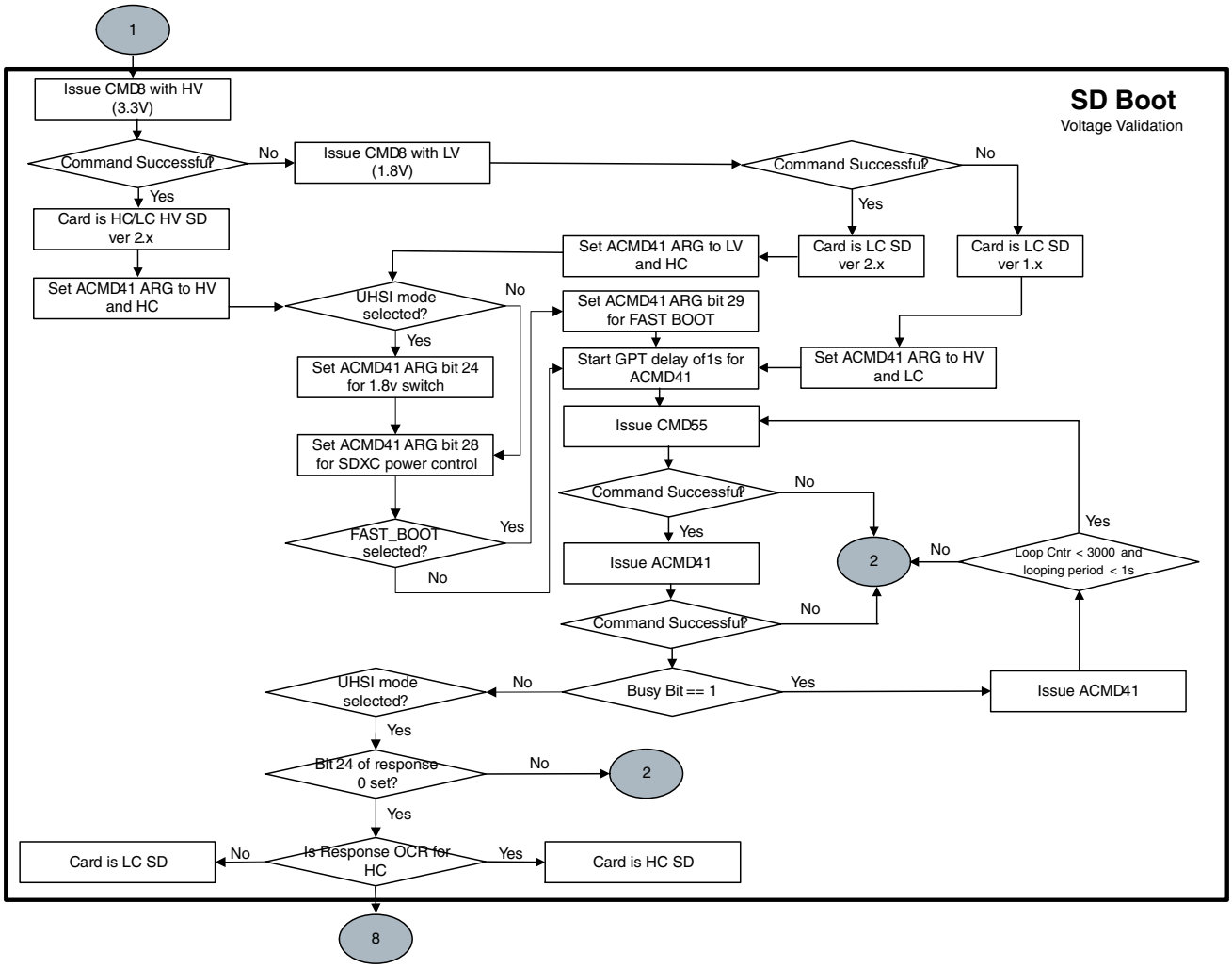


Figure 8-12. Expansion device (SD/eSD/SDXC) boot flow (3 of 6) part 1

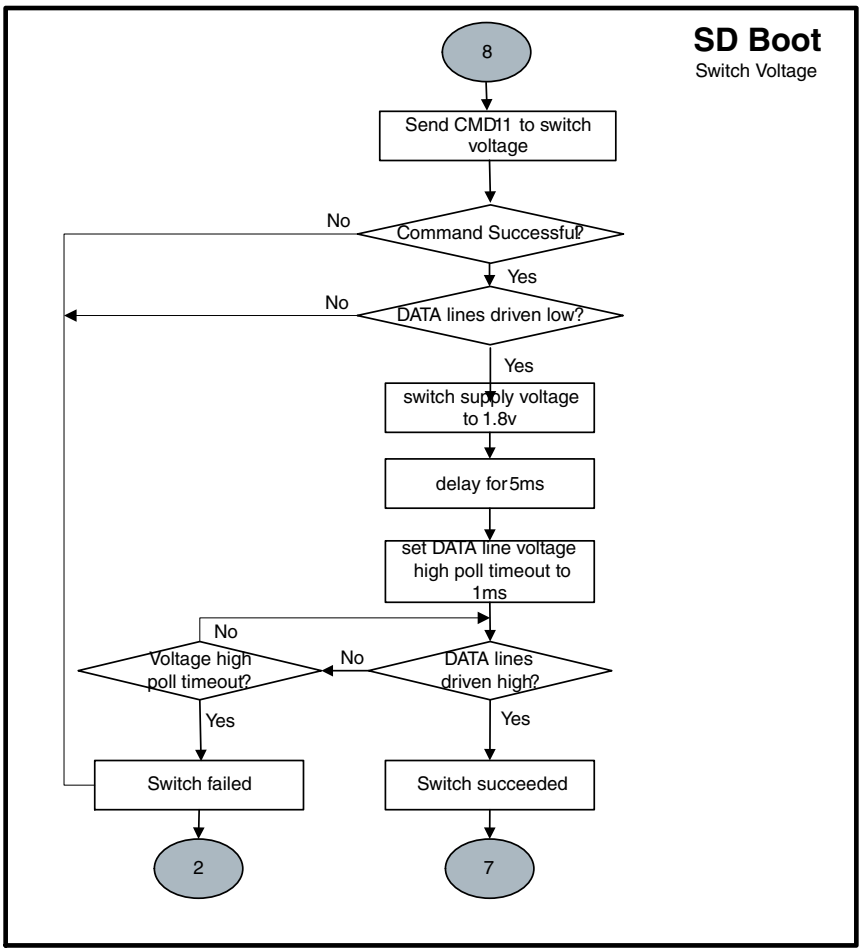


Figure 8-13. Expansion device (SD/eSD/SDXC) boot flow (3 of 6) part 2

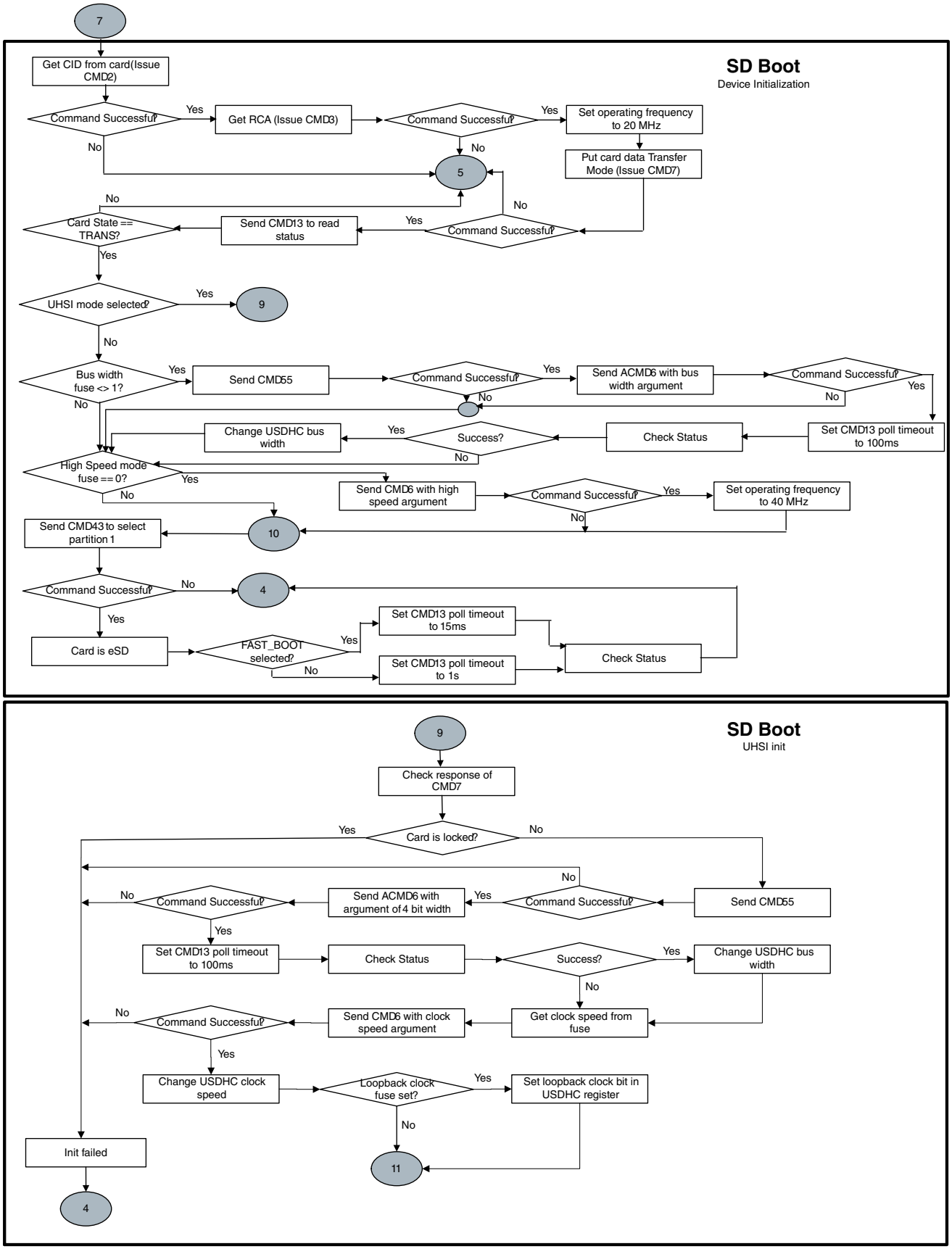


Figure 8-14. Expansion device (MMCSD/eSD/SDXC) boot flow (4 of 6)

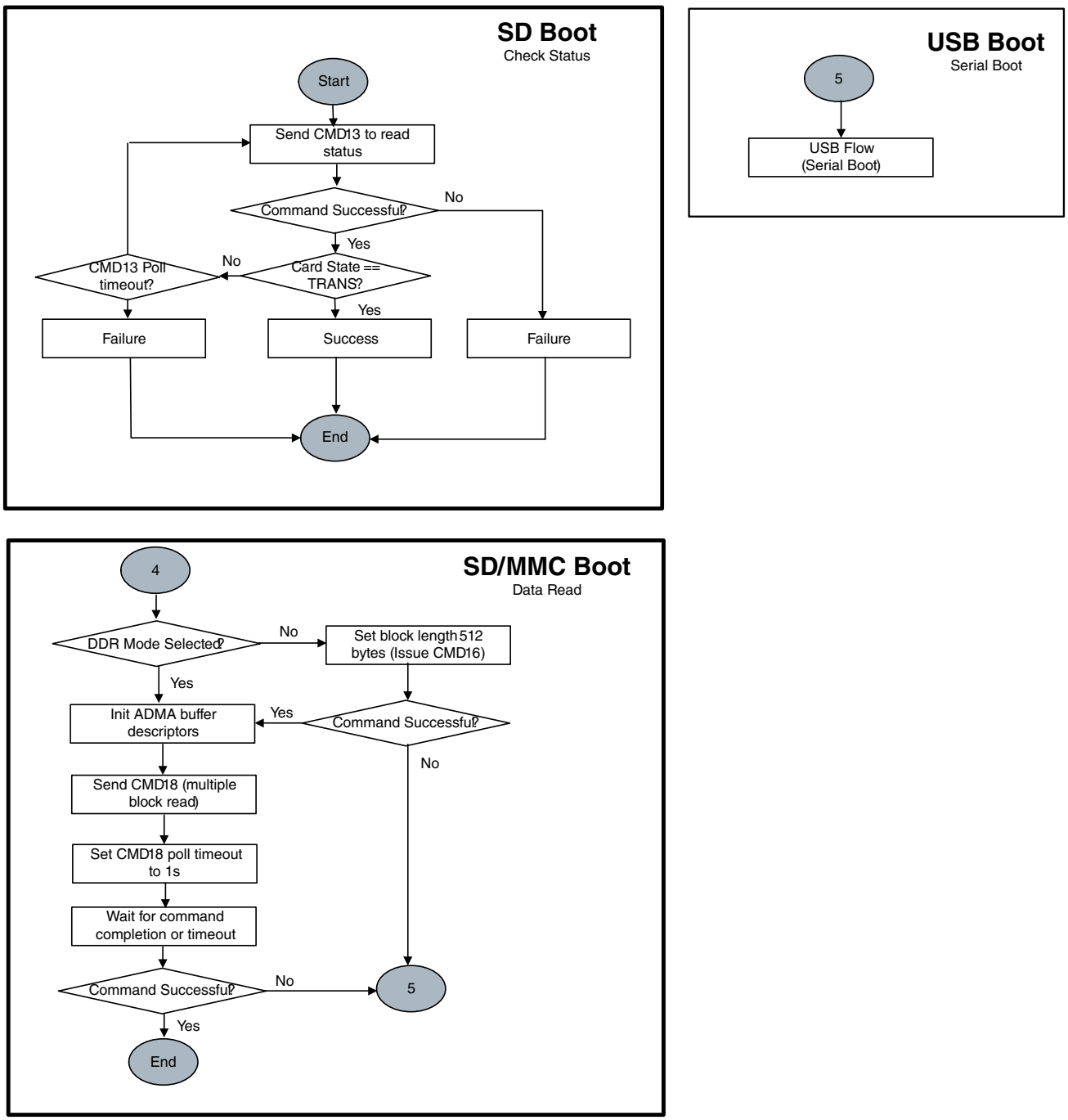
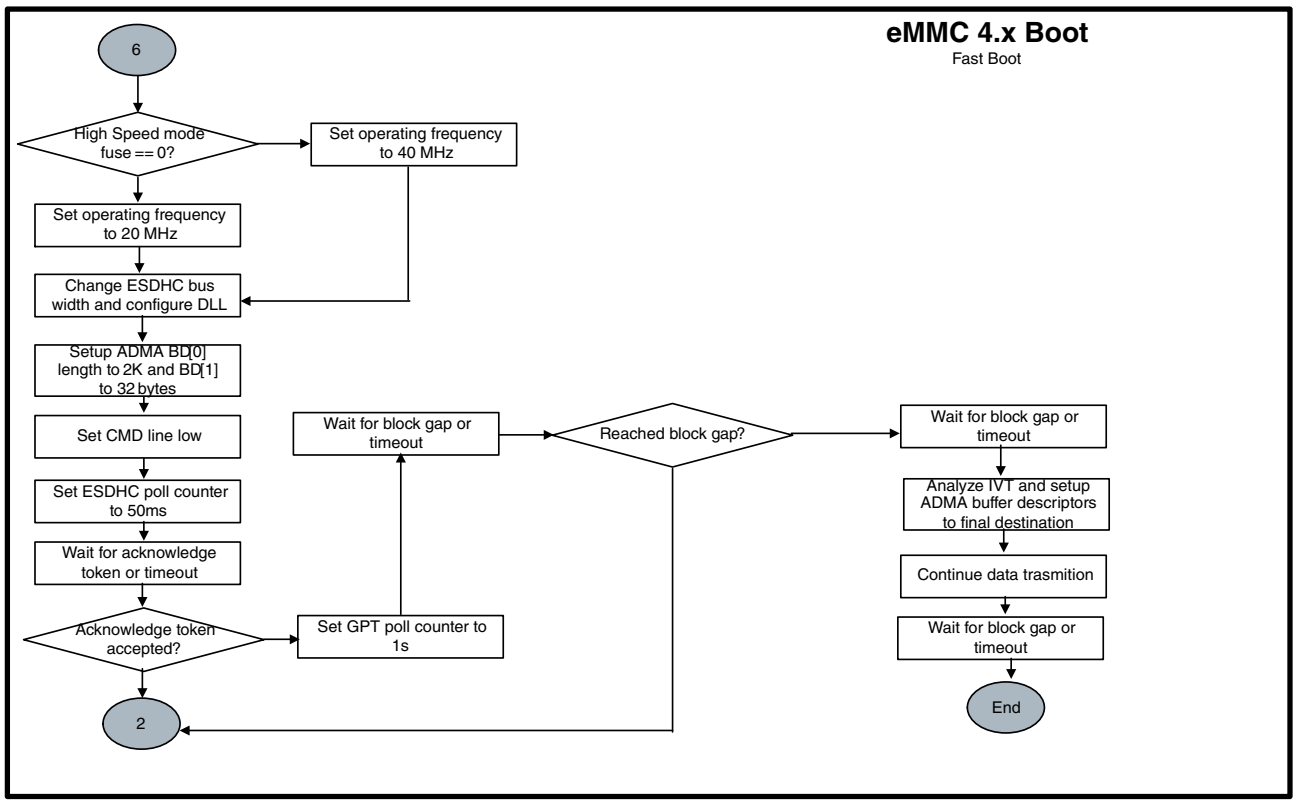


Figure 8-15. Expansion device (SD/eSD) boot flow (5 of 6)



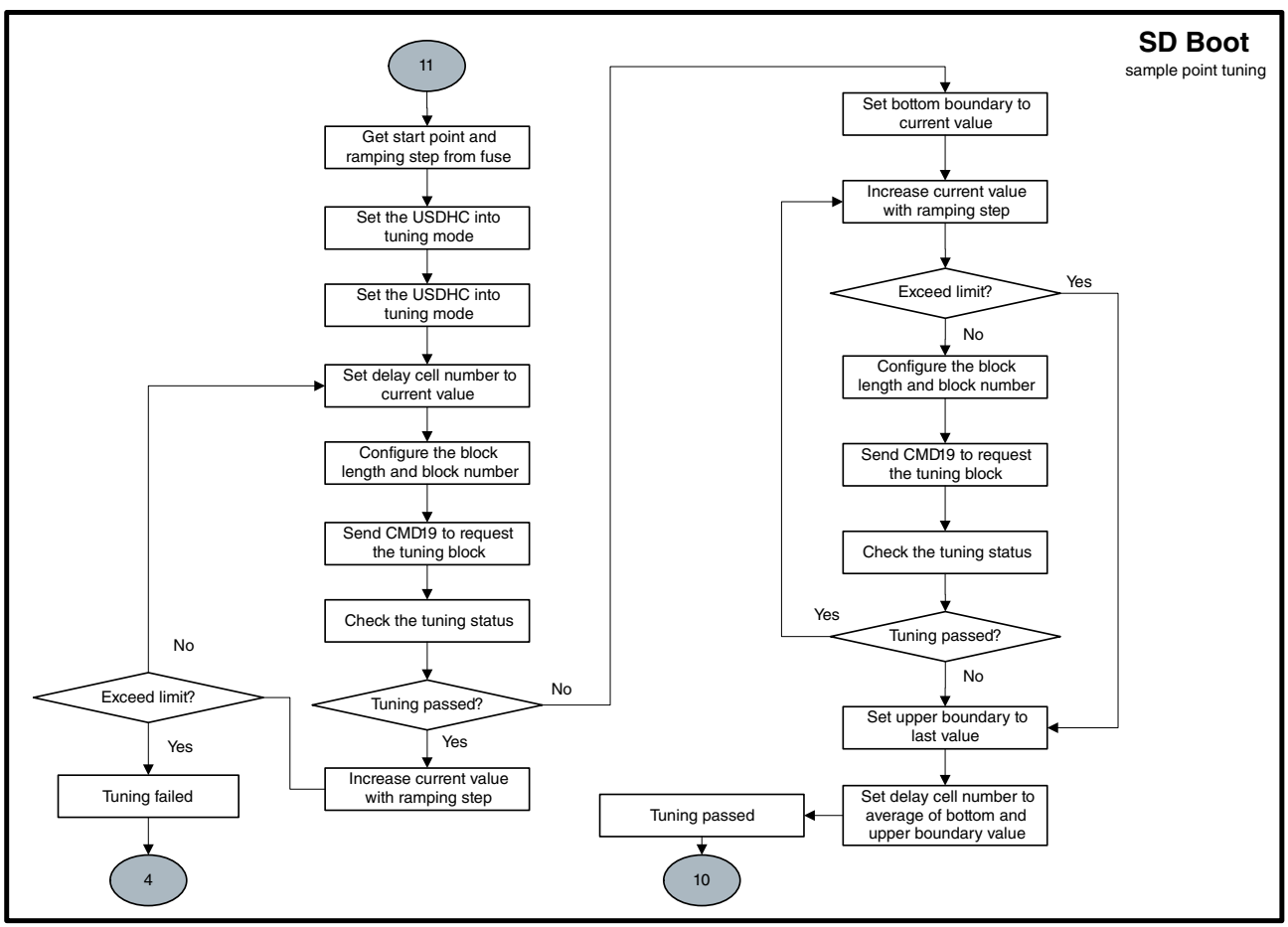


Figure 8-16. Expansion device boot flow (6 of 6)

8.5.3.3 SD, eSD, and SDXC

After the normal boot mode initialization begins, the SD/eSD/SDXC frequency is set to 347.22 kHz. During the identification phase, the SD/eSD/SDXC card voltage validation is performed. During the voltage validation, the boot code first checks with the high-voltage settings; if that fails, it checks with the low-voltage settings.

The capacity of the card is also checked. The boot code supports the high-capacity and low-capacity SD/eSD/SDXC cards after the voltage validation card initialization is done.

During the card initialization, the ROM boot code attempts to set the boot partition for all SD, eSD, and SDXC devices. If this fails, the boot code assumes that the card is a normal SD or SDXC card. If it does not fail, the boot code assumes it is an eSD card. After the initialization phase is over, the boot code switches to a higher frequency (25 MHz in the normal-speed mode or 50 MHz in the high-speed mode). The ROM also supports the FAST_BOOT mode booting from the eSD card. This mode can be selected by the BOOT_CFG1[4] (Fast Boot) fuse described in Table 8-14 .

For the UHSI cards, the clock speed fuses can be set to SDR50 or SDR104 on USDHC3 and USDHC4 ports. This enables the voltage switch process to set the signaling voltage to 1.8 V during the voltage validation. The bus width is fixed at a 4-bit width and a sampling point tuning process is needed to calibrate the number of the delay cells. If the SD Loopback Clock eFuse is set, the feedback clock comes directly from the loopback SD clock, instead of the card clock (by default). The SD clock speed can be selected by the BOOT_CFG1[3:2], and the SD Loopback Clock is selected by the BOOT_CFG1[0].

NOTE

SDR50 and SDR104 boot are not supported on USDHC1 and USDHC2, because there is no reset signal for those ports in the IOMUXC.

The UHSI calibration start value (MMC_DLL_DLY[6:0]) and the step value (BOOT_CFG2[7:5]) can be set to optimize the sample point tuning process.

If the SD Power Cycle Enable eFuse is 1, the ROM sets the SD_RST pad low, waits for 5 ms, and then sets the SD_RST pad high. If the SD_RST pad is connected to the SD power supply enable logic on board, it enables the power cycle of the SD card. This may be crucial in case the SD logic is in the 1.8 V states and must be reset to the 3.3 V states.

8.5.3.4 IOMUX configuration for SD/MMC

Table 8-17. SD/MMC IOMUX pin configuration

Signal	USDHC-1	USDHC-2	USDHC-3	USDHC-4
CLK	SD1_CLK.alt0	SD2_CLK.alt0	SD3_CLK.alt0	SD4_CLK.alt0
CMD	SD1_CMD.alt0	SD2_CMD.alt0	SD3_CMD.alt0	SD4_CMD.alt0
DAT0	SD1_DAT0.alt0	SD2_DAT0.alt0	SD3_DAT0.alt0	SD4_DAT0.alt1
DAT1	SD1_DAT1.alt0	SD2_DAT1.alt0	SD3_DAT1.alt0	SD4_DAT1.alt1
DAT2	SD1_DAT2.alt0	SD2_DAT2.alt0	SD3_DAT2.alt0	SD4_DAT2.alt1
DAT3	SD1_DAT3.alt0	SD2_DAT3.alt0	SD3_DAT3.alt0	SD4_DAT3.alt1
DAT4	NANDF_D0.alt1	NANDF_D4.alt1	SD3_DAT4.alt0	SD4_DAT4.alt1
DAT5	NANDF_D1.alt1	NANDF_D5.alt1	SD3_DAT5.alt0	SD4_DAT5.alt1
DAT6	NANDF_D2.alt1	NANDF_D6.alt1	SD3_DAT6.alt0	SD4_DAT6.alt1
DAT7	NANDF_D3.alt1	NANDF_D7.alt1	SD3_DAT7.alt0	SD4_DAT7.alt1
VSELECT			GPIO_18.alt2	NANDF_CS1.alt1
RESET [1]			SD3_RESET.alt0	NANDF_ALE.alt1
CD	GPIO_1.alt6	GPIO_4.alt6	-	-

1. Active-low

8.5.3.5 Redundant boot support for expansion device

The ROM supports the redundant boot for an expansion device. The primary or secondary image is selected, depending on the PERSIST_SECONDARY_BOOT setting. (see Table 8-6).

If the PERSIST_SECONDARY_BOOT is 0, the boot ROM uses address 0x400 for the primary image.

If the PERSIST_SECONDARY_BOOT is 1, the boot ROM reads the secondary image table from address 0x200 on the boot media and uses the address specified in the table for the secondary image.

Table 8-18. Secondary image table format

Reserved (chipNum)
Reserved (driveType)
tag
firstSectorNumber
Reserved (sectorCount)

Where:

- The tag is used as an indication of the valid secondary image table. It must be 0x00112233.
- The firstSectorNumber is the first 512-byte sector number of the secondary image.

For the secondary image support, the primary image must reserve the space for the secondary image table. See this figure for the typical structures layout on an expansion device.

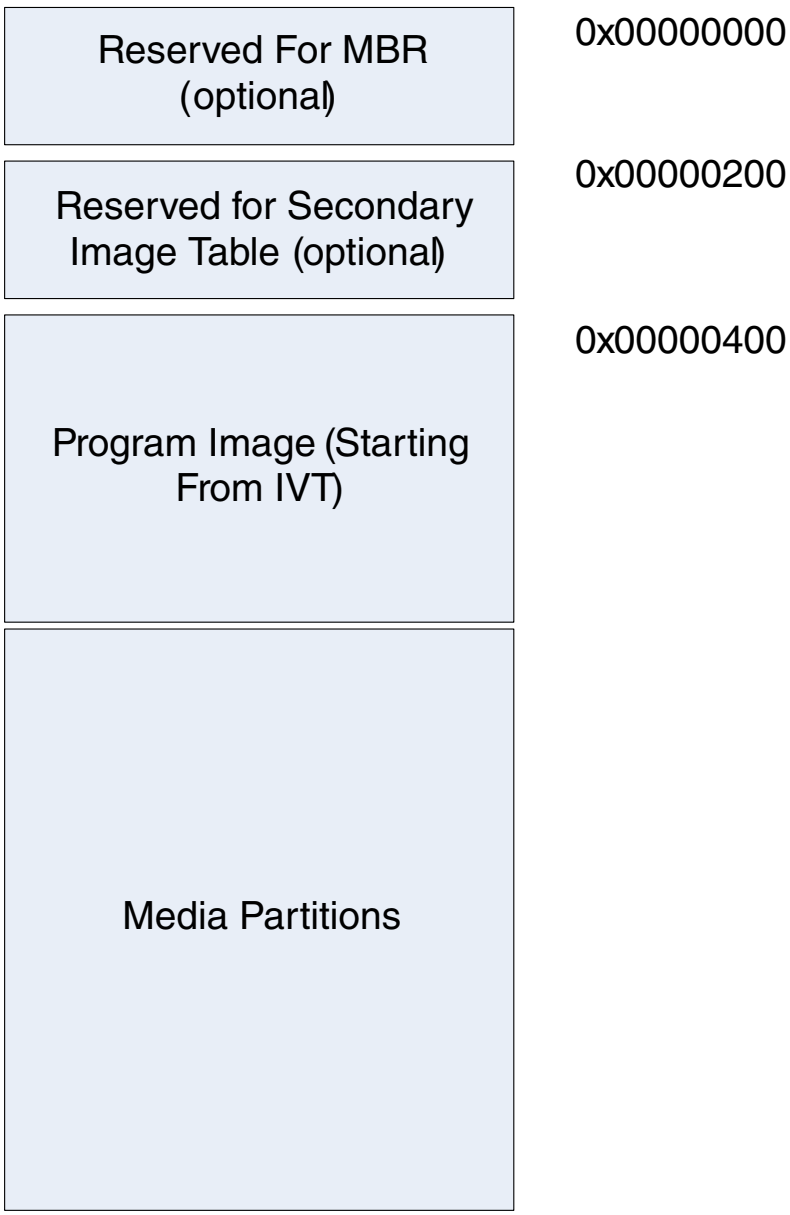


Figure 8-17. Expansion device structures layout

For the Closed mode, if there are failures during primary image authentication, the boot ROM turns on the PERSIST_SECONDARY_BOOT bit (see Table 8-6) and performs the software reset. (After the software reset, the secondary image is used.)

8.5.4 Serial ROM through SPI and I²C

The chip supports booting from serial memory devices, such as EEPROM and serial flash, using the SPI.

These ports are available for serial boot: eCSPI (eCSPI1, eCSPI2, eCSPI3, eCSPI4) and I²C controller (I2C1, I2C2 and I2C3) interfaces.

8.5.4.1 Serial ROM eFUSE configuration

The boot ROM code determines the type of device using the following parameters, either provided by the eFUSE settings or sampled on the I/O pins, during boot.

See this table for details:

Table 8-19. Serial ROM boot eFUSE descriptions

Fuse	Config	Definition	GPIO[1]	Shipped value	Settings
BOOT_CFG1[7:4]	OEM	Boot device selection	Yes	0000	0011 - Boot from the serial ROM
BOOT_CFG4[6]	OEM	EEPROM recovery enable	Yes	0	0 - Disabled EEPROM recovery 1 - Enabled EEPROM recovery
BOOT_CFG4[5:4]	OEM	CS select (SPI only)	Yes	00	00 - ECSPIx_SS0 01 - ECSPIx_SS1 10 - ECSPIx_SS2 11 - ECSPIx_SS3
BOOT_CFG4[3]	OEM	SPI addressing (SPI only)	Yes	0	0 - 2 B (16-bit) 1 - 3 B (24-bit)
BOOT_CFG4[2:0]	OEM	Port select	Yes	00	000 - ECSPI-1 001 - ECSPI-2 010 - ECSPI-3 011 - ECSPI-4 101 - I2C1 110 - I2C2 111 - I2C3

1. The setting can be overridden by the GPIO settings when the BT_FUSE_SEL fuse is intact. See GPIO Boot Overrides for the corresponding GPIO pin.

The ECPSI-1/ECPSI-2/ECPSI-3/ECPSI-4 block can be used as a boot device using the ECSPI interface for the serial ROM boot. The SPI interface is configured to operate at 15 MHz for 3-byte addressing devices and at 3.75 MHz for 2-byte addressing devices.

The I2C-1/I2C-2/I2C-3 block can be used as a boot device using the I^2C interface for the serial ROM boot. The I^2C interface is configured to operate at 343.75 kbit/s.

The boot ROM copies 4 KB of data from the serial ROM device to the internal RAM. After checking the Image Vector Table header value (0xD1) from the program image, the ROM code performs a DCD check. After a successful DCD extraction, the ROM code extracts the destination pointer and length of image from the Boot Data Structure to be copied to the RAM device from where the code execution occurs.

NOTE

The Initial 4 KB of program image must contain the IVT, DCD, and the Boot Data Structures.

8.5.4.2 I^2C boot

The boot flow when booting from the I^2C device is shown in Figure 8-18.

The boot ROM code reads the fuses BOOT_CFG1[7:4] (Boot Device Selection) and BOOT_CFG1[7:4] (Port select) to detect the EEPROM device type. The ROM program copies 4 KB of data from the EEPROM device to the internal RAM. The boot ROM code then copies the initial 4 KB of data as well as the rest of the image directly to the application destination extracted from the application image.

The chip uses the Device Select Code/Device Address in the following table to boot from the EEPROM.

Table 8-20. EEPROM via I^2C device select code

Bits	Device type identifier				Chip enable address[1]			R/W
	7	6	5	4	3	2	1	0
Device select code	1	0	1	0	0	0	0	R/W

1. These address bits must be configured at the memory device to match this '000' value.

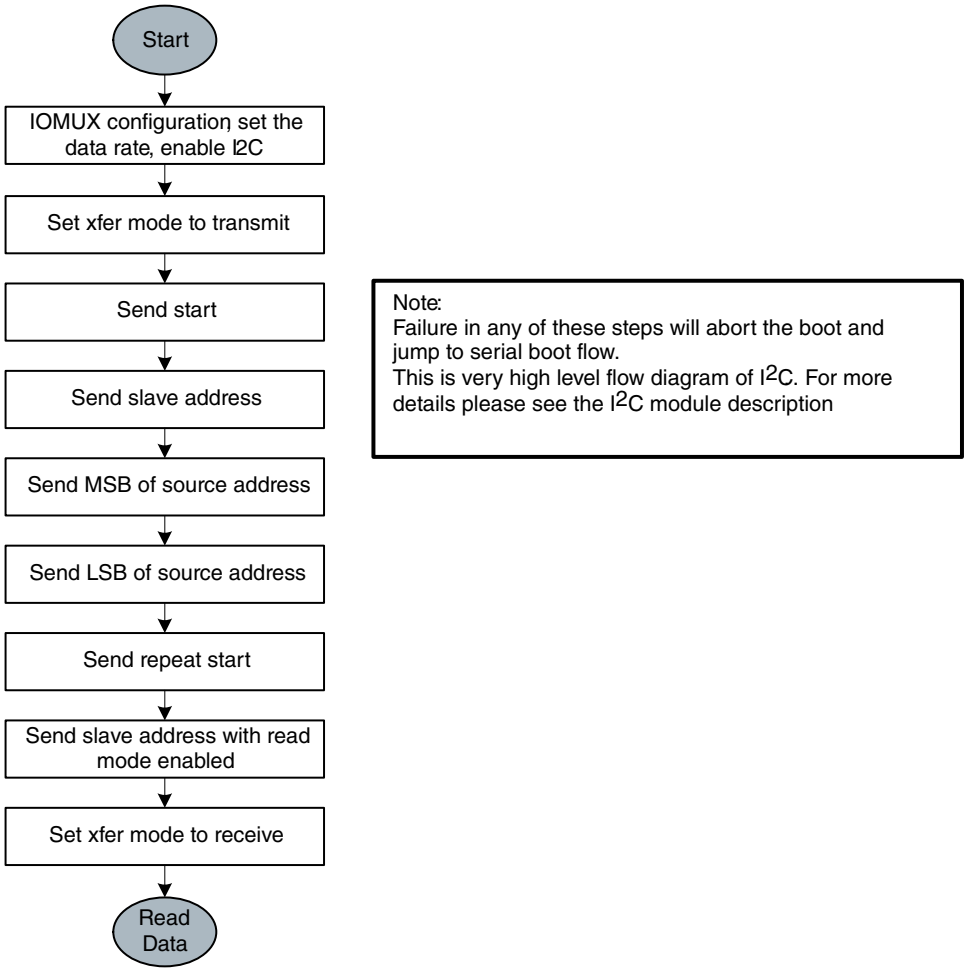


Figure 8-18. I²C flow chart

8.5.4.2.1 I²C IOMUX pin configuration

The contacts assigned to the signals used by the I²C blocks are shown in these tables:

Table 8-21. I²C IOMUX pin configuration

Signal	I2C-1	I2C-2	I2c-3
SDA	EIM_D28.alt1	EIM_D16.alt6	EIM_D18.alt6
SCL	EIM_D21.alt6	EIM_EB2.alt6	EIM_D17.alt6

8.5.4.3 ECSPI boot

The Enhanced Configurable SPI (ECSPI) interface is configured in the master mode and the EEPROM device is connected to the ECSPI interface as a slave.

The boot ROM code copies 4 KB of data from the EEPROM device to the internal RAM. If the DCD verification is successful, the ROM code copies the initial 4 KB of data, as well as the rest of the image extracted from the application image, directly to the application destination. The ECSPI can read data from the EEPROM using 2- or 3-byte addressing. Its burst length is 32 B.

NOTE

The Serial ROM Chip Select Number is determined by the BOOT_CFG4[5:4] (Chip Select) fuse.

When using the SPI as a boot device, the chip supports booting from both the serial EEPROM and serial flash devices. The boot code determines which device is being used by reading the appropriate eFUSE/I/O values at the boot (see Table 8-19 for details).

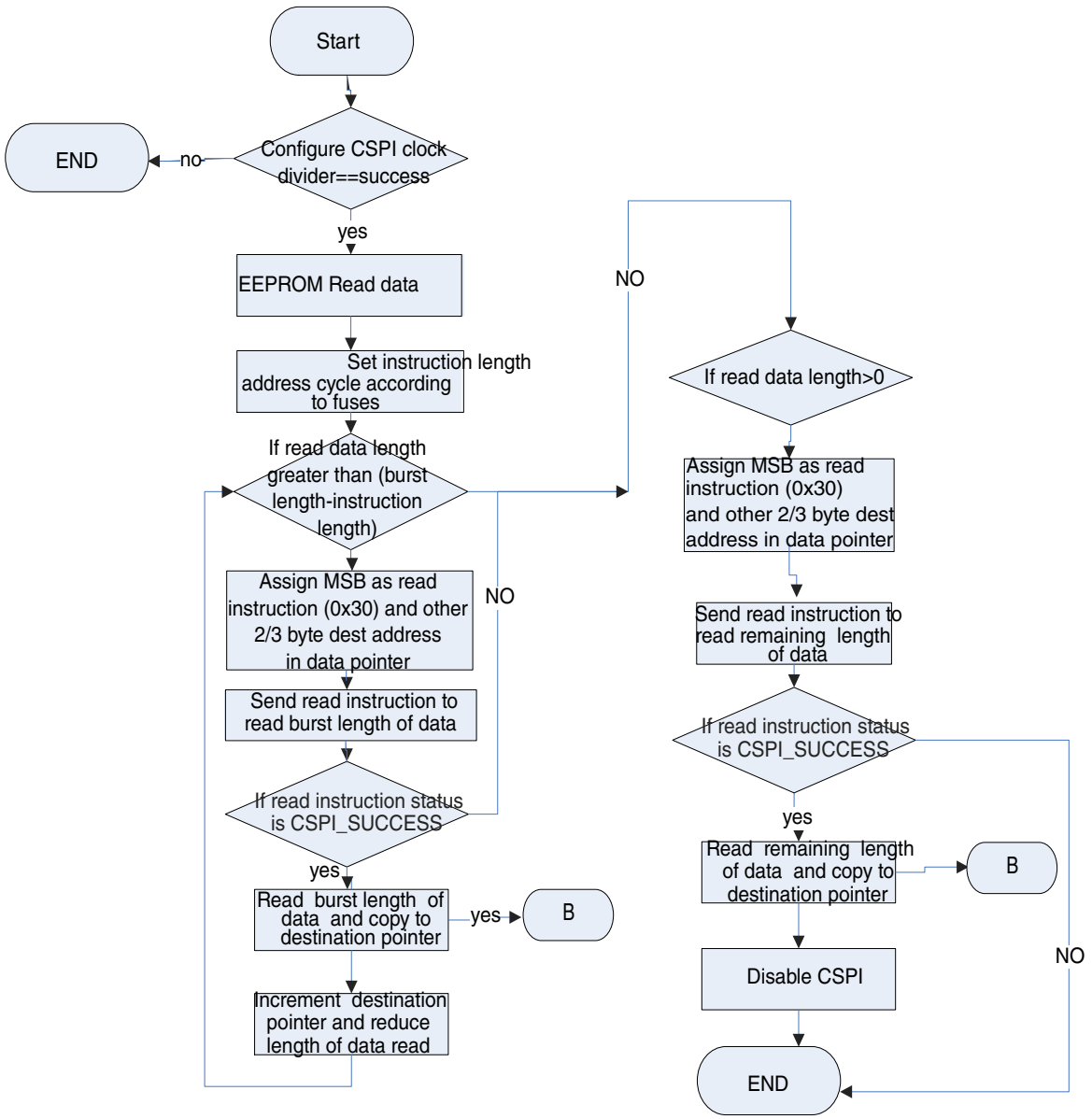


Figure 8-19. CSPI flow chart

8.5.4.3.1 ECSPI IOMUX pin configuration

The contacts assigned to the signals used by the CSPI blocks are shown in this table:

Table 8-22. ECSPI IOMUX pin configuration

Signal	ECSPI-1	ECSPI-2	ECSPI-3	ECSPI4	ECSPI-5
MISO	EIM_D17.alt1	CSI0_DAT10.alt2	DISP0_DAT2.alt2	EIM_D22.alt1	N/A
MOSI	EIM_D18.alt1	CSI0_DAT9.alt2	DISP0_DAT1.alt2	EIM_D28.alt2	N/A

Table continues on the next page...

Table 8-22. ECSPI IOMUX pin configuration (continued)

RDY	N/A[1]	N/A	N/A	N/A	N/A
SCLK	EIM_D16.alt1	CSI0_DAT8.alt2	DISP0_DAT0.alt2	EIM_D21.alt1	N/A
SS0	EIM_EB2.alt1	CSI0_DAT11.alt2	DISP0_DAT3.alt2	EIM_D20.alt1	N/A
SS1	EIM_D19.alt1	EIM_LBA.alt1	DISP0_DAT4.alt2	EIM_A25.alt1	N/A
SS2	EIM_D24.alt1	EIM_D24.alt4	DISP0_DAT5.alt2	EIM_D24.alt1	N/A
SS3	EIM_D25.alt1	EIM_D25.alt4	DISP0_DAT6.alt2	EIM_D25.alt1	N/A

1. The N/A in the ROM code indicates that the pins are not available or not used.

8.6 Program image

This section describes the data structures that are required to be included in the user's program image. The program image consists of:

- Image vector table—a list of pointers located at a fixed address that the ROM examines to determine where the other components of the program image are located.
- Boot data—a table that indicates the program image location, program image size in bytes, and the plugin flag.
- Device configuration data—IC configuration data.
- User code and data.

8.6.1 Image Vector Table and Boot Data

The Image Vector Table (IVT) is the data structure that the ROM reads from the boot device supplying the program image containing the required data components to perform a successful boot.

The IVT includes the program image entry point, a pointer to Device Configuration Data (DCD) and other pointers used by the ROM during the boot process.The ROM locates the IVT at a fixed address that is determined by the boot device connected to the Chip. The IVT offset from the base address and initial load region size for each boot device

type is defined in the table below. The location of the IVT is the only fixed requirement by the ROM. The remainder or the image memory map is flexible and is determined by the contents of the IVT.

Table 8-23. Image Vector Table Offset and Initial Load Region Size

Boot Device Type	Image Vector Table Offset	Initial Load Region Size
NOR	4 Kbyte = 0x1000 bytes	Entire Image Size
NAND	1 Kbyte = 0x400 bytes	4 Kbyte
OneNAND	256 bytes = 0x100 bytes	1 Kbyte
SD/MMC/eSD/eMMC/SDXC	1 Kbyte = 0x400 bytes	4 Kbyte
I2C/SPI EEPROM	1 Kbyte = 0x400 bytes	4 Kbyte

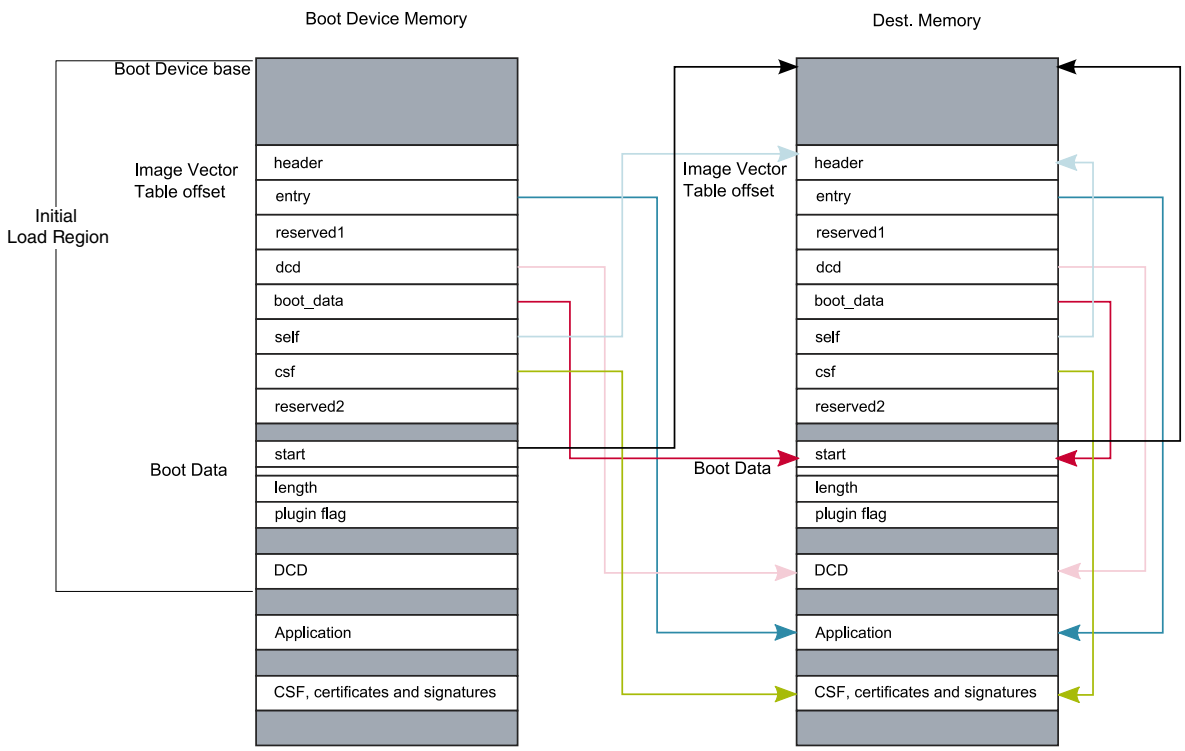


Figure 8-20. Image Vector Table

8.6.1.1 Image vector table structure

The IVT has the following format where each entry is a 32-bit word:

Table 8-24. IVT format

header
entry: Absolute address of the first instruction to execute from the image
reserved1: Reserved and should be zero
dcd: Absolute address of the image DCD. The DCD is optional so this field may be set to NULL if no DCD is required. See Device Configuration Data (DCD) for further details on the DCD.
boot data: Absolute address of the boot data
self: Absolute address of the IVT. Used internally by the ROM.
csf: Absolute address of the Command Sequence File (CSF) used by the HAB library. See High-Assurance Boot (HAB) for details on the secure boot using HAB. This field must be set to NULL when not performing a secure boot
reserved2: Reserved and should be zero

Figure 8-21 shows the IVT header format:

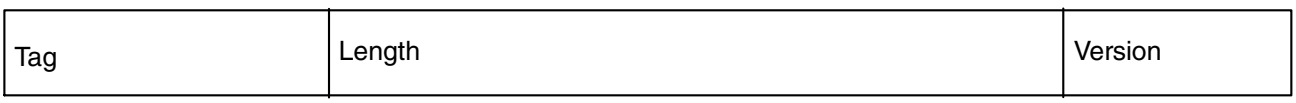


Figure 8-21. IVT header format

where:

```
Tag: A single byte field set to 0xD1
Length: a two byte field in big endian format containing the overall length of the IVT,
in bytes, including the header. (the length is fixed and must have a value of
32 bytes)
Version: A single byte field set to 0x40 or 0x41
```

8.6.1.2 Boot data structure

The boot data must follow the format defined in the table found here, each entry is a 32-bit word.

Table 8-25. Boot data format

start	Absolute address of the image
length	Size of the program image
plugin	Plugin flag (see Plugin image)

8.6.2 Device Configuration Data (DCD)

Upon reset, the chip uses the default register values for all peripherals in the system. However, these settings typically are not ideal for achieving the optimal system performance and there are even some peripherals that must be configured before they can be used.

The DCD is a configuration information contained in the program image (external to the ROM) that the ROM interprets to configure various peripherals on the chip.

For example, the EIM default settings allow the core to interface to a NOR flash device immediately after the reset. This allows the chip to interface with any NOR flash device, but has the disadvantage of slow performance. Additionally, some components (such as DDR) require some sequence of register programming as a part of the configuration before it is ready to be used. The DCD feature can be used to program the EIM registers and the MMDC registers to the optimal settings.

The ROM determines the location of the DCD table based on the information located in the Image Vector Table (IVT). See Image Vector Table and Boot Data for more details. The DCD table shown below is a big-endian byte array of the allowable DCD commands. The maximum size of the DCD is limited to 1768 B.



Figure 8-22. DCD data format

The DCD header is 4 B with the following format:

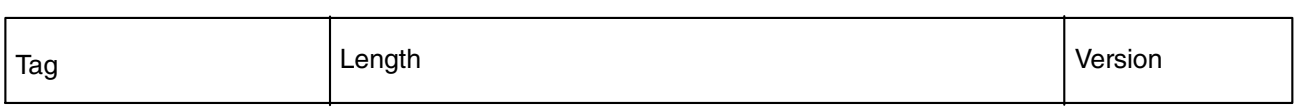


Figure 8-23. DCD header format

where:

Program image

```
Tag: A single-byte field set to 0xD2
Length: a two-byte field in the big-endian format containing the overall length of the DCD
(in bytes) including the header
Version: A single-byte field set to 0x41
```

8.6.2.1 Write data command

The write data command is used to write a list of given 1-, 2- or 4-byte values (or bitmasks) to a corresponding list of target addresses.

The format of the write data command (in a big-endian byte array) is shown in this table:

Table 8-26. Write data command format

Tag	Length	Parameter
Address		
Value/Mask		
[Address]		
[Value/Mask]		
...		
[Address]		
[Value/Mask]		

where:

```
Tag: a single-byte field set to 0xCC
Length: a two-byte field in a big-endian format, containing the length of the Write Data
Command (in bytes) including the header
Address: the target address to which the data must be written
Value/Mask: the data value (or bitmask) to be written to the preceding address
```

The parameter field is a single byte divided into the bitfields, as follows:

Table 8-27. Write data command parameter field

7	6	5	4	3	2	1	0
flags					bytes		

where

```
bytes: the width of the target locations in bytes (either 1, 2, or 4)
flags: control flags for the command behavior
Data Mask = bit 3: if set, only specific bits may be overwritten at the target address
(otherwise all bits may be overwritten)
Data Set = bit 4: if set, the bits at the target address are overwritten with this flag
(otherwise it is ignored)
```

One or more target address and value/bitmask pairs can be specified. The same bytes' and flags' parameters apply to all locations in the command.

When successful, this command writes to each target address in accordance with the flags as follows:

Table 8-28. Interpretation of write data command flags

"Mask"	"Set"	Action	Interpretation
0	0	*address = val_msk	Write value
0	1	*address = val_msk	Write value
1	0	*address &= ~val_msk	Clear bitmask
1	1	*address |= val_msk	Set bitmask

NOTE

If any of the target addresses does not have the same alignment as the data width indicated in the parameter field, none of the values are written.

If any of the values are larger or any of the bitmasks are wider than permitted by the data width indicated in the parameter field, none of the values are written.

If any of the target addresses do not lie within the allowed region, none of the values are written. The list of allowable blocks and target addresses for the chip are provided below.

Table 8-29. Valid DCD address ranges

Address range	Start address	Last address
IOMUX Control (IOMUXC) registers	0x020E0000	0x020E3FFF
CCM register set	0x020C4000	0x020C7FFF
ANADIG registers	0x020C8000	0x020C8FFF
MMDC register set	0x021B0000	0x021B7FFF
IRAM free space	0x00907000	0x00917FF0
EIM memory	0x08000000	0x0FFEFFFF
EIM registers	0x021B8000	0x021BBFFF
DDR	0x10000000	0xFFFFFFFF

8.6.2.2 Check data command

The check data command is used to test for a given 1-, 2-, or 4-byte bitmasks from a source address.

The check data command is a big-endian byte array with the format shown in this table:

Table 8-30. Check data command format

Tag	Length	Parameter
Address		
Mask		
[Count]		

where:

```
Tag: a single-byte field set to 0xCF
Length: a two-byte field in the big-endian format containing the length of the check data
command (in bytes) including the header
Address: the source address to test
Mask: the bit mask to test
Count: an optional poll count; If the count is not specified, this command polls
indefinitely
until the exit condition is met. If count = 0, this command behaves as for the NOP.
```

The parameter field is a single byte divided into bitfields, as follows:

Table 8-31. Check data command parameter field

7	6	5	4	3	2	1	0
flags					bytes		

where

```
bytes: the width of target locations in bytes (either 1, 2, or 4)
flags: control flags for the command behavior
Data Mask = bit 3: if set, only the specific bits may be overwritten at a target address
(otherwise all bits may be overwritten)
Data Set = bit 4: if set, the bits at the target address are overwritten with this flag
(otherwise it is ignored)
```

This command polls the source address until either the exit condition is satisfied, or the poll count is reached. The exit condition is determined by the flags as follows:

Table 8-32. Interpretation of check data command flags

"Mask"	"Set"	Action	Interpretation
0	0	(*address & mask) == 0	All bits clear
0	1	(*address & mask) == mask	All bits set
1	0	(*address & mask)!= mask	Any bit clear
1	1	(*address & mask)!= 0	Any bit set

NOTE

If the source address does not have the same alignment as the data width indicated in the parameter field, the value is not read.

If the bitmask is wider than permitted by the data width indicated in the parameter field, the value is not read.

8.6.2.3 NOP command

This command has no effect.

The format of the NOP command is a big-endian four-byte array, as shown in this table:

Table 8-33. NOP command format

Tag	Length	Undefined

where:

```
Tag: a single-byte field set to 0xC0
Length: a two-byte field in big endian containing the length of the NOP command in bytes
(fixed to a
value of 4)
Undefined: this byte is ignored and can be set to any value.
```

8.6.2.4 Unlock command

The unlock command is used to prevent specific engine features from being locked when exiting the ROM.

The format of the unlock command (in a big-endian byte array) is shown in this table:

Table 8-34. Unlock command format

Tag	Length	Eng
Value		
Value		
...		
Value		

where:

```
Tag: a single-byte field set to 0xB2
Eng: the engine to be left unlocked
Values: [optional] the unlock values required by the engine
```

NOTE

This command may not be used in the DCD structure if the SEC_CONFIG is configured as closed.

8.7 Plugin image

The ROM supports a limited number of boot devices. When using other devices as a boot source (for example, Ethernet, CDROM, or USB), the supported boot device must be used (typically serial ROM) as a firmware to provide the missing boot drivers. Additionally, the plugin can be customized to support boot drivers, which is more flexible when performing the device initialization, such as condition judging, delay assertion, or to apply custom settings to the boot device and memory system.

In addition to the standard images, the chip also supports plugin images. The plugin images return the execution to the ROM whereas the standard image does not.

The boot ROM detects the image type using the plugin flag of the boot data structure (see Boot data structure). If the plugin flag is 1, then the ROM uses the image as a plugin function. The function must initialize the boot device and copy the program image to the final location. At the end, the plugin function must return with the program image parameters. (See High-level boot sequence for details about the boot flow).

The boot ROM authenticates the plugin image before running the plugin function and then authenticates the program image.

The plugin function must follow the API described below:

typedef BOOLEAN (*plugin_download_f)(void **start, size_t *bytes, UINT32 *ivt_offset);

ARGUMENTS PASSED:

- start - the image load address on exit.
- bytes - the image size on exit.
- ivt_offset - the offset (in bytes) of the IVT from the image start address on exit.

RETURN VALUE:

- 1 - success
- 0 - failure

8.8 Serial Downloader

The Serial Downloader provides a means to download a program image to the chip over the USB serial connection.

In this mode, the ROM programs the WDOG1 for a 90-second time-out if the WDOG_ENABLE eFuse is 1 and continuously polls for the USB connection. If no activity is found on the USB OTG1 and the watchdog timer expires, the Arm core is reset.

NOTE

After the downloaded image is loaded, it is responsible for managing the watchdog resets properly.

This figure shows the USB boot flow:

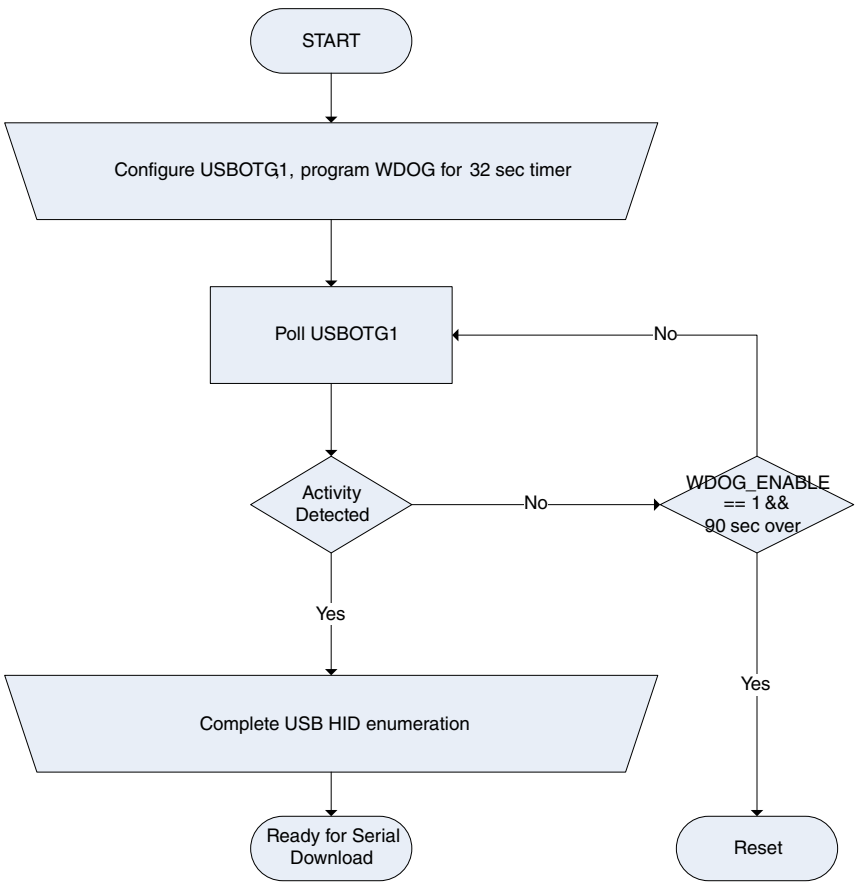


Figure 8-24. Serial Downloader boot flow

8.8.1 USB

The USB support is composed of the USBOH3 (USB OTG1 core controller, compliant with the USB 2.0 specification) and the USBPHY (HS USB transceiver).

The ROM supports the USB OTG port for boot purposes. The other USB ports on the chip are not supported for boot purposes.

The USB Driver is implemented as a USB HID class. A collection of four HID reports are used to implement the SDP protocol for data transfers, as described in Table 8-35.

Table 8-35. USB HID reports

Report ID (first byte)	Transfer endpoint	Direction	Length	Description
1	control OUT	Host to device	17 B	SDP command from the host to the device.
2	control OUT	Host to device	Up to 1025 B	Data associated with the report 1 SDP command.
3	interrupt	Device to host	5 B	HAB security configuration. The device sends 0x12343412 in the closed mode and 0x56787856 in the open mode.
4	interrupt	Device to host	Up to 65 B	Data in response to the SDP command in report 1.

8.8.1.1 USB configuration details

The USB OTG function device driver supports a high speed (HS for UTMI) non-stream mode with a maximal packet size of 512 B and a low-level USB OTG function.

The VID/PID and strings for the USB device driver are listed in the following table.

Table 8-36. VID/PID and strings for USB device driver

Descriptor	Value
VID (NXP vendor ID)	0x15A2
PID[1]	0x61
String Descriptor1 (manufacturer)	NXP Semiconductors
String Descriptor2 (product)	S Blank RIGEL SE Blank RIGEL NS Blank RIGEL

Table continues on the next page...

Table 8-36. VID/PID and strings for USB device driver (continued)

Descriptor	Value
String Descriptor4	NXP Flash
String Descriptor5	NXP Flash

1. Allocation based on the BPN (Before Part Number)

8.8.1.2 IOMUX configuration for USB

The interface signals of the UTMI PHY are not configured in the IOMUX. The UTMI PHY interface uses the dedicated contacts on the IC. See the chip data sheet for details.

8.8.2 Serial Download Protocol (SDP)

The 16-byte SDP command from the host to device is sent using the HID report 1.

This table describes the 16-byte SDP command data structure:

Table 8-37. 16-byte SDP command data structure

BYTE offset	Size	Name	Description
0	2	COMMAND TYPE	These commands are supported for the ROM: • 0x0101 READ_REGISTER • 0x0202 WRITE_REGISTER • 0x0404 WRITE_FILE • 0x0505 ERROR_STATUS • 0x0A0A DCD_WRITE • 0x0B0B JUMP_ADDRESS
2	4	ADDRESS	Only relevant for these commands: READ_REGISTER, WRITE_REGISTER, WRITE_FILE, DCD_WRITE, and JUMP_ADDRESS. For the READ_REGISTER and WRITE_REGISTER commands, this field is the address to a register. For the WRITE_FILE and JUMP_ADDRESS commands, this field is an address to the internal or external memory address.
6	1	FORMAT	Format of access, 0x8 for an 8-bit access, 0x10 for a 16-bit access, and 0x20 for a 32-bit access. Only relevant for the READ_REGISTER and WRITE_REGISTER commands.

Table continues on the next page...

Table 8-37. 16-byte SDP command data structure (continued)

BYTE offset	Size	Name	Description
7	4	DATA COUNT	Size of the data to read or write. Only relevant for the WRITE_FILE, READ_REGISTER, WRITE_REGISTER, and DCD_WRITE commands. For the WRITE_FILE and DCD_WRITE commands, the DATA COUNT is in the byte units.
11	4	DATA	The value to write. Only relevant for the WRITE_REGISTER command.
15	1	RESERVED	Reserved

8.8.2.1 SDP commands

The SDP commands are described in the following sections.

8.8.2.1.1 READ_REGISTER

The transaction for the READ_REGISTER command consists of these reports: Report1 for the command, Report3 for the security configuration, and Report4 for the response or the register value.

The register to read is specified in the ADDRESS field of the SDP command. The first device sends Report3 with the security configuration followed by the Report4 with the bytes read at a given address. If the count is greater than 64, multiple reports with the report id 4 are sent until the entire data requested by the host is sent. The STATUS is either 0x12343412 for the closed parts and 0x56787856 for the open or field return parts.

Report1, Command, Host to Device:

1	Valid values for the READ_REGISTER COMMAND, ADDRESS, FORMAT, DATA_COUNT

ID 16-byte SDP command

Report3, Response, Device to Host:

3	Four bytes indicating the security configuration

ID 4 bytes status

Report4, Response, Device to Host: first response report

4	Register value

ID 4 bytes of data containing the register value. If the number of bytes requested is less than 4, the remaining bytes must be ignored by the host.

Multiple reports of the report id 4 are sent until the entire requested data is sent.

Report4, Response, Device to Host: last response report

4	Register value

ID 64 bytes of data containing the register value. If the number of bytes requested is less than 64, the remaining bytes must be ignored by the host.

8.8.2.1.2 WRITE_REGISTER

The transaction for the WRITE_REGISTER command consists of these reports: Report1 for the command, Report3 for the security configuration and Report4 for the write status.

The host sends Report1 with the WRITE_REGISTER command. The register to write is specified in the ADDRESS field of the SDP command of Report1, with the FORMAT field set to the data type (number of bits to write, either 8, 16, or 32) and the value to write in the DATA field of the SDP command. The device writes the DATA to the register address and returns the WRITE_COMPLETE code using Report4 and the security configuration using Report3 to complete the transaction.

Report1, Command, Host to Device:

1	Valid values for WRITE_REGISTER COMMAND, ADDRESS, FORMAT, DATA_COUNT and DATA

ID 16-byte SDP command

Report3, Response, Device to Host:

3	4 bytes indicating the security configuration

ID 4 bytes status

Report4, Response, Device to Host:

4	WRITE_COMPLETE (0x128A8A12) status

ID 64 bytes data with the first 4 bytes to indicate that the write is completed with code 0x128A8A12. On failure, the device reports the HAB error status.

8.8.2.1.3 WRITE_FILE

The transaction for the WRITE_FILE command consists of these reports: Report1 for the command phase, Report2 for the data phase, Report3 for the HAB mode, and Report4 to indicate that the data are received in full.

The size of each Report2 is limited to 1024 bytes (limitation of the USB HID protocol). Hence, multiple Report2 packets are sent by the host in the data phase until the entire data is transferred to the device. When the entire data (DATA_COUNT bytes) is received, the device sends Report3 with the HAB mode and Report4 with 0x88888888, indicating that the file download completed.

Report1, Host to Device:

1	Valid values for WRITE_FILE COMMAND, ADDRESS, DATA_COUNT

ID 16-byte SDP command

==========Optional Begin==========

Host sends the ERROR_STATUS command to query if the HAB rejected the address

========== Optional End==========

Report2, Host to Device:

2	File data

ID Max 1024 bytes data per report

Report2, Host to Device:

2	File data

ID Max 1024 bytes data per report

Report3, Device to Host:

3	4 bytes indicating security configuration

ID 4 bytes status

Report4, Response, Device to Host:

4	COMPLETE (0x88888888) status

ID 64 bytes data with the first four bytes to indicate that the file download completed with code 0x88888888. On failure, the device reports the HAB error status.

8.8.2.1.4 ERROR_STATUS

The transaction for the SDP command ERROR_STATUS consists of three reports.

Report1 is used by the host to send the command; the device sends global error status in four bytes of Report4 after returning the security configuration in Report3. When the device receives the ERROR_STATUS command, it returns the global error status that is updated for each command. This command is useful to find out whether the last command resulted in a device error or succeeded.

Report1, Command, Host to Device:

1	ERROR_STATUS COMMAND

ID 16-byte SDP Command

Report3, Response, Device to Host:

3	Four bytes indicating the security configuration

ID 4 bytes status

Report4, Response, Device to Host:

4	Four bytes Error status

ID first 4 bytes status in 64 bytes Report4

8.8.2.1.5 DCD_WRITE

The SDP command DCD_WRITE is used by the host to send multiple register writes in one shot. This command is provided to speed up the process of programming the register writes (such as to configure an external RAM device).

The command goes with Report1 from the host with COMMAND TYPE set to DCD_WRITE, ADDRESS which is used as a temporary location of the DCD data, and DATA_COUNT to the number of bytes sent in the data out phase. In the data phase, the

host sends the data for a number of registers using Report2. The device completes the transaction with Report3 indicating the security configuration and Report4 with the WRITE_COMPLETE code 0x12828212.

Report1, Command, Host to Device:

1	DCD_WRITE COMMAND, ADDRESS, DATA_COUNT

ID 16-byte SDP Command

Report2, Data, Host to Device:

2	DCD binary data

ID Max 1024 bytes per report

Report3, Response, Device to Host:

3	Four bytes indicating the security configuration

ID 4 bytes status

Report4, Response, Device to Host:

4	WRITE_COMPLETE (0x128A8A12) status

ID 64 bytes report with the first four bytes to indicate that the write completed with the code 0x128A8A12. On failure, the device reports the HAB error status.

See Device Configuration Data (DCD) for the DCD format description.

8.8.2.1.6 JUMP_ADDRESS

The SDP command JUMP_ADDRESS is the last command that the host can send to the device. After this command, the device jumps to the address specified in the ADDRESS field of the SDP command and starts to execute.

This command usually follows after the WRITE_FILE command. The command is sent by the host in the command-phase of the transaction using Report1. There is no data phase for this command, but the device sends the status Report3 to complete the transaction. If the authentication fails, it also sends Report4 with the HAB error status.

Report1, Command, Host to Device:

1	JUMP_ADDRESS COMMAND, ADDRESS

ID 16-byte SDP Command

Report3, Response, Device to Host:

3	Four bytes indicating the security configuration

ID 4 bytes status

This report is sent by the device only in case of an error jumping to the given address, or if the device reports error in Report4, Response, Device to Host:

4	Four bytes HAB error status

ID 4 bytes status, 64 bytes report length

8.9 Recovery devices

The chip supports recovery devices. If the primary boot device fails, the boot ROM tries to boot from the recovery device using one of the I^2C or ECSPI ports.

To enable the recovery device, the BOOT_CFG4[6] fuse must be set. Additionally, the serial EEPROM fuses must be set as described in Serial ROM through SPI and I^2C.

8.10 USB low-power boot

The ROM supports the USB low-power boot. This feature enables a device with a dead or weak battery to power up and boot if the device is connected to a USB upstream port, no matter if the upstream port is a USB charger or a USB host/hub.

If a USB dedicated charger or a host/hub charger are connected, as soon as the device is connected to the upstream port, a stable current (max.1.5 A) can be supplied by the charger. If a USB host/hub is connected, a maximal current of 100 mA is supplied to the device and the device is able to power up to boot the image with less than 100 mA.

If the LPB_BOOT fuses are blown, the chip checks if there is a low-power condition via the GPIO_3 pad. If there is a low-power boot condition, the USB charger detection is activated. If there is no USB charger, the ROM initializes the USB as a device and applies division factors on the Arm, DDR, AXI, and AHB root clocks based on the

LPB_BOOT fuses value (see the table below). The polarity of the low-power boot condition on the GPIO_3 pad is set by the BT _LPB_POLARITY fuse (see the following figure).

Table 8-38. USB low-power boot frequencies

LPB_BOOT	Boot Frequencies=0	Boot Frequencies=1
00	ARM_CLK_ROOT=792 MHz MMDC_CH0_AXI_CLK_ROOT=528 MHz MMDC_CH1_AXI_CLK_ROOT=528 MHz AXI_CLK_ROOT=264 MHz AHB_CLK_ROOT=132 MHz	ARM_CLK_ROOT=396 MHz MMDC_CH0_AXI_CLK_ROOT=352 MHz MMDC_CH1_AXI_CLK_ROOT=352 MHz AXI_CLK_ROOT=176 MHz AHB_CLK_ROOT=88 MHz
01	ARM_CLK_ROOT=792 MHz MMDC_CH0_AXI_CLK_ROOT=528 MHz MMDC_CH1_AXI_CLK_ROOT=528 MHz AXI_CLK_ROOT=264 MHz AHB_CLK_ROOT=132 MHz	ARM_CLK_ROOT=396 MHz MMDC_CH0_AXI_CLK_ROOT=352 MHz MMDC_CH1_AXI_CLK_ROOT=352 MHz AXI_CLK_ROOT=176 MHz AHB_CLK_ROOT=88 MHz
10	ARM_CLK_ROOT=396 MHz MMDC_CH0_AXI_CLK_ROOT=264 MHz MMDC_CH1_AXI_CLK_ROOT=264 MHz AXI_CLK_ROOT=132 MHz AHB_CLK_ROOT=66 MHz	ARM_CLK_ROOT=264 MHz MMDC_CH0_AXI_CLK_ROOT=176 MHz MMDC_CH1_AXI_CLK_ROOT=176 MHz AXI_CLK_ROOT=88 MHz AHB_CLK_ROOT=44 MHz
11	ARM_CLK_ROOT=264 MHz MMDC_CH0_AXI_CLK_ROOT=132 MHz MMDC_CH1_AXI_CLK_ROOT=132 MHz AXI_CLK_ROOT=66 MHz AHB_CLK_ROOT=66 MHz	ARM_CLK_ROOT=132 MHz MMDC_CH0_AXI_CLK_ROOT=88 MHz MMDC_CH1_AXI_CLK_ROOT=88 MHz AXI_CLK_ROOT=44 MHz AHB_CLK_ROOT=44 MHz

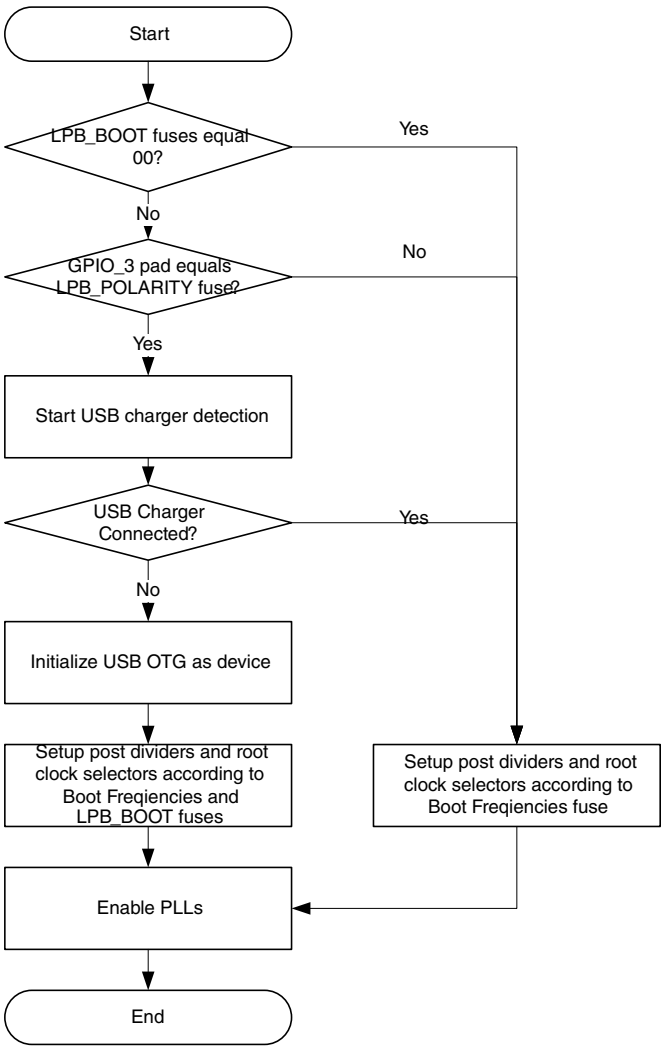


Figure 8-25. USB low-power boot flow

8.11 SD/MMC manufacture mode

When the internal boot and recover boot (if enabled) failed, the boot goes to the SD/MMC manufacture mode before the serial download mode. In the manufacture mode, one bit bus width is used despite of the fuse setting.

In the manufacture mode, the SD or MMC card will be scanned on the uSDHC1 and uSDHC2. If a card is detected and a valid boot image is found in the card, the boot image is loaded and executed. Pad of SD1_CD and SD2_CD are used to detect whether a card is inserted or not.

By default, the SD/MMC manufacture mode is enabled. Blow the fuse of the DISABLE_SDMMC_MFG to disable it.

NOTE

A secondary boot is not supported in the SD/MMC manufacture mode.

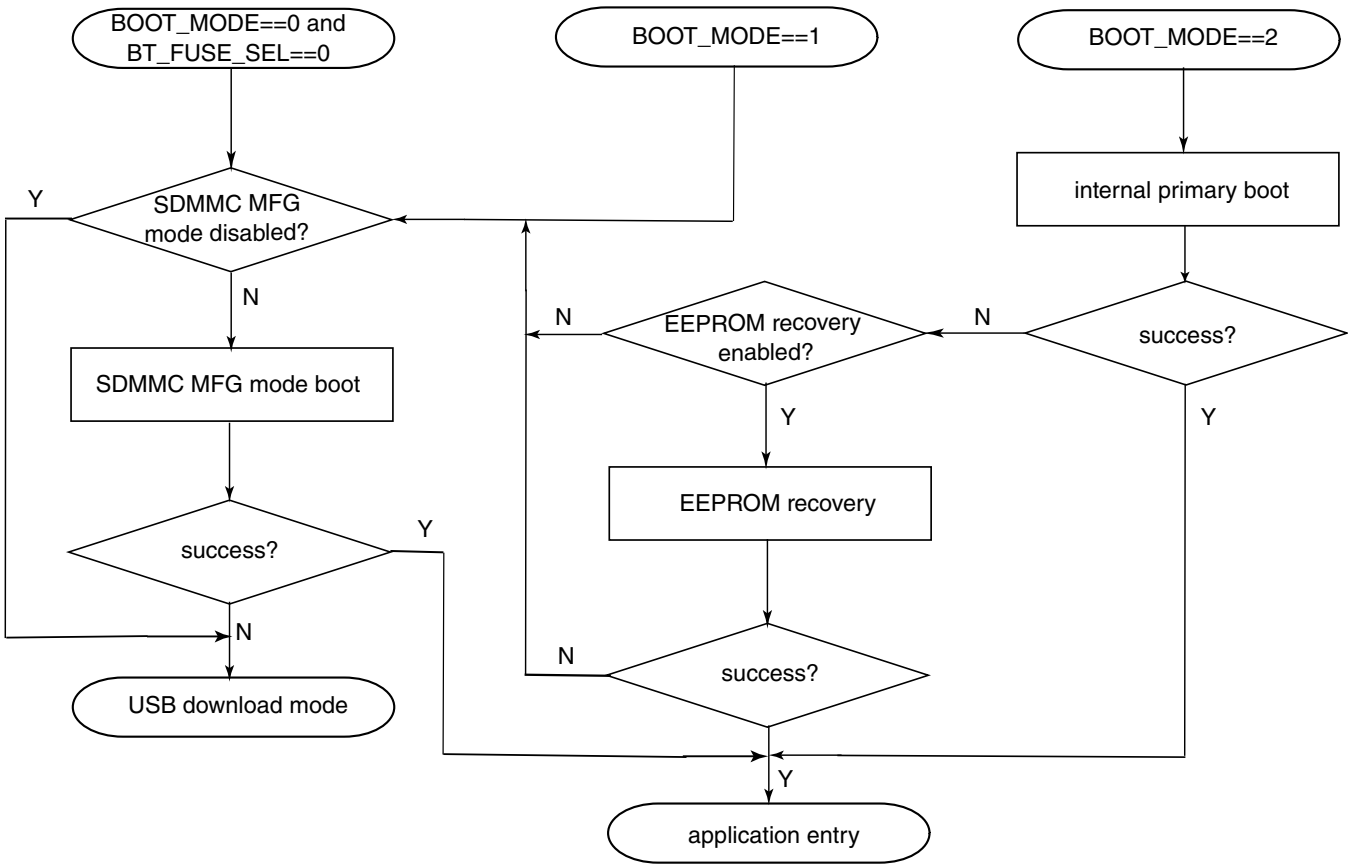


Figure 8-26. SD/MMC manufacture boot flow

8.11.1 Using manufacture mode / serial download mode with eMMC

Manufacture mode is intended to allow a system to boot from a SD/MMC card on a board with unprogrammed boot media or to upgrade the image on a boot device. For manufacture mode, the boot ROM assumes if there is a SD/MMC card present indicated by the uSDHC card detect (CD) signal pulled low, then there is a valid image on the card.

If an unprogrammed eMMC device is connected to the uSDHC port(s) on which manufacture mode is supported and the CD signal is low, the ROM will attempt manufacturing mode with the eMMC device, which does not contain a valid image. The ROM loads the invalid image, and this will cause the ROM code to fail to enter serial

download mode, resulting in a reset of the system by the ROM. To enter serial download mode in this case, the CD pin should be pulled up so the ROM does not detect a eMMC device present and will bypass the manufacture mode to enter serial download mode.

8.12 High-Assurance Boot (HAB)

The High Assurance Boot (HAB) component of the ROM protects against the potential threat of attackers modifying the areas of code or data in the programmable memory to make it behave in an incorrect manner. The HAB also prevents the attempts to gain access to features which must not be available.

The integration of the HAB feature with the ROM code ensures that the chip does not enter an operational state if the existing hardware security blocks detected a condition that may be a security threat or if the areas of memory deemed to be important were modified. The HAB uses the RSA digital signatures to enforce these policies.

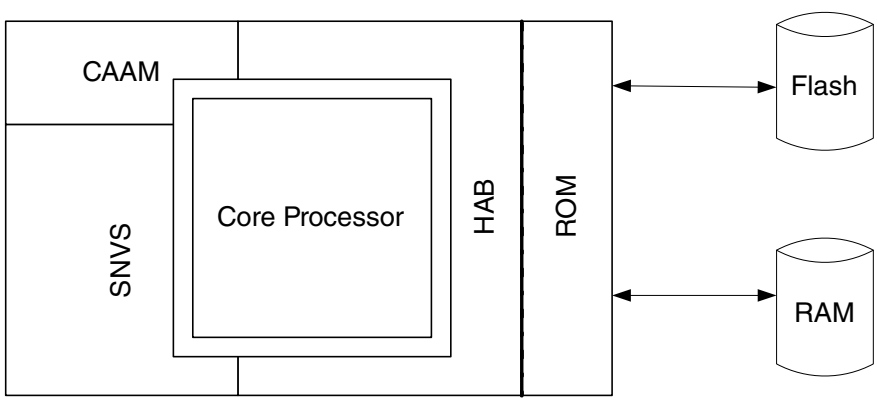


Figure 8-27. Secure boot components

The figure above illustrates the components used during a secure boot using HAB. The HAB interfaces with the SNVS to make sure that the system security state is as expected. The HAB also uses the CAAM hardware block to accelerate the SHA-256 message digest operations performed during the signature verifications and AES-128 operations for the encrypted boot operations. The HAB also includes a software implementation of SHA-256 for cases where a hardware accelerator can't be used. The RSA key sizes supported are 1024, 2048, and 3072 bits. The RSA signature verification operations are performed by a software implementation contained in the HAB library. The main features supported by the HAB are:

- X.509 public key certificate support
- CMS signature format support
- Proprietary encrypted boot support. Note that the encrypted boot depends on the CAAM hardware module. When the CAAM is disabled (when the EXPORT_CONTROL fuse is blown), the encrypted boot is not available.

NOTE

NXP provides the reference Code Signing Tool (CST) for key generation, certificate generation, and code signing for use with the HAB library. The CST can be found by searching for "IMX_CST_TOOL" at http://www.nxp.com.

NOTE

For further details on using the secure boot feature using HAB, refer to *Secure Boot on i.MX 50, i.MX 53, i.MX 6 and i.MX 7 Series using HABv4 (AN4581)*.

8.12.1 HAB API vector table addresses

For devices that perform a secure boot, the HAB library may be called by the boot stages that execute after the ROM code.

The RVT table contains the pointers to the HAB API functions. The address of the RVT table is shown in the following table.

Table 8-39. RVT table addresses

	HAB API vector table address	ROM API vector table address
Silicon revisions prior to 1.2	0x00000094	0x000000C0
Silicon revision 1.2 and later	0x00000098	0x000000C4

NOTE

For additional information on the secure boot including the HAB API, refer to *HABv4 RVT Guidelines and Recommendations (AN12263)*.

Chapter 9
Multimedia

9.1 Video Graphics Sub System

The chip video graphics subsystem consists of the dedicated modules found here.

- Video Processing Unit (VPU): a multi-standard high performance video/image CODEC
- Two Graphics Processing Units (GPUs):
  - 3D GPU: accelerating the generation of 3D graphics (OpenGL/ES)
  - 2D GPU: acceleration the generation of 2D graphics (BitBLT).
- One Image Processing Unit (IPU): providing connectivity to cameras and displays, related processing, synchronization and control.
- Display interface bridges: providing optional translation from the digital display interface supported by the IPU to other interfaces:
  - LVDS bridge (LDB): providing up to one LVDS interface
  - HDMI transmitter
  - MIPI/DSI transmitter
- MIPI/CSI-2 receiver
- Two (identical) Display Content Integrity Checker (DCIC) are used to authenticate sensitive displayed data.
- A Video Data Order Adapter (VDOA): used to re-order video data from the "tiled" order used by the VPU to the conventional raster-scan order needed by the IPU.

High level integration scheme of the i.MX 6Solo/6DualLite video/graphics system is provided by the figure below.

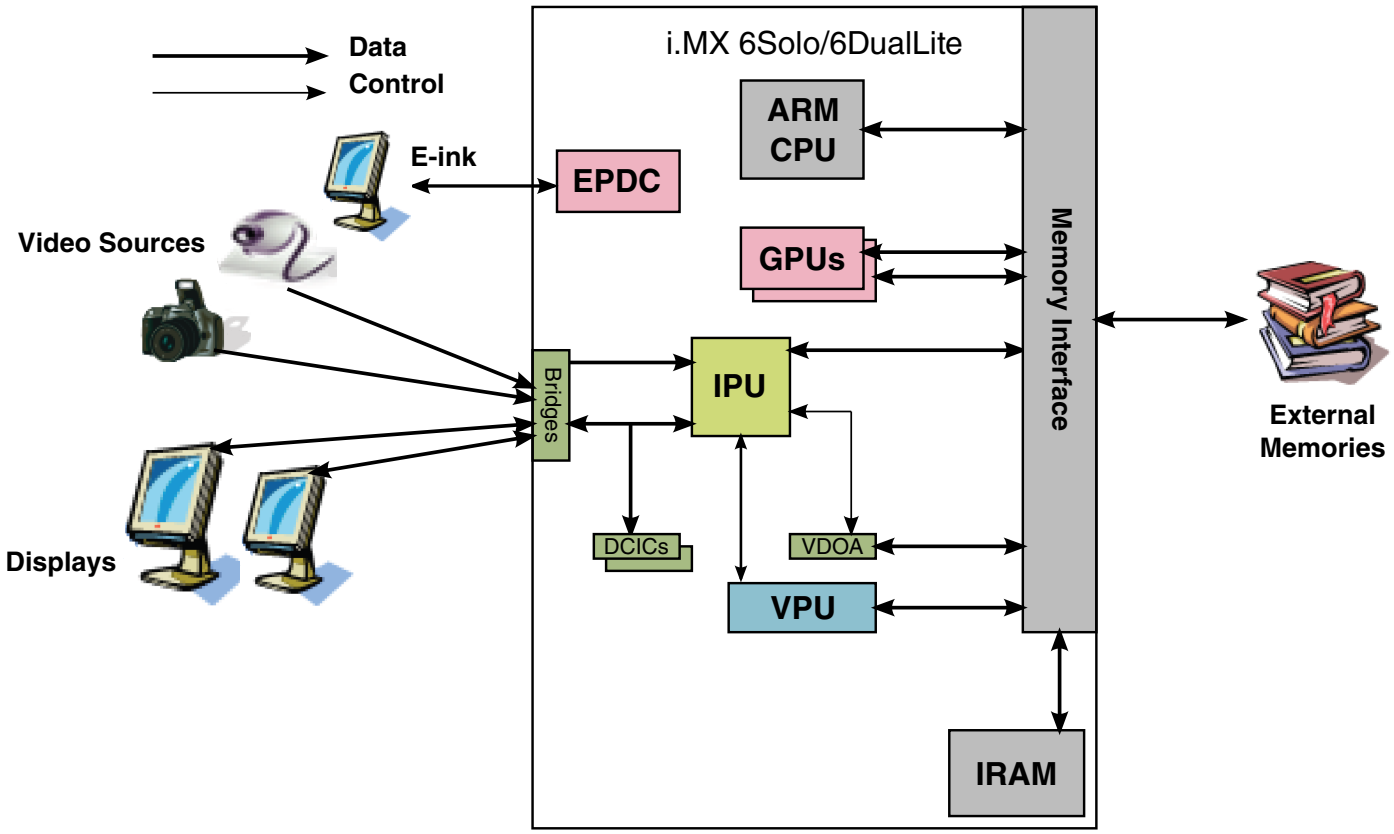


Figure 9-1. i.MX 6Solo/6DualLite Video-Graphics Subsystem

9.1.1 Display outputs

The i.MX 6Solo/6DualLite implements a robust muxing logic on the two display ports (2x per IPU), to the external interfaces, either direct, or via bridges (MIPI, LVDS, HDMI), per description below:

- Two parallel - driven directly by IPU; pixel clock at least up to 200 MHz (for external load of up to 10 pF).
  - Parallel interface works up to 200 MHz
  - HDMI interface works up to 270 MHz
- Two LVDS channels, driven by the LDB; pixel clock up to 170 MHz.
- One HDMI port (ver. 1.4) - driven by the HDMI transmitter: pixel clock up to 270 MHz (gated by the IPU capabilities)
- One MIPI/DSI port - driven by the MIPI/DSI transmitter; 2 data lanes at 1 GHz
- IPU display port (DI) can be connected to each of the above ports

- IPU has 2 display ports, up to two external ports can be active at any given time. (Additional asynchronous data flows can be sent though the parallel ports and the MIPI/DSI port.)
- Read access is supported as follows
  - For the Parallel0 port: through DI0
  - For the Parallel1 port: through DI1

- Inputs to either of the DCICs are taken from one of the following buses
  - For each of the parallel interfaces: probing the I/O loopback (essentially equivalent to probing the external wires).
  - For other integrated interfaces (e.g. LVDS): probing the DI1 output of the IPU (essentially equivalent to the inputs to the serializers)
  - For the data enable signal, two control signals are probed from each of the above buses.

The i.MX 6Solo/6DualLite also implements a EPDC for E-ink display.

For visual view of display signal routing, see the following figure. The chip MUX select signals are driven by configuration bits as specified in the IOMUX controller (IOMUXC) chapter in the IOMUXC_IOMUXC_GPR registers description fields.

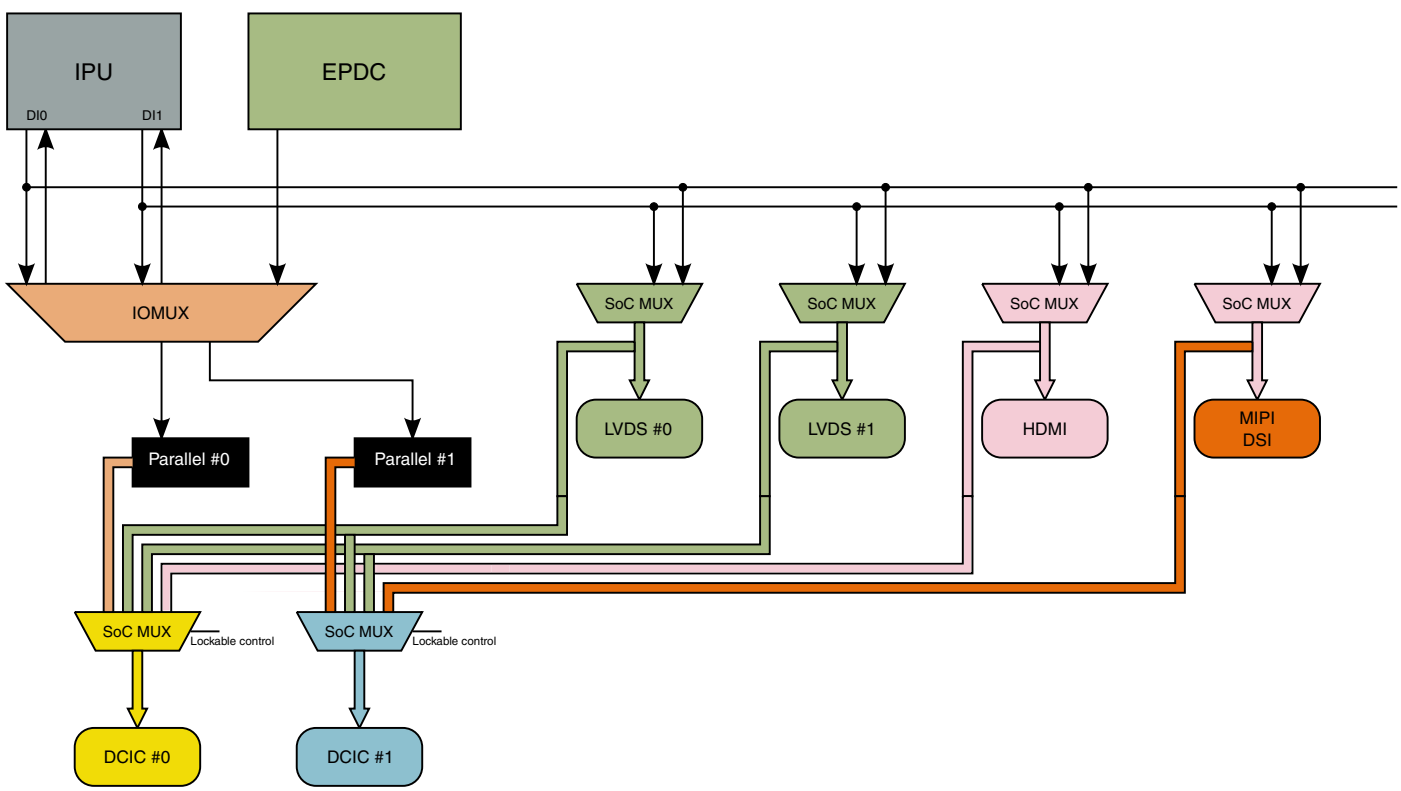


Figure 9-2. Display port muxing scheme

9.1.2 Video input

There are 2 video input ports (e.g. from image sensors)

See the figure below for details.

Video Input Ports Connectivity

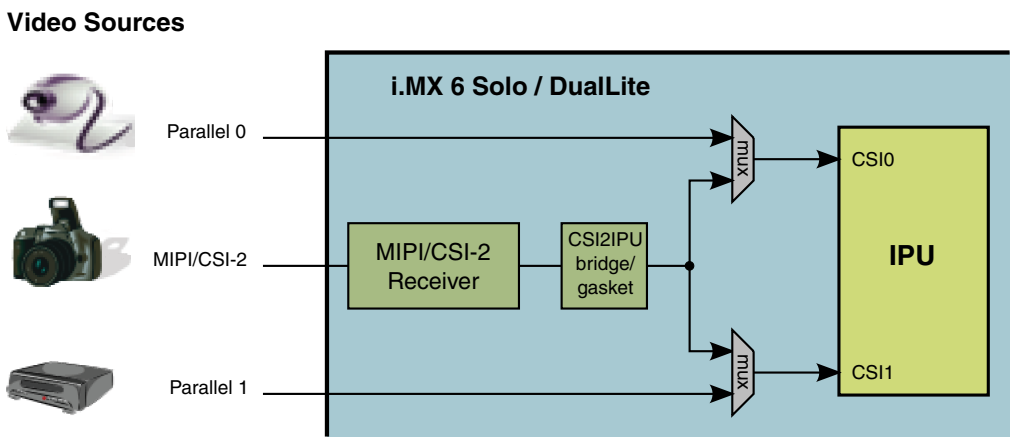


Figure 9-3. The Video Input Ports

One parallel port is connected directly to an IPU input port.

One MIPI/CSI-2 port- IPU receives two components per cycle from the MIPI_CSI2 interface. The maximum bandwidth of the interface is as follows:

- 250MByte/sec for 2 data lanes configuration (1000Mbps/lane)
- 125Mbyte/sec for 1 data lanes configuration (1000Mbps/lane)

IPU has two input ports, CSI0 and CSI1, which can receive data concurrently and independently. At any given time, an IPU input port may receive data either from a parallel external port or from the MIPI/CSI-2 receiver.

The MIPI/CSI-2 port can receive up to 4 concurrent data channels. Each data channel can be routed by one of the two CSI ports of the IPU. Pixel data can be further processed by the IPU. Other data types can be transferred through a CSI transparently as generic data to the system memory.

The IPU, VPU, VDOA and the GPUs have master AXI ports, providing access to system memory.

The modules are controlled (by the ARM CPU or the SDMA) as follows:

- The LDB is controlled by signals connected to top-level registers.
- All other modules have a host interface. For the VPU, VDOA and the DCIC's, this is a slave IP port and for the MIPI bridges, GPUs, and IPUs, this is a slave AHB port.
- The data flow between VPU, VDOA, GPUs and IPU is through system memory and it is normally controlled by the CPU. For spacial situations, a direct synchronization interface is provided:
  - An interface between the IPU and VPU, for low-latency video record.
  - An interface between the VDOA and IPU, for tight pipelining of data from the VDOA to the IPU, through the IRAM.

9.1.3 Synchronization Mechanisms

The i.MX 6Solo/6DualLite provides HW synchronization mechanism between IPU and VPU and between IPU and VDOA to reduce core intervention and enable lock-step operation.

9.1.4 Supported applications

The system described above supports a wide variety of video/graphics applications. The following table describes the main applications.

Table 9-1. Video/graphics applications

Application	Features
Display management	
Screen refresh	Up to two displays active simultaneously Pixel clock rate: • Single display - up to 270 MHz • Two displays (the sum of the two rates) - up to 270 MHz **NOTE:** The parallel interface works up to 200 MHz, and the HDMI interface works up to 270 MHz. Special features: • On-the-fly image conversion: rotation, inversion, resizing, color-space conversion and combining • On-the-fly image quality enhancement: color adjustment (including special effects) and gamut mapping, gamma correction and contrast stretching • Low-light compensation, allowing back-light reduction • Scrolling/panning
Updating the display buffer (from a background buffer)	Buffer either in system memory or in an external display controller (accessed through the display port) Special features:

Table continues on the next page...

Table 9-1. Video/graphics applications (continued)

Application	Features
	• On-the-fly processing - as for screen refresh • Optimized update - only modified parts are transferred • Synchronization with screen refresh, to prevent tearing • Scrolling (e.g. a running banner or a short animation)
Video	
Camera preview (displaying a view-finder window)	Input rate (from sensor): up to 270 MHz Additional features: • Window-of-interest • On-the-fly image conversion: de-interlacing, resizing, color space conversion, rotation, inversion • Combining with graphics • Synchronization, to prevent tearing
Still image capture	Rate: • Burst mode (off-line processing): up to 180 Mpixels/sec (while still leaving headroom for up to 35% blanking overhead) • Continuous: up to 160 Mpixels/sec Additional features: • Image quality enlacement before compression - as for camera preview • Image conversion before compression - as for camera preview; with independent parameters • Synchronization with a flash, a mechanical shutter and a mechanical iris
Motion Video Record	Resolution and rate: up to 1080p (1920x1080 at 60 fps in unit test without multimedia framework) Image processing before compression - as for still image capture
Motion Video Playback	Resolution and rate - up to 1080i/p (1920x1080) at 60 fps in unit test (without multimedia framework) Additional features: • Post-filtering: de-blocking and de-ringing • Image conversion after decompression: de-interlacing, rotation, inversion, resizing, color-space conversion and combining • Quality enhancement: color adjustment (including special effects) and gamut mapping, gamma correction and contrast stretching
Two-Way Video	Resolution and rate - up to 1080p (1920x1080) at 30 fps in unit test (without multimedia framework) Record and playback features - as above Various display options, including: • Two non-overlapping windows • Picture-in-picture

The following figure provides the sample processing flow of multimedia application.

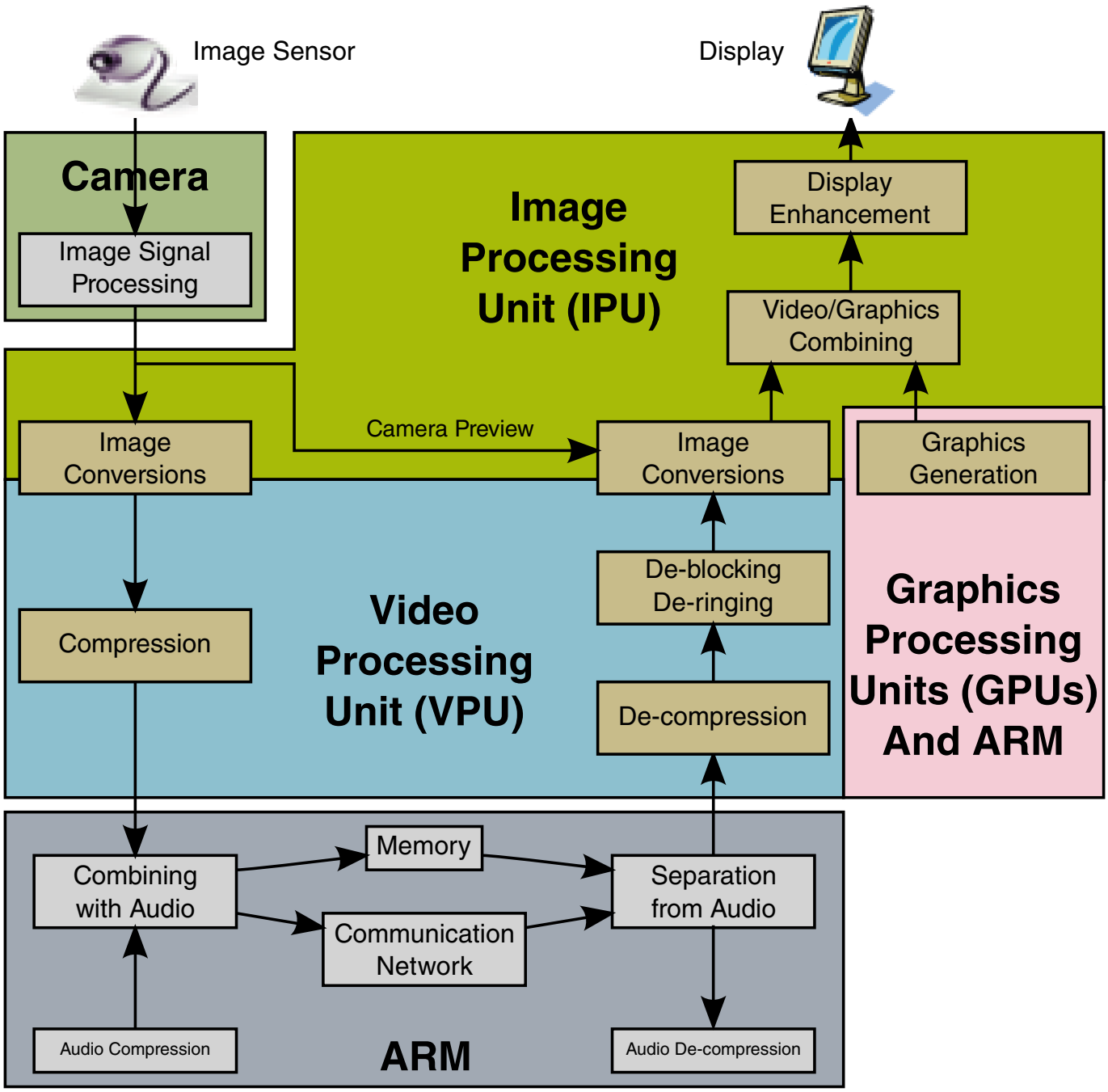


Figure 9-4. Processing flow for multimedia applications

- The camera preview mode-the upper part
- Video recording-the left part
- Video playback-the right part
- A video call-both sides

9.2 Image Processing Unit (IPU)

The following table shows the IPU IP parametric table.

Table 9-2. IPU IP Parametric Table

Name	IPU
Function	Connectivity to cameras and displays; related processing; synchronization and control
External I/O Pins Notes: This is the pinout of the IPU module At chip level, some of the pins are muxed and some are omitted. Additional GPIO pins are required to construct the connection. This is not included in this list.	Parallel Display port: 32 bit data, ~18 clocks and controls. Regular CMOS IO type, 198 MHz max. May be also connected to internal connectivity bridges. Slow Serial Display port: 2 bit data, 3 clocks and controls. Regular CMOS IO type, 120 MHz max. Parallel Sensor port: 20 bit data inputs, 5-6 clocks and controls. Regular CMOS IO type, MHz max. Camera strobe port: 6 camera control outputs. Regular CMOS IO type, 120 MHz max.
SoC Buses	AXI master - for accessing the memory AHB slave - for programming, control and direct access of the MCU to the display
Interrupts	Two interrupts: functional and error
DMA Requests	Includes an integral DMA controller with an AXI master port Also one DMA request to the SDMA
Number of instantiations	1
Clock sources and range	HSP_CLK - Internal high-speed processing clock: up to 270 MHz DI_CLK0, DI_CLK1 - Display interface clocks: up to 270 MHz

The goal of the IPU is to provide comprehensive support for the flow of data from an image sensor and/or to a display device. This support covers all aspects of these activities:

- Connectivity to relevant devices - cameras, displays, graphics coprocessors, TV encoders and decoders.

- Related image processing and manipulation: image enhancements and conversions, etc.
- Synchronization and control capabilities (to avoid tearing artifacts, for example)

This integrative approach leads to several significant advantages:

- Automation: The involvement of the MCU (Main Control Unit) in image management is minimized. In particular, display refresh/update and a camera preview (displaying the input from an image sensor) can be performed completely autonomously. The resulting benefits are reducing the overhead due to SW-HW synchronization, freeing the MCU to perform other tasks and reduced power consumption (when the MCU is idle and can be powered down).
- Optimal data path: Access to system memory is minimized. In particular, significant processing can be performed on-the-fly while receiving data from an image sensor and/or sending data to a display. System memory is used essentially only when a change in pixel order or frame rate is needed. The resulting benefits are reduced load on the system bus and further reduction of power consumption.
- Resource sharing: Maximal HW reuse for different applications, resulting with the support of a wide range of requirements with minimal hardware

The hardware reuse mentioned above is enabled by a sophisticated configurability of each hardware block. This configurability also allows the support of a wide range of external devices, data formats and operation modes.

A simplified block diagram of the IPU is given in the figure below. The role of each block is described in the table below.

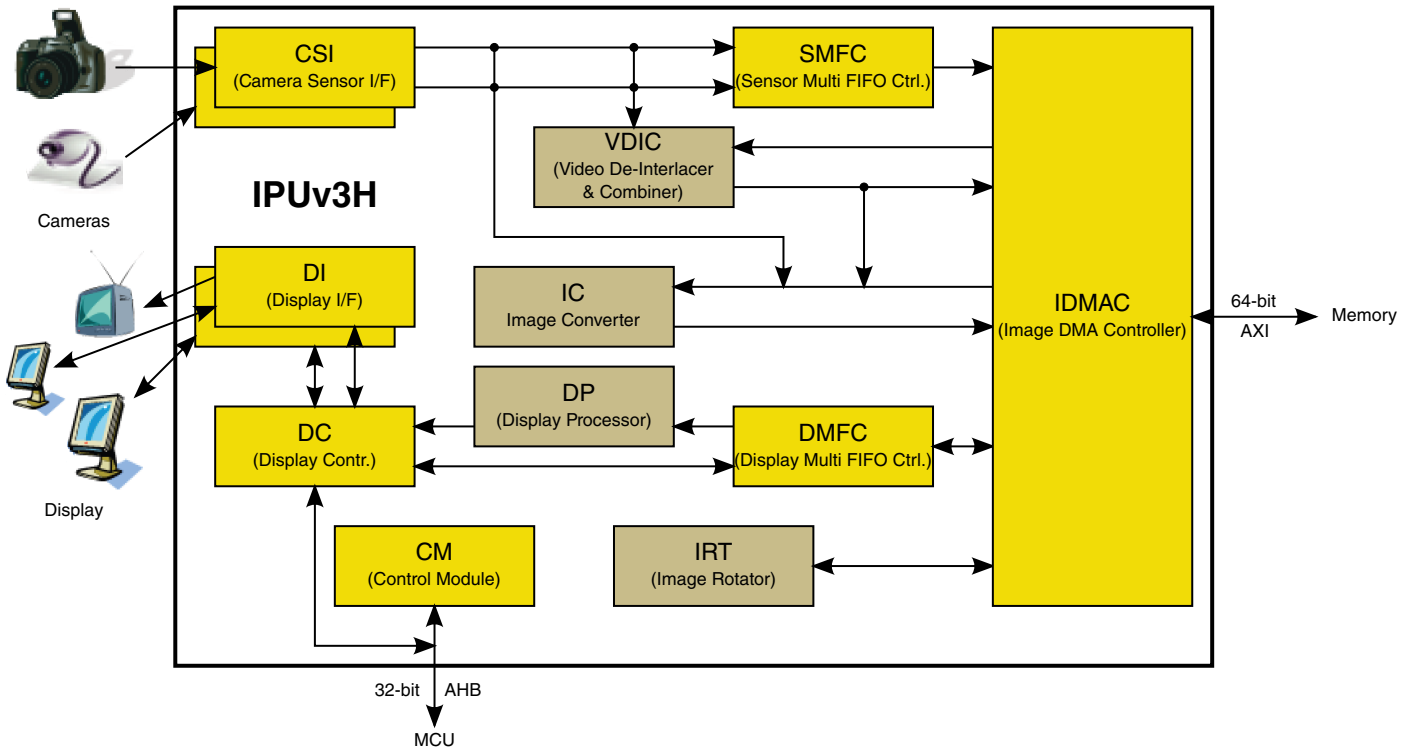


Figure 9-5. IPU - Block Diagram

Table 9-3. IPU - Block Description

Block	Description
CSI - Camera Sensor Interface	Controls a camera port; provides interface to an image sensor or a related device. IPUv3 includes 2 such blocks
DI - Display Interface	Provides interface to displays, display controllers and related devices. IPUv3 includes 2 such blocks
DC - Display Controller	Controls the display ports.
VDIC - Video De-Interlacer and Combiner	Performs de-interlacing - converting interlaced video to progressive - or combining
IC - Image Converter	Performs resizing, color conversion/correction, combining with graphics, and horizontal inversion
DP - Display Processor	Performs the processing required for data sent to display
IRT - Image Rotator	Performs rotation (90 or 180 degrees) and inversion (vertical/horizontal)
IDMAC - Image DMA Controller	Controls the memory port; transfers data to/from system memory
SMFC - Sensor Multi FIFO Controller	Controls FIFO's for output from the CSI's to system memory
DMFC - Display Multi FIFO Controller	Controls FIFO's for IDMAC channels related to the display system
CM - Control Module	Provides control and synchronization.

9.2.1 IPU External Ports

The IPU has the following ports:

- Two camera ports - controlled by a CSI module - providing a connection to image sensors and related devices.
- Two display ports - each controlled by a DI module - providing a connection to displays and related devices.
- Memory port - AXI (AHB V3.0) master, controlled by the IDMAC - providing connection to the system memory.
- AHB-lite slave port, providing connection to the ARM MCU (and to any other master connected to the ARM's cross-bar switch)

9.2.1.1 Camera Ports

The role of these ports is to receive input from video sources (e.g. image sensors) and to provide support for time-sensitive control signals to the camera. (Non-time-sensitive controls - e.g. configuration, reset - are performed by the MCU, through an I2C I/F or GPIO signals).

Each of the camera ports includes the following features:

- Direct connectivity to most relevant external devices.
- Parallel interface - up to 20-bit data bus
- Frame size: up to 8192 x 4096 pixels (including blanking intervals)
- Data formats supported include Raw (Bayer), RGB, YUV 4:4:4, YUV 4:2:2 and grayscale, up to 16 bits per value (component).
- Synchronization - video mode
  - The sensor is the master of the pixel clock (PIXCLK) and synchronization signals
  - Synchronization signals are received using either of the following methods:
    - Dedicated control signals -VSYNC, HSYNC - with programmable pulse width and polarity
    - Controls embedded in the data stream, following loosely the BT.656 protocol, with flexibility in code values and location.
- Synchronization - still image capture
  - The image capture is triggered by the MCU or by an external signal (e.g. a mechanical shutter)
  - Synchronized strobes are generated for up to 6 outputs - the sensor and camera peripherals (flash, mechanical shutter...)
- Additional features
  - Frame rate reduction, by periodic skipping of frames

- Downsizing x2, by skipping rows/columns
- Window-of-interest selection
- Pre-flash - for red-eye reduction and for measurements (e.g. focus) in low-light conditions

Several sensors can be connected to each of the CSI's. Simultaneous functionality (sending data) is supported as follows:

- Two streams can be received independently, each through a different CSI.
- Only one of the streams can be transferred to the VDIC or IC for on-the-fly processing, while the other one sent directly to system memory.

The input rate supported by the camera port is as follows:

- Peak: up to 240 MHz (values/sec)
- Average (assuming 35% blanking overhead), for YUV 4:2:2
  - Pixel in one cycle (e.g. BT.1120): up to 180 MP/sec, e.g. 12M pixels @ 15 fps
  - Pixel on two cycles (e.g. BT.656): up to 90 MP/sec, e.g. 6M pixels @ 15 fps.
- On-the-fly processing may be restricted to a lower input rate - see below.

9.2.1.2 Display Ports

The role of these ports is to communicate with display devices, either directly or through a controller (e.g. graphics accelerator) or a bridge (e.g. TV encoder or an LVDS interface bridge).

9.2.1.2.1 Access Modes

Two access modes are supported:

- Synchronous Access
- Asynchronous Access

9.2.1.2.2 Synchronous Access

In this mode, the IPU transfers a two-dimensional block of pixels to the display device, in synchronization with the screen refresh cycle.

This mode has a dual role:

- For a RAM-less display or a TV screen, this mode is used to perform the screen refresh process from a display buffer in system memory.
- For a "smart" display, this mode is used to transfer a rectangular block of pixels to the display's screen and, in some cases, also to the display buffer

In both cases, the IPU sends to the display all the synchronization signals controlling the screen refresh and the block transfer is synchronized with these signals. This synchronization means that tearing effects are avoided when using this mode.

9.2.1.2.3 Asynchronous Access

This is the main mode used for communicating with an external display controller (possibly in a smart display or a graphics accelerator). In this mode, the IPU performs random access - read/write - to the memory and registers of the controller.

The following access types are provided:

- Data transfer to the external device, after on-the-fly processing in the IPU.
- Data transfer (DMA) - read/write - between the host's system memory and the external device, through the IPU's memory port (controlled by the IDMAC); e.g. transfer of a rectangular block of pixels (possibly full screen).
- Host access - read/write - to an external device, through the AHB-slave port

9.2.1.2.4 Interface details

The display interface is very flexible and supports a wide variety of devices from major manufacturers.

The following interface types are provided (in each of the two display ports):

- A parallel video interface (for synchronous access) - up to 24-bit data bus.
  - Supports BT.656 (8-bit) and BT.1120 (16-bit) protocols
  - Supports HDTV standards SMPTE274 (1080i/p) and SMPTE296 (720p)
- A parallel bidirectional bus interface (for asynchronous access) - up to 32-bit data bus.
- A serial interface - 3-wire, 4-wire and 5-wire (two flavors) (for asynchronous access)

The supported formats for pixel data are: RGB and YUV 4:2:2.

For the interface clock, there are the following options (independently for each port)

- Derived from the IPU internal clock (master mode)
- Provided by an external source (slave mode)

The transfer rate supported

- When a single port is active, the pixel clock rate is up to 198 MHz
- When both ports are active
  - Each pixel clock rate may be up to MHz[1]
  - The sum of pixel clock rates is up to MHz

9.2.1.2.5 Connecting To Display Devices

IPU allows the connectivity to multiple display devices.

In particular, it supports the following setup:

- Primary LCD display.
- Second LCD display.
- External display; e.g. TV.

Simultaneous functionality of the above devices is possible in each of the following ways:

- Two devices can be accessed (synchronously or asynchronously) independently, each through a different port: each using any of the available interfaces.
- Two devices can time-share asynchronous accesses through the legacy serial and parallel interfaces, using the CS signals.

9.2.2 Processing

The IPU processes rectangular blocks of pixels. The processing is performed in four modules - VDIC, IC, DP and IRT.

(see Figure 9-5 and Table 9-3). Several time-shared data flows are supported, as described in the following table.

Table 9-4. Time-Shared Data Flows Through The IPU

Name	Number	Type	Flow	Target	Restrictions
Display Refresh/ Update	5 flows (at most two of them of type DS1)	DS1	Fmem -> DP -> Display	Synchronous Access (e.g. display refresh; controlled by the DI)	—
		DS2	Fmem -> DP -> Display	Asynchronous Access (e.g. display update)	—
		DS3	Fmem <-> Display	Generic Data Transfer	—
	1 flow	DS4	MCU <-> Display	Direct Access	—
Video Playback	1 flow	PL1	Bmem -> VDIC -> IC -> Bmem -> IRT -> Fmem + DSx	Main option	—
		PL2	Fmem -> IRT -> Bmem -> IC -> DP	Low power	Progressive source Large enough window

Table continues on the next page...

1. Specified pixel clocks frequencies are applicable for internal clocks, but may be limited by IO buffers speed capability. Final numbers are subjected to AC characterization.

Table 9-4. Time-Shared Data Flows Through The IPU (continued)

Name	Number	Type	Flow	Target	Restrictions
				(branching to DSx, as a video plane)	No other video flows
		PL3	Fmem -> VDIC -> IC -> DP	Low power (branching to DSx, as a video plane)	Interlaced source Large enough window No other video flows
Camera Preview	2 flows (VF2 may be used also as a playback flow)	VF1	Sensor -> IC -> Bmem -> IRT -> Fmem + DSx	main option	Single progressive input
		VF2	Sensor -> Fmem -> VDIC -> IC -> Bmem -> IRT -> Fmem + DSx	two inputs and/or interlaced input	When the VDIC is used for de-interlacing, one of the three input fields can go directly from the sensor to the VDIC
		VF4	Sensor -> IC -> Fmem + DS1	Low power RAM-less Display Single Display Buffer (in internal memory) Tearing-less	Single progressive input Refresh rate = 2x sensor frame rate Large enough window No other video flows
Video Record	2 flows	RCx	IC -> Bmem -> IRT -> Fmem	(branching from VFx)	
Graphic Overlays	2 flows	GF1	Fmem -> IC	(combining with the main flow	
	2 flows	GF2	Fmem -> DP		

9.2.2.1 Display Processor (DP)

The Display Processor performs all the processing required for data sent to a display.

- Combining 2 video/graphics planes
- Overlaying a simple HW cursor 32 x 32 pixels, uniform color; may be combined logically with the background.
- Color conversion/correction - linear (multiplicative and additive) Programmable; including:
  - YUV <-> RGB, YUV<->YUV conversions where YUV stands for any one of the color formats defined in the MPEG-4 standard
  - Adjustments: brightness, contrast, color saturation...
  - Special effects: gray-scale, color inversion, sephia, blue-tone...
  - Hue-preserving gamut mapping - for minimal color distortion
  - Applied to the output of combining or to one of the inputs
- Gamma correction and contrast stretching - programmable piecewise-linear map

The DP processes a single data flow at any given time, but it supports up to three data flows, by time sharing, one of them may be synchronous.

The data rate is up to 198M pixels/sec

9.2.2.2 Video de-interlacer (VDIC)

VDIC, the video de-interlacer and combiner module, has two operation modes

- De-interlacing: converts an interlaced video stream to progressive order.
- Combining: combines two video/graphics planes and a background color

The input and output to/from the VDIC is as follows:

- Input for de-interlacing: three consecutive fields
  - Source
    - The most recent field may come from the CSI or from system memory
    - The other two fields are read from memory
  - Field size: Supports up to 1080p
  - Pixel format: YUV 4:2:2/4:2:0, 8 bits/value
  - Typical video sources - SDTV: 480i30 (720x480 at 30 fps) or 576i25 (720x576 at 25 fps) and HDTV: 1080i30 (1920x1080 at 30 fps)
- Input for combining: two progressive video/graphics planes
  - Source: system memory
  - Plane size: up to 1920x1200 pixels.
  - Pixel format: RGB/YUV 4:2:2, 8 bits/value
- Output: progressive frame
  - Destination: to system memory or to the Image Converter.
  - Frame size: up to 1920x1200 pixels.
  - Rate: up to 198M pixels/sec.
  - Format: same as input format.

De-interlacing is performed using a high-quality 3-field filter, which is motion adaptive:

- For slow motion - retains the full resolution (of both top and bottom fields)
- For fast motion - prevents motion artifacts

VDIC supports a single video stream at any given time.

9.2.2.3 Image Converter (IC)

The Image Converter performs various operations on a video stream.

The operations performed are:

- Resizing:
  - Fully flexible resizing ratio Maximal downsizing ratio: 8:1. Subject to this limitation, any N->M resizing can be performed
  - Independent horizontal and vertical resizing ratios.
- Color conversion/correction - linear (multiplicative and additive) Programmable; including:
  - YUV <-> RGB, YUV<->YUV conversions where YUV stands for any one of the color formats defined in the MPEG-4 standard
  - Adjustments: brightness, contrast, color saturation...
  - Special effects: gray-scale, color inversion, sephia, blue-tone
- Combining with a graphics plane (e.g. application-specific overlay)
- Horizontal inversion

The IC supports three time-shared data flows: record, camera preview and playback (the first two share a common input).

The frame resolution supported is up to 4096x4096 for input and up to 1024x1024 for output. Wider frames can be processed by the IC by splitting them to vertical stripes.

Data throughput:

- For a single active task: up to 240M pixels/sec for input and up to 120M pixels/sec for output.
- For several active tasks: up to 200M pixels/sec for input and up to 100M pixels/sec for output.

9.2.2.4 Image Rotator (IRT)

The Image Rotator performs any combination of the following:

- 90-degree rotation
- Horizontal inversion
- Vertical inversion

The data throughput

- When a single task is active: up to 120M pixels/sec.
- When more tasks are active: up to 100M pixels/sec.

9.2.3 Automatic Procedures

The IPU is equipped with powerful control and synchronization capabilities to perform its tasks with minimal involvement of ARM and minimal use of memory.

In particular, it includes:

- An integrated DMA controller with an AXI master port, allowing autonomous access to the system memory.
- An integrated display controller, performing screen refresh of a RAM-less display.
- A page-flip double buffering mechanism, synchronizing read and write access to system memory, to prevent tearing effects.
- A double/triple buffer synchronization mechanism with a video/graphics source.
- Internal synchronization, e.g., between input from sensor and output to display.

As a result, in most cases, the MCU is involved only when it also performs part of the processing (e.g. video coding). In particular, the following procedures are performed by the IPU completely autonomously:

- Screen refresh for RAM-less displays
- Camera preview (displaying a view finder - a video-stream from the image sensor to the display).

NOTE

Direct camera to display flow is possible when the frame rate of the source and the destination is the same (typically the target display will be asynchronous display, where the display is updated at the rate of the source).

Typically, there are extended periods of time in which there is no other activity in the system. The MCU - being idle - can be put to a low-power mode, reducing the power consumption and extending significantly the battery life.

The IPU supports several techniques to reduce further the power consumption of the display system:

- Dynamic backlight control, with low-light compensation by image enhancement

Further features and capabilities of the automatic procedures include:

- Automatic display of a changing image (animation) or moving image (scrolling).
- The timing of the display update can be adjusted to avoid tearing.
- The video stream from an image sensor can be sent directly to the display buffer used for screen refresh, while avoiding tearing.

The clock sources received by the IPU are listed in the following table.

Table 9-5. IPU Clock Sources

Name	Symbol	Source	Rate	Comments
High-Speed Processing Clock	HSP_CLK	Clock control Module	Up to 270 MHz	
Display Interface Clocks	DI_CLK0 DI_CLK1	Clock control Module or an external PLL	Up to 220 MHz	Optional; needed for synchronization with interface bridges

9.3 LVDS Display Bridge (LDB)

LVDS Display Bridge (LDB) will be used to connect the IPU (Image Processing Unit) to the External LVDS Display Interface.

9.3.1 LDB Overview

Table 9-6. IP Parametric Table

Name	Description
Function	Connectivity to displays with LVDS interface
External I/O Pins Notes: Those are LVDS IO pads	LVDS Display port: 2 channels. Each channel consists of: • 1 clock pair • 4 data pairs Each pair contains - LVDS special differential pad (PadP, PadM). total of 20 IO pads.
SoC Buses	None. Only configuration signals.
Interrupts	None
DMA Requests	None
Number of instantiations	1
Clock sources and range	DI0_CLK, DI1_CLK - Display interface clock: 20-170 MHz DI0_SER_CLK, DI1_SER_CLK - Serializer clock: 140-595 MHz

The purpose of the LDB is to support flow of synchronous RGB data from the IPU to external display devices through the LVDS interface. This support covers all aspects of these activities:

- Connectivity to relevant devices - Displays with LVDS receivers.
- Arranging the data as required by the external display receiver and by LVDS display standards.
- Synchronization and control capabilities.

9.3.2 LDB External Ports

The LDB has the following ports:

- Two input parallel display ports.
- Two Output LVDS channels - Each channel consisting of 4 data pair, and 1 clock pair (pair=LVDS pad contains PadP, PadM).
- Control signals - to configure LDB parameters and operations.
- Clocks from SOC PLLs.

9.3.3 Input Parallel Display Ports

One or Two (DI0, DI1) parallel RGB input ports are supported (configurable). Only synchronous access mode is supported.

Each RGB data interface contains the following:

- RGB Data of 18 or 24 bits
- Pixel clock
- Control signals: HSYNC, VSYNC, DE, and 1 additional optional general purpose control.

Total of up to 28 bits per data interface are transferred per pixel clock cycle.

Rates supported:

- For dual-channel output: Up to 170 MHz pixel clock (e.g. UXGA - 1600x1200 @ 60 Hz + 35% blanking)
- For single-channel output: Up to 85 MHz per interface. (e.g. WXGA - 1366x768 @ 60 Hz + 35% blanking).

9.3.3.1 Output LVDS Ports

There are 2 LVDS channels. These inputs are used to communicate RGB data and controls to external LCD displays with LVDS interface, or through LVDS receivers.

The LVDS ports may be used as follows:

- One single-channel output
- One dual channel output: single input, split to two output channels
- Two identical outputs: single input sent to both output channels
- Two independent outputs: two inputs sent, each, to a different output channel

9.3.3.2 Control signals

Control signals are used to configure LDB operation.

9.4 Video Data Order Adapter (VDOA)

The goal of the VDOA is to re-order video data from the "tiled" order used by the VPU to the conventional raster-scan order needed by the IPU.

9.5 Display Content Integrity Checker (DCIC)

The goal of the DCIC is to verify that a safety-critical information sent to a display is not corrupted.

Such a verification is mandatory for warning icons in the instrument cluster of a car, to comply with the ASIL B (Automotive Safety Integrity Level B) specification. It is also required in other safety-sensitive systems.

9.6 Video Processing Unit (VPUv6)

The VPU is a multi-standard video codec (encoder/decoder) capable of handling multiple streams simultaneously by time multiplexing.

It is a very flexible block consisting of hardware accelerators surrounding a programmable core. The VPU presents to the system a register mapped interface that is controlled by the embedded processor. End users should only interface with the VPU using the API that is provided with each BSP Programmers User Guide. This API isolates the user from possible changes in the register level interface.

In the following table we can see a summary of the VPU specs. The VPU has its own DMA driven AXI masters that allow it to retrieve the required data directly from system memory (DDR and iRAM). The load in the host ARM platform is negligible because it only needs to interact with the VPU at the frame level.

Table 9-7. VPU Spec Summary

Name	VPU
Function	Decode video streams including optional video processing such as rotation, deringing and mirroring.
Supported encoders	MPEG-4 SP H.263 V2 + Annex J, K (RS=0 and ASO=0), and T H.264 BP, CBP MJPEG Baseline
Supported decoders	MPEG-2 MP, HP VC-1 SP, MP, AP MPEG-4 SP, ASP H.263 V2 + Annex J, K (RS=0 and ASO=0), and T H.264 BP, MP, HP H.264-MVC BP, MP, HP DivX v3,4,5 Real Video 8, 9, 10 MJPEG Baseline On2 VP8 AVS Jizhun
External I/O Pins (List, Type, Schmidt Trigger, Speed)	No external I/O pins are needed
SoC Buses (List, Type, Bandwidth)	64 bit AXI master for accessing the system memory and the optional secondary AXI but to iRAM IPBus slave for host control
Interrupts	Two interrupts
DMA Requests	Integrated DMA controller on the AXI master port
Endianness	64 and 32 bit BE/LE
Number of instantiations	1
Clock sources and range	Core clock: up to 328 MHz AXI bus clock: up to 328 MHz IP bus clock: up to 66 MHz

NOTE

RealNetworks video codec is disabled by default on i.MX 6 series processors. Please contact your FSL sales representative for more details.

The i.MX 6Solo/6DualLite processor has a high-performance video processing unit (VPU), which covers many standard and high definition video decoders and encoders as a multi-standard video codec engine as well as several important video processing such as rotation and mirroring.

9.6.1 Basic Structure

The following Figure 9-6 shows the basic structure of the VPU. The VPU is self contained except for memory accesses. It is connected to the memory subsystem through two AXI master ports. The ARM configures the VPU operation through the IP bus that is converted to APB in a gasket.

A new feature is added for allowing VPU encoding a frame even the IPU writes only a small portion of source video (from camera sensor) into the frame buffer. This feature is called IPU-VPU sub-frame synchronization. Four additional signals are added for implementing this feature with three of them from IPU to VPU, and one of them from VPU to IPU, as shown in Figure 9-6 for the signal category of "IPU VPU sub-frame sync"

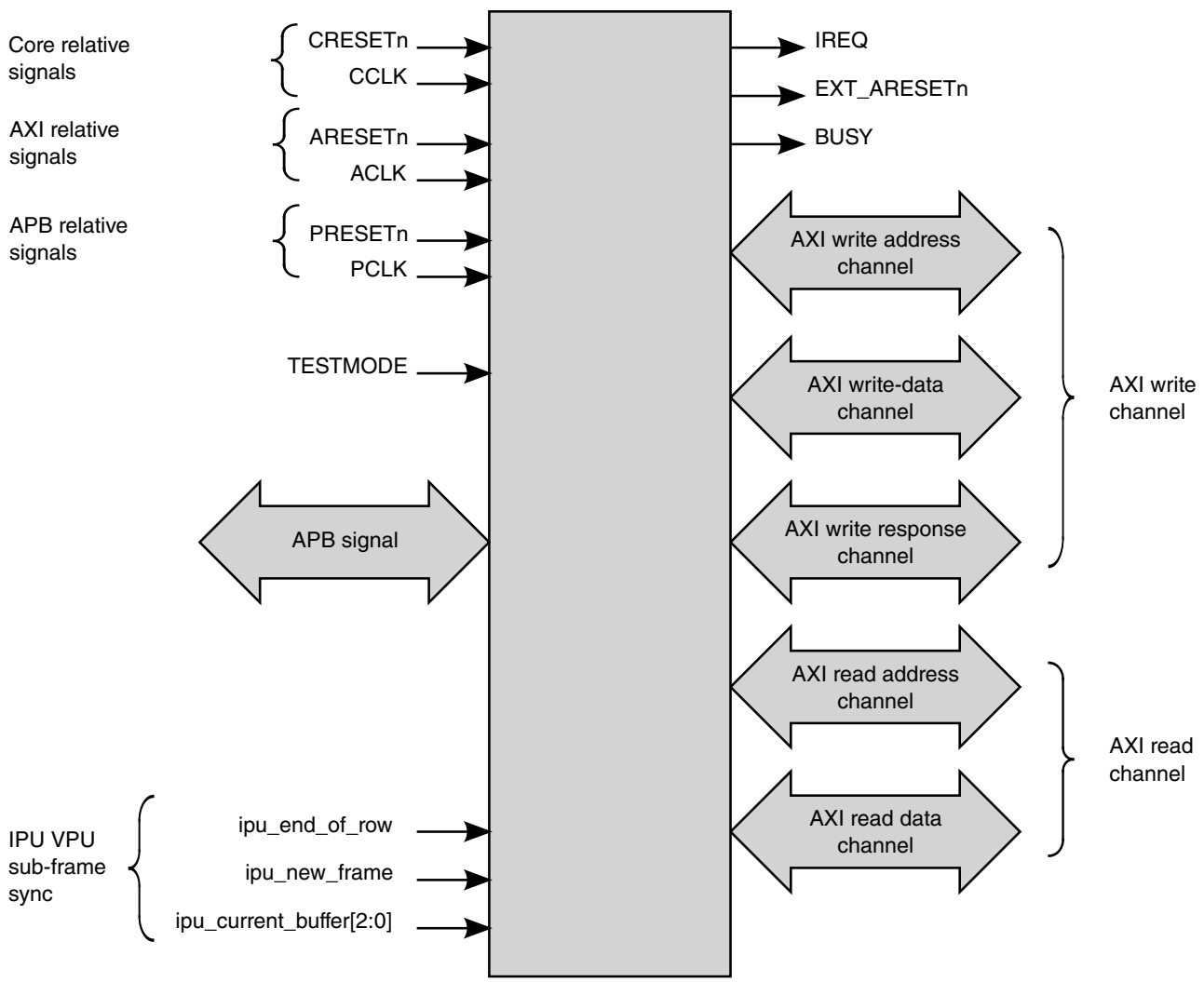


Figure 9-6. The Block Diagram of the VPU

9.6.2 Feature summary

The following table describes the VPU's encoding / decoding capabilities:

Table 9-8. VPU decoding/encoding capabilities

Dec/Enc	Standard	Profile	Resolution	Bitrate	Comments
HW Decoder	MPEG-2	Main-High	1080 i/p, 30fps	50 Mbps	1080p+SD at 30fps, 720p60
	MPEG4/XviD	SP/ASP	1080 i/p, 30fps	40 Mbps	—
	H.263	P0/P3	16CIF, 30 fps	20 Mbps	—
	H.264	BP/CBP/MP/HP	1080 i/p, 30 fps	50 Mbps	1080p+SD at 30 fps, 720p60
	H.264-MVC	BP/MP/HP	720p, 30 fps	—	—
	VC1	SP/MP/AP	1080 i/p, 30 fps	45 Mbps	1080p+SD at 30 fps, 720p60
	RV	8/9/10	1080 p, 30 fps	40 Mbps	—
	DivX	3/4/5/6	1080 i/p, 30 fps	40 Mbps	—
	On2 VP8	—	720p, 30 fps	20 Mbps	—
AVS	Jizhun	1080 i/p, 30fp	40 Mbps	—	—
MJPEG	Baseline	8192x8192	120 Mpixel/sec	Perf shown at 4:4:4 format	—
Hardware encoder	MPEG4	Simple	720p, 30 fps	12 Mbps	VPU can generate higher bitrate than the maximum specified by the corresponding standard.
	H.263	P0/P3	4CIF, 30 fps	8 Mbps	
	H.264	BP/CBP	1080p, 30 fps	14 Mbps	
	MJPEG	Baseline	8192 x 8192	160 Mpixel/sec	Perf shown at 4:2:2 format

VPU can also perform the following:

- On-the-fly (90 x n) degree simultaneous rotation and mirroring (n = 0,1,2,3).
- Post-processing
  - De-blocking filtering for MPEG-4
  - De-ringing filtering for MPEG-4 and H.264 decoder

9.7 OpenGL ES 3D Graphics Processing Unit (GPU3Dv5)

9.7.1 OpenGL Overview

9.7.2 OpenGL Features

Summary of features in the GPU3D includes:

- OpenGL ES 2.0 compliance, including extensions; OpenGL ES 1.1; OpenVG 1.1
- IEEE 32-bit floating-point pipeline
- Ultra-threaded, unified vertex and fragment shaders
- Low bandwidth at both high and low data rates
- Low CPU loading
- Up to 12 programmable elements per vertex
- Dependent texture operation with high-performance
- Alpha blending
- Depth and stencil compare
- Support for 8 fragment shader simultaneous textures
- Support for 4 vertex shader simultaneous textures
- Point sampling, bi-linear sampling, tri-linear filtering, and cubic textures
- Resolve and fast clear
- 8k x 8k texture size and 8k x 8k rendering target

9.7.3 OpenGL Block Diagram

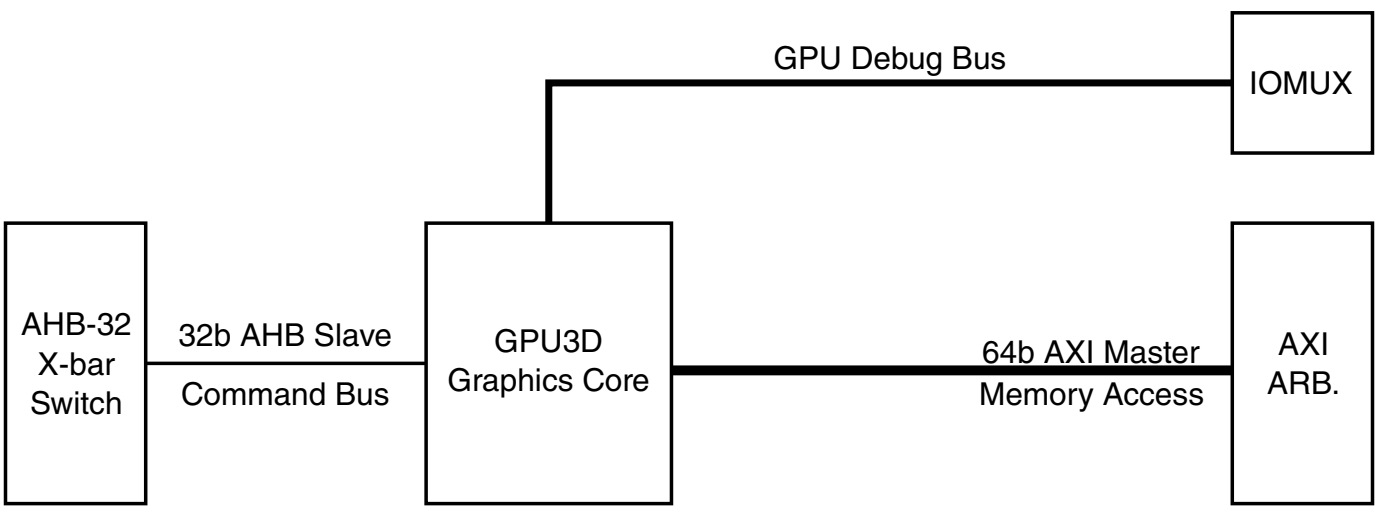


Figure 9-7. GPU System Integration Diagram

9.7.4 OpenGL Performance

- GPUv5 @ 532MHz (clock rate TBD)

9.7.5 OpenGL Software

API / Driver Support

- OpenGL ES 1.1, 2.0, and 3.0 (draft)
- OpenVG 1.1
- EGL 1.4
- OpenVG 1.1

Operating Systems

- Windows CE
- Linux Embedded and X11
- Android

9.8 2D Graphics Processing Unit (GPU2Dv2)

9.8.1 2D feature summary

GPU2D has the following features:

- Bit BLT and stretch BLT
- Rectangle fill and clear
- Line drawing
- High performance stretch and shrink
- Mono expansion for text rendering
- ROP2, ROP3, and ROP4
- Alpha blending including Java 2 Porter-Duff compositing blending rules
- Support rendering size of 32Kx32K
- 90/180/270 degree rotation
- Transparency by monochrome mask, chroma key, or pattern mask
- Color space conversion between YUV and RGB
- High quality image scaling, using up to 9x9 separable filter
- Bit-Blit Formats
    - A1R5G5B5 (source/destination)
    - A4R4G4B4 (source/destination)
    - X1R5G5B5 (source/destination)
    - X4R4G4B4 (source/destination)

    - R5G6B5 (source/destination)
    - X8R8G8B8 (source/destination)
    - A8R8G8B8 (source/destination)
    - 8-bit color index (source only)
    - A8 (source/destination)
    - 1-bit monochrome (source only)
- Filter Blit Formats
    - A1R5G5B5 (source/destination)
    - A4R4G4B4 (source/destination)
    - A8R8G8B8 (source/destination)
    - R5G6B5 (source/destination)
    - X1R5G5B5 (source/destination)
    - X4R4G4B4 (source/destination)
    - X8R8G8B8 (source/destination)
    - YUV (source only):
        - NV12 (4:2:0, 2 planes)
        - NV16 (4:2:2, 2 planes)
        - UYVY (4:2:2, interleave)
        - YUY2 (4:2:2, interleave)
        - YV12 (4:2:0, 3 planes)
    - 8-bit color index(source only)

9.8.2 2D Block Diagram

Figure 9-8 below presents the integration of the GPU2D core in i.MX 6Solo/6DualLite SoC.

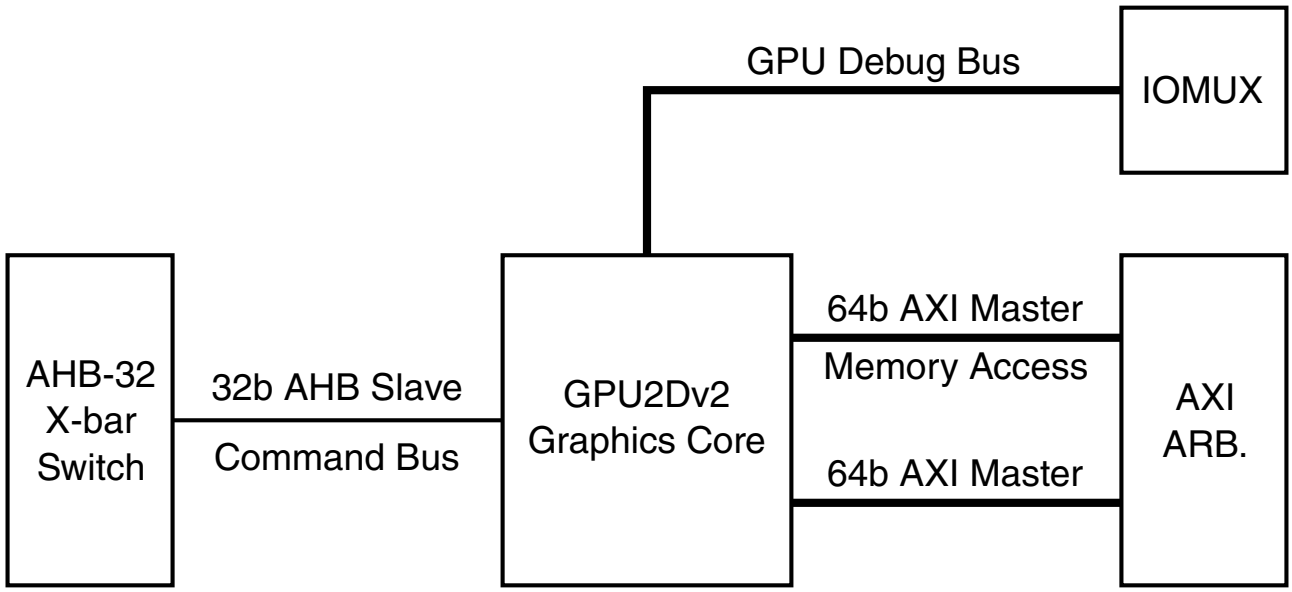


Figure 9-8. GPU2Dv2 (GPU2D) Interface Diagram

9.8.3 2D Performance

- Geometry Rate: 26.6M Triangles/sec
- Pixel Rate: 256M pixels/sec

9.8.4 2D Software

API / Driver Support

- GDI/DirectDraw
- DirectFB
- X11 EXA

Operating Systems

- Windows CE
- Linux Embedded and X11
- Android

9.9 HDMI TX - HD Multimedia Interface transmitter

9.9.1 HDMI Introduction

HDMI (High-Definition Multimedia Interface) is a compact audio/video interface for transmitting uncompressed digital video data and uncompressed/compressed digital audio data. HDMI connects digital audio/video sources-such as set-top boxes, Blu-ray Disc players, personal computers (PCs), video game consoles, and AV receivers to compatible digital audio devices, computer monitors, and digital televisions.

HDMI supports, on a single cable, any TV or PC video format, including standard, enhanced, and high-definition video, up to 8 channels of digital audio, and a Consumer Electronics Control (CEC) connection. The CEC allows HDMI devices to control each other when necessary and allows the user to operate multiple devices with one remote control handset.

Because HDMI is electrically compatible with the signals used by Digital Visual Interface (DVI), no signal conversion is necessary, nor is there a loss of video quality when a DVI-to-HDMI adapter is used.

The HDMI Transmitter (HDMI TX) consists of two parts:

- HDMI TX Controller
- HDMI TX PHY

Figure 9-9 depicts the HDMI TX integration scheme.

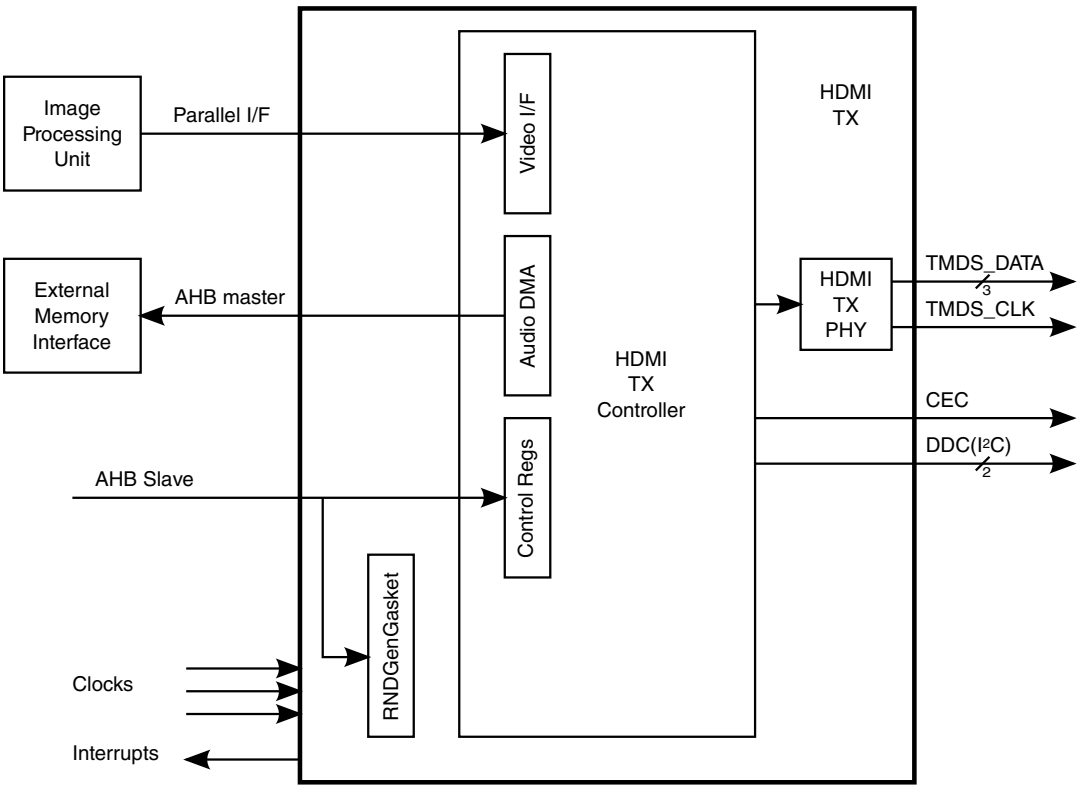


Figure 9-9. HDMI TX integration

The video interface is parallel with 24 bit/pixel data and separate signals for vertical (VSYNC) and horizontal (HSYNC) synchronization. The video data is supplied by the Image Processing Unit (IPU).

The audio data is read from an external memory by the internal Audio DMA via the master AHB interface. The HDMI TX is controlled via the AHB interface.

9.9.2 Features

The HDMI TX features are listed in Table 9-9.

Table 9-9. HDMI TX Features

Specification item	Requirement
General Features	
Standard Compliance	HDMI 1.4a

Table continues on the next page...

Table 9-9. HDMI TX Features (continued)

	HDMI CTS 1.4a DVI 1.0 HDCP 1.4
Consumer Electronic Control	Supported
Monitor Detection	Hot plug/unplug detection and link status monitor supported
Video Features	
Video Standard Compliance	EIA/CEA-861D
Supported Video Resolutions	Up to 1080p@120Hz HDTV display, up to QXGA graphics display, HDMI 1.4a 4K x 2K video formats HDMI 1.4a 3D video modes with up to 340MHz TMDS clock
Pixel Clock Frequency	From 25 MHz to 340 MHz
Input Data Formats	Parallel YCbCr 4:4:4 and parallel RGB 4:4:4 and parallel YCbCr 4:2:2
Input Color Depth	24/30/36/48 bits/pixel
Input Syncs Format	Separate HSYNC and VSYNC plus data enable control
Internal Video Processing	Interpolation YCbCr 4:2:2 to 4:4:4 Color space conversion YCbCr to RGB and vice versa
Audio Features	
Audio Standard Compliance	IEC60958, IEC61937
Supported Audio Formats	All audio formats as specified by the HDMI Specification Version 1.4a supported.
Audio Input Interfaces	Audio DMA
Audio Sampling Rate	Up to 192 kHz

9.10 Display / Sensor MIPI interfaces

9.10.1 Introduction

i.MX 6Solo/6DualLite Application Processor features MIPI DSI/CSI-2 interfaces that includes:

- MIPI DSI Host controller with associated MIPI D-PHY Tx (One clock lane, two Data lanes)
- MIPI CSI-2 Host controller with associated MIPI D-PHY Rx (one clock lane, two Data Lanes)

9.10.2 MIPI DSI

MIPI DSI is a high performance serial interconnect bus for mobile applications connecting display system to the host system.

Table 9-10. DSI Parametric Table

Name	DSI
Function	High speed serial interface controller for MIPI Display interface
Supported standard version	MIPI DSI Compliant DSI Version 1.01 DPI Version 2.0 DBI Version 2.0 DCS Version 1.02 PPI for D-PHY MIPI D-PHY Version 1.0
External I/O Pins (List, Type, Schmidt Trigger, Speed)	Analog at chip boundary: DATAP[1:0] DATAN[1:0] CLKP CLKN Supplies AVDD -2.5V general analog supply AVDDREF - 2.5V analog supply for D-PHY reference source (low noise) should be directly connected to package ball VDD - 1.1V digital supply GND
SoC Buses (List, Type, Bandwidth)	Synchronous Pixel Interface (DPI) Display Bus Interface (DBI) IPB slave interface x 32 bit PPI to D-PHY
Interrupts	1
DMA Requests	N/A
Num of instantiations	1
Clock sources and range	AHB clock (132MHz) Reference clock for Tx D-PHY - 6 ... 27 MHz

There is one instance of DSI port in the chip. This interface support from 80 Mbps up to 1 Gbps speed per data lane. The DSI Receiver core can mange one clock lane and up to 4 data lanes through the lane management, however two data lanes are implemented in the transmitter D-PHY, therefore the maximum throughput of display port is 2Gbps.

DSI host core is capable of supporting a variety of resolutions and formats:

- Resolution:
  - QQVGA
  - QCIF, QVGA
  - CIF
  - VGA
  - WVGA
  - SVGA
  - XVGA
- Pixel format:
  - RGB565
  - LRGB565
  - RGB666
  - RGB888

The DSI can support both command and video modes and up to four virtual channels to accommodate multiple displays.

- Command and video mode support (type 1, 2, 3, and 4 display architecture)
- Mode switching: low power and ultra low power
- Burst mode: dual video channel
- Non-burst mode: single video channel
- Bus turnaround
- Fault error recovery scheme

DPI and DBI could coexists in the system but only one of them could be active in a certain time moment

9.10.3 MIPI CSI-2

MIPI CSI-2 is a high performance serial interconnect bus for mobile applications connecting camera sensors to the host system.

Table 9-11. CSI Parametric Table

Name	CSI-2
Function	High speed serial interface controller for MIPI sensor interface
Supported standard version	MIPI CSI-2 Version 1.0 MIPI PPI interface for D-PHY MIPI D-PHY Version 1.0
External I/O Pins (List, Type, Schmidt Trigger, Speed)	Analog at chip boundary:

Table continues on the next page...

Table 9-11. CSI Parametric Table (continued)

Name	CSI-2
	DATAP[1:0] DATAN[1:0] CLKP CLKN Supplies AVDD -2.5V general analog supply AVDDREF - 2.5V analog supply for D-PHY reference source (low noise) should be directly connected to package ball VDD - 1.1V digital supply GND
SoC Buses (List, Type, Bandwidth)	Synchronous Pixel Interface - 32 bit, data formatting compliant to CSI-2 specification IPS slave interface x 32 bit
Interrupts	2
DMA Requests	N/A
Num of instantiations	1
Clock sources and range	AHB clock (132 MHz)

There is one instance of CSI-2 port in the chip. This interface supports from 80 Mbps up to 1 Gbps speed per data lane. The CSI-2 Receiver core can mange one clock lane and up to two data lanes through the lane management and de-packetization, providing a maximum throughput of 4 Gbps transfer rate.

9.10.4 D - PHY

D-PHY serves as physical layer for both MIPI DSI and CSI-2 interfaces.

D-PHY Transceiver unit is responsible for transmission and reception of data in High speed (HS) or Low Power (LP) mode. High speed mode is used for high-speed data transmission while the low power mode used for the control purpose. Point-to-point lane interconnect can be used for either data or clock signal transmission. High speed receiver is a differential line receiver while low- Power receiver is an un-terminated, single-ended receiver circuit.

Table 9-12. D-PHY Parametric Table

Name	D-PHY TX, D-PHY RX
Function	MIPI D-PHY dual mode transceiver
Supported standard version	D-PHY specification v1.0

Table continues on the next page...

Table 9-12. D-PHY Parametric Table (continued)

Name	D-PHY TX, D-PHY RX
External I/O Pins (List, Type, Schmidt Trigger, Speed)	One differential pair per lane. See CSI-2 and DSI parametric tables
SoC Buses (List, Type, Bandwidth)	PPI i/f
Interrupts	
DMA Requests	N/A
Num of instantiations	6 - 3 in DSI interface and 3 for CSI-2 interface
Clock sources and range	Reference clock - 6...27MHz

A single lane module is shown in the figure below.

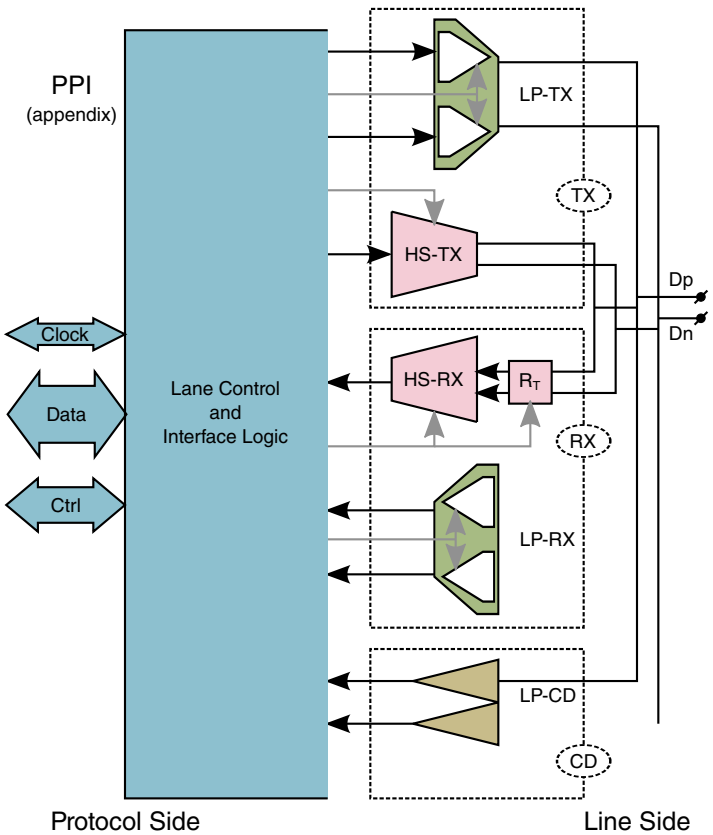


Figure 9-10. Single lane D-PHY- Block Diagram.

There is one instance of D-PHY TX (hard macro; two data lanes, one clock lane, PLL) and one instance of D-PHY RX (hard macro: four data lanes, one clock lane, no PLL) in i.MX 6Solo/6DualLite.

9.11 Audio subsystem

The audio subsystem consists of the following modules: SSI-1, SSI-2, SSI-3, AUDMUX, ESAI, SPDIF, ASRC, and MLB. In addition, the IOMUX must be appropriately configured to get signals in and out of the chip.

Audio Subsystem Module Overview provides an overview of each of the audio subsystem component modules, followed by a module-specific section.

9.11.1 Audio Subsystem Module Overview

The following figure shows a high level block diagram of the audio subsystem.

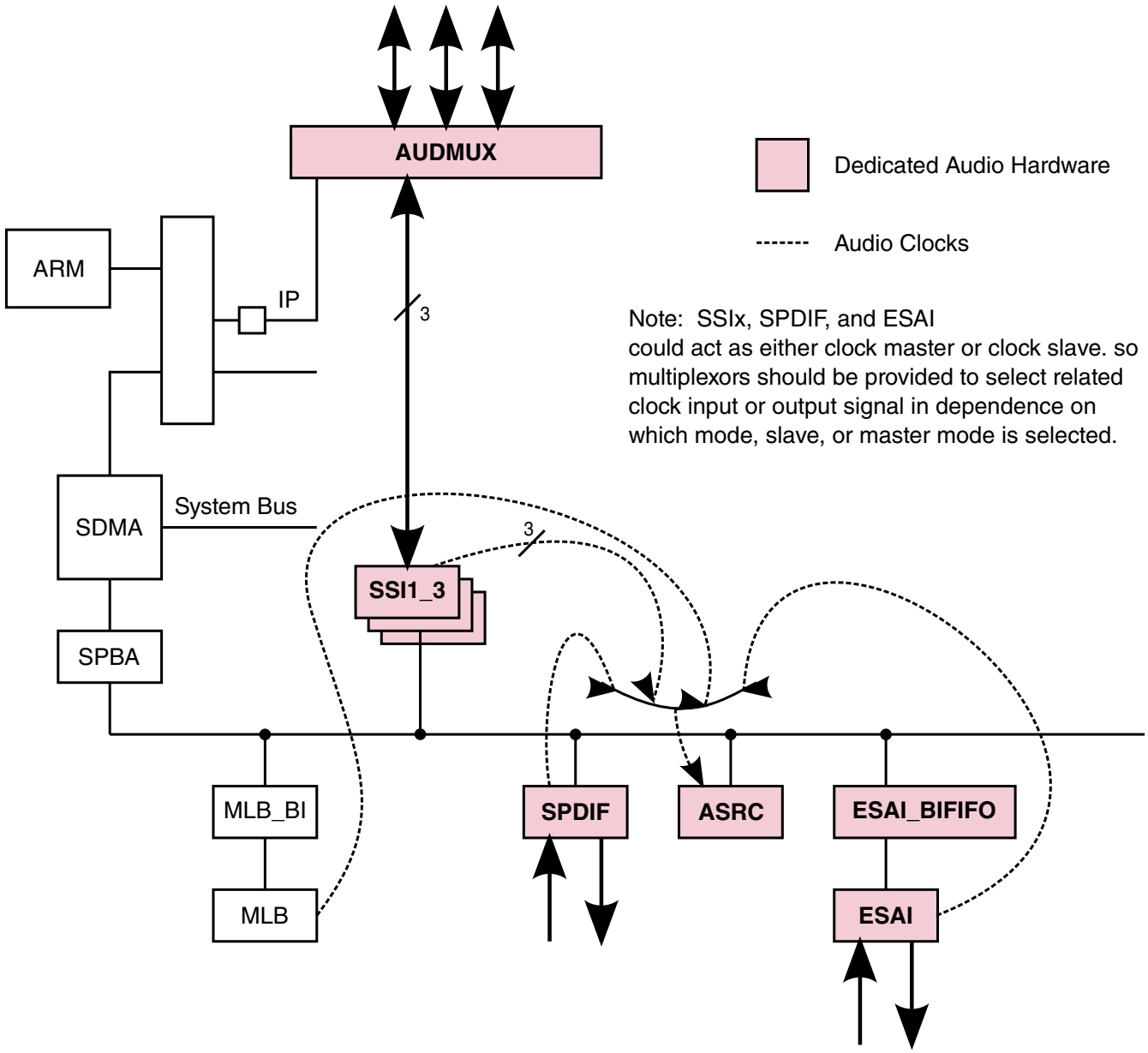


Figure 9-11. Audio subsystem block diagram

SSI1–3 are synchronous serial interfaces used to transfer audio data. SSI1–3 are on the shared peripheral bus. Instead of connecting to the IOMUX directly, their serial lines connect to the digital audio mux (AUDMUX).

AUDMUX provides flexible, programmable routing of the serial interfaces (SSI1, SSI2, or SSI3) to and from off-chip devices. AUDMUX routes audio data (and even splices together multiple time-multiplexed audio streams) but does not decode or process audio data itself. The ARM controls AUDMUX, but AUDMUX can route data even when the ARM is in a low-power mode.

ESAI (enhanced serial audio interface) provides a full-duplex serial port for serial communication with a variety of serial devices, including industry-standard codecs, SPDIF transceivers, and other processors. ESAI consists of independent transmitter and receiver sections, and each section has its own clock generator. ESAI is connected to the IOMUX and to the ESAI_BIFIFO module.

ESAI_BIFIFO (ESAI bus interface and FIFO) is the interface between the ESAI module and the shared peripheral bus. It contains the FIFOs used to buffer data to and from ESAI. It also provides the data word alignment and padding necessary to match the 24-bit data bus of the ESAI to the 32-bit data bus of the shared peripheral bus.

The SPDIF (Sony/Philips digital interface) audio module is a stereo transceiver that allows the processor to receive and transmit digital audio over it. The SPDIF receiver section includes a frequency measurement block that allows the precise measurement of an incoming sampling frequency. A recovered clock is provided by the SPDIF receiver section and may be used to drive both internal and external components in the system. SPDIF is connected to the shared peripheral bus.

ASRC (asynchronous sample rate converter) converts the sampling rate of a signal associated to an input clock into a signal associated to a different output clock. ASRC supports concurrent sample rate conversion of up to 10 channels of over 120dB THD+N. The sample rate conversion of each channel is associated to a pair of incoming and outgoing sampling rates. ASRC supports up to 3 sampling rate pairs. ASRC is connected to the shared peripheral bus.

9.11.2 Synchronous Serial Interface (SSI)

The Synchronous Serial Interface (SSI) is a full-duplex serial port that allows communication with external devices using a variety of serial protocols. The SSI supports a wide variety of protocols (SSI normal, SSI network, I2S, and AC-97), bit depths (up to 24 bits per word), and clock/frame sync options.

The SSI has two pairs of 15x32 FIFOs and hardware support for an external DMA controller in order to minimize its impact on system performance. The second pair of FIFOs provides hardware interleaving of a second audio stream which reduces CPU overhead in use cases where two timeslots are being used simultaneously.

The three SSIs may support three audio streams (possibly at different sample rates) simultaneously. SSI1, SSI2 and SSI3 are located on the Shared Peripheral Bus. Since the SDMA can directly access SSI1...SSI3 (being on the Shared Peripheral Bus), they can be used for high-bandwidth data transfers in order to optimize bus bandwidth consumption.

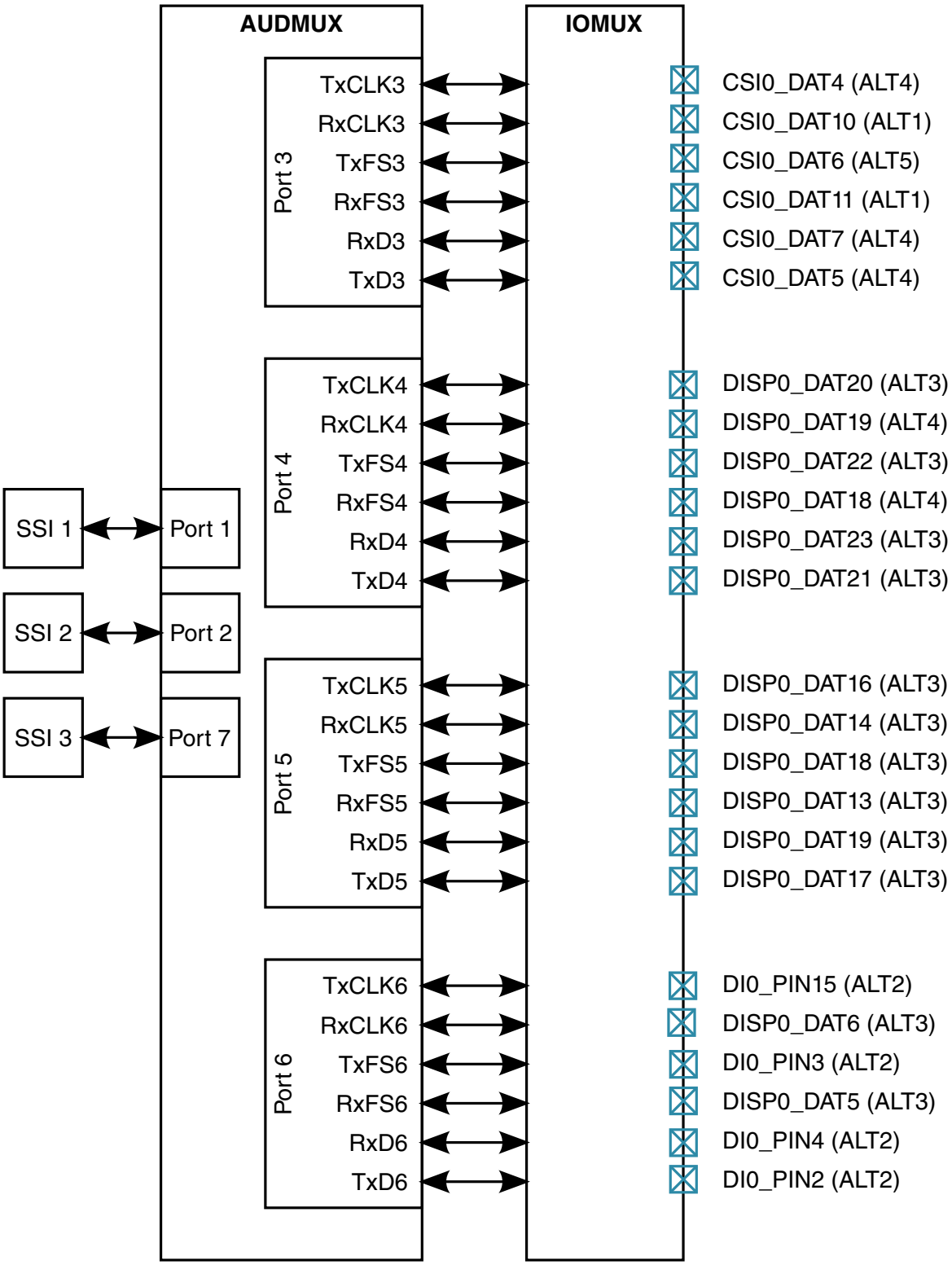
9.11.3 Digital Audio MUX (AUDMUX)

The Digital Audio Mux (AUDMUX) provides a programmable interconnect fabric for voice, audio, and synchronous data routing between host serial interfaces, such as SSI, and peripheral serial interfaces-that is, audio and voice codecs.

The AUDMUX includes two types of interfaces. Internal ports connect to the processor serial interfaces, and External ports connect to off-chip audio devices. A desired connectivity is achieved by configuring the appropriate host and peripheral ports.

The AUDMUX provides flexible, programmable routing of the on-chip serial interfaces to and from off-chip audio devices. The AUDMUX routes audio data (and even splices together multiple time-multiplexed audio streams) but does not decode or process audio data itself.

The following figure illustrates how the AUDMUX is connected in the system.



⊠ - Daisy Chain, i.e. more than one PAD is available (only one option shown above).

Figure 9-12. AUDMUX System Block Diagram

9.11.4 Enhanced Serial Audio Interface (ESAI)

The Enhanced Serial Audio Interface (ESAI) provides a full-duplex serial port for serial communication with a variety of serial devices, including industry-standard codecs, SPDIF transceivers, and other processors.

The ESAI consists of independent transmitter and receiver sections, each section with its own clock generator. All serial transfers are synchronized to a clock. Additional synchronization signals are used to delineate the word frames. The normal mode of operation is used to transfer data at a periodic rate, one word per period. The network mode is also intended for periodic transfers; however, it supports up to 32 words (time slots) per period. This mode can be used to build time division multiplexed (TDM) networks. In contrast, the on-demand mode is intended for non-periodic transfers of data and to transfer data serially at high speed when the data becomes available.

The ESAI has 12 pins for data and clocking connection to external devices. The ESAI is internally connected to the ESAI_BIFIFO, and does not connect directly to the shared peripheral bus. The ESAI interface is designed for a 24-bit data bus, while the shared peripheral data bus is 32-bit wide. Also, the ESAI data paths are only double buffered, not allowing efficient DMA service in the applications processor environment. The ESAI_BIFIFO allows increasing the data buffering and data width matching to the shared peripheral bus.

9.11.5 Sony/Philips Digital Interface (SPDIF)

The SPDIF module is a stereo that allows the processor transmit digital audio over it using the IEC60958 standard, consumer format. The chip provides one SPDIF transmitter with one output.

The SPDIF allows the handling of both SPDIF channel status (CS) and User (U) data.

For the SPDIF transmitter, the audio data is provided by the processor via the SPDIFTxLeft and SPDIFTxRight registers, and the data is stored in two 16-word-deep FIFOs, one for the right channel, the other for the left channel. The FIFOs support programmable watermark levels so that FIFO Empty service request can be triggered when the combined number of empty data words locations in both FIFOs is 8, 16, 24 or 32 words. It is recommended to program the watermark level to trigger a FIFO Empty service request when 16 word locations are empty. For optimal performance when servicing the FIFO Empty service request, the FIFOs should be written alternately, starting with the left channel FIFO. The Channel Status bits are also provided via the

corresponding registers. The SPDIF transmitter generates an SPDIF output bitstream in the biphase mark format (IEC 60958), which consists of audio data, channel status and user bits.

The data handled by the SPDIF module is 24-bit wide. The 24-bit SPDIF data is aligned in the 24 least significant bits of the 32-bit shared peripheral bus data word. The 8 most significant bits of the 32-bit word are ignored by the SPDIF Transmitter when data is being stored in the Transmit FIFOs from the peripheral bus. The 8 most significant bits of the 32-bit word are zeroed by the SPDIF Receiver module when the data is being read from the Receiver FIFOs to the peripheral bus.

Note that 16-bit data is left-aligned in the 24-bit word format of the SPDIF. When 16-bit data is to be transmitted, the 32-bit word to be written to the SPDIF Transmit FIFOs should be created as follows: the 16-bit data should be located in the middle two bytes of the 32-bit data word and the 8 bits of the LSB must be set to zero, while the 8 bits of the MSB will be ignored.

The SPDIF Transmit clock is generated by the SPDIF internal clock generator module and the clock sources are from outside of the SPDIF block. The ESAI, SSI's and MLB clock sources should provide a clock that is at least 64 x Fs, where Fs is the sampling frequency. The external clock source should provide at least 128 x Fs. Clocks of higher frequency may be provided as long as the multiplication factor is a power of 2 (for example, 128x, 256x or 512x). Also, clock frequency precision of 100ppm or better should be provided.

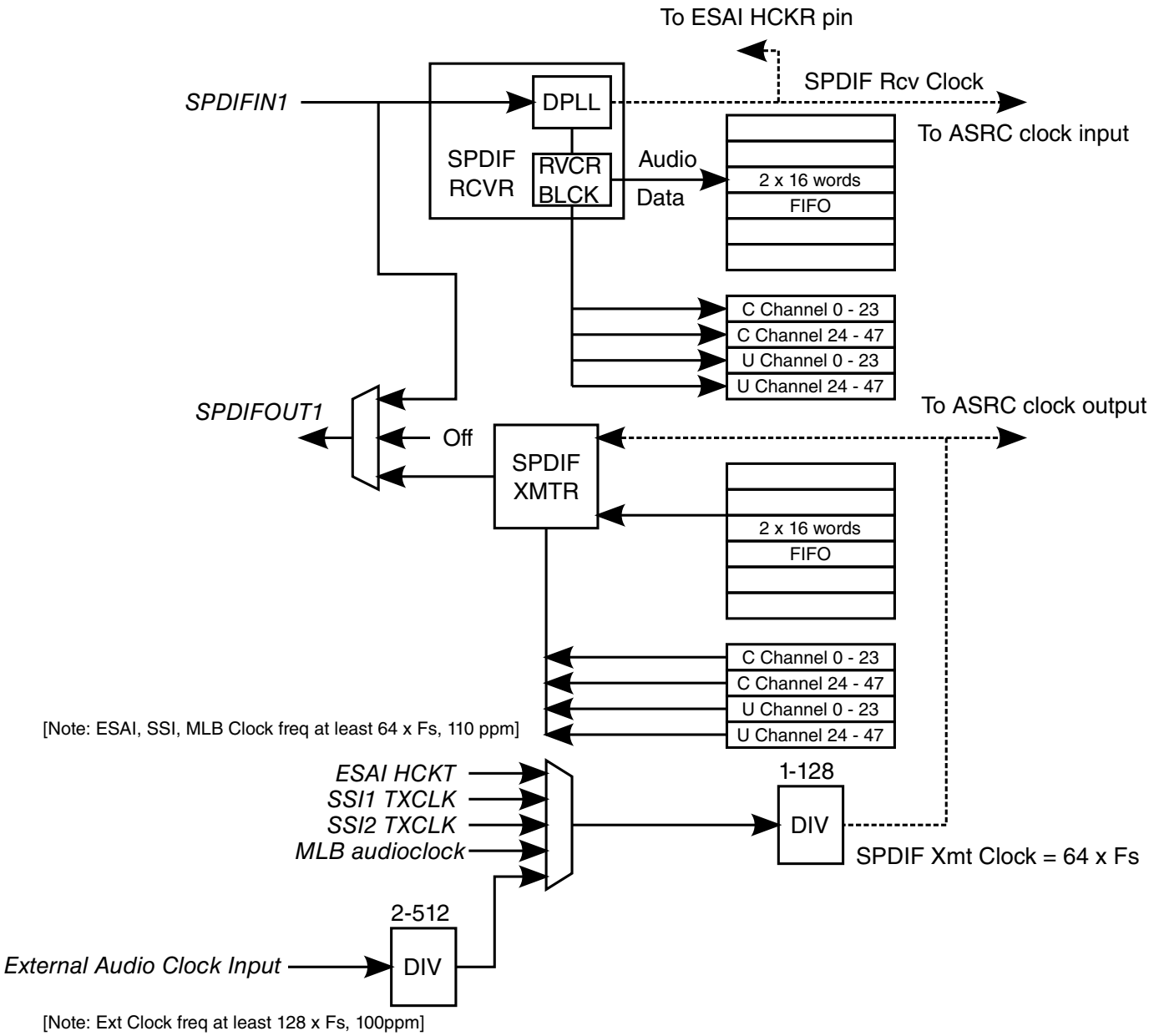


Figure 9-13. SPDIF Transceiver Clock Diagram

9.11.6 Asynchronous Sample Rate Converter (ASRC)

The incoming audio data may be received from various sources at different sampling rates. The outgoing audio data may have different sampling rates, and it can also be associated to output clocks that are asynchronous to the input clocks.

ASRC converts the sampling rate of a signal associated to an input clock into a signal associated to a different output clock. ASRC supports concurrent sample rate conversion of up to 10 channels of about -120dB THD+N. The sampling rate conversion of each channel is associated to a pair of incoming and outgoing sampling rates. ASRC supports up to 3 sampling rate pairs.

In the real-time audio use case, both input/output sampling rate clocks are activated. Both sampling rate clocks are directly connected to ASRC, and the ratio estimation of the input clocks to output clocks is used to perform the sample rate conversion in ASRC hardware.

The SSI1, SSI2, SSI3, S/PDIF, ESAI, and MLB audio modules can act as either a clock master or a clock slave. Multiplexors are provided on a chip level to select which of the module's clock signal, input or output, will be connected to ASRC input in dependence of the module's operational mode as it is shown in Figure 9-14 and Table 9-13. Figure 9-14 presents the multiplexor cell used for all ASRC input clocks. Table 9-13 shows the detailed clocks, multiplexor control and data bits for each ASRC input clock. For the audio blocks clocks that are directly conected to ASRC (without the multiplexor scheme) see Table 9-14.

In the non-real time streaming audio use case, the input sampling rate clock does not need to be provided. Instead the ideal-ratio value conversion is set in the ASRC interface registers. In this case only the output sampling rate clock must be provided, and the fixed ratio of the input to the output in the register is used to perform the sample rate conversion.

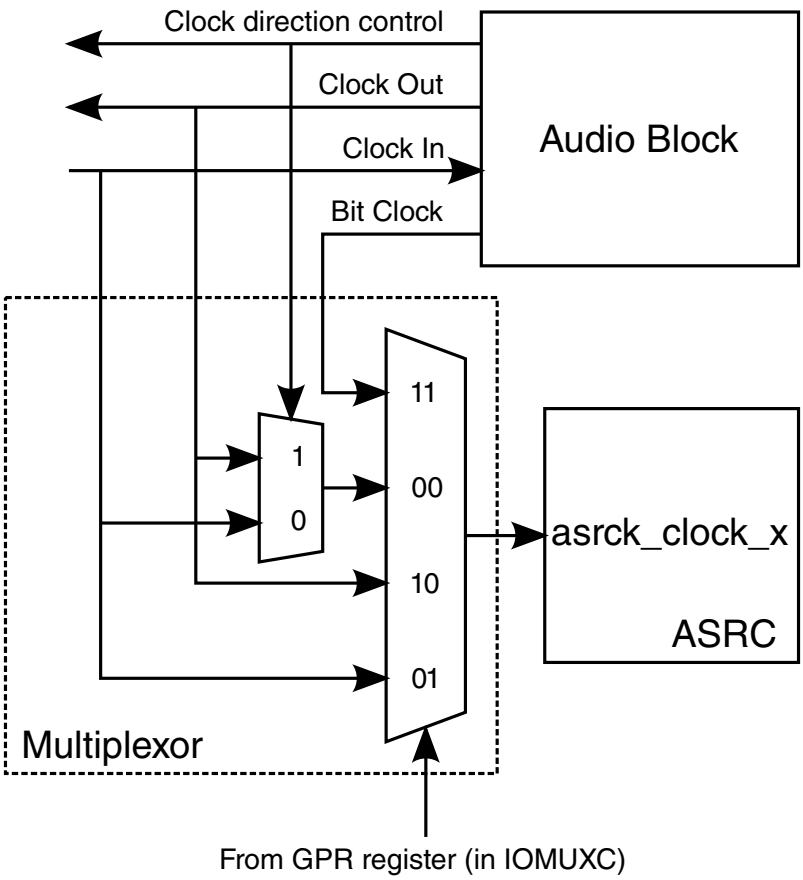


Figure 9-14. Muxing scheme template for ASRC input clocks

Table 9-13. ASRC Muxed Input Clocks

ASRC Clock Input	Audio block source	MUX2:1 Control	MUX2:1 Input 0	MUX2:1 Input 1	MUX4:1 Control	MUX4:1 Input 00	MUX4:1 Input 01	MUX4:1 Input 10	MUX4:1 Input 11
asrck_clock_1	SSI-1 (Rx)	SRCR[RXDIR][1]	External SRCK	InternalSRCK	GPR0[17:16][2]	MUX2:1 output	External SRCK	InternalSRCK	RX bit clock
asrck_clock_9	SSI-1 (Tx)	STCR[TXDIR]	External STCK	InternalSTCK	GPR0[17:16]	MUX2:1 output	External STCK	InternalSTCK	TX bit clock
asrck_clock_2	SSI-2 (Rx)	SRCR[RXDIR]	External SRCK	InternalSRCK	GPR0[19:18]	MUX2:1 output	External SRCK	InternalSRCK	RX bit clock
asrck_clock_a	SSI-2 (Tx)	STCR[TXDIR]	External STCK	InternalSTCK	GPR0[19:18]	MUX2:1 output	External STCK	InternalSTCK	TX bit clock
asrck_clock_3	SSI-3 (Rx)	SRCR[RXDIR]	External SRCK	InternalSRCK	GPR0[21:20]	MUX2:1 output	External SRCK	InternalSRCK	RX bit clock
asrck_clock_b	SSI-3 (Tx)	STCR[TXDIR]	External STCK	InternalSTCK	GPR0[21:20]	MUX2:1 output	External STCK	InternalSTCK	TX bit clock
asrck_clock_0	ESAI (Rx)	RCCR[RCKD]	External SCKR	InternalSCKR	GPR0[23:22]	MUX2:1 output	External SCKR	InternalSCKR	N/A[3]
asrck_clock_8	ESAI (Tx)	TCCR[TCKD]	External SCKT	InternalSCKT	GPR0[23:22]	MUX2:1 output	External SCKT	InternalSCKT	N/A

1. Register[bit] format, SRCR is the regsiter name in the block specified in the "Audio block source" column, while RXDIR is name field/bit in the register.
2. GPR0 is the General Register 0 in the IOMUX controller. See IOMUXC chapter for more details.
3. Not in used, the clock value is "0".

Table 9-14. ASRC Direct Clocks

ASRC Clock Input	Block driving clock source	Block output clock
asrck_clock_4	SPDIF (Rx)	SPDIF Rx Clock
asrck_clock_c	SPDIF (Tx)	SPDIF Tx Clock
asrck_clock_6	External (from PAD)	Via IOMUX (from PAD) KEY_ROW3 (ALT1) [default] GPIO_0 (ALT3) GPIO_18 (ALT4)
asrck_clock_7	Reserved	Reserved[1]
asrck_clock_d	CCM	SPDIF1 clock root

1. Not in used, the clock value is "0".

Chapter 10
Clock and Power Management

10.1 Introduction

This chapter describes the Clock and Power Management architecture of the SoC.

The chip targets applications where low power consumption, long battery life, always-on and instant-on capabilities are paramount, and where there is no need for active cooling. To achieve these capabilities, the primary focus of the chip's design is reducing current consumption as much as possible, while simultaneously enabling the maximum level of peak performance and a balanced level of sustained performance for target applications. To achieve this, the chip architecture uses a wide range of power-management techniques and their combinations for maximum system design flexibility.

This chapter contains information about:

- Structural components of the power and clock management systems of the chip
- Power, clock and thermal management techniques supported by the chip

All given numerical values are typical or examples. For accurate values one should refer to the datasheet.

10.2 Device Power Management Architecture Components

To provide a clean and versatile architecture supporting a wide range of power-management techniques, clocks, and power rails are managed resources.

For each rail, two levels of management are defined: the first level is centralized or SoC-level resource management, and the second is a local or "module level" resource management.

The high level architectural view of the clock, power and thermal management system of the chip is presented in the figure below.

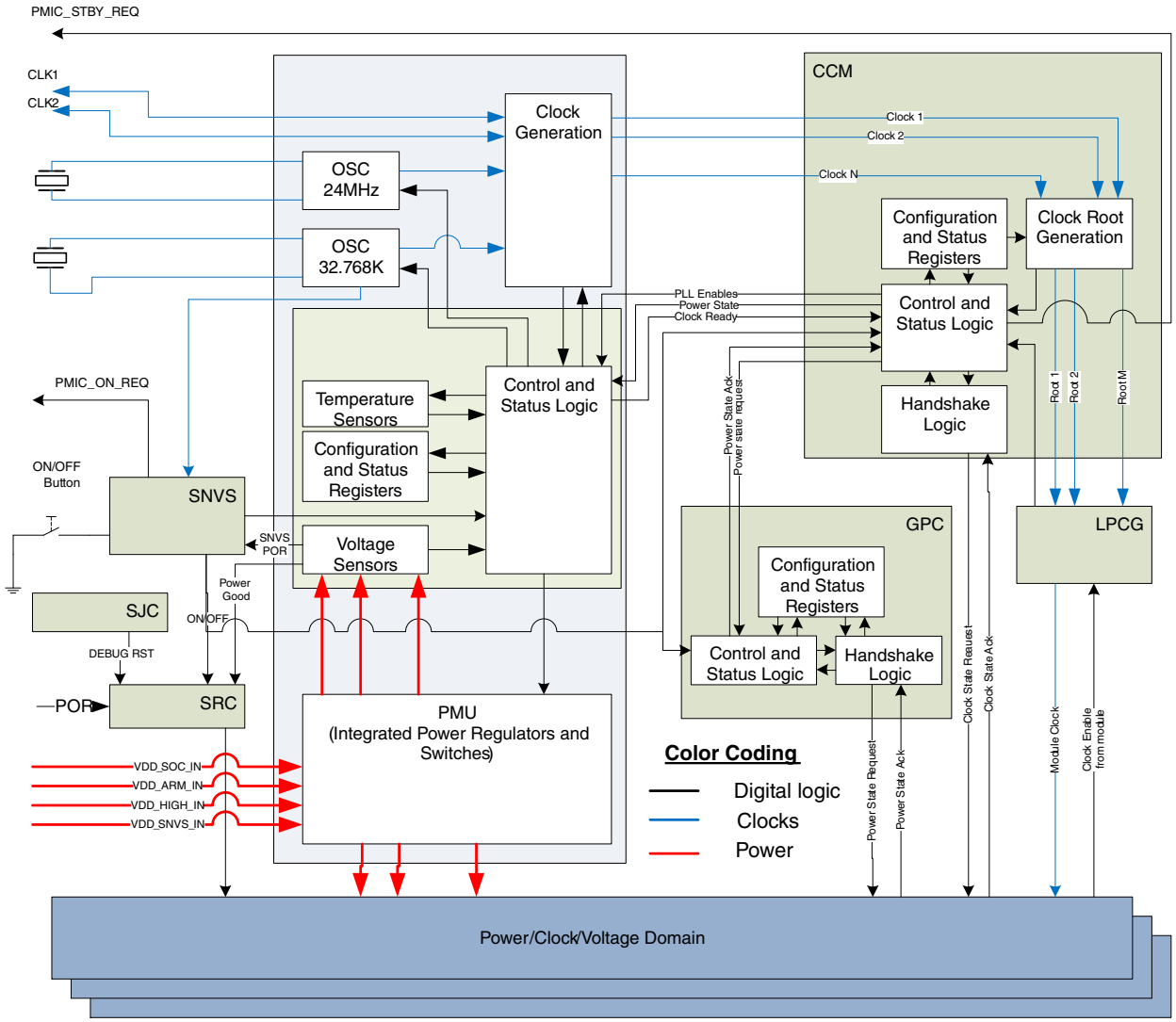


Figure 10-1. Power and clock management framework

10.2.1 Centralized components of clock generation and management

Centralized components of the clock generation and management sub-system are implemented in the following blocks:

- CCM (Clock Control Module): The CCM module provides control for primary (source level) and secondary (root level) clock generation, division, distribution, synchronization, and coarse-level gating. See Clock Controller Module (CCM)) for

information on the CCM architecture, functional description and programming model.
- LPCG (Low Power Clock Gating): This module distributes the clocks to all blocks in the SoC and handles block level software-controllable and automated clock gating. See Clock Controller Module (CCM) for information on the LPCG architecture and functional description.

10.2.2 Centralized components of power generation, distribution and management

Centralized components of the power generation, distribution and management sub-system are implemented in the following blocks:

- Power Management Unit (PMU). See Power Management Unit (PMU) for information on the PMU architecture, functional description and programming model.
- General Power Controller (GPC). See General Power Controller (GPC) for information on the GPC architecture, functional description and programming model.

10.2.3 Reset generation and distribution system

Power and clock management are accompanied with an appropriate reset generation and distribution system, centralized functions of which are implemented by the System Reset Controller (SRC).

10.2.4 Power and clock management framework

Together, the modules listed above provide enhanced power-management features with centralized control for the clock, reset, and power-management signals on the SoC.

The centralized management framework defines the managed components of the power-management architecture. These components are called the clock, power, and voltage domains.

NOTE

A domain is a group of modules or functional blocks that share a common resource entity (for example, common clock root, common power source, or a common power switch). The software component managing shared resources should take

into account the joint constraints of all the modules belonging to that resource domain.

10.3 Clock Management

10.3.1 Centralized components of clock management system

The clock generation and management system is built around the CCM and LPCG blocks.

A high level block diagram of the clock management system in the SoC environment is shown in the following figure.

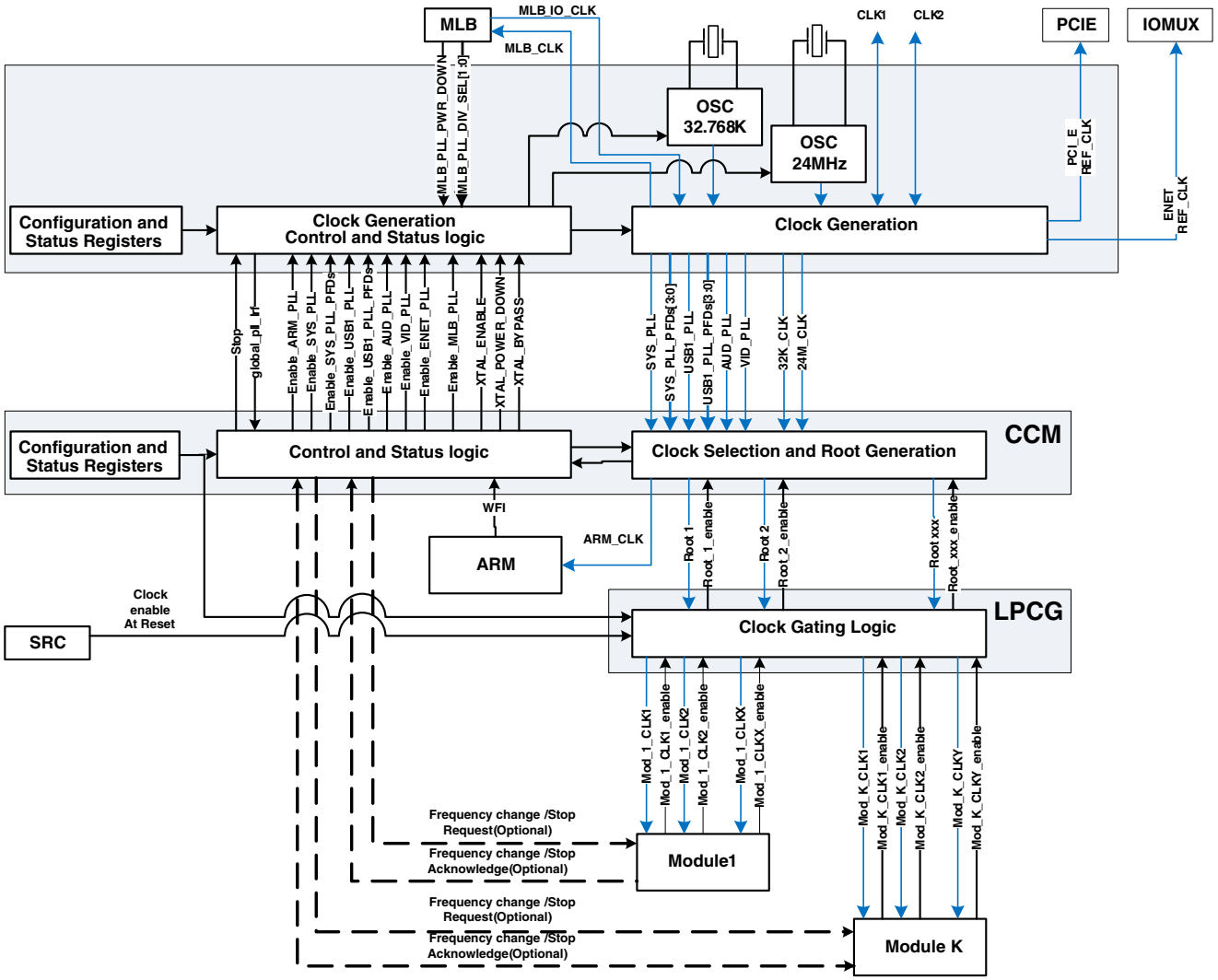


Figure 10-2. Clock Management System

A high level block diagram of the clock generation is shown in the figure below.

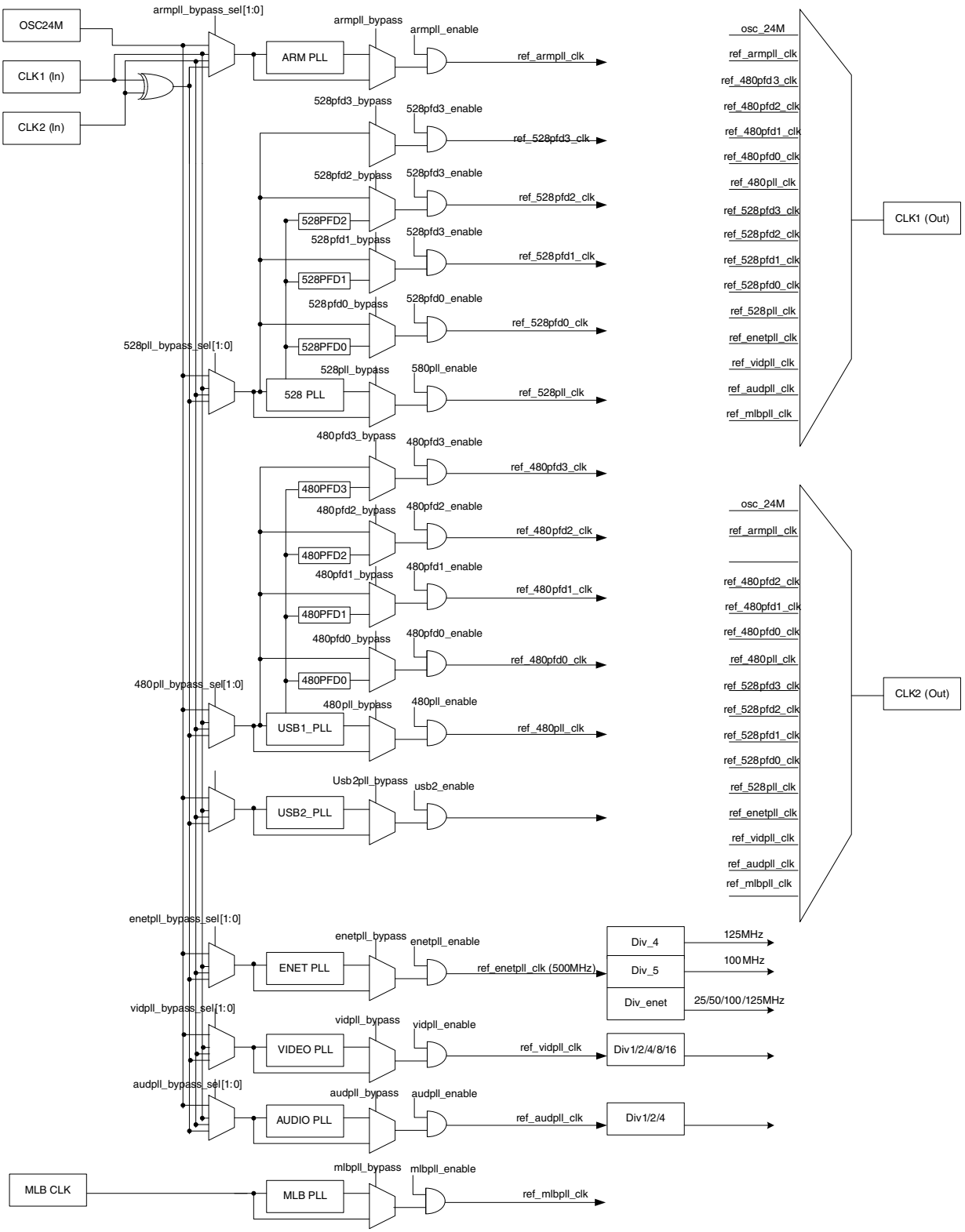


Figure 10-3. Primary Clock Generation

10.3.2 Clock generation

The clock generation section includes the components detailed in the following sections.

10.3.2.1 Crystal Oscillator (XTALOSC)

The Crystal Oscillator block is comprised of both the high frequency oscillator (typical frequency is 24 MHz) and the low frequency real time clock oscillator (typical frequency of 32.768 KHz). Each of these oscillators is implemented as a biased amplifier that, when combined with a suitable external quartz crystal and external load capacitors, implements an oscillator. See Crystal Oscillator (XTALOSC) for details of the XTALOSC block.

10.3.2.2 Low Voltage Differential Signaling (LVDS) I/O ports

There is two LVDS I/O ports used for clock generation. The low jitter differential I/O portsare provided to input and output clocks. It can take input clocks from outside of the SoC and provide them to the PLLs or to the other modules, or it can take the outputs of the PLLs and provide them outside of the SoC as a functional or reference clock.

10.3.2.3 PLLs

Eight PLLs are included in the clock generation section. Two of these PLLs are each equipped with four Phase Fractional Dividers (PFDs) in order to generate additional frequencies.

NOTE

Each PFD works independently by interpolating the VCO of the PLL to which it is connected. It effectively takes the PLL VCO frequency and produces 18/N x Fvco at its output where N ranges from 12 to 35. PFD is a completely digital design with no analog components or feedback loops. The frequency switch time is much faster than a PLL because keeping the base PLL locked and changing the integer N only changes the logical combination of the interpolated outputs of the VCO. Note that the PFD not only enables faster frequency changes than a PLL, but also allows the configuration to be safely changed "on-the-fly" without going through the output clock disabling/enabling process.

The eight PLLs are listed below:

- PLL1 (also referred to as ARM_PLL) - This is the PLL clocking the Arm core complex. It is a programmable integer frequency multiplier capable of output frequency of up to 1.3 GHz (may exceed chip capabilities, see datasheet for more information).
- PLL2 (also referred to as System_PLL or 528_PLL) - PLL2 generally runs at a fixed multiplier of 22, producing 528 MHz output frequency with 24 MHz reference from XTALOSC. It has the capability to spread spectrum the generated signal. Besides the main output, this PLL drives four PFDs (528_PFD0...528_PFD3). The main PLL output and its PFD outputs are used as inputs for many clock roots. These do not require exact/constant frequency and can be changed as a part of dynamic frequency scaling procedure and/or can be spread-spectrum modulated. Typically, this PLL is a clock source for internal system buses, internal processing logic, DDR interface, NAND/NOR interface modules, etc.
- PLL3 (also referred to as USB1_PLL or 480 PLL) - PLL3 is used in conjunction with the first instance of USB PHY (USB0 PHY, also known as OTG PHY). This PLL also drives four PFDs (480_PFD0...480_PFD3) and runs at a fixed multiplier of 20. This results in a VCO frequency of 480 MHz with a 24 MHz oscillator. The main PLL output and its PFD outputs are used as inputs for many clock roots that require constant frequency, such as UART, CAN and other serial interfaces, audio interfaces, etc.
- 480_PLL2 (also referred to as USB2_PLL) - This PLL provides clock exclusively to USB2 PHY (also known as HOST PHY). It runs at a fixed multiplier of 20, resulting in a VCO frequency of 480 MHz with a 24 MHz oscillator.
- PLL4 (also referred to as an Audio PLL) - This is a fractional multiplier PLL used for generating a low jitter and high precision audio clock with standardized audio frequencies. The PLLs oscillator frequency range is from 650 MHz to 1300 MHz, and the frequency resolution is better than 1 Hz. This clock is mainly used as a clock for serial audio interfaces and as a reference clock for external audio codecs. It is equipped with a divider on its output and can generate divided by 1, 2 or 4 from the PLL VCO frequency.
- PLL5 (also referred to as a Video PLL) - This is a fractional multiplier PLL used for generating a low jitter and high precision video clock with standardized video frequencies. The PLLs oscillator frequency range is from 650 MHz to 1300 MHz, and the frequency resolution is better than 1 Hz. This clock is mainly used as a clock for display and video interfaces. It is equipped with dividers on its output and can generate clock divided by 1, 2, 4, 8 or16 from the PLL VCO frequency
- PLL6 (also referred to as PLL_ENET) - This PLL implements a fixed 20+(5/6) multiplier. With a 24 MHz input, it has a VCO frequency of 500 MHz. This PLL is used to generate:
- 125 MHz for the PCIe serial interface and reduced gigabit ethernet interface.

- 50 or 25 MHz for the external ethernet interface.
- MLB_PLL - This PLL takes the Media Link Bus (MLB) interface clock, multiplies it up by 1, 2, or 4, and delay compensates so that the MLB data stream can be captured.

10.3.2.3.1 General PLL Control and Status Functions

Each PLLs configuration and control functions are accessible individually through its PFDs and global configuration and status registers.

Reference input clock for any of the PLLs could be selected individually by the BYPASS_CLK_SRC field of the PLL control register. See CCM Analog Memory Map/ Register Definition for more information.

Each of the PLLs could be individually configured to "Bypass", "Output disabled" and "Power Down" modes.

When configured in "Bypass" PLL pass directly its input reference clocks to the PLL output. Bypassing the PLL is done by setting the BYPASS bit in the control register. For the PLL equipped with PFDs the input reference clock is also bypassed to all PFDs outputs.

When configured in output disabled mode (ENABLE=0), the PLL's output is completely gated and there is neither a bypass clock nor PLL generated clock that propagates to PLL output. Each PLL output has an individual "Output Enable" control bit. The PFDs are gated by the ENABLE bit of their associated PLL. Each PFD does have an associated clock gate bit that can be used to turn it off individually.

When configured in "Power Down mode" most of the PLL circuitry is switched off. Neither main PLL output nor PFD outputs are available in this mode.

When the related PLL is powered up from the power down state or made to go through a relock cycle due to PLL reprogramming, it is required that the related PFDx_CLKGATE bit in CCM_ANALOG_PFD_480n or CCM_ANALOG_PFD_528n, be cycled on and off (1 to 0) after PLL lock. The PFDs can be in the clock gated state during PLL relock but must be clock un-gated only after lock is achieved. See the engineering bulletin, Configuration of Phase Fractional Dividers (EB790) at www.nxp.com for procedure details.

Individual PLL status is reflected in "PLL Lock" bits of the PLL control registers. PLL enable logic which monitors the register value change is implemented to gate off the PLL outputs during the "lock in" period.

Outputs are generated to be sent out by monitoring the individual PLL lock flags and filtering out any random initial edges.

Individual PLL Lock ready flags are first "ORED" with "enables" and then "ANDED" together to generate the global PLL lock ready flag that reflects status of all PLLs enabled in certain moment.

CCM Memory Map/Register Definition and CCM Analog Memory Map/Register Definition contains detailed descriptions of the memory mapped registers and control functions of the clock generation sub-module.

10.3.2.4 CCM

CCM includes:

- Clock root generation logic - This sub-block provides the registers that control most of the secondary clock source programming, including both the primary clock source selection and the clock dividers. The clock roots are each individual clocks to the core, system buses (AXI, AHB, IPG) and all other SoC peripherals, among those are serial clocks, baud clocks, and special functional clocks. Most of clock roots are specific per module.
- CCM, in coordination with GPC, PMU and SRC, manages the Power modes, namely RUN, WAIT and STOP modes. The gating of the peripheral clocks is programmable in RUN and WAIT modes.

CCM manages the frequency scaling procedure for:

- Arm core clock - "on the fly" without clock interruption, by either shifting between PLL sources PLL clock change or by changing the divider ratio.
- changing of the DDR memory controller clock. See MMDC handshake and Self refresh and Frequency change entry/exit for more details.
- Peripheral root clock - by using programmable divider. The division factor can change on the fly without loss of clocks.

NOTE

On-the-fly frequency changing for synchronous interfaces like serial audio interfaces, video and display interfaces, or general purpose serial interfaces (e.g. UART, CAN) may cause synchronization loss and should not be done.

10.3.2.5 Low Power Clock Gating unit (LPCG)

The LPCG block receives the root clocks from CCM and splits them to clock branches for each block. The clock branches are individually gated clocks.

The enables for those gates can come from three sources:

- Clock enable signal from CCM - This signal is generated depending on the power mode the system is in. For each power mode, it is defined in the software using the configuration of the CGR bits in CCM.
- Clock enable signal from the block - This signal is generated by the block based on its internal logic. Not every enable signal from the block is used. Each clock enable signal from the block can be overridden based on the programmable bit in CCM.
- Clock enable signal from the reset controller (SRC) - This signal will enable the clock during the reset procedure.

10.3.3 Peripheral components of clock management system

The figure found here shows the clock interface of a functional module in the system.

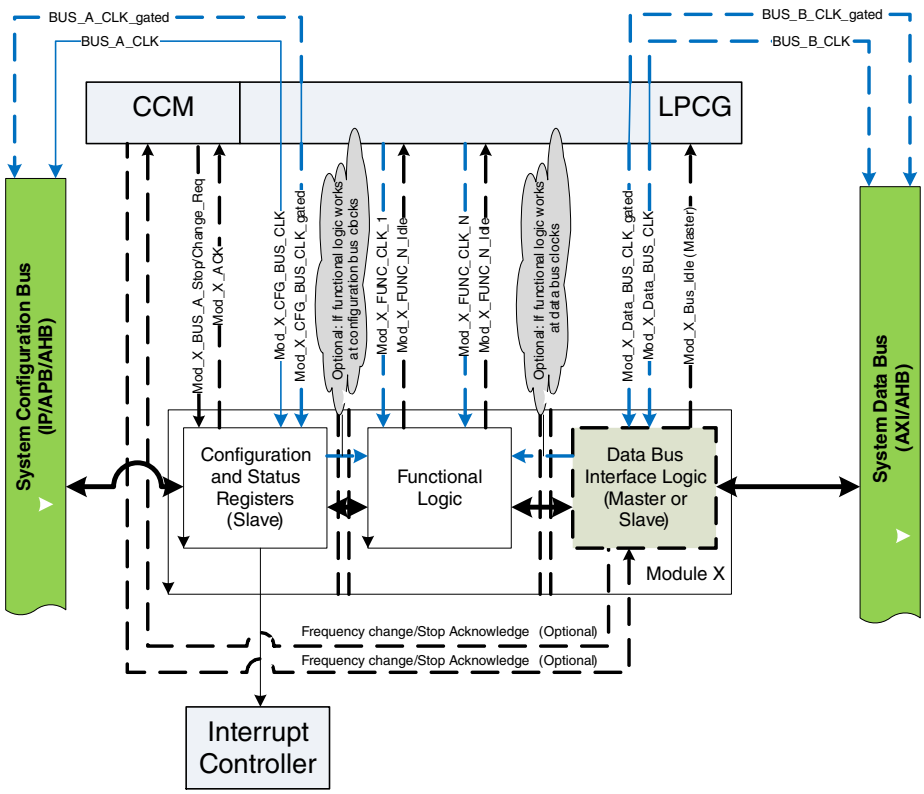


Figure 10-4. Clock interface of the functional module in the system

10.3.3.1 Interface and functional clock

Each block within the SoC has specific clock input characteristic requirements. Based on the characteristics of the clocks delivered to modules, the clocks are divided into two categories: bus interface clocks and functional clocks.

The bus interface clocks have the following characteristics:

- They ensure proper communication between any block/subsystem and the system buses.
- In most cases, they supply the system interface and configuration registers of the block.
- A typical block has one system bus clock, but blocks with multiple interface clocks may also exist (that is, when a block is connected to multiple buses).
- The bus interface clocks are always fed by the outputs of the CCM/LPCG.
- Clock management for this type of clock is always implemented at the system level because it requires coordinated clock management between the block and system buses.

Functional clocks have the following characteristics:

- They supply the functional part of a block or a subsystem.
- Typically, these clocks are completely asynchronous and independent from the bus interface clock of the same block.
- A block can have one or more functional clocks. Some functional clocks are mandatory, while others are optional for its functioning. A block needs its mandatory clock(s) to be operational. The optional clocks are used for specific features and can be shut down without stopping the block activity (in the case of IPU, that could be the clock for a display interface or a camera sensor interface).
- The functional clocks are fed either by a CCM/LPCG block functional clock output, or by some other clock source, such as a clock output of another block or an external signal coming from IOMUX.

10.3.3.2 Block level clock management

Each block in the system may also have specific clock requirements. Certain module clocks must be active when operating in some specific modes, or may be gated in some others. Generally, the activation and gating of the module clocks are managed by LPCG. Hence, the LPCG block must be programmed properly and, in case of hardware controllable clock gating, peripheral module should provide signals indicating when to activate and when to gate the module clocks.

The LPCG block differentiates the clock-management behavior for device modules based on whether the block can initiate transactions on the device interconnect (called master module), or if it cannot initiate transactions and only responds to the transactions initiated by the master (called slave module). Thus, two hardware-based clock-management protocols are used:

- Master protocol - Clock-management protocol between the CCM/LPCG and blocks that can be bus master
- Slave protocol - Clock-management protocol between the CCM/LPCG and slave modules

10.3.3.2.1 Master clock protocol

This protocol is used to indicate that a master module is ready to initiate a transaction on the device interconnect and requests specific (both functional and interface) clocks. The CCM/LPCG block ensures that the required clocks are active when the master module requests that the CCM/LPCG enable them. The module is said to be functional after the required clocks are activated.

Similarly, when the master module no longer requires the clocks, it informs the LPCG/CCM block and the LPCG/CCM can then gate the clocks to the module and all the clock precedents that are not used by other blocks. The master module is then said to be in clock-gated or partially clock gated mode.

Examples of modules supporting master clock protocol areGPU3D, VPU,GPU2D, VDOA and USDHC. Please see details in chapters describing these modules and in the CCM enable override register (CCM_CMEOR).

10.3.3.2.2 Slave clock protocol

This hardware protocol allows CCM to control the state of a slave module. CCM informs the slave module, through assertion of a stop/change request, when its clocks (both interface and functional) can be changed or gated. The slave acknowledges the request and CCM is then allowed to gate or change the clocks to the block.

Similarly, a clock-gated slave module may need to be woken up because of some event or a service request from a master module. In this situation, CCM enables the clocks to the module and then de-asserts the stop request to signal the module to wake up.

Examples of modules supporting slave clock protocol are CAN,EPIT and GPT. Please see details in chapters describing these modules and in the CCM Module Enable Overide Register (CCM_CMEOR). See CCM Memory Map/Register Definition for more details.

The protocol in both "master" and "slave" cases is completely hardware-controlled, but software should configure the clock management behavior for the module in two places: in the CCM registers associated with the block and in the block configuration registers.

10.3.3.3 Clock Domain(s)

A clock domain is a group of blocks fed by clock signals controlled by the same clock controls in CCM. By gating the clocks in a clock domain, the clocks to all the blocks belonging to that clock domain can be gated/activated, either by software control or by hardware control associated with block activity. Thus, a clock domain allows efficient control of the dynamic power consumption of the domain.

The device is partitioned into multiple clock domains and each clock domain is controlled by an associated group of clock gating cells within the LPCG block. This allows the CCM/LPCG to individually activate and gate each clock domain of the system.

Examples of clock domains are: Main AXI bus clock domain, VPU clock domain, GPU3D clock domain, etc.

10.3.3.4 Domain level clock management

The domain clock manager can automatically (based on hardware conditions) and manage the bus interface clocks within the clock domain. The functional clocks within the clock domain are managed through software settings.

10.3.3.5 Domain dependencies

A domain dependency is a hierarchical relationship between two clock domains. Clock domain "X" is said to depend on a clock domain "Y" when a block in clock domain "Y" provides services (or even just a clock) to a block in clock domain "X". As a result, clock domain "Y" must be active whenever clock domain "X" is active.

The dependency between two clock domains may also exist if one clock domain serves to ensure communication between two blocks (for example, the clock domain of the device interconnect).

10.4 Power management

10.4.1 Centralized Components of Power Management System

The power generation and management system is built around the PMU and GPC blocks.

A high level block diagram of the power management system in the SoC environment is shown in the figure below.

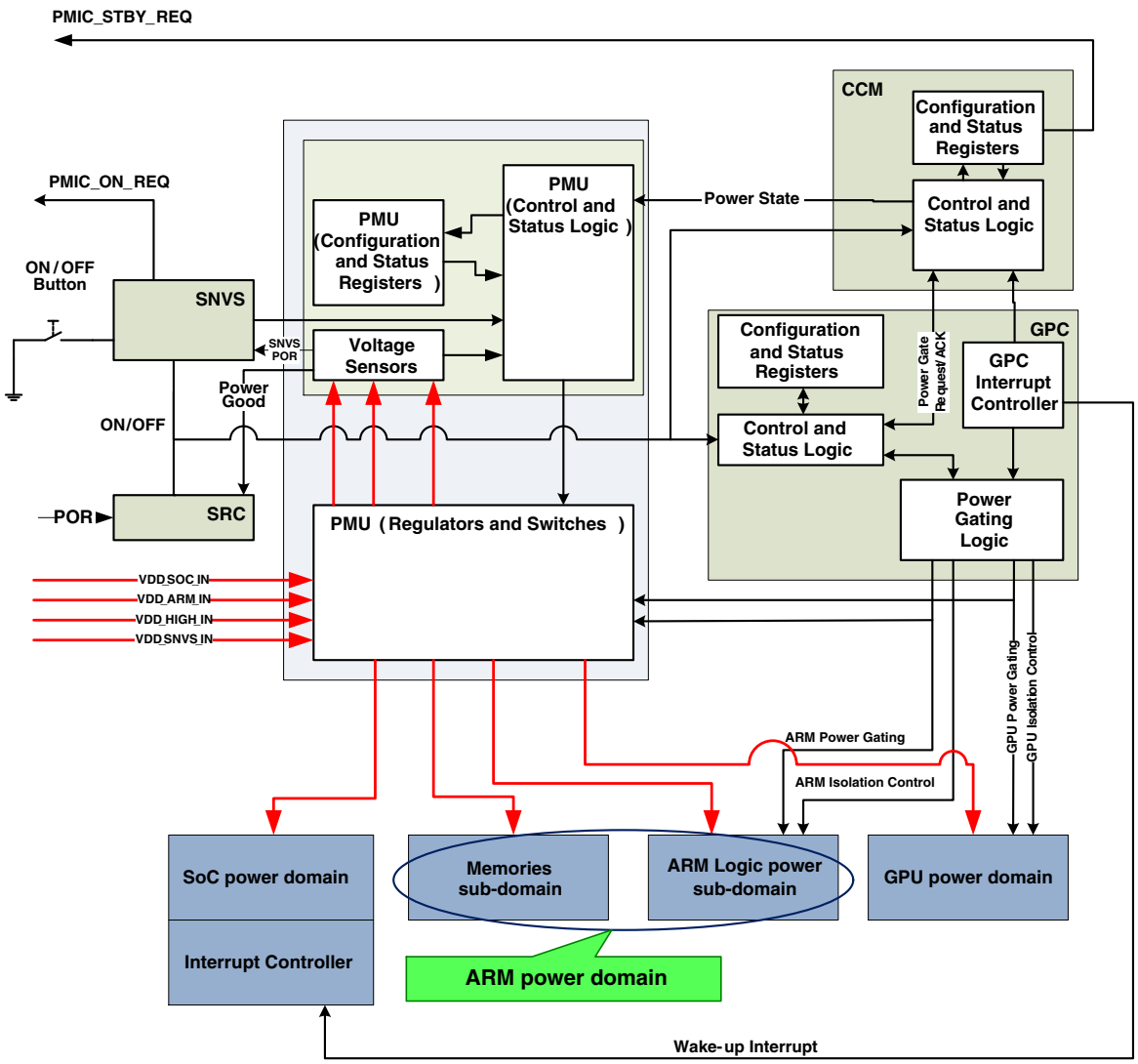


Figure 10-5. Power Management System

10.4.1.1 Integrated PMU

The first component of the power management system, referred to as the integrated PMU, is designed to simplify the external power interface.

It consists of a set of secondary power supplies that enable SoC operations from just two or three primary supplies. The high level block diagram of the power tree, utilizing the integrated PMU, is shown below.

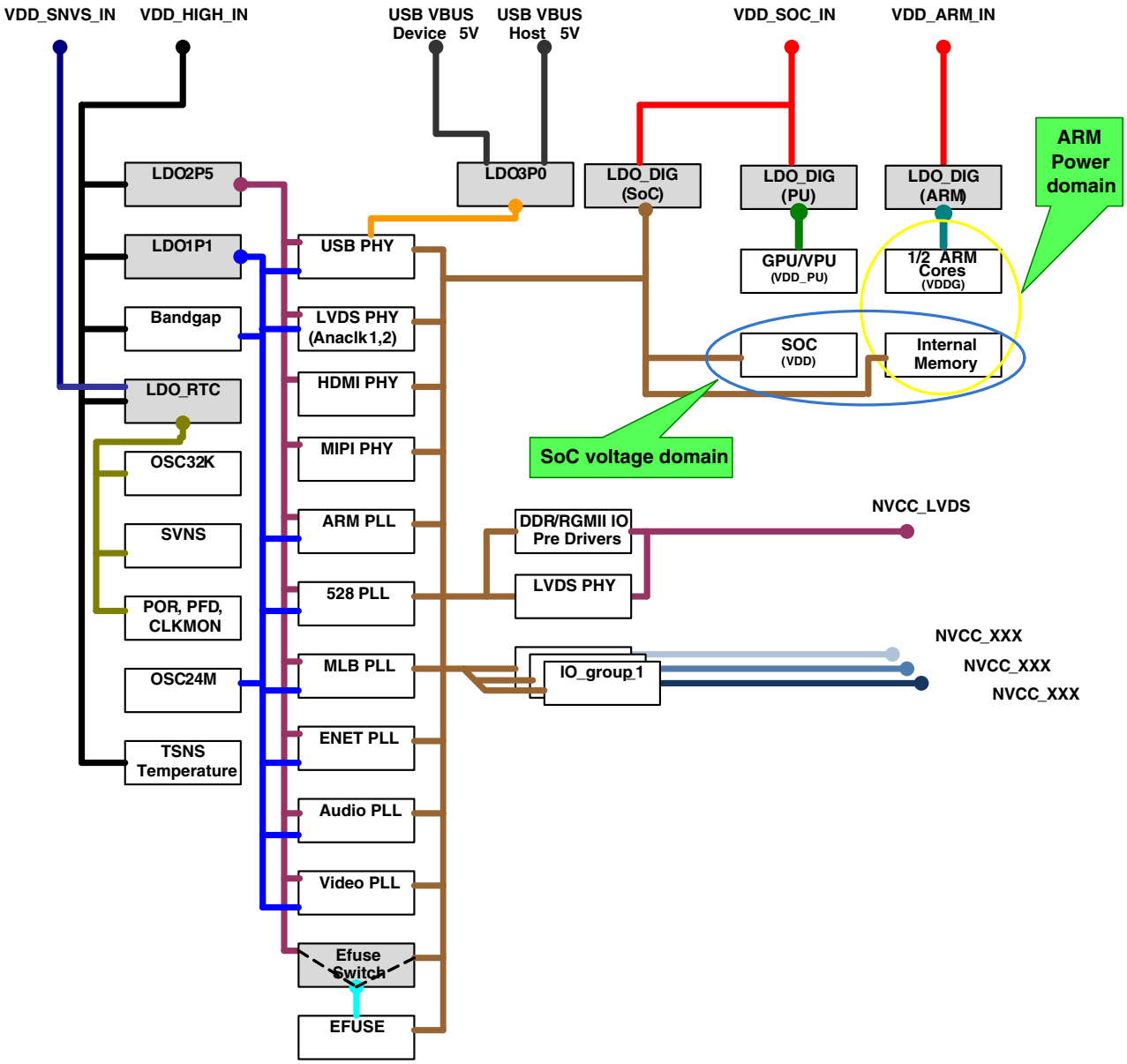


Figure 10-6. i.MX 6Solo/6DualLite Power Tree

The integrated PMU includes the following components:

- Three Digital LDO regulators
- Two Analog LDO regulators
- USB LDO
- SNVS regulator
- Reverse well biasing

See Power Management Unit (PMU) for further details on integrated PMU functional description and programmability.

10.4.1.1.1 Digital LDO Regulators

The integrated PMU includes three digital LDO regulators: LDO_ARM, LDO_PU, and LDO_SOC. These regulators provide power to the ARM_Core power domain, the combined VPU, IPU, and GPU power domain, and the rest of the SoC logic (except always-ON SNVS domain).

NOTE

The name "digital" only refers to the type of load. It is not related to the LDO design or feature set.

The digital LDO regulators can operate in the following modes:

- Internal Bypass - The regulation pass device (FET) is switched fully on, passing the external input voltage to the load unaltered. The analog part of the regulator is powered down in this state, removing any loss other than the IR drop through the power grid and the FET. Be aware that a period of time (see datasheet) is required to switch from the internal digital bypass mode to the analog regulation mode. Typically it takes less than 100us. Please refer to PMU for further details on bypass and power gate configuration.
- External Bypass - The input and output of the regulator are shorted externally to the SoC. If operating in this configuration, enable the internal bypass early in the startup sequence before attempting high frequency/high power operation. Be aware that internal power gating is not available in this mode.
- Power Gate - The regulation FET is switched fully off, limiting the current draw from the supply. The analog part of the regulator is powered down, limiting the power consumption. The output voltage will fall to a level where the residual leakage of the power FET balances with the leakage of the load. Power gating is applicable to Arm and PU power domains.
- Analog regulation mode - The regulation FET is controlled such that the output voltage of the regulator equals the programmed target voltage. The target voltage is fully programmable in 25mV steps.

These modes allow the regulators to implement voltage scaling and power gating, and allow bypass when an external high power efficient regulator is used as a direct source for some of the SoC loads.

These digital regulators also feature brownout detection, which is helpful to sense when supplies are starting to collapse. Note that the core will be interrupted on a brownout. Please see details in Miscellaneous Control Register (PMU_MISC2*n*).

For further details of LDO programming and configuration please refer to Digital Regulator Core Register (PMU_REG_CORE*n*).

The power management system is built under assumption that in typical applications the single (and simple) shared power supply will be used for Arm core domain and SoC domain. The combined load gains some efficiency, especially in low power modes and saves BoM significantly.

The DVFS in a typical cost/complexity optimized application is considered by mean of internal LDO. In"full speed" modes LDO bypass is considered in both domains. The dynamic voltage scaling to low load workpoints for Arm domain is implemented by programming associated LDO.

In highly power-optimized systems, it is possible to use multiple external DCDC buck converters and bypass internal LDO for high power domains. The obvious trade-off is in the increased complexity of the external power supply components and the associated increase in the BoM and board design complexity.

10.4.1.1.2 Analog LDO regulators

There are two analog LDO regulators used for general system purposes:

- LDO_1P1 - The LDO_1P1 (VDD_HIGH_IN, NVCC_PLL) linearly regulates down a higher supply voltage (2.8V-3.3V) to produce a nominal 1.1V output voltage. This regulator supplies digital portions of USB PHYs, PLLs, and the internal 24 MHz oscillator.

10.4.1.1.3 USB LDO

The USB_LDO linearly regulates down the USB VBUS input voltages (typically 5V) to produce a nominal 3.0V output voltage. This regulator has a built in power-mux that allows the user to run the regulator from either one of the VBUS supplies when both are present. If only one of the VBUS voltages is present, the regulator automatically selects that supply. Current limit is also included to help the system keep the in-rush current within limits as required in USB 2.0 specification. This regulator supplies only low speed and full speed transceivers of USB PHYs.

10.4.1.1.4 SNVS regulator

The SNVS regulator takes the SNVS_IN supply and generates the SNVS_CAP supply, which in turn powers the real time clock and low power section of the SNVS modules. If VDDHIGH_IN is present, then the SNVS_IN supply is internally shorted to the VDDHIGH_IN supply to allow coin cell recharging if necessary.

10.4.1.1.5 Reverse well biasing

The reverse well biasing module on the SoC consists of a self-clocked/self-regulating charge-pump circuit, used to generate a negative bias voltage for the floating PWELL, and a low-power regulator. The low-power regulator is used to generate a positive bias voltage for the NWELL of the digital logic cells on the SOC power domain. Static leakage reduction can be achieved during SoC low-power modes through the use of these reverse well bias voltages. Please refer to the CCM_CLPCR register in CCM Memory Map/Register Definition and PMU_MISC0 in PMU Memory Map/Register Definition for details about well bias control and Static " for functional description of the reverse well biasing.

10.4.1.2 GPC - General Power Controller

The GPC block includes the sub-blocks listed here.

- Power Gating Controller (PGC) - This sub-block of GPC has the following functions:
  - Provides the user with the ability to switch off power to a target subsystem.
  - Generates power-up and power-down control sequences. This includes interaction with CCM/LPCG and SRC, and control for clock and reset generation for power domains affected by power gating.
  - Provides programmable registers that adjust the timing of the power control signals.
- Controls the CPU power domain and the combined GPU/VPU power domain.
- Wake-up interrupt controller - This controller initiates the system wake-up from low power modes when only low frequency real time clock remains active, and thus the Generic Interrupt Controller (GIC) can not handle synchronous interrupt signals. Additional features are as follows:
  - Supports up to 128 interrupts
  - Provides an option to mask/unmask each interrupt
  - Detects interrupts and generates the wake up signal

See General Power Controller (GPC) for further details on GPC, its sub-blocks, and information on its functional description and programmability.

10.4.1.3 SRC - System reset Controller

The reset controller is responsible for the generation of all reset signals and boot configuration decoding.

It determines the source and the type of reset, such as POR, WARM, and COLD, and performs the necessary reset signal qualifications. SRC is capable of generating reset sequences in the following conditions:

- in interaction with external PMIC, based on external POR_B signal and "power ready" signals generated by the integrated PMU
- or in interaction with the integrated PMU only, based on its "power ready" signal.

Based on the type of reset, the reset logic generates the reset sequence for either the entire SoC or for the blocks that are power-gated.

See System Reset Controller (SRC) for further details on SRC functional description and programmability.

10.4.1.4 Power domain(s)

A power domain is a group of blocks or sub-blocks fed by power sources controlled by the same power controls in GPC.

Some power domains can be split into a logic sub-domain and a memory sub-domain. The memory sub-domain in such case may contain two entities:

- Memory array(s) - Powered by a dedicated voltage rail enabling memory retention while core is OFF.
- Memory interface logic - Powered by the same voltage source as the logic sub-domain of the power domain.

Signals crossing power domain boundaries or sub-domain boundaries are passed through proper isolation and/or level-shifting cells to ensure robust operations of the SoC when some of domains are power gated or working at a reduced voltage.

The following figure shows the power domain interface within the system.

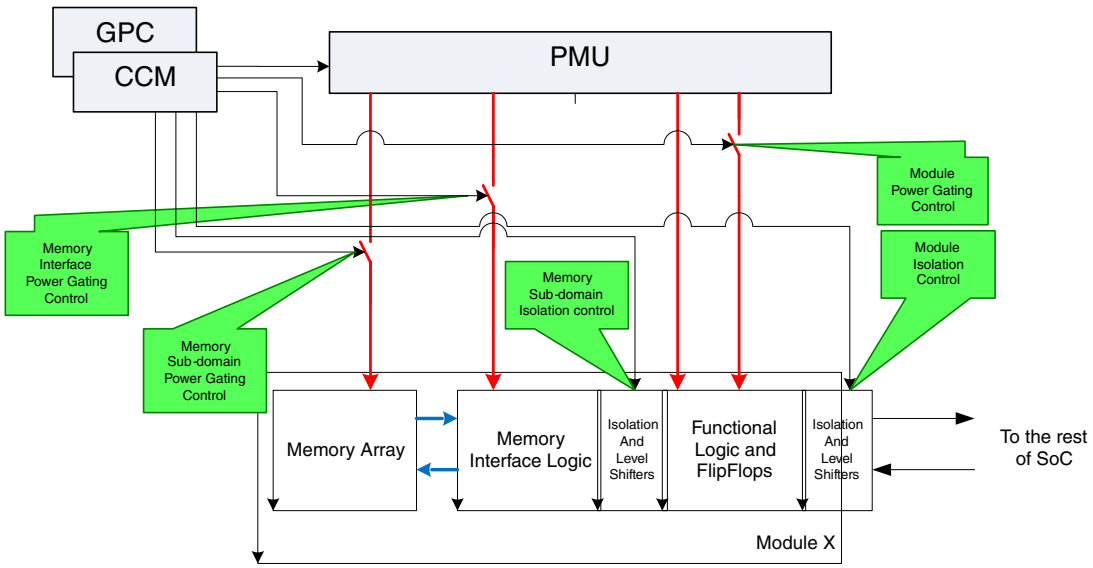


Figure 10-7. Power domain interface

10.4.1.4.1 Power distribution

The power distribution tree is comprised of multiple power domains. The main power domains are:

- Arm - The Arm domain contains the Arm Core platform (except for memory arrays and interface logic). This domain can be supplied either from an integrated power supply or from an external controllable regulator, preferably high efficiency DCDC converter.
- Arm Memory array - Memory arrays are connected to a separate and dedicated power domain (separated from the Main logic and Arm domain). In normal operation mode (functional mode), the memory arrays domain voltage level should be kept equal to (same as) the rest of the core logic domains (Main, Arm). Please refer to the Datasheet for further information about voltage level difference between domains allowed in different power modes.
- Combined GPU and VPU domain - The combined GPU/VPU domain contains all GPU3D and GPU2D engines and VPU engine.
- SNVS/RTC low power domain - The SRTC domain contains only counter, comparator and compared data of the on-chip RTC. This domain should be supplied from an external single cell LiION battery and/or an external pre-regulated power supply.
- Analog domain - The analog domain contains the PLLs, LDOs and USB PHY. The domain supplies should be constant to allow continuous clock during any dynamic voltage scaling techniques. The digital supply should be provided from an internal

regulator, and can be combined with the memory array supply. The analog supply should be provided from internal low noise regulator.
- Main SoC logic - The main SoC logic domain contains the rest of the logic of the SoC.

From a DVFS and Power Gating standpoint, the following digital logic domains are affected:

- Cortex-A9 Core Platform - DVFS and power gating.
- Arm Cortex-A9 memories - Power gating only.
- GPU3D and VPU - Power gating only.

See table below for details of the system power domains layout and dependencies.

10.4.1.4.2 Domain Memory and domain logic state retention in case of Power Gating

The following is the list of relevant memories and logic domains with the description of their state-retention support:

- Cortex-A9 Core Platform is sub-divided into three sub-domains listed below:
- Cortex-A9 Core Platform logic: The software state retention for all logic is implemented in this domain. That means that the content of relevant registers should be be stored in some memory retaining its state (L2 cache for example) while the logic domain is power-gated. Details on how to implement the software retention can be found in the Cortex-A9 Core Platform TRM.

- GPU3D, GPU2D and VPU: These three modules can be power gated (together) independently of Cortex-A9 power gating, because it resides on a separate power domain. State retention is supported neither for GPU3D, GPU2D and VPU logic nor for their internal memories.
- ANALOG PHY IPs - PCIe and HDMI - hardware state-retention since its supplies are driven by non-gated supplies.
- SoC - hardware state-retention in Standby mode. Reverse Well Biasing is applicable for this domain.
- SNVS_LP - hardware state-retention even when SoC supplies are removed.

10.4.1.4.3 Power Gating Domain Management

The following bullets provide the sequence required for power-gating the relevant power-domains:

10.4.1.4.3.1 Arm Core Platform

1. Copy through software all the Core configuration registers to a powered-on memory
2. Configure the GPC/PGC CPU registers in PGC Memory Map/Register Definition as follows to power-down the core on the next "WFI" instruction:
   - Configure the GPC/PGC PGC_CPU_PDNSCR Register ISO and ISO2SW bits. The ISO field determines the delay between the power-down request and enabling the platform isolation. The ISO2SW field determines the delay between the platform isolation and the actual power-off switch to the supplies.
   - Configure the GPC/PGC PGC_CPU_PUPSCR Register SW and SW2ISO bits. The SW field determines the delay between the power-up request and the actual power-up of the supplies. The SW2ISO field determines the delay between asserting power toggle and negating platform isolation.
   - Configure the GPC PGC PGC_CPU_CTRL PCR bit to allow the power down of the platform
   - Arm Core Platform should execute a "WFI" instruction.

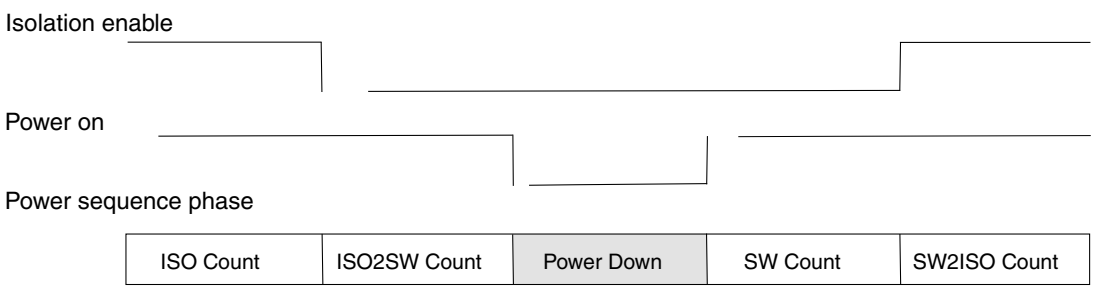


Figure 10-8. Arm Core Platform isolation and power on switch flow

10.4.1.4.3.2 GPU3D, GPU2D and VPU

1. Configure the CCM CGR bits (CCM Memory Map/Register Definition) to disable the GPU3D, GPU2D and VPU clocks.
2. Configure the GPC/PGC Registers (GPC Memory Map/Register Definition) as follows to power-down isolate the GPU3D, GPU2D and VPU logic from the rest of the SoC logic :

Configure the GPC/PGC PDNSCR Register ISO bits. These bits determine the delay between the power-down request to enabling the LDO domain isolation.

Configure the GPC/PGC PUPSCR Register SW2ISO bits. These bits determine the power-up request to the LDO domain isolation disabling.

Configure the GPC/PGC CTRL[PCR] bit to allow the power down of the block.

Configure the GPC/PGC GPC_CNTR to power down GPU3D, GPU2D and VPU

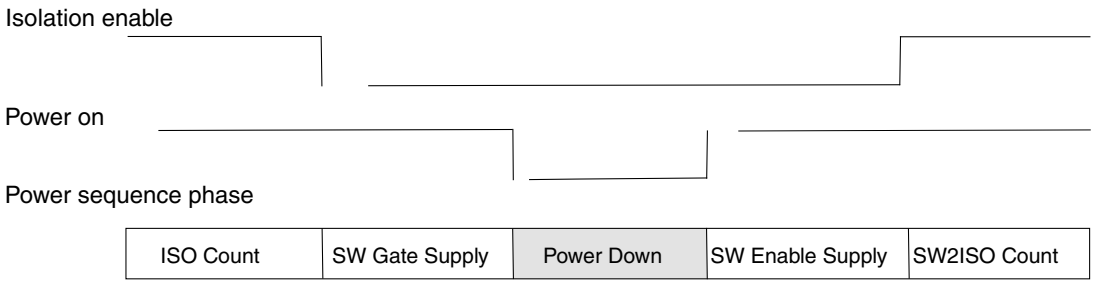


Figure 10-9. GPUs and VPU isolation and power on switch flow

10.4.1.4.3.3 SoC

For additional power reduction it is possible to do the following:

- Power-down the internal oscillator by configuring the following bits CCM_CCR[COSC_EN] (CCM Memory Map/Register Definition). This can be done only in case there is no dependency on 24 MHz XTAL for wake-up.
- Enable reverse well biasing by configuring the CCM_CLPCR[WB_PER_AT_LPM] bit (CCM Memory Map/Register Definition).
- It is possible to turn off and turn on the PMIC supplies to the SoC even when the SoC supplies are off. Since SNVS_LP is powered through an "always on" supply, configuring the SNVS_LP DP_EN to "1" allows changing the PMIC_ON_REQ pad (SoC on/off supply indication to the PMIC) through the ONOFF pad.

10.4.1.4.4 Power Gating domain dependencies

There are 3 power domains that need to be isolated in different power-down cases:

- Cortex-A9 Core Platform - Isolation needs to be enabled before power-down. This is taken care of automatically after CCM and PGC are configured and the Cortex-A9 Core Platform executes the "WFI" instruction.
- GPU3D, GPU2D and VPU - Isolation needs to be enabled before power-down. This should be taken care through software configuration of the PGC.
- SNVS_LP - Different from the 2 domains above, the SNVS_LP isolation isolates the signals coming from the SoC to the SNVS_LP. This is required for saving the contents of the SNVS_LP (such as the real-time clock). The isolation is activated in 2 ways:
  - Automatically through the power-fail detector in the PMU
  - Through software configuration

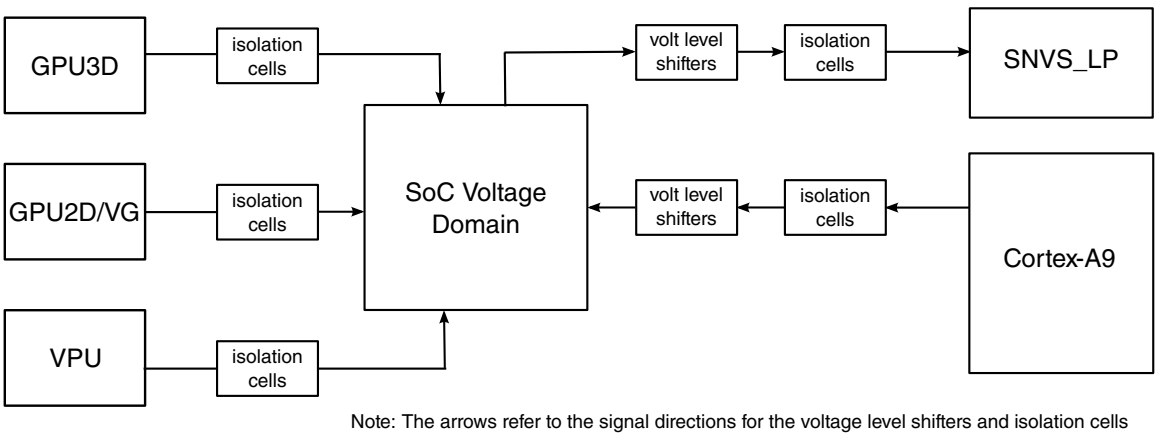


Figure 10-10. Isolation cells and Voltage level shifters placing

10.4.1.5 Voltage domains

The list found here states the different voltage domains and their scalability in regarding to power-saving in dynamic and static scenarios.

- Arm - Cortex-A9 Core Platform including L1 cache - Scalable voltage in both dynamic and static scenarios
- SoC and PU LDO Domains - Scalable voltage only in static scenarios
- ANALOG components including PCIe, LVDS andHDMI, MIPI, LVDS and PLLs - Fixed voltage
- I/O - Fixed voltage
- SNVS_LP - Fixed voltage

10.4.1.6 Voltage domain management

10.4.1.6.1 Dynamic

10.4.1.6.1.1 Voltage Scaling

A simplistic way to reduce power consumption in dynamic scenarios is to scale down the Arm, SoC and PU LDOs voltage according to the allowed voltage points and corresponding frequencies specified in datasheet.

10.4.1.6.2 Static

10.4.1.6.2.1 Standby Leakage reduction (SLR)

Standby leakage reduction is a power-management technique utilizing:

- Reduced supply voltage for relevant domains
- Reverse well-biasing in STOP, WAIT, or Deep Sleep Mode (DSM) modes

With SLR, the device switches into low-power active system modes automatically or in response to user requests during system Stop, Wait, or DSM modes (that is, in situations when no application is started and no system activity is presented).

When applying SLR, the system remains in the lowest static power mode while retaining logic and memory states. This technique trades static power consumption for wake-up latency while maintaining fast system response time suitable for most applications.

See CCM Control Register (CCM_CCR), CCM Low Power Control Register (CCM_CLPCR) and PMU Miscellaneous Register 0 (PMU_MISC0) for further details on SLR programmability options.

The following describes the flow for applying standby voltage and reverse well bias to the SoC:

- Configure the external PMIC standby voltage, refer to chip datasheet.
- Configure CCM_CCR[RBC_EN] bits to bypass and disable PMU regulators in the next Arm "WFI" execution.
- Configure CCM_CCR[REG_BYP_COUNT] bits to allow proper voltage restoration by the external PMIC when exiting standby.
- Arm Core Platform executes the "WFI" instruction that completes the software sequence putting the SoC into low power mode
- After that, the reverse well bias will be applied automatically (if enabled) with appropriate delay. Please refer to CCM_CLPCR and CCM_CCR Registers in CCM for further information on reverse bias enabling and counters configuration.

10.4.1.6.2.2 ANALOG PHYs IPs

The PCIe and HDMI analog PHYs can also be configured to consume less power when the are in a non-active state. Details on how to put the IPs in low power mode can be found in each of the IP documentation. Further reduction can be achieved by settings the ENABLE_WEAK_LINREG in the PMU PMU_REG_2P5 register.

10.4.1.7 System domains layout

The following table describes the different power modes.

NOTE

Wakeup time is the hardware perspective, and doesn't reflect the time it takes software to resume drivers and perform system operations.

Table 10-1. Aimed power modes

Mode	Description (Status of main power domains)	Wake-up capability	Wakeup time	Applicable use case
RUN	Power supplies are on, all clocks are on.	N/A	N/A	All
WAIT	Power supplies are on. Arm executes WFI command, Option to gate off clocks for specific modules based on programmable bits.	Based on interrupt from any module	Immediate (Arm exit WFI)	IDLE process of operating system.
STOP	Power supplies are on. Arm executes WFI command. VPU and GPU are in power gating PLLs are off, hence all system clocks are off.	Based on interrupt from modules that can track operation with low frequency clock- like Keypad press, GPIO, Timer, CAN activity, etc.	~50us	Suspend state of operating system - system is on but is waiting for operation.
Standby	- PLLs disabled - Low voltage could be applied to Arm and SoC power inputs - Chip regulator bypass if reduce supply voltage, - Well Bias could be activated - CPU regulator disabled for power gating - PU regulator is disabled in software - xtal enabled	Same as above	~220 us	Low power standby mode with limited reaction on external events
Deep Sleep	Same as above but with OSC24M disabled	Interrupts from modules those could generate asynchronous interrupts (Keypad, GPIO, SNVS)	~1400us	
SRTC	Only SNVS domain and OSC32K keep alive	SRTC security alert / SRTC timer expire. Based on each one of those, SRTC module will generate request to PMIC to perform POR sequence and walk-up the system. The system will go through boot process.	~1700us to WARM boot COLD boot, depends on boot procedure and boot device	

Table 10-1. Aimed power modes

Mode	Description (Status of main power domains)	Wake-up capability	Wakeup time	Applicable use case
		PMIC can also receive press on "ON" button to walk-up the system. This should be connected directly to PMIC.		

There is a single hardware signal coming into PMU which sets the PMU in either of two "STOP" states. The STOP state is implemented is controlled by the PMU_MISC0[STOP_MODE_CONFIG] bit (See PMU Memory Map/Register Definition). It is recommended that the blocks be configured for safe powerdown/up through the registers before asserting the stop_mode signal. Blocks not described in the section below are unaffected by stop_mode.

If the stop_mode_config is set to zero, thus in the STOP mode all blocks powered down in minimum power configuration.

If the stop_mode_config is set to one, thus in the STOP mode some of the blocks remain powered and in different states as defined in the table below.

Table 10-2. STOP mode configuration

Block	STOP_MODE_CONFIG=0	STOP_MODE_CONFIG=1
reg1p1	off	on
reg2p5	off	on
reg3p0	off/on depending on vbus. Uses crude local reference if vbus is present	off/on depending on vbus . Uses analog central bandgap if VBUS is present.
reg_core	bypassed if not power gated.	bypassed if not power gated
reg_pu	bypassed if not power gated.	bypassed if not power gated.
reg_soc	bypassed	bypassed
bandgap	off	functional
temp_sensor	off	off
well_bias	hardware controlled	hardware controlled
All PLLs	off	off
OSC24M	off	Controlled by CCM configuration
LVDS clock I/O CLK1 and CLK2	off	off

10.4.2 Power management techniques

The device supports the power-management techniques with the features found here.

- Partitioning of the device into voltage, power, clock, and reset domains
- Domain isolation that allows flexible configurations of domains on/off states to form use cases targeting various applications
- Clock tree with selective clock-gating conditions and almost independent clock roots
- Power, reset, and clock control hardware mechanism to manage sleep and wake-up dependencies of power domains
- Software-controllable and hardware-controllable clock gating for functional modules and buses
- Memory retention and state retention capability (Software State Retention for Arm core) for preserving memory contents and device state in low-power modes
- Support for low-power device modes input/output (I/O) pad configuration for minimum power
- Variety of operating modes to optimize device performance and wake-up times
- Thermal monitoring and thermal aware performance management

Many of the low power features are fully or partially software controllable and can be configured for the specific requirements of a target system.

Combining these techniques, the system designer may meet tight requirements of low-power standby and operational modes while maintaining high performance for time-critical tasks.

10.4.2.1 Power saving techniques

The table below lists power saving techniques supported by the SoC in their connection to different components of power consumption.

Table 10-3. Power saving design/architecture and power saving techniques

Techniques	Active SoC Power	Standby SoC Power	System Power
Temperature Monitoring, and active frequency throttling	√		
Arm Core Platform SRPG (Software)		√	
Arm Core Platform Power Gating		√	
Cortex-M4 Asymmetric multicore	√	√	√
VPU and GPU3D Power Gating	√[1]	√	
Clock gating (automatic dynamic and forced)	√		
Integrated PMU (IR drop, efficiency, accuracy)	√		√
C4 package (IR drop, thermal)	√		
Display Backlight optimization (IPU, SW)			√
Architecture: L2 cache, Video / Audio / Graphics acceleration	√		
Architecture: PCIe, LVDS, HDMI, USB integration			√
Low Power DDR: LPDDR2, LV-DDR3			√

1. Applicable to use cases where GPU3D and VPU operation is not required.

10.4.2.2 Thermal-aware power management

The temperature sensor block (TEMPMON) implements a temperature sensor/conversion function. The block features an alarm function that can raise an interrupt signal if the temperature is above a specified threshold.

Software may implement temperature aware DVFS for the Arm domain and the GPU domain, as well as temperature aware frequency scaling for other system components to ensure that both the frequency and voltage is lowered when the die temperature is above the specified limit.

Software may also implement temperature aware task scheduling to ensure that non-critical tasks are suspended when the die temperature is above the specified limit.

See Temperature Monitor (TEMPMON) for further details on temperature monitor functions and programmability options.

10.4.2.3 Peripheral Power management

10.4.2.3.1 Main memory power management

Main system memory, DDR3, and LPDDR2 are some of the most power-hungry system components, but the SoC provides several options to manage DDR power.

Automated power saving modes are supported by the MMDC hardware. This feature allows the DDR memory to automatically enter self-refresh mode when there are no DDR accesses for a configurable time. The default setting is 1024 clock cycles which can be optimized based on the customer use case and application.

See Power Saving and Clock Frequency Change modes for further details on MMDC power saving features and programmability options.

Software may support DDR frequency scaling. Automated frequency changing procedure is supported by MMDC and CCM modules.

NOTE

DDR frequency changes cost extra time and power. Slowing requestors while keeping DDR at full speed may increase total system power. Software may also implement Cooperative Dynamic Frequency Scaling in order to keep the system balanced, (that is, keep the system in balance when DDR

throughput is equal or slightly higher than total amount of requests generated by all requestors).

Reducing the DDR frequency while in DLL-ON mode may be not efficient because:

- Reduction in DDR frequency will cause bus duty cycle to increase and thus reduces chance of automatic MMDC power saving (place memory into SR).
- Total amount of read/write operation does not change (power is per-operation).
- The termination is active longer, though, lowering frequency below 396 MHz may enable lowering drive strengths and termination.

When possible at lower performance use cases, software may switch DDR3 to DLL-off mode. This allows it to greatly reduce DDR3 frequency and thus disable or reduce termination and drive strength, which significantly reduces the power consumption of the DDR3 interface.

A good strategy for many types of workloads is to combine most activity in bursts (natively possible, for example, for typical multimedia applications, communication, etc.) and run this segment at maximal speed, then switch to DLL-OFF mode to support background activity (communication, display refresh, housekeeping).

The DRAM Interface power dissipation depends on many variables, however, proper termination and drive strength is key for power and thermal performance. Memory and controllers provide a host of programmable options for the drive strength of the output buffers and for the on-die termination impedance.

The ideal settings for drive strength and ODT also depend on the clock frequency to ensure that inter-symbol interference (ISI) effects are not introduced.

DDR PHY power is proportional to the amount and type of bus activity.

In cases where the DDR is placed into self-refresh, software can configure DDR I/O to be floated or lowered to the minimum drive allowed by JEDEC.

Modifying the DDR drive strength must be done by code that is executing from a memory region other than DDR (for example, IRAM). No access to DDR (including page table walks, cache misses, alternate bus master accesses, etc.) is allowed while the DDR I/O pads are being re-configured.

10.4.2.3.2 IO power reduction

Software configures IO to low power modes:

- PHYs - make sure that all unused PHYs are placed to lowest power state. Please refer relevant chapter for further information about different PHYs

- Digital IOs - Make sure all unnecessary PU/PD are disabled and IO are switched to either minimal drive strength or to input mode (when applicable)
- Set DDR type IO to CMOS mode if possible, specifically RGMII segment
- Disable all unused LVDS pads (ANALOG, MLB PHY, LDB)

10.4.2.4 Examples of External Power Supply Interface

This section presents the example of external power supply interfacing to the chip.

The scenario based on integrated PMU system is presented in the following figure. This scenario minimizes BoM and board design complexity.

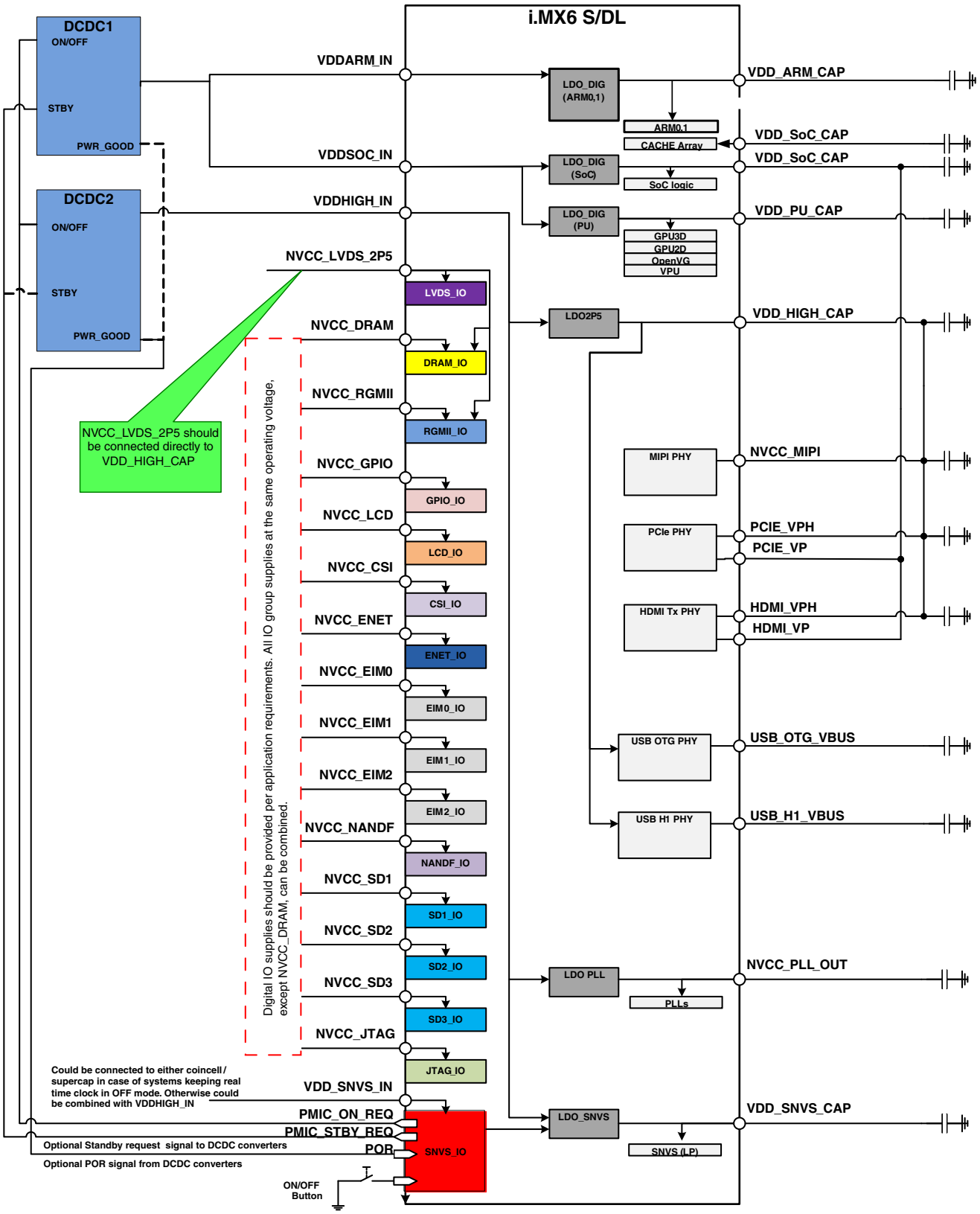


Figure 10-11. Supplying i.MX 6Solo/6DualLite power using integrated PMU

The scenario based on external programmable regulators is presented below: This scenario could be used in highly power optimized systems.

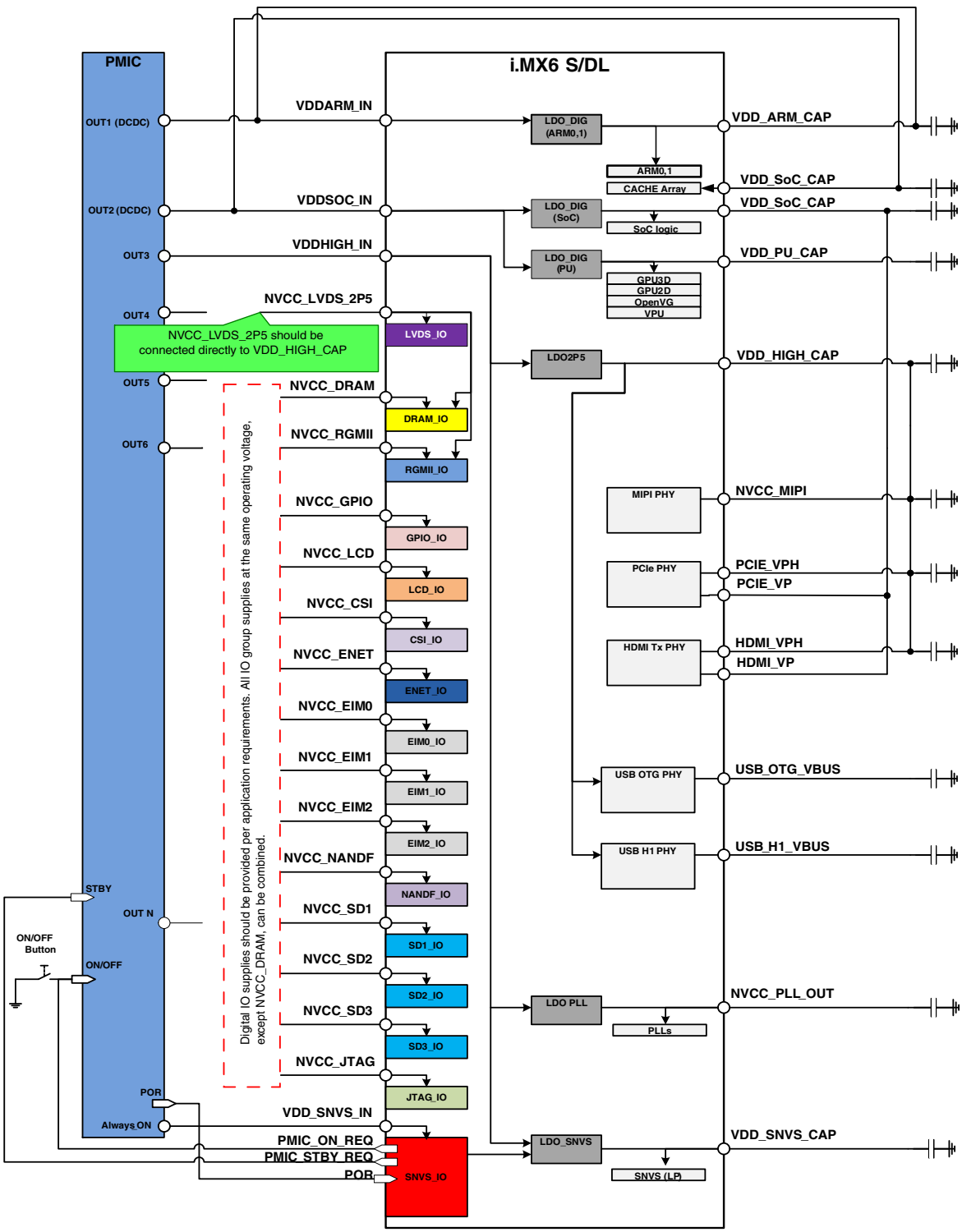


Figure 10-12. Supplying i.MX 6Solo/6DualLite Power Using External Programmable PMIC

10.5 ONOFF (Button)

The chip supports the use of a button input signal to request main SoC power state changes (i.e. On or Off) from the PMU.

The button is used to power On and Off the SoC using an always-on (e.g., coincell battery-backed) power domain.

- When the chip main power supply is Off, a button press greater in duration than 750 ms asserts an output signal to request power from a power IC to power up the SoC.
- When the chip main power supply is On, a button press between 750 ms and 5 seconds will send an interrupt to the core to request that software bring down the SoC safely. Software may respond to the interrupt by saving the processor state and then setting a control bit that requests to the power IC the removal of the main power supply.
- Button presses greater than 5 seconds, when the SoC is powered, results in a direct hardware power down request signal to the power IC without providing a software interrupt first. This long button press initiates a hardware-enforced mode which is applicable when software is unable to power Off the device.

The button is connected to input ONOFF and the power IC is connected the chip output PMIC_ON_REQ. The chip must have TEST_MODE deasserted. An always-On supply (e.g. coincell) is needed in the system for this feature.

Chapter 11
System Security

11.1 Overview

Security is a common requirement for platforms built using the chip, although the specific needs vary greatly depending on the platform and market. The type and cost of assets to be protected on a portable consumer device are very different from those to be protected on automotive or industrial platforms, and the same applies to the kind of attacks and level of resources threatening those assets. The platform designer must select an appropriate set of counter measures to meet the relevant platform security needs.

For the platform designer to meet the requirements for each market, the chip incorporates a range of security features which can be used individually or in concert to underpin the platform security architecture. Most of the chip security features provide protection against particular kinds of attack and can be configured at various levels according to the required degree of protection. These features are designed to work together and can be integrated with appropriate software to create defensive layers. In addition to protection features, the chip includes a general purpose accelerator to enhance the performance of selected industry standard cryptographic algorithms.

The following is an introduction to the chip security components.

- Cryptographic module with AES 128/256 symmetric algorithms, random number generator entropy source and NIST-validated DRBG, cryptographic key protection, run-time integrity checking.
- High Assurance Boot (HAB) feature in the System Boot up to RSA-4096 signature verification
- Secure Non Volatile Storage (SNVS)
- TrustZone (TZ) Architecture in the ARM Cortex A9 Platform, TrustZone aware Interrupt Controller (GIC) and TrustZone Watchdog Timer (WDOG-2)
- TrustZone Address Space Controller (TZC-380) - providing security address region control functions on DDR memory space.
- On-chip RAM (OCRAM) with TrustZone protection using OCRAM controller.

- 16 Kbyte of on-chip Secure RAM
- On chip OTP (OCOTP) with on-chip electrical fuses
- Central Security Unit (CSU)
- Secure JTAG Controller (SJC)
- Locked mode in the Smart Direct Memory Access (SDMA) controller
- For DTCP (Digital Transmission Content Protection) adopters, DTCP functionality to support DTCP-MOST and DTCP-IP.
- 12 tamper pins supported
- Hardware Cryptographic Accelerators
  - Symmetric: AES-128, AES-192, AES-256, DES, 3DES, and ARC4
  - Hash Message Digest and HMAC: SHA-1, SHA-224, SHA-256, and MD-5
- True and Pseudo Random Number Generator

Detailed descriptions of the components are provided in the *Multimedia Applications Processor Security Reference Manual for i.MX 6Dual/6Quad and i.MX 6Solo/6DualLite*.

11.2 Central Security Unit (CSU)

11.2.1 CSU Overview

The CSU manages the system security policy for peripheral access on the SoC. The CSU allows trusted code to set individual security access privilges on each of the peripherals, using one of eight security access privilege levels. Also, according to programmed policy, the CSU may assign bus master security privileges during bus transactions.

11.2.2 CSU Features

The Central Security Unit (CSU) sets access control policies between bus masters and bus slaves, allowing peripherals to be separated into distinct security domains. This protects against unauthorized access to data e.g. when software programs a DMA bus master to access addresses that the software itself is prohibited from accessing directly. Configuring DMA bus master privileges in the CSU consistent with software privileges defends against such attempted accesses.

CSU has the following security related features:

- Peripheral access policy - Appropriate bus master privileges and identity are required to access each peripheral.
- Masters privilege policy - CSU overrides bus master privilege signals, i.e. user/supervisor secure/non-secure, according to access control policy.

11.2.3 CSU Functional Description

The CSU enables secure software to set bus privilege security policy within the platform.

Security policies may be set, and optionally locked in the CSU registers. These privilege values may originate in the command sequence file (CSF) which is processed by the High Assurance Boot (HAB) itself or by an HAB authenticated image which executes after the initial boot ROM phase.

11.2.3.1 CSU Peripheral Access Policy

According to its programmed policy, the CSU determines the bus master privileges and the masters that are allowed to access each of the slave peripherals.

There are four security modes of operation (i.e. bus privileges) in the system distinguished by security (TrustZone/non-TrustZone) and privilege (Supervisor/User) setting of the module. Below is the list of these security modes from the highest security level to the lowest:

- TrustZone (Secure) Privilege (Supervisor) Mode - Highest Security Level
- TrustZone (Secure) non-Privilege (User) Mode - Medium Security Level
- non-TrustZone (Regular) Privilege (Supervisor) Mode - Medium Security Level
- non-TrustZone (Regular) non-Privilege (User) Mode - Lowest Security Level

This functionality is implemented as follows:

The Configure Slave Level (CSL) Register value for a specified peripheral resource defines the output signal -- csu_sec_level for that peripheral. The value of this signal determines by what master privileges a peripheral is accessible. The relationship between the value of the csu_sec_level signal and security operation mode is shown in the table below. The CSL registers reside in the CSU module. Details, describing CSL register fields and how they are programmed to control access privileges for specific peripherals, can be found in the Security Reference Manual.

Table 11-1. Permission Access Table

CSU_SEC_LEVEL[2:0]	Non-Secure User Mode	Non-Secure Spvr Mode	Secure (TZ) User Mode	Secure (TZ) Spvr Mode	CSL register value
(0) 000	RD+WR	RD+WR	RD+WR	RD+WR	8'b1111_1111
(1) 001	None	RD+WR	RD+WR	RD+WR	8'b1011_1011
(2) 010	RD	RD	RD+WR	RD+WR	8'b0011_1111
(3) 011	None	RD	RD+WR	RD+WR	8'b0011_1011
(4) 100	None	None	RD+WR	RD+WR	8'b0011_0011
(5) 101	None	None	None	RD+WR	8'b0010_0010
(6) 110	None	None	RD	RD	8'b0000_0011
(7) 111	None	None	None	None	Any other value

11.3 Cryptographic Acceleration and Assurance Module (CAAM)

11.3.1 CAAM Overview

CAAM is a cryptographic acceleration device that accelerates block encryption algorithms, stream cipher algorithms, hashing algorithms, and random number generation. It has an integral DMA engine that allows CAAM to fetch its command programs, read input data and write the resulting output.

CAAM works with the SNVS to provide platform assurance features, including support for High Assurance Boot, detection of and response to potential tamper events, and shortterm and long-term protection of secret data such as public keypairs, Digital Rights Management keys and proprietary software.

CAAM provides the following features:

- Cryptographic acceleration of hashing, encryption and decryption
- Offloading of cryptographic functions via programmable command descriptor language
- Two independent queues for commands and results
- Register Bus Interface used for configuration, control and status
- DMA, including scatter/gather support for data
- Automatic short-term encryption/decryption of cryptographic keys
- Long-term protection of secret data via device-specific encryption keys

- Secure Memory for access-controlled protection of critical data, and automatic zeroization in the case of tampering
- Random Number Generation using a true random number generator and a NISTcompliant pseudo random number generator

11.4 Secure Non-Volatile Storage (SNVS)

11.4.1 SNVS Overview

SNVS is a hardware device that includes a security state machine and security violation detection circuits that, together with High Assurance Boot software, determine whether the chip is currently in a secure state.

When the security state machine indicates a secure state, the SNVS allows CAAM use of special cryptographic keys to decrypt long-term secrets such as public/private keypairs, Digital Rights Management keys and proprietary software. When the SNVS detects a potential security violation, such as a tamper alert, the SNVS signals CAAM to erase sensitive data such as cryptographic keys, and sends an interrupt to alert the Operating System of the event. The SNVS also includes a general purpose real-time counter.

The SNVS includes the following features:

- Security State Machine driven by High Assurance Boot software and tamper detection circuits
- Master Key Control that protects the integrity and secrecy of the Master Key (OTPMK) stored in fuses
- Tamper detection circuits that detect JTAG events, power glitches, Master Key ECC check failure, and software-reported and hardware-reported security violations
- 256-bit Zeroizable Master Key that can be automatically erased in the event of a security breach
- Tamper-protected Secure Realtime Counter that continues running when the chip is powered off
- Non-volatile Monotonic Counter used to protect against “roll-back” attacks
- Non-volatile General Purpose Register can be used to store a 32-bit value across power cycles
- Non-Secure Real Time Counter with programmable alarm and periodic interrupt

11.4.2 Tamper Detection

Tamper Detection is a special mechanism provided through a chip pin to signal when the device encounters unauthorized opening or tampering.

When not in use, the Tamper Detection signal is pulled-down internally. In case of use, it should be connected to a Tamper Detection contact in a target system (Normally closed, pulled-up to the VDD_SNVS_IN).

An always-ON power supply (coincell battery) should be present in the system. If the tamper detection feature is enabled by software then opening of the tamper contact:

- Switches system power ON with a Tamper Detection alarm interrupt asserted (for software reaction)
- Activate security related hardware (e.g. automatic and immediate erasure of the Zeroizable Master Key and deny access and erase secure memory contents)

11.5 Data Co-Processor (DCP)

For security purposes, the Data Co-Processor (DCP) provides hardware acceleration for the cryptographic algorithms. The features of DCP are:

- Encryption Algorithms: AES-128 (ECB and CBC modes)
- Hashing Algorithms: SHA-1 and SHA-256
- Key selection from the SNVS, DCP internal key storage, or general memory
- Internal Memory for storing up to four AES-128 keys—when a key is written to a key slot it can be read only by the DCP AES-128 engine
- IP slave interface
- DMA

11.6 High-Assurance Boot (HAB)

The HAB, which is the high-assurance boot feature in the system boot ROM, detects and prevents the execution of unauthorized software (malware) during the boot sequence.

When the unauthorized software is permitted to gain control of the boot sequence, it can be used for a variety of goals, such as exposing stored secrets; circumventing access controls to sensitive data, services, or networks, or for repurposing the platform. The unauthorized software can enter the platform during upgrades or reprovisioning, or when booting from the USB connections or removable devices.

The HAB protects against unauthorized software by:

- Using digital signatures to recognize the authentic software. This enables you to boot the device to a known initial state and run the software signed by the device manufacturer.

11.7 System JTAG Controller (SJC)

The JTAG port provides debug access to hardware blocks, including the Arm processor and the system bus. This enables program control and manipulation as well as visibility to the chip peripherals and memory.

The JTAG port must be accessible during initial platform development, manufacturing tests, and general troubleshooting. Given its capabilities, JTAG manipulation is a known attack vector for accessing sensitive data and gaining control over software execution. The System JTAG Controller (SJC) protects against the whole range of attacks based on unauthorized JTAG manipulation. It also provides a JTAG port that conforms to the IEEE 1149.1 and IEEE 1149.6 (AC) standards for BSR (boundary-scan) testing.

The SJC provides these security levels:

- The JTAG Disabled-JTAG use is permanently blocked.
- The No-Debug-All security sensitive JTAG features are permanently blocked.
- The Secure JTAG-JTAG use is restricted (as in the No-Debug level) unless a secret-key challenge/response protocol is successfully executed.
- The JTAG Enabled-JTAG use is unrestricted.

The security levels are selected via the e-fuse configuration.

NOTE

For the final production hardware, the most secure and preferred method is to ensure that the JTAG interface is not pinned or routed out at all. The same applies to any other trace, diagnostic or test ports including SDP.

Chapter 12
ARM Cortex A9 MPCore Platform (ARM)

12.1 Overview

The Cortex-A9 Core Platform consists of an ARM®Cortex®-A9 MPCore processor, which includes a NEON co-processor, a private timer and watch-dog, and 32 KB + 32 KB L1 data and instruction caches per core.

The Cortex-A9 Core Platform consists of a unified 512 KB L2 cache, SCU (Snoop Control Unit), and Generic Interrupt Controller (GIC). In addition, the Cortex-A9 Core Platform includes various components composing the ARM CoreSight debug/Trace system, including PTM and a 2 / 3[1] x CTI, 2xCTM, and 16KB ETB.

The Cortex-A9 processor utilizes two AXI-64 master ports connected from the SCU to the Level 2 Cache. The L2 cache also utilizes 2x AXI-64 to access the L3 memory or other SoC peripherals in a symmetric way.

The core supports debug through real-time trace via PTM, and static debug via JTAG.

The core platform supports static debug through the debug logic to SOC. This includes the capability of real time trace via ARM's CoreSight PTM, ETB and TPIU modules. The CTI and CTM modules allow cross-triggering of internal and external trigger sources.

12.2 External Signals

The following table describes the external signals of ARM:

Table 12-1. ARM External Signals

Signal	Description	Pad	Mode	Direction
ARM_EVENTI	Input event signal	GPIO_5	ALT7	I

Table continues on the next page...

1. Depending on specific part: i.MX 6Solo, i.MX 6DualLite.

Table 12-1. ARM External Signals (continued)

Signal	Description	Pad	Mode	Direction
ARM_EVENTO	Output event signal	CSI0_PIXCLK	ALT7	O
ARM_TRACE00	Trace signal	CSI0_VSYNC	ALT7	O
ARM_TRACE01	Trace signal	CSI0_DAT4	ALT7	O
ARM_TRACE02	Trace signal	CSI0_DAT5	ALT7	O
ARM_TRACE03	Trace signal	CSI0_DAT6	ALT7	O
ARM_TRACE04	Trace signal	CSI0_DAT7	ALT7	O
ARM_TRACE05	Trace signal	CSI0_DAT8	ALT7	O
ARM_TRACE06	Trace signal	CSI0_DAT9	ALT7	O
ARM_TRACE07	Trace signal	CSI0_DAT10	ALT7	O
ARM_TRACE08	Trace signal	CSI0_DAT11	ALT7	O
ARM_TRACE09	Trace signal	CSI0_DAT12	ALT7	O
ARM_TRACE10	Trace signal	CSI0_DAT13	ALT7	O
ARM_TRACE11	Trace signal	CSI0_DAT14	ALT7	O
ARM_TRACE12	Trace signal	CSI0_DAT15	ALT7	O
ARM_TRACE13	Trace signal	CSI0_DAT16	ALT7	O
ARM_TRACE14	Trace signal	CSI0_DAT17	ALT7	O
ARM_TRACE15	Trace signal	CSI0_DAT18	ALT7	O
ARM_TRACE_CLK	Clock signal	CSI0_DATA_EN	ALT7	O
ARM_TRACE_CTL	Control signal	CSI0_MCLK	ALT7	O

12.3 Platform configuration

The Bus, Cortex A9 Core, and L2 Cache configuration options are contained in the following subsections.

Table 12-2. Cortex-A9 revision

Core	MP004-BU-50000-r2p10-0rel0
Neon	AT397-BU-50001- r2p0-00rel0
PL310	PL310-BU-00000-r3p2-00rel0

12.3.1 Platform and SCU configuration

Table 12-3. Cortex-A9 configuration

Option	Selected Value	Comments
MP_MODE	Yes	Multi-Processor mode
POWER_DOMAIN_WRAPPER	No	Wrappers to support power off of individual cores.
PTM_INTERFACE_PRESENT	Yes	Use PTM as part of Trace/Debug logic.
PARITY	Yes	Using RAM arrays which support parity.
CORE_NUM	2/1[1]	Number of cores
INT_NUM	128	Number of interrupts (SPIs) in GIC
ACP_PRESENT	No	Accelerator Coherency Port (ACP)
MASTER_NUM	2	Number of 64-bit AXI output master ports.

1. Depending on type of part - i.MX 6Solo, i.MXDualLite.

12.3.2 Core configuration

Table 12-4. Cortex-A9 Core configuration

Option	Selected Value	Comments
DCACHESIZE	32	L1 Data cache size
ICACHESIZE	32	L1 Instruction cache size
TLBSIZE	128	
JAZELLE_PRESENT	Yes	Providing ARM's Jazelle technology hardware extensions.
FPU_PRESENT	No	The FPU functions are provided by NEON, thus additional FPU cannot be used.
NEON_PRESENT	Yes	Use MPE, NEON Co-Processor and FPU
PRELOAD_ENGINE_PRESENT	No	May only be beneficial in Video processing.

12.3.3 PL310 L2 Cache configuration

Table 12-5. PL310 L2 Cache configuration

Option	Selected Value	Comments
Cache way size	32 KB	(For total of 512 KB L2 size)
Number of cache ways	16	Performance enhancement versus 8 ways
RAM latencies	4	
Data RAM banking	Yes	Significantly improves cache throughput

Table continues on the next page...

Table 12-5. PL310 L2 Cache configuration (continued)

Option	Selected Value	Comments
Slave port 1 present	Yes	
Master port 1 present	Yes	
Parity logic	Yes	For military / surveillance applications, and side ease the process of identify memory related issues.
Lockdown by master	Yes	Increase L2 optimization
Lockdown by line	Yes	Increase L2 optimization
AXI ID width	5	
Address filtering	No	Help in timing closure, not required for symmetric AXI bus connectivity scheme.
Speculative read	Yes	Performance boost, when used with CortexA9.
Size of L2 cache	512 KB	Size is implied by Cache-Size times cache-ways (i.e. 512 KB)

12.3.4 Endian Modes

The Cortex-A9 Core Platform supports little endian mode only. Big Endian is not supported even though both modes are supported by the Cortex-A9 processor.

12.3.5 Memory Parity error support

The i.MX 6Solo/6DualLite ARM Cortex A9 MPCore ™ platform supports Parity Fail signals for several of the RAM arrays.

Parity fail indication is provided by "PARITYFAILn[7:0]" and "PARITYFAILSCU[N:0]" buses for system notification. On parity error event, these signals will be ORed together to provide a single event (interrupt) to the core, in case of any parity fail event.

12.4 Performance and Power

This section will discuss the operational conditions and performance goals for the Cortex-A9 Core Platform.

12.4.1 Low-Power design

The Cortex-A9 Core Platform low-power design is based on these characteristics:

- Symmetric processing and ARM design by using the same clock frequency on cores

- Low leakage of LP process
- C4 package

As a result, the proposed power modes and power management scheme is as follows:

- The same voltage level must be used for the logic part of the whole platform.
- Separate voltage for L1 memories array is required to allow DVFS.
- On-chip power switches are not utilized by gating the power of the platform directly from the regulator.

12.4.1.1 SRPG (State Retention Power Gating)

ARM core SRPG is implemented by software save & restore of essential configuration registers prior to the complete power-down of the entire ARM platform.

Save & restore utilizes "Dormant mode" for the primary core (L1 cache flushed, L2 preserved), and power down mode of all other cores, as follows:

Power Down flow[2]

- Power down request
- Save core's essential registers and platform registers to L2/DDR memory by Dormant mode routines.
- Perform L1 cache clean operation on all cores
- Enter WFI state
- Power Gating of all cores' and platform logic

Power Up flow[2]

- Power up request to GPC external controller (interrupt)
- Supply power
- Reset
- Restore registers of platform and cores from memory by Dormant mode routines.

12.4.1.2 Dynamic Voltage and Frequency Scaling (DVFS)

The Cortex-A9 Core Platform has been designed, in conjunction with external control logic and software, to support dynamic scaling of voltage and frequency.

12.4.2 Clocks, frequency goals

2. Dormant mode implementation information, provided by ARM, is preliminary.

12.4.2.1 ARM Clock

For ARM clock please see the product data sheet.

12.4.2.2 Bus Clocks

The AXI master ports are designed to run at half the frequency of the Cortex A9 core clock.

The on-platform debug components clock is asynchronous to the ARM core clock.

12.4.2.3 Debug Clocks

The ARM platform contains several debug components that use different internal clock names, but are all derived from the same ahb_clk clock root. These clocks, listed with their default values, are as follows.

- Trace buffer (ETB) clock - 133 MHz
- Trace port (TPIU) clock - 133 MHz
- AHB clock - 133 MHz

12.5 Core Platform Sub-Blocks details

12.5.1 ARM Cortex A9 MPCore ™ Processor

The information presented in this section focuses on design aspects of the ARM Cortex A9 MPCore ™ in the AP subsystem.

The ARM Cortex A9 is a high-performance, low-power, synthesisable processor with an L1 cache subsystem that provides full virtual memory capabilities. The Cortex A9 processor implements the ARMv7-A architecture and runs 32-bit ARM instructions, 16-bit and 32-bit Thumb2 instructions, and 8-bit Java™ byte-codes in Jazelle state.

The ARM Cortex A9 MPCore ™ processor in the chip consists of:

- Two (2) Cortex A9 processors in a cluster and a Snoop Control Unit (SCU) that can be used to ensure coherency within the cluster.

- A set of private memory-mapped peripherals, including a global timer and a watchdog and private timer for each Cortex-A9 processor present in the cluster.
- An integrated Interrupt Controller is an implementation of the Generic Interrupt Controller architecture. The integrated Interrupt Controller registers sit beside the timers and watchdog control registers in the private memory region of the Cortex-A9 MPCore.

Individual Cortex-A9 processors in the Cortex-A9 MPCore cluster are symmetrically implemented with hardware configurations as specified in Platform configuration.

12.5.2 Media Processing Engine (MPE - NEON)

The Media Processing Engine (MPE) implements ARM NEON technology, a media and signal processing architecture that adds instructions targeted at audio, video, 3-D graphics, image, and speech processing.

Advanced SIMD instructions are available in both ARM and Thumb states. The MPE also implements a VFPv3-D32 Floating-Point Unit.

The ARM Cortex A9 MPCore ™ Platform includes MPE per core.

12.5.3 Generic Interrupt Controller (GIC)

The Cortex-A9 MPCore contains an integrated interrupt controller that shares the same programmer's model as the PL390 (GIC), although there are implementation-specific differences.

12.5.3.1 Interrupt Controller Features

- 128 interrupt sources
- The Cortex-A9 multiprocessor contains the following types of interrupts:
  - Up to 16 Software Generated Interrupts (SGIs)
  - Private Peripheral Interrupt (PPI) - an interrupt generated by a peripheral that is specific to a single Cortex-A9 processor (there are 5 PPIs for each Cortex-A9 processor interface)

- Shared Peripheral Interrupt (SPI) - an interrupt generated by a peripheral which the Interrupt Controller can route to any or all Cortex-A9 processor interfaces. The Interrupt Controller supports a maximum of 224 SPIs .
- Lockable Shared Peripheral Interrupts (LSPI) - there are 31 LSPIs, interrupts 32-62. The user can configure and then lock these interrupts against further change using CFGSDISABLE. The LSPIs are present only if the SPIs are present.

For more informtation, see ARM MPCORE Technical Reference Manual.

12.5.3.2 About the Interrupt Controller

The Interrupt Controller is a single functional unit that is located in a Cortex-A9 multiprocessor design.

The Interrupt Controller is memory-mapped. The Cortex-A9 processors access it by using a private interface through the SCU.

12.5.3.3 Interrupt Controller Clock frequency

The interrupt controller's clock period is 2x multiple of the main clock period.

The watchdogs and timers use the same clock as the interrupt controller.

12.5.3.4 TrustZone support

The Interrupt Controller permits all implemented interrupts to be individually defined as Secure or Non-secure.

The user can program Secure interrupts to use either the IRQ or FIQ interrupt mechanism of a Cortex-A9 processor through the FIQen bit in the ICPICR Register.

Non-secure interrupts are always signalled using the IRQ mechanism of a Cortex-A9 processor.

12.5.4 Instruction and data caches (L1)

The Cortex-A9 processor is configured with a 32 Kbyte Instruction Cache and a 32 Kbyte Data Cache.

12.5.4.1 L1 features

- Four-way set associative cache
- Virtually indexed and physically addressed
- Capable of providing two words per cycle for all requesting sources
- Eight 32-bit words per cache line
- 128 indexes per tag RAM

12.5.5 L2 Cache and controller (PL310)

The platform includes unified (data / instruction) L2 cache unit, based on the cache controller IP by ARM.

The Cortex-A9 processor utilizes 2x AXI-64 master ports connected from the SCU to the Level 2 Cache. The L2 cache also utilizes 2x AXI-64 to access the L3 memory or other SoC peripherals in a symmetric way.

See Table 12-5 for more information.

12.6 Debug and Trace Sub-blocks (CoreSight components)

This section gives a brief overview of the modules that are implemented within the Cortex-A9 Core Platform.

The Cortex-A9 Core Platform debug blocks are part of the overall CoreSight debug system which include the 16KB ETB, 2 x CTM's, 2 / 3[3] x CTI's, ATB replicator, DAP, TPIU and APB address decode.

The CoreSight™ compatible Program Flow Trace Macrocell (PTM) provides control for ARM software tracing and debug. The Cross Trigger Interface (CTI) is included in the Cortex-A9 platform to provide a common programming model for use by the debug tools, control the trigger sources, and interface to the Cross Trigger Matrix (CTM). The debug is controlled via an ARM Debug Access Port (DAP).

For details of the full CoreSight debug subsystem, see the System Debug chapter.

3. Depending on specific part: i.MX 6Solo, i.MX 6DualLite.

12.6.1 Debug Access Port (DAP)

The Debug Access Port (DAP) is an implementation of an ARM Debug Interface version 5.1 (ADIv5.1) comprised of a number of components supplied in a single configuration.

All the supplied components fit into the various architectural components for Debug Ports (DPs), which are used to access the DAP from an external debugger and Access Ports (APs), to access on-chip system resources.

The debug port and access ports together are referred to as the DAP. The DAP provides real-time access for the debugger without halting the processor to:

- AMBA system memory and peripheral registers
- All debug configuration registers

The DAP also provides debugger access to JTAG scan chains of system components, (to non-CoreSight compliant processors, for example).

12.6.2 Program Trace Macrocell (PTM)

The PTM unit is a nonintrusive trace macrocell that filters and compresses instruction trace for use in system debugging and system profiling.

The PTM unit has an external interface outside of the processor called the *Advanced Trace Bus* (ATB) interface. The PTM is an evolvement of the ETM, designed for the Cortex A9 cores, handling program trace only.

The Cortex A9 PTM provides real time instruction trace for the Cortex A9. It's designed to be used with the CoreSight Design Kit.

Real time tracing is controlled by specifying a set of filtering and triggering resources which include address and data comparators, counters and sequencers.

Two main schemes can be used for connecting PTMs:

1. Single PTM - shared by all cores and resources.
2. Nx PTMs, where N is numbers of cores in the system.

12.6.2.1 Program Flow Trace (PFT)

The CoreSight Program Flow Trace Macrocell (PTM) is based on the Program Flow Trace (PFT) architecture. The PTM generates information that trace tools use to reconstruct the execution of all or part of a program.

The PFT architecture assumes the trace tools can access a copy of the code being traced. For this reason, the PTM generates trace only at certain points in program execution, called waypoints. This reduces the amount of trace data generated by the PTM compared to the ETM protocol. Waypoints are changes in the program flow or events, such as an exception. The trace tools use waypoints to follow the flow of program execution.

For full reconstruction of the program flow, the PTM traces:

- Indirect branches, with target address and condition code
- Direct branches with only the condition code
- Instruction barrier instructions
- Exceptions, with indication of where the exception occurred
- Changes in processor instruction set state
- Changes in processor security state
- Context-ID changes
- Entry to and return from Debug state when Halting Debug-mode is enabled

You can also configure the PTM to trace:

- Cycle count between traced waypoints
- Global system timestamps
- Target addresses for taken direct branches

PTM components include the following main components:

- Processor Interface to monitor the behavior of the processor.
- Trace Generation to create a real-time trace stream.
- Filtering and Triggering Resources used to affect when trace is generated and to control the capturing of trace by the trace tools.
- Main FIFO (72 bytes) flattens out any bursts in the trace stream, and signals an overflow in the trace when it becomes full, halting trace generation until the FIFO empties.
- ATB interface for PTM output.
- APB interface to access the PTM registers.

12.6.3 Cross Trigger Interface (CTI)

This block controls the Trigger Interface (TI). The CTI combines and maps the trigger requests, and broadcasts them to all other interfaces on the ECT as channel events.

When the CTI receives a channel event, it maps this onto a trigger output. This enables subsystems to cross trigger with each other. The receiving and transmitting of triggers is performed through the TI. Each CTI has 3/5 trigger inputs.

See System Debug for more information on the CTI.

12.6.4 Embedded Trace Buffer (ETB)

The ETB provides on-chip storage of trace data using 32-bit, 16KB RAM.

The ETB accepts trace data from the Cortex-A9 via an ATB port (passing through a replicator in between). Providing an on-chip buffer alleviates the pin count, bandwidth, and pad design requirements associated with sending trace data to a debugger directly through package pins in near real-time.

Features:

- compiled memory for the trace buffer and can be used as general purpose memory
- AMBA Peripheral Bus programming interface for configuration and memory access

12.6.4.1 AMBA Trace Bus (ATB) Replicator

The ATB Replicator enables two trace sinks (ETB and an off platform port generally connected to a Trace Port Interface Unit-TPIU) to be wired together and receive ATB trace data from the same trace source (PTM).

There are no programmable registers. It takes incoming trace data from a single source (PTM) and replicates it as multiple masters.

Chapter 13
AHB to IP Bridge (AIPSTZ)

13.1 Overview

This section provides an overview of the AHB to IP Bridge (AIPSTZ). This particular peripheral is designed as the bridge between AHB bus and peripherals with the lower bandwidth IP Slave (IPS) buses.

13.1.1 Features

The following list summarizes the key features of the bridge:

- The bridge supports the IPS slave bus signals. This interface is only meant for slave peripherals.
- The bridge supports 8-, 16-, and 32-bit IPS peripherals. (Accesses larger than the size of a peripheral are not supported, except to 32-bit memory.)
- The bridge supports a pair of IPS accesses for 64-bit and certain misaligned AHB transfers to 32-bit memory in 64-bit platforms.
- The bridge directly supports up to 32 16-Kbyte external IPS peripherals, and 2 global external IPS peripheral spaces. The bridge occupies 1 MBytes of total address space.
- The bridge provides configurable per-block and per-master access protections. More details on the protection features and configuration can be found in the Security Reference Manual
- Peripheral read transactions require a minimum of 2 hclk clocks, and unbuffered write transactions require a minimum of 3 hclk clocks.
- The bridge uses one single asynchronous reset and one global clock.

13.2 Clocks

The following table describes the clock sources for AIPSTZ. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 13-1. AIPSTZ Clocks

Clock name	Clock Root	Description
hclk	ahb_clk_root	Module clock

13.3 Functional Description

The AIPS bridge serves as a protocol translator between the AHB system bus and the IP bus.

Support is provided for generating a pair of 32-bit IP bus accesses when targeted by a 64-bit system bus access, or a misaligned access which crosses a 32-bit boundary. No other bus-sizing access support is provided.

The AHB to IP bridge is the interface between the AHB and on-chip IPS peripherals, which are sub-blocks containing readable/writable control and status registers.

The AHB master reads and writes these registers through the AIPSTZ. The bridge generates block enables, the block address, transfer attributes, byte enables and write data as inputs to the IPS peripherals. The bridge captures read data from the IPS interface and drives it on the AHB.

Each bridge that connects to the IPS (or peripherals) are referred as AIPS. The chip has several separate AIPS modules, and peripherals are grouped and assigned under each AIPS block. The list of peripherals are indicated as n-1, ... and n-*x* for AIPS-1, ... and AIPS-*x* respectively.

AIPS occupies a 1-Mbyte portion of the address space. The register maps of the IPS peripherals are located on 16-Kbyte boundaries. Each IPS peripheral is allocated one 16-Kbyte block of the memory map, and is activated by one of the block enables from the bridge. Up to thirty-two 16-Kbyte external IPS peripherals may be implemented, occupying contiguous blocks of 16-Kbytes. Two global external IPS block enables are available for the remaining address space to allow for customization and expansion of addressed peripheral devices. In addition, a single "non-global" block enable is also asserted whenever any of the thirty-two non-global block enables is asserted.

The bridge is responsible for indicating to IPS peripherals if an access is in supervisor or user mode. It may block user mode accesses to certain IPS peripherals or it may allow the individual IPS peripherals to determine if user mode accesses are allowed. In addition, peripherals may be designated as write-protected.

The bridge supports the notion of "trusted" masters for security purposes. Masters may be individually designated as trusted for reads, trusted for writes, or trusted for both reads and writes, as well as being forced to look as though all accesses from a master are in user-mode privilege level. Refer to AIPSTZ Memory Map/Register Definition for more information.

All peripheral devices are expected to only require aligned accesses equal to or smaller in size than the peripheral size. An exception to this rule is supported for 32-bit peripherals to allow memory to be placed on the IPS.

13.4 Access Protections

The AIPSTZ bridge provides programmable access protections for both masters and peripherals. It allows the privilege level of a master to be overridden, forcing it to user-mode privilege, and allows masters to be designated as trusted or untrusted.

Peripherals may require supervisor privilege level for access, may restrict access to a trusted master only, and may be write-protected. IP bus peripherals are subject to access control policies set in both CSU registers and AIPSTZ registers. An access is blocked if it is denied by either policy.

13.5 Access Support

Aligned 64-bit accesses, aligned and misaligned word and half word accesses, as well as byte accesses are supported for 32-bit peripherals. Misaligned accesses are supported to allow memory to be placed on the IPS.

Peripheral registers must not be misaligned, although no explicit checking is performed by the AIPS bridge. The bridge will perform two IPS transfers for 64-bit accesses, word accesses with byte offsets of 1, 2, or 3, and for half word accesses with a byte offset of 3. All other accesses will be performed with a single IPS transfer.

Only aligned half word and byte accesses are supported for 16-bit peripherals. All other accesses types are unsupported, and results of such accesses are undefined. They are not terminated with an error response.

Only byte accesses are supported for 8-bit peripherals. All other accesses types are unsupported, and results of such accesses are undefined. They are not terminated with an error response.

13.6 Initialization Information

The AIPS bridge should be programmed before use.

The following registers should be initialized: The Master Privilege Registers (AIPSTZ_MPRs), the Peripheral Access Control registers (AIPSTZ_PACRs), and the Off-platform Peripheral Access Control registers (AIPSTZ_OPACRs) described in AIPSTZ Memory Map/Register Definition.

13.6.1 Security Block

The AIPSTZ contains a security block that is connected to each off-platform peripheral. This block filters accesses based on write/read, non-secure, and supervisor signals.

Each peripheral can be individually configured to allow or deny each of the following transactions as described in the table below:

Table 13-2. Peripheral Access Configuration options

Config Bit	Write	Non-Secure	Supervisor	Meaning
0	0	0	0	Secure User Read
1	0	0	1	Secure Supervisor Read
2	0	1	0	Non-Secure User Read
3	0	1	1	Non-Secure Supervisor Read
4	1	0	0	Secure User Write
5	1	0	1	Secure Supervisor Write
6	1	1	0	Non-Secure User Write
7	1	1	1	Non-Secure Supervisor Write

Each peripheral has a security configuration (sec_config_X) input for determining whether to allow or deny a given access type. These are 8-bit vectors, with each bit corresponding to one of the transactions above as listed in the Config Bit column of Table 13-2. If the bit is asserted (1'b1), the transaction is allowed. If the bit is negated (1'b0), the transaction is not allowed.

For example, if peripheral 0 is configured as follows:

sec_config_0 [7:0] = 8'b0011_0011

This peripheral can only be accessed by secure transactions. Bits 0, 1, 4, and 5 are asserted and these bits refer to the four types of secure transactions. If an insecure transaction is attempted to this peripheral, it will result in an error.

Eight bits per peripheral across an entire system can result in a large number of configuration bits that must be assigned and controlled, most likely in a series of registers in another block. To reduce the number of register bits required predefined sets of security profiles can be defined and encapsulated in an external security translation block. The table below describes one set of security profiles that has been proposed for use with the AIPSTZ.

Table 13-3. Security Levels

CSU_SEC_LEVEL	Non-Secure User	Non-Secure Supervisor	Secure User	Secure Supervisor
0	RD+WR	RD+WR	RD+WR	RD+WR
1	NOT ALLOWED	RD+WR	RD+WR	RD+WR
2	Read Only	Read Only	RD+WR	RD+WR
3	NOT ALLOWED	Read Only	RD+WR	RD+WR
4	NOT ALLOWED	NOT ALLOWED	RD+WR	RD+WR
5	NOT ALLOWED	NOT ALLOWED	NOT ALLOWED	RD+WR
6	NOT ALLOWED	NOT ALLOWED	Read Only	Read Only
7	NOT ALLOWED	NOT ALLOWED	NOT ALLOWED	NOT ALLOWED

Information regarding CSU is provided in the Security Reference Manual. Contact your NXP representative for information about obtaining this document.

A 3-bit input, 8-bit output translation block can be used such that only three register bits are required to set the security profile and the translation block will drive the correct 8-bit configuration vector. Each peripheral connected to the AIPSTZ would require this translation block. The top level AIPSTZ has this three bit input line `csu_sec_level[2:0]' corresponding to each peripheral X.

13.7 Off-Platform Peripherals Index

The off-platform peripherals index allocation is shown in the following table.

Table 13-4. External IPS Peripherals

OPAC ID	AIPSTZ1	AIPSTZ2
OPAC31	Reserved	Reserved
OPAC30	LCDIF	I2C4

Table continues on the next page...

Table 13-4. External IPS Peripherals (continued)

OPAC ID	AIPSTZ1	AIPSTZ2
OPAC29	EPDC	UART5
OPAC28	PXP	UART4
OPAC27	SDMA	UART3
OPAC26	DCIC2	UART2
OPAC25	DCIC1	VDOA
OPAC24	IOMUXC	MIPI (DSI port)
OPAC23	PGC_ARM, PGC_PU, GPC	MIPI (CSI port)
OPAC22	SRC	AUDMUX
OPAC21	EPIT2	TZASC2
OPAC20	EPIT1	TZASC1
OPAC19	SNVS_HP	Perfmon3
OPAC18	USBPHY2, USBPHY1, ANALOG: (PLLs, PFDs, Regulators, LDOs, Temp Sensor)	Perfmon2
OPAC17	CCM	Perfmon1
OPAC16	WDOG2	CSU
OPAC15	WDOG1	OCOTP_CTRL
OPAC14	KPP	EIM
OPAC13	GPIO7	MMDC (port 1)
OPAC12	GPIO6	MMDC
OPAC11	GPIO5	ROMCP
OPAC10	GPIO4	I2C3
OPAC9	GPIO3	I2C2
OPAC8	GPIO2	I2C1
OPAC7	GPIO1	uSDHC4
OPAC6	GPT	uSDHC3
OPAC5	CAN2	uSDHC2
OPAC4	CAN1	uSDHC1
OPAC3	PWM4	MLB150
OPAC2	PWM3	ENET
OPAC1	PWM2	USBOH3 (USB)
OPAC0	PWM1	USBOH3 (pl301)
Global external IPS peripheral		
OPAC ID	**AIPSTZ1**	**AIPSTZ2**
OPAC33	VPU	ARM Cortex A9 MPCore Platform
OPAC32	SPBA Peripherals: ASRC, SSI3, SSI2, SSI1, ESAI, UART1, eCSPI4, eCSPI3, eCSPI2, eCSPI1, SPDIF.	CAAM

Table 13-5. External IPS Peripherals

OPAC ID	AIPSTZ1	AIPSTZ2	AIPSTZ3	AIPSTZ4
OPAC31	Reserved	QoSC	Reserved	Reserved
OPAC30	CSU	Reserved	ENET1	PLATFORM_CTRL
OPAC29	RDC	PERFMON2	SDMA	Reserved
OPAC28	SEMAPHORE2	PERFMON1	Reserved	Reserved
OPAC27	SEMAPHORE1	Reserved	QSPI	MTR
OPAC26	GPC	Reserved	Reserved	Reserved
OPAC25	SRC	Reserved	Reserved	Reserved
OPAC24	CCM	Reserved	CSI2	TZASC
OPAC23	SNVS_HP	Reserved	MIPI_CSI_PHY2	Reserved
OPAC22	ANA_PLL	Reserved	MIPI_CSI2	Reserved
OPAC21	OCOTP_CTRL	Reserved	uSDHC2	Reserved
OPAC20	IOMUXC_GPR	Reserved	uSDHC1	Reserved
OPAC19	IOMUXC	Reserved	Reserved	Reserved
OPAC18	LCDIF	Reserved	Reserved	Reserved
OPAC17	ROMCP	Reserved	Reserved	Reserved
OPAC16	Reserved	GPT4	Reserved	Reserved
OPAC15	GPT3	GPT5	Reserved	Reserved
OPAC14	GPT2	GPT6	Reserved	Reserved
OPAC13	GPT1	Reserved	Reserved	Reserved
OPAC12	SDMA2	System Counter_CTRL	SEMAPHORE-HS	Reserved
OPAC11	Reserved	System Counter_CMP	MU-B	Reserved
OPAC10	WDOG3	System Counter_RD	MU-A	Reserved
OPAC9	WDOG2	PWM4	CSI1	Reserved
OPAC8	WDOG1	PWM3	MIPI_CSI_PHY1	Reserved
OPAC7	ANA_OSC	PWM2	MIPI_CSI1	Reserved
OPAC6	ANA_TSENSOR	PWM1	UART4	Reserved
OPAC5	Reserved	Reserved	I2C4	Reserved
OPAC4	GPIO5	Reserved	I2C3	HDMI_SEC
OPAC3	GPIO4	Reserved	I2C2	DC_MST3
OPAC2	GPIO3	Reserved	I2C1	DC_MST2
OPAC1	GPIO2	Reserved	MIPI_DSI	DC_MST1
OPAC0	GPIO1	Reserved	MIPI_PHY	DC_MST0
Global external IPS peripheral				
OPAC ID	**AIPSTZ1**	**AIPSTZ2**	**AIPSTZ3**	**AIPSTZ4**
OPAC33	Reserved	Reserved	CAAM	Reserved
OPAC32	SPBA2: SAI4, SAI5, SAI6, SAI2, SAI1	Reserved	SPBA1: SAI3, SAI2, SPDIF2, UART2, UART3, UART1, eCSPI3, eCSPI2, eCSPI1, SPDIF1.	HDMI_CTRL

13.8 AIPSTZ Memory Map/Register Definition

The memory map for the AIPS SW-visible registers is shown in the table below.

The MPROT and OPACR fields are 4 bits in width. Some bits may be reserved depending on device.

AIPSTZ memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
207_C000	Master Priviledge Registers (AIPSTZ1_MPR)	32	R/W	7700_0000h	13.8.1/554
207_C040	Off-Platform Peripheral Access Control Registers (AIPSTZ1_OPACR)	32	R/W	4444_4444h	13.8.2/557
207_C044	Off-Platform Peripheral Access Control Registers (AIPSTZ1_OPACR1)	32	R/W	4444_4444h	13.8.3/560
207_C048	Off-Platform Peripheral Access Control Registers (AIPSTZ1_OPACR2)	32	R/W	4444_4444h	13.8.4/563
207_C04C	Off-Platform Peripheral Access Control Registers (AIPSTZ1_OPACR3)	32	R/W	4444_4444h	13.8.5/566
207_C050	Off-Platform Peripheral Access Control Registers (AIPSTZ1_OPACR4)	32	R/W	4444_4444h	13.8.6/569
217_C000	Master Priviledge Registers (AIPSTZ2_MPR)	32	R/W	7700_0000h	13.8.1/554
217_C040	Off-Platform Peripheral Access Control Registers (AIPSTZ2_OPACR)	32	R/W	4444_4444h	13.8.2/557
217_C044	Off-Platform Peripheral Access Control Registers (AIPSTZ2_OPACR1)	32	R/W	4444_4444h	13.8.3/560
217_C048	Off-Platform Peripheral Access Control Registers (AIPSTZ2_OPACR2)	32	R/W	4444_4444h	13.8.4/563
217_C04C	Off-Platform Peripheral Access Control Registers (AIPSTZ2_OPACR3)	32	R/W	4444_4444h	13.8.5/566
217_C050	Off-Platform Peripheral Access Control Registers (AIPSTZ2_OPACR4)	32	R/W	4444_4444h	13.8.6/569

13.8.1 Master Priviledge Registers (AIPSTZx_MPR)

Each AIPSTZ_MPR specifies 16 4-bit fields defining the access privilege level associated with a bus master in the platform, as well as specifying whether write accesses from this master are bufferable shown in Table 13-6

The registers provide one field per bus master, where field 15 corresponds to master 15, field 14 to master 14,... field 0 to master 0 (typically the processor core). The master index allocation is shown in the table below.

Table 13-6. MPROT Field

Bit	Field	Description
3	MBW	**Master Buffer Writes** - This bit determines whether the AIPSTZ is enabled to buffer writes from this master.
2	MTR	**Master Trusted for Reads** - This bit determines whether the master is trusted for read accesses.
1	MTW	**Master Trusted for Writes** - This bit determines whether the master is trusted for write accesses.
0	MPL	**Master Privilege Level** - This bit determines how the privilege level of the master is determined.

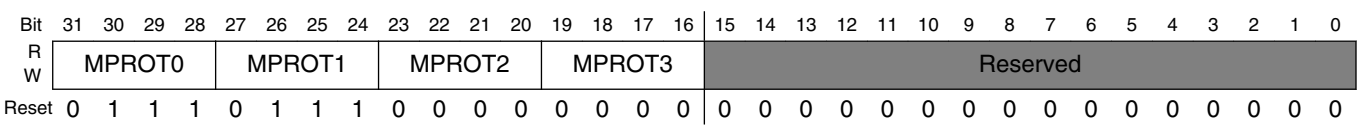
NOTE

The reset value is set to 0000_0000_7700_0000, which makes master 0 and master 1 (Arm CORE) the trusted masters. Trusted software can change the settings after reset.

Table 13-7. Master Index Allocation

Master Index	Master Name	Comments
Master 0	All masters excluding Arm core, SDMA and CAAM	Share the same number allocation.
Master 1	Arm CORE	
Master 2	CAAM	
Master 3	SDMA	
Master 4-15	Reserved	

Address: Base address + 0h offset



AIPSTZx_MPR field descriptions

Field	Description
31–28 MPROT0	Master 0 Priviledge, Buffer, Read, Write Control xxx0 **MPL0** — Accesses from this master are forced to user-mode (ips_supervisor_access is forced to zero) regardless of the hprot[1] access attribute. xxx1 **MPL1** — Accesses from this master are not forced to user-mode. The hprot[1] access attribute is used directly to determine ips_supervisor_access. xx0x **MTW0** — This master is not trusted for write accesses. xx1x **MTW1** — This master is trusted for write accesses. x0xx **MTR0** — This master is not trusted for read accesses.

Table continues on the next page...

AIPSTZx_MPR field descriptions (continued)

Field	Description
	x1xx **MTR1** — This master is trusted for read accesses. 0xxx **MBW0** — Write accesses from this master are not bufferable 1xxx **MBW1** — Write accesses from this master are allowed to be buffered
27–24 MPROT1	Master 1 Priviledge, Buffer, Read, Write Control xxx0 **MPL0** — Accesses from this master are forced to user-mode (ips_supervisor_access is forced to zero) regardless of the hprot[1] access attribute. xxx1 **MPL1** — Accesses from this master are not forced to user-mode. The hprot[1] access attribute is used directly to determine ips_supervisor_access. xx0x **MTW0** — This master is not trusted for write accesses. xx1x **MTW1** — This master is trusted for write accesses. x0xx **MTR0** — This master is not trusted for read accesses. x1xx **MTR1** — This master is trusted for read accesses. 0xxx **MBW0** — Write accesses from this master are not bufferable 1xxx **MBW1** — Write accesses from this master are allowed to be buffered
23–20 MPROT2	Master 2 Priviledge, Buffer, Read, Write Control xxx0 **MPL0** — Accesses from this master are forced to user-mode (ips_supervisor_access is forced to zero) regardless of the hprot[1] access attribute. xxx1 **MPL1** — Accesses from this master are not forced to user-mode. The hprot[1] access attribute is used directly to determine ips_supervisor_access. xx0x **MTW0** — This master is not trusted for write accesses. xx1x **MTW1** — This master is trusted for write accesses. x0xx **MTR0** — This master is not trusted for read accesses. x1xx **MTR1** — This master is trusted for read accesses. 0xxx **MBW0** — Write accesses from this master are not bufferable 1xxx **MBW1** — Write accesses from this master are allowed to be buffered
19–16 MPROT3	Master 3 Priviledge, Buffer, Read, Write Control. xxx0 **MPL0** — Accesses from this master are forced to user-mode (ips_supervisor_access is forced to zero) regardless of the hprot[1] access attribute. xxx1 **MPL1** — Accesses from this master are not forced to user-mode. The hprot[1] access attribute is used directly to determine ips_supervisor_access. xx0x **MTW0** — This master is not trusted for write accesses. xx1x **MTW1** — This master is trusted for write accesses. x0xx **MTR0** — This master is not trusted for read accesses. x1xx **MTR1** — This master is trusted for read accesses. 0xxx **MBW0** — Write accesses from this master are not bufferable 1xxx **MBW1** — Write accesses from this master are allowed to be buffered
-	This field is reserved. Reserved

13.8.2 Off-Platform Peripheral Access Control Registers (AIPSTZx_OPACR)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_OPACR has the following format shown in Table 13-8

Table 13-8. OPAC Field

Bit	Field	Description
3	BW	**Buffer Writes** - This bit determines whether write accesses to this peripheral are allowed to be buffered.[1]
2	SP	**Supervisor Protect** - This bit determines whether the peripheral requires supervisor privilege level for access.
1	WP	**Write Protect** - This bit determines whether the peripheral allows write accesses.
0	TP	**Trusted Protect** - This bit determines whether the peripheral allows accesses from an untrusted master.

1. Buffered writes are not available for AIPSTZ. This bit should be set to '0'.

Address: Base address + 40h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OPAC0				OPAC1				OPAC2				OPAC3				OPAC4				OPAC5				OPAC6				OPAC7			
Reset	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

AIPSTZx_OPACR field descriptions

Field	Description
31–28 OPAC0	Off-platform Peripheral Access Control 0 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
27–24 OPAC1	Off-platform Peripheral Access Control 1

Table continues on the next page...

AIPSTZx_OPACR field descriptions (continued)

Field	Description
	xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
23–20 OPAC2	Off-platform Peripheral Access Control 2 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
19–16 OPAC3	Off-platform Peripheral Access Control 3 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
15–12 OPAC4	Off-platform Peripheral Access Control 4 xxx0 **TP0** — Accesses from an untrusted master are allowed.

Table continues on the next page...

AIPSTZx_OPACR field descriptions (continued)

Field	Description
	xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
11–8 OPAC5	Off-platform Peripheral Access Control 5 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
7–4 OPAC6	Off-platform Peripheral Access Control 6 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
OPAC7	Off-platform Peripheral Access Control 7 xxx0 **TP0** — Accesses from an untrusted master are allowed.

Table continues on the next page...

AIPSTZx_OPACR field descriptions (continued)

Field	Description
	xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.

13.8.3 Off-Platform Peripheral Access Control Registers (AIPSTZx_OPACR1)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_OPACR has the following format shown in Table 13-8

Address: Base address + 44h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OPAC8				OPAC9				OPAC10				OPAC11				OPAC12				OPAC13				OPAC14				OPAC15			
Reset	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

AIPSTZx_OPACR1 field descriptions

Field	Description
31–28 OPAC8	Off-platform Peripheral Access Control 8 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus.

Table continues on the next page...

AIPSTZx_OPACR1 field descriptions (continued)

Field	Description
	0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
27–24 OPAC9	Off-platform Peripheral Access Control 9 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
23–20 OPAC10	Off-platform Peripheral Access Control 10 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
19–16 OPAC11	Off-platform Peripheral Access Control 11 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.

Table continues on the next page...

AIPSTZx_OPACR1 field descriptions (continued)

Field	Description
15–12 OPAC12	Off-platform Peripheral Access Control 12 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
11–8 OPAC13	Off-platform Peripheral Access Control 13 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
7–4 OPAC14	Off-platform Peripheral Access Control 14 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
OPAC15	Off-platform Peripheral Access Control 15

Table continues on the next page...

AIPSTZx_OPACR1 field descriptions (continued)

Field	Description
	xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.

13.8.4 Off-Platform Peripheral Access Control Registers (AIPSTZx_OPACR2)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_OPACR has the following format shown in Table 13-8

Address: Base address + 48h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OPAC16				OPAC17				OPAC18				OPAC19				OPAC20				OPAC21				OPAC22				OPAC23			
Reset	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

AIPSTZx_OPACR2 field descriptions

Field	Description
31–28 OPAC16	Off-platform Peripheral Access Control 16 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus.

Table continues on the next page...

AIPSTZx_OPACR2 field descriptions (continued)

Field	Description
	0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
27–24 OPAC17	Off-platform Peripheral Access Control 17 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
23–20 OPAC18	Off-platform Peripheral Access Control 18 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
19–16 OPAC19	Off-platform Peripheral Access Control 19 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.

Table continues on the next page...

AIPSTZx_OPACR2 field descriptions (continued)

Field	Description
15–12 OPAC20	Off-platform Peripheral Access Control 20 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
11–8 OPAC21	Off-platform Peripheral Access Control 21 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
7–4 OPAC22	Off-platform Peripheral Access Control 22 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
OPAC23	Off-platform Peripheral Access Control 23

Table continues on the next page...

AIPSTZx_OPACR2 field descriptions (continued)

Field	Description
	xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.

13.8.5 Off-Platform Peripheral Access Control Registers (AIPSTZx_OPACR3)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_OPACR has the following format shown in Table 13-8

Address: Base address + 4Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OPAC24				OPAC25				OPAC26				OPAC27				OPAC28				OPAC29				OPAC30				OPAC31			
Reset	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

AIPSTZx_OPACR3 field descriptions

Field	Description
31–28 OPAC24	Off-platform Peripheral Access Control 24 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus.

Table continues on the next page...

AIPSTZx_OPACR3 field descriptions (continued)

Field	Description
	0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
27–24 OPAC25	Off-platform Peripheral Access Control 25 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
23–20 OPAC26	Off-platform Peripheral Access Control 26 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
19–16 OPAC27	Off-platform Peripheral Access Control 27 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.

Table continues on the next page...

AIPSTZx_OPACR3 field descriptions (continued)

Field	Description
15–12 OPAC28	Off-platform Peripheral Access Control 28 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
11–8 OPAC29	Off-platform Peripheral Access Control 29 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
7–4 OPAC30	Off-platform Peripheral Access Control 30 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
OPAC31	Off-platform Peripheral Access Control 31

Table continues on the next page...

AIPSTZx_OPACR3 field descriptions (continued)

Field	Description
	xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.

13.8.6 Off-Platform Peripheral Access Control Registers (AIPSTZx_OPACR4)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_OPACR has the following format shown in Table 13-8

Address: Base address + 50h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OPAC32				OPAC33				Reserved																							
Reset	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

AIPSTZx_OPACR4 field descriptions

Field	Description
31–28 OPAC32	Off-platform Peripheral Access Control 32 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus.

Table continues on the next page...

AIPSTZx_OPACR4 field descriptions (continued)

Field	Description
	0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
27–24 OPAC33	Off-platform Peripheral Access Control 33 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
-	This field is reserved. Reserved

Chapter 14
AHB-to-APBH Bridge with DMA (APBH-Bridge-DMA)

14.1 Overview

The AHB-to-APBH bridge provides the chip with an inexpensive peripheral attachment bus running on the AHB's HCLK. (The H in APBH denotes that the APB*H* is synchronous to HCLK.)

As shown in the figure below, the AHB-to-APBH bridge includes the AHB-to-APB PIO bridge for a memory-mapped I/O to the APB devices, as well as a central DMA facility for devices on this bus and a vectored interrupt controller for the Arm core. Each one of the APB peripherals, including the vectored interrupt controller, is documented in their respective chapters.

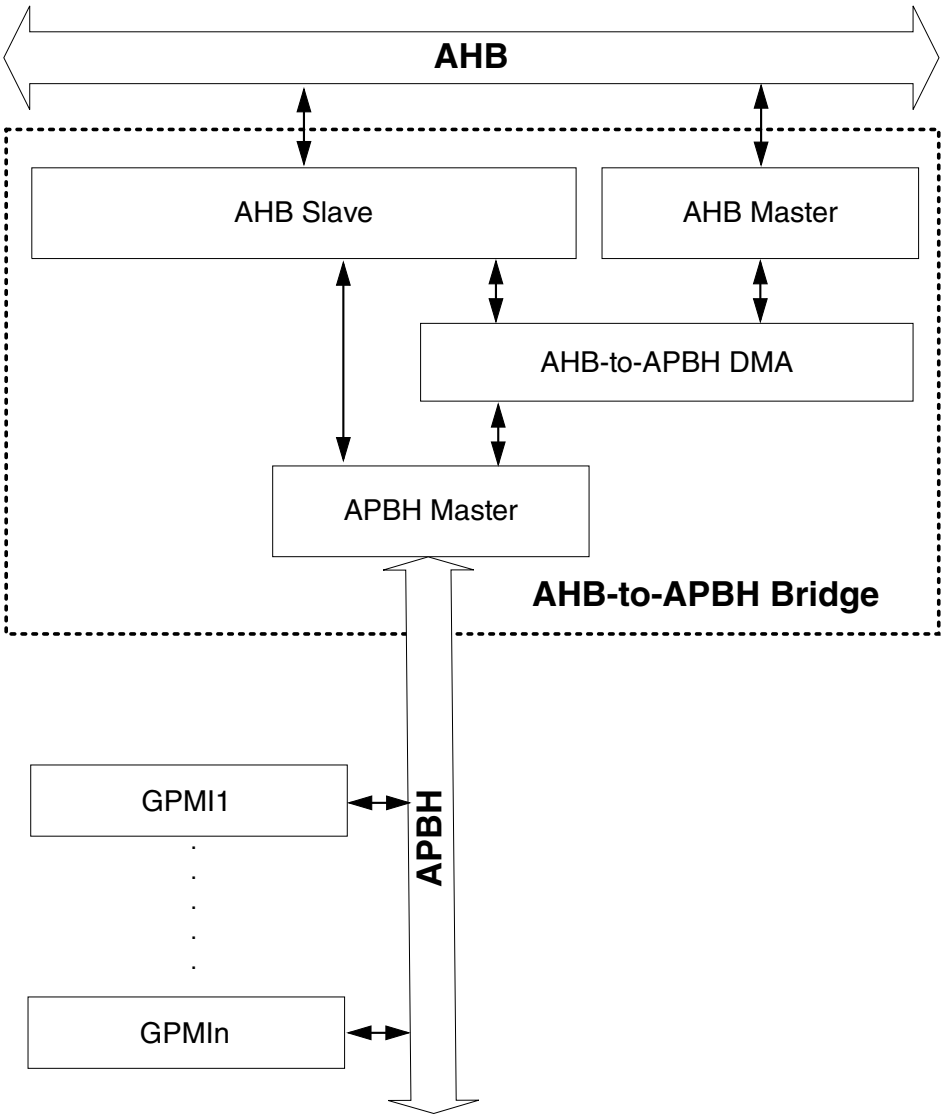


Figure 14-1. AHB-to-APBH Bridge DMA Block Diagram

The DMA controller uses the APBH bus to transfer read and write data to and from each peripheral. There is no separate DMA bus for these devices. Contention between the DMA's use of the APBH bus and the AHB-to-APB bridge functions' use of the APBH is mediated by an internal arbitration logic. For contention between these two units, the DMA is favored and the AHB slave will report "not ready" through its HREADY output until the bridge transfer can complete. The arbiter tracks repeated lockouts and inverts the priority, guaranteeing the Arm platform every fourth transfer on the APB.

14.2 Clocks

The table found here describes the clock sources for APBH. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 14-1. APBH Clocks

Clock name	Clock Root	Description
hclk	usdhc3_clk_root	Module clock

14.3 APBH DMA

The DMA supports four channels of DMA services, as shown in the following table. The shared DMA resource allows each independent channel to follow a simple chained command list. Command chains are built up using the general structure, as shown in Figure 14-2.

Table 14-2. APBH DMA channel assignments

APBH DMA Channel #	Usage
0	GPMI0
1	GPMI1
2	GPMI2
3	GPMI3

A single command structure or channel command word specifies a number of operations to be performed by the DMA in support of a given device. Thus, the Arm platform can set up large units of work, chaining together many DMA channel command words, pass them off to the DMA, and have no further concern for the device until the DMA completion interrupt occurs. The goal is to have enough intelligence in the DMA and the devices to keep the interrupt frequency from any device below 1 KHz (arrival intervals longer than 1 ms).

A single command structure can issue 32-bit PIO write operations to key registers in the associated device using the same APB bus and controls that it uses to write DMA data bytes to the device. For example, this allows a chain of operations to be issued to the GPMI controller to send NAND command bytes, address bytes, and data transfers where the command and the address structure is completely under software control, but the administration of that transfer is handled autonomously by the DMA. Each DMA structure can have 0–15 PIO words appended to it. The CMDPIOWORDS field, if non-zero, instructs the DMA engine to copy these words to the APB, beginning at the first register address offset for the peripheral and incrementing the register offset each cycle.

The DMA master generates only normal read/write transfers to the APBH. It does *not* generate set, clear, or toggle (SCT) transfers.

After any requested PIO words have been transferred to the peripheral, the DMA examines the two-bit command field in the channel command structure. Table 14-3 shows the four commands implemented by the DMA.

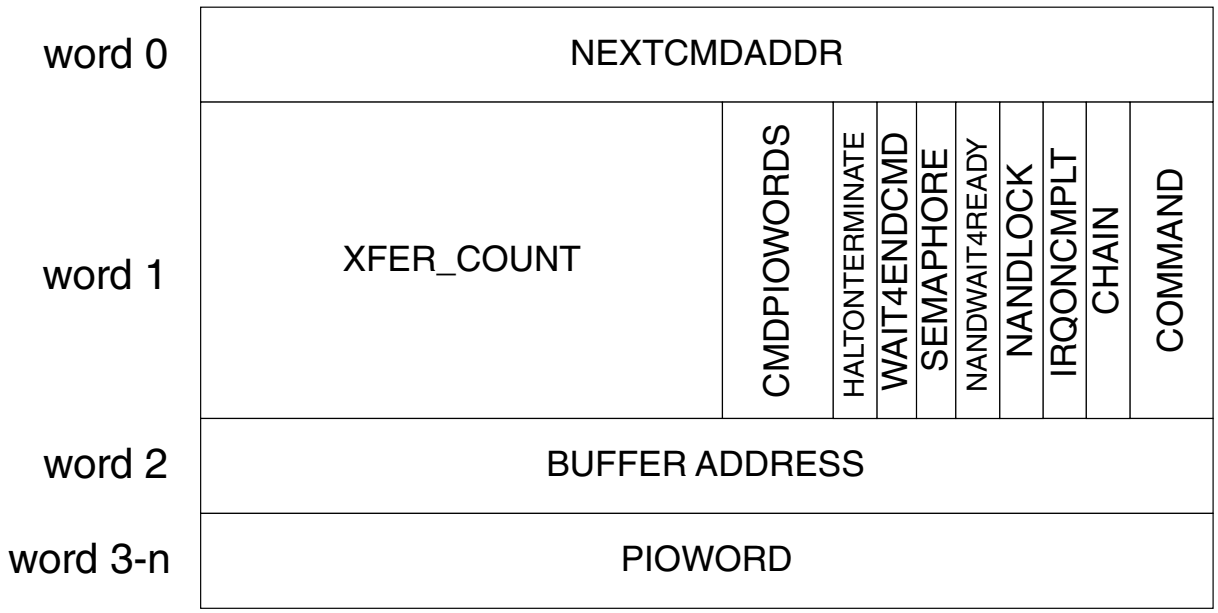


Figure 14-2. AHB-to-APBH Bridge DMA channel command structure

Table 14-3. APBH DMA commands

DMA Command	Usage
00	NO_DMA_XFER. Perform any requested PIO word transfers, but terminate the command before any DMA transfer.
01	DMA_WRITE. Perform any requested PIO word transfers, then perform a DMA transfer from the peripheral for the specified number of bytes.
10	DMA_READ. Perform any requested PIO word transfers and then perform a DMA transfer to the peripheral for the specified number of bytes.
11	DMA_SENSE. Perform any requested PIO word transfers, then perform a conditional branch to the next chained device. Follow the NEXTCMD_ADDR pointer if the peripheral sense is false. Follow the BUFFER_ADDRESS as a chain pointer if the peripheral sense line is true. This command becomes a no-operation for any channel other than a GPMI channel.

DMA_WRITE operations copy data bytes to the system memory (on-chip RAM or SDRAM) from the associated peripheral.

DMA_READ operations copy data bytes to the APB peripheral from the system memory. The DMA engine contains a shared byte aligner that aligns bytes from system memory to or from the peripherals. Peripherals always assume little-endian-aligned data arrives or departs on their 32-bit APB. The DMA_READ transfer uses the BUFFER_ADDRESS word in the command structure to point to the DMA data buffer to be read by the DMA_READ command.

The NO_DMA_XFER command is used to write PIO words to a device without performing any DMA data byte transfers. This command is useful in such applications as activating the NAND devices CHECKSTATUS operation. The check status command reads a status byte from the NAND device, performs an XOR and MASK against an expected value supplied as part of the PIO transfer. Once the read check completes (see NAND Read Status Polling Example), the NO_DMA_XFER command completes. The result in the peripheral is that its sense line is driven by the results of the comparison. The sense flip-flop is only updated by CHECKSTATUS for the device that is executed. At some future point, the chain contains a DMA command structure with the fourth and final command value, that is, the DMA_SENSE command.

As each DMA command completes, it triggers the DMA to load the next DMA command structure in the chain. The normal flow list of DMA commands is found by following the NEXTCMD_ADDR pointer in the DMA command structure. The DMA_SENSE command uses the DMA buffer pointer word of the command structure to point to an alternate DMA command structure chain or list. The DMA_SENSE command examines the sense line of the associated peripheral. If the sense line is false, then the DMA follows the standard list found whose next command is found from the pointer in the NEXTCMD_ADDR word of the command structure. If the sense line is true, then the DMA follows the alternate list whose next command is found from the pointer in the DMA Buffer Pointer word of the DMA_SENSE command structure (see Figure 14-2). The sense command ignores the CHAIN bit, so that both pointers must be valid when the DMA comes to a sense command.

If the wait-for-end-command bit (WAIT4ENDCMD) is set in a command structure, the DMA channel waits for the device to signal completion of a command by toggling the endcmd signal before proceeding to load and execute the next command structure. Then, if DECREMENT_SEMAPHORE is set, the semaphore is decremented after the end command is seen.

A detailed bit-field view of the DMA command structure is shown in the following table, which shows a field that specifies the number of bytes to be transferred by this DMA command. The transfer-count mechanism is duplicated in the associated peripheral, either as an implied or as a specified count in the peripheral.

Table 14-4. DMA channel command word in system memory

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
NEXT_COMMAND_ADDRESS																															

Table continues on the next page...

Table 14-4. DMA channel command word in system memory (continued)

Number DMA Bytes to Transfer	Number PIO Words to Write				HALTONTERMINATE	WAIT4ENDCMD	DECREMENT SEMAPHORE	NANDwAIT4READY	NANDLOCK	IRQ_COMPLETE	CHAIN	COMMAND	
DMA Buffer or Alternate CCW													
Zero or More PIO Words to Write to the Associated Peripheral Starting at its Base Address on the APBH Bus													

Figure 14-2 also shows the CHAIN bit in bit 2 of the second word of the command structure. This bit is set to 1, if the NEXT_COMMAND_ADDRESS contains a pointer to another DMA command structure. If a null pointer (0) is loaded into the NEXT_COMMAND_ADDRESS, it is not detected by the DMA hardware. Only the CHAIN bit indicates whether a valid list exists beyond the current structure.

If the IRQ_COMPLETE bit is set in the command structure, then the last act of the DMA before loading the next command is to set the interrupt-status bit corresponding to the current channel. The sticky interrupt request bit in the DMA CSR remains set until cleared by the software. It can be used to interrupt the Arm platform.

The NAND_LOCK bit is monitored by the DMA channel arbiter. Once a NAND channel (from channel 0 to channel 3) succeeds in the arbiter with its NAND_LOCK bit set, then the arbiter ignores the other NAND channels until a command is completed in which the NAND_LOCK is not set. Notice that the semantic here is that the NAND_LOCK state is to limit scheduling of a non-locked DMA. A DMA channel can go from unlocked to locked in the arbiter at the beginning of a command when the NAND_LOCK bit is set. When the last DMA command of an atomic sequence is completed, the lock should be removed. To accomplish this, the last command does not have the NAND_LOCK bit. It is still locked in the atomic state within the arbiter when the command starts, so that it is the only NAND command that can be executed. At the end, it drops from the atomic state within the arbiter.

The NAND_WAIT4READY bit also has a special use for GPMI channels (from channel 0 to channel 3), i.e., the NAND device channels. The GPMI peripheral supplies a sample of the ready line from the NAND device. This ready value is used to hold off of a command with this bit set until the ready line is asserted to 1. Once the arbiter sees a command with a wait-for-ready set, it holds off that channel until ready is asserted.

Receiving an IRQ for HALTONTERMINATE (HOT) is a feature in the APBH DMA descriptor that allows GPMI to signal to the DMA engine that an error has occurred. If a command is stalled due to an error, a HOT signal is sent from the peripheral to the DMA engine and causes an IRQ after terminating the DMA descriptor being executed.

Therefore, it is recommended that software use this signal as follows:

- Always set HALTONTERMINATE to 1 in a DMA descriptor. That way, if a peripheral signals HOT, the transfer will end, leaving the peripheral block and the DMA engine synchronized (but at the end of a command).
- When an IRQ from an APBH channel is received, and the IRQ is determined to be due to an error (as opposed to an IRQONCOMPLETE interrupt) the software should:
  - Reset the channel.
  - Determine the error from error reporting in the peripheral block, then manage the error in the peripheral that is attached to that channel in whatever appropriate way exists for that device (software recovery, device reset, block reset, etc).

Each channel has an eight-bit counting semaphore that controls whether it is in the idle state. When the semaphore is non-zero, the channel is ready to run, process commands and perform DMA transfers. Whenever a command finishes its DMA transfer, it checks the DECREMENT_SEMAPHORE bit. If set, it decrements the counting semaphore. If the semaphore goes to 0 as a result, then the channel enters the idle state and remains there until the semaphore is incremented by the software. When the semaphore goes to non-zero and the channel is in its idle state, then it uses the value in the APBH_CHn_NXTCMDAR register (next command address register) to fetch a pointer to the next command to process.

NOTE

This is a double indirect case. This method allows the software to append to a running command list under the protection of the counting semaphore.

To start processing the first time, software creates the command list to be processed. It writes the address of the first command into the APBH_CHn_NXTCMDAR register, and then writes 1 to the counting semaphore in APBH_CHn_SEMA. The DMA channel loads APBH_CHn_CURCMDAR register and then enters the normal state machine processing for the next command. When the software writes a value to the counting semaphore, it is added to the semaphore count by hardware, protecting the case where both hardware and software are trying to change the semaphore on the same clock edge.

Software can examine the value of APBH_CHn_CURCMDAR at any time to determine the location of the command structure currently being processed.

14.4 NAND Read Status Polling Example

The following figure shows a more complicated scenario.

This subset of a NAND device workload shows that the first two command structures are used during the data-write phase of an NAND device write operation (CLE and ALE transfers omitted for clarity).

- After writing the data, one must wait until the NAND device status register indicates that the write charge has been transferred. This is built into the workload using a check status command in the NAND in a loop created from the next two DMA command structures.
- The NO_DMA_TRANSFER command is shown here performing the read check, followed by a DMA_SENSE command to branch the DMA command structure list, based on the status of a bit in the external NAND device.

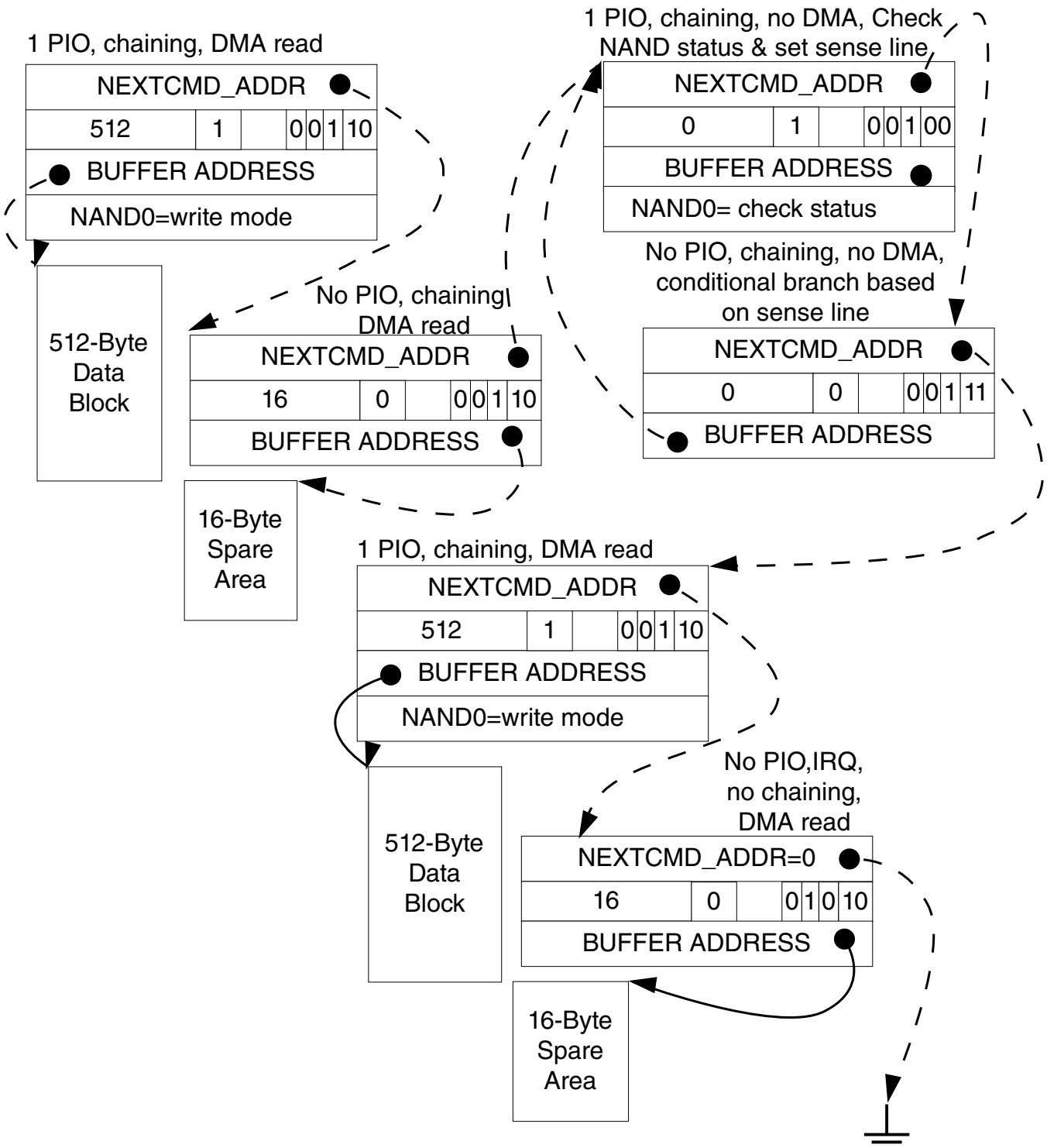


Figure 14-3. AHB-to-APBH Bridge DMA NAND Read Status Polling with DMA Sense Command

The example in the above figure shows the workload continuing immediately to the next NAND page transfer. However, one could perform a second sense operation to see if an error has occurred after the write. One could then point the sense command alternate branch at a NO_DMA_XFER command with the interrupt bit set. If the CHAIN bit is not

set on this failure branch, then the Arm platform is interrupted immediately, and the channel process is also immediately terminated in the presence of a workload-detected NAND error bit.

Note that each word of the three-word DMA command structure corresponds to a PIO register of the DMA that is accessible on the APBH bus. Normally, the DMA copies the next command structure onto these registers for processing at the start of each command by following the value of the pointer previously loaded into the NEXTCMD_ADDR register.

To start DMA processing for the first command, initialize the PIO registers of the desired channel, as follows:

- First, load the next command address register with a pointer to the first command to be loaded.
- Then, write 1 to the counting semaphore register. This causes the DMA to schedule the targeted channel for the DMA command structure load, just as if it had finished its previous command.

14.5 APBH Memory Map/Register Definition

APBH Hardware Register Format Summary

APBH memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
11_0000	AHB to APBH Bridge Control and Status Register 0 (APBH_CTRL0)	32	R/W	E000_0000h	14.5.1/586
11_0004	AHB to APBH Bridge Control and Status Register 0 (APBH_CTRL0_SET)	32	R/W	E000_0000h	14.5.1/586
11_0008	AHB to APBH Bridge Control and Status Register 0 (APBH_CTRL0_CLR)	32	R/W	E000_0000h	14.5.1/586
11_000C	AHB to APBH Bridge Control and Status Register 0 (APBH_CTRL0_TOG)	32	R/W	E000_0000h	14.5.1/586
11_0010	AHB to APBH Bridge Control and Status Register 1 (APBH_CTRL1)	32	R/W	0000_0000h	14.5.2/588
11_0014	AHB to APBH Bridge Control and Status Register 1 (APBH_CTRL1_SET)	32	R/W	0000_0000h	14.5.2/588
11_0018	AHB to APBH Bridge Control and Status Register 1 (APBH_CTRL1_CLR)	32	R/W	0000_0000h	14.5.2/588

Table continues on the next page...

APBH memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
11_001C	AHB to APBH Bridge Control and Status Register 1 (APBH_CTRL1_TOG)	32	R/W	0000_0000h	14.5.2/588
11_0020	AHB to APBH Bridge Control and Status Register 2 (APBH_CTRL2)	32	R/W	0000_0000h	14.5.3/591
11_0024	AHB to APBH Bridge Control and Status Register 2 (APBH_CTRL2_SET)	32	R/W	0000_0000h	14.5.3/591
11_0028	AHB to APBH Bridge Control and Status Register 2 (APBH_CTRL2_CLR)	32	R/W	0000_0000h	14.5.3/591
11_002C	AHB to APBH Bridge Control and Status Register 2 (APBH_CTRL2_TOG)	32	R/W	0000_0000h	14.5.3/591
11_0030	AHB to APBH Bridge Channel Register (APBH_CHANNEL_CTRL)	32	R/W	0000_0000h	14.5.4/596
11_0034	AHB to APBH Bridge Channel Register (APBH_CHANNEL_CTRL_SET)	32	R/W	0000_0000h	14.5.4/596
11_0038	AHB to APBH Bridge Channel Register (APBH_CHANNEL_CTRL_CLR)	32	R/W	0000_0000h	14.5.4/596
11_003C	AHB to APBH Bridge Channel Register (APBH_CHANNEL_CTRL_TOG)	32	R/W	0000_0000h	14.5.4/596
11_0040	AHB to APBH DMA Device Assignment Register (APBH_DEVSEL)	32	R	0000_0000h	14.5.5/597
11_0050	AHB to APBH DMA burst size (APBH_DMA_BURST_SIZE)	32	R/W	0055_5555h	14.5.6/598
11_0060	AHB to APBH DMA Debug Register (APBH_DEBUG)	32	R/W	0000_0000h	14.5.7/599
11_0100	APBH DMA Channel n Current Command Address Register (APBH_CH0_CURCMDAR)	32	R	0000_0000h	14.5.8/600
11_0110	APBH DMA Channel n Next Command Address Register (APBH_CH0_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_0120	APBH DMA Channel n Command Register (APBH_CH0_CMD)	32	R	0000_0000h	14.5.10/ 601
11_0130	APBH DMA Channel n Buffer Address Register (APBH_CH0_BAR)	32	R	0000_0000h	14.5.11/ 603
11_0140	APBH DMA Channel n Semaphore Register (APBH_CH0_SEMA)	32	R/W	0000_0000h	14.5.12/ 604
11_0150	AHB to APBH DMA Channel n Debug Information (APBH_CH0_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605
11_0160	AHB to APBH DMA Channel n Debug Information (APBH_CH0_DEBUG2)	32	R	0000_0000h	14.5.14/ 608
11_0170	APBH DMA Channel n Current Command Address Register (APBH_CH1_CURCMDAR)	32	R	0000_0000h	14.5.8/600
11_0180	APBH DMA Channel n Next Command Address Register (APBH_CH1_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_0190	APBH DMA Channel n Command Register (APBH_CH1_CMD)	32	R	0000_0000h	14.5.10/ 601
11_01A0	APBH DMA Channel n Buffer Address Register (APBH_CH1_BAR)	32	R	0000_0000h	14.5.11/ 603

Table continues on the next page...

APBH memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
11_01B0	APBH DMA Channel n Semaphore Register (APBH_CH1_SEMA)	32	R/W	0000_0000h	14.5.12/ 604
11_01C0	AHB to APBH DMA Channel n Debug Information (APBH_CH1_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605
11_01D0	AHB to APBH DMA Channel n Debug Information (APBH_CH1_DEBUG2)	32	R	0000_0000h	14.5.14/ 608
11_01E0	APBH DMA Channel n Current Command Address Register (APBH_CH2_CURCMDAR)	32	R	0000_0000h	14.5.8/600
11_01F0	APBH DMA Channel n Next Command Address Register (APBH_CH2_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_0200	APBH DMA Channel n Command Register (APBH_CH2_CMD)	32	R	0000_0000h	14.5.10/ 601
11_0210	APBH DMA Channel n Buffer Address Register (APBH_CH2_BAR)	32	R	0000_0000h	14.5.11/ 603
11_0220	APBH DMA Channel n Semaphore Register (APBH_CH2_SEMA)	32	R/W	0000_0000h	14.5.12/ 604
11_0230	AHB to APBH DMA Channel n Debug Information (APBH_CH2_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605
11_0240	AHB to APBH DMA Channel n Debug Information (APBH_CH2_DEBUG2)	32	R	0000_0000h	14.5.14/ 608
11_0250	APBH DMA Channel n Current Command Address Register (APBH_CH3_CURCMDAR)	32	R	0000_0000h	14.5.8/600
11_0260	APBH DMA Channel n Next Command Address Register (APBH_CH3_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_0270	APBH DMA Channel n Command Register (APBH_CH3_CMD)	32	R	0000_0000h	14.5.10/ 601
11_0280	APBH DMA Channel n Buffer Address Register (APBH_CH3_BAR)	32	R	0000_0000h	14.5.11/ 603
11_0290	APBH DMA Channel n Semaphore Register (APBH_CH3_SEMA)	32	R/W	0000_0000h	14.5.12/ 604
11_02A0	AHB to APBH DMA Channel n Debug Information (APBH_CH3_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605
11_02B0	AHB to APBH DMA Channel n Debug Information (APBH_CH3_DEBUG2)	32	R	0000_0000h	14.5.14/ 608
11_02C0	APBH DMA Channel n Current Command Address Register (APBH_CH4_CURCMDAR)	32	R	0000_0000h	14.5.8/600
11_02D0	APBH DMA Channel n Next Command Address Register (APBH_CH4_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_02E0	APBH DMA Channel n Command Register (APBH_CH4_CMD)	32	R	0000_0000h	14.5.10/ 601
11_02F0	APBH DMA Channel n Buffer Address Register (APBH_CH4_BAR)	32	R	0000_0000h	14.5.11/ 603
11_0300	APBH DMA Channel n Semaphore Register (APBH_CH4_SEMA)	32	R/W	0000_0000h	14.5.12/ 604

Table continues on the next page...

APBH memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
11_0310	AHB to APBH DMA Channel n Debug Information (APBH_CH4_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605
11_0320	AHB to APBH DMA Channel n Debug Information (APBH_CH4_DEBUG2)	32	R	0000_0000h	14.5.14/ 608
11_0330	APBH DMA Channel n Current Command Address Register (APBH_CH5_CURCMDAR)	32	R	0000_0000h	14.5.8/600
11_0340	APBH DMA Channel n Next Command Address Register (APBH_CH5_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_0350	APBH DMA Channel n Command Register (APBH_CH5_CMD)	32	R	0000_0000h	14.5.10/ 601
11_0360	APBH DMA Channel n Buffer Address Register (APBH_CH5_BAR)	32	R	0000_0000h	14.5.11/ 603
11_0370	APBH DMA Channel n Semaphore Register (APBH_CH5_SEMA)	32	R/W	0000_0000h	14.5.12/ 604
11_0380	AHB to APBH DMA Channel n Debug Information (APBH_CH5_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605
11_0390	AHB to APBH DMA Channel n Debug Information (APBH_CH5_DEBUG2)	32	R	0000_0000h	14.5.14/ 608
11_03A0	APBH DMA Channel n Current Command Address Register (APBH_CH6_CURCMDAR)	32	R	0000_0000h	14.5.8/600
11_03B0	APBH DMA Channel n Next Command Address Register (APBH_CH6_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_03C0	APBH DMA Channel n Command Register (APBH_CH6_CMD)	32	R	0000_0000h	14.5.10/ 601
11_03D0	APBH DMA Channel n Buffer Address Register (APBH_CH6_BAR)	32	R	0000_0000h	14.5.11/ 603
11_03E0	APBH DMA Channel n Semaphore Register (APBH_CH6_SEMA)	32	R/W	0000_0000h	14.5.12/ 604
11_03F0	AHB to APBH DMA Channel n Debug Information (APBH_CH6_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605
11_0400	AHB to APBH DMA Channel n Debug Information (APBH_CH6_DEBUG2)	32	R	0000_0000h	14.5.14/ 608
11_0410	APBH DMA Channel n Current Command Address Register (APBH_CH7_CURCMDAR)	32	R	0000_0000h	14.5.8/600
11_0420	APBH DMA Channel n Next Command Address Register (APBH_CH7_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_0430	APBH DMA Channel n Command Register (APBH_CH7_CMD)	32	R	0000_0000h	14.5.10/ 601
11_0440	APBH DMA Channel n Buffer Address Register (APBH_CH7_BAR)	32	R	0000_0000h	14.5.11/ 603
11_0450	APBH DMA Channel n Semaphore Register (APBH_CH7_SEMA)	32	R/W	0000_0000h	14.5.12/ 604
11_0460	AHB to APBH DMA Channel n Debug Information (APBH_CH7_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605

Table continues on the next page...

APBH memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
11_0470	AHB to APBH DMA Channel n Debug Information (APBH_CH7_DEBUG2)	32	R	0000_0000h	14.5.14/ 608
11_0480	APBH DMA Channel n Current Command Address Register (APBH_CH8_CURCMDAR)	32	R	0000_0000h	14.5.8/600
11_0490	APBH DMA Channel n Next Command Address Register (APBH_CH8_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_04A0	APBH DMA Channel n Command Register (APBH_CH8_CMD)	32	R	0000_0000h	14.5.10/ 601
11_04B0	APBH DMA Channel n Buffer Address Register (APBH_CH8_BAR)	32	R	0000_0000h	14.5.11/ 603
11_04C0	APBH DMA Channel n Semaphore Register (APBH_CH8_SEMA)	32	R/W	0000_0000h	14.5.12/ 604
11_04D0	AHB to APBH DMA Channel n Debug Information (APBH_CH8_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605
11_04E0	AHB to APBH DMA Channel n Debug Information (APBH_CH8_DEBUG2)	32	R	0000_0000h	14.5.14/ 608
11_04F0	APBH DMA Channel n Current Command Address Register (APBH_CH9_CURCMDAR)	32	R	0000_0000h	14.5.8/600
11_0500	APBH DMA Channel n Next Command Address Register (APBH_CH9_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_0510	APBH DMA Channel n Command Register (APBH_CH9_CMD)	32	R	0000_0000h	14.5.10/ 601
11_0520	APBH DMA Channel n Buffer Address Register (APBH_CH9_BAR)	32	R	0000_0000h	14.5.11/ 603
11_0530	APBH DMA Channel n Semaphore Register (APBH_CH9_SEMA)	32	R/W	0000_0000h	14.5.12/ 604
11_0540	AHB to APBH DMA Channel n Debug Information (APBH_CH9_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605
11_0550	AHB to APBH DMA Channel n Debug Information (APBH_CH9_DEBUG2)	32	R	0000_0000h	14.5.14/ 608
11_0560	APBH DMA Channel n Current Command Address Register (APBH_CH10_CURCMDAR)	32	R	0000_0000h	14.5.8/600
11_0570	APBH DMA Channel n Next Command Address Register (APBH_CH10_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_0580	APBH DMA Channel n Command Register (APBH_CH10_CMD)	32	R	0000_0000h	14.5.10/ 601
11_0590	APBH DMA Channel n Buffer Address Register (APBH_CH10_BAR)	32	R	0000_0000h	14.5.11/ 603
11_05A0	APBH DMA Channel n Semaphore Register (APBH_CH10_SEMA)	32	R/W	0000_0000h	14.5.12/ 604
11_05B0	AHB to APBH DMA Channel n Debug Information (APBH_CH10_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605
11_05C0	AHB to APBH DMA Channel n Debug Information (APBH_CH10_DEBUG2)	32	R	0000_0000h	14.5.14/ 608

Table continues on the next page...

APBH memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
11_05D0	APBH DMA Channel n Current Command Address Register (APBH_CH11_CURCMDAR)	32	R	0000_0000h	14.5.8/600
11_05E0	APBH DMA Channel n Next Command Address Register (APBH_CH11_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_05F0	APBH DMA Channel n Command Register (APBH_CH11_CMD)	32	R	0000_0000h	14.5.10/ 601
11_0600	APBH DMA Channel n Buffer Address Register (APBH_CH11_BAR)	32	R	0000_0000h	14.5.11/ 603
11_0610	APBH DMA Channel n Semaphore Register (APBH_CH11_SEMA)	32	R/W	0000_0000h	14.5.12/ 604
11_0620	AHB to APBH DMA Channel n Debug Information (APBH_CH11_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605
11_0630	AHB to APBH DMA Channel n Debug Information (APBH_CH11_DEBUG2)	32	R	0000_0000h	14.5.14/ 608
11_0640	APBH DMA Channel n Current Command Address Register (APBH_CH12_CURCMDAR)	32	R	0000_0000h	14.5.8/600
11_0650	APBH DMA Channel n Next Command Address Register (APBH_CH12_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_0660	APBH DMA Channel n Command Register (APBH_CH12_CMD)	32	R	0000_0000h	14.5.10/ 601
11_0670	APBH DMA Channel n Buffer Address Register (APBH_CH12_BAR)	32	R	0000_0000h	14.5.11/ 603
11_0680	APBH DMA Channel n Semaphore Register (APBH_CH12_SEMA)	32	R/W	0000_0000h	14.5.12/ 604
11_0690	AHB to APBH DMA Channel n Debug Information (APBH_CH12_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605
11_06A0	AHB to APBH DMA Channel n Debug Information (APBH_CH12_DEBUG2)	32	R	0000_0000h	14.5.14/ 608
11_06B0	APBH DMA Channel n Current Command Address Register (APBH_CH13_CURCMDAR)	32	R	0000_0000h	14.5.8/600
11_06C0	APBH DMA Channel n Next Command Address Register (APBH_CH13_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_06D0	APBH DMA Channel n Command Register (APBH_CH13_CMD)	32	R	0000_0000h	14.5.10/ 601
11_06E0	APBH DMA Channel n Buffer Address Register (APBH_CH13_BAR)	32	R	0000_0000h	14.5.11/ 603
11_06F0	APBH DMA Channel n Semaphore Register (APBH_CH13_SEMA)	32	R/W	0000_0000h	14.5.12/ 604
11_0700	AHB to APBH DMA Channel n Debug Information (APBH_CH13_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605
11_0710	AHB to APBH DMA Channel n Debug Information (APBH_CH13_DEBUG2)	32	R	0000_0000h	14.5.14/ 608
11_0720	APBH DMA Channel n Current Command Address Register (APBH_CH14_CURCMDAR)	32	R	0000_0000h	14.5.8/600

Table continues on the next page...

APBH memory map (continued)

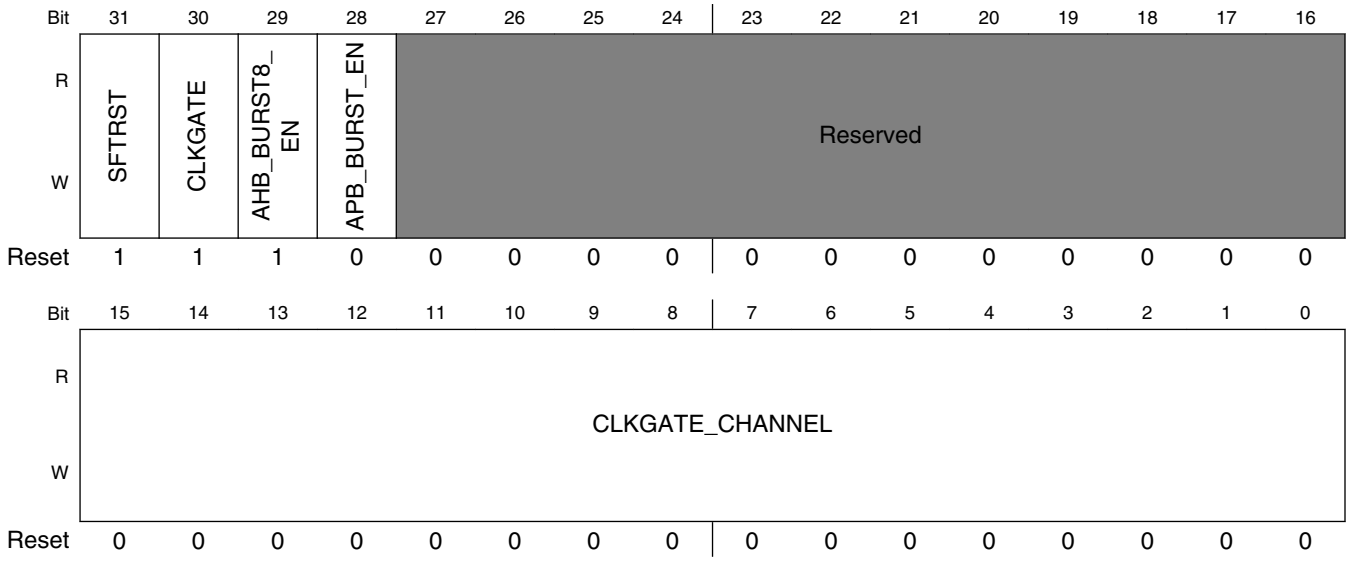
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
11_0730	APBH DMA Channel n Next Command Address Register (APBH_CH14_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_0740	APBH DMA Channel n Command Register (APBH_CH14_CMD)	32	R	0000_0000h	14.5.10/ 601
11_0750	APBH DMA Channel n Buffer Address Register (APBH_CH14_BAR)	32	R	0000_0000h	14.5.11/ 603
11_0760	APBH DMA Channel n Semaphore Register (APBH_CH14_SEMA)	32	R/W	0000_0000h	14.5.12/ 604
11_0770	AHB to APBH DMA Channel n Debug Information (APBH_CH14_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605
11_0780	AHB to APBH DMA Channel n Debug Information (APBH_CH14_DEBUG2)	32	R	0000_0000h	14.5.14/ 608
11_0790	APBH DMA Channel n Current Command Address Register (APBH_CH15_CURCMDAR)	32	R	0000_0000h	14.5.8/600
11_07A0	APBH DMA Channel n Next Command Address Register (APBH_CH15_NXTCMDAR)	32	R/W	0000_0000h	14.5.9/601
11_07B0	APBH DMA Channel n Command Register (APBH_CH15_CMD)	32	R	0000_0000h	14.5.10/ 601
11_07C0	APBH DMA Channel n Buffer Address Register (APBH_CH15_BAR)	32	R	0000_0000h	14.5.11/ 603
11_07D0	APBH DMA Channel n Semaphore Register (APBH_CH15_SEMA)	32	R/W	0000_0000h	14.5.12/ 604
11_07E0	AHB to APBH DMA Channel n Debug Information (APBH_CH15_DEBUG1)	32	R	00A0_0000h	14.5.13/ 605
11_07F0	AHB to APBH DMA Channel n Debug Information (APBH_CH15_DEBUG2)	32	R	0000_0000h	14.5.14/ 608
11_0800	APBH Bridge Version Register (APBH_VERSION)	32	R	0301_0000h	14.5.15/ 609

14.5.1 AHB to APBH Bridge Control and Status Register 0 (APBH_CTRL0*n*)

The APBH CTRL 0 provides overall control of the AHB to APBH bridge and DMA.

This register contains module softreset, clock gating, channel clock gating/freeze bits.

Address: 11_0000h base + 0h offset + (4d × i), where i=0d to 3d



APBH_CTRL0*n* field descriptions

Field	Description
31 SFTRST	Set this bit to zero to enable normal APBH DMA operation. Set this bit to one (default) to disable clocking with the APBH DMA and hold it in its reset (lowest power) state. This bit can be turned on and then off to reset the APBH DMA block to its default state.
30 CLKGATE	This bit must be set to zero for normal operation. When set to one it gates off the clocks to the block.
29 AHB_BURST8_EN	Set this bit to one (default) to enable AHB 8-beat burst. Set to zero to disable 8-beat burst on AHB interface.
28 APB_BURST_EN	Set this bit to one to enable apb master do a continous transfers when a device request a burst dma. Set to zero will treat a burst dma request as 4/8 individual requests.
27–16 RSVD0	This field is reserved. Reserved, always set to zero.
CLKGATE_CHANNEL	These bits must be set to zero for normal operation of each channel. When set to one they gate off the individual clocks to the channels. 0x0001 **NAND0** — 0x0002 **NAND1** — 0x0004 **NAND2** — 0x0008 **NAND3** — 0x0010 **NAND4** — 0x0020 **NAND5** — 0x0040 **NAND6** — 0x0080 **NAND7** — 0x0100 **SSP** —

14.5.2 AHB to APBH Bridge Control and Status Register 1 (APBH_CTRL1*n*)

The APBH CTRL one provides overall control of the interrupts generated by the AHB to APBH DMA. This register contains the per channel interrupt status bits and the per channel interrupt enable bits. Each channel has a dedicated interrupt vector in the vectored interrupt controller.

EXAMPLE

```
                    BF_WR(APBH_CTRL1, CH5_CMDCMPLT_IRQ, 0);  // use bitfield write macro
                    BF_APBH_CTRL1.CH5_CMDCMPLT_IRQ = 0;      // or, assign to register
struct's bitfield
```

Address: 11_0000h base + 10h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CH15_CMDCMPLT_IRQ_EN	CH14_CMDCMPLT_IRQ_EN	CH13_CMDCMPLT_IRQ_EN	CH12_CMDCMPLT_IRQ_EN	CH11_CMDCMPLT_IRQ_EN	CH10_CMDCMPLT_IRQ_EN	CH9_CMDCMPLT_IRQ_EN	CH8_CMDCMPLT_IRQ_EN	CH7_CMDCMPLT_IRQ_EN	CH6_CMDCMPLT_IRQ_EN	CH5_CMDCMPLT_IRQ_EN	CH4_CMDCMPLT_IRQ_EN	CH3_CMDCMPLT_IRQ_EN	CH2_CMDCMPLT_IRQ_EN	CH1_CMDCMPLT_IRQ_EN	CH0_CMDCMPLT_IRQ_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH15_CMDCMPLT_IRQ	CH14_CMDCMPLT_IRQ	CH13_CMDCMPLT_IRQ	CH12_CMDCMPLT_IRQ	CH11_CMDCMPLT_IRQ	CH10_CMDCMPLT_IRQ	CH9_CMDCMPLT_IRQ	CH8_CMDCMPLT_IRQ	CH7_CMDCMPLT_IRQ	CH6_CMDCMPLT_IRQ	CH5_CMDCMPLT_IRQ	CH4_CMDCMPLT_IRQ	CH3_CMDCMPLT_IRQ	CH2_CMDCMPLT_IRQ	CH1_CMDCMPLT_IRQ	CH0_CMDCMPLT_IRQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APBH_CTRL1*n* field descriptions

Field	Description
31 CH15_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 15.
30 CH14_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 14.

Table continues on the next page...

APBH_CTRL1*n* field descriptions (continued)

Field	Description
29 CH13_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 13.
28 CH12_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 12.
27 CH11_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 11.
26 CH10_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 10.
25 CH9_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 9.
24 CH8_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 8.
23 CH7_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 7.
22 CH6_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 6.
21 CH5_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 5.
20 CH4_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 4.
19 CH3_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 3.
18 CH2_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 2.

Table continues on the next page...

APBH_CTRL1*n* field descriptions (continued)

Field	Description
17 CH1_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 1.
16 CH0_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 0.
15 CH15_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 15. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
14 CH14_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 14. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
13 CH13_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 13. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
12 CH12_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 12. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
11 CH11_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 11. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
10 CH10_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 10. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
9 CH9_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 9. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
8 CH8_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 8. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
7 CH7_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 7. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
6 CH6_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 6. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.

Table continues on the next page...

APBH_CTRL1*n* field descriptions (continued)

Field	Description
5 CH5_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 5. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
4 CH4_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 4. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
3 CH3_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 3. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
2 CH2_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 2. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
1 CH1_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 1. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
0 CH0_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 0. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.

14.5.3 AHB to APBH Bridge Control and Status Register 2 (APBH_CTRL2*n*)

The APBH CTRL 2 provides channel error interrupts generated by the AHB to APBH DMA. This register contains the per channel interrupt status bits and the per channel interrupt enable bits. Each channel has a dedicated interrupt vector in the vectored interrupt controller.

EXAMPLE

```
                    BF_WR(APBH_CTRL1, CH5_CMDCMPLT_IRQ, 0);  // use bitfield write macro
                    BF_APBH_CTRL1.CH5_CMDCMPLT_IRQ = 0;      // or, assign to register
struct's bitfield
```

Address: 11_0000h base + 20h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CH15_ERROR_STATUS	CH14_ERROR_STATUS	CH13_ERROR_STATUS	CH12_ERROR_STATUS	CH11_ERROR_STATUS	CH10_ERROR_STATUS	CH9_ERROR_STATUS	CH8_ERROR_STATUS	CH7_ERROR_STATUS	CH6_ERROR_STATUS	CH5_ERROR_STATUS	CH4_ERROR_STATUS	CH3_ERROR_STATUS	CH2_ERROR_STATUS	CH1_ERROR_STATUS	CH0_ERROR_STATUS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH15_ERROR_IRQ	CH14_ERROR_IRQ	CH13_ERROR_IRQ	CH12_ERROR_IRQ	CH11_ERROR_IRQ	CH10_ERROR_IRQ	CH9_ERROR_IRQ	CH8_ERROR_IRQ	CH7_ERROR_IRQ	CH6_ERROR_IRQ	CH5_ERROR_IRQ	CH4_ERROR_IRQ	CH3_ERROR_IRQ	CH2_ERROR_IRQ	CH1_ERROR_IRQ	CH0_ERROR_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APBH_CTRL2*n* field descriptions

Field	Description
31 CH15_ERROR_ STATUS	Error status bit for APBH DMA Channel 15. Valid when corresponding Error IRQ is set. 1 - AHB bus error

Table continues on the next page...

APBH_CTRL2*n* field descriptions (continued)

Field	Description
	0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
30 CH14_ERROR_ STATUS	Error status bit for APBH DMA Channel 14. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
29 CH13_ERROR_ STATUS	Error status bit for APBH DMA Channel 13. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
28 CH12_ERROR_ STATUS	Error status bit for APBH DMA Channel 12. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
27 CH11_ERROR_ STATUS	Error status bit for APBH DMA Channel 11. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
26 CH10_ERROR_ STATUS	Error status bit for APBH DMA Channel 10. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
25 CH9_ERROR_ STATUS	Error status bit for APBH DMA Channel 9. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
24 CH8_ERROR_ STATUS	Error status bit for APBH DMA Channel 8. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination.

Table continues on the next page...

APBH_CTRL2*n* field descriptions (continued)

Field	Description
	0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
23 CH7_ERROR_STATUS	Error status bit for APBX DMA Channel 7. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
22 CH6_ERROR_STATUS	Error status bit for APBX DMA Channel 6. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
21 CH5_ERROR_STATUS	Error status bit for APBX DMA Channel 5. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
20 CH4_ERROR_STATUS	Error status bit for APBX DMA Channel 4. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
19 CH3_ERROR_STATUS	Error status bit for APBX DMA Channel 3. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
18 CH2_ERROR_STATUS	Error status bit for APBX DMA Channel 2. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
17 CH1_ERROR_STATUS	Error status bit for APBX DMA Channel 1. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.

Table continues on the next page...

APBH_CTRL2*n* field descriptions (continued)

Field	Description
16 CH0_ERROR_STATUS	Error status bit for APBX DMA Channel 0. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
15 CH15_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 15. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
14 CH14_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 14. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
13 CH13_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 13. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
12 CH12_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 12. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
11 CH11_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 11. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
10 CH10_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 10. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
9 CH9_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 9. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
8 CH8_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 8. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
7 CH7_ERROR_IRQ	Error interrupt status bit for APBX DMA Channel 7. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
6 CH6_ERROR_IRQ	Error interrupt status bit for APBX DMA Channel 6. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
5 CH5_ERROR_IRQ	Error interrupt status bit for APBX DMA Channel 5. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
4 CH4_ERROR_IRQ	Error interrupt status bit for APBX DMA Channel 4. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
3 CH3_ERROR_IRQ	Error interrupt status bit for APBX DMA Channel 3. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
2 CH2_ERROR_IRQ	Error interrupt status bit for APBX DMA Channel 2. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.

Table continues on the next page...

APBH_CTRL2*n* field descriptions (continued)

Field	Description
1 CH1_ERROR_ IRQ	Error interrupt status bit for APBX DMA Channel 1. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
0 CH0_ERROR_ IRQ	Error interrupt status bit for APBX DMA Channel 0. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.

14.5.4 AHB to APBH Bridge Channel Register (APBH_CHANNEL_CTRL*n*)

The APBH CHANNEL CTRL provides reset/freeze control of each DMA channel. This register contains individual channel reset/freeze bits.

Address: 11_0000h base + 30h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RESET_CHANNEL																FREEZE_CHANNEL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APBH_CHANNEL_CTRL*n* field descriptions

Field	Description
31–16 RESET_ CHANNEL	Setting a bit in this field causes the DMA controller to take the corresponding channel through its reset state. The bit is reset after the channel resources are cleared. 0x0001 **NAND0** — 0x0002 **NAND1** — 0x0004 **NAND2** — 0x0008 **NAND3** — 0x0010 **NAND4** — 0x0020 **NAND5** — 0x0040 **NAND6** — 0x0080 **NAND7** — 0x0100 **SSP** —
FREEZE_ CHANNEL	Setting a bit in this field will freeze the DMA channel associated with it. This field is a direct input to the DMA channel arbiter. When frozen, the channel is deined access to the central DMA resources. 0x0001 **NAND0** — 0x0002 **NAND1** — 0x0004 **NAND2** — 0x0008 **NAND3** — 0x0010 **NAND4** — 0x0020 **NAND5** — 0x0040 **NAND6** —

Table continues on the next page...

APBH_CHANNEL_CTRL*n* field descriptions (continued)

Field	Description
	0x0080 **NAND7** — 0x0100 **SSP** —

14.5.5 AHB to APBH DMA Device Assignment Register (APBH_DEVSEL)

This register allows reassignment of the APBH device connected to the DMA Channels.

In this chip, APBH DMA channel resource is enough for high speed peripherals, so this register is of no use and reserved.

Address: 11_0000h base + 40h offset = 11_0040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APBH_DEVSEL field descriptions

Field	Description
31–30 CH15	This field is reserved. Reserved.
29–28 CH14	This field is reserved. Reserved.
27–26 CH13	This field is reserved. Reserved.
25–24 CH12	This field is reserved. Reserved.
23–22 CH11	This field is reserved. Reserved.
21–20 CH10	This field is reserved. Reserved.
19–18 CH9	This field is reserved. Reserved.
17–16 CH8	This field is reserved. Reserved.
15–14 CH7	This field is reserved. Reserved.
13–12 CH6	This field is reserved. Reserved.

Table continues on the next page...

APBH_DEVSEL field descriptions (continued)

Field	Description
11–10 CH5	This field is reserved. Reserved.
9–8 CH4	This field is reserved. Reserved.
7–6 CH3	This field is reserved. Reserved.
5–4 CH2	This field is reserved. Reserved.
3–2 CH1	This field is reserved. Reserved.
CH0	This field is reserved. Reserved.

14.5.6 AHB to APBH DMA burst size (APBH_DMA_BURST_SIZE)

This register programs the apbh burst size of the APBH DMA devices when a DMA burst request is issued.

This register provides a mechanism for assigning the device.

Address: 11_0000h base + 50h offset = 11_0050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		CH8	
Reset	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH7		CH6		CH5		CH4		CH3		CH2		CH1		CH0	
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

APBH_DMA_BURST_SIZE field descriptions

Field	Description
31–30 CH15	This field is reserved. Reserved.
29–28 CH14	This field is reserved. Reserved.
27–26 CH13	This field is reserved. Reserved.
25–24 CH12	This field is reserved. Reserved.
23–22 CH11	This field is reserved. Reserved.

Table continues on the next page...

APBH_DMA_BURST_SIZE field descriptions (continued)

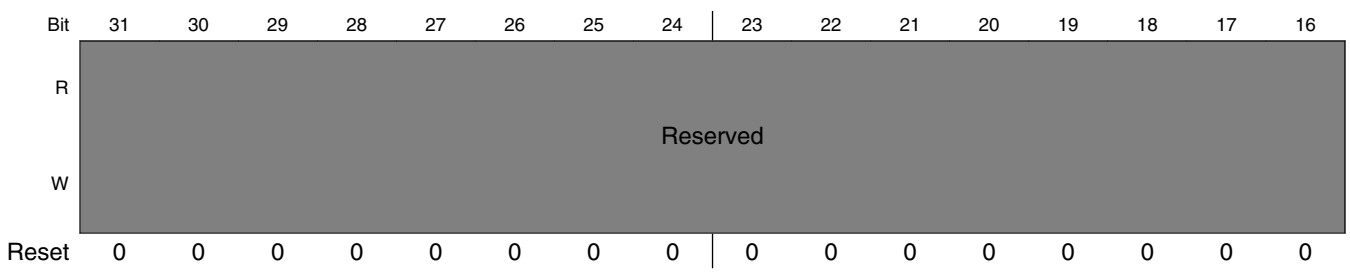
Field	Description
21–20 CH10	This field is reserved. Reserved.
19–18 CH9	This field is reserved. Reserved.
17–16 CH8	DMA burst size for SSP. 0x0 **BURST0** — 0x1 **BURST4** — 0x2 **BURST8** —
15–14 CH7	DMA burst size for GPMI channel 7. Do not change. GPMI only support burst size 4.
13–12 CH6	DMA burst size for GPMI channel 6. Do not change. GPMI only support burst size 4.
11–10 CH5	DMA burst size for GPMI channel 5. Do not change. GPMI only support burst size 4.
9–8 CH4	DMA burst size for GPMI channel 4. Do not change. GPMI only support burst size 4.
7–6 CH3	DMA burst size for GPMI channel 3. Do not change. GPMI only support burst size 4.
5–4 CH2	DMA burst size for GPMI channel 2. Do not change. GPMI only support burst size 4.
3–2 CH1	DMA burst size for GPMI channel 1. Do not change. GPMI only support burst size 4.
CH0	DMA burst size for GPMI channel 0. Do not change. GPMI only support burst size 4.

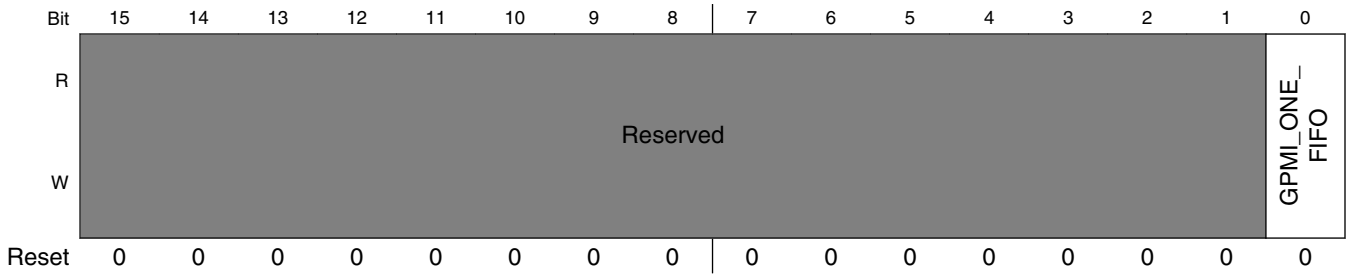
14.5.7 AHB to APBH DMA Debug Register (APBH_DEBUG)

This register is for debug purpose.

The debug register is for internal use only. Not recommend for customer useage.

Address: 11_0000h base + 60h offset = 11_0060h





APBH_DEBUG field descriptions

Field	Description
31–1 -	This field is reserved. Reserved, always set to zero.
0 GPMI_ONE_FIFO	Set to 0ne and the 8 GPMI channels will share the DMA FIFO, and when set to zero, the 8 GPMI channels will use its own DMA FIFO.

14.5.8 APBH DMA Channel n Current Command Address Register (APBH_CH*n*_CURCMDAR)

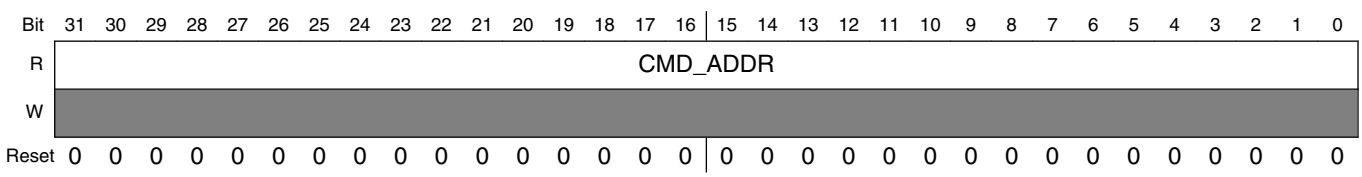
The APBH DMA channel n current command address register points to the multiword command that is currently being executed. Commands are threaded on the command address.

APBH DMA Channel n is controlled by a variable sized command structure. This register points to the command structure currently being executed.

EXAMPLE

```
pCurCmd = (apbh_chn_cmd_t *) APBH_CHn_CURCMDAR_RD(0);              // read the whole
register, since there is only one field
pCurCmd = (apbh_chn_cmd_t *) BF_RDn(APBH_CHn_CURCMDAR, 0, CMD_ADDR);  // or, use multi-
register bitfield read macro
pCurCmd = (apbh_chn_cmd_t *) APBH_CHn_CURCMDAR(0).CMD_ADDR;        // or, assign from
bitfield of indexed register's struct
```

Address: 11_0000h base + 100h offset + (112d × i), where i=0d to 15d



APBH_CH*n*_CURCMDAR field descriptions

Field	Description
CMD_ADDR	Pointer to command structure currently being processed for channel n.

14.5.9 APBH DMA Channel n Next Command Address Register (APBH_CH*n*_NXTCMDAR)

The APBH DMA Channel n Next Command Address register contains the address of the next multiword command to be executed. Commands are threaded on the command address. Set CHAIN to 1 in the DMA command word to process command lists.

APBH DMA Channel n is controlled by a variable sized command structure. Software loads this register with the address of the first command structure to process and increments the Channel n semaphore to start processing. This register points to the next command structure to be executed when the current command is completed.

EXAMPLE

```
APBH_CHn_NXTCMDAR_WR(0, (reg32_t) pCommandTwoStructure);         // write the entire
register, since there is only one field
BF_WRn(APBH_CHn_NXTCMDAR, 0, (reg32_t) pCommandTwoStructure);       // or, use multi-
register bitfield write macro
APBH_CHn_NXTCMDAR(0).CMD_ADDR = (reg32_t) pCommandTwoStructure;  // or, assign to bitfield
of indexed register's struct
```

Address: 11_0000h base + 110h offset + (112d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMD_ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APBH_CH*n*_NXTCMDAR field descriptions

Field	Description
CMD_ADDR	Pointer to next command structure for channel n.

14.5.10 APBH DMA Channel n Command Register (APBH_CH*n*_CMD)

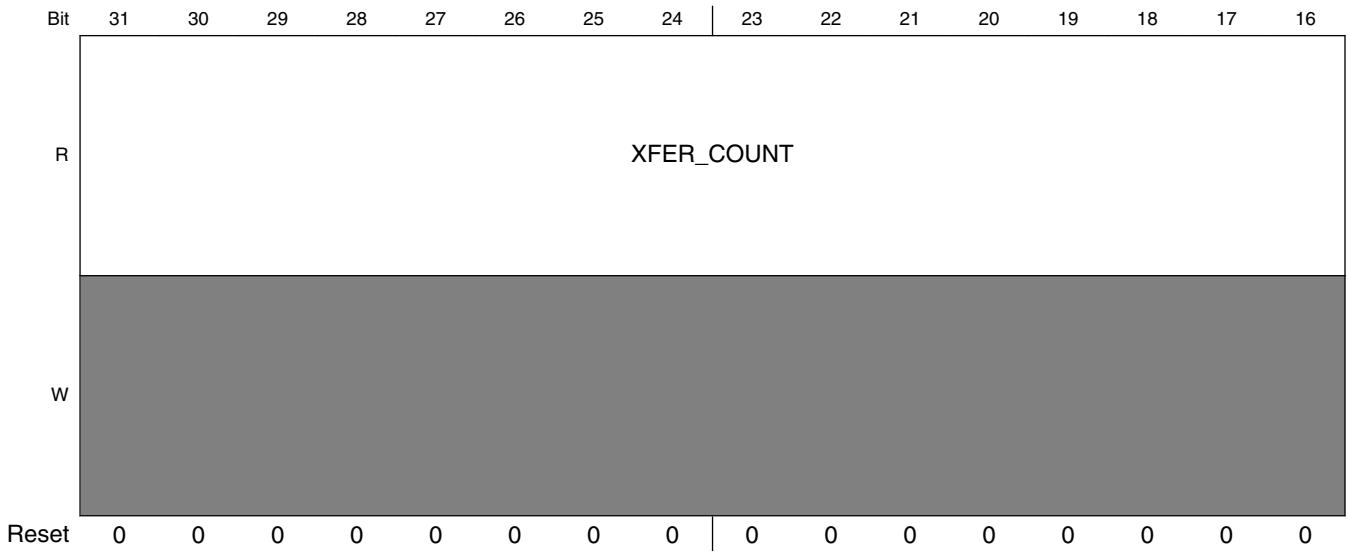
The APBH DMA Channel n command register specifies the DMA transaction to perform for the current command chain item.

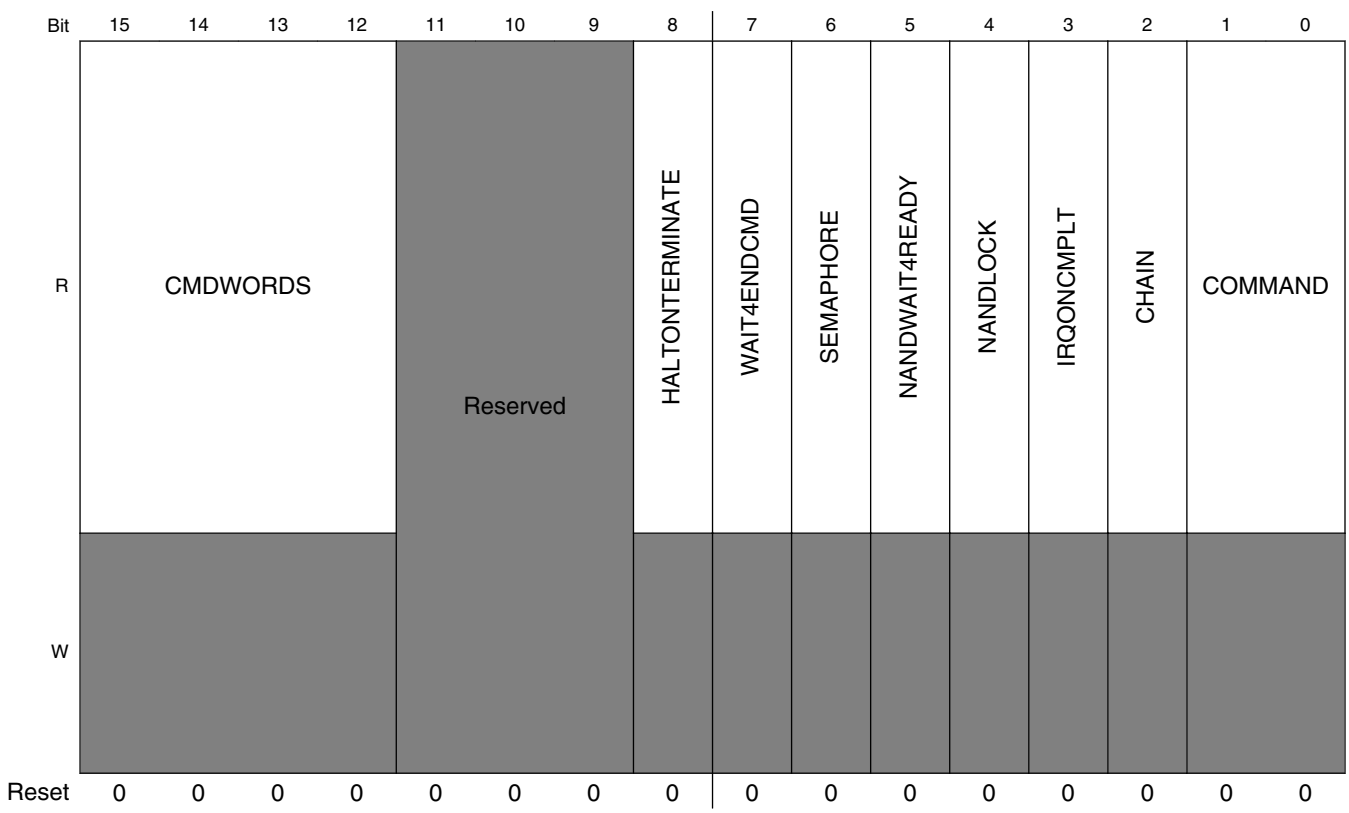
The command register controls the overall operation of each DMA command for this channel. It includes the number of bytes to transfer to or from the device, the number of APB PIO command words included with this command structure, whether to interrupt at command completion, whether to chain an additional command to the end of this one and whether this transfer is a read or write DMA transfer.

EXAMPLE

```
apbh_chn_cmd_t  dma_cmd;
dma_cmd.XFER_COUNT = 512;                                    // transfer 512 bytes
dma_cmd.COMMAND = BV_APBH_CHn_CMD_COMMAND__DMA_WRITE;  // transfer to system memory from
peripheral device
dma_cmd.CHAIN = 1;                                           // chain an additional command
structure on to the list
dma_cmd.IRQONCMPLT = 1;                                      // generate an interrupt on
completion of this command structure
```

Address: 11_0000h base + 120h offset + (112d × i), where i=0d to 15d





APBH_CH*n*_CMD field descriptions

Field	Description
31–16 XFER_COUNT	This field indicates the number of bytes to transfer to or from the appropriate PIO register in the GPMI0 device. A value of 0 indicates a 64 KBytes transfer.
15–12 CMDWORDS	This field indicates the number of command words to send to the GPMI0, starting with the base PIO address of the GPMI0 control register and incrementing from there. Zero means transfer NO command words
11–9 -	This field is reserved. Reserved, always set to zero.
8 HALTONTERMINATE	A value of one indicates that the channel will immeditately terminate the current descriptor and halt the DMA channel if a terminate signal is set. A value of 0 will still cause an immediate terminate of the channel if the terminate signal is set, but the channel will continue as if the count had been exhausted, meaning it will honor IRQONCMPLT, CHAIN, SEMAPHORE, and WAIT4ENDCMD.
7 WAIT4ENDCMD	A value of one indicates that the channel will wait for the end of command signal to be sent from the APBH device to the DMA before starting the next DMA command.
6 SEMAPHORE	A value of one indicates that the channel will decrement its semaphore at the completion of the current command structure. If the semaphore decrements to zero, then this channel stalls until software increments it again.
5 NANDWAIT4READY	A value of one indicates that the NAND DMA channel will will wait until the NAND device reports "ready" before executing the command. It is ignored for non-NAND DMA channels.
4 NANDLOCK	A value of one indicates that the NAND DMA channel will remain "locked" in the arbiter at the expense of other NAND DMA channels. It is ignored for non-NAND DMA channels.
3 IRQONCMPLT	A value of one indicates that the channel will cause the interrupt status bit to be set upon completion of the current command, i.e. after the DMA transfer is complete.
2 CHAIN	A value of one indicates that another command is chained onto the end of the current command structure. At the completion of the current command, this channel will follow the pointer in APBH_CHn_CMDAR to find the next command.
COMMAND	This bitfield indicates the type of current command: 0x0 **NO_DMA_XFER** — Perform any requested PIO word transfers but terminate command before any DMA transfer. 0x1 **DMA_WRITE** — Perform any requested PIO word transfers and then perform a DMA transfer from the peripheral for the specified number of bytes. 0x2 **DMA_READ** — Perform any requested PIO word transfers and then perform a DMA transfer to the peripheral for the specified number of bytes. 0x3 **DMA_SENSE** — Perform any requested PIO word transfers and then perform a conditional branch to the next chained device. Follow the NEXCMD_ADDR pointer if the perpheral sense is true. Follow the BUFFER_ADDRESS as a chain pointer if the peripheral sense line is false.

14.5.11 APBH DMA Channel n Buffer Address Register (APBH_CH*n*_BAR)

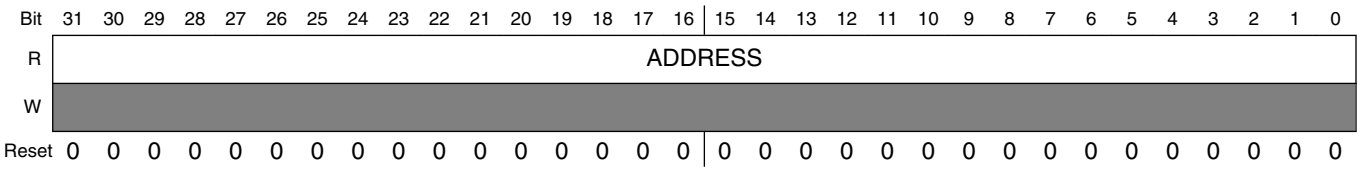
The APBH DMA Channel n buffer address register contains a pointer to the data buffer for the transfer. For immediate forms, the data is taken from this register. This is a byte address which means transfers can start on any byte boundary.

This register holds a pointer to the data buffer in system memory. After the command values have been read into the DMA controller and the device controlled by this channel, then the DMA transfer will begin, to or from the buffer pointed to by this register.

EXAMPLE

```
apbh_chn_bar_t  dma_data;
dma_data.ADDRESS = (reg32_t) pDataBuffer;
```

Address: 11_0000h base + 130h offset + (112d × i), where i=0d to 15d



APBH_CH*n*_BAR field descriptions

Field	Description
ADDRESS	Address of system memory buffer to be read or written over the AHB bus.

14.5.12 APBH DMA Channel n Semaphore Register (APBH_CH*n*_SEMA)

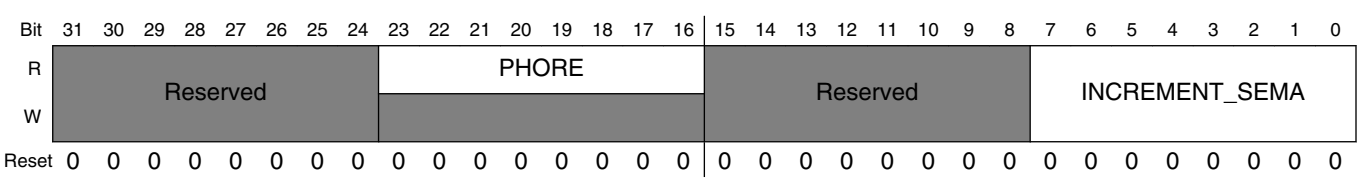
The APBH DMA Channel n semaphore register is used to synchronize the Arm platform instruction stream and the DMA chain processing state.

Each DMA channel has an 8 bit counting semaphore that is used to synchronize between the program stream and and the DMA chain processing. DMA processing continues until the DMA attempts to decrement a semaphore that has already reached a value of zero. When the attempt is made, the DMA channel is stalled until software increments the semaphore count.

EXAMPLE

```
BF_WR(APBH_CHn_SEMA, 0, INCREMENT_SEMA, 2);     // increment semaphore by two
current_sema = BF_RD(APBH_CHn_SEMA, 0, PHORE);  // get instantaneous value
```

Address: 11_0000h base + 140h offset + (112d × i), where i=0d to 15d



APBH_CH*n*_SEMA field descriptions

Field	Description
31–24 -	This field is reserved. Reserved, always set to zero.
23–16 PHORE	This read-only field shows the current (instantaneous) value of the semaphore counter.
15–8 -	This field is reserved. Reserved, always set to zero.
INCREMENT_ SEMA	The value written to this field is added to the semaphore count in an atomic way such that simultaneous software adds and DMA hardware substracts happening on the same clock are protected. This bit field reads back a value of 0x00. Writing a value of 0x02 increments the semaphore count by two, unless the DMA channel decrements the count on the same clock, then the count is incremented by a net one.

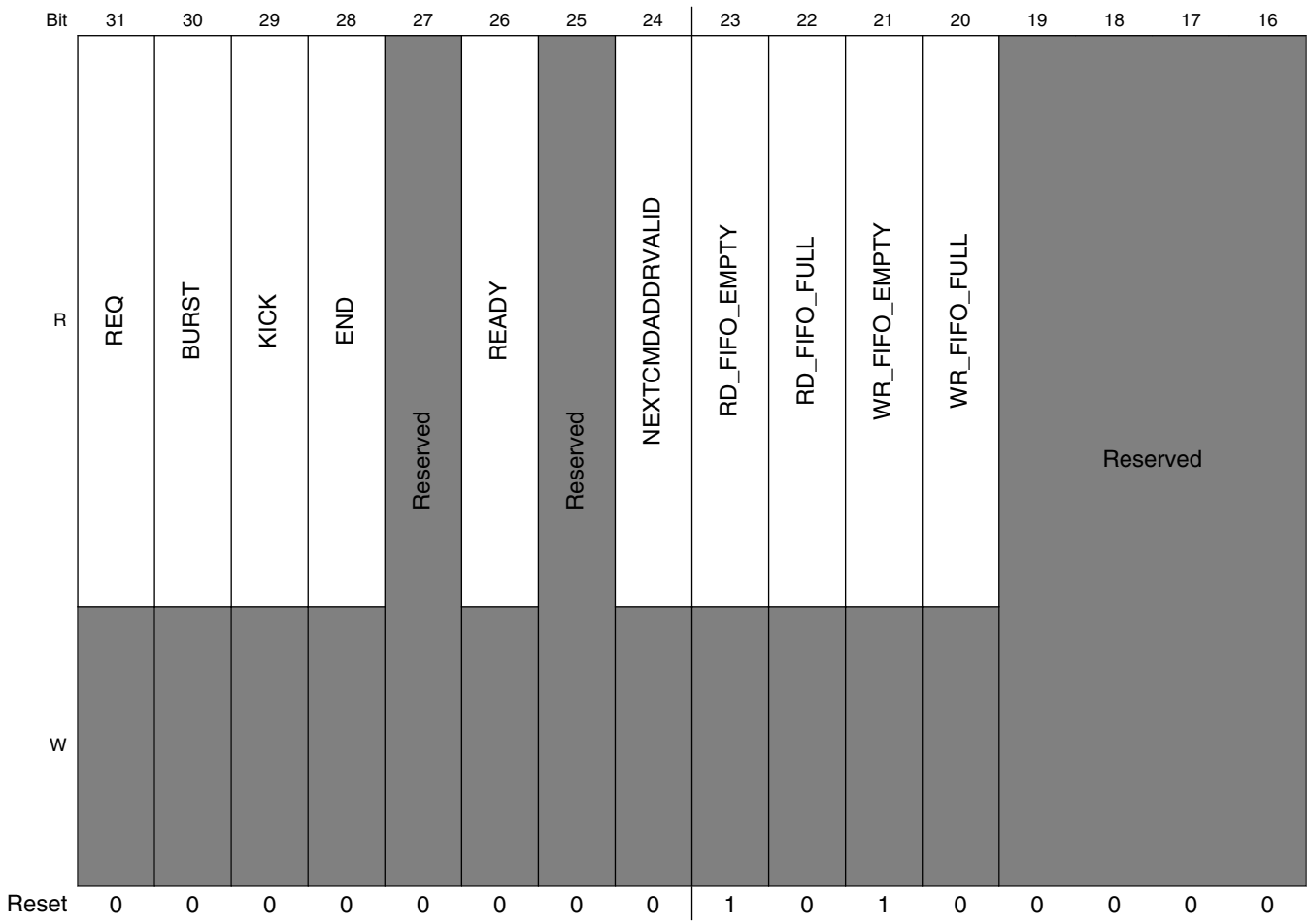
14.5.13 AHB to APBH DMA Channel n Debug Information (APBH_CH*n*_DEBUG1)

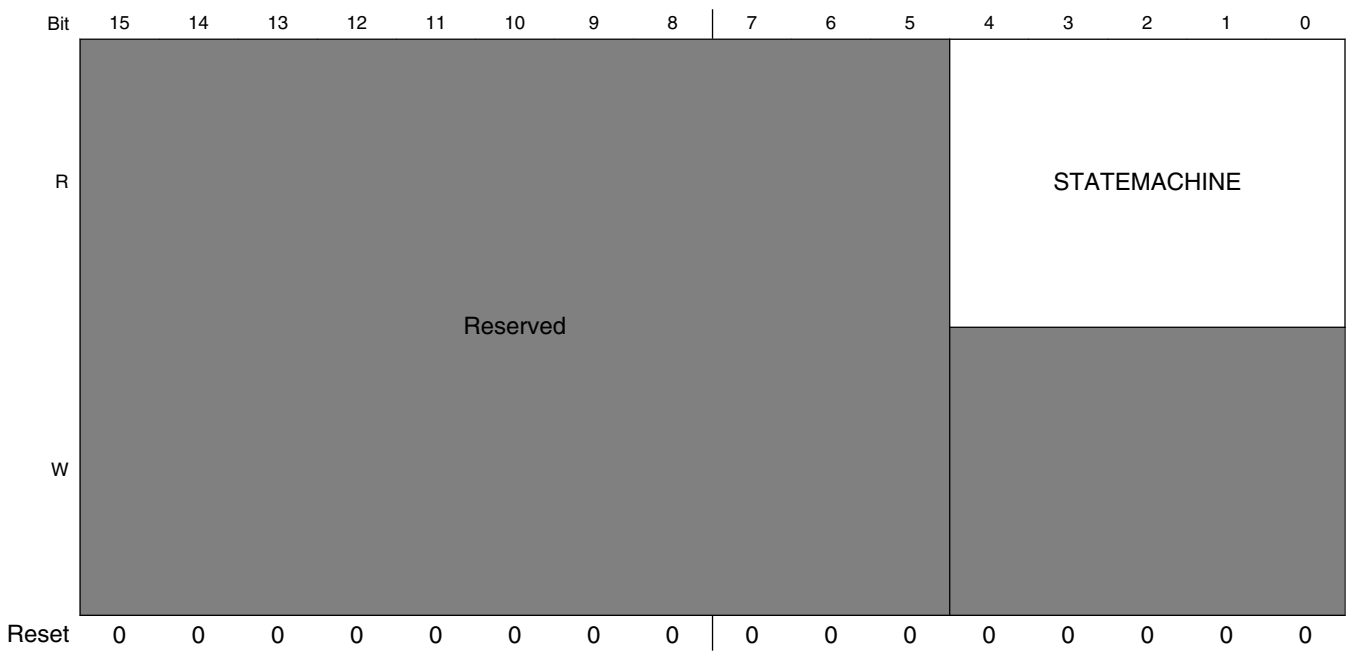
This register gives debug visibility into the APBH DMA Channel n state machine and controls.

This register allows debug visibility of the APBH DMA Channel n.

APBH Memory Map/Register Definition

Address: 11_0000h base + 150h offset + (112d × i), where i=0d to 15d





APBH_CH*n*_DEBUG1 field descriptions

Field	Description
31 REQ	This bit reflects the current state of the DMA Request Signal from the APB device
30 BURST	This bit reflects the current state of the DMA Burst Signal from the APB device
29 KICK	This bit reflects the current state of the DMA Kick Signal sent to the APB Device
28 END	This bit reflects the current state of the DMA End Command Signal sent from the APB Device
27 SENSE	This field is reserved. This bit is reserved for this DMA Channel and always reads 0.
26 READY	This bit is reserved for this DMA Channel and always reads 0.
25 LOCK	This field is reserved. This bit is reserved for this Channel and always reads 0.
24 NEXTCMDADDRVALID	This bit reflects the internal bit which indicates whether the channel's next command address is valid.
23 RD_FIFO_EMPTY	This bit reflects the current state of the DMA Channel's Read FIFO Empty signal.
22 RD_FIFO_FULL	This bit reflects the current state of the DMA Channel's Read FIFO Full signal.
21 WR_FIFO_EMPTY	This bit reflects the current state of the DMA Channel's Write FIFO Empty signal.
20 WR_FIFO_FULL	This bit reflects the current state of the DMA Channel's Write FIFO Full signal.
19–5 RSVD1	This field is reserved. Reserved
STATEMACHINE	PIO Display of the DMA Channel n state machine state. 0x00 **IDLE** — This is the idle state of the DMA state machine. 0x01 **REQ_CMD1** — State in which the DMA is waiting to receive the first word of a command. 0x02 **REQ_CMD3** — State in which the DMA is waiting to receive the third word of a command. 0x03 **REQ_CMD2** — State in which the DMA is waiting to receive the second word of a command. 0x04 **XFER_DECODE** — The state machine processes the descriptor command field in this state and branches accordingly. 0x05 **REQ_WAIT** — The state machine waits in this state for the PIO APB cycles to complete. 0x06 **REQ_CMD4** — State in which the DMA is waiting to receive the fourth word of a command, or waiting to receive the PIO words when PIO count is greater than 1. 0x07 **PIO_REQ** — This state determines whether another PIO cycle needs to occur before starting DMA transfers. 0x08 **READ_FLUSH** — During a read transfers, the state machine enters this state waiting for the last bytes to be pushed out on the APB. 0x09 **READ_WAIT** — When an AHB read request occurs, the state machine waits in this state for the AHB transfer to complete. 0x0C **WRITE** — During DMA Write transfers, the state machine waits in this state until the AHB master arbiter accepts the request from this channel. 0x0D **READ_REQ** — During DMA Read transfers, the state machine waits in this state until the AHB master arbiter accepts the request from this channel.

Table continues on the next page...

APBH_CH*n*_DEBUG1 field descriptions (continued)

Field	Description
	0x0E **CHECK_CHAIN** — Upon completion of the DMA transfers, this state checks the value of the Chain bit and branches accordingly. 0x0F **XFER_COMPLETE** — The state machine goes to this state after the DMA transfers are complete, and determines what step to take next. 0x14 **TERMINATE** — When a terminate signal is set, the state machine enters this state until the current AHB transfer is completed. 0x15 **WAIT_END** — When the Wait for Command End bit is set, the state machine enters this state until the DMA device indicates that the command is complete. 0x1C **WRITE_WAIT** — During DMA Write transfers, the state machine waits in this state until the AHB master completes the write to the AHB memory space. 0x1D **HALT_AFTER_TERM** — If HALTONTERMINATE is set and a terminate signal is set, the state machine enters this state and effectively halts. A channel reset is required to exit this state 0x1E **CHECK_WAIT** — If the Chain bit is a 0, the state machine enters this state and effectively halts. 0x1F **WAIT_READY** — When the NAND Wait for Ready bit is set, the state machine enters this state until the GPMI device indicates that the external device is ready.

14.5.14 AHB to APBH DMA Channel n Debug Information (APBH_CH*n*_DEBUG2)

This register gives debug visibility for the APB and AHB byte counts for DMA Channel n.

This register allows debug visibility of the APBH DMA Channel n.

Address: 11_0000h base + 160h offset + (112d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	APB_BYTES																AHB_BYTES															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APBH_CH*n*_DEBUG2 field descriptions

Field	Description
31–16 APB_BYTES	This value reflects the current number of APB bytes remaining to be transfered in the current transfer.
AHB_BYTES	This value reflects the current number of AHB bytes remaining to be transfered in the current transfer.

14.5.15 APBH Bridge Version Register (APBH_VERSION)

This register always returns a known read value for debug purposes it indicates the version of the block.

This register indicates the RTL version in use.

EXAMPLE

```
if (APBH_VERSION.B.MAJOR != 3)
   Error();
```

Address: 11_0000h base + 800h offset = 11_0800h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR								MINOR								STEP															
W																																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APBH_VERSION field descriptions

Field	Description
31–24 MAJOR	Fixed read-only value reflecting the MAJOR field of the RTL version.
23–16 MINOR	Fixed read-only value reflecting the MINOR field of the RTL version.
STEP	Fixed read-only value reflecting the stepping of the RTL version.

Chapter 15
Asynchronous Sample Rate Converter (ASRC)

15.1 Overview

The Asynchronous Sample Rate Converter (ASRC) converts the sampling rate of a signal associated with an input clock into a signal associated with a different output clock.

The ASRC supports concurrent sample rate conversion of up to 10 channels of about -120dB THD+N. The ASRC supports up to three sampling rate pairs.

The incoming audio data to this chip may be received from various sources at different sampling rates. The outgoing audio data of this chip may have different sampling rates and it can also be associated with output clocks that are asynchronous to the input clocks.

The ASRC is implemented as a co-processor in hardware, with minimal ARM Platform intervention required.

The following figure is a system view of the connection between the ASRC block and other blocks.

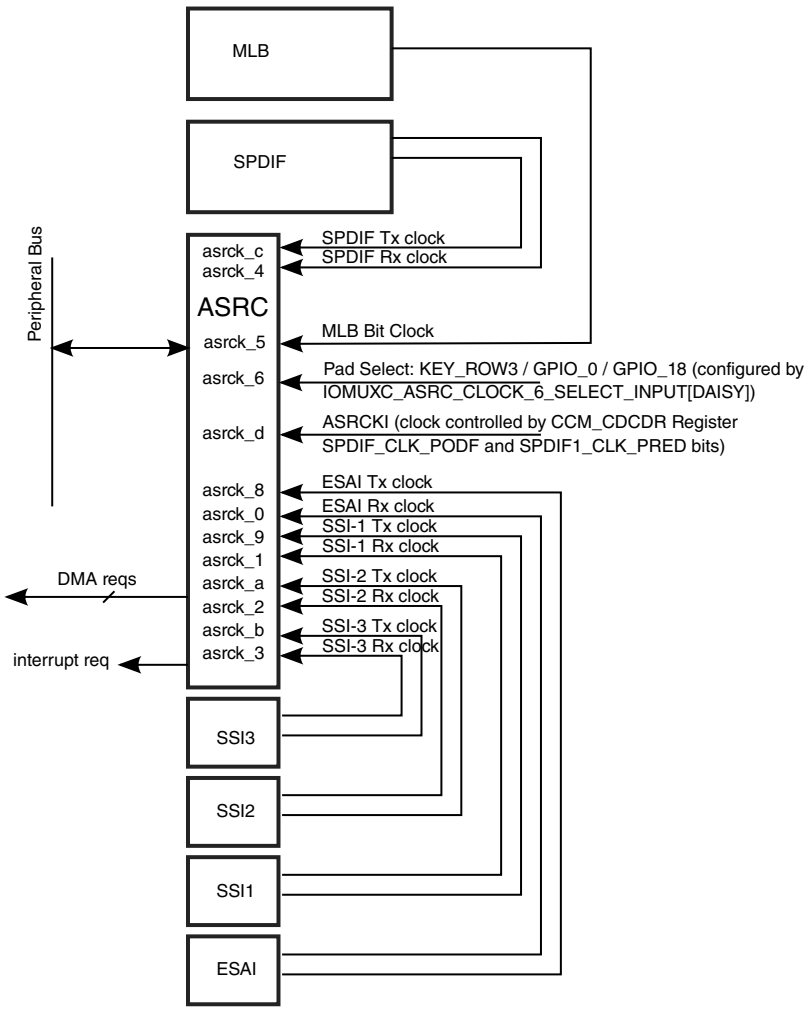


Figure 15-1. General System Overview

The following figure is the ASRC block diagram.

The red dotted line designates the ASRC block. Objects outside the dotted line represent SoC-level resources.

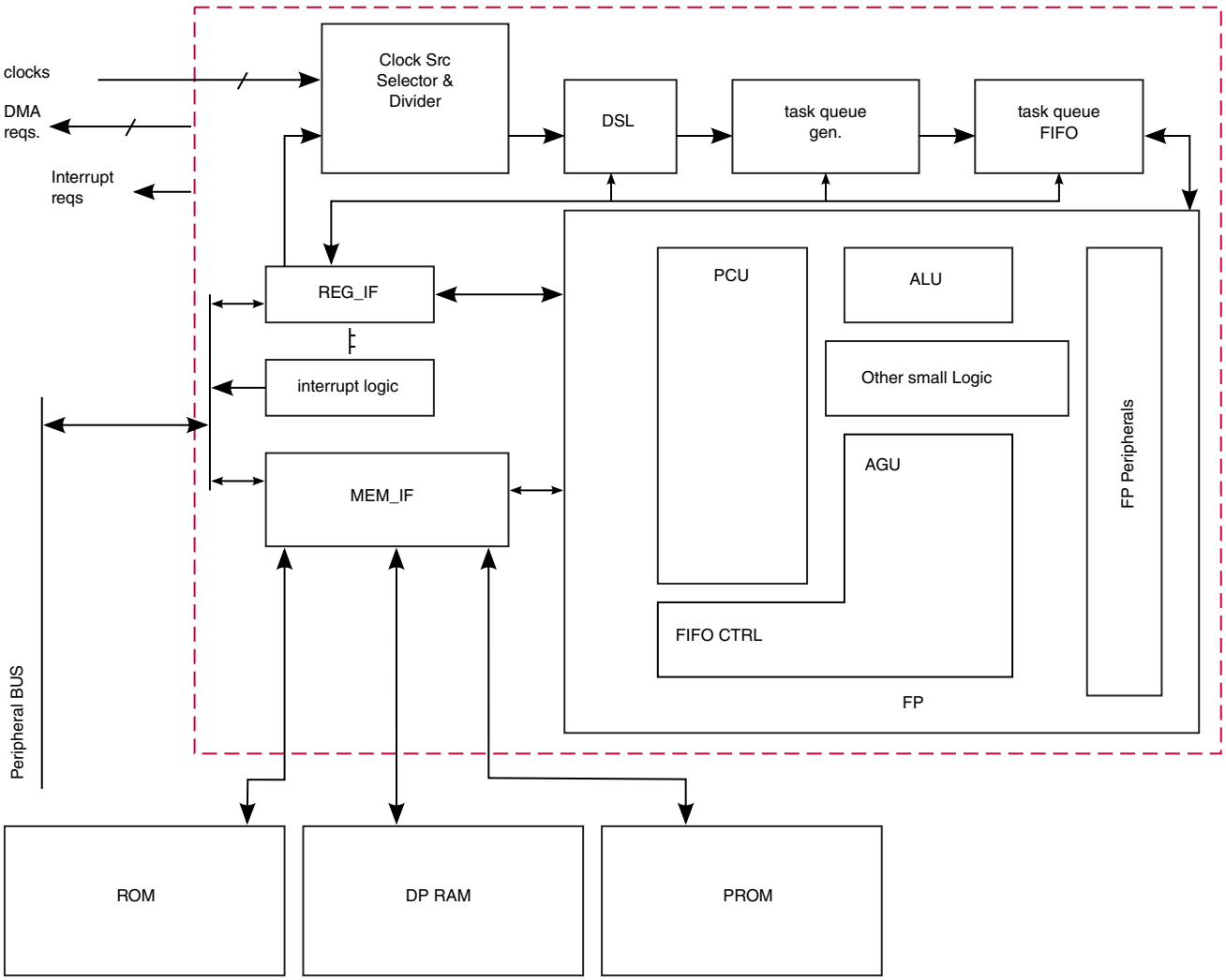


Figure 15-2. ASRC block diagram

15.1.1 Features

Table 15-1. ASRC Specifications

Parameters	Test Conditions	Minimum	Typical	Maximum	Unit
Channels Supported		0 [1]	n (0-10)	10	
Pairs of Rate Conversion		1	-	3	
THD+N	120 MHz < Fs_{ASRC}[2] <160 MHz		-120		dB
Dynamic Range				144	dB
Settling Time			40		ms
Comment:					

1. When a pair has zero channels, the pair will be disabled, although the pair enable bit may be set in ASRCTR register.
2. Fs_{ASRC} is the processing clock of ASRC block.

Other Features:

- Any number (0-10) of contiguous channels can be associated to one of the sampling rate pairs.
- Support user-programmable threshold for the input/output FIFOs.
- Support flexible 8/16/24-bit width of input data, and 16/24-bit width of output data.
- Designed for rate conversion between 44.1 kHz, 32 kHz, 48 kHz, 96 kHz, and 192 kHz. The useful signal bandwidth is below 24 kHz.
- Other input sampling rates in the range of 8 kHz to 200 kHz is also supported, but possibly with less desirable bandwidth.
- Other output sampling rates in the range of 30 kHz to 200 kHz is also supported, but possibly with less desirable bandwidth.
- Automatic accommodation to slow variations in the incoming and outgoing sampling rates.
- Linear phase
- Tolerant to sample clock jitter

Clock/Data Connections

- The sampling rate clocks are directly connected to the ASRC block, the ratio estimation of the input clocks with output clocks are done in ASRC hardware when both input/output sampling clocks are physically available.
- When both the input sampling clock and the output sampling clock are physically available, the rate conversion can work by configuring the physical clocks.
- When the input sampling clock is not physically available, the rate conversion can still work by setting ideal-ratio values into ASRC interface registers.
- The clock signals come from the following blocks:
  - ESAI, receiving bit clock and transmitting bit clock
  - SPDIF, receiving bit clock and transmitting bit clock
  - other audio peripherals etc.
  - MLB bit clock
- The exchange of audio data is done by the processor accessing ASRC block through registers defined on shared peripheral bus.

15.1.2 Modes of Operation

See ASRC Memory Map/Register Definition for a definition of the registers and parameters used in ASRC.

15.1.2.1 Data Transfer Schemes

15.1.2.1.1 Data Input Modes

The input mode for each of the three channel sets may be set independently. Three modes of supplying data to the ASRC input FIFOs are available:

- Polling
- Interrupt
- DMA

In all input-data transfer schemes, the ASRC fetches data from each enabled FIFO and processes the data sample-by-sample after each rising edge of the associated input sampling clock until the FIFO level reaches a threshold.

After the threshold is reached, the ASRC requests data. The FIFO size for each channel set is 64 samples and the threshold is set at 32 samples. The threshold can be defined by interface registers ASRMCRx, x=A, B or C.

If the ASRC attempts to fetch data from an empty FIFO, an error is generated and the ASRSTR_AOLE bit is set. If the ASRC overload interrupt is enabled (ASRIER_AOLIE bit is set), an interrupt is generated.

When writing data to an input FIFO, you must ensure that it is in a predefined sequence. For example, when writing to an input FIFO, the sequence should be: channel_0, channel_1, channel_2,..., channel_n, channel_0, channel_1, channel_2, etc. Here channel_n stands for the data intended for the n-th channel. The hardware will re-allocate each data to its corresponding channel FIFO. The channel being re-allocated is shown by ASRCCR_ACIx, x=A,B or C.

Mode 1 (Polling Mode)

Polling mode is the default mode following power-on or individual reset, and is selected by clearing the associated channel set A, B, or C data-input interrupt enable bit (ASRIER_ADIEx, where x=A, B or C). In this mode, data-input interrupts are disabled. When the FIFO level is below the threshold, the associated status bit (ASRSTR_AIDEx, where x=A, B, or C) is set. To clear the status bit, the FIFO must be written with enough data to raise the level above the threshold.

Mode 2 (Interrupt Mode)

The ASRC input FIFOs can also be serviced by interrupts. To enable interrupts, the corresponding data-input interrupt enable bits (ASRIER_ADIEx, where x=A, B, or C) should be set. An interrupt is automatically generated any time the input FIFO level is below the threshold. The interrupt is cleared when enough data is written to the FIFO to raise the level above the threshold.

Mode 3 (DMA Mode)

The ASRC input FIFOs can also be filled using DMA. In this mode, the data-input interrupt-enable bits (ASRIER_ADIEx, where x=A, B, or C) should be cleared and the DMA controller should be configured to use the ASRC as a request source.

15.1.2.1.2 Data Output Modes

The output mode for each of the 3 channel sets (A, B, and C) may be set independently.

Three modes of retrieving data from the ASRC output FIFOs are available:
- Polling
- Interrupt
- DMA

In all output-data transfer schemes, the ASRC places a processed sample into the associated output FIFO. After a threshold is reached, the ASRC requests that data be transferred out of the FIFO.

The FIFO size for each channel set is 64 samples and the threshold is set at 32 samples. The threshold can be defined by interface registers ASRMCRx, x=A, B or C.

If the ASRC attempts to place data into a FIFO that is already full, an error is generated and the ASRSTR_AOLE bit is set. If the ASRC overload interrupt is enabled (ASRIER_AOLIE bit is set), an interrupt is generated.

Each output FIFO is organized in the same channel order in which the associated input FIFO was written.

Three transfer modes are supported by Interface Block.

Mode 1 (Polling Mode)

The ASRC output FIFOs can be serviced by polling. In this mode, ensure the associated output-data interrupt enable bit (ASRIER_ADOEx, where x=A, B, or C) is cleared. In this mode, all output-data interrupts are disabled. Any time the output FIFO exceeds the threshold the associated status bit (ASRSTR_AODFx, where x=A, B, or C) is set. To clear the status bit, enough data must be read from the associated output FIFO to lower the level below the threshold.

Mode 2 (Interrupt Mode)

The ASRC output FIFOs may also be serviced using interrupts. To enable this mode, the corresponding output-data interrupt-enable bits (ASRIER_ADOEx, where x=A, B, or C) should be set. Any time the output FIFO level exceeds the threshold, an interrupt is automatically generated. The interrupt is cleared when enough data is read from the FIFO to lower the level below the threshold.

Mode 3 (DMA Mode)

The ASRC output FIFOs can also be read using DMA. In this mode, the output-data interrupt-enable bits (ASRIER_ADOEx, where x=A, B, or C) should be cleared and the DMA controller should be configured to use the ASRC as a request source.

15.1.2.2 Word Alignment Supported

15.1.2.2.1 Input Data Alignment Modes

The position and length of input data word to the input data FIFOs ASRDIA, ASRDIB, ASRDIC are programmable. The control bits are defined in ASRMCR1x {x=A, B, or C}. It supports the following modes.

Table 15-2. Input Data Alignment

Format	Bit Number																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8-bit LSB Aligned																									7	6	5	4	3	2	1	0
8-bit MSB Aligned																	7	6	5	4	3	2	1	0								
16-bit LSB Aligned																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit MSB Aligned	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
24-bit LSB Aligned									23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24-bit MSB Aligned	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								

15.1.2.2.2 Output Data Alignment Modes

The position and length of output data word from the output data FIFOs ASRDOA, ASRDOB, ASRDOC are programmable. The control bits are defined in ASRMCR1x {x=A, B, or C}. It supports the following modes.

Table 15-3. Output Data Alignment

Format	Bit Number																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit LSB Aligned																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table continues on the next page...

Table 15-3. Output Data Alignment (continued)

Format	Bit Number																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit LSB Aligned with Sign Extension	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit MSB Aligned	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
24-bit LSB Aligned									23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24-bit LSB Aligned with Sign Extension	23	23	23	23	23	23	23	23	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24-bit MSB Aligned	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								

15.2 Clocks

The table found here describes the clock sources for ASRC.

Please see clock control block for clock setting, configuration and gating information.

Table 15-4. ASRC Clocks

Clock name	Clock Root	Description
asrck_clock_d	spdif1_clk_root	ASRC module clock (SPDIF clock)
ipg_clk	ahb_clk_root	Peripheral clock
mem_clk	ahb_clk_root	Peripheral access clock

15.3 Interrupts

ASRC has several interrupts events.

The priorities are shown in the following table.

Table 15-5. Interrupt Priorities/Vector

Priority	Offset	Description
lowest	0x0	ASRC Pair A input data needed
	0x2	ASRC Pair B input data needed

Table continues on the next page...

Table 15-5. Interrupt Priorities/Vector (continued)

Priority	Offset	Description
	0x4	ASRC Pair C input data needed
	0x6	ASRC Pair A output data ready
	0x8	ASRC Pair B output data ready
	0xA	ASRC Pair C output data ready
highest	0xC	ASRC Overload

15.4 DMA requests

ASRC has six DMA requests. They are directly connected to the lowest six status bits in the ASRSTR register. For some ARM platforms, the six status bits are directly connected to SDMA or HDMA as DMA event signals.

Table 15-6. DMA requests

Type	Description
0	ASRC Pair A input data needed
1	ASRC Pair B input data needed
2	ASRC Pair C input data needed
3	ASRC Pair A output data ready
4	ASRC Pair B output data ready
5	ASRC Pair C output data ready

15.5 Functional Description

This section provides a complete functional description of the block.

15.5.1 Algorithm Description

15.5.1.1 Signal processing flow

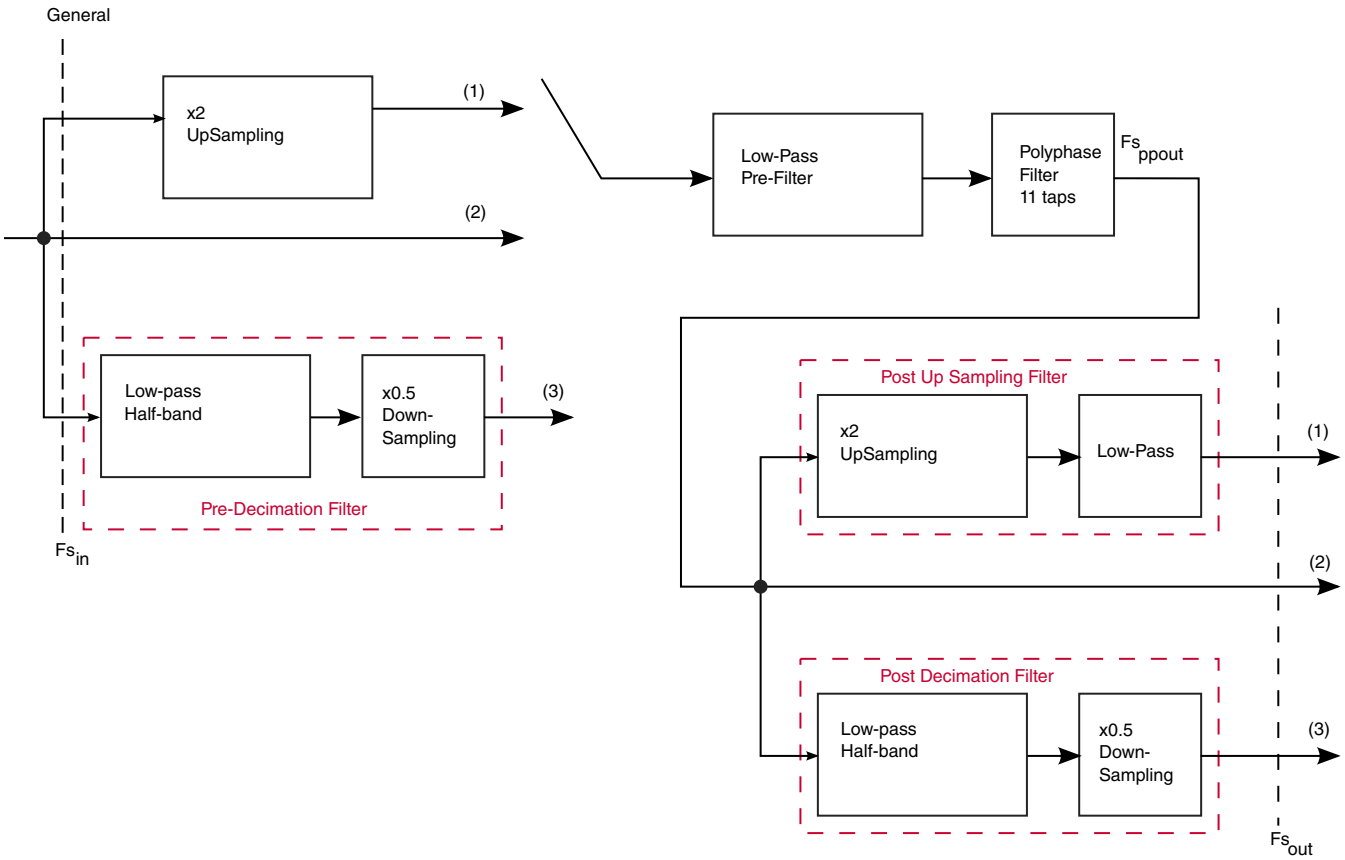


Figure 15-3. Signal processing configurations

The figure above shows the possible configurations of the ASRC. Each configuration consists of 2 to 4 stages.

- x2 up-sampling rate expander (zero insertion only) (input branch 1), direct connection (input branch 2), or low-pass pre decimation filter (consisting of a low-pass half-band FIR filter with x0.5 downsampling rate decimator) (input branch 3),
- low-pass pre-filter, the low-pass bandwidth is at most 0.25 x Fs, where Fs is the sampling rate of the input signal to this low-pass pre-filter,
- polyphase filter,
- x2 post upsampling filter (consisting of a x2 up-sampling rate expander (zero insertion only) with low-pass half-band FIR filter) (output branch 1), direct connection (output branch 2), or low-pass post decimation filter (consisting of a low-pass half-band FIR filter with x0.5 downsampling rate decimator) (output branch 3).

By flowing through different processing branches and different setups of the pre-filter, this ASRC scheme can be used to handle different rate conversion requirements.

- Configuration (a): Input Branch 1+Output Branch 1:

The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/2$.
The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}/2$.

- Configuration (b): Input Branch 1+Output Branch 2:

  The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/2$.
  The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}$.

- Configuration (c): Input Branch 1+Output Branch 3:

  The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/2$.
  The signal sampling rate of the polyphase filter output is $Fs_{ppout} = 2Fs_{out}$.

- Configuration (d): Input Branch 2+Output Branch 1:

  The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/4$.
  The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}/2$.

- Configuration (e): Input Branch 2+Output Branch 2:

  The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/4$.
  The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}$.

- Configuration (f): Input Branch 2+Output Branch 3:

  The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/4$.
  The signal sampling rate of the polyphase filter output is $Fs_{ppout} = 2Fs_{out}$.

- Configuration (g): Input Branch 3+Output Branch 1:

  The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/8$.
  The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}/2$.

- Configuration (h): Input Branch 3+Output Branch 2:

  The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/8$.
  The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}$.

- Configuration (i): Input Branch 3+Output Branch 3:

  The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/8$.
  The signal sampling rate of the polyphase filter output is $Fs_{ppout} = 2Fs_{out}$.

Table 15-7. Pre-processing, post-processing options

{Pre_Proc, Post_Proc}		Fsout (KHz)											
		8	12	16	24	32	44.1	48	64	88.2	96	128	192
Fsin (KHz)	8	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}
	12	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}
	16	{1,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}
	24	{1,2}	{1,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}
	32	{1,2}	{1,2}	{1,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}
	44.1	{2,2}	{1,2}	{1,2}	{0,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}
	48	{2,2}	{1,2}	{1,2}	{1,2}	{0,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}
	64	{2,2}	{2,2}	{1,2}	{1,2}	{0,2}	{0,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}
	88.2	NA	{2,2}	{2,2}	{1,2}	{1,2}	{1,2}	{1,2}	{1,1}	{1,1}	{1,1}	{1,1}	{1,1}
	96	NA	{2,2}	{2,2}	{1,2}	{1,2}	{1,2}	{1,2}	{1,1}	{1,1}	{1,1}	{1,1}	{1,1}
	128	NA	NA	{2,2}	{2,2}	{1,2}	{1,2}	{1,2}	{1,1}	{1,1}	{1,1}	{1,1}	{1,1}
	192	NA	NA	NA	{2,2}	{2,2}	{2,2}	{2,2}	{2,1}	{2,1}	{2,1}	{2,1}	{2,1}

NOTE: In the {Pre_Proc, Post_Proc} pair, the meaning of the values are:

Pre_Proc:

- 0 --- Pre-processing Branch 1 as shown in Figure 15-3
- 1 --- Pre-processing Branch 2 as shown in Figure 15-3
- 2 --- Pre-processing Branch 3 as shown in Figure 15-3, decimation-by-2

Post_Proc:

- 0 --- Post-processing Branch 1 as shown in Figure 15-3
- 1 --- Post-processing Branch 2 as shown in Figure 15-3
- 2 --- Post-processing Branch 3 as shown in Figure 15-3

The latencies of the different option can be roughly calculated as follows:

- For PreProc = 0, PostProc = 1 : min latency = constant_A / input-sample-rate + constant_B / output-sample-rate
- For PreProc = 0, PostProc = 0 : min latency = constant_A / input-sample-rate + constant_C / output-sample-rate
- For PreProc = 1, PostProc = 1 : min latency = constant_D / input-sample-rate + constant_B / output-sample-rate

The constants above (e.g., constant_A means the Constant for Preproc = 0, constant_B means the Constant for Postproc = 1, ...) are only influenced by the PreProc/PostProc and (input/output) sampling rate to which they are connected. Input latencies have no relationship with the output latencies, but both elements add together to form the total latencies.

For a rough estimation, the constants can be set as:

- Constant for Preproc = 0: 39

- Constant for Preproc = 1: 78.5
- Constant for Preproc = 2: 235
- Constant for Postproc = 0: 42.5
- Constant for Postproc = 1: 8.5
- Constant for Postproc = 2: 172

The max latency can be derived from this value by using the following formula (where 32 means the input/output FIFO depth that will arouse data transfer):

- max latency = min latency + 32 / input-sample-rate + 32 / output-sample-rate

15.5.1.2 Operation of the Filter

15.5.1.2.1 Support of Physical Clocks

This design supports physical sampling clocks. The clocks can be provided by:

- Sony/Phillips digital interface (SPDIF)
- Enhanced Serial Audio Interface (ESAI),
- Media Local Bus controller (MLB)
- Core master clock derivative as ASRCK1

Figure 15-4. Clock Source Selector and Divider

Software can set the ASRC Clock Source Register (ASRCSR) and the Clock Divider Register to select the desired clock source and divide it to the needed sample rate clock for use by the ASRC. The clocks have the following restriction. If the prescaler is set to 1, the clock divider can only be set to 1 and the clock source must have a 50% duty cycle.

15.6 Startup Procedure

The following example shows the normal setup procedure for the ASRC block.

```
#include "asrc_common.h"

#include "stdio.h"

#include "soc_api.h"

int incnt=0;

int outcnt=0;

#include "wy_ideal_ratio_dataini_part.h"

WORD   IdealRatio_High=0x04; //

WORD   IdealRatio_Low=0x0; //
```

```
void asrc_config_alloc(WORD ASRCTR_VAL, WORD ASRIER_VAL, WORD ASRCNCR_VAL,
                           WORD ASRCFG_VAL, WORD ASRCDR1_VAL, WORD ASRCDR2_VAL,
                           WORD ASRCSR_VAL)

{  // Disable ASRC

    reg32_write(ASRC_ASRCTR,0x0);

    reg32_write(ASRC_ASRCTR,ASRCTR_VAL);

    reg32_write(ASRC_ASRIER,ASRIER_VAL);

    reg32_write(ASRC_ASRIEM,0x0);

    reg32_write(ASRC_ASRCNCR,ASRCNCR_VAL);

    reg32_write(ASRC_ASRCFG,ASRCFG_VAL);

    reg32_write(ASRC_ASRCDR1,ASRCDR1_VAL);

    reg32_write(ASRC_ASRCDR2,ASRCDR2_VAL);

    reg32_write(ASRC_ASRCSR, ASRCSR_VAL);

    reg32_write(ASRC_ASRPM1, 0x7fffff);

    reg32_write(ASRC_ASRPM2, 0x255555);

    reg32_write(ASRC_ASRPM3, 0xff7280);

    reg32_write(ASRC_ASRPM4, 0xff7280);

    reg32_write(ASRC_ASRPM5, 0xff7280);

    reg32_write(ASRC_ASRQFIFO1,0x001f00);

    reg32_write(ASRC_ASRMCRA,0x001f00);

    reg32_write(ASRC_ASRMCRB,0x001f00);

    reg32_write(ASRC_ASRMCRC,0x001f00);

}

 void sim_ideal_ratio()

  {

    WORD tmp32bit;

    #define ASRSTR_AIDEA_MASK  0x1

    #define ASRSTR_AODFA_MASK  0x1 <<3

    #define ASRSTR_AOLE_MASK   0x1<<6

    #define ASRCTR_DBG_EN             1<<23

    #define ASRCTR_IDRA               1<<13

    #define ASRCTR_USRA               1<<14

    #define ASRC_CLK_PRED_RSTRICTED 0<<28

    #define ASRC_CLK_PRED_DFLT      1<<28 // default: 596MHz div by 2
```

```
#define ASRC_CLK_PRED_DIV3      2<<28 // 596MHz div by 3

#define ASRC_CLK_PRED_DIV4      3<<28 // 596MHz div by 4

#define ASRC_CLK_PRED_DIV5      4<<28 // 596MHz div by 5

#define ASRC_CLK_PRED_DIV6      5<<28 // 596MHz div by 6

#define ASRC_CLK_PRED_DIV7      6<<28 // 596MHz div by 7

#define ASRC_CLK_PRED_DIV8      7<<28 // 596MHz div by 8

#define ECSPI_CLK_PRED_DFLT     1<<25

#define ECSPI_CLK_PODF_DFLT     1<<19

#define ASRC_CLK_PODF_DIV1      0<<9  // pred output divide by 1 again

#define ASRC_CLK_PODF_DIV2      1<<9  // pred output divide by 2 again

#define ASRC_CLK_PODF_DIV3      2<<9  // pred output divide by 3 again

#define ASRC_CLK_PODF_DIV4      3<<9  // pred output divide by 4 again

#define ASRC_CLK_PODF_DFLT      4<<9  // default: pred output divide by 5 again

#define ASRC_CLK_PODF_DIV6      5<<9  // pred output divide by 6 again

#define ASRC_CLK_PODF_DIV7      6<<9  // pred output divide by 7 again

#define ASRC_CLK_PODF_DIV25     24<<9  // pred output divide by 7 again

#define IEEE_CLK_PRED_DFLT      1<<6  //

#define IEEE_CLK_PODF_DFLT      4     //

#define ASR_HFA_HFB         0

#define ASR_PREMODA_UP2         0<<6

#define ASR_PREMODA_DIR         1<<6

#define ASR_PREMODA_DN2         2<<6

#define ASR_PREMODA_PAS         3<<6

#define ASR_POSTMODA_UP2         0<<8

#define ASR_POSTMODA_DIR         1<<8

#define ASR_POSTMODA_DN2         2<<8

 reg32_write(CCM_CSCDR2, ASRC_CLK_PRED_DIV8|ECSPI_CLK_PRED_DFLT|ECSPI_CLK_PODF_DFLT|
ASRC_CLK_PODF_DIV25|IEEE_CLK_PRED_DFLT|IEEE_CLK_PODF_DFLT);

 // Disable the ASRC

 reg32_write(ASRC_ASRCTR, 0x0);

 // program AHB clocks

 tmp32bit = reg32_read(CCM_CBCDR);

 tmp32bit = tmp32bit & (~0x00001C00);
```

```
      //tmp32bit = tmp32bit | (0x00000C00); // AHB 100MHz // divided-by-4

      //tmp32bit = tmp32bit | (0x00001000); // AHB 80MHz // divided-by-5

      //tmp32bit = tmp32bit | (0x00001400); // AHB 66MHz // divided-by-6

      //tmp32bit = tmp32bit | (0x00001800); // AHB 57MHz // divided-by-7

      tmp32bit = tmp32bit | (0x00001C00); // AHB 50MHz // divided-by-8

      reg32_write(CCM_CBCDR, tmp32bit); // enable all perihperal clocks during all modes,
except stop mode

      while ( (reg32_read(CCM_CDHIPR) & 0x00008) != 0);

             asrc_config_alloc ( 0x002 | ASRCTR_IDRA | ASRCTR_USRA,    // ASRCTR_VAL, Use
Ratio input, use ideal ratio, Enable Pair A,

             0x0, //0x09,                                              // ASRIER_VAL, Open
PairA input and output interrupt

             0x002,     // ASRCNCR_VAL, assign 2 channels to Pair A

             ASR_PREMODA_DIR | ASR_POSTMODA_DIR | ASR_HFA_HFB,         // ASRCFG_VAL,
POSTMODA=downsampling by 2 ; PREMODA=downsampling by 2

             0x03b03b , // ASRCDR1_VAL, AOCPA=3(FoutA/(2^3)); AICPA=3(FinA/(2^3));
AOCDA=7(div 8); AICDA=7(div 8);

             0x0  ,     // ASRCDR2_VAL,

             0x00d00d   // ASRCSR_VAL,  AOCSA=d: bit clock d: ASRCK1 clk from CCM; AICSA=d:
bit clock d: ASRCK1 clock from CCM;

             );

      reg32_write(ASRC_ASRIDRHA, 0x04); //

      reg32_write(ASRC_ASRIDRLA, 0x0); // Ideal Ratio is set to be 1.

      #define OUTFIFO_THRESH_0  8<<12

      #define INFIFO_THRESH_1   32

      reg32_write(ASRC_ASRMCRA, OUTFIFO_THRESH_0 | INFIFO_THRESH_1);

      reg32clrbit(ASRC_ASRMCRA, 23);   // zeroize Pair A buffers

      reg32setbit(ASRC_ASRMCRA, 21);   // stall conversion in case of near full/near empty
condition

      reg32clrbit(ASRC_ASRMCRA, 20); // Do not bypass polyA filter

      // Set ASRC Interrupt

      //CAPTURE_INTERRUPT(ASRC_INT_ROUTINE, asrc_handler);

      //enable_hdler(ASRC_INT_NUM);

      disable_hdler(ASRC_INT_NUM);

      incnt=0;

      outcnt=0;

      reg32setbit(ASRC_ASRCTR,0); // enable ASRC
```

```
      #define ASRCFG_INIA_FINISH  0x1<<21
      while ( (reg32_read(ASRC_ASRCFG) & ASRCFG_INIA_FINISH) == 0); // wait for ini finished.
      // Polling
          while (outcnt < 100) <
      {
              int ii;
              if ( (reg32_read(ASRC_ASRSTR) & ASRSTR_AIDEA_MASK) != 0 )
               {
                 for (ii=0;ii<2;ii++)</codeblock
                   {
                      reg32_write(ASRC_ASRDIA,asrc_input_array[incnt]);     // feed in input
data
                      reg32_write(ASRC_ASRDIA,asrc_input_array[incnt]);     // feed in input
data
                      incnt=(incnt+1)%128;
                   }
                }
              if ( (reg32_read(ASRC_ASRSTR) & ASRSTR_AODFA_MASK) != 0 )
               {
                   for (ii=0;ii<2;ii++)<
                     {
                       WORD TempRdOut;
                       TempRdOut=reg32_read(ASRC_ASRDOA);  // get output data
                       TempRdOut=reg32_read(ASRC_ASRDOA);  // get output data
                       outcnt=outcnt+1;
                     }
                 }
              if ( (reg32_read(ASRC_ASRSTR) & ASRSTR_AOLE_MASK) != 0 )
               {
                   reg32_write(ASRC_ASRSTR,ASRSTR_AOLE_MASK);      // clear overloading
errors
                }
             }
         reg32clrbit(ASRC_ASRCTR,0); // disable ASRC
   }
```

15.7 ASRC Memory Map/Register Definition

All useful registers are listed in the memory map below. The access of undefined registers will behave as normal registers.

All the interface registers are LSB aligned except the input FIFOs and the output FIFOs, and each register has only 24 effective bits.

The input FIFO and output FIFO word alignment can be defined using ASRMCR1{A,B,C} registers in 32-bit interface system.

ASRC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
203_4000	ASRC Control Register (ASRC_ASRCTR)	32	R/W	0000_0000h	15.7.1/630
203_4004	ASRC Interrupt Enable Register (ASRC_ASRIER)	32	R/W	0000_0000h	15.7.2/633
203_400C	ASRC Channel Number Configuration Register (ASRC_ASRCNCR)	32	R/W	0000_0000h	15.7.3/634
203_4010	ASRC Filter Configuration Status Register (ASRC_ASRCFG)	32	R/W	0000_0000h	15.7.4/636
203_4014	ASRC Clock Source Register (ASRC_ASRCSR)	32	R/W	0000_0000h	15.7.5/638
203_4018	ASRC Clock Divider Register 1 (ASRC_ASRCDR1)	32	R/W	0000_0000h	15.7.6/641
203_401C	ASRC Clock Divider Register 2 (ASRC_ASRCDR2)	32	R/W	0000_0000h	15.7.7/642
203_4020	ASRC Status Register (ASRC_ASRSTR)	32	R	0000_0000h	15.7.8/643
203_4040	ASRC Parameter Register n (ASRC_ASRPMn1)	32	R/W	0000_0000h	15.7.9/646
203_4044	ASRC Parameter Register n (ASRC_ASRPMn2)	32	R/W	0000_0000h	15.7.9/646
203_4048	ASRC Parameter Register n (ASRC_ASRPMn3)	32	R/W	0000_0000h	15.7.9/646
203_404C	ASRC Parameter Register n (ASRC_ASRPMn4)	32	R/W	0000_0000h	15.7.9/646
203_4050	ASRC Parameter Register n (ASRC_ASRPMn5)	32	R/W	0000_0000h	15.7.9/646
203_4054	ASRC ASRC Task Queue FIFO Register 1 (ASRC_ASRTFR1)	32	R/W	0000_0000h	15.7.10/ 647
203_405C	ASRC Channel Counter Register (ASRC_ASRCCR)	32	R/W	0000_0000h	15.7.11/ 648
203_4060	ASRC Data Input Register for Pair x (ASRC_ASRDIA)	32	W	0000_0000h	15.7.12/ 649
203_4064	ASRC Data Output Register for Pair x (ASRC_ASRDOA)	32	R	0000_0000h	15.7.13/ 649
203_4068	ASRC Data Input Register for Pair x (ASRC_ASRDIB)	32	W	0000_0000h	15.7.12/ 649
203_406C	ASRC Data Output Register for Pair x (ASRC_ASRDOB)	32	R	0000_0000h	15.7.13/ 649

Table continues on the next page...

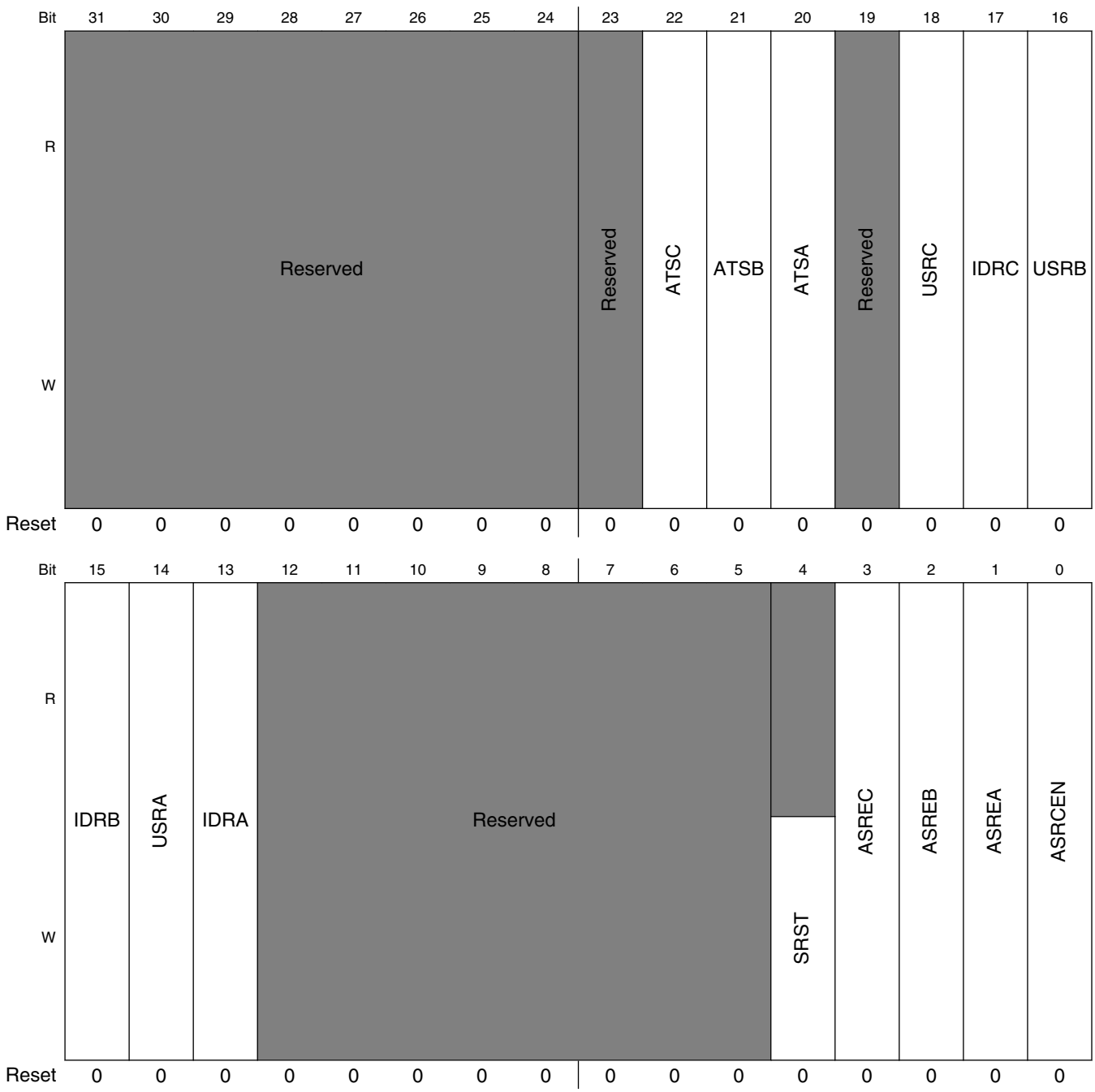
ASRC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
203_4070	ASRC Data Input Register for Pair x (ASRC_ASRDIC)	32	W	0000_0000h	15.7.12/ 649
203_4074	ASRC Data Output Register for Pair x (ASRC_ASRDOC)	32	R	0000_0000h	15.7.13/ 649
203_4080	ASRC Ideal Ratio for Pair A-High Part (ASRC_ASRIDRHA)	32	R/W	0000_0000h	15.7.14/ 650
203_4084	ASRC Ideal Ratio for Pair A -Low Part (ASRC_ASRIDRLA)	32	R/W	0000_0000h	15.7.15/ 650
203_4088	ASRC Ideal Ratio for Pair B-High Part (ASRC_ASRIDRHB)	32	R/W	0000_0000h	15.7.16/ 651
203_408C	ASRC Ideal Ratio for Pair B-Low Part (ASRC_ASRIDRLB)	32	R/W	0000_0000h	15.7.17/ 651
203_4090	ASRC Ideal Ratio for Pair C-High Part (ASRC_ASRIDRHC)	32	R/W	0000_0000h	15.7.18/ 652
203_4094	ASRC Ideal Ratio for Pair C-Low Part (ASRC_ASRIDRLC)	32	R/W	0000_0000h	15.7.19/ 652
203_4098	ASRC 76 kHz Period in terms of ASRC processing clock (ASRC_ASR76K)	32	R/W	0000_0A47h	15.7.20/ 653
203_409C	ASRC 56 kHz Period in terms of ASRC processing clock (ASRC_ASR56K)	32	R/W	0000_0DF3h	15.7.21/ 654
203_40A0	ASRC Misc Control Register for Pair A (ASRC_ASRMCRA)	32	R/W	0000_0000h	15.7.22/ 655
203_40A4	ASRC FIFO Status Register for Pair A (ASRC_ASRFSTA)	32	R	0000_0000h	15.7.23/ 657
203_40A8	ASRC Misc Control Register for Pair B (ASRC_ASRMCRB)	32	R/W	0000_0000h	15.7.24/ 658
203_40AC	ASRC FIFO Status Register for Pair B (ASRC_ASRFSTB)	32	R	0000_0000h	15.7.25/ 660
203_40B0	ASRC Misc Control Register for Pair C (ASRC_ASRMCRC)	32	R/W	0000_0000h	15.7.26/ 661
203_40B4	ASRC FIFO Status Register for Pair C (ASRC_ASRFSTC)	32	R	0000_0000h	15.7.27/ 663
203_40C0	ASRC Misc Control Register 1 for Pair X (ASRC_ASRMCR1A)	32	R/W	0000_0000h	15.7.28/ 664
203_40C4	ASRC Misc Control Register 1 for Pair X (ASRC_ASRMCR1B)	32	R/W	0000_0000h	15.7.28/ 664
203_40C8	ASRC Misc Control Register 1 for Pair X (ASRC_ASRMCR1C)	32	R/W	0000_0000h	15.7.28/ 664

15.7.1 ASRC Control Register (ASRC_ASRCTR)

The ASRC control register (ASRCTR) is a read/write register that controls the ASRC operations.

Address: 203_4000h base + 0h offset = 203_4000h



ASRC_ASRCTR field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 -	This field is reserved. Reserved. Should be written as zero for compatibility.
22 ATSC	ASRC Pair C Automatic Selection For Processing Options When this bit is 1, pair C will automatic update its pre-processing and post-processing options (ASRCFG: PREMODC, ASRCFG:POSTMODC see ASRC Misc Control Register 1 for Pair C) based on the frequencies it detected. To use this option, the two parameter registers(ASR76K and ASR56K) should be set correctly (see ASRC Misc Control Register 1 for Pair C and ASRC Misc Control Register 1 for Pair C). When this bit is 0, the user is responsible for choosing the proper processing options for pair C. This bit should be disabled when {USRC, IDRC}={1,1}.
21 ATSB	ASRC Pair B Automatic Selection For Processing Options When this bit is 1, pair B will automatic update its pre-processing and post-processing options (ASRCFG: PREMODB, ASRCFG:POSTMODB see ASRC Misc Control Register 1 for Pair C) based on the frequencies it detected. To use this option, the two parameter registers(ASR76K and ASR56K) should be set correctly (see ASRC Misc Control Register 1 for Pair C and ASRC Misc Control Register 1 for Pair C). When this bit is 0, the user is responsible for choosing the proper processing options for pair B. This bit should be disabled when {USRB, IDRB}={1,1}.
20 ATSA	ASRC Pair A Automatic Selection For Processing Options When this bit is 1, pair A will automatic update its pre-processing and post-processing options (ASRCFG: PREMODA, ASRCFG:POSTMODA see ASRC Misc Control Register 1 for Pair C) based on the frequencies it detected. To use this option, the two parameter registers(ASR76K and ASR56K) should be set correctly (see ASRC Misc Control Register 1 for Pair C and ASRC Misc Control Register 1 for Pair C). When this bit is 0, the user is responsible for choosing the proper processing options for pair A. This bit should be disabled when {USRA, IDRA}={1,1}.
19 -	This field is reserved. Reserved. Should be written as zero for compatibility.
18 USRC	Use Ratio for Pair C Use ratio as the input to ASRC. This bit is used in conjunction with IDRC control bit.
17 IDRC	Use Ideal Ratio for Pair C When USRC=0, this bit has no usage. When USRC=1 and IDRC=0, ASRC internal measured ratio will be used. When USRC=1 and IDRC=1, the idea ratio from the interface register ASRIDRHC, ASRIDRLC will be used. It is suggested to manually set ASRCFG:POSTMODC, ASRCFG:PREMODC according to Table 15-7 in this case.
16 USRB	Use Ratio for Pair B Use ratio as the input to ASRC. This bit is used in conjunction with IDRB control bit.
15 IDRB	Use Ideal Ratio for Pair B When USRB=0, this bit has no usage. When USRB=1 and IDRB=0, ASRC internal measured ratio will be used. When USRB=1 and IDRB=1, the idea ratio from the interface register ASRIDRHB, ASRIDRLB will be used.It is suggested to manually set ASRCFG:POSTMODB, ASRCFG:PREMODB according to Table 15-7 in this case.

Table continues on the next page...

ASRC_ASRCTR field descriptions (continued)

Field	Description
14 USRA	Use Ratio for Pair A Use ratio as the input to ASRC. This bit is used in conjunction with IDRA control bit.
13 IDRA	Use Ideal Ratio for Pair A When USRA=0, this bit has no usage. When USRA=1 and IDRA=0, ASRC internal measured ratio will be used. When USRA=1 and IDRA=1, the idea ratio from the interface register ASRIDRHA, ASRIDRLA will be used. It is suggested to manually set ASRCFG:POSTMODA, ASRCFG:PREMODA according to Table 15-7 in this case.
12–5 -	This field is reserved. Reserved. Should be written as zero for compatibility.
4 SRST	Software Reset This bit is self-clear bit. Once it is been written as 1, it will generate a software reset signal inside ASRC. After 9 cycles of the ASRC processing clock, this reset process will stop, and this bit will be cleared automatically.
3 ASREC	ASRC Enable C Enable the operation of the conversion C of ASRC. When ASREC is cleared, operation of conversion C is disabled.
2 ASREB	ASRC Enable B Enable the operation of the conversion B of ASRC. When ASREB is cleared, operation of conversion B is disabled.
1 ASREA	ASRC Enable A Enable the operation of the conversion A of ASRC. When ASREA is cleared, operation of conversion A is disabled.
0 ASRCEN	ASRC Enable Enable the operation of ASRC.

15.7.2 ASRC Interrupt Enable Register (ASRC_ASRIER)

Address: 203_4000h base + 4h offset = 203_4004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								AFPWE	AOLIE	ADOEC	ADOEB	ADOEA	ADIEC	ADIEB	ADIEA
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRIER field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–8 -	This field is reserved. Reserved. Should be written as zero for compatibility.
7 AFPWE	FP in Wait State Interrupt Enable Enables the FP in wait state interrupt. 1 interrupt enabled 0 interrupt disabled
6 AOLIE	Overload Interrupt Enable Enables the overload interrupt. 1 interrupt enabled 0 interrupt disabled
5 ADOEC	Data Output C Interrupt Enable Enables the data output C interrupt. 1 interrupt enabled 0 interrupt disabled
4 ADOEB	Data Output B Interrupt Enable Enables the data output B interrupt. 1 interrupt enabled 0 interrupt disabled

Table continues on the next page...

ASRC_ASRIER field descriptions (continued)

Field	Description
3 ADOEA	Data Output A Interrupt Enable Enables the data output A interrupt. 1 interrupt enabled 0 interrupt disabled
2 ADIEC	Data Input C Interrupt Enable Enables the data input C interrupt. 1 interrupt enabled 0 interrupt disabled
1 ADIEB	Data Input B Interrupt Enable Enables the data input B interrupt. 1 interrupt enabled 0 interrupt disabled
0 ADIEA	Data Input A Interrupt Enable Enables the data input A Interrupt. 1 interrupt enabled 0 interrupt disabled

15.7.3 ASRC Channel Number Configuration Register (ASRC_ASRCNCR)

The ASRC channel number configuration register (ASRCNCR) is a read/write register that sets the number of channels used by each ASRC conversion pair.

There are 10 channels available for distribution among 3 conversion pairs, they are ordered as 0,1,...,9. The bottom [0, ANCA-1] channels are used for pair A, the top [10-ANCC, 9] channels are used for pair C, and the [ANCA, ANCA+ANCB-1] channels are allocated for pair B. In case that ANCA=0, then the [0, ANCB-1] channels are assigned for pair B.

Address: 203_4000h base + Ch offset = 203_400Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								Reserved												ANCC				ANCB				ANCA			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRCNCR field descriptions

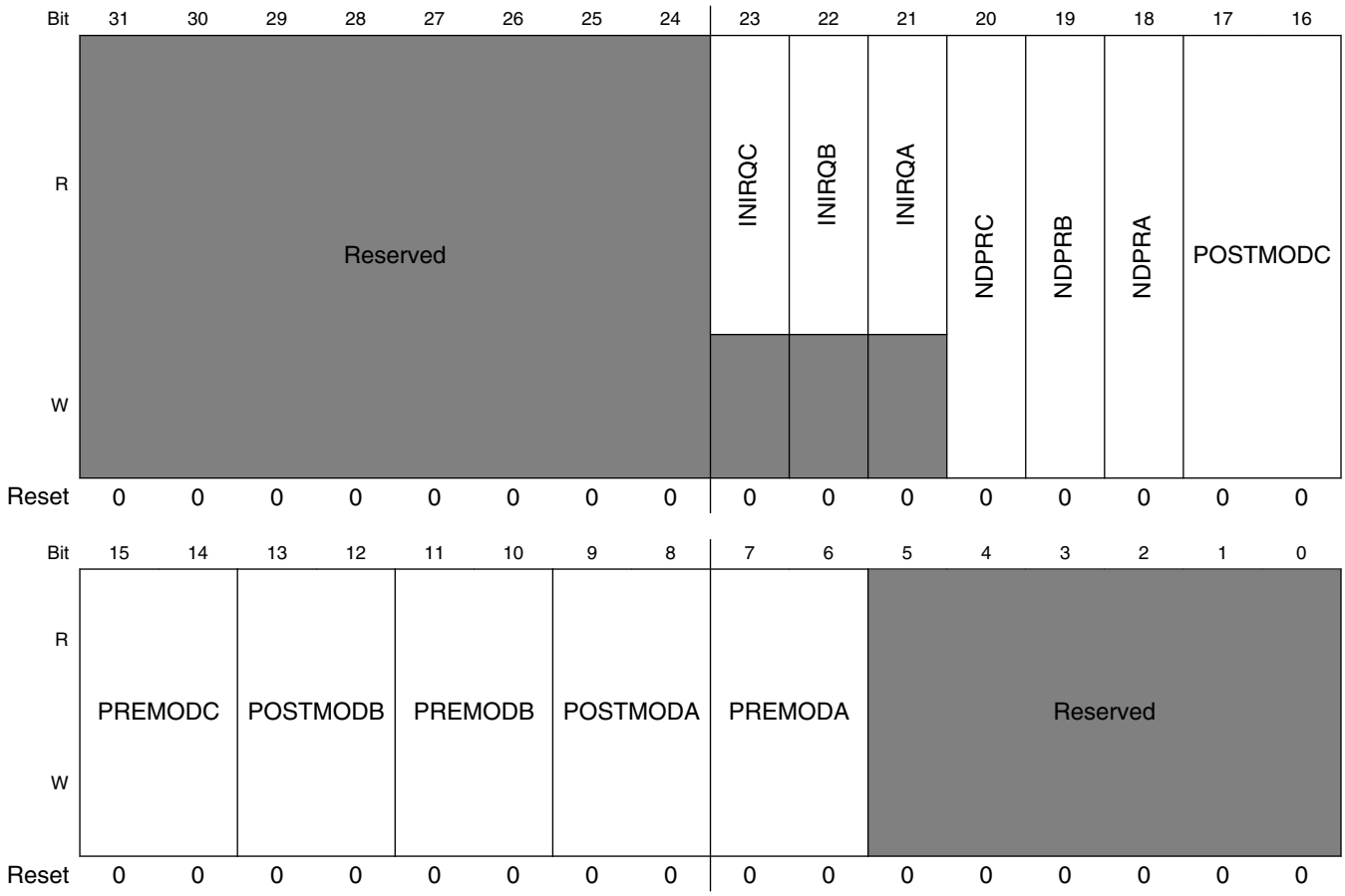
Field	Description
31–24 -	This field is reserved. Reserved
23–12 -	This field is reserved. Reserved. Should be written as zero for compatibility.
11–8 ANCC	Number of C Channels[1] 0000 0 channels in C (Pair C is disabled) 0001 1 channel in C 0010 2 channels in C 0011 3 channels in C 0100 4 channels in C 0101 5 channels in C 0110 6 channels in C 0111 7 channels in C 1000 8 channels in C 1001 9 channels in C 1010 10 channels in C 1011-1111 Should not be used.
7–4 ANCB	Number of B Channels 0000 0 channels in B (Pair B is disabled) 0001 1 channel in B 0010 2 channels in B 0011 3 channels in B 0100 4 channels in B 0101 5 channels in B 0110 6 channels in B 0111 7 channels in B 1000 8 channels in B 1001 9 channels in B 1010 10 channels in B 1011-1111 Should not be used.
ANCA	Number of A Channels 0000 0 channels in A (Pair A is disabled) 0001 1 channel in A 0010 2 channels in A 0011 3 channels in A 0100 4 channels in A 0101 5 channels in A 0110 6 channels in A 0111 7 channels in A 1000 8 channels in A 1001 9 channels in A 1010 10 channels in A 1011-1111 Should not be used.

1. ANCC+ANCB+ANCA<=10. Hardware is not checking the constraint. Programmer should take the responsibility to ensure the constraint is satisfied.

15.7.4 ASRC Filter Configuration Status Register (ASRC_ASRCFG)

The ASRC configuration status register (ASRCFG) is a read/write register that sets and/or automatically senses the ASRC operations.

Address: 203_4000h base + 10h offset = 203_4010h



ASRC_ASRCFG field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 INIRQC	Initialization for Conversion Pair C is served When this bit is 1, it means the initialization for conversion pair C is served. This bit is cleared by disabling the ASRC conversion pair (ASRCTR:ASREC=0 or ASRCTR:ASRCEN=0).
22 INIRQB	Initialization for Conversion Pair B is served When this bit is 1, it means the initialization for conversion pair B is served. This bit is cleared by disabling the ASRC conversion pair (ASRCTR:ASREB=0 or ASRCTR:ASRCEN=0).

Table continues on the next page...

ASRC_ASRCFG field descriptions (continued)

Field	Description
21 INIRQA	Initialization for Conversion Pair A is served When this bit is 1, it means the initialization for conversion pair A is served. This bit is cleared by disabling the ASRC conversion pair (ASRCTR:ASREA=0 or ASRCTR:ASRCEN=0).
20 NDPRC	Not Use Default Parameters for RAM-stored Parameters For Conversion Pair C 0 Use default parameters for RAM-stored parameters. Override any parameters already in RAM. 1 Don't use default parameters for RAM-stored parameters. Use the parameters already stored in RAM.
19 NDPRB	Not Use Default Parameters for RAM-stored Parameters For Conversion Pair B 0 Use default parameters for RAM-stored parameters. Override any parameters already in RAM. 1 Don't use default parameters for RAM-stored parameter. Use the parameters already stored in RAM.
18 NDPRA	Not Use Default Parameters for RAM-stored Parameters For Conversion Pair A 0 Use default parameters for RAM-stored parameters. Override any parameters already in RAM. 1 Don't use default parameters for RAM-stored parameters. Use the parameters already stored in RAM.
17–16 POSTMODC	Post-Processing Configuration for Conversion Pair C These bits will be read/write by user if ASRCTR:ATSC=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSC=1 (see ASRC Misc Control Register 1 for Pair C). These bits set the selection of the post-processing configuration. 00 Select Upsampling-by-2 as defined in Signal Processing Flow. 01 Select Direct-Connection as defined in Signal Processing Flow. 10 Select Downsampling-by-2 as defined in Signal Processing Flow.
15–14 PREMODC	Pre-Processing Configuration for Conversion Pair C These bits will be read/write by user if ASRCTR:ATSC=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSC=1 (see ASRC Misc Control Register 1 for Pair C). These bits set the selection of the pre-processing configuration. 00 Select Upsampling-by-2 as defined in Signal processing flow 01 Select Direct-Connection as defined in Signal processing flow 10 Select Downsampling-by-2 as defined in Signal processing flow 11 Select passthrough mode. In this case, POSTMODC[1-0] have no use.
13–12 POSTMODB	Post-Processing Configuration for Conversion Pair B These bits will be read/write by user if ASRCTR:ATSB=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSB=1 (see ASRC Misc Control Register 1 for Pair C). These bits set the selection of the post-processing configuration. 00 Select Upsampling-by-2 as defined in Signal processing flow 01 Select Direct-Connection as defined in Signal processing flow 10 Select Downsampling-by-2 as defined in Signal processing flow
11–10 PREMODB	Pre-Processing Configuration for Conversion Pair B These bits will be read/write by user if ASRCTR:ATSB=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSB=1 (see ASRC Misc Control Register 1 for Pair C). These bits set the selection of the pre-processing configuration. 00 Select Upsampling-by-2 as defined in Signal processing flow 01 Select Direct-Connection as defined in Signal processing flow

Table continues on the next page...

ASRC_ASRCFG field descriptions (continued)

Field	Description
	10 Select Downsampling-by-2 as defined in Signal processing flow 11 Select passthrough mode. In this case, POSTMODB[1-0] have no use.
9–8 POSTMODA	Post-Processing Configuration for Conversion Pair A These bits will be read/write by user if ASRCTR:ATSA=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSA=1 (see ASRC Misc Control Register 1 for Pair C). These bits set the selection of the post-processing configuration. 00 Select Upsampling-by-2 as defined in Signal processing flow 01 Select Direct-Connection as defined in Signal processing flow 10 Select Downsampling-by-2 as defined in Signal processing flow
7–6 PREMODA	Pre-Processing Configuration for Conversion Pair A These bits will be read/write by user if ASRCTR:ATSA=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSA=1 (see ASRC Misc Control Register 1 for Pair C). These bits set the selection of the pre-processing configuration. 00 Select Upsampling-by-2 as defined in Signal processing flow 01 Select Direct-Connection as defined in Signal processing flow 10 Select Downsampling-by-2 as defined in Signal processing flow 11 Select passthrough mode. In this case, POSTMODA[1-0] have no use.
-	This field is reserved. Reserved. Should be written as zero for compatibility.

15.7.5 ASRC Clock Source Register (ASRC_ASRCSR)

The ASRC clock source register (ASRCSR) is a read/write register that controls the sources of the input and output clocks of the ASRC.

Address: 203_4000h base + 14h offset = 203_4014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								AOCSC				AOCSB				AOCSA				AICSC				AICSB				AICSA			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRCSR field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–20 AOCSC	**Output Clock Source C** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3

Table continues on the next page...

ASRC_ASRCSR field descriptions (continued)

Field	Description
	0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D 1110 bit clock E 1111 clock disabled, connected to zero
19–16 AOCSB	**Output Clock Source B** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D 1110 bit clock E 1111 clock disabled, connected to zero
15–12 AOCSA	**Output Clock Source A** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D

Table continues on the next page...

ASRC_ASRCSR field descriptions (continued)

Field	Description
	1110 bit clock E 1111 clock disabled, connected to zero
11–8 AICSC	**Input Clock Source C** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D 1110 bit clock E 1111 clock disabled, connected to zero
7–4 AICSB	**Input Clock Source B** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D 1110 bit clock E 1111 clock disabled, connected to zero
AICSA	**Input Clock Source A** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6

Table continues on the next page...

ASRC_ASRCSR field descriptions (continued)

Field	Description
	0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D 1110 bit clock E 1111 clock disabled, connected to zero

15.7.6 ASRC Clock Divider Register 1 (ASRC_ASRCDR1)

The ASRC clock divider register (ASRCDR1) is a read/write register that controls the division factors of the ASRC input and output clock sources.

Address: 203_4000h base + 18h offset = 203_4018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								AOCDB			AOCPB			AOCDA		AOCPA				AICDB			AICPB			AICDA			AICPA			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRCDR1 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–21 AOCDB	Output Clock Divider B Specify the divide ratio of the output clock divider B. The divide ratio may range from 1 to 8 (AOCDB[2:0] = 000 to 111).
20–18 AOCPB	Output Clock Prescaler B Specify the prescaling factor of the output prescaler B. The prescaling ratio may be any power of 2 from 1 to 128.
17–15 AOCDA	Output Clock Divider A Specify the divide ratio of the output clock divider A. The divide ratio may range from 1 to 8 (AOCDA[2:0] = 000 to 111).
14–12 AOCPA	Output Clock Prescaler A Specify the prescaling factor of the output prescaler A. The prescaling ratio may be any power of 2 from 1 to 128.
11–9 AICDB	Input Clock Divider B Specify the divide ratio of the input clock divider B. The divide ratio may range from 1 to 8 (AICDB[2:0] = 000 to 111).

Table continues on the next page...

ASRC_ASRCDR1 field descriptions (continued)

Field	Description
8–6 AICPB	Input Clock Prescaler B Specify the prescaling factor of the input prescaler B. The prescaling ratio may be any power of 2 from 1 to 128.
5–3 AICDA	Input Clock Divider A Specify the divide ratio of the input clock divider A. The divide ratio may range from 1 to 8 (AICDA[2:0] = 000 to 111).
AICPA	Input Clock Prescaler A Specify the prescaling factor of the input prescaler A. The prescaling ratio may be any power of 2 from 1 to 128.

15.7.7 ASRC Clock Divider Register 2 (ASRC_ASRCDR2)

The ASRC clock divider register (ASRCDR2) is a read/write register that controls the division factors of the ASRC input and output clock sources.

Address: 203_4000h base + 1Ch offset = 203_401Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								Reserved												AOCDC			AOCPC			AICDC			AICPC		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

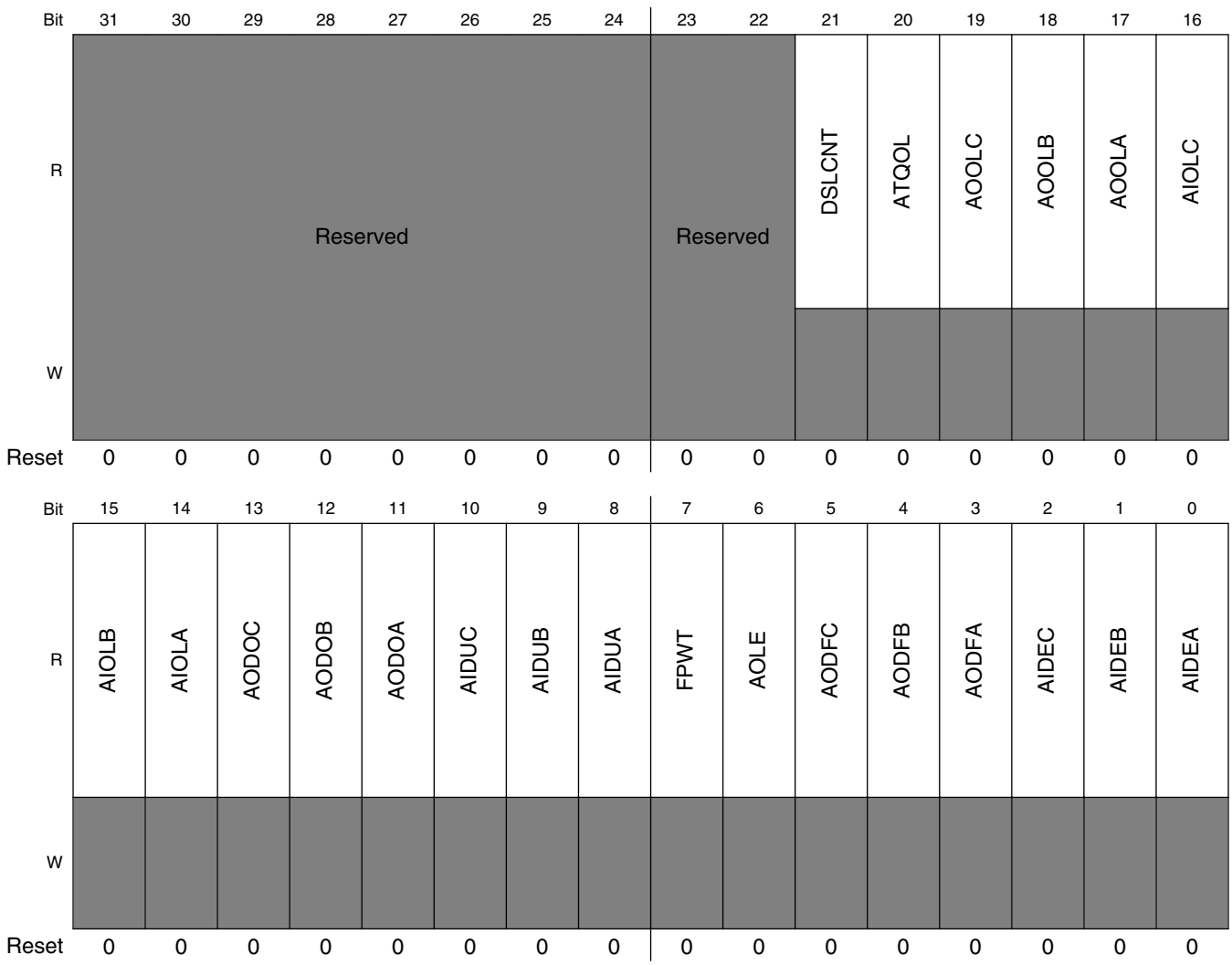
ASRC_ASRCDR2 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–12 -	This field is reserved. Reserved. Should be written as zero for compatibility.
11–9 AOCDC	Output Clock Divider C Specify the divide ratio of the output clock divider C. The divide ratio may range from 1 to 8 (AOCDC[2:0] = 000 to 111).
8–6 AOCPC	Output Clock Prescaler C Specify the prescaling factor of the output prescaler C. The prescaling ratio may be any power of 2 from 1 to 128.
5–3 AICDC	Input Clock Divider C Specify the divide ratio of the input clock divider C. The divide ratio may range from 1 to 8 (AICDC[2:0] = 000 to 111).
AICPC	Input Clock Prescaler C Specify the prescaling factor of the input prescaler C. The prescaling ratio may be any power of 2 from 1 to 128.

15.7.8 ASRC Status Register (ASRC_ASRSTR)

The ASRC status register (ASRSTR) is a read/write register used by the processor core to examine the status of the ASRC block and clear the overload interrupt request and AOLE flag bit. Read the status register will return the current state of ASRC.

Address: 203_4000h base + 20h offset = 203_4020h



ASRC_ASRSTR field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–22 -	This field is reserved. Reserved. Should be written as zero for compatibility.

Table continues on the next page...

ASRC_ASRSTR field descriptions (continued)

Field	Description
21 DSLCNT	DSL Counter Input to FIFO ready When set, this bit indicates that new DSL counter information is stored in the internal ASRC FIFO. When clear, this bit indicates that new DSL counter information is in the process of storage into the internal ASRC FIFO. When ASRIER:AFPWE=1, the rising edge of this signal will propose an interrupt request. Writing any value with this bit set will clear the interrupt request proposed by the rising edge of this bit.
20 ATQOL	Task Queue FIFO overload When set, this bit indicates that task queue FIFO logic is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRSTR:AOLE as 1.
19 AOOLC	Pair C Output Task Overload When set, this bit indicates that pair C output task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRSTR:AOLE as 1.
18 AOOLB	Pair B Output Task Overload When set, this bit indicates that pair B output task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRSTR:AOLE as 1.
17 AOOLA	Pair A Output Task Overload When set, this bit indicates that pair A output task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRSTR:AOLE as 1.
16 AIOLC	Pair C Input Task Overload When set, this bit indicates that pair C input task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRSTR:AOLE as 1.
15 AIOLB	Pair B Input Task Overload When set, this bit indicates that pair B input task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRSTR:AOLE as 1.
14 AIOLA	Pair A Input Task Overload When set, this bit indicates that pair A input task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRSTR:AOLE as 1.
13 AODOC	Output Data Buffer C has overflowed When set, this bit indicates that output data buffer C has overflowed. When clear, this bit indicates that output data buffer C has not overflowed The bit is cleared when writing ASRSTR:AOLE as 1.
12 AODOB	Output Data Buffer B has overflowed When set, this bit indicates that output data buffer B has overflowed. When clear, this bit indicates that output data buffer B has not overflowed

Table continues on the next page...

ASRC_ASRSTR field descriptions (continued)

Field	Description
	The bit is cleared when writing ASRSTR:AOLE as 1.
11 AODOA	Output Data Buffer A has overflowed When set, this bit indicates that output data buffer A has overflowed. When clear, this bit indicates that output data buffer A has not overflowed The bit is cleared when writing ASRSTR:AOLE as 1.
10 AIDUC	Input Data Buffer C has underflowed When set, this bit indicates that input data buffer C has underflowed. When clear, this bit indicates that input data buffer C has not underflowed. The bit is cleared when writing ASRSTR:AOLE as 1.
9 AIDUB	Input Data Buffer B has underflowed When set, this bit indicates that input data buffer B has underflowed. When clear, this bit indicates that input data buffer B has not underflowed. The bit is cleared when writing ASRSTR:AOLE as 1.
8 AIDUA	Input Data Buffer A has underflowed When set, this bit indicates that input data buffer A has underflowed. When clear, this bit indicates that input data buffer A has not underflowed. The bit is cleared when writing ASRSTR:AOLE as 1.
7 FPWT	FP is in wait states This bit is for debug only. When set, this bit indicates that ASRC is in wait states. When clear, this bit indicates that ASRC is not in wait states.
6 AOLE	Overload Error Flag When set, this bit indicates that the task rate is too high for the ASRC to handle. The reasons for overload may be: - too high input clock frequency, - too high output clock frequency, - incorrect selection of the pre-filter, - low ASRC processing clock, - too many channels, - underrun, - or any combination of the reasons above. Since the ASRC uses the same hardware resources to perform various tasks, the real reason for the overload is not straight forward, and it should be carefully analyzed by the programmer. If ASRIER:AOLIE=1, an interrupt will be proposed when this bit is set. Write any value with this bit set as one into the status register will clear this bit and the interrupt request proposed by this bit.
5 AODFC	Number of data in Output Data Buffer C is greater than threshold

Table continues on the next page...

ASRC_ASRSTR field descriptions (continued)

Field	Description
	When set, this bit indicates that number of data already existing in ASRDORC is greater than threshold and the processor can read data from ASRDORC. When AODFC is set, the ASRC generates data output C interrupt request to the processor, if enabled (that is, ASRIER:ADOEC = 1). A DMA request is always generated when the AODFC bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.
4 AODFB	Number of data in Output Data Buffer B is greater than threshold When set, this bit indicates that number of data already existing in ASRDORB is greater than threshold and the processor can read data from ASRDORB. When AODFB is set, the ASRC generates data output B interrupt request to the processor, if enabled (that is, ASRIER:ADOEB = 1). A DMA request is always generated when the AODFB bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.
3 AODFA	Number of data in Output Data Buffer A is greater than threshold When set, this bit indicates that number of data already existing in ASRDORA is greater than threshold and the processor can read data from ASRDORA. When AODFA is set, the ASRC generates data output A interrupt request to the processor, if enabled (that is, ASRIER:ADOEA = 1). A DMA request is always generated when the AODFA bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.
2 AIDEC	Number of data in Input Data Buffer C is less than threshold When set, this bit indicates that number of data still available in ASRDIRC is less than threshold and the processor can write data to ASRDIRC. When AIDEC is set, the ASRC generates data input C interrupt request to the processor, if enabled (that is, ASRIER:ADIEC = 1). A DMA request is always generated when the AIDEC bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.
1 AIDEB	Number of data in Input Data Buffer B is less than threshold When set, this bit indicates that number of data still available in ASRDIRB is less than threshold and the processor can write data to ASRDIRB. When AIDEB is set, the ASRC generates data input B interrupt request to the processor, if enabled (that is, ASRIER:ADIEB = 1). A DMA request is always generated when the AIDEB bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.
0 AIDEA	Number of data in Input Data Buffer A is less than threshold When set, this bit indicates that number of data still available in ASRDIRA is less than threshold and the processor can write data to ASRDIRA. When AIDEA is set, the ASRC generates data input A interrupt request to the processor, if enabled (that is, ASRIER:ADIEA = 1). A DMA request is always generated when the AIDEA bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.

15.7.9 ASRC Parameter Register n (ASRC_ASRPMn*n*)

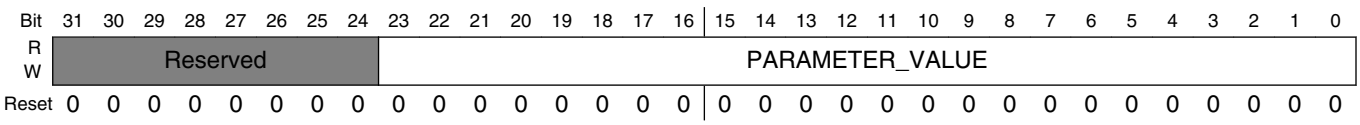
Parameter registers determine the performance of ASRC.

The parameter registers must be initialized by software before ASRC is enabled. Recommended values are given in ASRC Misc Control Register 1 for Pair C below,

Table 15-8. ASRC Parameter Registers (ASRPM1~ASRPM5)

Register	Offset	Access	Reset Value	Recommend Value
asrcpm1	0x40	R/W	0x00_0000	0x7fffff
asrcpm2	0x44	R/W	0x00_0000	0x255555
asrcpm3	0x48	R/W	0x00_0000	0xff7280
asrcpm4	0x4C	R/W	0x00_0000	0xff7280
asrcpm5	0x50	R/W	0x00_0000	0xff7280

Address: 203_4000h base + 40h offset + (4d × i), where i=0d to 4d



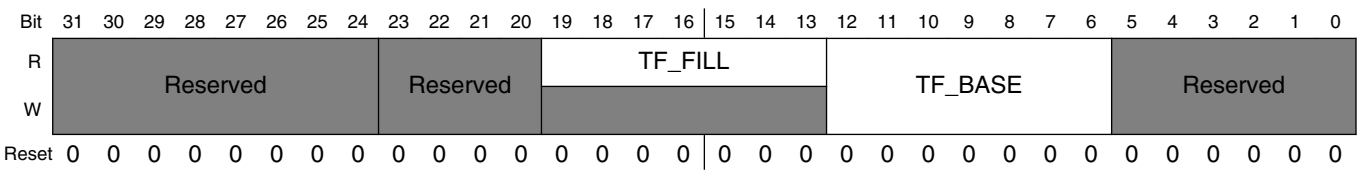
ASRC_ASRPMn*n* field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
PARAMETER_ VALUE	See recommended values table.

15.7.10 ASRC ASRC Task Queue FIFO Register 1 (ASRC_ASRTFR1)

The register defines and shows the parameters for ASRC inner task queue FIFOs.

Address: 203_4000h base + 54h offset = 203_4054h



ASRC_ASRTFR1 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–20 -	This field is reserved. Reserved. Should be written as zero for compatibility.
19–13 TF_FILL	Current number of entries in task queue FIFO.

Table continues on the next page...

ASRC_ASRTFR1 field descriptions (continued)

Field	Description
12–6 TF_BASE	Base address for task queue FIFO. Set to 0x7C.
-	This field is reserved. Reserved. Should be written as zero for compatibility.

15.7.11 ASRC Channel Counter Register (ASRC_ASRCCR)

The ASRC channel counter register (ASRCCR) is a read/write register that sets and reflects the current specific input/output FIFO being accessed through shared peripheral bus for each ASRC conversion pair.

Address: 203_4000h base + 5Ch offset = 203_405Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								ACOC				ACOB				ACOA				ACIC				ACIB				ACIA			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

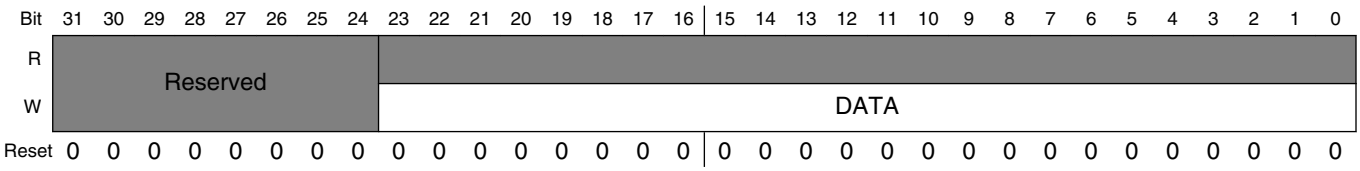
ASRC_ASRCCR field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–20 ACOC	The channel counter for Pair C's output FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair C's output FIFO's usage. The value can be any value between [0, ANCC-1]
19–16 ACOB	The channel counter for Pair B's output FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair B's output FIFO's usage. The value can be any value between [0, ANCB-1]
15–12 ACOA	The channel counter for Pair A's output FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair A's output FIFO's usage. The value can be any value between [0, ANCA-1]
11–8 ACIC	The channel counter for Pair C's input FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair C's input FIFO's usage. The value can be any value between [0, ANCC-1]
7–4 ACIB	The channel counter for Pair B's input FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair B's input FIFO's usage. The value can be any value between [0, ANCB-1]
ACIA	The channel counter for Pair A's input FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair A's input FIFO's usage. The value can be any value between [0, ANCA-1]

15.7.12 ASRC Data Input Register for Pair *x* (ASRC_ASRDI*n*)

These registers are the interface registers for the audio data input of pair A,B,C respectively. They are backed by FIFOs.

Address: 203_4000h base + 60h offset + (8d × i), where i=0d to 2d



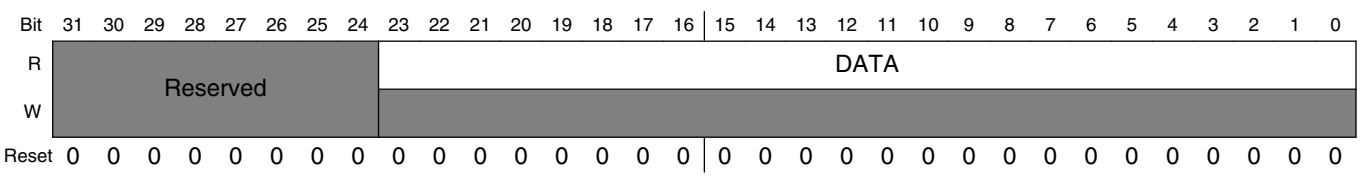
ASRC_ASRDI*n* field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
DATA	Audio data input

15.7.13 ASRC Data Output Register for Pair *x* (ASRC_ASRDO*n*)

These registers are the interface registers for the audio data output of pair A,B,C respectively. They are backed by FIFOs.

Address: 203_4000h base + 64h offset + (8d × i), where i=0d to 2d



ASRC_ASRDO*n* field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
DATA	Audio data output

15.7.14 ASRC Ideal Ratio for Pair A-High Part (ASRC_ASRIDRHA)

The ideal ratio registers (ASRIDRHA, ASRIDRLA) hold the ratio value IDRATIOA. IDRATIOA = $Fs_{inA}/Fs_{outA} = Ts_{outA}/Ts_{inA}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRA, IDRA}=2'b11.

Address: 203_4000h base + 80h offset = 203_4080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								Reserved																IDRATIOA_H							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRIDRHA field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–8 -	This field is reserved. Reserved
IDRATIOA_H	IDRATIOA[31:24]. High part of ideal ratio value for pair A

15.7.15 ASRC Ideal Ratio for Pair A -Low Part (ASRC_ASRIDRLA)

The ideal ratio registers (ASRIDRHA, ASRIDRLA) hold the ratio value IDRATIOA. IDRATIOA = $Fs_{inA}/Fs_{outA} = Ts_{outA}/Ts_{inA}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRA, IDRA}=2'b11.

Address: 203_4000h base + 84h offset = 203_4084h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								IDRATIOA_L																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRIDRLA field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
IDRATIOA_L	IDRATIOA[23:0]. Low part of ideal ratio value for pair A

15.7.16 ASRC Ideal Ratio for Pair B-High Part (ASRC_ASRIDRHB)

The ideal ratio registers (ASRIDRHB, ASRIDRLB) hold the ratio value IDRATIOB. IDRATIOB = $Fs_{inB}/Fs_{outB} = Ts_{outB}/Ts_{inB}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRB, IDRB}=2'b11.

Address: 203_4000h base + 88h offset = 203_4088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								Reserved																IDRATIOB_H							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRIDRHB field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–8 -	This field is reserved. Reserved
IDRATIOB_H	IDRATIOB[31:24]. High part of ideal ratio value for pair B.

15.7.17 ASRC Ideal Ratio for Pair B-Low Part (ASRC_ASRIDRLB)

The ideal ratio registers (ASRIDRHB, ASRIDRLB) hold the ratio value IDRATIOB. IDRATIOB = $Fs_{inB}/Fs_{outB} = Ts_{outB}/Ts_{inB}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRB, IDRB}=2'b11.

Address: 203_4000h base + 8Ch offset = 203_408Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								IDRATIOB_L																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

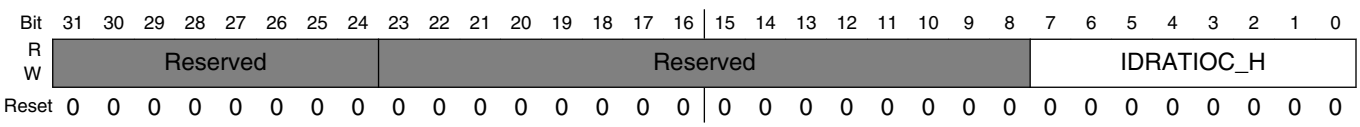
ASRC_ASRIDRLB field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
IDRATIOB_L	IDRATIOB[23:0]. Low part of ideal ratio value for pair B.

15.7.18 ASRC Ideal Ratio for Pair C-High Part (ASRC_ASRIDRHC)

The ideal ratio registers (ASRIDRHC, ASRIDRLC) hold the ratio value IDRATIOC. IDRATIOC = $Fs_{inC}/Fs_{outC} = Ts_{outC}/Ts_{inC}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRC, IDRC}=2'b11.

Address: 203_4000h base + 90h offset = 203_4090h



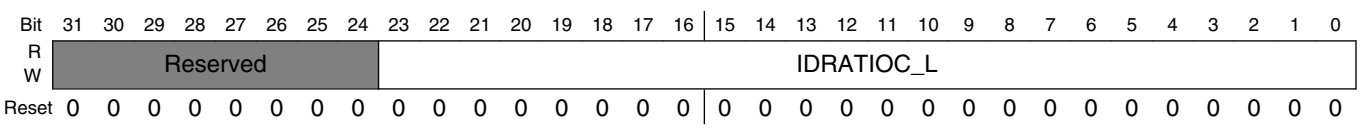
ASRC_ASRIDRHC field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–8 -	This field is reserved. Reserved
IDRATIOC_H	IDRATIOC[31:24]. High part of ideal ratio value for pair C.

15.7.19 ASRC Ideal Ratio for Pair C-Low Part (ASRC_ASRIDRLC)

The ideal ratio registers (ASRIDRHC, ASRIDRLC) hold the ratio value IDRATIOC. IDRATIOC = $Fs_{inC}/Fs_{outC} = Ts_{outC}/Ts_{inC}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRC, IDRC}=2'b11.

Address: 203_4000h base + 94h offset = 203_4094h



ASRC_ASRIDRLC field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
IDRATIOC_L	IDRATIOC[23:0]. Low part of ideal ratio value for pair C.

15.7.20 ASRC 76 kHz Period in terms of ASRC processing clock (ASRC_ASR76K)

The register (ASR76K) holds the period of the 76 kHz sampling clock in terms of the ASRC processing clock with frequency Fs_{ASRC}. ASR76K = Fs_{ASRC}/Fs_{76k}. Reset value is 0x0A47 which assumes that Fs_{ASRC}=200 MHz. This register is used to help the ASRC internal logic to decide the pre-processing and the post-processing options automatically (see ASRC Misc Control Register 1 for Pair C and ASRC Misc Control Register 1 for Pair C). In a system when Fs_{ASRC} = 133 MHz, the value should be assigned explicitly as 0x06D6 in user application code.

Address: 203_4000h base + 98h offset = 203_4098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								Reserved								ASR76K															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	1	1

ASRC_ASR76K field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–17 -	This field is reserved. Reserved
ASR76K	Value for the period of the 76 kHz sampling clock.

15.7.21 ASRC 56 kHz Period in terms of ASRC processing clock (ASRC_ASR56K)

The register (ASR56K) holds the period of the 56 kHz sampling clock in terms of the ASRC processing clock with frequency Fs_{ASRC}. ASR56K = Fs_{ASRC}/Fs_{56k}. Reset value is 0x0DF3 which assumes that Fs_{ASRC} = 200 MHz. This register is used to help the ASRC internal logic to decide the pre-processing and the post-processing options automatically (see ASRC Misc Control Register 1 for Pair C and ASRC Misc Control Register 1 for Pair C). In a system when Fs_{ASRC} = 133 MHz, the value should be assigned explicitly as 0x0947 in user application code.

Address: 203_4000h base + 9Ch offset = 203_409Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								Reserved								ASR56K															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	1	0	0	1	1

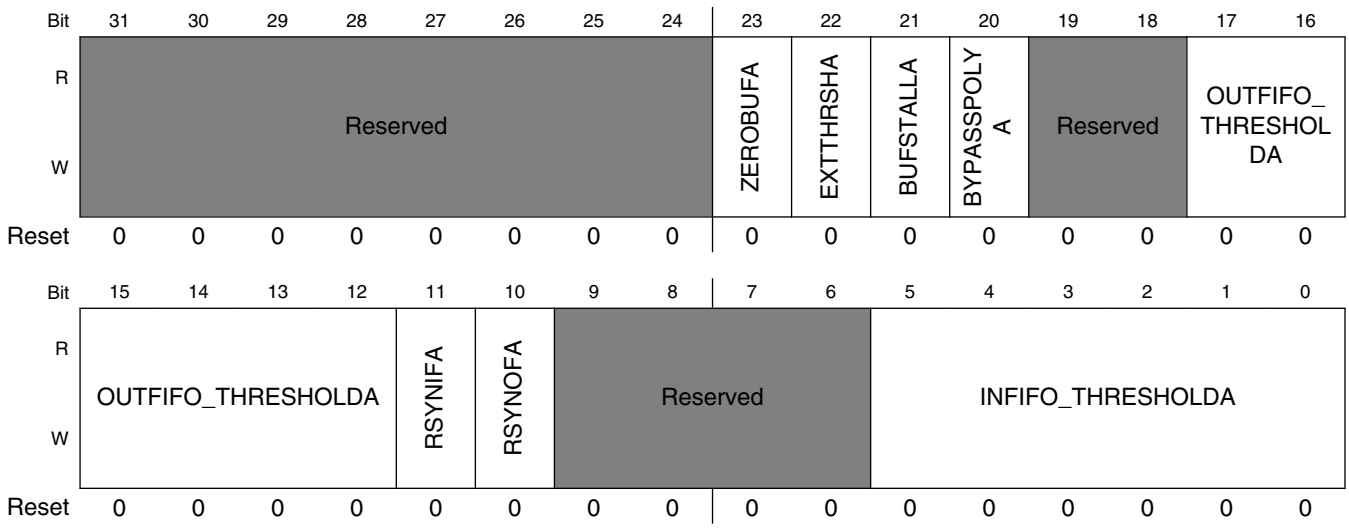
ASRC_ASR56K field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–17 -	This field is reserved. Reserved
ASR56K	Value for the period of the 56 kHz sampling clock

15.7.22 ASRC Misc Control Register for Pair A (ASRC_ASRMCRA)

The register (ASRMCRA) is used to control Pair A internal logic.

Address: 203_4000h base + A0h offset = 203_40A0h



ASRC_ASRMCRA field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 ZEROBUFA	Initialize buf of Pair A when pair A is enabled. Always clear option. This bit is used to control whether the buffer is to be zeroized when pair A is enabled. 1 Don't zeroize the buffer 0 Zeroize the buffer
22 EXTTHRSHA	Use external thresholds for FIFO control of Pair A This bit will determine whether the FIFO thresholds externally defined in this register is used to control ASRC internal FIFO logic for pair A. 1 Use external defined thresholds. 0 Use default thresholds.
21 BUFSTALLA	Stall Pair A conversion in case of Buffer Near Empty/Full Condition This bit will determine whether the near empty/full FIFO condition will stall the rate conversion for pair A. This option can only work when external ratio is used. Near empty condition is the condition when input FIFO has less than 4 useful samples per channel. Near full condition is the condition when the output FIFO has less than 4 vacant sample words to fill per channel.

Table continues on the next page...

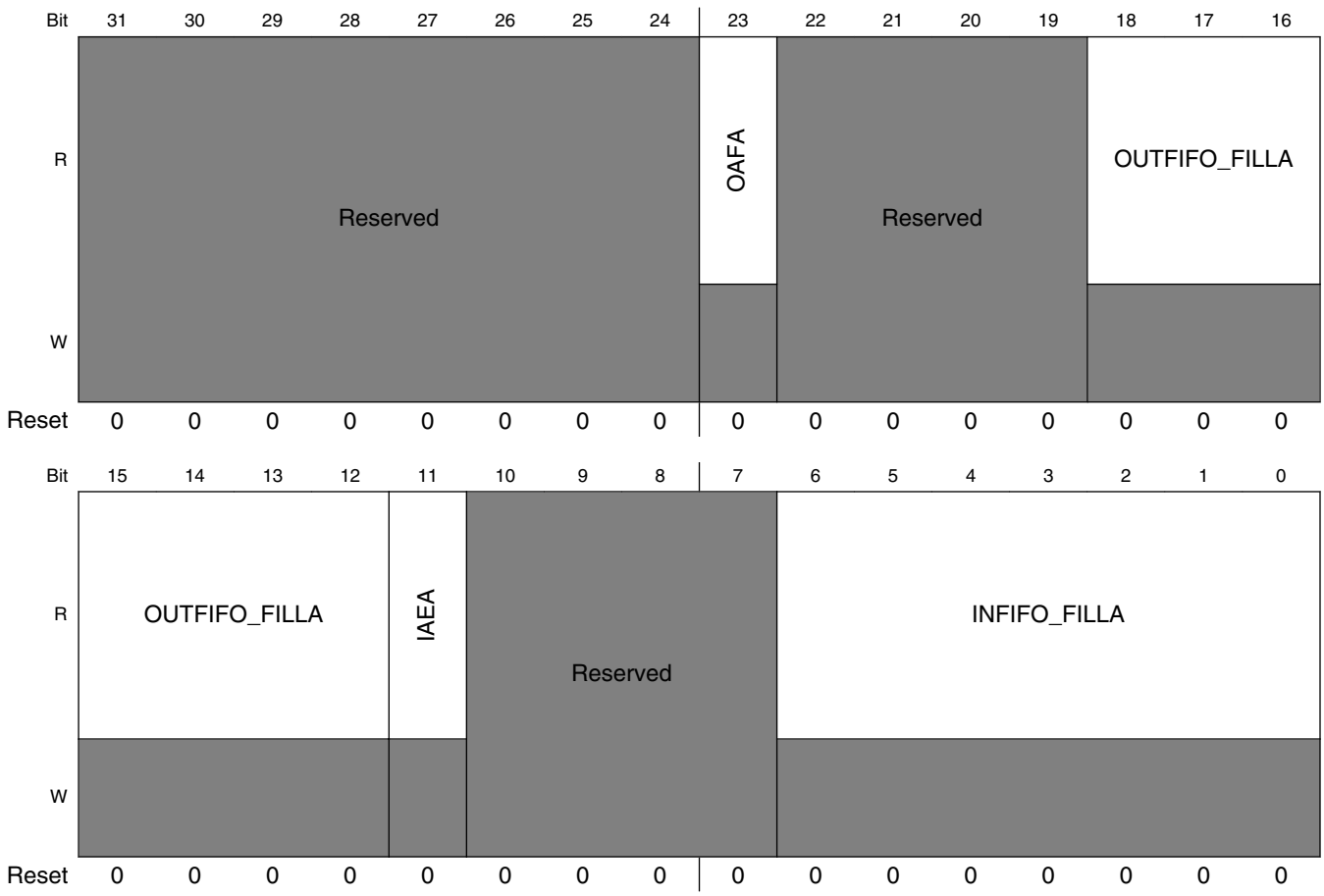
ASRC_ASRMCRA field descriptions (continued)

Field	Description
	1 Stall Pair A conversion in case of near empty/full FIFO conditions. 0 Don't stall Pair A conversion even in case of near empty/full FIFO conditions.
20 BYPASSPOLYA	Bypass Polyphase Filtering for Pair A This bit will determine whether the polyphase filtering part of Pair A conversion will be bypassed. 1 Bypass polyphase filtering. 0 Don't bypass polyphase filtering.
19–18 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
17–12 OUTFIFO_ THRESHOLDA	The threshold for Pair A's output FIFO per channel These bits stand for the threshold for Pair A's output FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of output FIFO fillings of the pair is greater than n samples per channel, the output data ready flag is set; when the number of output FIFO fillings of the pair is less than or equal to n samples per channel, the output data ready flag is automatically cleared.
11 RSYNIFA	Re-sync Input FIFO Channel Counter If bit set, force ASRCCR:ACIA=0. If bit clear, untouch ASRCCR:ACIA.
10 RSYNOFA	Re-sync Output FIFO Channel Counter If bit set, force ASRCCR:ACOA=0. If bit clear, untouch ASRCCR:ACOA.
9–6 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
INFIFO_ THRESHOLDA	The threshold for Pair A's input FIFO per channel These bits stand for the threshold for Pair A's input FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of input FIFO fillings of the pair is less than n samples per channel, the input data needed flag is set; when the number of input FIFO fillings of the pair is greater than or equal to n samples per channel, the input data needed flag is automatically cleared.

15.7.23 ASRC FIFO Status Register for Pair A (ASRC_ASRFSTA)

The register (ASRFSTA) is used to show Pair A internal FIFO conditions.

Address: 203_4000h base + A4h offset = 203_40A4h



ASRC_ASRFSTA field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 OAFA	Output FIFO is near Full for Pair A This bit is to indicate whether the output FIFO of Pair A is near full.
22–19 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
18–12 OUTFIFO_FILLA	The fillings for Pair A's output FIFO per channel These bits stand for the fillings for Pair A's output FIFO per channel. Possible range is [0,64].

Table continues on the next page...

ASRC_ASRFSTA field descriptions (continued)

Field	Description
11 IAEA	Input FIFO is near Empty for Pair A This bit is to indicate whether the input FIFO of Pair A is near empty.
10–7 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
INFIFO_FILLA	The fillings for Pair A's input FIFO per channel These bits stand for the fillings for Pair A's input FIFO per channel. Possible range is [0,64].

15.7.24 ASRC Misc Control Register for Pair B (ASRC_ASRMCRB)

The register (ASRMCRB) is used to control Pair B internal logic.

Address: 203_4000h base + A8h offset = 203_40A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								ZEROBUFB	EXTTHRSHB	BUFSTALLB	BYPASSPOLYB	Reserved		OUTFIFO_THRESHOLDB	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OUTFIFO_THRESHOLDB				RSYNIFB	RSYNOFB	Reserved				INFIFO_THRESHOLDB					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRMCRB field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 ZEROBUFB	Initialize buf of Pair B when pair B is enabled This bit is used to control whether the buffer is to be zeroized when pair B is enabled. 1 Don't zeroize the buffer 0 Zeroize the buffer
22 EXTTHRSHB	Use external thresholds for FIFO control of Pair B

Table continues on the next page...

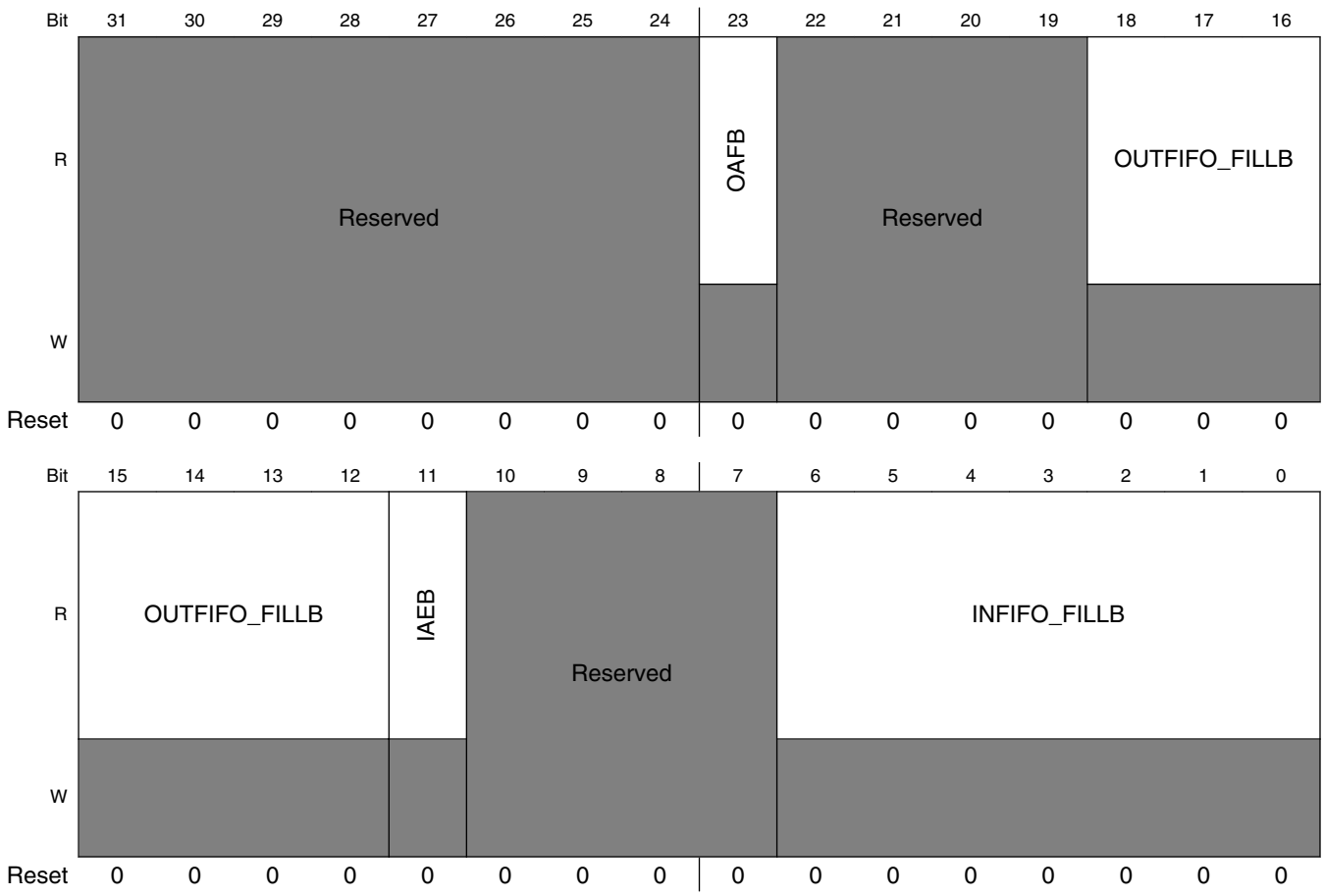
ASRC_ASRMCRB field descriptions (continued)

Field	Description
	This bit will determine whether the FIFO thresholds externally defined in this register is used to control ASRC internal FIFO logic for pair B. 1 Use external defined thresholds. 0 Use default thresholds.
21 BUFSTALLB	Stall Pair B conversion in case of Buffer Near Empty/Full Condition This bit will determine whether the near empty/full FIFO condition will stall the rate conversion for pair B. This option can only work when external ratio is used. Near empty condition is the condition when input FIFO has less than 4 useful samples per channel. Near full condition is the condition when the output FIFO has less than 4 vacant sample words to fill per channel. 1 Stall Pair B conversion in case of near empty/full FIFO conditions. 0 Don't stall Pair B conversion even in case of near empty/full FIFO conditions.
20 BYPASSPOLYB	Bypass Polyphase Filtering for Pair B This bit will determine whether the polyphase filtering part of Pair B conversion will be bypassed. 1 Bypass polyphase filtering. 0 Don't bypass polyphase filtering.
19–18 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
17–12 OUTFIFO_ THRESHOLDB	The threshold for Pair B's output FIFO per channel These bits stand for the threshold for Pair B's output FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of output FIFO fillings of the pair is greater than n samples per channel, the output data ready flag is set; when the number of output FIFO fillings of the pair is less than or equal to n samples per channel, the output data ready flag is automatically cleared.
11 RSYNIFB	Re-sync Input FIFO Channel Counter If bit set, force ASRCCR:ACIB=0. If bit clear, untouch ASRCCR:ACIB.
10 RSYNOFB	Re-sync Output FIFO Channel Counter If bit set, force ASRCCR:ACOB=0. If bit clear, untouch ASRCCR:ACOB.
9–6 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
INFIFO_ THRESHOLDB	The threshold for Pair B's input FIFO per channel These bits stand for the threshold for Pair B's input FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of input FIFO fillings of the pair is less than n samples per channel, the input data needed flag is set; when the number of input FIFO fillings of the pair is greater than or equal to n samples per channel, the input data needed flag is automatically cleared.

15.7.25 ASRC FIFO Status Register for Pair B (ASRC_ASRFSTB)

The register (ASRFSTB) is used to show Pair B internal FIFO conditions.

Address: 203_4000h base + ACh offset = 203_40ACh



ASRC_ASRFSTB field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 OAFB	Output FIFO is near Full for Pair B This bit is to indicate whether the output FIFO of Pair B is near full.
22–19 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
18–12 OUTFIFO_FILLB	The fillings for Pair B's output FIFO per channel These bits stand for the fillings for Pair B's output FIFO per channel. Possible range is [0,64].

Table continues on the next page...

ASRC_ASRFSTB field descriptions (continued)

Field	Description
11 IAEB	Input FIFO is near Empty for Pair B This bit is to indicate whether the input FIFO of Pair B is near empty.
10–7 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
INFIFO_FILLB	The fillings for Pair B's input FIFO per channel These bits stand for the fillings for Pair B's input FIFO per channel. Possible range is [0,64].

15.7.26 ASRC Misc Control Register for Pair C (ASRC_ASRMCRC)

The register (ASRMCRC) is used to control Pair C internal logic.

Address: 203_4000h base + B0h offset = 203_40B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								ZEROBUFC	EXTTHRSHC	BUFSTALLC	BYPASSPOLYC	Reserved		OUTFIFO_THRESHOLDC	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OUTFIFO_THRESHOLDC				RSYNIFC	RSYNOFC	Reserved				INFIFO_THRESHOLDC					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRMCRC field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 ZEROBUFC	Initialize buf of Pair C when pair C is enabled This bit is used to control whether the buffer is to be zeroized when pair C is enabled. 1 Don't zeroize the buffer 0 Zeroize the buffer
22 EXTTHRSHC	Use external thresholds for FIFO control of Pair C

Table continues on the next page...

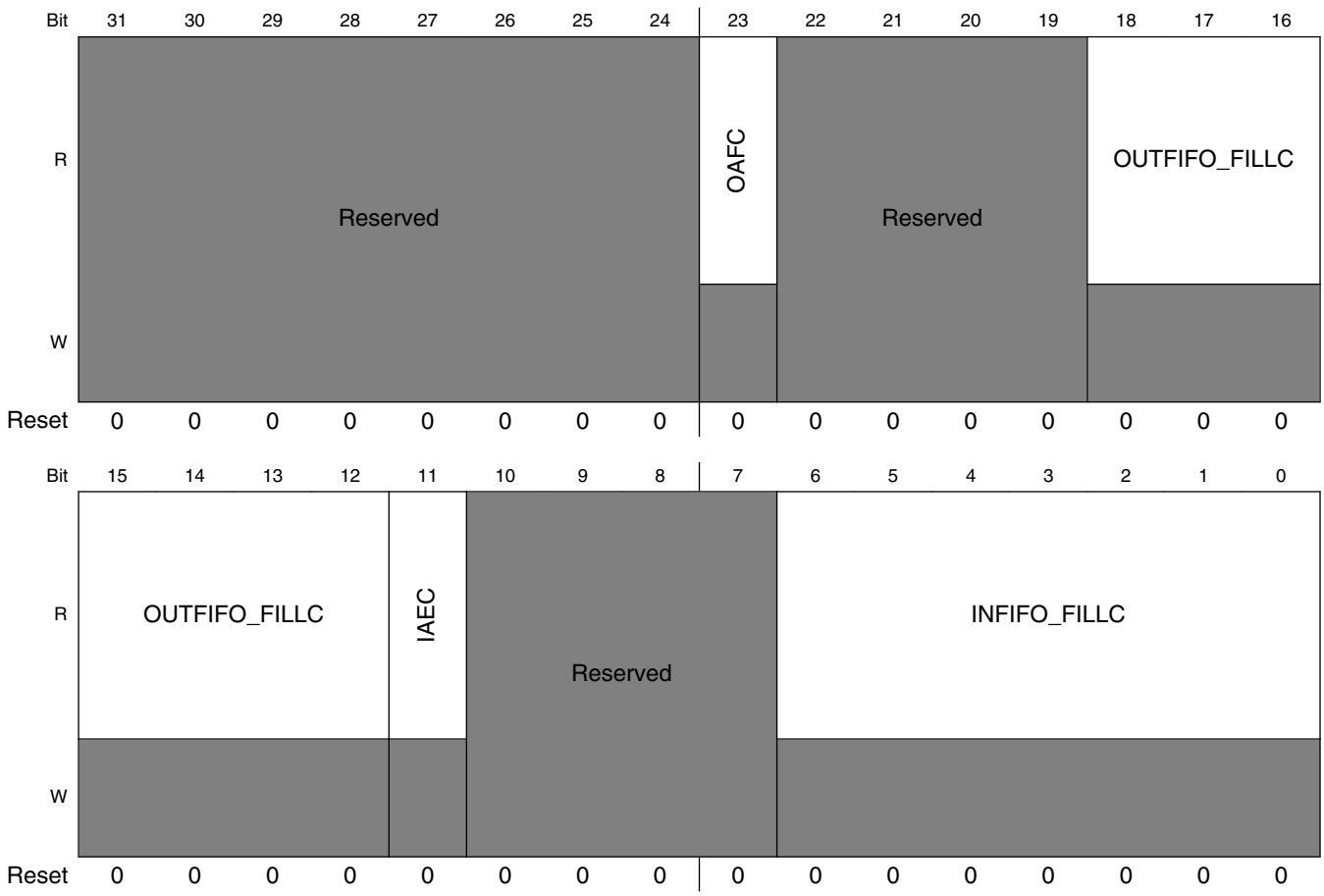
ASRC_ASRMCRC field descriptions (continued)

Field	Description
	This bit will determine whether the FIFO thresholds externally defined in this register is used to control ASRC internal FIFO logic for pair C. 1 Use external defined thresholds. 0 Use default thresholds.
21 BUFSTALLC	Stall Pair C conversion in case of Buffer Near Empty/Full Condition This bit will determine whether the near empty/full FIFO condition will stall the rate conversion for pair C. This option can only work when external ratio is used. Near empty condition is the condition when input FIFO has less than 4 useful samples per channel. Near full condition is the condition when the output FIFO has less than 4 vacant sample words to fill per channel. 1 Stall Pair C conversion in case of near empty/full FIFO conditions. 0 Don't stall Pair C conversion even in case of near empty/full FIFO conditions.
20 BYPASSPOLYC	Bypass Polyphase Filtering for Pair C This bit will determine whether the polyphase filtering part of Pair C conversion will be bypassed. 1 Bypass polyphase filtering. 0 Don't bypass polyphase filtering.
19–18 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
17–12 OUTFIFO_ THRESHOLDC	The threshold for Pair C's output FIFO per channel These bits stand for the threshold for Pair C's output FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of output FIFO fillings of the pair is greater than n samples per channel, the output data ready flag is set; when the number of output FIFO fillings of the pair is less than or equal to n samples per channel, the output data ready flag is automatically cleared.
11 RSYNIFC	Re-sync Input FIFO Channel Counter If bit set, force ASRCCR:ACIC=0. If bit clear, untouch ASRCCR:ACIC.
10 RSYNOFC	Re-sync Output FIFO Channel Counter If bit set, force ASRCCR:ACOC=0. If bit clear, untouch ASRCCR:ACOC.
9–6 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
INFIFO_ THRESHOLDC	The threshold for Pair C's input FIFO per channel These bits stand for the threshold for Pair C's input FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of input FIFO fillings of the pair is less than n samples per channel, the input data needed flag is set; when the number of input FIFO fillings of the pair is greater than or equal to n samples per channel, the input data needed flag is automatically cleared.

15.7.27 ASRC FIFO Status Register for Pair C (ASRC_ASRFSTC)

The register (ASRFSTC) is used to show Pair C internal FIFO conditions.

Address: 203_4000h base + B4h offset = 203_40B4h



ASRC_ASRFSTC field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 OAFC	Output FIFO is near Full for Pair C This bit is to indicate whether the output FIFO of Pair C is near full.
22–19 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
18–12 OUTFIFO_FILLC	The fillings for Pair C's output FIFO per channel These bits stand for the fillings for Pair C's output FIFO per channel. Possible range is [0,64].

Table continues on the next page...

ASRC_ASRFSTC field descriptions (continued)

Field	Description
11 IAEC	Input FIFO is near Empty for Pair C This bit is to indicate whether the input FIFO of Pair C is near empty.
10–7 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
INFIFO_FILLC	The fillings for Pair C's input FIFO per channel These bits stand for the fillings for Pair C's input FIFO per channel. Possible range is [0,64].

15.7.28 ASRC Misc Control Register 1 for Pair X (ASRC_ASRMCR1*n*)

The register (ASRMCR1x) is used to control Pair *x* internal logic (for data alignment etc.).

The bit assignment for all the input data formats is the same as that supported by the .

Address: 203_4000h base + C0h offset + (4d × i), where i=0d to 2d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				IWD			IMSB	Reserved					OMSB	OSGN	OW16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRMCR1*n* field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–12 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
11–9 IWD	Data Width of the input FIFO These three bits will determine the bitwidth for the audio data into ASRC All other settings not shown are reserved. 3'b000 24-bit audio data. 3'b001 16-bit audio data.

Table continues on the next page...

ASRC_ASRMCR1*n* field descriptions (continued)

Field	Description
	3'b010 8-bit audio data.
8 IMSB	Data Alignment of the input FIFO This bit will determine the data alignment of the input FIFO. 1 MSB aligned. 0 LSB aligned.
7–3 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
2 OMSB	Data Alignment of the output FIFO This bit will determine the data alignment of the output FIFO. 1 MSB aligned. 0 LSB aligned.
1 OSGN	Sign Extension Option of the output FIFO This bit will determine the sign extension option of the output FIFO. 1 Sign extension. 0 No sign extension.
0 OW16	Bit Width Option of the output FIFO This bit will determine the bit width option of the output FIFO. 1 16-bit output data 0 24-bit output data.

Chapter 16
Digital Audio Multiplexer (AUDMUX)

16.1 Overview

The Digital Audio Multiplexer (AUDMUX) provides a programmable interconnect device for voice, audio, and synchronous data routing between Synchronous Serial Interface Controller (SSI) and audio/voice codec's (also known as coder-decoders) peripheral serial interfaces.

This section includes a top level diagram that shows the functional organization of the block, including all off-chip signals.

AUDMUX allows the users to reconfigure the audio system signal routing through programming, as opposed to altering the PCB design. The full description of the block is in Functional Description.

Figure 16-1 shows the block diagram.

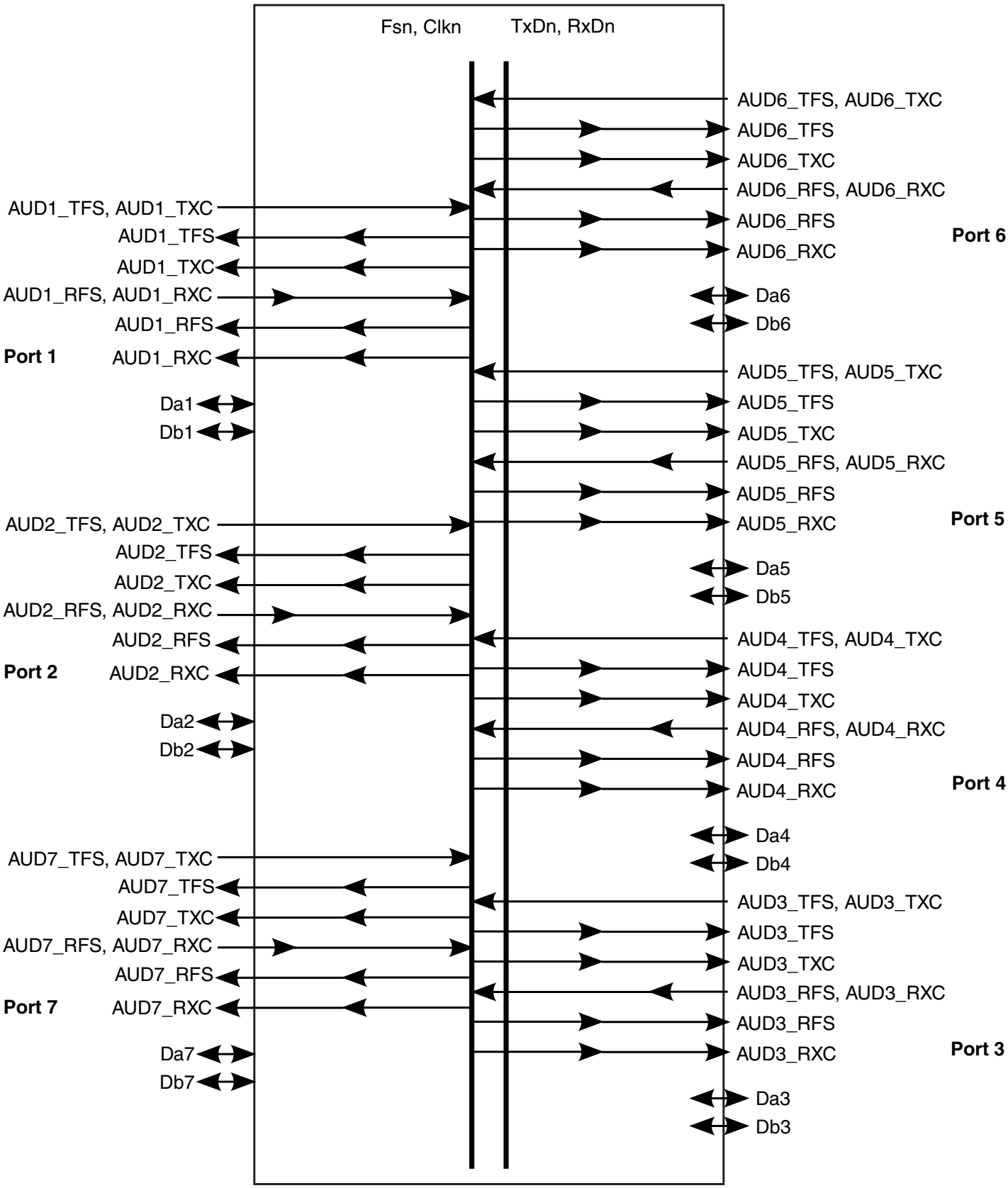


Figure 16-1. AUDMUX Block Diagram

AUDMUX supports single source to single destination connectivity or single source to multiple destination connectivity.

AUDMUX includes two types of interfaces: Internal ports and External ports. Internal ports are hard wired to Synchronous Serial Interface Controller (SSI). The connection between each SSI and AUDMUX's Internal ports cannot be modified, however, routing of the signals connected to each of the internal port can be routed within AUDMUX. External ports are connected to IOMUX module where the ports connect to off-chip audio devices and serial interfaces of other processors. The connectivity of the External port and IOMUX cannot be configured, but the output or input of the signal can be routed easily by setting the appropriate AUDMUX registers.

16.1.1 Features

Key features of the block include:

- Three internal ports
- Four external ports
- Full 6-wire SSI interfaces for asynchronous receive and transmit
- Configurable 4-wire (synchronous) or 6-wire (asynchronous) peripheral interfaces
- Independent Tx/Rx Frame sync and clock direction selection for host or peripheral
- Each host interface's capability to connect to any other host or peripheral interface in a point-to-point or point-to-multipoint (network mode)

16.1.2 Modes and Operations

The AUDMUX supports the modes described in Operating Modes.

16.2 External Signals

The following table describes the external signals of AUDMUX:

Table 16-1. AUDMUX External Signals

Signal	Description	Pad	Mode	Direction
AUD3_RXC (RXC)	Receive clock signal	CSI0_DAT10	ALT1	IO
AUD3_RXD (RXD)	Data receive signal	CSI0_DAT7	ALT4	IO
AUD3_RXFS (RXFS)	Receive Frame sync signal	CSI0_DAT11	ALT1	IO
AUD3_TXC (TXC)	Transmit clock signal	CSI0_DAT4	ALT4	IO

Table continues on the next page...

Table 16-1. AUDMUX External Signals (continued)

Signal	Description	Pad	Mode	Direction
AUD3_TXD (TXD)	Data transmit signal	CSI0_DAT5	ALT4	IO
AUD3_TXFS (TXFS)	Transmit Frame sync signal	CSI0_DAT6	ALT4	IO
AUD4_RXC (RXC)	Receive clock signal	DISP0_DAT19	ALT4	IO
		SD2_CMD	ALT3	
AUD4_RXD (RXD)	Data receive signal	DISP0_DAT23	ALT3	IO
		SD2_DAT0	ALT3	
AUD4_RXFS (RXFS)	Receive Frame sync signal	DISP0_DAT18	ALT4	IO
		SD2_CLK	ALT3	
AUD4_TXC (TXC)	Transmit clock signal	DISP0_DAT20	ALT3	IO
		SD2_DAT3	ALT3	
AUD4_TXD (TXD)	Data transmit signal	DISP0_DAT21	ALT3	IO
		SD2_DAT2	ALT3	
AUD4_TXFS (TXFS)	Transmit Frame sync signal	DISP0_DAT22	ALT3	IO
		SD2_DAT1	ALT3	
AUD5_RXC (RXC)	Receive clock signal	DISP0_DAT14	ALT3	IO
		EIM_D25	ALT6	
AUD5_RXD (RXD)	Data receive signal	DISP0_DAT19	ALT3	IO
		KEY_ROW1	ALT2	
AUD5_RXFS (RXFS)	Receive Frame sync signal	DISP0_DAT13	ALT3	IO
		EIM_D24	ALT6	
AUD5_TXC (TXC)	Transmit clock signal	DISP0_DAT16	ALT3	IO
		KEY_COL0	ALT2	
AUD5_TXD (TXD)	Data transmit signal	DISP0_DAT17	ALT3	IO
		KEY_ROW0	ALT2	
AUD5_TXFS (TXFS)	Transmit Frame sync signal	DISP0_DAT18	ALT3	IO
		KEY_COL1	ALT2	
AUD6_RXC (RXC)	Receive clock signal	DISP0_DAT6	ALT3	IO
AUD6_RXD (RXD)	Data receive signal	DI0_PIN4	ALT2	IO
AUD6_RXFS (RXFS)	Receive Frame sync signal	DISP0_DAT5	ALT3	IO
AUD6_TXC (TXC)	Transmit clock signal	DI0_PIN15	ALT2	IO
AUD6_TXD (TXD)	Data transmit signal	DI0_PIN2	ALT2	IO
AUD6_TXFS (TXFS)	Transmit Frame sync signal	DI0_PIN3	ALT2	IO

16.3 Clocks

This section provides information about AUDMUX clocking including clock inputs and the clock diagram.

The following table describes the clock source for AUDMUX. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 16-2. AUDMUX Clocks

Clock name	Clock Root	Description
ipg_clk_s	ipg_clk_root	Peripheral access clock

16.3.1 Clock Inputs

The IP Bus read/write clock-peripheral clock (ipg_clk_s) is an input to the AUDMUX. It is used for all AUDMUX register accesses. It is driven only when there is an AUDMUX access on the IP Bus.

16.3.2 Clock Diagram

The figure below shows the clocking used in the AUDMUX.

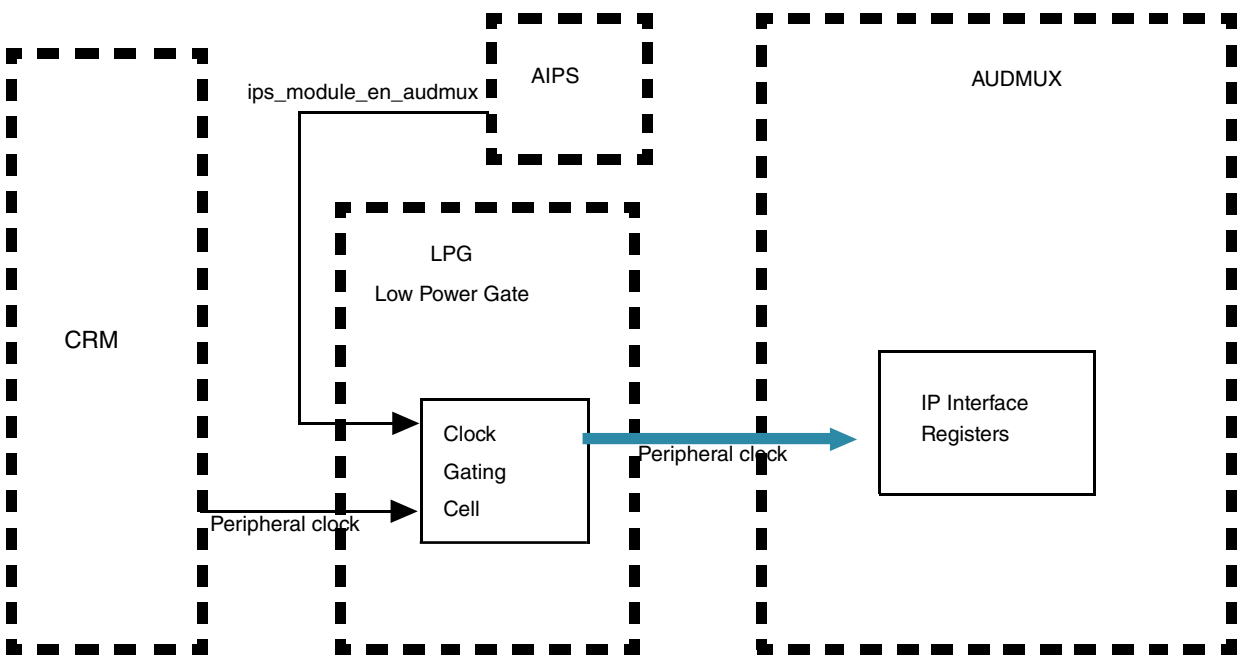


Figure 16-2. AUDMUX Clocking Scheme

16.3.3 Clocking Restrictions

- Since the AUDMUX requires only peripheral clock, it places no restrictions on the bus frequency.
- All registers in the AUDMUX are control registers so their values will not change frequently. These values will be programmed when changing between use cases (not during operation in a particular mode).

16.4 Default Register Configuration

There are two configuration registers for each port. Each pair of configuration registers is identical for each port; however, the default values following a reset differ as shown in the Memory Map.

Default Port Configuration, describes the default configuration of the ports.

16.4.1 Default Port Configuration

After a reset, each port defaults to normal mode (PDCR*n*[MODE] = 0) with synchronous timing mode (PTCR*n*[SYN] = 1) enabled.

The default port-to-port connections are as follows:

- Port 1 to Port 6
  - Port 6 provides the clock and frame sync.
- Port 2 to Port 5
  - Port 5 provides the clock and frame sync.
- Port 3 to Port 4
  - Port 4 provides the clock and frame sync.
- Port 7 to Port 7 (in data loopback mode)
  - Clock and frame syncs are inputs.

16.5 Functional Description

This section provides a complete functional description of the AUDMUX.

16.5.1 Operating Modes

This section describes all functional operation modes of the AUDMUX.

Figure 16-1 shows the AUDMUX block diagram.

All of the ports are essentially identical; there is no functional difference among Ports 1 through 7. The main difference is whether a port is hard wired to synchronous serial interface (SSI) or hard wired to the chip's pads. Each of the connection is hard wired to specific AUDMUX port. AUDMUX provides flexibility in routing the signal within the module, but all Internal and External port connections are fixed to specific configuration.

All ports can be configured as four- or six-wire interfaces. When configured as a six-wire interface, Receive Frame Sync (RXFS) and Receive Clock (RXC) signals of SSI interface enable the serial interface to be used in asynchronous mode with separate receive and transmit clocks.

AUDMUX supports both Normal mode (not to be confused with SSI’s Normal Mode), External Network mode and Internal Network mode. The definition of each mode will be given in the next section.

All ports have a TXRXEN bit to provide flexibility in supporting network mode configurations. The TXRXEN bit reverses the functions of transmit and receive data lines where the transmit line is configured as receive and transmit line to be configured as transmit line. This function is provided so that mastership of the serial bus can be passed among multiple external devices connected to a single port.

In addition to supporting the External Network mode (default), all ports support an internal network mode:

- With internal network mode, single point-to-multipoint network configuration with an arbitrary number of slaves can be supported if the external slaves are put into the high-impedance state (as defined in the SSI network mode protocol) and have pull-up resistors on their TxD pins. (Alternatively, this can be viewed as requiring a pull-up resistor on the corresponding AUDMUX RxD pin.)

Bit clock direction selection enables each port to be configured as a master or slave in the flow.

Possible scenarios include:

- SSI (hard wired to internal port) transmits data to a voice codec and a BT (Blue tooth) codec (both on external Port 4Port 5) and the Bottom Connector (on external Port 5Port 6) simultaneously using network mode. SSI is configured as the master.
- An external processor (external port - Port 4Port 3) drives a voice codec and a BT codec (both on external Port 5Port 4) and the Bottom Connector (on Port 6Port 5) simultaneously using network mode. The external processor is the master.

16.5.1.1 Port Receive Data Modes

Each port has logic to select which data lines are used to create the RXD line for the corresponding host interface.

Figure 16-3 shows the logic used to create the RXD line for Port 1. This logic has the following modes of operation (as determined by MODE:

- Normal (not to be confused with SSI's normal mode)
- Internal network mode

The subsequent sections describe the various modes of the port receive data logic. The following terms are used to define the operation of the AUDMUX:

- Network mode- Time-Division Multiplexed protocol for sending unique data to multiple devices on a serial bus or single devices with multi-channel capabilities.
- Internal network mode-Physical bus configuration where multiple serial buses are effectively connected within the AUDMUX via digital logic to create point-to multipoint connectivity. An arbitrary number of devices are supported. Devices must be put into the high-impedance state as specified by the network mode protocol.
- External network mode-Physical bus configuration where multiple serial buses are electrically connected together on a printed circuit board (that is, external to the AUDMUX). Devices must put their TXD lines into the high-impedance state as specified by the network mode protocol. TXD lines of devices must be pulled high.

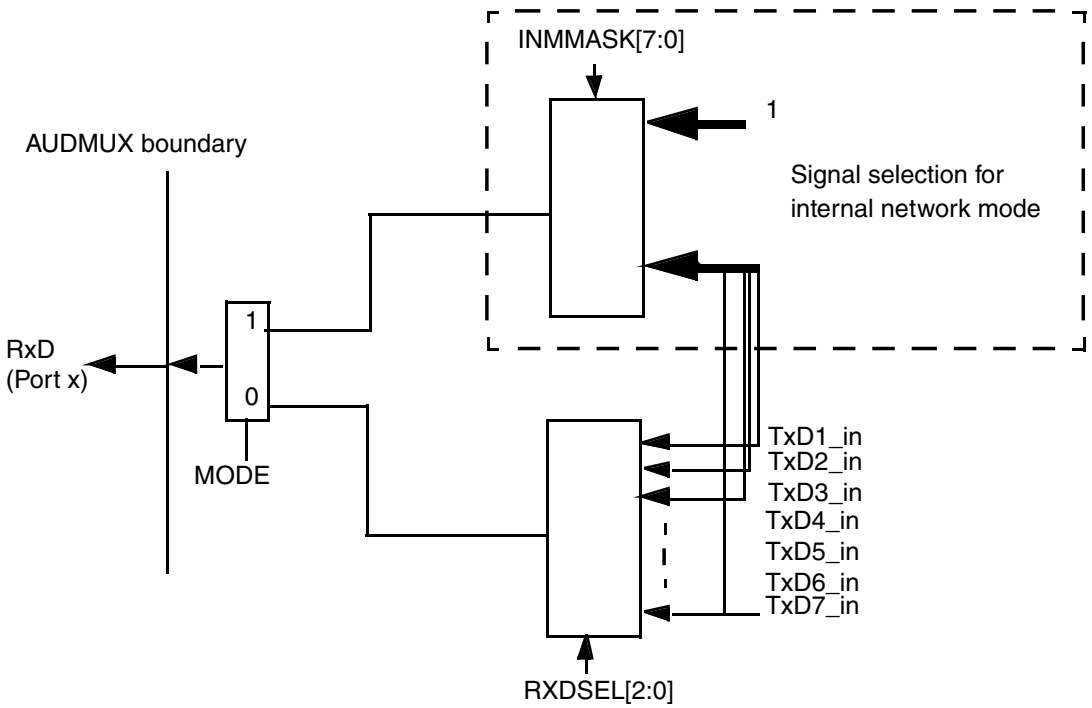


Figure 16-3. Receive Data Logic for Port x

16.5.1.1.1 Normal Mode

In normal mode (MODE = 0) , not to be confused with SSI's Normal Mode, the port is connected in a single device point-to-single device port configuration (as a master or a slave) and the RXDSEL[2:0] setting selects the transmit signal from any port. In normal mode, any data format can be used (that is, SSI normal mode, SSI network mode, AC-97, and others.

If a user wishes to transmit SSI1's data (TXD) to port 5, PDCR5's RXDSEL[2:0] must be set to 000b. If the clock (TCK) and frame clock (TFS) are sent out to the external device from SSI1, PTCR5's TFSEL[3:0] and TCSEL[3:0] must be set to 0000b (port1) and 0000b (port1), while setting PTCR5's TFSDIR and TCLKDIR to 1.

If either or both of the clocks are to be received from external device to SSI1, PTCR5's TCLKDIR and/or TFSDIR must be set to 0 (input), and PTCR1's TFSEL[3:0] and/or TCLKDIR[3:0] to 0100b (port5).

Likewise, if a user wishes to receive serial data from Port 4 and send the receiving data to SSI2's RXD, one must set PDCR2's RXDSEL[2:0] to 11b. If the frame (RXFS) and bit clocks (RXC) are to be received from the external device, PTCR2's RFSEL[3:0] and RCSEL[3:0] must be set to 011b while PTCR4's RFSDIR and RCLKDIR set to 0.

16.5.1.1.2 Internal Network Mode

In internal network mode (MODE = 1) , the output of the AND gate is routed (via the output of the port) to the RXD signal of the corresponding host interface.

The INMMASK bit vector selects the transmit signals of the ports that are to be connected in network mode. The transmit signals received at the AUDMUX ports (TxDn_in) are ANDed together to form the output. In internal network mode, only one device can be transmitting in its predesignated timeslot and all other transmit signals must remain high (be in high-impedance state and pulled-up). Therefore, non-active signals in the selection will be high and do not influence the output of the AND gate.

Network mode is a protocol where a master SSI is connected to more than one slave SSI device and communication occurs on a time-slotted frame. Though network mode can allow master-slave and slave-slave communication, internal network mode supports only master-slave communication.

There are two scenarios where internal network mode can be used with external network mode:

1. Slave-only devices are attached to an external port.
2. A master device is attached to an external port and all slave devices connected to the same external port are disabled.

NOTE

When internal network mode is enabled at an external port, RXDSEL[3:0] for RxDn_obe selection is ignored and RxD_obe is always driven high (that is, asserted for all timeslots). All slave devices connected to the same port must be disabled.

Internal Network Mode Example 1

SSI_m and SSI_n are used with Port 3 in internal network mode as shown in Figure 16-4. No pull-up resistors are required because the interfaces combined in internal network mode are on-chip interfaces.

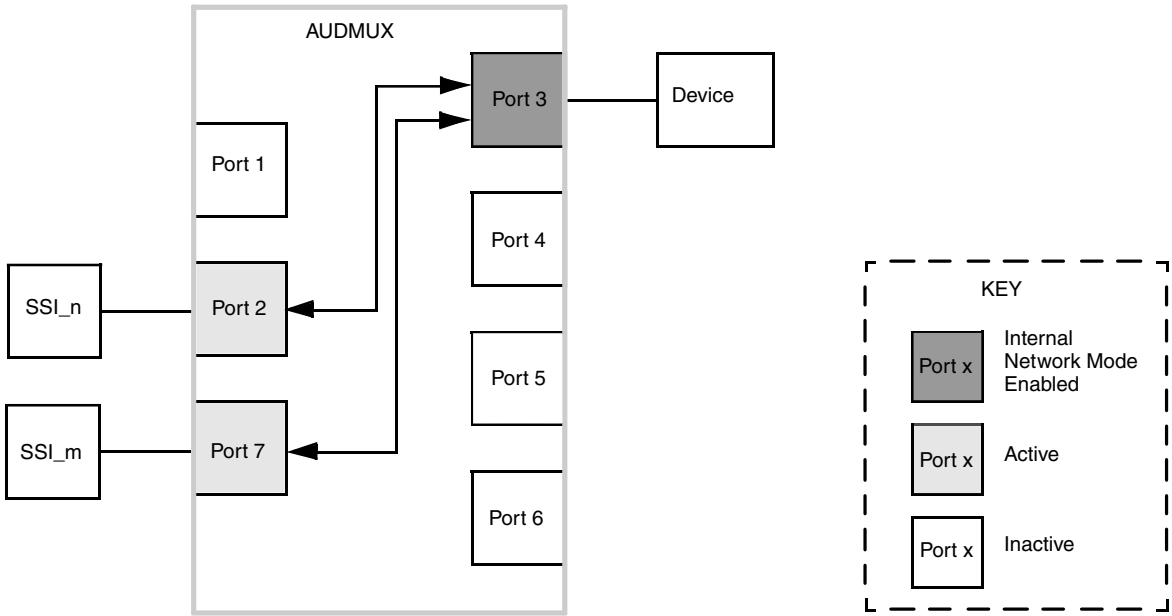


Figure 16-4. Block Diagram For Example 1

See Figure 16-5 for the timing diagram of Example 1. The clock and frame sync signals show the bit and frame timing for the serial bus. The vertical dashed lines divide the frame into four timeslots.

The data lines for SSI_m and SSI_n (as well as their output enables) are shown. Note that the on-chip interfaces drive a logic '1' when their output enables are logic '0'. The combined TXD line, which is the logical AND of the individual TXD lines, is used for Port 3's TXD line.

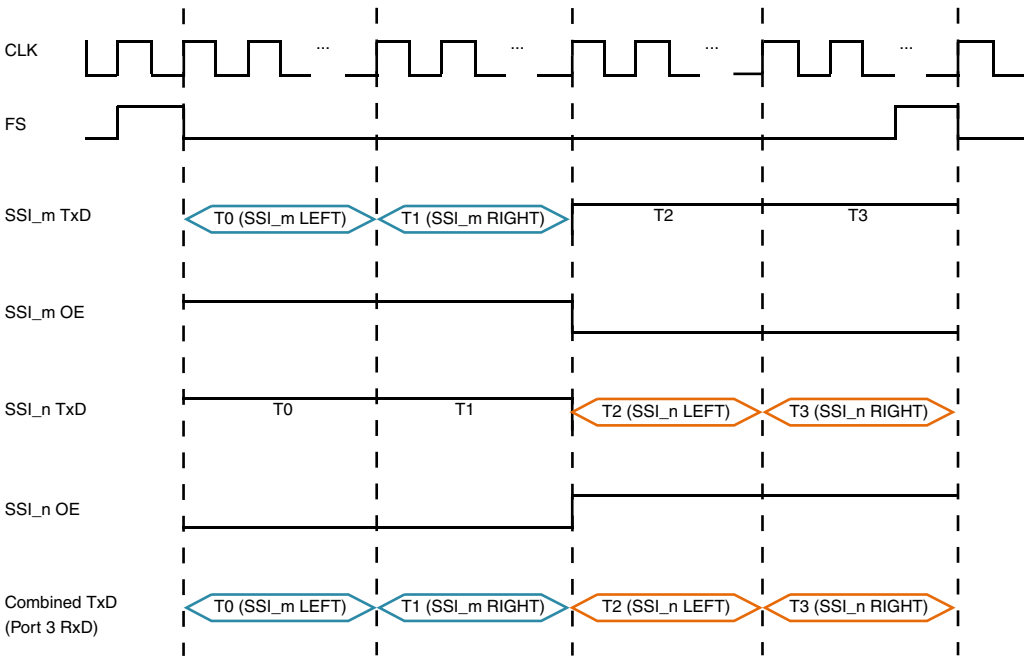


Figure 16-5. Example Using Internal Ports For Transmit Data

Internal Network Mode Example 2

The SSI, Port 3, and Port 4 are used with Port 5 in internal network mode, as shown in the following figure. Note that Port 3 and Port 4 are external ports. Therefore, pull-up resistors are required on the Port 3 RXD and Port 4 RXD pins. This example shows the timing associated with using adjacent timeslots for the SSI, Port 3 and Port 4 .

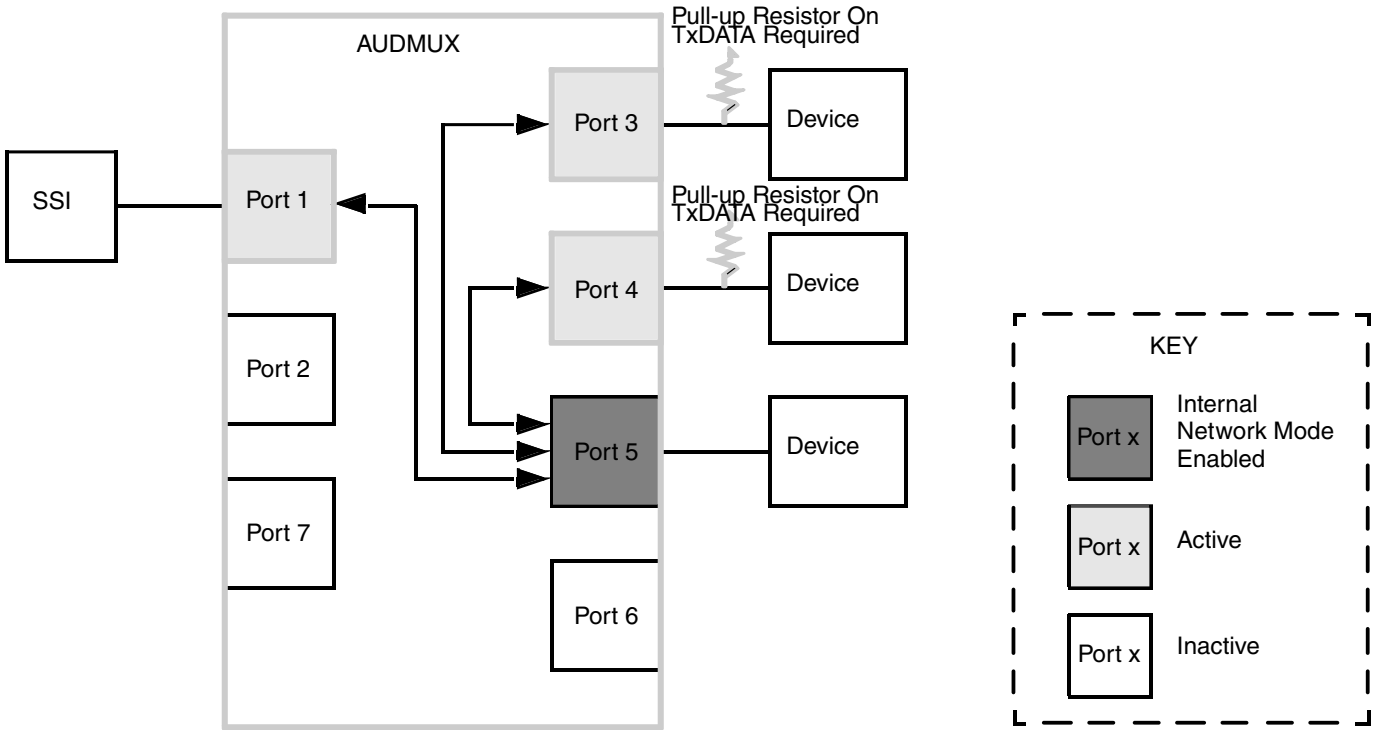


Figure 16-6. Block Diagram For Example 2

The resistance value of the pull-up resistors must be sufficiently high such that a value of '0' can be pulled up to logic '1' within half of a period of the bitclock. The required resistance must be no larger than:

Rmax = 1 / (2 * f_{bc} * C) where:

- f_{bc} is the frequency of the bitclock
- C is the total system capacitance (ICs, board traces, and so on)

The following figure shows the timing diagram for this example. The clock and frame sync signals show the bit and frame timing for the serial bus. The vertical dashed lines divide the frame into four timeslots.

The data lines for the SSI, Port 3 and Port 4 are shown. Note that the SSI transmits a logic '1' when its corresponding output enable is a logic '0'. The data lines from Port 3 and Port 4 at the pad are pulled high by pull-up resistors when they are in the high-impedance state. The data lines from Port 3 and Port 4 at the AUDMUX are pure digital signals and are constantly driven. The combined TXD line, which is the logical AND of the SSI, Port 3 and Port 4's TXD lines, is used for Port 5's TXD line.

Note the highlighted areas in the Figure 16-7. This shows the transition time that occurs while a TXD line is being pulled high. In this example, this transition time is a maximum of 1/2 the period of the serial bitclock. This prevents corruption of the first data bit of the next timeslot. It is critical that the pull-up resistance is sufficient for the given bitclock frequency and system capacitance.

Note that hysteresis should be enabled at Port 3's RXD pad and Port 4's RXD pad to prevent the digital signals created by the pad from toggling rapidly during the pull-up period. The pads typically require a transition within 25ns unless hysteresis is enabled. Instead of using hysteresis, one could select a pull-up resistor sufficiently high to pull-up the signal at the pad within 25 ns; however, that would result in a higher resistance value and higher current drain.

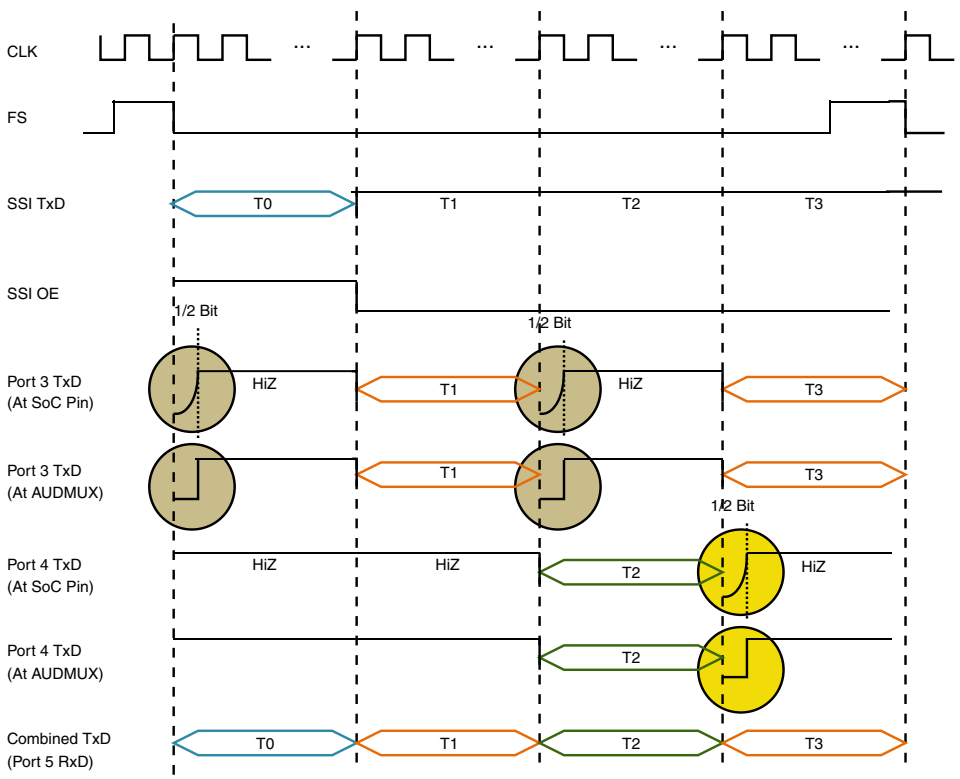


Figure 16-7. Example Using External Ports for Transmit Data in Consecutive Timeslots

Internal Network Mode Example 3

The SSI and Port 3 are used with Port 5 in internal network mode as shown in the following figure. Note that Port 3 is an external port. Therefore, a pull-up resistor is required on the Port 3TXD pin. This example shows the timing associated with inserting empty timeslots after the timeslots have been used by external ports.

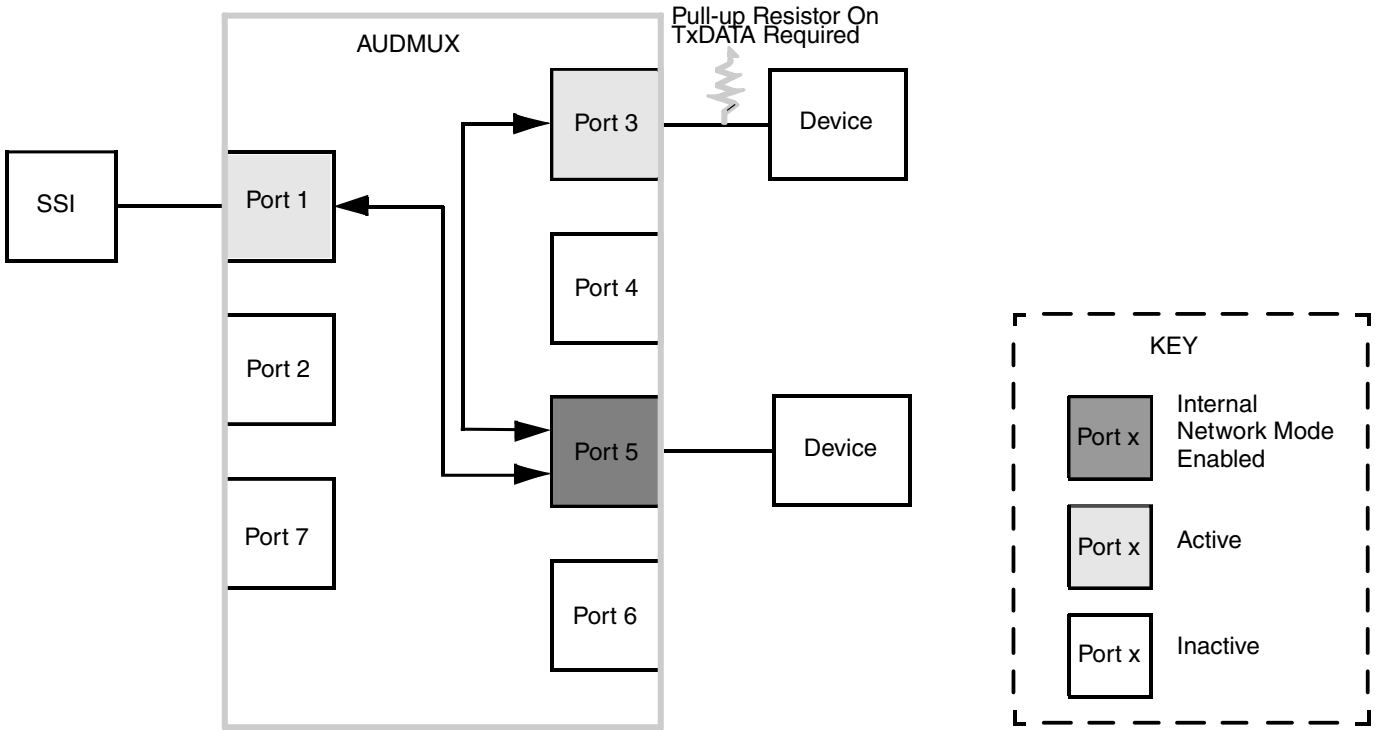


Figure 16-8. Block Diagram For Example 3

The resistance value of the pull-up resistors must be sufficiently high such that a value of '0' can be pulled up to logic '1' by the time that the next occupied timeslot occurs. This allows a much weaker pull-up to be used as compared to Example 2. The required resistance must be no larger than:

Rmax = (4 * n + 1) / (2* f_{bc} * C) where:

- n is the number of bits per timeslot
- f_{bc} is the frequency of the bitclock
- C is the total system capacitance (ICs, board traces, and so on)

The figure below shows the timing diagram for this example. The clock and frame sync signals show the bit and frame timing for the serial bus. The vertical dashed lines divide the frame into four timeslots.

The data lines for the SSI and Port 3 are shown. Note that the SSI transmits a logic '1' when its corresponding output enable is a logic '0'. The data line from Port 3 at the pad is pulled high by a pull-up resistor when they are in the high-impedance state. The data line from Port 3 at the AUDMUX is a pure digital signal and is constantly driven. The combined TXD line, which is the logical AND of the SSI and Port 3's TXD lines, is used for Port 5's RXD line.

Note the highlighted area in the Figure 16-9. This shows the transition time that occurs while Port 3's TXD line is being pulled high. In this example, this transition time is a maximum of two timeslots plus 1/2 the period of the serial bitclock. This prevents corruption of the first data bit of the next timeslot. It is critical that the pull-up resistance is sufficient for the given bitclock frequency and system capacitance.

Note that hysteresis must be enabled at Port 3's RXD pad to prevent the digital signal created by the pad from toggling rapidly during the extended pull-up period. The pads typically require a transition within 25 ns unless hysteresis is enabled.

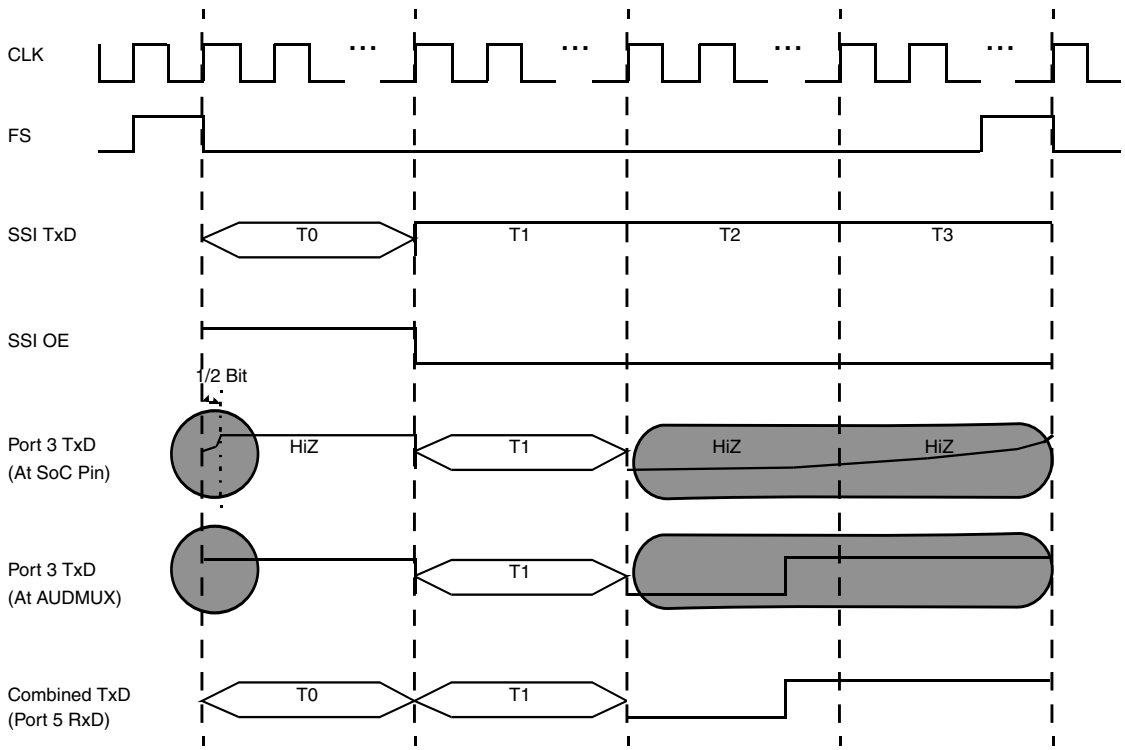


Figure 16-9. Example Using External Ports For Transmit Data In Nonconsecutive Timeslots

16.5.1.1.3 Transmit Data Output Enable Assertion

The TXD line from the internal network mode master (connected at any internal port) is put into the high-impedance state at the pad depending upon the assertion or deassertion of TxD_obe, its corresponding output enable generated by the network mode master.

In the case of an external network mode master (connected at an external port), the corresponding TxD_obe is always asserted after the port data register configuration.

16.5.1.2 Tx/Rx Switch and External Network Mode

External network mode is the traditional network mode connection. It is called external network mode to differentiate from the internal network mode. In external network mode, devices are connected to a single external port in a star or multi-drop configuration.

In network mode, there can be only one master (driving the frame sync and clock source) with the other devices configured in normal slave mode or network slave mode. Unlike internal network mode, both master-slave and slave-slave communication can take place in external network mode. Codec devices transmit on a single timeslot while processor serial interfaces (that is, SSI) can process more than one timeslot of data while in network master or slave mode.

The following figure shows the Tx/Rx data switch. RxD_obe is the output buffer enable signal and RxD_out is the data transmit signal from the serial interface. The TxD_in signal is the receive data signal going towards the RXDSEL muxes of all ports.

D_TxRx is the data pin which serves as the chip-level transmit data pin when the TxRx switch is not enabled. D_RxTx is the data pin which serves as the chip-level receive data pin when the TxRx switch is not enabled. The roles of these pins are reversed when the TxRx switch is enabled.

When TXRXEN is disabled (TXRXEN=0), RxD_out is routed to D_TxRx and D_RxTx is routed to TxD_in. The output buffer enable, selected by RXDSEL[2:0], is routed to Db_obe.

When the Tx/Rx switch is enabled (TXRXEN=1), RxD_out is routed to D_RxTx and D_TxRx is routed to TxD_in. The output buffer enable, selected by RXDSEL[2:0], is routed to Da_obe.

If the RXDSELn[2:0] field for any Port *n* is configured to select data from an internal port, the output buffer enable is selected by RXDSELn[2:0] and is routed to Da*n*_obe/ Db*n*_obe. In the case when the RXDSELn[2:0] field for Port *n* is configured to select data from an external port, the output buffer enable is always high and routed to Dan_obe/Dbn_obe, depending on the TXRXENn switch configuration.

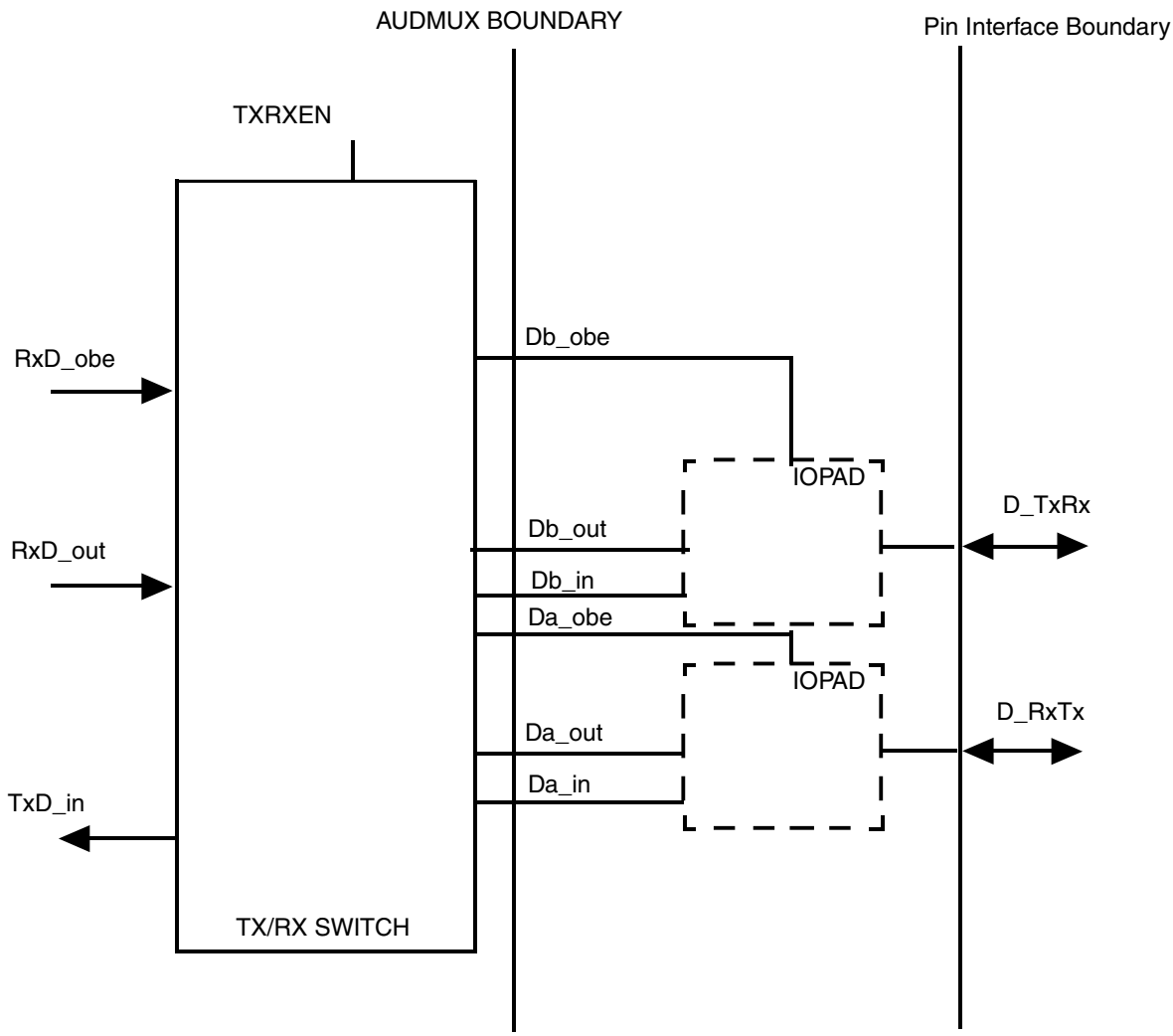


Figure 16-10. Tx/Rx Switch

16.5.1.3 Timing Modes

The AUDMUX ports are constructed as 6-wire interfaces. However, they can be used either in synchronous or asynchronous modes as determined by the SYN bit.

16.5.1.3.1 Synchronous Mode (4-Wire Interface)

In Synchronous mode, each port set in SYN=1 will be a 4-wire interface (that is, RXD, TXD, TXC, TXFS). In this setup, receive data timing is determined by TXC and TXFS..

As shown in the following figure, SSI signals interfaced to Port x signals are routed to Port y. When SYN = 1 the 6-wire signal from SSI is reduced to 4-wire within the internal logic.

TFS_in, RFS_in, TCLK_in, and RCLK_in are the input frame sync and bit clocks from the serial interface (Port x) with their corresponding output buffer enable signals (_obe). TFS_out, RFS_out, TCLK_out, and RCLK_out are the frame sync and bit clocks that are transmitted to the serial interface from the other ports.

The TFS_out and TCLK_out are selected at Port x by the TFSEL and TCSEL mux settings, respectively. RFS_out and RCLK_out are selected at Port x by the RFSEL and RCSEL mux settings, respectively. Similarly, in the external direction, Port y is configured as a 4-wire port; TFSEL selects the FS_obe and FS_out signals. In this mode, the configuration of RFSEL and RCSEL is not used, since the RFS_out and RCLK_out pins at Port y are not available.

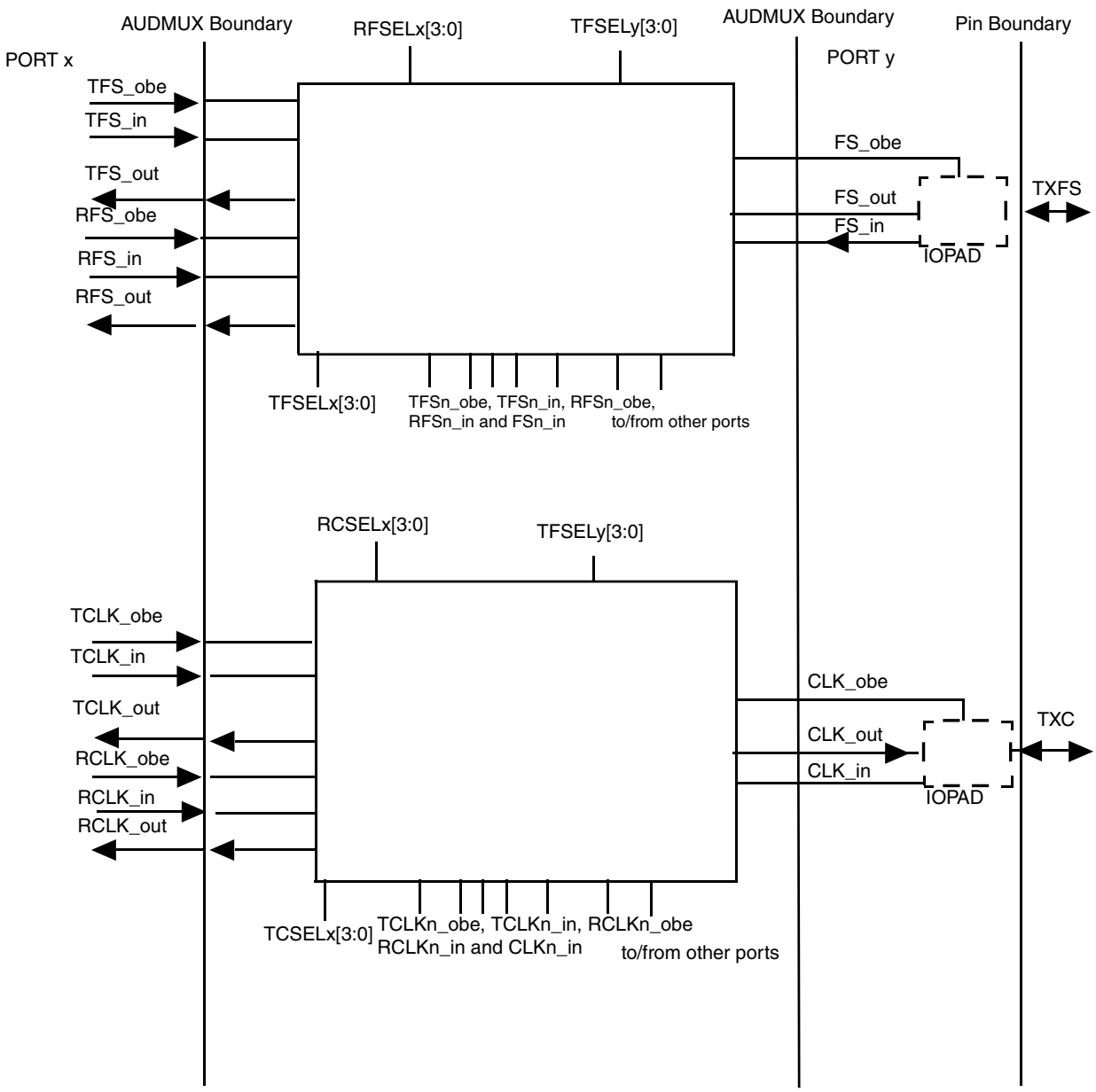


Figure 16-11. Frame Sync and Clock Routing When External Port Is 4-Wire

16.5.1.3.2 Asynchronous Mode (6-Wire Interface)

In Asynchronous mode, the port has a 6-wire interface (meaning RXD, TXD, TXC, TXFS, RXC, RXFS). This mode has additional receive clock (RXC) and frame sync (RXFS) signals as compared to the synchronous or 4-wire interface.

As shown in the figures below, Port x signals can be routed to Port y, producing 6-wire to 6-wire port connectivity.

TFS_in, RFS_in, TCLK_in, and RCLK_in are input frame sync and bit clocks from the serial interface (Port x) with their corresponding output buffer enable signals (_obe). TFS_out, RFS_out, TCLK_out, and RCLK_out are the frame sync and bit clocks that are transmitted to the serial interface from the other ports.

TFS_out and TCLK_out are selected by the TFSEL and TCSEL mux settings, respectively. RFS_out and RCLK_out are selected by the RFSEL and RCSEL mux settings, respectively. Similarly, in the external direction, the TFSEL selects the TxFS_obe and TxFS_out signals and TCSEL selects the TxCLK_obe and TxClk_out signals. The RFSEL selects the RxFS_obe and RxFS_out signals and RCSEL selects the RxCLK_obe and RxCLK_out signals.

NOTE

Because FS_in and CLK_in from external interfaces are also routed to the TFSEL and TCSEL muxes of the external ports, these signals do not have corresponding buffer enable signals. Consequently, their corresponding inputs to the TFSEL and TCSEL mux of the external ports have to be tied high.

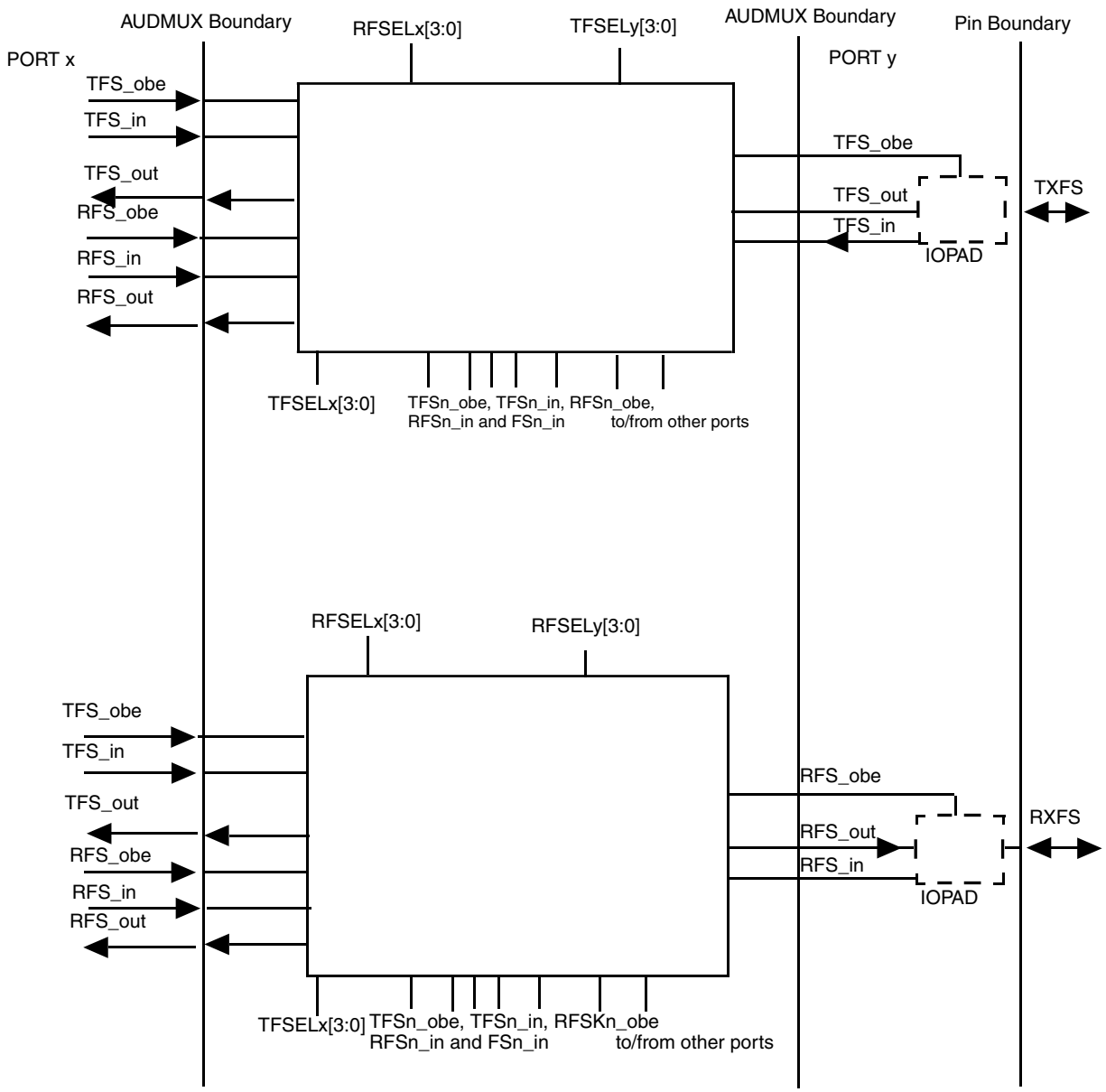


Figure 16-12. Frame Sync Routing When External Port Is 6-Wire

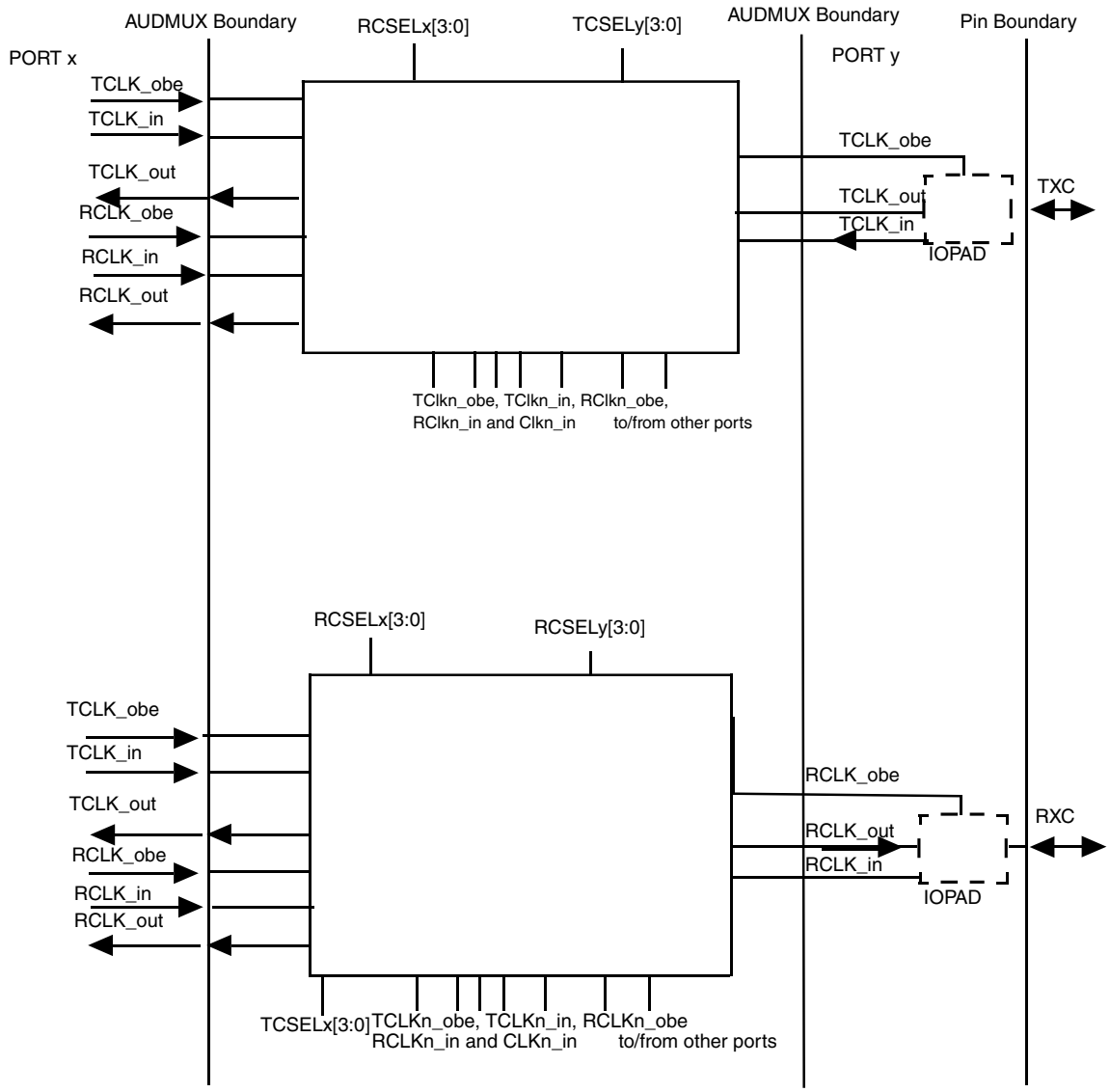


Figure 16-13. Clock Routing When External Port Is 6 Wire

16.5.2 Connectivity Between Ports

Four basic types of connections are provided by the AUDMUX:

- Internal port to external port
- External port to external port
- Internal port to internal port
- Loopback

The corresponding data connections are described in the following sections.

16.5.2.1 Internal Port to External Port Connectivity

The internal port is connected to a processor's serial interface. TxD_obe is the buffer enable signal from the serial interface, TxD_in is the input transmit data from the serial interface to the AUDMUX, and RxD_out is the receive data output from the AUDMUX to the serial interface.

RXDSEL[2:0] of the external port selects the buffer enable signal (TxD_obe) and transmit data output (TxD_out) signal from the TxD_obe and RxD_in signals. RXDSEL[2:0] is a common signal to both selection muxes.

NOTE

Because buffer TxD_in signals from external interfaces do not have corresponding buffer enable signals, their buffer enable signals into the selection mux are tied high. This will ensure that selection of TxD_in, as RxD_out will also drive the RxD_obe output high.

Transmit Data from the serial interface goes into the RXDSEL data mux and comes out as RxD_out. RxD_out is routed to Da_TxRx when TXRXEN is disabled and to D_RxTx when TXRXEN is enabled. Similarly, D_RxTx is routed to TxD_in when TXRXEN is disabled and D_TxRx is routed to TxD_in when TXRXEN is enabled. The routing of frame syncs is shown in Figure 16-12 and the routing of interface clocks is shown in Figure 16-10.

If internal network mode is disabled, then RXDSEL selects the TxD_in, which is sent from the AUDMUX to the serial interface connected at Port x. When the internal network mode is selected, RxD_out is constructed by ANDing selected TxD_in signals from the ports (as determined by INMMASK).

If there is more than one device attached to the external port at D_TxRx and D_RxTx and one of the devices is a network master, then two conditions must be noted:

1. When the external master is enabled in network mode, then the serial interface at Port x must be configured as a slave (normal or network mode). No Tx/Rx switching is required.
2. When the external master is disabled and the serial interface at Port x and other slave devices must communicate, then the serial interface at Port x must be configured as a network mode master and the Tx/Rx switch at Port y must be enabled (TXRXEN=1). This will ensure that the transmit and receive paths are connected appropriately.

To communicate with more than one port, internal network mode can be enabled at Port x. In internal network mode, it is possible to communicate with any device attached to the other ports. Internal network mode shall be enabled at the port that is the SSI network mode master.

16.5.2.2 External Port to External Port Connectivity

External ports can communicate with external ports directly.

External ports can communicate together in three ways:

1. Each port's receive logic is configured in normal mode (MODE = 0) . Each port's RXDSEL[2:0] field is configured to select the other port's transmit data. Bit fields associated with clock/frame sync selection and direction are configured for each port. Either port can be the master.
2. One port is configured in internal network mode (MODE = 1) . All desired data lines are combined by the AND gate as determined by INMMASK[7:0]. Since an external port is being used as the internal network mode master, all other devices on the same AUDMUX port as the internal network mode master must be disabled. This configuration can be used with a combination of internal and external ports. All external ports must have a pull-up resistor on its RXD pin. Bit fields associated with clock/frame sync selection and direction are configured for each port. Any port can be the master.

16.5.2.3 Internal Port to Internal Port Connectivity

Internal ports can communicate with other internal ports directly, thereby providing a means for synchronous interprocessor communication.

Internal ports can communicate together in two ways:

1. Each port's receive logic is configured in normal mode (MODE = 0) . Each port's RXDSEL[2:0] field is configured to select the other port's transmit data. Bit fields associated with clock/frame sync selection and direction are configured for each port. Either port can be the master.
2. One port is configured in internal network mode (MODE = 1) . All desired data lines are combined by the AND gate as determined by INMMASK[7:0]. This configuration can be used with a combination of internal and external ports. All external ports must have a pull-up resistor on its RXD pin. Bit fields associated with clock/frame sync selection and direction are configured for each port. Any port can be the master.

16.5.2.4 Loopback Connectivity

AUDMUX ports can communicate with themselves in order to provide loopback functionality. Port x can route its TXD signal to its own RxD_out signal by setting RXDSELx[2:0] to its own port number. This is supported by all ports in the AUDMUX.

In addition, ports can provide loopback support in internal network mode. With internal network mode, the internal network mode master can loop its TXD signal (combined with those of other ports, if desired) back into its RxD_out signal. Port x's INMMASK should be set such that bit (x - 1) is clear in order to enable the loopback.

16.6 AUDMUX Memory Map/Register Definition

This section includes the block memory map and detailed descriptions of all registers. For the base address of a specific sub-block instantiation, see the system memory map in this manual.

The AUDMUX memory map is shown in the following table.

AUDMUX memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21D_8000	Port Timing Control Register 1 (AUDMUX_PTCR1)	32	R/W	AD40_0800h	16.6.1/693
21D_8004	Port Data Control Register 1 (AUDMUX_PDCR1)	32	R/W	0000_A000h	16.6.2/695
21D_8008	Port Timing Control Register 2 (AUDMUX_PTCR2)	32	R/W	A500_0800h	16.6.3/696
21D_800C	Port Data Control Register 2 (AUDMUX_PDCR2)	32	R/W	0000_8000h	16.6.4/698
21D_8010	Port Timing Control Register 3 (AUDMUX_PTCR3)	32	R/W	9CC0_0800h	16.6.5/699
21D_8014	Port Data Control Register 3 (AUDMUX_PDCR3)	32	R/W	0000_6000h	16.6.6/701
21D_8018	Port Timing Control Register 4 (AUDMUX_PTCR4)	32	R/W	0000_0800h	16.6.7/702
21D_801C	Port Data Control Register 4 (AUDMUX_PDCR4)	32	R/W	0000_4000h	16.6.8/704
21D_8020	Port Timing Control Register 5 (AUDMUX_PTCR5)	32	R/W	0000_0800h	16.6.9/705
21D_8024	Port Data Control Register 5 (AUDMUX_PDCR5)	32	R/W	0000_2000h	16.6.10/ 707
21D_8028	Port Timing Control Register 6 (AUDMUX_PTCR6)	32	R/W	0000_0800h	16.6.11/ 708
21D_802C	Port Data Control Register 6 (AUDMUX_PDCR6)	32	R/W	0000_0000h	16.6.12/ 710
21D_8030	Port Timing Control Register 7 (AUDMUX_PTCR7)	32	R/W	0000_0800h	16.6.13/ 711

Table continues on the next page...

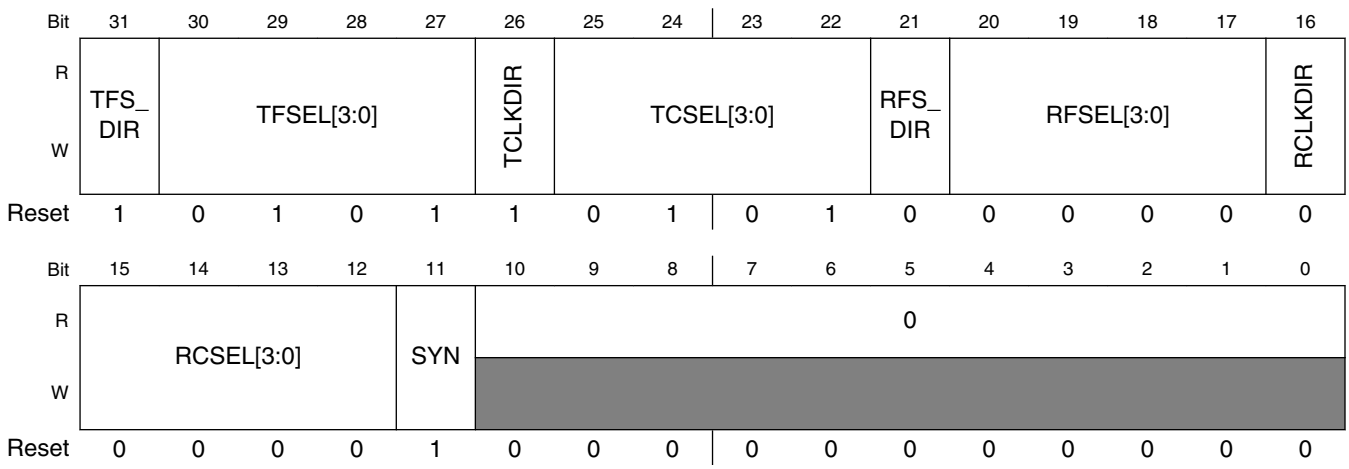
AUDMUX memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21D_8034	Port Data Control Register 7 (AUDMUX_PDCR7)	32	R/W	0000_C000h	16.6.14/ 713

16.6.1 Port Timing Control Register 1 (AUDMUX_PTCR1)

PTCR1 is the Port Timing Control Register for Port 1.

Address: 21D_8000h base + 0h offset = 21D_8000h



AUDMUX_PTCR1 field descriptions

Field	Description
31 TFS_DIR	Transmit Frame Sync Direction Control. This bit sets the direction of the TXFS pin of the interface as an output or input. When set as an input, the TFSEL settings are ignored. When set as an output, the TFSEL settings determine the source port of the frame sync. 0 TXFS is an input. 1 TXFS is an output.
30–27 TFSEL[3:0]	Transmit Frame Sync Select. Selects the source port from which TXFS is sourced. 0xxx Selects TXFS from port. 1xxx Selects RXFS from port. x000 Port 1 x110 Port 7 x111 Reserved
26 TCLKDIR	Transmit Clock Direction Control. This bit sets the direction of the TXC pin of the interface as an output or input. When set as an input, the TCSEL settings are ignored. When set as an output, the TCSEL settings determine the source port of the clock.

Table continues on the next page...

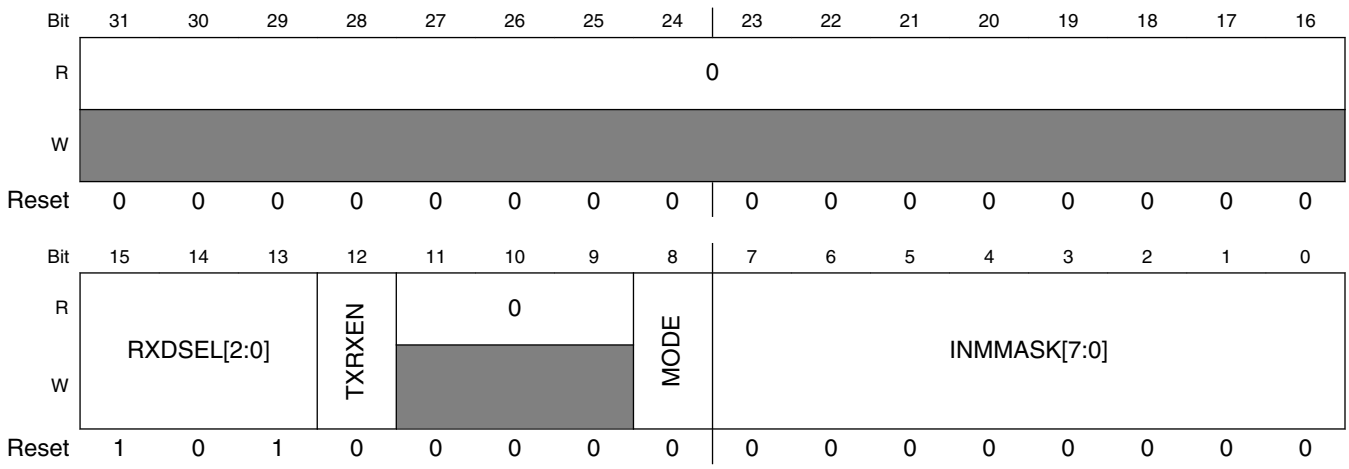
AUDMUX_PTCR1 field descriptions (continued)

Field	Description
	0 TXC is an input. 1 TXC is an output.
25–22 TCSEL[3:0]	Transmit Clock Select. Selects the source port from which TXC is sourced. 0xxx Selects TXC from port. 1xxx Selects RXC from port. x000 Port 1 x110 Port 7 x111 Reserved
21 RFS_DIR	Receive Frame Sync Direction Control. This bit sets the direction of the RXFS pin of the interface as an output or input. When set as an input, the RFSEL settings are ignored. When set as an output, the RFSEL settings determine the source port of the frame sync. 0 RXFS is an input. 1 RXFS is an output.
20–17 RFSEL[3:0]	Receive Frame Sync Select. Selects the source port from which RXFS is sourced. RXFS can be sourced from TXFS and RXFS from other ports. 0xxx Selects TXFS from port. 1xxx Selects RXFS from port. x000 Port 1 x110 Port 7 x111 Reserved
16 RCLKDIR	Receive Clock Direction Control. This bit sets the direction of the RXC pin of the interface as an output or input. When set as an input, the RCSEL settings are ignored. When set as an output, the RCSEL settings determine the source port of the clock. **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 RXC is an input. 1 RXC is an output.
15–12 RCSEL[3:0]	Receive Clock Select. Selects the source port from which RXC is sourced. RXC can be sourced from TXC and RXC from other ports. 0xxx Selects TXC from port. 1xxx Selects RXC from port. x000 Port 1 x110 Port 7 x111 Reserved
11 SYN	Synchronous/Asynchronous Select. When SYN is set, synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals (that is, the port is a 4-wire interface). When SYN is cleared, asynchronous mode is chosen and separate clock and frame sync signals are used for the transmit and receive sections (that is, the port is a 6-wire interface). **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 Asynchronous mode 1 Synchronous mode (default)
Reserved	This read-only field is reserved and always has the value 0.

16.6.2 Port Data Control Register 1 (AUDMUX_PDCR1)

PDCR1 is the Port Data Control Register for Port 1.

Address: 21D_8000h base + 4h offset = 21D_8004h



AUDMUX_PDCR1 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–13 RXDSEL[2:0]	Receive Data Select. Selects the source port for the RXD data. RXDSEL is ignored if MODE = 1 (that is, Internal Network Mode is enabled). xxx Port number for RXD 000 Port 1 110 Port 7 111 Reserved
12 TXRXEN	Transmit/Receive Switch Enable. Swaps the transmit and receive signals. 0 No switch (Transmit Pin = Transmit, Receive Pin = Receive) 1 Switch (Transmit Pin = Receive, Receive Pin = Transmit)
11–9 Reserved	This read-only field is reserved and always has the value 0.
8 MODE	Mode Select. This field selects the mode in which the port is to operate. The modes of operation include the following: • Normal mode, in which the RXD from the port selected by RXDSEL is routed to the port. • Internal Network mode in which RXD from other ports are ANDed together. RXDSEL is ignored. INMMASK determines which RXD signals are ANDed together. 0 Normal mode 1 Internal Network mode

Table continues on the next page...

AUDMUX_PDCR1 field descriptions (continued)

Field	Description
INMMASK[7:0]	Internal Network Mode Mask. Bit mask that selects the ports from which the RXD signals are to be ANDed together for internal network mode. Bit 6 represents RXD from Port 7 and bit0 represents RXD from Port 1. 0 Includes RXDn for ANDing 1 Excludes RXDn from ANDing

16.6.3 Port Timing Control Register 2 (AUDMUX_PTCR2)

PTCR2 is the Port Timing Control Register for Port 2.

Address: 21D_8000h base + 8h offset = 21D_8008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TFS_DIR	TFSEL[3:0]				TCLKDIR	TCSEL[3:0]				RFS_DIR	RFSEL[3:0]				RCLKDIR
Reset	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RCSEL[3:0]				SYN	0										
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

AUDMUX_PTCR2 field descriptions

Field	Description
31 TFS_DIR	Transmit Frame Sync Direction Control. This bit sets the direction of the TXFS pin of the interface as an output or input. When set as an input, the TFSEL settings are ignored. When set as an output, the TFSEL settings determine the source port of the frame sync. 0 TXFS is an input. 1 TXFS is an output.
30–27 TFSEL[3:0]	Transmit Frame Sync Select. Selects the source port from which TXFS is sourced. 0xxx Selects TXFS from port. 1xxx Selects RXFS from port. x000 Port 1 x110 Port 7 x111 Reserved

Table continues on the next page...

AUDMUX_PTCR2 field descriptions (continued)

Field	Description
26 TCLKDIR	Transmit Clock Direction Control. This bit sets the direction of the TXC pin of the interface as an output or input. When set as an input, the TCSEL settings are ignored. When set as an output, the TCSEL settings determine the source port of the clock. 0 TXC is an input. 1 TXC is an output.
25–22 TCSEL[3:0]	Transmit Clock Select. Selects the source port from which TXC is sourced. 0xxx Selects TXC from port. 1xxx Selects RXC from port. x000 Port 1 x110 Port 7 x111 Reserved
21 RFS_DIR	Receive Frame Sync Direction Control. This bit sets the direction of the RXFS pin of the interface as an output or input. When set as an input, the RFSEL settings are ignored. When set as an output, the RFSEL settings determine the source port of the frame sync. 0 RXFS is an input. 1 RXFS is an output.
20–17 RFSEL[3:0]	Receive Frame Sync Select. Selects the source port from which RXFS is sourced. RXFS can be sourced from TXFS and RXFS from other ports. 0xxx Selects TXFS from port. 1xxx Selects RXFS from port. x000 Port 1 x110 Port 7 x111 Reserved
16 RCLKDIR	Receive Clock Direction Control. This bit sets the direction of the RXC pin of the interface as an output or input. When set as an input, the RCSEL settings are ignored. When set as an output, the RCSEL settings determine the source port of the clock. **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 RXC is an input. 1 RXC is an output.
15–12 RCSEL[3:0]	Receive Clock Select. Selects the source port from which RXC is sourced. RXC can be sourced from TXC and RXC from other ports. 0xxx Selects TXC from port. 1xxx Selects RXC from port. x000 Port 1 x110 Port 7 x111 Reserved
11 SYN	Synchronous/Asynchronous Select. When SYN is set, synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals (that is, the port is a 4-wire interface). When SYN is cleared, asynchronous mode is chosen and separate clock and frame sync signals are used for the transmit and receive sections (that is, the port is a 6-wire interface). **NOTE:** RCLKDIR and SYN should not be changed at the same time.

Table continues on the next page...

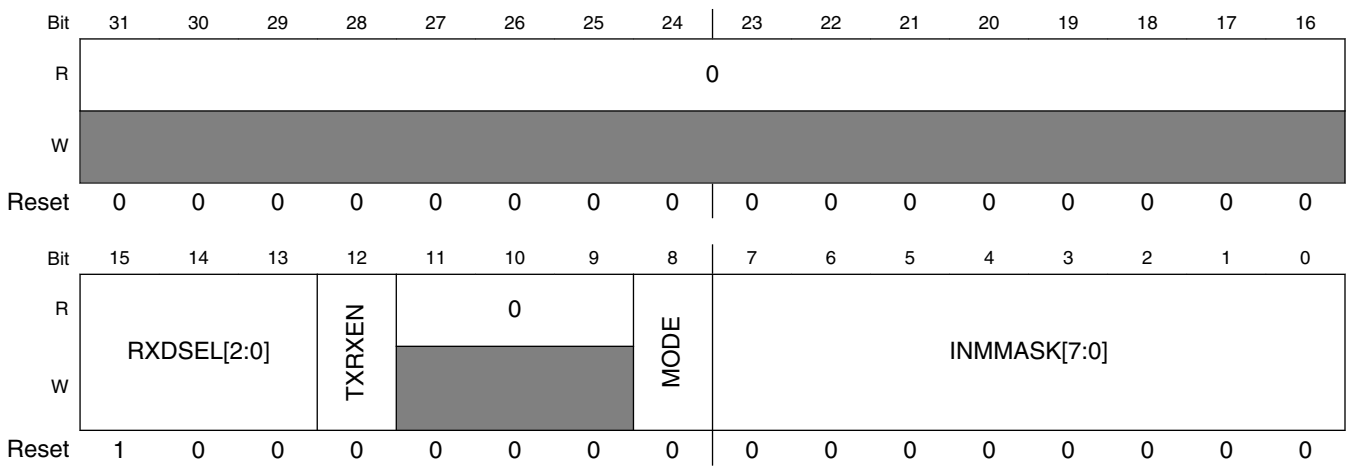
AUDMUX_PTCR2 field descriptions (continued)

Field	Description
	0 Asynchronous mode 1 Synchronous mode (default)
Reserved	This read-only field is reserved and always has the value 0.

16.6.4 Port Data Control Register 2 (AUDMUX_PDCR2)

PDCR2 is the Port Data Control Register for Port 2.

Address: 21D_8000h base + Ch offset = 21D_800Ch



AUDMUX_PDCR2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–13 RXDSEL[2:0]	Receive Data Select. Selects the source port for the RXD data. RXDSEL is ignored if MODE = 1 (that is, Internal Network Mode is enabled). xxx Port number for RXD 000 Port 1 110 Port 7 111 Reserved
12 TXRXEN	Transmit/Receive Switch Enable. Swaps the transmit and receive signals. 0 No switch (Transmit Pin = Transmit, Receive Pin = Receive) 1 Switch (Transmit Pin = Receive, Receive Pin = Transmit)
11–9 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

AUDMUX_PDCR2 field descriptions (continued)

Field	Description
8 MODE	Mode Select. This field selects the mode in which the port is to operate. The modes of operation include the following: • Normal mode, in which the RXD from the port selected by RXDSEL is routed to the port. • Internal Network mode in which RXD from other ports are ANDed together. RXDSEL is ignored. INMMASK determines which RXD signals are ANDed together. 0 Normal mode 1 Internal Network mode
INMMASK[7:0]	Internal Network Mode Mask. Bit mask that selects the ports from which the RXD signals are to be ANDed together for internal network mode. Bit 6 represents RXD from Port 7 and bit0 represents RXD from Port 1. 0 Includes RXDn for ANDing 1 Excludes RXDn from ANDing

16.6.5 Port Timing Control Register 3 (AUDMUX_PTCR3)

PTCR3 is the Port Timing Control Register for Port 3.

Address: 21D_8000h base + 10h offset = 21D_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TFS_DIR	TFSEL[3:0]				TCLKDIR	TCSEL[3:0]				RFS_DIR	RFSEL[3:0]				RCLKDIR
Reset	1	0	0	1	1	1	0	0	1	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RCSEL[3:0]				SYN	0										
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

AUDMUX_PTCR3 field descriptions

Field	Description
31 TFS_DIR	Transmit Frame Sync Direction Control. This bit sets the direction of the TXFS pin of the interface as an output or input. When set as an input, the TFSEL settings are ignored. When set as an output, the TFSEL settings determine the source port of the frame sync. 0 TXFS is an input. 1 TXFS is an output.

Table continues on the next page...

AUDMUX_PTCR3 field descriptions (continued)

Field	Description
30–27 TFSEL[3:0]	Transmit Frame Sync Select. Selects the source port from which TXFS is sourced. 0xxx Selects TXFS from port. 1xxx Selects RXFS from port. x000 Port 1 x110 Port 7 x111 Reserved
26 TCLKDIR	Transmit Clock Direction Control. This bit sets the direction of the TXC pin of the interface as an output or input. When set as an input, the TCSEL settings are ignored. When set as an output, the TCSEL settings determine the source port of the clock. 0 TXC is an input. 1 TXC is an output.
25–22 TCSEL[3:0]	Transmit Clock Select. Selects the source port from which TXC is sourced. 0xxx Selects TXC from port. 1xxx Selects RXC from port. x000 Port 1 x110 Port 7 x111 Reserved
21 RFS_DIR	Receive Frame Sync Direction Control. This bit sets the direction of the RXFS pin of the interface as an output or input. When set as an input, the RFSEL settings are ignored. When set as an output, the RFSEL settings determine the source port of the frame sync. 0 RXFS is an input. 1 RXFS is an output.
20–17 RFSEL[3:0]	Receive Frame Sync Select. Selects the source port from which RXFS is sourced. RXFS can be sourced from TXFS and RXFS from other ports. 0xxx Selects TXFS from port. 1xxx Selects RXFS from port. x000 Port 1 x110 Port 7 x111 Reserved
16 RCLKDIR	Receive Clock Direction Control. This bit sets the direction of the RXC pin of the interface as an output or input. When set as an input, the RCSEL settings are ignored. When set as an output, the RCSEL settings determine the source port of the clock. **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 RXC is an input. 1 RXC is an output.
15–12 RCSEL[3:0]	Receive Clock Select. Selects the source port from which RXC is sourced. RXC can be sourced from TXC and RXC from other ports. 0xxx Selects TXC from port. 1xxx Selects RXC from port. x000 Port 1 x110 Port 7 x111 Reserved

Table continues on the next page...

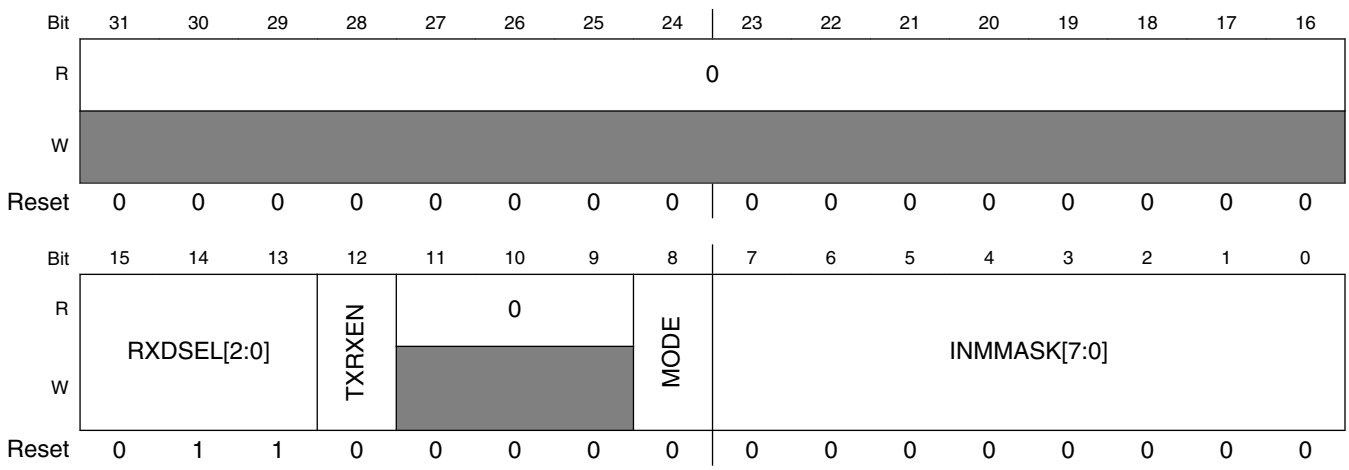
AUDMUX_PTCR3 field descriptions (continued)

Field	Description
11 SYN	Synchronous/Asynchronous Select. When SYN is set, synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals (that is, the port is a 4-wire interface). When SYN is cleared, asynchronous mode is chosen and separate clock and frame sync signals are used for the transmit and receive sections (that is, the port is a 6-wire interface). **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 Asynchronous mode 1 Synchronous mode (default)
Reserved	This read-only field is reserved and always has the value 0.

16.6.6 Port Data Control Register 3 (AUDMUX_PDCR3)

PDCR3 is the Port Data Control Register for Port 3.

Address: 21D_8000h base + 14h offset = 21D_8014h



AUDMUX_PDCR3 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–13 RXDSEL[2:0]	Receive Data Select. Selects the source port for the RXD data. RXDSEL is ignored if MODE = 1 (that is, Internal Network Mode is enabled). xxx Port number for RXD 000 Port 1 110 Port 7 111 Reserved

Table continues on the next page...

AUDMUX_PDCR3 field descriptions (continued)

Field	Description
12 TXRXEN	Transmit/Receive Switch Enable. Swaps the transmit and receive signals. 0 No switch (Transmit Pin = Transmit, Receive Pin = Receive) 1 Switch (Transmit Pin = Receive, Receive Pin = Transmit)
11–9 Reserved	This read-only field is reserved and always has the value 0.
8 MODE	Mode Select. This field selects the mode in which the port is to operate. The modes of operation include the following: • Normal mode, in which the RXD from the port selected by RXDSEL is routed to the port. • Internal Network mode in which RXD from other ports are ANDed together. RXDSEL is ignored. INMMASK determines which RXD signals are ANDed together. 0 Normal mode 1 Internal Network mode
INMMASK[7:0]	Internal Network Mode Mask. Bit mask that selects the ports from which the RXD signals are to be ANDed together for internal network mode. Bit 6 represents RXD from Port 7 and bit0 represents RXD from Port 1. 0 Includes RXDn for ANDing 1 Excludes RXDn from ANDing

16.6.7 Port Timing Control Register 4 (AUDMUX_PTCR4)

Port Timing Control Register for Port 4

Address: 21D_8000h base + 18h offset = 21D_8018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TFS_DIR	TFSEL[3:0]				TCLKDIR	TCSEL[3:0]				RFS_DIR	RFSEL[3:0]				RCLKDIR
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RCSEL[3:0]				SYN	0										
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

AUDMUX_PTCR4 field descriptions

Field	Description
31 TFS_DIR	Transmit Frame Sync Direction Control. This bit sets the direction of the TXFS pin of the interface as an output or input. When set as an input, the TFSEL settings are ignored. When set as an output, the TFSEL settings determine the source port of the frame sync. 0 TXFS is an input. 1 TXFS is an output.
30–27 TFSEL[3:0]	Transmit Frame Sync Select. Selects the source port from which TXFS is sourced. 0xxx Selects TXFS from port. 1xxx Selects RXFS from port. x000 Port 1 x110 Port 7 x111 Reserved
26 TCLKDIR	Transmit Clock Direction Control. This bit sets the direction of the TXC pin of the interface as an output or input. When set as an input, the TCSEL settings are ignored. When set as an output, the TCSEL settings determine the source port of the clock. 0 TXC is an input. 1 TXC is an output.
25–22 TCSEL[3:0]	Transmit Clock Select. Selects the source port from which TXC is sourced. 0xxx Selects TXC from port. 1xxx Selects RXC from port. x000 Port 1 x110 Port 7 x111 Reserved
21 RFS_DIR	Receive Frame Sync Direction Control. This bit sets the direction of the RXFS pin of the interface as an output or input. When set as an input, the RFSEL settings are ignored. When set as an output, the RFSEL settings determine the source port of the frame sync. 0 RXFS is an input. 1 RXFS is an output.
20–17 RFSEL[3:0]	Receive Frame Sync Select. Selects the source port from which RXFS is sourced. RXFS can be sourced from TXFS and RXFS from other ports. 0xxx Selects TXFS from port. 1xxx Selects RXFS from port. x000 Port 1 x110 Port 7 x111 Reserved
16 RCLKDIR	Receive Clock Direction Control. This bit sets the direction of the RXC pin of the interface as an output or input. When set as an input, the RCSEL settings are ignored. When set as an output, the RCSEL settings determine the source port of the clock. **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 RXC is an input. 1 RXC is an output.
15–12 RCSEL[3:0]	Receive Clock Select. Selects the source port from which RXC is sourced. RXC can be sourced from TXC and RXC from other ports.

Table continues on the next page...

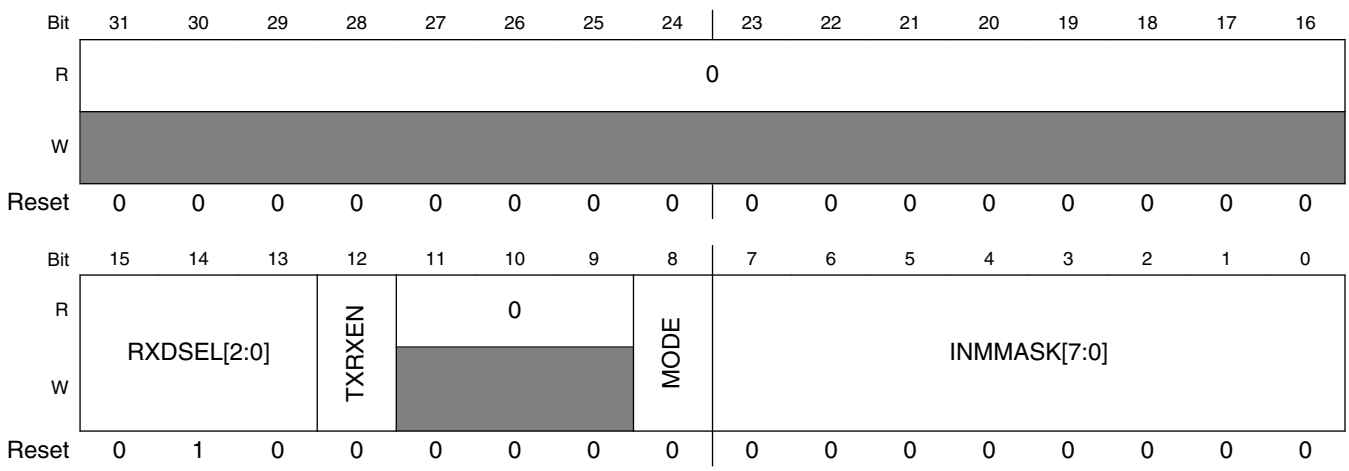
AUDMUX_PTCR4 field descriptions (continued)

Field	Description
	0xxx Selects TXC from port. 1xxx Selects RXC from port. x000 Port 1 x110 Port 7 x111 Reserved
11 SYN	Synchronous/Asynchronous Select. When SYN is set, synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals (that is, the port is a 4-wire interface). When SYN is cleared, asynchronous mode is chosen and separate clock and frame sync signals are used for the transmit and receive sections (that is, the port is a 6-wire interface). **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 Asynchronous mode 1 Synchronous mode (default)
Reserved	This read-only field is reserved and always has the value 0.

16.6.8 Port Data Control Register 4 (AUDMUX_PDCR4)

PDCR4 is the Port Data Control Register for Port 4.

Address: 21D_8000h base + 1Ch offset = 21D_801Ch



AUDMUX_PDCR4 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–13 RXDSEL[2:0]	Receive Data Select. Selects the source port for the RXD data. RXDSEL is ignored if MODE = 1 (that is, Internal Network Mode is enabled).

Table continues on the next page...

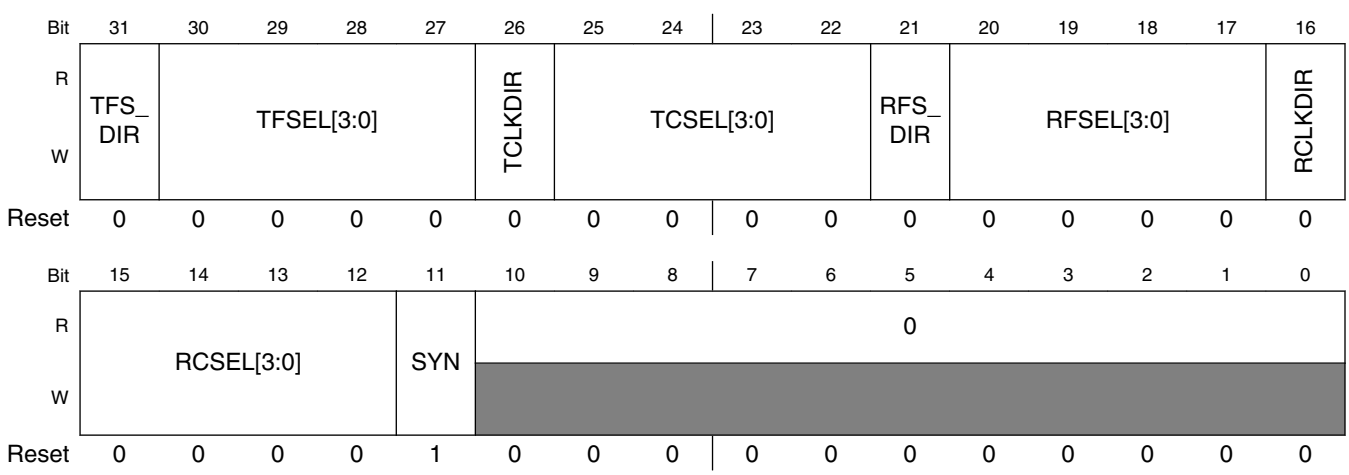
AUDMUX_PDCR4 field descriptions (continued)

Field	Description
	xxx Port number for RXD 000 Port 1 110 Port 7 111 Reserved
12 TXRXEN	Transmit/Receive Switch Enable. Swaps the transmit and receive signals. 0 No switch (Transmit Pin = Transmit, Receive Pin = Receive) 1 Switch (Transmit Pin = Receive, Receive Pin = Transmit)
11–9 Reserved	This read-only field is reserved and always has the value 0.
8 MODE	Mode Select. This field selects the mode in which the port is to operate. The modes of operation include the following: • Normal mode, in which the RXD from the port selected by RXDSEL is routed to the port. • Internal Network mode in which RXD from other ports are ANDed together. RXDSEL is ignored. INMMASK determines which RXD signals are ANDed together. 0 Normal mode 1 Internal Network mode
INMMASK[7:0]	Internal Network Mode Mask. Bit mask that selects the ports from which the RXD signals are to be ANDed together for internal network mode. Bit 6 represents RXD from Port 7 and bit0 represents RXD from Port 1. 0 Includes RXDn for ANDing 1 Excludes RXDn from ANDing

16.6.9 Port Timing Control Register 5 (AUDMUX_PTCR5)

Port Timing Control Register for Port 5

Address: 21D_8000h base + 20h offset = 21D_8020h



AUDMUX_PTCR5 field descriptions

Field	Description
31 TFS_DIR	Transmit Frame Sync Direction Control. This bit sets the direction of the TXFS pin of the interface as an output or input. When set as an input, the TFSEL settings are ignored. When set as an output, the TFSEL settings determine the source port of the frame sync. 0 TXFS is an input. 1 TXFS is an output.
30–27 TFSEL[3:0]	Transmit Frame Sync Select. Selects the source port from which TXFS is sourced. 0xxx Selects TXFS from port. 1xxx Selects RXFS from port. x000 Port 1 x110 Port 7 x111 Reserved
26 TCLKDIR	Transmit Clock Direction Control. This bit sets the direction of the TXC pin of the interface as an output or input. When set as an input, the TCSEL settings are ignored. When set as an output, the TCSEL settings determine the source port of the clock. 0 TXC is an input. 1 TXC is an output.
25–22 TCSEL[3:0]	Transmit Clock Select. Selects the source port from which TXC is sourced. 0xxx Selects TXC from port. 1xxx Selects RXC from port. x000 Port 1 x110 Port 7 x111 Reserved
21 RFS_DIR	Receive Frame Sync Direction Control. This bit sets the direction of the RXFS pin of the interface as an output or input. When set as an input, the RFSEL settings are ignored. When set as an output, the RFSEL settings determine the source port of the frame sync. 0 RXFS is an input. 1 RXFS is an output.
20–17 RFSEL[3:0]	Receive Frame Sync Select. Selects the source port from which RXFS is sourced. RXFS can be sourced from TXFS and RXFS from other ports. 0xxx Selects TXFS from port. 1xxx Selects RXFS from port. x000 Port 1 x110 Port 7 x111 Reserved
16 RCLKDIR	Receive Clock Direction Control. This bit sets the direction of the RXC pin of the interface as an output or input. When set as an input, the RCSEL settings are ignored. When set as an output, the RCSEL settings determine the source port of the clock. **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 RXC is an input. 1 RXC is an output.
15–12 RCSEL[3:0]	Receive Clock Select. Selects the source port from which RXC is sourced. RXC can be sourced from TXC and RXC from other ports.

Table continues on the next page...

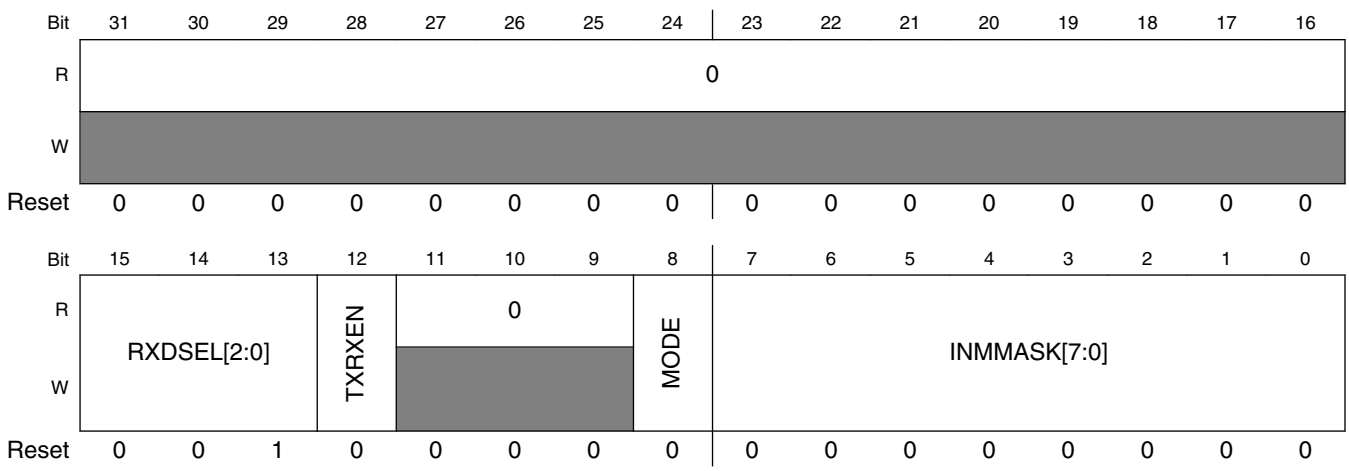
AUDMUX_PTCR5 field descriptions (continued)

Field	Description
	0xxx Selects TXC from port. 1xxx Selects RXC from port. x000 Port 1 x110 Port 7 x111 Reserved
11 SYN	Synchronous/Asynchronous Select. When SYN is set, synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals (that is, the port is a 4-wire interface). When SYN is cleared, asynchronous mode is chosen and separate clock and frame sync signals are used for the transmit and receive sections (that is, the port is a 6-wire interface). **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 Asynchronous mode 1 Synchronous mode (default)
Reserved	This read-only field is reserved and always has the value 0.

16.6.10 Port Data Control Register 5 (AUDMUX_PDCR5)

PDCR5 is the Port Data Control Register for Port 5.

Address: 21D_8000h base + 24h offset = 21D_8024h



AUDMUX_PDCR5 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–13 RXDSEL[2:0]	Receive Data Select. Selects the source port for the RXD data. RXDSEL is ignored if MODE = 1 (that is, Internal Network Mode is enabled).

Table continues on the next page...

AUDMUX_PDCR5 field descriptions (continued)

Field	Description
	xxx Port number for RXD 000 Port 1 110 Port 7 111 Reserved
12 TXRXEN	Transmit/Receive Switch Enable. Swaps the transmit and receive signals. 0 No switch (Transmit Pin = Transmit, Receive Pin = Receive) 1 Switch (Transmit Pin = Receive, Receive Pin = Transmit)
11–9 Reserved	This read-only field is reserved and always has the value 0.
8 MODE	Mode Select. This field selects the mode in which the port is to operate. The modes of operation include the following: • Normal mode, in which the RXD from the port selected by RXDSEL is routed to the port. • Internal Network mode in which RXD from other ports are ANDed together. RXDSEL is ignored. INMMASK determines which RXD signals are ANDed together. 0 Normal mode 1 Internal Network mode
INMMASK[7:0]	Internal Network Mode Mask. Bit mask that selects the ports from which the RXD signals are to be ANDed together for internal network mode. Bit 6 represents RXD from Port 7 and bit0 represents RXD from Port 1. 0 Includes RXDn for ANDing 1 Excludes RXDn from ANDing

16.6.11 Port Timing Control Register 6 (AUDMUX_PTCR6)

Port Timing Control Register for Port 6

Address: 21D_8000h base + 28h offset = 21D_8028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TFS_DIR	TFSEL[3:0]				TCLKDIR	TCSEL[3:0]				RFS_DIR	RFSEL[3:0]				RCLKDIR
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RCSEL[3:0]				SYN	0										
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

AUDMUX_PTCR6 field descriptions

Field	Description
31 TFS_DIR	Transmit Frame Sync Direction Control. This bit sets the direction of the TXFS pin of the interface as an output or input. When set as an input, the TFSEL settings are ignored. When set as an output, the TFSEL settings determine the source port of the frame sync. 0 TXFS is an input. 1 TXFS is an output.
30–27 TFSEL[3:0]	Transmit Frame Sync Select. Selects the source port from which TXFS is sourced. 0xxx Selects TXFS from port. 1xxx Selects RXFS from port. x000 Port 1 x110 Port 7 x111 Reserved
26 TCLKDIR	Transmit Clock Direction Control. This bit sets the direction of the TXC pin of the interface as an output or input. When set as an input, the TCSEL settings are ignored. When set as an output, the TCSEL settings determine the source port of the clock. 0 TXC is an input. 1 TXC is an output.
25–22 TCSEL[3:0]	Transmit Clock Select. Selects the source port from which TXC is sourced. 0xxx Selects TXC from port. 1xxx Selects RXC from port. x000 Port 1 x110 Port 7 x111 Reserved
21 RFS_DIR	Receive Frame Sync Direction Control. This bit sets the direction of the RXFS pin of the interface as an output or input. When set as an input, the RFSEL settings are ignored. When set as an output, the RFSEL settings determine the source port of the frame sync. 0 RXFS is an input. 1 RXFS is an output.
20–17 RFSEL[3:0]	Receive Frame Sync Select. Selects the source port from which RXFS is sourced. RXFS can be sourced from TXFS and RXFS from other ports. 0xxx Selects TXFS from port. 1xxx Selects RXFS from port. x000 Port 1 x110 Port 7 x111 Reserved
16 RCLKDIR	Receive Clock Direction Control. This bit sets the direction of the RXC pin of the interface as an output or input. When set as an input, the RCSEL settings are ignored. When set as an output, the RCSEL settings determine the source port of the clock. **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 RXC is an input. 1 RXC is an output.
15–12 RCSEL[3:0]	Receive Clock Select. Selects the source port from which RXC is sourced. RXC can be sourced from TXC and RXC from other ports.

Table continues on the next page...

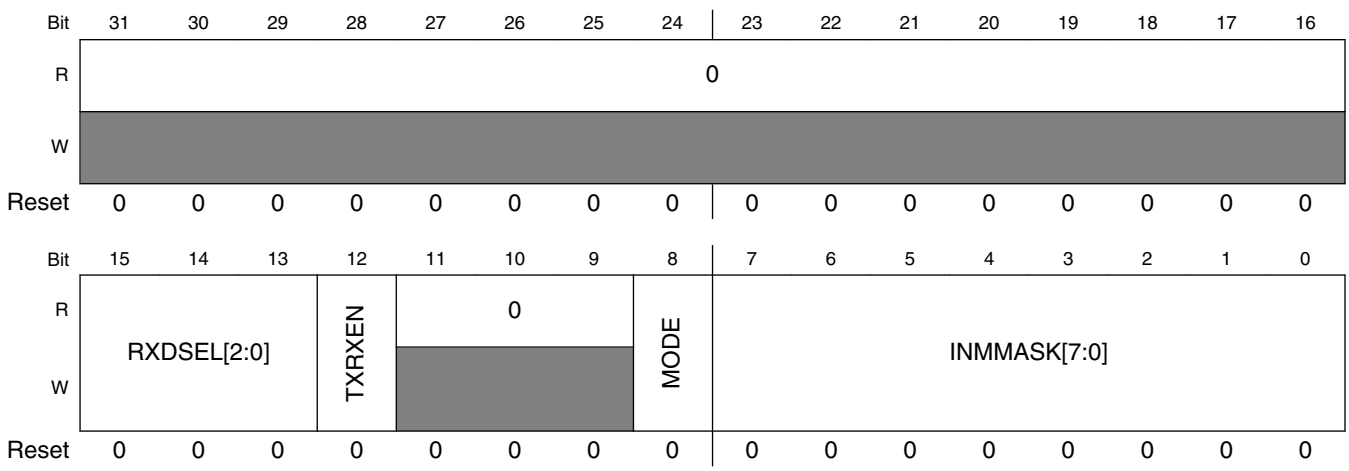
AUDMUX_PTCR6 field descriptions (continued)

Field	Description
	0xxx Selects TXC from port. 1xxx Selects RXC from port. x000 Port 1 x110 Port 7 x111 Reserved
11 SYN	Synchronous/Asynchronous Select. When SYN is set, synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals (that is, the port is a 4-wire interface). When SYN is cleared, asynchronous mode is chosen and separate clock and frame sync signals are used for the transmit and receive sections (that is, the port is a 6-wire interface). **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 Asynchronous mode 1 Synchronous mode (default)
Reserved	This read-only field is reserved and always has the value 0.

16.6.12 Port Data Control Register 6 (AUDMUX_PDCR6)

PDCR6 is the Port Data Control Register for Port 6.

Address: 21D_8000h base + 2Ch offset = 21D_802Ch



AUDMUX_PDCR6 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–13 RXDSEL[2:0]	Receive Data Select. Selects the source port for the RXD data. RXDSEL is ignored if MODE = 1 (that is, Internal Network Mode is enabled).

Table continues on the next page...

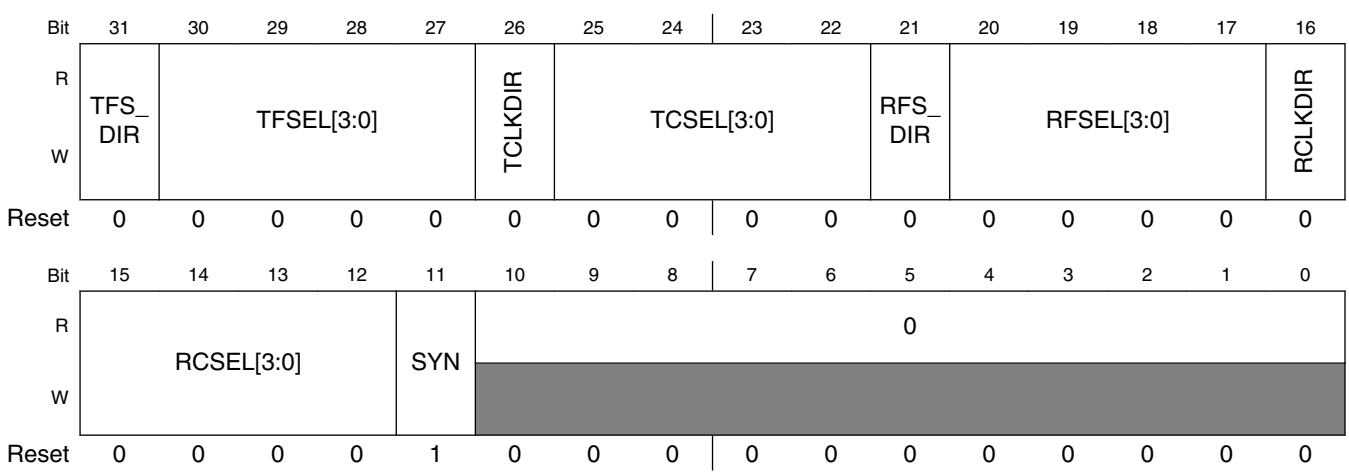
AUDMUX_PDCR6 field descriptions (continued)

Field	Description
	xxx Port number for RXD 000 Port 1 110 Port 7 111 Reserved
12 TXRXEN	Transmit/Receive Switch Enable. Swaps the transmit and receive signals. 0 No switch (Transmit Pin = Transmit, Receive Pin = Receive) 1 Switch (Transmit Pin = Receive, Receive Pin = Transmit)
11–9 Reserved	This read-only field is reserved and always has the value 0.
8 MODE	Mode Select. This field selects the mode in which the port is to operate. The modes of operation include the following: • Normal mode, in which the RXD from the port selected by RXDSEL is routed to the port. • Internal Network mode in which RXD from other ports are ANDed together. RXDSEL is ignored. INMMASK determines which RXD signals are ANDed together. 0 Normal mode 1 Internal Network mode
INMMASK[7:0]	Internal Network Mode Mask. Bit mask that selects the ports from which the RXD signals are to be ANDed together for internal network mode. Bit 6 represents RXD from Port 7 and bit0 represents RXD from Port 1. 0 Includes RXDn for ANDing 1 Excludes RXDn from ANDing

16.6.13 Port Timing Control Register 7 (AUDMUX_PTCR7)

Port Timing Control Register for Port 7

Address: 21D_8000h base + 30h offset = 21D_8030h



AUDMUX_PTCR7 field descriptions

Field	Description
31 TFS_DIR	Transmit Frame Sync Direction Control. This bit sets the direction of the TXFS pin of the interface as an output or input. When set as an input, the TFSEL settings are ignored. When set as an output, the TFSEL settings determine the source port of the frame sync. 0 TXFS is an input. 1 TXFS is an output.
30–27 TFSEL[3:0]	Transmit Frame Sync Select. Selects the source port from which TXFS is sourced. 0xxx Selects TXFS from port. 1xxx Selects RXFS from port. x000 Port 1 x110 Port 7 x111 Reserved
26 TCLKDIR	Transmit Clock Direction Control. This bit sets the direction of the TXC pin of the interface as an output or input. When set as an input, the TCSEL settings are ignored. When set as an output, the TCSEL settings determine the source port of the clock. 0 TXC is an input. 1 TXC is an output.
25–22 TCSEL[3:0]	Transmit Clock Select. Selects the source port from which TXC is sourced. 0xxx Selects TXC from port. 1xxx Selects RXC from port. x000 Port 1 x110 Port 7 x111 Reserved
21 RFS_DIR	Receive Frame Sync Direction Control. This bit sets the direction of the RXFS pin of the interface as an output or input. When set as an input, the RFSEL settings are ignored. When set as an output, the RFSEL settings determine the source port of the frame sync. 0 RXFS is an input. 1 RXFS is an output.
20–17 RFSEL[3:0]	Receive Frame Sync Select. Selects the source port from which RXFS is sourced. RXFS can be sourced from TXFS and RXFS from other ports. 0xxx Selects TXFS from port. 1xxx Selects RXFS from port. x000 Port 1 x110 Port 7 x111 Reserved
16 RCLKDIR	Receive Clock Direction Control. This bit sets the direction of the RXC pin of the interface as an output or input. When set as an input, the RCSEL settings are ignored. When set as an output, the RCSEL settings determine the source port of the clock. **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 RXC is an input. 1 RXC is an output.
15–12 RCSEL[3:0]	Receive Clock Select. Selects the source port from which RXC is sourced. RXC can be sourced from TXC and RXC from other ports.

Table continues on the next page...

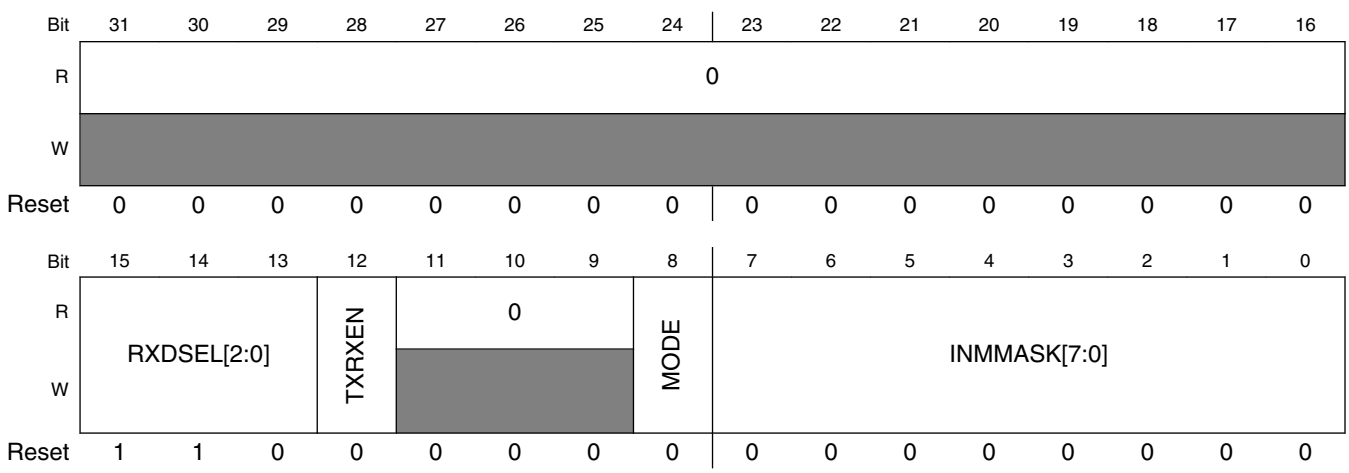
AUDMUX_PTCR7 field descriptions (continued)

Field	Description
	0xxx Selects TXC from port. 1xxx Selects RXC from port. x000 Port 1 x110 Port 7 x111 Reserved
11 SYN	Synchronous/Asynchronous Select. When SYN is set, synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals (that is, the port is a 4-wire interface). When SYN is cleared, asynchronous mode is chosen and separate clock and frame sync signals are used for the transmit and receive sections (that is, the port is a 6-wire interface). **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 Asynchronous mode 1 Synchronous mode (default)
Reserved	This read-only field is reserved and always has the value 0.

16.6.14 Port Data Control Register 7 (AUDMUX_PDCR7)

PDCR7 is the Port Data Control Register for Port 7.

Address: 21D_8000h base + 34h offset = 21D_8034h



AUDMUX_PDCR7 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–13 RXDSEL[2:0]	Receive Data Select. Selects the source port for the RXD data. RXDSEL is ignored if MODE = 1 (that is, Internal Network Mode is enabled).

Table continues on the next page...

AUDMUX_PDCR7 field descriptions (continued)

Field	Description
	xxx Port number for RXD 000 Port 1 110 Port 7 111 Reserved
12 TXRXEN	Transmit/Receive Switch Enable. Swaps the transmit and receive signals. 0 No switch (Transmit Pin = Transmit, Receive Pin = Receive) 1 Switch (Transmit Pin = Receive, Receive Pin = Transmit)
11–9 Reserved	This read-only field is reserved and always has the value 0.
8 MODE	Mode Select. This field selects the mode in which the port is to operate. The modes of operation include the following: • Normal mode, in which the RXD from the port selected by RXDSEL is routed to the port. • Internal Network mode in which RXD from other ports are ANDed together. RXDSEL is ignored. INMMASK determines which RXD signals are ANDed together. 0 Normal mode 1 Internal Network mode
INMMASK[7:0]	Internal Network Mode Mask. Bit mask that selects the ports from which the RXD signals are to be ANDed together for internal network mode. Bit 6 represents RXD from Port 7 and bit0 represents RXD from Port 1. 0 Includes RXDn for ANDing 1 Excludes RXDn from ANDing

Chapter 17
40-BIT Correcting ECC Accelerator (BCH)

17.1 Overview

The hardware ECC accelerator provides a forward error-correction function for improving the reliability of various storage media that may be attached to the device.

For example, NAND flash devices use a spare area to store ecc codes to correct some hard bit errors in data stored within the device, allowing higher device yields and, therefore, lower NAND device costs.

The Bose, Ray-Chaudhuri, Hocquenghem (BCH) Encoder and Decoder module is capable of correcting from 2 to 40 single bit errors within a block of data no larger than about 1900 bytes (512 bytes or 1024 bytes are typical) in applications such as protecting data and resources stored on modern NAND flash devices. The correction level in the BCH block is programmable to provide flexibility for varying applications and configurations of flash page size. The design can be programmed to encode protection of 2 to 40 bit errors when writing flash and to correct the corresponding number of errors on decode. The correction level when decoding MUST be programmed to the same correction level as was used during the encode phase.

BCH-codes are a type of block-code, which implies that all error-correction is performed over a block of N-symbols. The BCH operation will be performed over GF(2^{13} = 8192) or GF(2^{14} = 16384), which is the Galois Field consisting of 8191 or 16383 one-bit symbols. BCH-encoding (or encode for any block-code) can be performed by two algorithms: systematic encoding or multiplicative encoding. Systematic encoding is the process of reading all the symbols which constitute a block, dividing continuously these symbols by the generator polynomial for the GF(8192) or GF(16384) and appending the resulting ***t*** parity symbols to the block to create a BCH codeword (where ***t*** is the number of correctable bits).

The BCH encode process creates *t**13 (or *t**14)-bit parity symbols for each data block when the data is written to the flash device. The parity symbols are written to the flash device after the corresponding data block, and together these are collectively called the codeword. The codeword can be used during the decode process to correct errors that occur in either the data or parity blocks.

The BCH decoder processes code words in a 4-step fashion:

1. Syndrome Calculation (SC): This is the process of reading in all of the symbols of the codeword and continuously dividing by the generator polynomial for the field. 2**t* syndromes must be calculated for each codeword and inspection of the syndromes determines if there are errors: a non-zero set of syndromes indicates one or more errors. This process is implemented parallel hardware to minimize processing time since it must be done every time the decode is performed.
2. Key Equation Solver (KES): The syndromes represent 2t-linear equations with 2t-unknown variables. The process of solving these equations and selecting from the numerous solutions constitutes the KES module. When the KES block completes its operations, it generates an error locator polynomial (sigma) that is used in the proceeding block to determine the locations and values of the errors.
3. Chien Search (CS): This block takes input from the KES block and uses the Chien Algorithm for finding the locations of the errors based on the error locator polynomial. The method basically involves substituting all 8191 symbols from the GF(8192) or 16383 symbols from the GF(16383) into the locator polynomial. All evaluations that produce a zero solution indicate locations of the various errors. Since each located error corresponds to a single bit, the bit in the original data may be corrected by simply flipping the polarity of the incorrect location.
4. Correction: this block has to convert the symbol index and mask information to memory byte indexes and masks.

The BCH block, shown in the figure, was designed to operate in a pipelined fashion to maximize throughput. Aside from the initial latency to fill the pipeline stages, the BCH throughput is about 7/4 cycles/byte. Thus, the bottleneck in performing NAND reads and error corrections is the BCH rate. Current GPMI read rates are approximately 1/2 cycles/byte maximally for the current generation of NAND flash. Fortunately, BCH has a different master clock from GPMI, this gives some flexibility to match the throughput rate. The CPU is not directly involved in generating parity symbols, checking for errors, or correcting them.

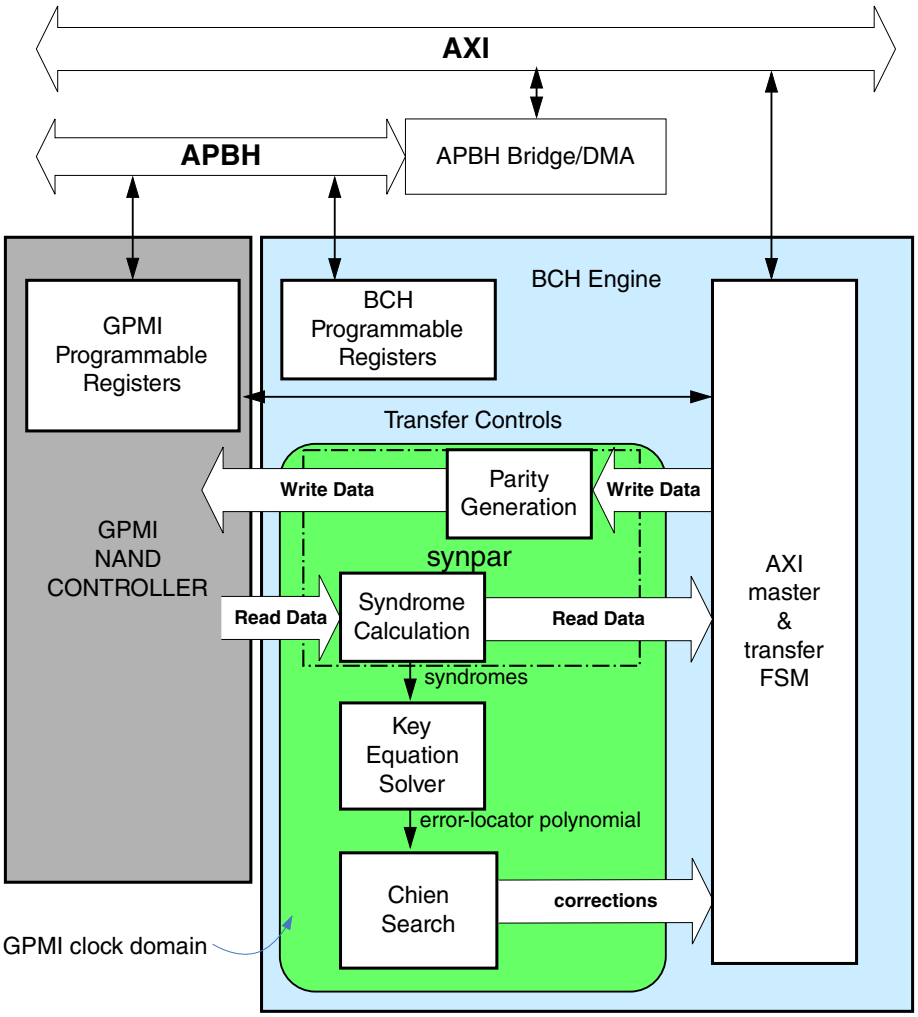


Figure 17-1. Hardware BCH Accelerator

17.2 Operation

Before performing any NAND flash read or write operations, software should first program the BCH's flash layout registers (see Flash Page Layout) to specify how data is to be formatted on the flash device. The BCH hardware allows full programmability over the flash page layout to enable users flexibility in balancing ECC correction levels and ever-changing flash page sizes.

To initiate a NAND Flash write, software will program a GPMI DMA operation. The DMA need only program the GPMI control registers (and handle the requisite flash addressing handshakes) since the BCH will handle all data operations using its AXI bus interface. The BCH will then send the data to the GPMI controller to be written to flash

as it computes the parity symbols. At the end of each data block the BCH will insert the parity symbols into the data stream so that the GPMI sees only a continuous stream of data to be written.

NAND Flash read operations operate in a similar manner. As the GPMI controller reads the device, all data is sent to the BCH hardware for error detection/correction. The BCH controller writes all incoming read data to system memory and in parallel computes the syndromes used to detect bit errors. If errors are detected within a block, the BCH hardware activates the error correction logic to determine where bit errors have occurred and ultimately correct them in the data buffer in system memory. After an entire flash page has been read and corrected, the BCH will signal an interrupt to the CPU.

The figure below indicates how data read from the GPMI is operated on within the BCH hardware. As the BCH receives data from the GPMI (top row), it is written to memory by the BCH's Bus Interface Unit (BIU) (second row). For blocks requiring correction, the KES logic will be activated after the entire block has been received. Once the error locator polynomial has been computed, the corrections are determined by the Chien Search and fed back to the BIU, which performs a read, modify, write operation on the buffer in memory to correct the data.

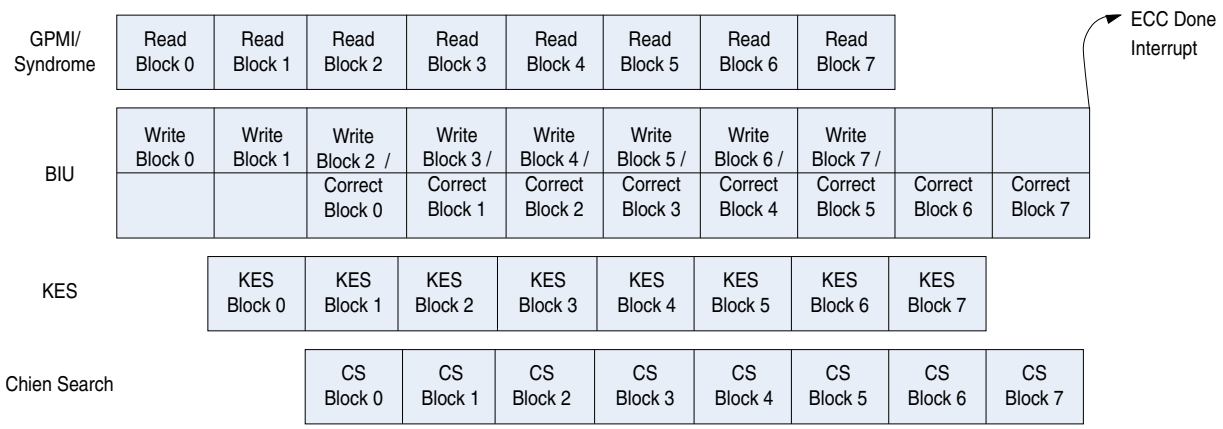


Figure 17-2. Block Pipeline while Reading Flash

17.2.1 BCH Limitations and Assumptions

- The BCH is programmable to support 2 to 40 bit error correction. ECC0 is supported as a pass-through, non-correcting mode.
- Data block sizes must be a multiple of 4 bytes and be aligned in system memory.
- The BCH supports a programmable number of metadata/auxiliary data bytes, from 0 to 255.

- Metadata will be written at the beginning of the flash page to facilitate fast access for filesystem operations.
- Metadata may be treated as an independent block for ECC purposes or combined with the first data block to conserve bits in the flash.
- The BCH does not support a partial page write (this can be accomplished by programming the BCH layout registers such that the BCH only sees a portion of the page).
- Flash read operations can read the entire page or the first block on the page.
- The BCH also supports a memory-to-memory mode of operation that does not require the use of DMA or the GPMI.

17.2.2 Flash Page Layout

The BCH supports a fully programmable flash page layout. The BCH maintains four independent layout registers that can describe four completely different NAND devices or layouts.

When the BCH initiates an operation, it selects one of the layouts by using the chip select as an index into the BCH_LAYOUTSELECT register that determines which layout should be used for the operation.

Three possible (generic) flash layout schemes are supported, as indicated in the figure below. (In each case, the metadata size may also be programmed to 0 bytes). Metadata may either be combined with the first block of data or the size of the first data block can be programmed to 0 to allow the metadata to be protected by its own ECC parity bits.

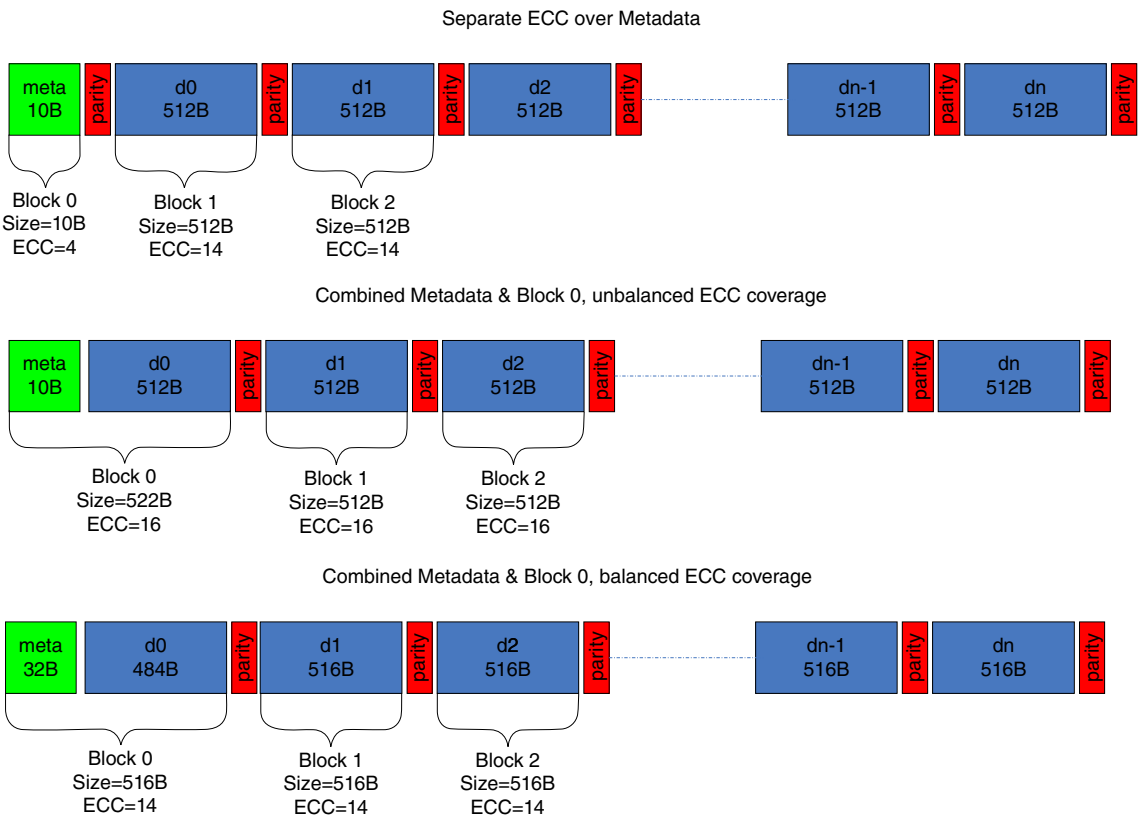


Figure 17-3. FLASH Page Layout Options

Each layout is determined by a pair of registers that define the following parameters:

- DATA0_SIZE: Indicates the number of data bytes in the first block on the page (this should not include parity or metadata bytes). This should be set to 0 when the metadata is to be covered separately with its own ECC. This must be a multiple of 4 bytes.
- ECC0: Indicates the ECC level to be used for the first block on the flash (data0+metadata).
- META_SIZE: Indicates the number of bytes (from 0-255) that are stored as metadata.
- NBLOCKS: Indicates the number of subsequent DATAN blocks on the flash, or the number of blocks following the DATA0 block.
- DATAN_SIZE: Indicates the number of data bytes in all subsequent data blocks. This MUST be a multiple of 4 bytes.
- ECCN: Indicates the ECC level to be used for the subsequent data blocks.

- GF0 or GFN: Indicates the Galois field the meta / data blocks are using
- PAGE_SIZE: Indicates the total number of bytes available per page on the physical flash device. This includes the spare area and is typically 4096+128, 4096+218, or 2048+64 bytes.

17.2.3 Determining the ECC layout for a device

Since the BCH is programmable, a system can trade off ECC levels for flash size and layout configurations.

The following examples indicate how to determine a valid layout based on the required storage space and flash size. For all cases, the size of the parity will be 13 (or 14 for GF(2^{14}))*ECC level ***bits***-- so for ECC8, 13 (or 14) bytes are required (per block).

17.2.3.1 4K+218 flash, 10 bytes metadata, 512 byte data blocks, separate metadata, Assuming GF(2^{13})

In this case, we have 8 data blocks each consisting of 512 bytes. Since the flash has 218 spare bytes (1744 bits), first estimate an ECC level for the data blocks by first subtracting the number of metadata bytes from the spare bytes (218 – 10 = 208 bytes = 1664 bits) then dividing the number of bits by 8 (number of blocks) and then by 13 (bits per ECC level).

(218 - 10) x 8 = 1664/13(8) = 16

Therefore all the data blocks could be covered by ECC16 if the metadata had no parity. This isn't acceptable, so assume ECC14 for all the data blocks. Now calculate the number of free bits for the metadata parity as

1664 - (14) x 13 x 8 = 208

Therefore, 208 bits remain for metadata parity. Dividing by 13 (bits/ECC) gives 16, so the metadata can be covered with ECC16. The settings for this device would then be

Table 17-1. Settings for 4K+218 FLASH

Setting	Value
PAGE_SIZE	4096+218=4314=0x10DA
META_SIZE	10=0x0A
DATA0_SIZE	0
ECC0	16=0x10

Table continues on the next page...

Table 17-1. Settings for 4K+218 FLASH (continued)

Setting	Value
GF0	GF(2^{13})
DATAN_SIZE	512=0x200 (in register interface, assigned as 0x80)
ECCN	14=0x0E
GFN	GF(2^{13})
NBLOCKS	8

17.2.3.2 4K+128 flash, 10 bytes metadata, 1024 byte data blocks, separate metadata, assuming GF(2^{13}) for data and GF(2^{14}) for metadata

This flash will have 118 bytes available for ECC (after subtracting the metadata size), therefore, 994 bits.

Dividing by 4*14 (number of blocks * ECC level) we get 17.75, therefore we can support ECC16 on the data blocks. The number of free spare bits becomes 944 - 16 * 4 * 14 = 944 - 896 = 48, divided by 13 = 3.69, therefore the metadata can be also covered by ECC2.

Table 17-2. Settings for 4K+128 FLASH

Setting	Value
PAGE_SIZE	4096+128=4224=0x1080
META_SIZE	10=0x0A
DATA0_SIZE	0
ECC0	2
GF0	GF(2^{13})
DATAN_SIZE	1024=0x400 (in register interface, assigned as 0x100)
ECCN	16
GFN	GF(2^{14})
NBLOCKS	4

In this case, there will be additional unused spare bits, with the BCH will pad out with zeros.

17.2.4 Data Buffers in System Memory

While the data on the flash is interleaved with parity symbols, the BCH assumes that the data buffers in memory are contiguous.

Metadata read from the flash will be stored to the location pointed to by the GPMI_AUXILIARY register and data will be written to the address specified in the GPMI_PAYLOAD register as is shown in the following figure where the block length is 512 bytes for example. Since the number of blocks on a flash page is programmable, the BCH also writes individual block correction status to the auxiliary pointer at the word-aligned address following the end of the metadata. Optionally, the computed syndromes may also be written to the auxiliary area if the DEBUGSYNDROME bit is set in the control register.

As blocks complete processing, the bus master will accumulate the status for each block and write it to the auxiliary data buffer following the metadata. The metadata area will be padded with 0's until the next word boundary and the status for blocks 0-3 will be written to the next word. The status for subsequent blocks will then be written to the buffer. The status for the first block (metadata block) is also stored in the STATUS_BLK0 register in the BCH_STATUS register. The completion codes for the blocks are indicated in the Table 17-3. Note that the definition of the bytes and their ordering in the auxiliary and payload storage areas are user defined. When this data is read back from the flash and put into memory, it will resemble the original buffer that was written out to the flash.

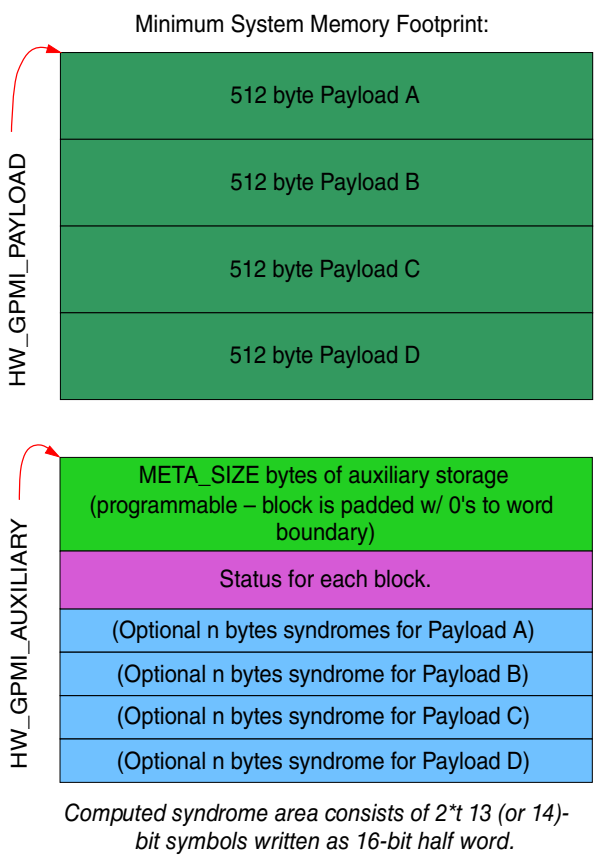


Figure 17-4. BCH Data Buffers in Memory

Table 17-3. Status Block Completion Codes

Code	Description
0xFF	Block is erased
0xFE	Block is uncorrectable
0x00	No errors found
0x01-0x28	Number of errors corrected

The following figure shows the layout of the bytes within the status field.

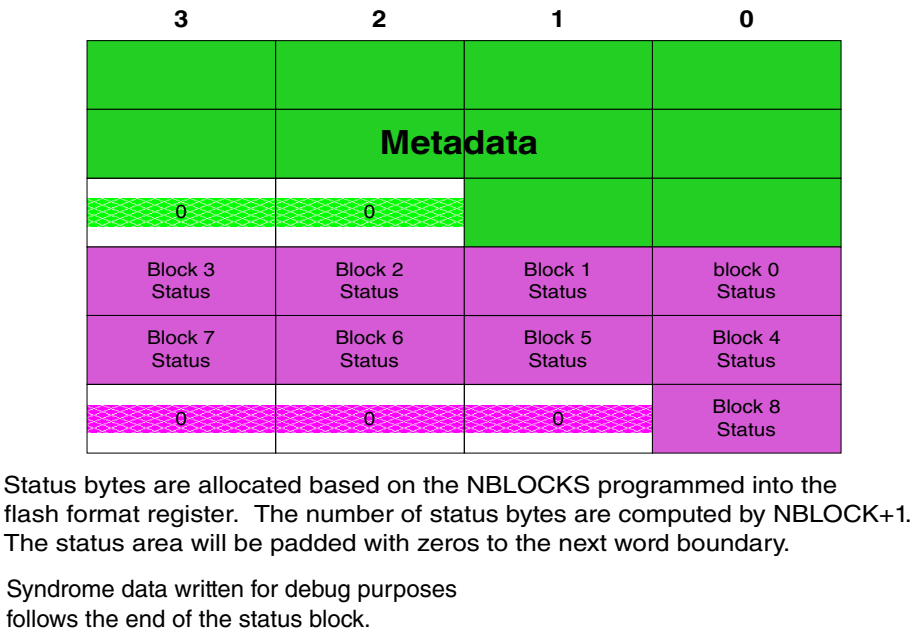


Figure 17-5. Memory-to-Memory Operations

17.3 Memory to Memory (Loopback) Operation

The BCH supports a memory-to-memory mode of operation where both the encoded and decoded buffers reside in system memory.

This can be useful for applications where data must be protected by ECC, but the storage device does not reside on the GPMI bus.

The BCH operation in memory to memory mode is much simpler than in GPMI mode since DMAs are not required to manage the operation. Instead, software simply writes the BCH_DATAPTR and BCH_METAPTR with the addresses of the data and metadata (auxiliary) buffers and the BCH_ENCODEPTR with the address of the buffer for encoded data. To initiate the operation, software simply sets the M2M_ENCODE and

M2M_ENABLE bits in the control register. The BCH can be programmed to either issue an interrupt at the end of the operation or software may poll the status bits for completion.

Memory to memory decode operations work in a similar manner. The encoded data address is written to the BCH_ENCODEPTR and the data and meta pointers are written to buffers that correspond to the desired decoded data addresses. To initiate a decode, software must set the M2M_ENCODE bit to 0 while writing the M2M_ENABLE bit. Note that the addresses written to the BCH_DATAPTR, BCH_METAPTR and BCH_ENCODEPTR registers should always be aligned on a 4 byte boundary. In other words, the 2 lower bits of the address should always be written with zeros.

17.4 Programming the BCH/GPMI Interfaces

Programming the BCH for NAND operations consists largely of disabling the soft reset and clock bits (SFTRST and CLKGATE) from the BCH_CTRL register and then programming the flash layout registers to correspond to the format of the attached NAND device(s).

The BCH_LAYOUTSELECT register should also be programmed to map the chip select of each attached device into one of the four layout registers.

The bulk of the programming is actually applied to the GPMI through PIO operations embedded in DMA command structures. The DMA will perform all the requisite handshaking with the GPMI interface to negotiate the address portion of the transfer, then the BCH will handle all the movement of data from memory to the GPMI (writes) or the GPMI to memory (reads). The BCH will direct all data blocks to the buffer pointed to by the PAYLOAD_BUFFER and the metadata will be written to the AUXILIARY_BUFFER. Both of these registers are located in the GPMI PIO data space and are communicated to the BCH hardware at the beginning of the transfer. Thus, the normal multi-NAND DMA based device interleaving is preserved, that is, four NANDs on four separate chip selects can be scheduled for read or write operations using the BCH. Whichever channel finishes its ready wait first and enters the DMA arbiter with its lock bit set owns the GPMI command interface and through it owns the BCH resources for the duration of its processing.

17.4.1 BCH Encoding for NAND Writes

The BCH encoder flowchart in Figure 17-6 shows the detailed steps involved in programming and using the BCH encoder. This flowchart shows how to use the BCH block with the GPMI.

To use the BCH encoder with the GPMI's DMA, create a DMA command chain containing ten descriptor structures, as shown in Figure 17-8 and detailed in the DMA structure code example that follows it in DMA Structure Code Example. The ten descriptors perform the following tasks:

1. Disable the BCH block (in case it was enabled) and issue NAND write setup command byte (under CLE) and address bytes (under ALE).
2. Configure and enable the BCH and GPMI blocks to perform the NAND write.
3. Disable the BCH block and issue NAND write execute command byte (under CLE).
4. Wait for the NAND device to finish writing the data by watching the ready signal.
5. Check for NAND timeout through PSENSE.
6. Issue NAND status command byte (under CLE).
7. Read the status and compare against expected.
8. If status is incorrect or incomplete, branch to error handling descriptor chain.
9. Otherwise, write is complete and emit GPMI interrupt.

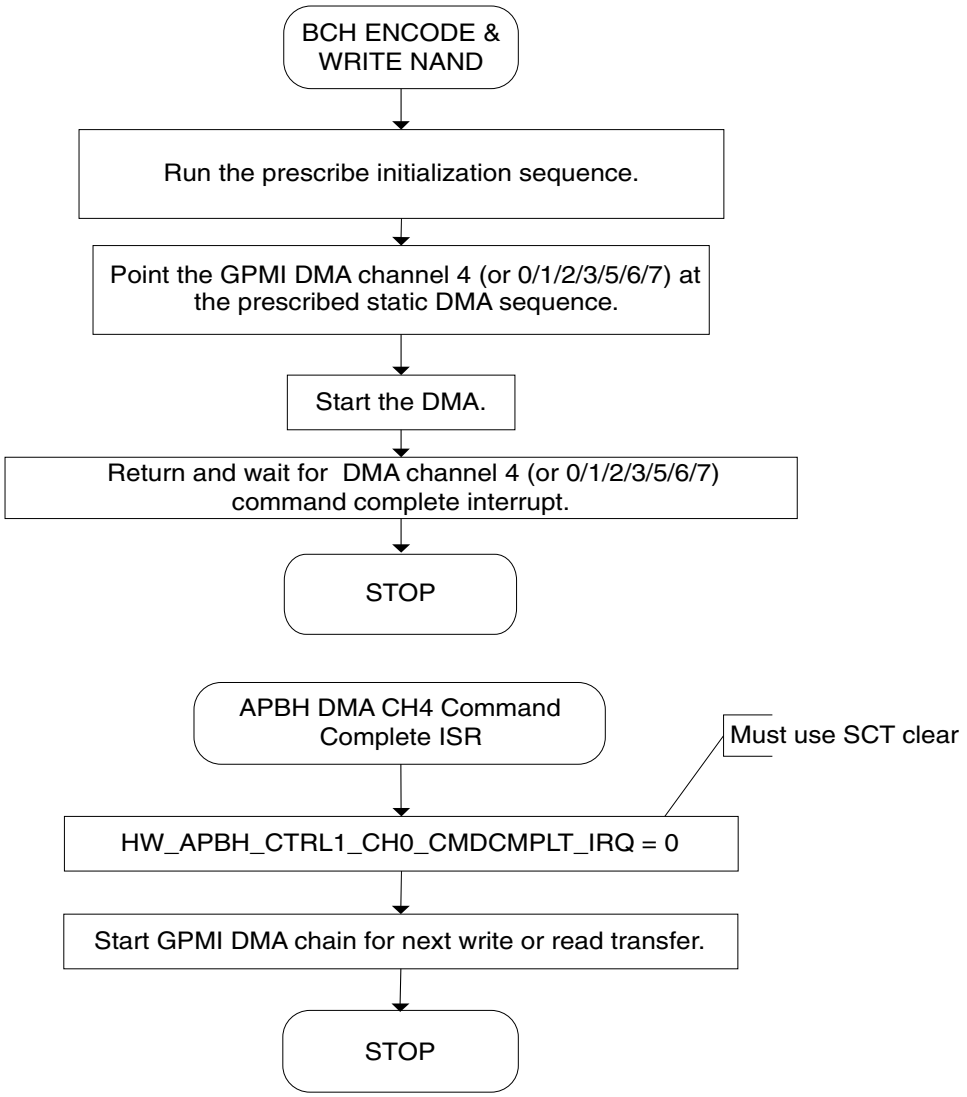


Figure 17-6. BCH Encode Flowchart

Descriptor Legend

NEXT CMD ADDR										
CMD	<=	xfer_count	cmdwords	wait4endcmd	semaphore	nandwait4ready	nandlock	irqoncmplt	chain	command
BUFFER ADDR										
HW_GPMI_CTRL0	<=	command_mode	word_length	lock_cs	CS	address	address_increment	xfer_count		
HW_GPMI_COMPARE	<=	mask				reference				
HW_GPMI_ECCCTRL	<=	ecc_cmd			enable_ecc				buffer_mask	
HW_GPMI_ECCCOUNT										
HW_GPMI_PAYLOAD										
HW_GPMI_AUXILIARY										

Figure 17-7. BCH DMA Descriptor Legend

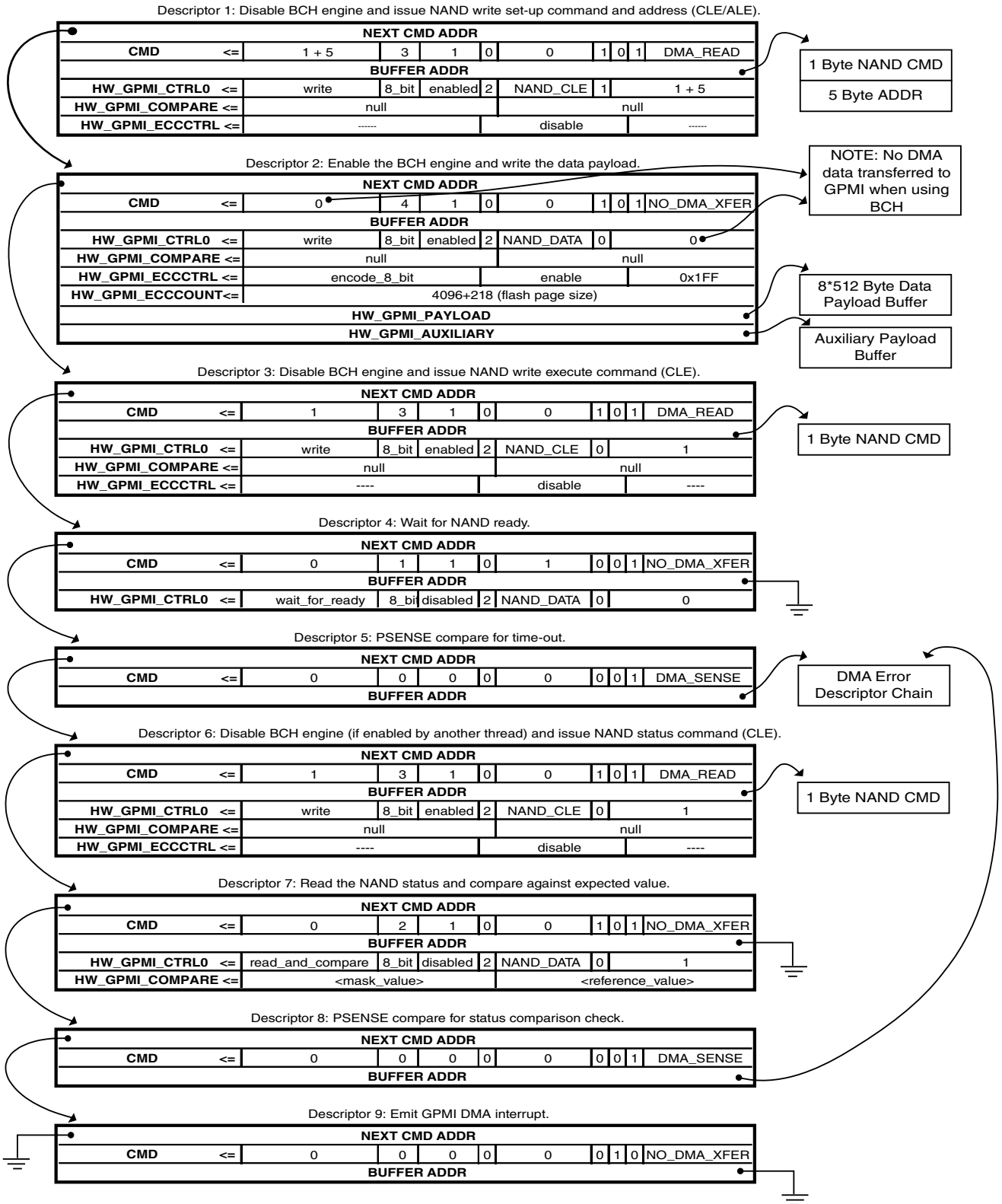


Figure 17-8. BCH Encode DMA Descriptor Chain

17.4.1.1 DMA Structure Code Example

The following code sample illustrates the coding for one write transaction involving 4096 bytes of data payload (eight 512-byte blocks) and 10 bytes of auxiliary payload (also referred to as metadata) to a 4K NAND page sitting on GPMI CS2.

```
//-------------------------------------------------------------------------
// generic DMA/GPMI/ECC descriptor struct, order sensitive!
//-------------------------------------------------------------------------
typedef struct {
  // DMA related fields
  unsigned int dma_nxtcmdar;
  unsigned int dma_cmd;
  unsigned int dma_bar;
  // GPMI related fields
  unsigned int gpmi_ctrl0;
  unsigned int gpmi_compare;
  unsigned int gpmi_eccctrl;
  unsigned int gpmi_ecccount;
  unsigned int gpmi_data_ptr;
  unsigned int gpmi_aux_ptr;
} GENERIC_DESCRIPTOR;
//-------------------------------------------------------------------------
// allocate 10 descriptors for doing a NAND ECC Write
//-------------------------------------------------------------------------
GENERIC_DESCRIPTOR write[10];
//-------------------------------------------------------------------------
// DMA descriptor pointer to handle error conditions from psense checks
//-------------------------------------------------------------------------
unsigned int * dma_error_handler;
//-------------------------------------------------------------------------
// 8 byte NAND command and address buffer
// any alignment is ok, it is read by the GPMI DMA
//   byte 0 is write setup command
//   bytes 1-5 is the NAND address
//   byte 6 is write execute command
//   byte 7 is status command
//-------------------------------------------------------------------------
unsigned char nand_cmd_addr_buffer[8];
//-------------------------------------------------------------------------
// 4096 byte payload buffer used for reads or writes
// needs to be word aligned
//-------------------------------------------------------------------------
unsigned int write_payload_buffer[(4096/4)];
//-------------------------------------------------------------------------
// 65 byte meta-data to be written to NAND
// needs to be word aligned
//-------------------------------------------------------------------------
unsigned int write_aux_buffer[65];
//-------------------------------------------------------------------------
// Descriptor 1: issue NAND write setup command (CLE/ALE)
//-------------------------------------------------------------------------
write[0].dma_nxtcmdar = &write[1];                              // point to the next descriptor
write[0].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (1 + 5)|   // 1 byte command, 5 byte address
                   BF_APBH_CHn_CMD_CMDWORDS      (3)    |   // send 3 words to the GPMI
                 BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)    |  // wait for command to finish
before
                                                               //    continuing
                   BF_APBH_CHn_CMD_SEMAPHORE     (0)    |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)    |
                   BF_APBH_CHn_CMD_NANDLOCK      (1)    |   // prevent other DMA channels
from
                                                               //    taking over
                   BF_APBH_CHn_CMD_IRQONCMPLT    (0)    |
                   BF_APBH_CHn_CMD_CHAIN         (1)    |   // follow chain to next command
```

```
                BV_FLD(APBH_CHn_CMD, COMMAND, DMA_READ); // read data from DMA, write to
NAND
write[0].dma_bar = &nand_cmd_addr_buffer;           // byte 0 write setup, bytes 1 - 5 NAND
address
// 3 words sent to the GPMI
write[0].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)     | // write to the NAND
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)     |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      ENABLED)   |
                    BF_GPMI_CTRL0_CS              (2)       | // must correspond to NAND CS
used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_CLE) |
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (1)       | // send command and
address
                    BF_GPMI_CTRL0_XFER_COUNT       (1 + 5);    // 1 byte command, 5 byte
address
write[0].gpmi_compare = NULL;                            // field not used but necessary to
set eccctrl
write[0].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, DISABLE); // disable the ECC block
//-------------------------------------------------------------------------
// Descriptor 2: write the data payload (DATA)
//-------------------------------------------------------------------------
write[1].dma_nxtcmdar = &write[2];                        // point to the next descriptor
write[1].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT      (0)|  // NOTE: No DMA data transfer
                   BF_APBH_CHn_CMD_CMDWORDS        (4)|  // send 4 words to the GPMI
                   BF_APBH_CHn_CMD_WAIT4ENDCMD     (1)|  // Wait to end
                   BF_APBH_CHn_CMD_SEMAPHORE       (0)|
                   BF_APBH_CHn_CMD_NANDWAIT4READY  (0)|
                   BF_APBH_CHn_CMD_NANDLOCK        (1)|  // maintain resource lock
                   BF_APBH_CHn_CMD_IRQONCMPLT      (0)|
                   BF_APBH_CHn_CMD_CHAIN           (1)|  // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, DMA_NO_XFER);  // No data transferred
write[1].dma_bar = &write_payload_buffer;                  // pointer for the 4K byte
data area
// 4 words sent to the GPMI
write[1].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)     | // write to the NAND
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)     |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      ENABLED)   |
                    BF_GPMI_CTRL0_CS                (2)       | // must correspond to NAND
CS used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA)|
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)       |
                      BF_GPMI_CTRL0_XFER_COUNT         (0);        // NOTE: this field
contains
                                                                 //  the total amount
                              // DMA transferred to GPMI via DMA (0)!
write[1].gpmi_compare = NULL;                             // field not used but necessary
to
set eccctrl
write[1].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ECC_CMD,    ENCODE_8_BIT) | // specify t = 8
mode
                        BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, ENABLE)       | // enable ECC module
                     BF_GPMI_ECCCTRL_BUFFER_MASK   (0x1FF);          // write all 8 data
blocks
                                                                   //   and 1 aux block
write[1].gpmi_ecccount = BF_GPMI_ECCCOUNT_COUNT(4096+218);      // specify number of bytes
                                                                //   written to NAND
write[1].gpmi_data_pointer = &write_payload_pointer;            // data buffer address
write[1].gpmi_aux_pointer  = &write_aux_pointer;                // metadata pointer
//-------------------------------------------------------------------------
// Descriptor 3: issue NAND write execute command (CLE)
//-------------------------------------------------------------------------
write[2].dma_nxtcmdar = &write[3];                          // point to the next descriptor
write[2].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (1) |     // 1 byte command
                   BF_APBH_CHn_CMD_CMDWORDS      (3) |     // send 3 words to the GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1) |     // wait for command to finish
before
                                                          //    continuing
                   BF_APBH_CHn_CMD_SEMAPHORE     (0) |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0) |
                   BF_APBH_CHn_CMD_NANDLOCK      (1) |     // maintain resource lock
```

```
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0) |
                  BF_APBH_CHn_CMD_CHAIN         (1) |     // follow chain to next command
                BV_FLD(APBH_CHn_CMD, COMMAND, DMA_READ);   // read data from DMA, write to
NAND
write[2].dma_bar = &nand_cmd_addr_buffer[6];             // point to byte 6, write execute
command
// 3 words sent to the GPMI
write[2].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)    | // write to the NAND
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)    |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      ENABLED)  |
                    BF_GPMI_CTRL0_CS               (2)      | // must correspond to NAND CS
used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_CLE) |
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)       |
                      BF_GPMI_CTRL0_XFER_COUNT         (1);       // 1 byte command
write[2].gpmi_compare = NULL;                           // field not used but necessary to set
eccctrl
write[2].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, DISABLE); // disable the ECC block
//--------------------------------------------------------------------------
// Descriptor 4: wait for ready (CLE)
//--------------------------------------------------------------------------
write[3].dma_nxtcmdar = &write[4];                       // point to the next descriptor

write[3].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0) |   // no dma transfer
                   BF_APBH_CHn_CMD_CMDWORDS      (1) |   // send 1 word to the GPMI
                   BF_APBH_CHn_CMD_WAIT4ENDCMD   (1) |   // wait for command to finish before
                                                         //      continuing
                   BF_APBH_CHn_CMD_SEMAPHORE     (0) |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(1) |   // wait for nand to be ready
                   BF_APBH_CHn_CMD_NANDLOCK      (0) |   // relinquish nand lock
                   BF_APBH_CHn_CMD_IRQONCMPLT    (0) |
                   BF_APBH_CHn_CMD_CHAIN         (1) |   // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);          // no dma transfer
write[3].dma_bar = NULL;                                                // field not used
// 1 word sent to the GPMI
write[3].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WAIT_FOR_READY) | // wait for NAND
ready
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)         |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)      |
                  BF_GPMI_CTRL0_CS              (2)         | // must correspond to NAND CS
used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA)     |
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)            |
                      BF_GPMI_CTRL0_XFER_COUNT         (0);
//--------------------------------------------------------------------------
// Descriptor 5: psense compare (time out check)
//--------------------------------------------------------------------------
write[4].dma_nxtcmdar = &write[5];                         // point to the next descriptor
write[4].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)       | // no dma transfer
                   BF_APBH_CHn_CMD_CMDWORDS      (0)       | // no words sent to GPMI
                   BF_APBH_CHn_CMD_WAIT4ENDCMD   (0)       | // do not wait to continue
                   BF_APBH_CHn_CMD_SEMAPHORE     (0)       |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)       |
                   BF_APBH_CHn_CMD_NANDLOCK      (0)       |
                   BF_APBH_CHn_CMD_IRQONCMPLT    (0)       |
                   BF_APBH_CHn_CMD_CHAIN         (1)       | // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, DMA_SENSE);  // perform a sense check
write[4].dma_bar = dma_error_handler;                // if sense check fails, branch to error
handler
//--------------------------------------------------------------------------
// Descriptor 6: issue NAND status command (CLE)
//--------------------------------------------------------------------------
write[5].dma_nxtcmdar = &write[6];                         // point to the next descriptor
write[5].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (1)     |  // 1 byte command
                   BF_APBH_CHn_CMD_CMDWORDS      (3)     |  // send 3 words to the GPMI
                 BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)     |  // wait for command to finish
before
continuing
                   BF_APBH_CHn_CMD_SEMAPHORE     (0)     |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)     |
```

```
                    BF_APBH_CHn_CMD_NANDLOCK      (1)    |  // prevent other DMA channels from
taking over
                     BF_APBH_CHn_CMD_IRQONCMPLT    (0)    |
                     BF_APBH_CHn_CMD_CHAIN         (1)    |  // follow chain to next command
                  BV_FLD(APBH_CHn_CMD, COMMAND, DMA_READ);  // read data from DMA, write to
NAND
write[5].dma_bar = &nand_cmd_addr_buffer[7];                 // point to byte 7, status
command
write[5].gpmi_compare = NULL;                              // field not used but necessary to set
eccctrl
write[5].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, DISABLE); // disable the ECC block
// 3 words sent to the GPMI
write[5].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)    | // write to the NAND
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)    |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      ENABLED)  |
                    BF_GPMI_CTRL0_CS               (2)     | // must correspond to NAND CS
used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_CLE) |
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)       |
                      BF_GPMI_CTRL0_XFER_COUNT         (1);        // 1 byte command
//--------------------------------------------------------------------------
// Descriptor 7: read status and compare (DATA)
//--------------------------------------------------------------------------
write[6].dma_nxtcmdar = &write[7];                           // point to the next descriptor
write[6].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)  |    // no dma transfer
                   BF_APBH_CHn_CMD_CMDWORDS      (2)  |    // send 2 words to the GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)  |    // wait for command to finish
before
                                                           //      continuing
                    BF_APBH_CHn_CMD_SEMAPHORE     (0)  |
                    BF_APBH_CHn_CMD_NANDWAIT4READY(0)  |
                    BF_APBH_CHn_CMD_NANDLOCK      (1)  |    // maintain resource lock
                    BF_APBH_CHn_CMD_IRQONCMPLT    (0)  |
                    BF_APBH_CHn_CMD_CHAIN         (1)  |    // follow chain to next command
                    BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);  // no dma transfer
write[6].dma_bar = NULL;                                     // field not used
// 2 word sent to the GPMI
write[6].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, READ_AND_COMPARE) | // read from the
                                                                  //       NAND and
//                                                               // compare to expect
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)     |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)  |
                   BF_GPMI_CTRL0_CS             (2)       |   // must correspond to NAND CS
used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA) |
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)        |
                      BF_GPMI_CTRL0_XFER_COUNT         (1);
write[6].gpmi_compare = <MASK_AND_REFERENCE_VALUE>;  // NOTE: mask and reference values are
NAND
                                                          // SPECIFIC to evaluate the NAND
status
//--------------------------------------------------------------------------
// Descriptor 8: psense compare (time out check)
//--------------------------------------------------------------------------
write[7].dma_nxtcmdar = &write[8];                             // point to the next descriptor
write[7].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)         | // no dma transfer
                   BF_APBH_CHn_CMD_CMDWORDS      (0)         | // no words sent to GPMI
                   BF_APBH_CHn_CMD_WAIT4ENDCMD   (0)         | // do not wait to continue
                   BF_APBH_CHn_CMD_SEMAPHORE     (0)         |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)         |
                   BF_APBH_CHn_CMD_NANDLOCK      (0)         | // relinquish nand lock
                   BF_APBH_CHn_CMD_IRQONCMPLT    (0)         |
                   BF_APBH_CHn_CMD_CHAIN         (1)         | // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, DMA_SENSE);  // perform a sense check
write[7].dma_bar = dma_error_handler;              // if sense check fails, branch to error
handler
//--------------------------------------------------------------------------
// Descriptor 9: emit GPMI interrupt
//--------------------------------------------------------------------------
write[8].dma_nxtcmdar = NULL;                                // not used since this is
```

```
last
descriptor
write[8].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT     (0)          | // no dma transfer
                   BF_APBH_CHn_CMD_CMDWORDS       (0)          | // no words sent to GPMI
                   BF_APBH_CHn_CMD_WAIT4ENDCMD    (0)          | // do not wait to continue
                   BF_APBH_CHn_CMD_SEMAPHORE      (0)          |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)           |
                   BF_APBH_CHn_CMD_NANDLOCK       (0)          |
                   BF_APBH_CHn_CMD_IRQONCMPLT     (1)          | // emit GPMI interrupt
                   BF_APBH_CHn_CMD_CHAIN          (0)          | // terminate DMA chain
processing
                   BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);  // no dma transfer
```

17.4.1.2 Using the BCH Encoder

To use the BCH encoder, first turn off the module-wide soft reset bit in both the GPMI and BCH blocks before starting any DMA activity.

Turning off the soft reset must take place by itself, prior to programming the rest of the control registers. Turn off the BCH bus master soft reset bit. Turn off the clock gate bits.

Program the remainder of the GPMI, BCH and APBH DMA as follows:

```
// bring APBH out of reset
APBH_CTRL0_CLR(BM_APBH_CTRL0_SFRST);
APBH_CTRL0_CLR(BM_APBH_CTRL0_CLKGATE);

// bring BCH out of reset
BCH_CTRL_CLR(BM_BCH_CTRL_SFTRST);
BCH_CTRL_CLR(BM_BCH_CTRL_CLKGATE);

// bring gpmi out of reset
GPMI_CTRL0_CLR(BM_GPMI_CTRL0_SFTRST);
GPMI_CTRL0_CLR(BM_GPMI_CTRL0_CLKGATE);
GPMI_CTRL1_SET(BM_GPMI_CTRL1_DEV_RESET | // deassert reset
                   BM_GPMI_CTRL1_BCH_MODE  );    // enable BCH mode

// enable pinctrl
PINCTRL_CTRL_WR(0x00000000);

// enable gpmi pins
PINCTRL_MUXSEL0_CLR(0x0000ffff);    // data bits
PINCTRL_MUXSEL1_CLR(0x03ffffff);    // control bits
PINCTRL_MUXSEL8_CLR(0x0003f3ff);    // control bits
PINCTRL_MUXSEL8_SET(0x00015155);    // control bits
```

Note that for writing NANDs (ECC encoding), only GPMI DMA command complete interrupts are used. The BCH engine is used for writing to the NAND but may optionally produces an interrupt. From the sample code in DMA Structure Code Example :

- DMA descriptor 1 prepares the NAND for data write by using the GPMI to issue a write setup command byte under CLE, then sends a 5-byte address under ALE. The BCH engine is disabled and not used for these commands.

- DMA descriptor 2 enables the BCH engine for encoding to begin the initial writing of the NAND data by specifying where the data and auxiliary payload are coming from in system memory.
- DMA descriptor 3 issues the write commit command byte under CLE to the NAND.
- DMA descriptor 4 waits for the NAND to complete the write commit/transfer by watching the NAND's ready line status. This descriptor relinquishes the NANDLOCK on the GPMI to enable the other DMA channels to initiate NAND transactions on different NAND CS lines.
- DMA descriptor 6 issues a NAND status command byte under "CLE" to check the status of the NAND device following the page write.
- DMA descriptor 7 reads back the NAND status and compares the status with an expected value. If there are differences, then the DMA processing engine follows an error-handling DMA descriptor path.
- DMA descriptor 8 disables the BCH engine and emits a GPMI interrupt to indicate that the NAND write has been completed.

17.4.2 BCH Decoding for NAND Reads

When a page is read from NAND flash, BCH syndromes will be computed and, if correctable errors are found, they will be corrected on a per block basis within the NAND page. This decoding process is fully overlapped with other NAND data reads and with CPU execution. The BCH decoder flowchart in the figure below shows the steps involved in programming the decoder. The hardware flow of reading and decoding a 4096-byte page is shown in Figure 17-10.

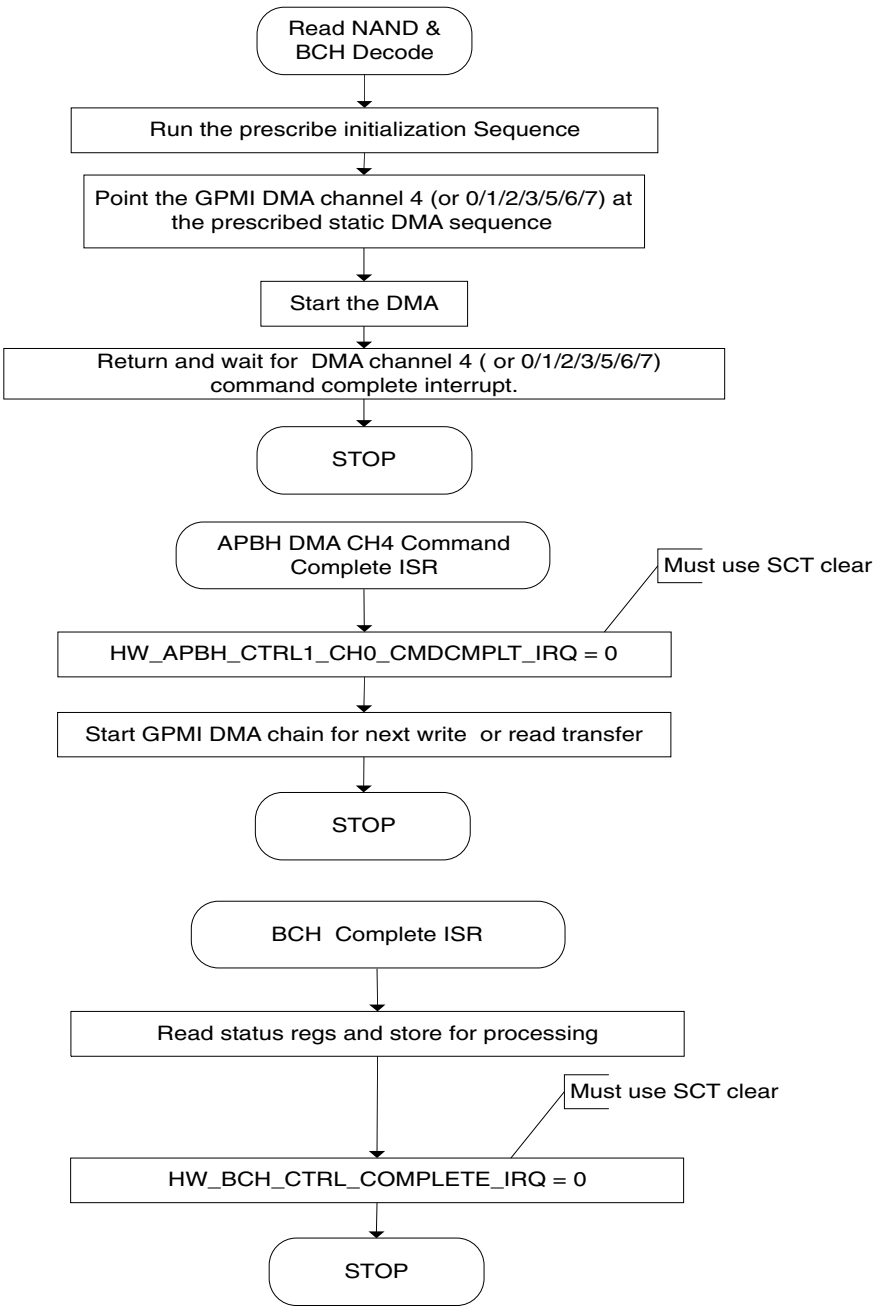


Figure 17-9. BCH Decode Flowchart

Conceptually, an APHB DMA Channel (0,1,2 or 3) command chain with seven command structures linked together is used to perform the BCH decode operation (as shown in Figure 17-10).

Note

The GPMI's DMA command structures controls the BCH decode operation.

To use the BCH decoder with the GPMI's DMA, create a DMA command chain containing seven descriptor structures, as shown in the figure below and detailed in the DMA structure code example that follows it in DMA Structure Code Example. The seven DMA descriptors perform the following tasks:

1. Issue NAND read setup command byte (under "CLE") and address bytes (under "ALE").
2. Issue NAND read execute command byte (under "CLE").
3. Wait for the NAND device to complete accessing the block data by watching the ready signal.
4. Check for NAND timeout through "PSENSE".
5. Configure and enable the BCH block and read the NAND block data.
6. Disable the BCH block.
7. Descriptor NOP to allow NANDLOCK in the previous descriptor to the thread-safe.

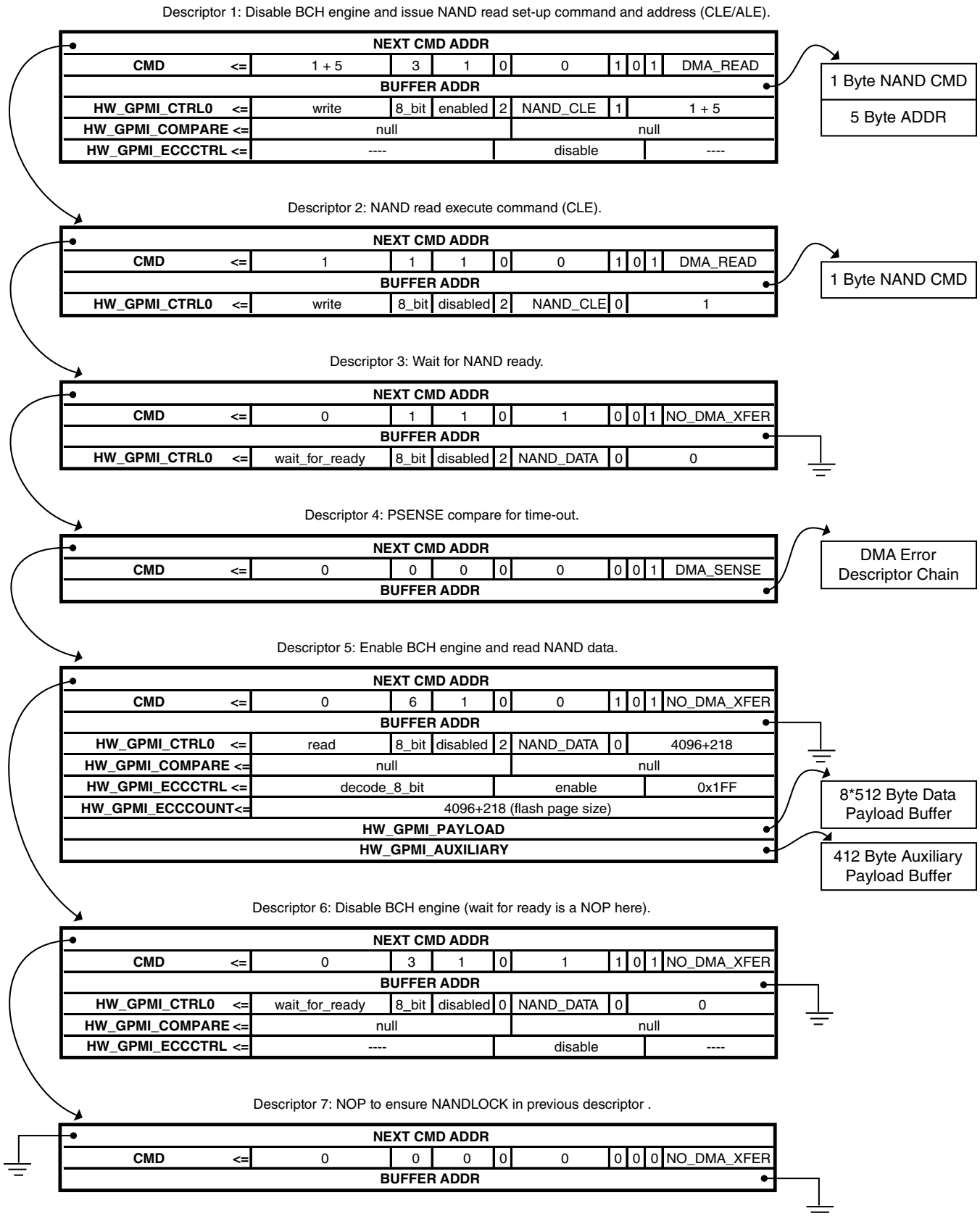


Figure 17-10. BCH Decode DMA Descriptor Chain

17.4.2.1 DMA Structure Code Example

The following sample code illustrates the coding for one read transaction, consisting of a seven DMA command structure chain for reading all 4096 bytes of payload data (eight 512-byte blocks) and 65 bytes of metadata with the associative parity bytes (8 * (18) + 9) from a 4K NAND page sitting on GPMI CS2.

```
//-----------------------------------------------------------------------------
// generic DMA/GPMI/ECC descriptor struct, order sensitive!
//-----------------------------------------------------------------------------
typedef struct {
  // DMA related fields
  unsigned int dma_nxtcmdar;
  unsigned int dma_cmd;
  unsigned int dma_bar;
  // GPMI related fields
  unsigned int gpmi_ctrl0;
  unsigned int gpmi_compare;
  unsigned int gpmi_eccctrl;
  unsigned int gpmi_ecccount;
  unsigned int gpmi_data_ptr;
  unsigned int gpmi_aux_ptr;
} GENERIC_DESCRIPTOR;
//-----------------------------------------------------------------------------
// allocate 7 descriptors for doing a NAND ECC Read
//-----------------------------------------------------------------------------
GENERIC_DESCRIPTOR read[7];
//-----------------------------------------------------------------------------
// DMA descriptor pointer to handle error conditions from psense checks
//-----------------------------------------------------------------------------
unsigned int * dma_error_handler;
//-----------------------------------------------------------------------------
// 7 byte NAND command and address buffer
// any alignment is ok, it is read by the GPMI DMA
//   byte 0 is read setup command
//   bytes 1-5 is the NAND address
//   byte 6 is read execute command
//-----------------------------------------------------------------------------
unsigned char nand_cmd_addr_buffer[7];
//-----------------------------------------------------------------------------
// 4096 byte payload buffer used for reads or writes
// needs to be word aligned
//-----------------------------------------------------------------------------
unsigned int read_payload_buffer[(4096/4)];
//-----------------------------------------------------------------------------
// 412 byte auxiliary buffer used for reads
// needs to be word aligned
//-----------------------------------------------------------------------------
unsigned int read_aux_buffer[(412/4)];
//-----------------------------------------------------------------------------
// Descriptor 1: issue NAND read setup command (CLE/ALE)
//-----------------------------------------------------------------------------
read[0].dma_nxtcmdar = &read[1];                              // point to the next descriptor
read[0].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (1 + 5) |  // 1 byte command, 5 byte address
                  BF_APBH_CHn_CMD_CMDWORDS      (3)     |  // send 3 words to the GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)     |  // wait for command to finish
                                                           //     before continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)     |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(0)     |
                  BF_APBH_CHn_CMD_NANDLOCK      (1)     |  // prevent other DMA channels from
                                                           //    taking over
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)     |
```

```
                BF_APBH_CHn_CMD_CHAIN         (1)      |   // follow chain to next command
                BV_FLD(APBH_CHn_CMD, COMMAND, DMA_READ);  // read data from DMA, write to
NAND
read[0].dma_bar = &nand_cmd_addr_buffer;            // byte 0 read setup, bytes 1 - 5 NAND
address
// 3 words sent to the GPMI
read[0].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)    | // write to the NAND
                     BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)    |
                     BV_FLD(GPMI_CTRL0, LOCK_CS,      ENABLED)  |
                     BF_GPMI_CTRL0_CS                 (2)       | // must correspond to NAND
CS used
                     BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_CLE) |
                     BF_GPMI_CTRL0_ADDRESS_INCREMENT  (1)       | // send command and address
                     BF_GPMI_CTRL0_XFER_COUNT         (1 + 5);    // 1 byte command, 5 byte
address
read[0].gpmi_compare = NULL;                        // field not used but necessary to set
eccctrl
read[0].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, DISABLE); // disable the ECC block
//-------------------------------------------------------------------------
// Descriptor 2: issue NAND read execute command (CLE)
//-------------------------------------------------------------------------
read[1].dma_nxtcmdar = &read[2];                          // point to the next descriptor
read[1].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (1)    |   // 1 byte read command
                  BF_APBH_CHn_CMD_CMDWORDS      (1)    |   // send 1 word to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)    |   // wait for command to finish
before
                                                        //    continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)    |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(0)    |
                  BF_APBH_CHn_CMD_NANDLOCK      (1)    |   // prevent other DMA channels from
                                                        //     taking over
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)    |
                  BF_APBH_CHn_CMD_CHAIN         (1)    |   // follow chain to next command
                  BV_FLD(APBH_CHn_CMD, COMMAND, DMA_READ);     // read data from DMA, write
to NAND
read[1].dma_bar = &nand_cmd_addr_buffer[6];              // point to byte 6, read execute
command
// 1 word sent to the GPMI
read[1].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)    | // write to the NAND
                     BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)    |
                     BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED) |
                     BF_GPMI_CTRL0_CS                 (2)       | // must correspond to NAND
CS used
                     BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_CLE) |
                     BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)       |
                     BF_GPMI_CTRL0_XFER_COUNT         (1);        // 1 byte command
//-------------------------------------------------------------------------
// Descriptor 3: wait for ready (DATA)
//-------------------------------------------------------------------------
read[2].dma_nxtcmdar = &read[3];                          // point to the next descriptor
read[2].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)      | // no dma transfer
                  BF_APBH_CHn_CMD_CMDWORDS      (1)      | // send 1 word to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)      | // wait for command to finish
before
                                                        //     continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)      |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(1)      | // wait for nand to be ready
                  BF_APBH_CHn_CMD_NANDLOCK      (0)      | // relinquish nand lock
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)      |
                  BF_APBH_CHn_CMD_CHAIN         (1)      | // follow chain to next command
                  BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);   // no dma transfer
read[2].dma_bar = NULL;                                    // field not used
// 1 word sent to the GPMI
read[2].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WAIT_FOR_READY) | // wait for NAND
ready
                     BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)          |
                     BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)       |
                     BF_GPMI_CTRL0_CS                 (2)             | // must correspond
to NAND CS used
                     BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA)      |
```

```
                    BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)             |
                    BF_GPMI_CTRL0_XFER_COUNT         (0);
//-----------------------------------------------------------------------------
// Descriptor 4: psense compare (time out check)
//-----------------------------------------------------------------------------
read[3].dma_nxtcmdar = &read[4];                                     // point to the next
descriptor
read[3].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)        |      // no dma transfer
                  BF_APBH_CHn_CMD_CMDWORDS      (0)        |      // no words sent to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (0)        |      // do not wait to continue
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)        |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(0)        |
                  BF_APBH_CHn_CMD_NANDLOCK      (0)        |
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)        |
                  BF_APBH_CHn_CMD_CHAIN         (1)        |      // follow chain to next
command
                  BV_FLD(APBH_CHn_CMD, COMMAND, DMA_SENSE);      // perform a sense check
read[3].dma_bar = dma_error_handler;                        // if sense check fails, branch to
error handler
//-----------------------------------------------------------------------------
// Descriptor 5: read 4K page plus 65 byte meta-data Nand data
//               and send it to ECC block (DATA)
//-----------------------------------------------------------------------------
read[4].dma_nxtcmdar = &read[5];                               // point to the next descriptor
read[4].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)    | // no dma transfer
                  BF_APBH_CHn_CMD_CMDWORDS      (6)    | // send 6 words to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)    | // wait for command to finish before
                                                         //       continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)    |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(0)    |
                  BF_APBH_CHn_CMD_NANDLOCK      (1)    | // prevent other DMA channels from
taking over
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)    | // ECC block generates BCH interrupt
                                                         //      on completion
                  BF_APBH_CHn_CMD_CHAIN         (1)    | // follow chain to next command
                  BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);    // no DMA transfer,
                                                                 // ECC block handles
transfer
read[4].dma_bar = NULL;                                        // field not used
// 6 words sent to the GPMI
read[4].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, READ)       |  // read from the NAND
                     BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)      |
                     BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)   |
                     BF_GPMI_CTRL0_CS                 (2)         |  // must correspond to
NAND CS used
                     BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA)  |
                     BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)         |
                     BF_GPMI_CTRL0_XFER_COUNT         (4096+218);    // eight 512 byte data
blocks
                                                                     // metadata, and parity

read[4].gpmi_compare = NULL;                                // field not used but necessary to set
eccctrl
// GPMI ECCCTRL PIO This launches the 4K byte transfer through BCH's
// bus master. Setting the ECC_ENABLE bit redirects the data flow
// within the GPMI so that read data flows to the BCH engine instead
// of flowing to the GPMI's DMA channel.
read[4].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ECC_CMD,    DECODE_8_BIT) |    // specify t = 8
mode
                       BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, ENABLE)       |    // enable ECC
module
                       BF_GPMI_ECCCTRL_BUFFER_MASK     (0X1FF);  // read all 8 data blocks
and 1 aux block
read[4].gpmi_ecccount = BF_GPMI_ECCCOUNT_COUNT(4096+218);        // specify number of bytes
                                                                 //        read from NAND
read[4].gpmi_data_ptr = &read_payload_buffer;                    // pointer for the 4K byte
                                                                 //    data area
read[4].gpmi_aux_ptr = &read_aux_buffer;                         // pointer for the 65 byte
aux area +
                                                                 // parity and syndrome
```

```
bytes for both
                                                                  // data and aux blocks.
//------------------------------------------------------------------------------
// Descriptor 6: disable ECC block
//------------------------------------------------------------------------------
read[5].dma_nxtcmdar = &read[6];                                  // point to the next descriptor
read[5].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)       | // no dma transfer
                  BF_APBH_CHn_CMD_CMDWORDS      (3)       | // send 3 words to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)       | // wait for command to finish
before
                                                            //     continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)       |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(1)       | // wait for nand to be ready
                  BF_APBH_CHn_CMD_NANDLOCK      (1)       | // need nand lock to be
                                                            // thread safe while turn-off BCH
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)       |
                  BF_APBH_CHn_CMD_CHAIN         (1)       | // follow chain to next command
                  BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);      // no dma transfer
read[5].dma_bar = NULL;                                           // field not used
// 3 words sent to the GPMI
read[5].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, READ)       |
                     BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)      |
                     BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)   |
                     BF_GPMI_CTRL0_CS                 (2)         | // must correspond to
NAND CS used
                     BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA)  |
                     BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)         |
                     BF_GPMI_CTRL0_XFER_COUNT         (0);
read[5].gpmi_compare = NULL;                                      // field not used but necessary to set
eccctrl
read[5].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, DISABLE);  // disable the ECC block
//------------------------------------------------------------------------------
// Descriptor 7: deassert nand lock
//------------------------------------------------------------------------------
read[6].dma_nxtcmdar = NULL;                                      // not used since this is last
descriptor
read[6].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)    |    // no dma transfer
                  BF_APBH_CHn_CMD_CMDWORDS      (0)    |    // no words sent to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (0)    |    // wait for command to finish
before
                                                            //        continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)    |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(0)    |
                  BF_APBH_CHn_CMD_NANDLOCK      (0)    |    // relinquish nand lock
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)    |    // BCH engine generates interrupt
                  BF_APBH_CHn_CMD_CHAIN         (0)    |    // terminate DMA chain processing
                  BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);      // no dma transfer
read[6].dma_bar = NULL;                                           // field not used
```

17.4.2.2 Using the Decoder

As illustrated in Figure 17-10 and the sample code in DMA Structure Code Example :

- DMA descriptor 1 prepares the NAND for data read by using the GPMI to issue a NAND read setup command byte under CLE, then sends a 5-byte address under ALE. The BCH engine is not used for these commands.
- DMA descriptor 2 issues a one-byte read execute command to the NAND device that triggers its read access. The NAND then goes not ready.

- DMA descriptor 3 performs a wait for ready operation allowing the DMA chain to remain dormant until the NAND device completes its read access time.
- DMA descriptor 5 handles the reading and error correction of the NAND data. This command's PIOs activate the BCH engine to write the read NAND data to system memory and to process it for any errors that need to be corrected. This DMA descriptor contains two PIO values that are system memory addresses pointing to the PAYLOAD data area and to the AUXILIARY data area. These addresses are used by the BCH engine's AHB master to move data into system memory and to correct it. While this example is reading an entire 4K page—payload plus metadata—it is equally possible to read just one 512-byte payload block or just the uniquely protected metadata block in a single 7 DMA structure transfer.
- DMA descriptor 6 disables the BCH engine with the NANDLOCK asserted. This is necessary to ensure that the GPMI resource is not arbitrated to another DMA channel when multiple DMA channels are active concurrently.
- DMA descriptor 7 de-asserts the NANDLOCK to free up the GPMI resource to another channel.

As the BCH block receives data from the GPMI:

- The decoder transforms the read NAND data block into a BCH code word and computes the codeword syndrome.
- If no errors are present, then the BCH block can immediately report back to firmware. This report is passed as the BCH_CTRL_COMPLETE_IRQ interrupt status bit and the associated status registers in BCH_STATUS0/1 registers.
- If an error is present, then the BCH block corrects the necessary data block or parity block bytes, if possible (not all errors are correctable).

As the BCH decoder reads the data and parity blocks, it records a special condition, i.e., that all of the bits of a payload data block or metadata block are one, including any associated parity bytes. The all-ones case for both parity and data indicates an erased block in the NAND device.

The BCH_STATUS0 register contains a 4-bit field that indicates the final status of the auxiliary block. A value of 0x0 indicates no errors found for a block.

- A value of 1 to 20 inclusive indicates that many correctable errors were found and fixed.
- A value of 0xFE indicates uncorrectable errors detected on the block.

- A value of 0xFF indicates that the block was in the special ALL ONES state and is therefore considered to be an ERASED block.
- All other values are disallowed by the hardware design.

Recall that up to eight NAND devices can have DMA chains in-flight at once, i.e. they can all be contending for access to the GPMI data bus. It is impossible to predict which NAND device will enter the BCH engine with a transfer first, because each chain includes a wait4ready command structure. As a result, firmware should look at the BCH_STATUS0_COMPLETED_CE bit field to determine which block is being reported in the status register. There is also a 16-bit HANDLE field in the GPMI_ECCCTRL register that is passed down the pipeline with each transaction. This handle field can be used to speed firmware's detection of which transaction is being reported.

These examples of reading and writing have focused on full page transfers of 4K page NAND devices. Other device configurations can be specified by changing the ECCOUNT field in the GPMI registers and reprogramming the BCH's FLASHnLAYOUTm registers.

The BCH and GPMI blocks are designed to be very efficient at reading single 512 (or 1024)-byte pages in one transaction. With no errors, the transaction takes less than 20 HCLKs longer than the time to read the raw data from the NAND.

To summarize, the APBH DMA command chain for a BCH decode operation is shown in Figure 17-10. Seven DMA command structures must be present for each NAND read transaction decoded by the BCH. The seven DMA command structures for multiple NAND read transaction blocks can be chained together to make larger units of work for the BCH, and each will produce an appropriate error report in the BCH PIO space. Multiple NAND devices can have such multiple chains scheduled. The results can come back out of order with respect to the multiple chains.

17.4.3 Interrupts

There are two interrupt sources used in processing BCH protected NAND read and write transfers.

Since all BCH operations are initiated by GPMI DMA command structures, the DMA completion interrupt for the GPMI is an important ISR. Both of the flow charts of Figure 17-6 and Figure 17-9 show the GPMI DMA complete ISR skeleton. In both reads and writes, the GPMI DMA completion interrupt is used to schedule work ***INTO*** the error correction pipeline. As the front end processing completes, the DMA interrupt is

generated and additional work, such as DMA chains, are passed to the GPMI DMA to keep it *fed*. For write operations, this is the only interrupt that is generated for processing the NAND write transfer.

For reads, however, two interrupts are needed. Every read is started by a GPMI DMA command chain and the front end queue is fed as described above. The back end of the read pipeline is drained by monitoring the BCH completion interrupt found int HW_BCH_CTRL_COMPLETE_IRQ.

An BCH transaction consists of reading or writing all of the blocks requested in the HW_GPMI_ECCCTRL_BUFFER_MASK bit field. As every read transaction completes, it posts the status of all of the blocks to the HW_BCH_STATUS0 and HW_BCH_STATUS1 registers and sets the completion interrupt. The five stages of the BCH read pipeline completes, one in the GPMI and four in the BCH, are independently stalled as they complete and try to deliver to the next stage in the data flow. Several of these stages can be skipped if no-errors are found or once an uncorrectable error is found in a block.

In any case, the final stage will stall if the status register is busy waiting for the CPU to take status register results. The hardware monitors the state of the HW_BCH_CTRL_COMPLETE_IRQ bit. If it is still set when the last pipeline stage is ready to post data, then the stage will stall. It follows that the next previous stage will stall when it is ready to hand off work to the final stage, and so on up the pipeline.

CAUTION

It is important that firmware read the STATUS0/1 results and save them before clearing the interrupt request bit. Otherwise, a transaction and its results could be completely lost.

17.5 Behavior During Reset

A soft reset (SFTRST) can take multiple clock periods to complete, so do NOT set CLKGATE when setting SFTRST.

The reset process gates the clocks automatically. The example code is shown below.

```
// A soft reset can take multiple clocks to complete, so do NOT gate the
// clock when setting soft reset. The reset process will gate the clock
// automatically. Poll until this has happened before subsequently
// preparing soft-reset and clock gate
  BCH_CTRL_CLR(BM_BCH_CTRL_SFTRST);
  BCH_CTRL_CLR(BM_BCH_CTRL_CLKGATE);
// asserting soft-reset
  BCH_CTRL_SET(BM_BCH_CTRL_SFTRST);
// waiting for confirmation of soft-reset
  while (!BCH_CTRL.B.CLKGATE)
```

```
  {
// busy wait
  }
// Done.
  BCH_CTRL_CLR(BM_BCH_CTRL_SFTRST);
  BCH_CTRL_CLR(BM_BCH_CTRL_CLKGATE);
```

17.6 BCH Memory Map/Register Definition

BCH Hardware Register Format Summary

BCH memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
11_4000	Hardware BCH ECC Accelerator Control Register (BCH_CTRL)	32	R/W	C000_0000h	17.6.1/746
11_4004	Hardware BCH ECC Accelerator Control Register (BCH_CTRL_SET)	32	R/W	C000_0000h	17.6.1/746
11_4008	Hardware BCH ECC Accelerator Control Register (BCH_CTRL_CLR)	32	R/W	C000_0000h	17.6.1/746
11_400C	Hardware BCH ECC Accelerator Control Register (BCH_CTRL_TOG)	32	R/W	C000_0000h	17.6.1/746
11_4010	Hardware ECC Accelerator Status Register 0 (BCH_STATUS0)	32	R	0000_0010h	17.6.2/749
11_4020	Hardware ECC Accelerator Mode Register (BCH_MODE)	32	R/W	0000_0000h	17.6.3/750
11_4030	Hardware BCH ECC Loopback Encode Buffer Register (BCH_ENCODEPTR)	32	R/W	0000_0000h	17.6.4/751
11_4040	Hardware BCH ECC Loopback Data Buffer Register (BCH_DATAPTR)	32	R/W	0000_0000h	17.6.5/751
11_4050	Hardware BCH ECC Loopback Metadata Buffer Register (BCH_METAPTR)	32	R/W	0000_0000h	17.6.6/752
11_4070	Hardware ECC Accelerator Layout Select Register (BCH_LAYOUTSELECT)	32	R/W	E4E4_E4E4h	17.6.7/752
11_4080	Hardware BCH ECC Flash 0 Layout 0 Register (BCH_FLASH0LAYOUT0)	32	R/W	070A_4080h	17.6.8/754
11_4090	Hardware BCH ECC Flash 0 Layout 1 Register (BCH_FLASH0LAYOUT1)	32	R/W	10DA_4080h	17.6.9/756
11_40A0	Hardware BCH ECC Flash 1 Layout 0 Register (BCH_FLASH1LAYOUT0)	32	R/W	070A_4080h	17.6.10/ 757
11_40B0	Hardware BCH ECC Flash 1 Layout 1 Register (BCH_FLASH1LAYOUT1)	32	R/W	10DA_4080h	17.6.11/ 759
11_40C0	Hardware BCH ECC Flash 2 Layout 0 Register (BCH_FLASH2LAYOUT0)	32	R/W	070A_4080h	17.6.12/ 760
11_40D0	Hardware BCH ECC Flash 2 Layout 1 Register (BCH_FLASH2LAYOUT1)	32	R/W	10DA_4080h	17.6.13/ 762

Table continues on the next page...

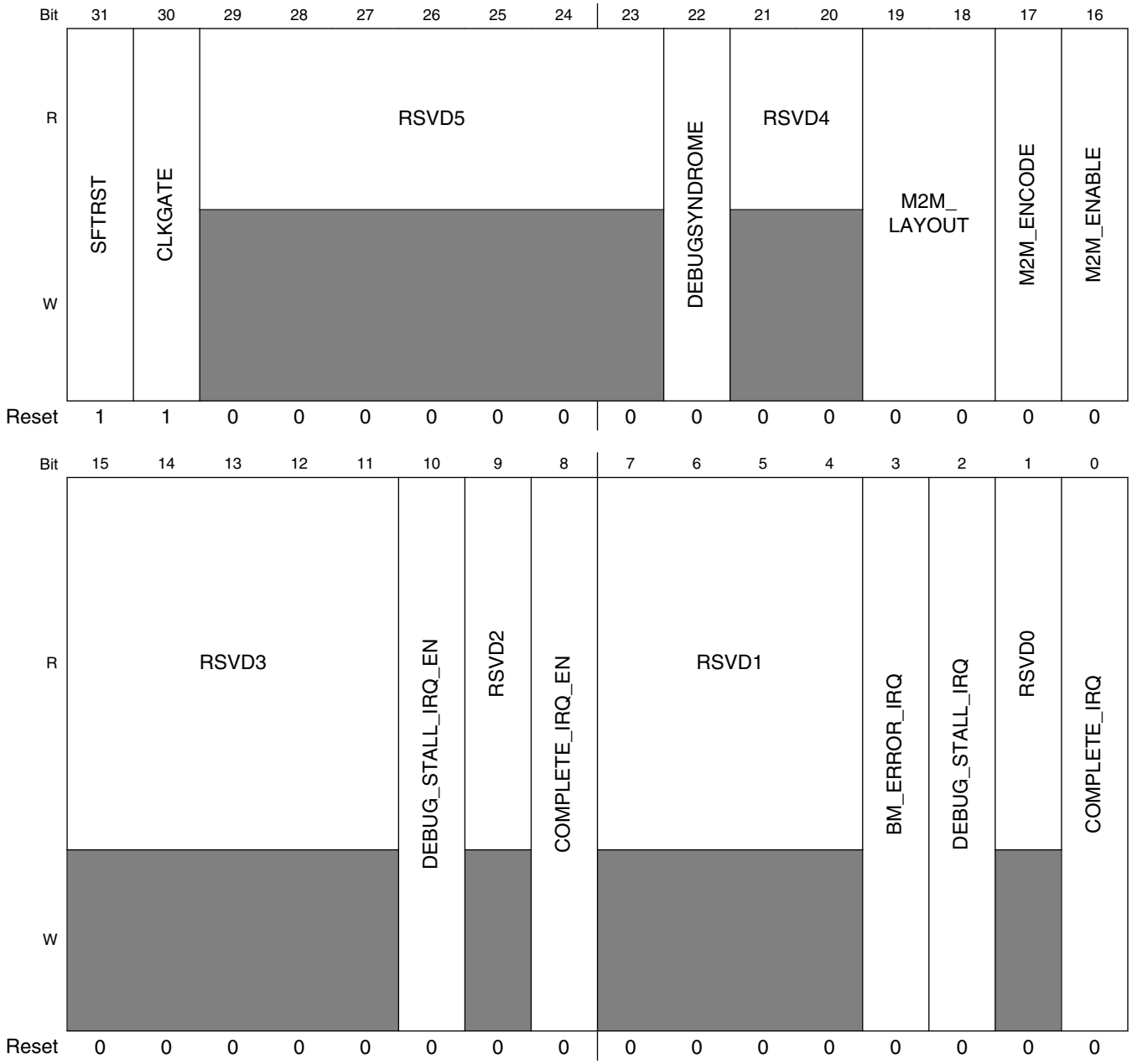
BCH memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
11_40E0	Hardware BCH ECC Flash 3 Layout 0 Register (BCH_FLASH3LAYOUT0)	32	R/W	070A_4080h	17.6.14/ 763
11_40F0	Hardware BCH ECC Flash 3 Layout 1 Register (BCH_FLASH3LAYOUT1)	32	R/W	10DA_4080h	17.6.15/ 765
11_4100	Hardware BCH ECC Debug Register0 (BCH_DEBUG0)	32	R/W	0000_0000h	17.6.16/ 766
11_4104	Hardware BCH ECC Debug Register0 (BCH_DEBUG0_SET)	32	R/W	0000_0000h	17.6.16/ 766
11_4108	Hardware BCH ECC Debug Register0 (BCH_DEBUG0_CLR)	32	R/W	0000_0000h	17.6.16/ 766
11_410C	Hardware BCH ECC Debug Register0 (BCH_DEBUG0_TOG)	32	R/W	0000_0000h	17.6.16/ 766
11_4110	KES Debug Read Register (BCH_DBGKESREAD)	32	R	0000_0000h	17.6.17/ 768
11_4120	Chien Search Debug Read Register (BCH_DBGCSFEREAD)	32	R	0000_0000h	17.6.18/ 769
11_4130	Syndrome Generator Debug Read Register (BCH_DBGSYNDGENREAD)	32	R	0000_0000h	17.6.19/ 769
11_4140	Bus Master and ECC Controller Debug Read Register (BCH_DBGAHBMREAD)	32	R	0000_0000h	17.6.20/ 770
11_4150	Block Name Register (BCH_BLOCKNAME)	32	R	2048_4342h	17.6.21/ 770
11_4160	BCH Version Register (BCH_VERSION)	32	R	0100_0000h	17.6.22/ 771
11_4170	Hardware BCH ECC Debug Register 1 (BCH_DEBUG1)	32	R/W	0000_0000h	17.6.23/ 772

17.6.1 Hardware BCH ECC Accelerator Control Register (BCH_CTRL*n*)

The BCH CTRL provides overall control of the hardware ECC accelerator

Address: 11_4000h base + 0h offset + (4d × i), where i=0d to 3d



BCH_CTRL*n* field descriptions

Field	Description
31 SFTRST	Set this bit to 0 to enable normal BCH operation. Set this bit to 1 (default) to disable clocking with the BCH and hold it in its reset (lowest power) state. This bit can be turned on and then off to reset the BCH block to its default state. This bit resets all state machines except for the AHB master state machine 0x0 **RUN** — Allow BCH to operate normally. 0x1 **RESET** — Hold BCH in reset.
30 CLKGATE	This bit must be set to 0 for normal operation. When set to 1 it gates off the clocks to the block.

Table continues on the next page...

BCH_CTRL*n* field descriptions (continued)

Field	Description
	0x0 **RUN** — Allow BCH to operate normally. 0x1 **NO_CLKS** — Do not clock BCH gates in order to minimize power consumption.
29–23 RSVD5	This field is reserved. This read-only field is reserved and always has the value 0.
22 DEBUGSYNDROME	(For debug purposes only). Enable write of computed syndromes to memory on BCH decode operations. Computed syndromes will be written to the auxiliary buffer after the status block. Syndromes will be written as padded 16-bit values.
21–20 RSVD4	This field is reserved. This read-only field is reserved and always has the value 0
19–18 M2M_LAYOUT	Selects the flash page format for memory-to-memory operations.
17 M2M_ENCODE	Selects encode (parity generation) or decode (correction) mode for memory-to-memory operations.
16 M2M_ENABLE	NOTE! WRITING THIS BIT INITIATES A MEMORY-TO-MEMORY OPERATION. The BCH module must be inactive (not processing data from the GPMI) when this bit is set. The M2M_ENCODE and M2M_LAYOUT bits as well as the ENCODEPTR, DATAPTR, and METAPTR registers are used for memory-to-memory operations and must be correctly programmed before writing this bit.
15–11 RSVD3	This field is reserved. This read-only field is reserved and always has the value 0
10 DEBUG_STALL_ IRQ_EN	1 = interrupt on debug stall mode is enabled. The IRQ is raised on every block
9 RSVD2	This field is reserved. This read-only field is reserved and always has the value 0.
8 COMPLETE_IRQ_ EN	1 = interrupt on completion of correction is enabled.
7–4 RSVD1	This field is reserved. This read-only field is reserved and always has the value 0.
3 BM_ERROR_IRQ	AHB Bus interface Error Interrupt Status. Write a 1 to the SCT clear address to clear the interrupt status bit.
2 DEBUG_STALL_IRQ	DEBUG STALL Interrupt Status. Write a 1 to the SCT clear address to clear the interrupt status bit.
1 RSVD0	This field is reserved. This read-only field is reserved and always has the value 0.
0 COMPLETE_IRQ	This bit indicates the state of the external interrupt line. Write a 1 to the SCT clear address to clear the interrupt status bit. NOTE: subsequent ECC completions will be held off as long as this bit is set. Be sure to read the data from BCH_STATUS0, 1 before clearing this interrupt bit.

17.6.2 Hardware ECC Accelerator Status Register 0 (BCH_STATUS0)

The BCH STAT register provides visibility into the run-time status of the BCH and status information when processing is complete. It provides overall status of the hardware ECC accelerator.

Address: 11_4000h base + 10h offset = 11_4010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HANDLE												COMPLETED_CE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STATUS_BLK0								RSVD1			ALLONES	CORRECTED	UNCORRECTABLE	RSVD0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

BCH_STATUS0 field descriptions

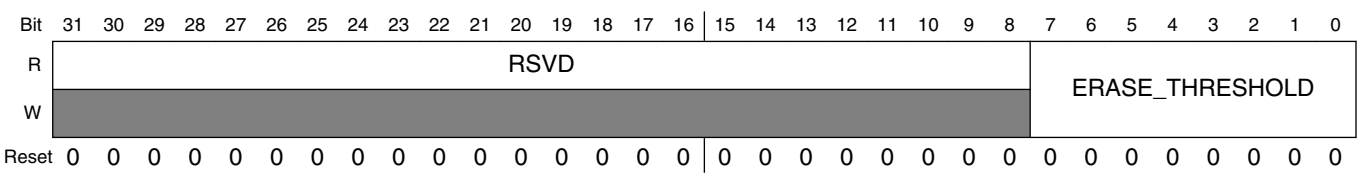
Field	Description
31–20 HANDLE	Software supplies a 12 bit handle for this transfer as part of the GPMI DMA PIO operation that started the transaction. That handle passes down the pipeline and ends up here at the time the BCH interrupt is signaled.
19–16 COMPLETED_CE	This is the chip enable number corresponding to the NAND device from which this data came.
15–8 STATUS_BLK0	Count of symbols in error during processing of first block of flash (metadata block). The number of errors reported will be in the range of 0 to the ECC correction level for block 0. 0x00 **ZERO** — No errors found on block. 0x01 **ERROR1** — One error found on block. 0x02 **ERROR2** — One errors found on block. 0x03 **ERROR3** — One errors found on block. 0x04 **ERROR4** — One errors found on block. 0xFE **UNCORRECTABLE** — Block exhibited uncorrectable errors. 0xFF **ERASED** — Page is erased.
7–5 RSVD1	This field is reserved. This read-only field is reserved and always has the value 0.
4 ALLONES	1 = All data bits of this transaction are ONE.
3 CORRECTED	1 = At least one correctable error encountered during last processing cycle.
2 UNCORRECTABLE	1 = Uncorrectable error encountered during last processing cycle.
RSVD0	This field is reserved. This read-only field is reserved and always has the value 0.

17.6.3 Hardware ECC Accelerator Mode Register (BCH_MODE)

The BCH MODE register provides additional mode controls.

Contains additional global mode controls for the BCH engine.

Address: 11_4000h base + 20h offset = 11_4020h



BCH_MODE field descriptions

Field	Description
31–8 RSVD	This field is reserved.

Table continues on the next page...

BCH_MODE field descriptions (continued)

Field	Description
	This read-only field is reserved and always has the value 0.
ERASE_ THRESHOLD	This value indicates the maximum number of zero bits on a flash subpage for it to be considered erased. For SLC NAND devices, this value should be programmed to 0 (meaning that the entire page should consist of bytes of 0xFF. For MLC NAND devices, bit errors may occur on reads (even on blank pages), so this threshold can be used to tune the erased page checking algorithm.

17.6.4 Hardware BCH ECC Loopback Encode Buffer Register (BCH_ENCODEPTR)

When performing memory to memory operations, indicates the address of the encode buffer. This register should be programmed before writing a 1 to the M2M_ENABLE bit in the CTRL register.

For memory to memory operations, this register is used as the pointer to the encoded data, which is an output when encoding and an input while decoding.

Address: 11_4000h base + 30h offset = 11_4030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BCH_ENCODEPTR field descriptions

Field	Description
ADDR	Address pointer to encode buffer. This is the source for decode operations and the destination for encode operations. This value must be aligned on a 4 bytes boundary.

17.6.5 Hardware BCH ECC Loopback Data Buffer Register (BCH_DATAPTR)

When performing memory to memory operations, indicates the address of the data buffer.

For memory to memory operations, this register is used as the pointer to the data to encode or the destination buffer for decode operations.

Address: 11_4000h base + 40h offset = 11_4040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BCH_DATAPTR field descriptions

Field	Description
ADDR	Address pointer to data buffer. This is the source for encode operations and the destination for decode operations. This register should be programmed before writing a 1 to the M2M_ENABLE bit in the CTRL register. This value must be aligned on a 4 byte boundary.

17.6.6 Hardware BCH ECC Loopback Metadata Buffer Register (BCH_METAPTR)

When performing memory to memory operations, indicates the address of the metadata buffer.

For memory to memory operations, this register is used as the pointer to the metadata to encode or the extracted metadata for decode operations.

Address: 11_4000h base + 50h offset = 11_4050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BCH_METAPTR field descriptions

Field	Description
ADDR	Address pointer to metadata buffer. This is the source for encode metadata read operations and the destination for metadata decode operations. This register should be programmed before writing a 1 to the M2M_ENABLE bit in the CTRL register. This value must be aligned on a 4 bytes boundary.

17.6.7 Hardware ECC Accelerator Layout Select Register (BCH_LAYOUTSELECT)

The BCH LAYOUTSELECT register provides a mapping of chip selects to layout registers.

When the BCH engine receives a request to process a data block from the GPMI interface, it will use this register to map the incoming chip select to one of the four possible flash layout registers

Address: 11_4000h base + 70h offset = 11_4070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CS15_SELECT		CS14_SELECT		CS13_SELECT		CS12_SELECT		CS11_SELECT		CS10_SELECT		CS9_SELECT		CS8_SELECT	
Reset	1	1	1	0	0	1	0	0	1	1	1	0	0	1	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CS7_SELECT		CS6_SELECT		CS5_SELECT		CS4_SELECT		CS3_SELECT		CS2_SELECT		CS1_SELECT		CS0_SELECT	
Reset	1	1	1	0	0	1	0	0	1	1	1	0	0	1	0	0

BCH_LAYOUTSELECT field descriptions

Field	Description
31–30 CS15_SELECT	Selects which layout is used for chip select 15.
29–28 CS14_SELECT	Selects which layout is used for chip select 14.
27–26 CS13_SELECT	Selects which layout is used for chip select 13.
25–24 CS12_SELECT	Selects which layout is used for chip select 12.
23–22 CS11_SELECT	Selects which layout is used for chip select 11.
21–20 CS10_SELECT	Selects which layout is used for chip select 10.
19–18 CS9_SELECT	Selects which layout is used for chip select 9.
17–16 CS8_SELECT	Selects which layout is used for chip select 8.
15–14 CS7_SELECT	Selects which layout is used for chip select 7.
13–12 CS6_SELECT	Selects which layout is used for chip select 6.
11–10 CS5_SELECT	Selects which layout is used for chip select 5.
9–8 CS4_SELECT	Selects which layout is used for chip select 4.
7–6 CS3_SELECT	Selects which layout is used for chip select 3.
5–4 CS2_SELECT	Selects which layout is used for chip select 2.
3–2 CS1_SELECT	Selects which layout is used for chip select 1.
CS0_SELECT	Selects which layout is used for chip select 0.

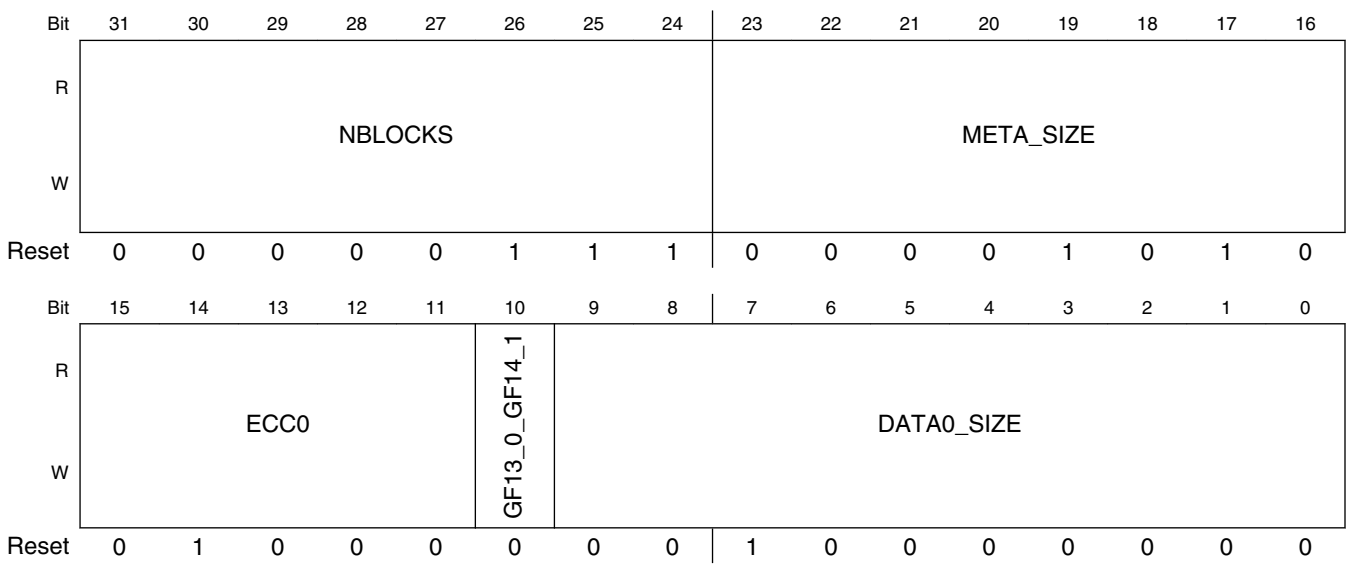
17.6.8 Hardware BCH ECC Flash 0 Layout 0 Register (BCH_FLASH0LAYOUT0)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH0LAYOUT1 register to control the format for the devices selecting layout 0 in the LAYOUTSELECT register.

Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings that are used by the BCH block while reading / writing the flash page to control data, metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks.

See sections Flash Page Layout and Determining the ECC layout for a device for more detail information on setting up the flash layout registers.

Address: 11_4000h base + 80h offset = 11_4080h



BCH_FLASH0LAYOUT0 field descriptions

Field	Description
31–24 NBLOCKS	Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that 8 subsequent blocks are present for a total of 9 blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.

Table continues on the next page...

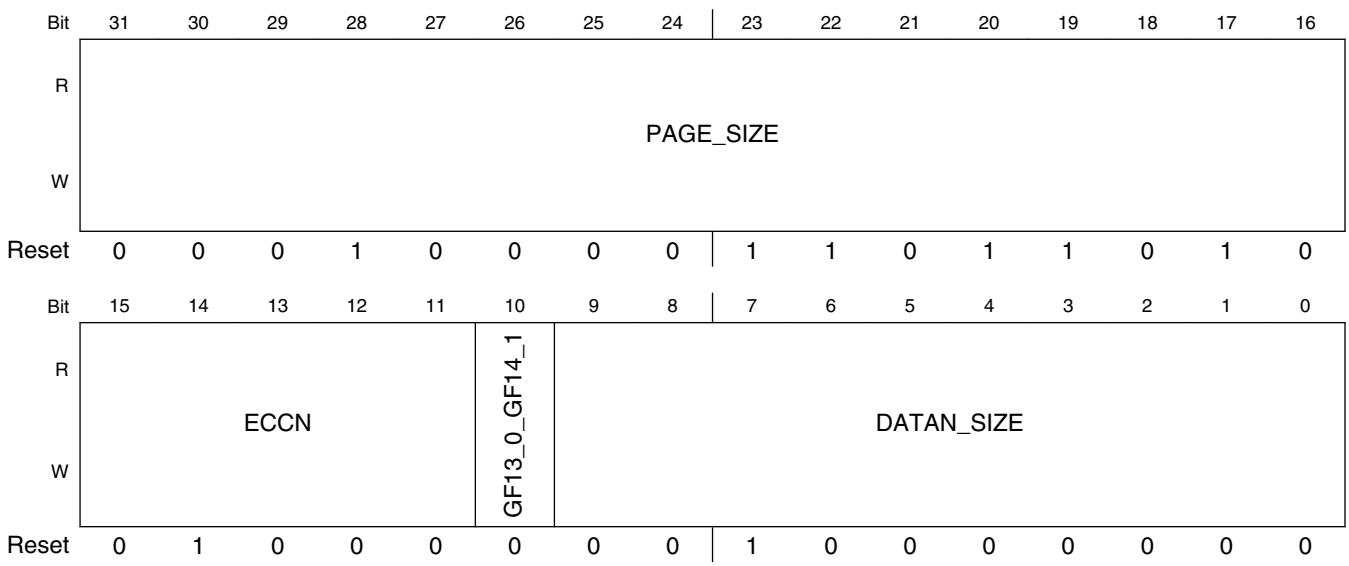
BCH_FLASH0LAYOUT0 field descriptions (continued)

Field	Description
23–16 META_SIZE	Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design support from 0 to 255 bytes for metadata—if set to 0, no metadata will be stored. Metadata is stored before the associated data in block 0. If the DATA0_SIZE field is programmed to a 0, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and covered by a single parity block.
15–11 ECC0	Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field. 0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed 0x3 **ECC6** — ECC 6 to be performed 0x4 **ECC8** — ECC 8 to be performed 0x5 **ECC10** — ECC 10 to be performed 0x6 **ECC12** — ECC 12 to be performed 0x7 **ECC14** — ECC 14 to be performed 0x8 **ECC16** — ECC 16 to be performed 0x9 **ECC18** — ECC 18 to be performed 0xA **ECC20** — ECC 20 to be performed 0xB **ECC22** — ECC 22 to be performed 0xC **ECC24** — ECC 24 to be performed 0xD **ECC26** — ECC 26 to be performed 0xE **ECC28** — ECC 28 to be performed 0xF **ECC30** — ECC 30 to be performed 0x10 **ECC32** — ECC 32 to be performed 0x11 **ECC34** — ECC 34 to be performed 0x12 **ECC36** — ECC 36 to be performed 0x13 **ECC38** — ECC 38 to be performed 0x14 **ECC40** — ECC 40 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATA0_SIZE	Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to 0, the first block only contains metadata.

17.6.9 Hardware BCH ECC Flash 0 Layout 1 Register (BCH_FLASH0LAYOUT1)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH0LAYOUT0 register to control the format for the device selecting layout 0 in the LAYOUTSELECT register.

Address: 11_4000h base + 90h offset = 11_4090h



BCH_FLASH0LAYOUT1 field descriptions

Field	Description
31–16 PAGE_SIZE	Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accomodate different flash configurations that may be available in the future.
15–11 ECCN	Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata). 0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed 0x3 **ECC6** — ECC 6 to be performed 0x4 **ECC8** — ECC 8 to be performed 0x5 **ECC10** — ECC 10 to be performed 0x6 **ECC12** — ECC 12 to be performed 0x7 **ECC14** — ECC 14 to be performed 0x8 **ECC16** — ECC 16 to be performed 0x9 **ECC18** — ECC 18 to be performed 0xA **ECC20** — ECC 20 to be performed

Table continues on the next page...

BCH_FLASH0LAYOUT1 field descriptions (continued)

Field	Description
	0xB **ECC22** — ECC 22 to be performed 0xC **ECC24** — ECC 24 to be performed 0xD **ECC26** — ECC 26 to be performed 0xE **ECC28** — ECC 28 to be performed 0xF **ECC30** — ECC 30 to be performed 0x10 **ECC32** — ECC 32 to be performed 0x11 **ECC34** — ECC 34 to be performed 0x12 **ECC36** — ECC 36 to be performed 0x13 **ECC38** — ECC 38 to be performed 0x14 **ECC40** — ECC 40 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATAN_SIZE	Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page. The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

17.6.10 Hardware BCH ECC Flash 1 Layout 0 Register (BCH_FLASH1LAYOUT0)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH1LAYOUT1 register to control the format for the devices selecting layout 1 in the LAYOUTSELECT register.

Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings that are used by the BCH block while reading / writing the flash page to control data, metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks.

See sections Flash Page Layout and Determining the ECC layout for a device for more detail information on setting up the flash layout registers.

Address: 11_4000h base + A0h offset = 11_40A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	NBLOCKS								META_SIZE							
W																
Reset	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ECC0					GF13_0_GF14_1	DATA0_SIZE									
W																
Reset	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0

BCH_FLASH1LAYOUT0 field descriptions

Field	Description
31–24 NBLOCKS	Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that 8 subsequent blocks are present for a total of 9 blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.
23–16 META_SIZE	Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design supports from 0 to 255 bytes for metadata—if set to 0, no metadata will be stored. Metadata is stored before the associated data is in block 0. If the DATA0_SIZE field is programmed to a 0, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and covered by a single parity block.
15–11 ECC0	Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field. 0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed 0x3 **ECC6** — ECC 6 to be performed 0x4 **ECC8** — ECC 8 to be performed 0x5 **ECC10** — ECC 10 to be performed 0x6 **ECC12** — ECC 12 to be performed 0x7 **ECC14** — ECC 14 to be performed 0x8 **ECC16** — ECC 16 to be performed 0x9 **ECC18** — ECC 18 to be performed 0xA **ECC20** — ECC 20 to be performed 0xB **ECC22** — ECC 22 to be performed 0xC **ECC24** — ECC 24 to be performed 0xD **ECC26** — ECC 26 to be performed 0xE **ECC28** — ECC 28 to be performed 0xF **ECC30** — ECC 30 to be performed 0x10 **ECC32** — ECC 32 to be performed 0x11 **ECC34** — ECC 34 to be performed

Table continues on the next page...

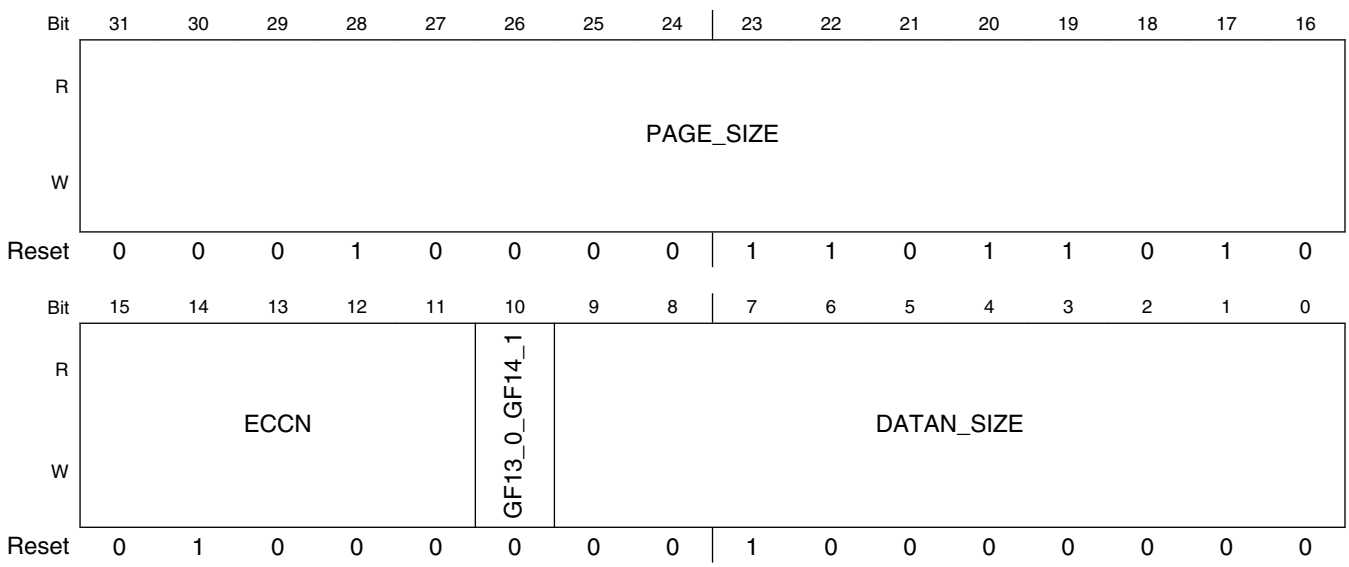
BCH_FLASH1LAYOUT0 field descriptions (continued)

Field	Description
	0x12 **ECC36** — ECC 36 to be performed 0x13 **ECC38** — ECC 38 to be performed 0x14 **ECC40** — ECC 40 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATA0_SIZE	Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to 0, the first block will contains metadata.

17.6.11 Hardware BCH ECC Flash 1 Layout 1 Register (BCH_FLASH1LAYOUT1)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH1LAYOUT0 register to control the format for the device selecting layout 1 in the LAYOUTSELECT register.

Address: 11_4000h base + B0h offset = 11_40B0h



BCH_FLASH1LAYOUT1 field descriptions

Field	Description
31–16 PAGE_SIZE	Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accomodate different flash configurations that may be available in the future.
15–11 ECCN	Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata).

Table continues on the next page...

BCH_FLASH1LAYOUT1 field descriptions (continued)

Field	Description
	0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed 0x3 **ECC6** — ECC 6 to be performed 0x4 **ECC8** — ECC 8 to be performed 0x5 **ECC10** — ECC 10 to be performed 0x6 **ECC12** — ECC 12 to be performed 0x7 **ECC14** — ECC 14 to be performed 0x8 **ECC16** — ECC 16 to be performed 0x9 **ECC18** — ECC 18 to be performed 0xA **ECC20** — ECC 20 to be performed 0xB **ECC22** — ECC 22 to be performed 0xC **ECC24** — ECC 24 to be performed 0xD **ECC26** — ECC 26 to be performed 0xE **ECC28** — ECC 28 to be performed 0xF **ECC30** — ECC 30 to be performed 0x10 **ECC32** — ECC 32 to be performed 0x11 **ECC34** — ECC 34 to be performed 0x12 **ECC36** — ECC 36 to be performed 0x13 **ECC38** — ECC 38 to be performed 0x14 **ECC40** — ECC 40 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATAN_SIZE	Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page. The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

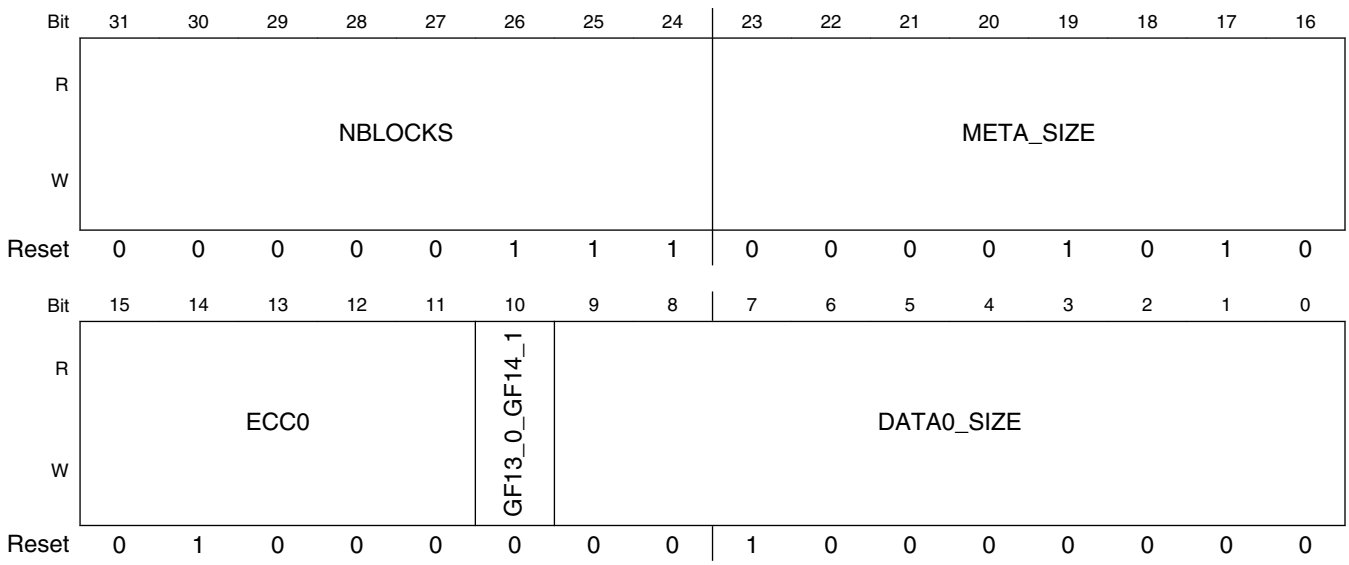
17.6.12 Hardware BCH ECC Flash 2 Layout 0 Register (BCH_FLASH2LAYOUT0)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH2LAYOUT1 register to control the format for the devices selecting layout 2 in the LAYOUTSELECT register.

Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings that are used by the BCH block while reading / writing the flash page to control data, metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks.

See sections Flash Page Layout and Determining the ECC layout for a device for more detail information on setting up the flash layout registers.

Address: 11_4000h base + C0h offset = 11_40C0h



BCH_FLASH2LAYOUT0 field descriptions

Field	Description
31–24 NBLOCKS	Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that eight subsequent blocks are present for a total of nine blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.
23–16 META_SIZE	Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design support from 0 to 255 bytes for metadata—if set to 0, no metadata will be stored. Metadata is stored before the associated data in block 0. If the DATA0_SIZE field is programmed to a 0, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and will be covered by a single parity block.
15–11 ECC0	Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field. 0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed 0x3 **ECC6** — ECC 6 to be performed 0x4 **ECC8** — ECC 8 to be performed 0x5 **ECC10** — ECC 10 to be performed 0x6 **ECC12** — ECC 12 to be performed 0x7 **ECC14** — ECC 14 to be performed 0x8 **ECC16** — ECC 16 to be performed 0x9 **ECC18** — ECC 18 to be performed 0xA **ECC20** — ECC 20 to be performed 0xB **ECC22** — ECC 22 to be performed 0xC **ECC24** — ECC 24 to be performed 0xD **ECC26** — ECC 26 to be performed

Table continues on the next page...

BCH_FLASH2LAYOUT0 field descriptions (continued)

Field	Description
	0xE **ECC28** — ECC 28 to be performed 0xF **ECC30** — ECC 30 to be performed 0x10 **ECC32** — ECC 32 to be performed 0x11 **ECC34** — ECC 34 to be performed 0x12 **ECC36** — ECC 36 to be performed 0x13 **ECC38** — ECC 38 to be performed 0x14 **ECC40** — ECC 40 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATA0_SIZE	Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to 0, the first block will only contain metadata.

17.6.13 Hardware BCH ECC Flash 2 Layout 1 Register (BCH_FLASH2LAYOUT1)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH2LAYOUT0 register to control the format for the device selecting layout 2 in the LAYOUTSELECT register.

Address: 11_4000h base + D0h offset = 11_40D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PAGE_SIZE															
Reset	0	0	0	1	0	0	0	0	1	1	0	1	1	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECCN					GF13_0_GF14_1	DATAN_SIZE									
Reset	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0

BCH_FLASH2LAYOUT1 field descriptions

Field	Description
31–16 PAGE_SIZE	Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accomodate different flash configurations that may be available in the future.
15–11 ECCN	Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata). 0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed 0x3 **ECC6** — ECC 6 to be performed 0x4 **ECC8** — ECC 8 to be performed 0x5 **ECC10** — ECC 10 to be performed 0x6 **ECC12** — ECC 12 to be performed 0x7 **ECC14** — ECC 14 to be performed 0x8 **ECC16** — ECC 16 to be performed 0x9 **ECC18** — ECC 18 to be performed 0xA **ECC20** — ECC 20 to be performed 0xB **ECC22** — ECC 22 to be performed 0xC **ECC24** — ECC 24 to be performed 0xD **ECC26** — ECC 26 to be performed 0xE **ECC28** — ECC 28 to be performed 0xF **ECC30** — ECC 30 to be performed 0x10 **ECC32** — ECC 32 to be performed 0x11 **ECC34** — ECC 34 to be performed 0x12 **ECC36** — ECC 36 to be performed 0x13 **ECC38** — ECC 38 to be performed 0x14 **ECC40** — ECC 40 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATAN_SIZE	Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page. The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

17.6.14 Hardware BCH ECC Flash 3 Layout 0 Register (BCH_FLASH3LAYOUT0)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH3LAYOUT1 register to control the format for the devices selecting layout 3 in the LAYOUTSELECT register.

Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings that are used by the BCH block while reading / writing the flash page to control data,

metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks.

See sections Flash Page Layout and Determining the ECC layout for a device for more detail information on setting up the flash layout registers.

Address: 11_4000h base + E0h offset = 11_40E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NBLOCKS								META_SIZE							
Reset	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECC0					GF13_0_GF14_1	DATA0_SIZE									
Reset	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0

BCH_FLASH3LAYOUT0 field descriptions

Field	Description
31–24 NBLOCKS	Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that 8 subsequent blocks are present for a total of 9 blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.
23–16 META_SIZE	Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design support from 0 to 255 bytes for metadata—if set to 0, no metadata will be stored. Metadata is stored before the associated data in block 0. If the DATA0_SIZE field is programmed to a 0, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and will be covered by a single parity block.
15–11 ECC0	Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field. 0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed 0x3 **ECC6** — ECC 6 to be performed 0x4 **ECC8** — ECC 8 to be performed 0x5 **ECC10** — ECC 10 to be performed 0x6 **ECC12** — ECC 12 to be performed 0x7 **ECC14** — ECC 14 to be performed

Table continues on the next page...

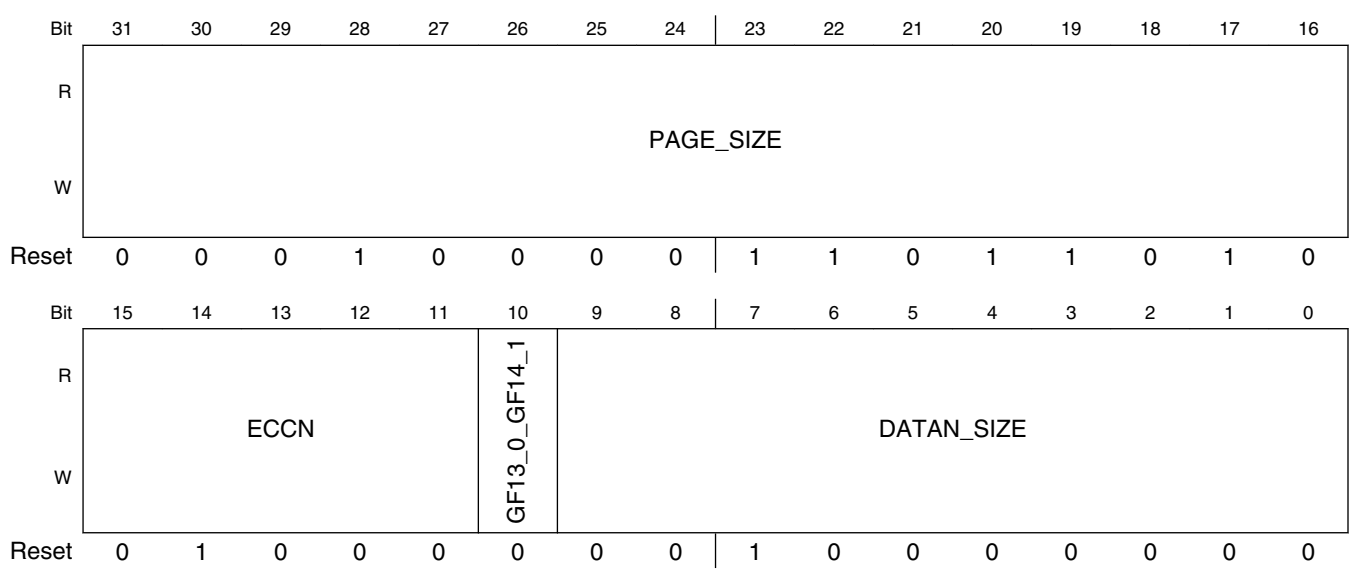
BCH_FLASH3LAYOUT0 field descriptions (continued)

Field	Description
	0x8 **ECC16** — ECC 16 to be performed 0x9 **ECC18** — ECC 18 to be performed 0xA **ECC20** — ECC 20 to be performed 0xB **ECC22** — ECC 22 to be performed 0xC **ECC24** — ECC 24 to be performed 0xD **ECC26** — ECC 26 to be performed 0xE **ECC28** — ECC 28 to be performed 0xF **ECC30** — ECC 30 to be performed 0x10 **ECC32** — ECC 32 to be performed 0x11 **ECC34** — ECC 34 to be performed 0x12 **ECC36** — ECC 36 to be performed 0x13 **ECC38** — ECC 38 to be performed 0x14 **ECC40** — ECC 40 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATA0_SIZE	Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to 0, the first block will only contain metadata.

17.6.15 Hardware BCH ECC Flash 3 Layout 1 Register (BCH_FLASH3LAYOUT1)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH3LAYOUT0 register to control the format for the device selecting layout 3 in the LAYOUTSELECT register.

Address: 11_4000h base + F0h offset = 11_40F0h



BCH_FLASH3LAYOUT1 field descriptions

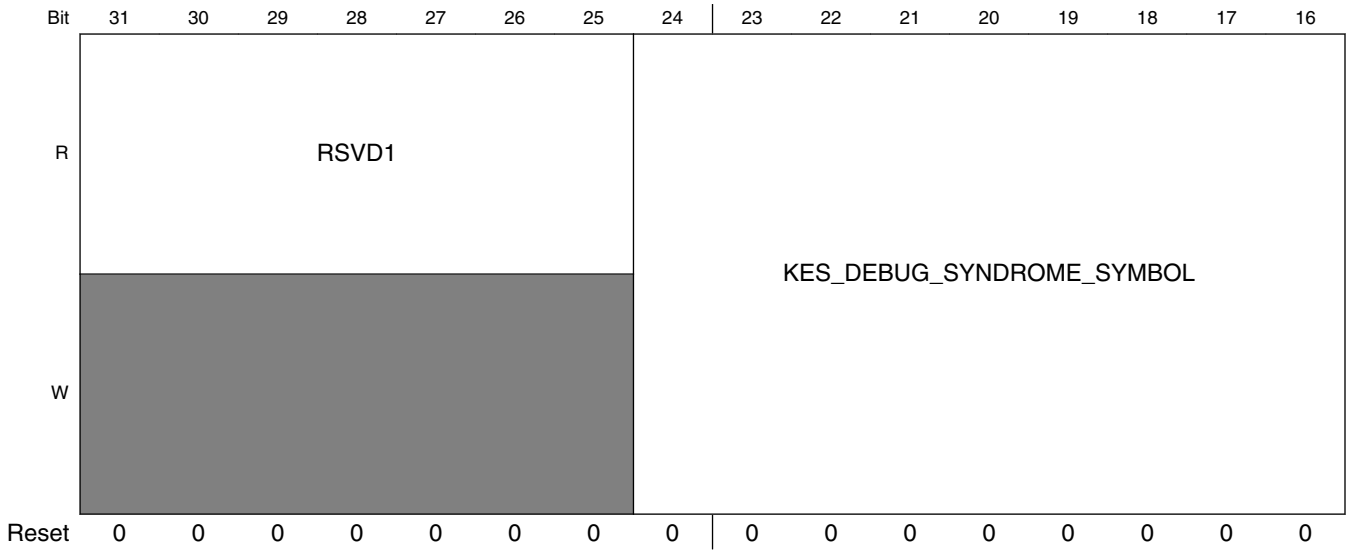
Field	Description
31–16 PAGE_SIZE	Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accomodate different flash configurations that may be available in the future.
15–11 ECCN	Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata). 0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed 0x3 **ECC6** — ECC 6 to be performed 0x4 **ECC8** — ECC 8 to be performed 0x5 **ECC10** — ECC 10 to be performed 0x6 **ECC12** — ECC 12 to be performed 0x7 **ECC14** — ECC 14 to be performed 0x8 **ECC16** — ECC 16 to be performed 0x9 **ECC18** — ECC 18 to be performed 0xA **ECC20** — ECC 20 to be performed 0xB **ECC22** — ECC 22 to be performed 0xC **ECC24** — ECC 24 to be performed 0xD **ECC26** — ECC 26 to be performed 0xE **ECC28** — ECC 28 to be performed 0xF **ECC30** — ECC 30 to be performed 0x10 **ECC32** — ECC 32 to be performed 0x11 **ECC34** — ECC 34 to be performed 0x12 **ECC36** — ECC 36 to be performed 0x13 **ECC38** — ECC 38 to be performed 0x14 **ECC40** — ECC 40 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATAN_SIZE	Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page. The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

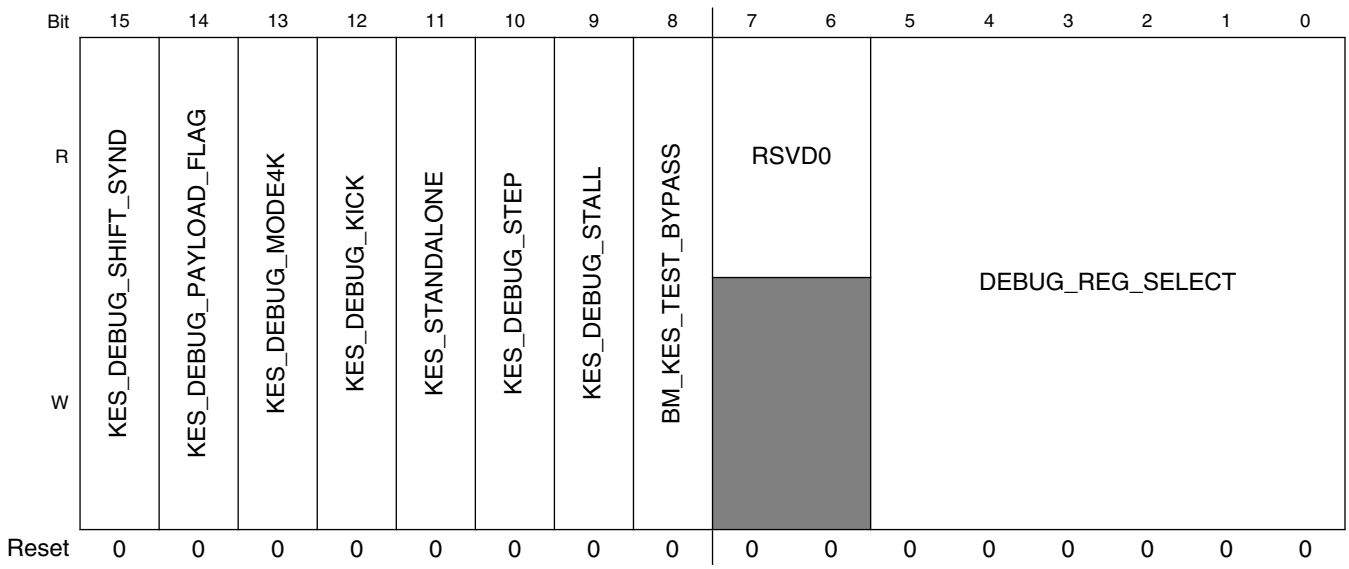
17.6.16 Hardware BCH ECC Debug Register0 (BCH_DEBUG0*n*)

The hardware BCH accelerator internal state machines and signals can be seen in the ECC debug register.

The BCH_DEBUG0 register provides access to various internal state information which might prove useful during hardware debug and validation.

Address: 11_4000h base + 100h offset + (4d × i), where i=0d to 3d





BCH_DEBUG0*n* field descriptions

Field	Description
31–25 RSVD1	This field is reserved. This read-only field is reserved and always has the value 0.
24–16 KES_DEBUG_ SYNDROME_ SYMBOL	The 9 bit value in this bit field shifts into the syndrome register array at the input of the KES engine whenever BCH_DEBUG0_KES_DEBUG_SHIFT_SYND is toggled. 0x0 **NORMAL** — Bus master address generator for SYND_GEN writes operates normally. 0x1 **TEST_MODE** — Bus master address generator always addresses last four bytes in Auxiliary block.
15 KES_DEBUG_ SHIFT_SYND	Toggling this bit causes the value in BCH_DEBUG0_KES_SYNDROME_SYMBOL to be shift into the syndrome register array at the input to the KES engine. After shifting in 16 symbols, one can kick off both KES and CF cycles by toggling BCH_DEBUG0_KES_DEBUG_KICK. Make sure that set KES_BCH_DEBUG0_KES_STANDALONE mode to 1 before kicking.

Table continues on the next page...

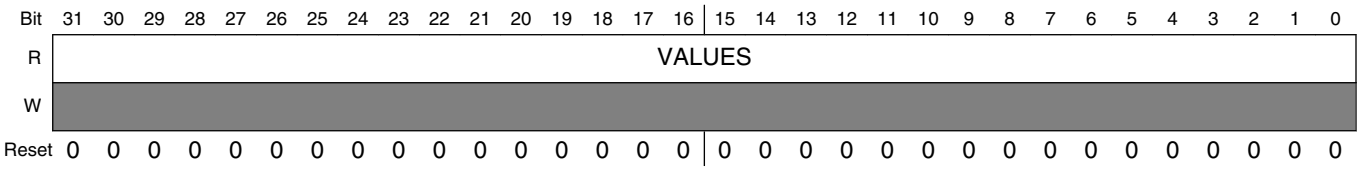
BCH_DEBUG0*n* field descriptions (continued)

Field	Description
14 KES_DEBUG_ PAYLOAD_FLAG	When running the stand alone debug mode on the error calculator, the state of this bit is presented to the KES engine as the input payload flag. 0x1 **DATA** — Payload is set for 512 bytes data block. 0x1 **AUX** — Payload is set for 65 or 19 bytes auxiliary block.
13 KES_DEBUG_ MODE4K	When running the stand alone debug mode on the error calculator, the state of this bit is presented to the KES engine as the input mode (4K or 2K pages). 0x1 **4k** — Mode is set for 4K NAND pages. 0x1 **2k** — Mode is set for 2K NAND pages.
12 KES_DEBUG_ KICK	Toggling causes KES engine FSM to start as if kick by the Bus Master. This allows stand alone testing of the KES and Chien Search engines. Be sure to set KES_BCH_DEBUG0_KES_STANDALONE mode to 1 before kicking.
11 KES_ STANDALONE	Set to one, cause the KES engine to suppress toggling the KES_BM_DONE signal to the bus master and suppress toggling the CF_BM_DONE signal by the CF engine. 0x0 **NORMAL** — Bus master address generator for SYND_GEN writes operates normally. 0x1 **TEST_MODE** — Bus master address generator always addresses last four bytes in Auxiliary block.
10 KES_DEBUG_ STEP	Toggling this bit causes the KES FSM to skip passed the stall state if it is in DEBUG_STALL mode and completed processing a block.
9 KES_DEBUG_ STALL	Set to one to cause KES FSM to stall after notifying Chien search engine to start processing its block but before notifying the bus master that the KES computation is complete. This allows a diagnostic to stall the FSM after each blocks key equations are solved. This also has the effect of stalling the CSFE search engine so it's state can be examined after it finishes processing the KES stalled block. 0x0 **NORMAL** — KES FSM proceeds to next block supplied by bus master. 0x1 **WAIT** — KES FSM waits after current equations are solved and the search engine is started.
8 BM_KES_TEST_ BYPASS	1 = Point all SYND_GEN writes to dummy area at the end of the AUXILLIARY block so that diagnostics can preload all payload, parity bytes and computed syndrome bytes for test the KES engine. 0x0 **NORMAL** — Bus master address generator for SYND_GEN writes operates normally. 0x1 **TEST_MODE** — Bus master address generator always addresses last four bytes in Auxiliary block.
7–6 RSVD0	This field is reserved. This read-only field is reserved and always has the value 0.
DEBUG_REG_ SELECT	The value loaded in this bit field is used to select the internal register state view of KES engine or the Chien search engine.

17.6.17 KES Debug Read Register (BCH_DBGKESREAD)

The hardware BCH ECC accelerator key equation solver internal state machines and signals can be seen in the ECC debug registers.

Address: 11_4000h base + 110h offset = 11_4110h



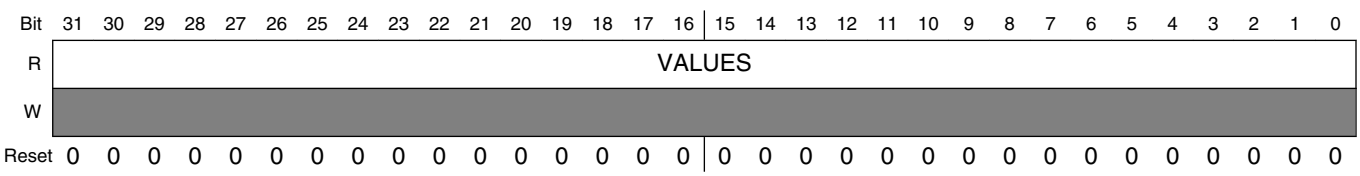
BCH_DBGKESREAD field descriptions

Field	Description
VALUES	This register returns the ROM BIST CRC value after a BIST test.

17.6.18 Chien Search Debug Read Register (BCH_DBGCSFEREAD)

The hardware BCH ECC accelerator Chien Search internal state machines and signals can be seen in the ECC debug registers.

Address: 11_4000h base + 120h offset = 11_4120h



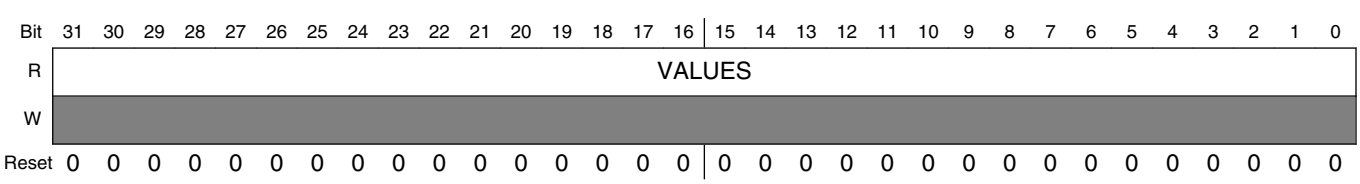
BCH_DBGCSFEREAD field descriptions

Field	Description
VALUES	Reserved

17.6.19 Syndrome Generator Debug Read Register (BCH_DBGSYNDGENREAD)

The hardware BCH ECC accelerator syndrome generator internal state machines and signals can be seen in the ECC debug registers.

Address: 11_4000h base + 130h offset = 11_4130h



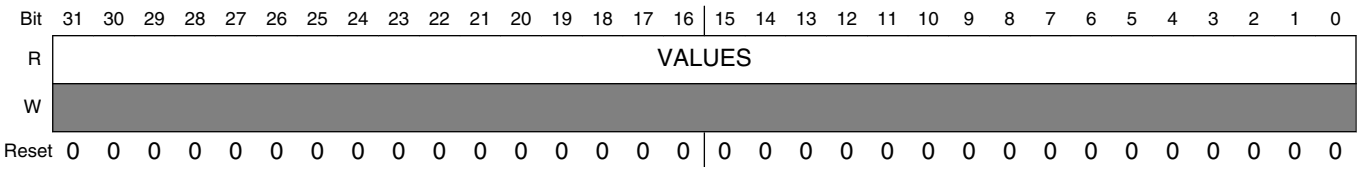
BCH_DBGSYNDGENREAD field descriptions

Field	Description
VALUES	Reserved

17.6.20 Bus Master and ECC Controller Debug Read Register (BCH_DBGAHBMREAD)

The hardware BCH ECC accelerator bus master, ECC controller internal state machines, and signals can be seen in the ECC debug registers.

Address: 11_4000h base + 140h offset = 11_4140h



BCH_DBGAHBMREAD field descriptions

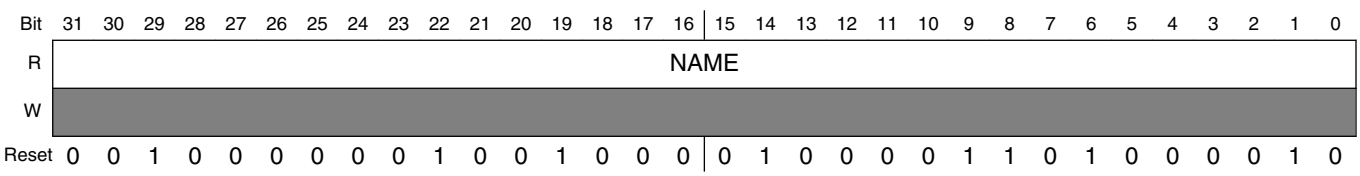
Field	Description
VALUES	Reserved

17.6.21 Block Name Register (BCH_BLOCKNAME)

Read only view of the block name string BCH.

Fixed pattern read only value is for test purposes. It can be read as an ASCII string with the zero termination coming from the first byte of the BLOCKVERSION register.

Address: 11_4000h base + 150h offset = 11_4150h



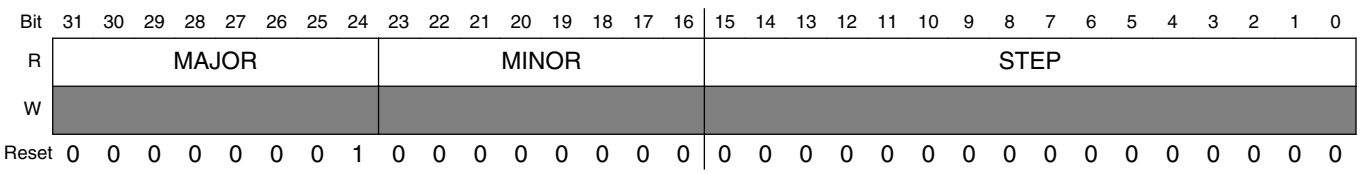
BCH_BLOCKNAME field descriptions

Field	Description
NAME	The name is in the ASCII characters BCH (0x20, H, C, B).

17.6.22 BCH Version Register (BCH_VERSION)

This register always returns a known read value for debug purposes and indicates the version of the block and RTL version in use.

Address: 11_4000h base + 160h offset = 11_4160h



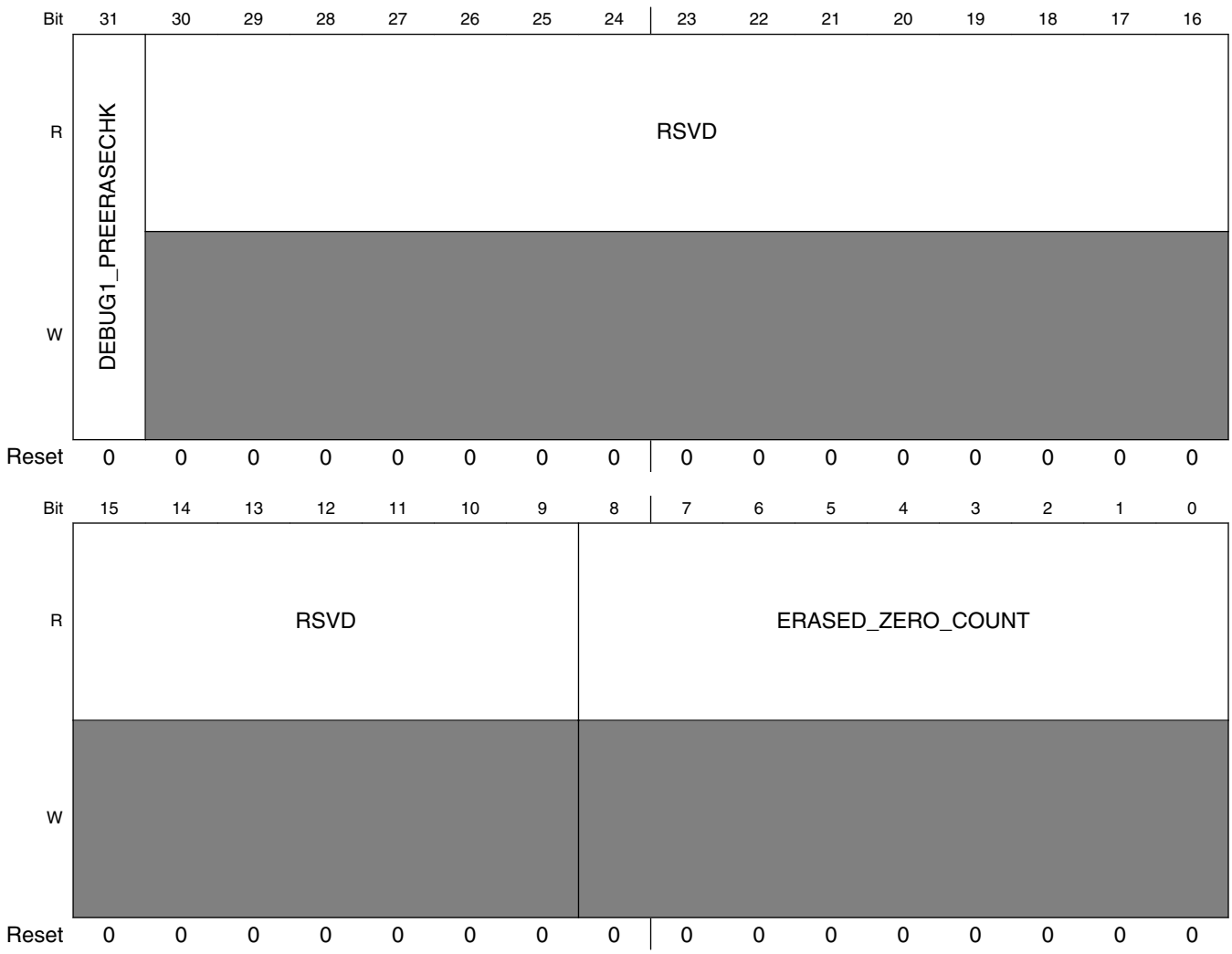
BCH_VERSION field descriptions

Field	Description
31–24 MAJOR	Fixed read-only value indicates the MAJOR field of the RTL version.
23–16 MINOR	Fixed read-only value indicates the MINOR field of the RTL version.
STEP	Fixed read-only value reflecting the stepping of the RTL version.

17.6.23 Hardware BCH ECC Debug Register 1 (BCH_DEBUG1)

The BCH_DEBUG1 register provides erased zero count information and pre-erase check.

Address: 11_4000h base + 170h offset = 11_4170h



BCH_DEBUG1 field descriptions

Field	Description
31 DEBUG1_ PREERASECHK	Blank page enables pre-erase check. 0x0 Turn off pre-erase check 0x1 Turn on pre-erase check
30–9 RSVD	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

BCH_DEBUG1 field descriptions (continued)

Field	Description
ERASED_ ZERO_COUNT	The zero counts on one page.

Chapter 18
Clock Controller Module (CCM)

18.1 Overview

The Clock Control Module (CCM) generates and controls clocks to the various modules in the design and manages low power modes. This module uses the available clock sources to generate the clock roots.

The Clock Controller Module controls the following functions:

- Uses the available clock sources to generate clock roots to various parts of the chip:
  - PLL1 also referenced as ARM PLL
  - PLL2 also referenced as System PLL
  - PLL3 also referenced as USB1 PLL
  - PLL4 also referenced as Audio PLL
  - PLL5 also referenced as Video PLL
  - PLL6 also referenced as ENET PLL
  - PLL7 also referenced as USB2 PLL (This PLL is only used by the USB UTM interface through a direct connection.)
  - PLL8 also referenced as MLB PLL
- Uses programmable bits to control frequencies of the clock roots.
- Controls the low power mechanism.
- Provides control signals to LPCG for gating clocks.
- Provides handshake with SRC for reset performance.
- Provides handshake with GPC for support of power gating operations.

18.1.1 Features

The CCM includes these distinctive features:

- Provides root clock to SoC modules based on several source clocks.
- Arm core root clock is generated from a dedicated source clock.

- Includes separate dividers to control generation of core and bus root clocks (AXI, AHB, IPG).
- Includes separate dividers and clock source selectors for each serial root clock.
- Optional external clocks to bypass PLL clocks.
- Selects clock signals to output on CCM_CLKO1 and CCM_CLKO2 onto the pads for observability.
- Controllable registers are accessible via IP bus.
- Manages the Low Power Modes, namely RUN, WAIT and STOP. The gating of the peripheral clocks is programmable in RUN and WAIT modes.
- Manages frequency scaling procedure for Arm core clock by shifting between PLL sources, without loss of clocks.
- Manages frequency scaling procedure for peripheral root clock by programmable divider. The division is done on the fly without loss of clocks.
- Interface for the following IPs:
  - PLL - Interfaces for each PLL
  - LPCG - Low Power Clock Gating unit
  - SRC - System Reset Controller
  - GPC - General Power Controller

18.1.2 CCM Block Diagram

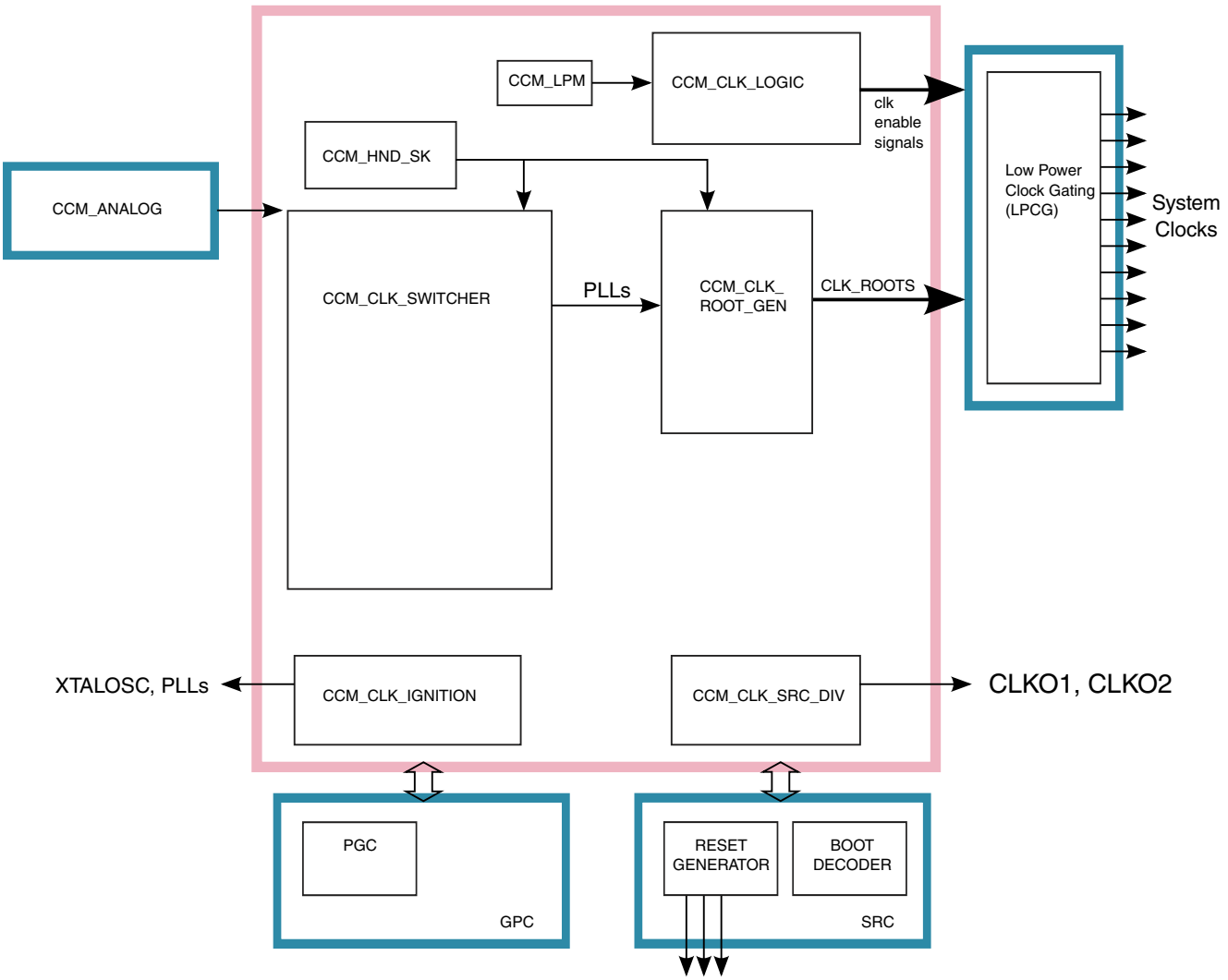


Figure 18-1. Block Diagram

CCM contains the following sub-blocks:

Table 18-1. CCM Sub-blocks

CCM_CLK_SRC_DIV Sub-block	Description
CCM_CLK_IGNITION	Manages the ignition process. This module starts its functionality after CCM comes out of reset. It manages the process that begins with starting the OSC, PLLs and finishes with creation of stable output root clocks after reset.
CCM_CLK_SWITCHER	Receives the clock outputs of the PLLs, together with the bypass clocks for the PLLs, and generates switcher clock outputs (pll1_sw_clk, pll3_sw_clk) for the CCM_CLK_ROOT_GEN sub-module.
CCM_CLK_ROOT_GEN	Receives the main clocks (PLLs / PFDs) and generates the output root clocks.
CCM_CLK_LOGIC	Generates the clock enables. It generates the clock enable signals based on info from CCM_LPM and CCM_IP. The clock enables are used in LPCG to turn off and on the split clocks.

Table continues on the next page...

Table 18-1. CCM Sub-blocks (continued)

CCM_CLK_SRC_DIV Sub-block	Description
CCM_LPM	Manages the low power modes of the IC.
CCM_CLK_SRC_DIV	Muxes different clocks to two output clocks, CCM_CLKO1 and CCM_CLKO2, for observability. These output clocks are connected to the pads and can provide support for testing.
CCM_HND_SK	Manages the handshake when changing certain root clock dividers that require handshake.

18.2 External Signals

The following table describes the external signals of CCM:

Table 18-2. CCM External Signals

Signal	Description	Pad	Mode	Direction
CCM_CLKO1	Observability clock 1 output	CSI0_MCLK	ALT3	O
		GPIO_0	ALT0	
		GPIO_19	ALT3	
		GPIO_5	ALT3	
CCM_CLKO2	Observability clock 2 output	GPIO_3	ALT4	O
		NANDF_CS2	ALT4	
CCM_PMIC_STBY_REQ	Goes to PMIC_STBY_REQ pin, which notifies external power management IC to move from functional voltage to standby voltage.	PMIC_STBY_REQ	No Muxing (ALT0)	O
CCM_REF_EN_B	Enable external reference clock (CKIH)	GPIO_9	ALT3	O
CCM_PLL3_BYP	PLL3 bypass clock	RGMII_TD1	ALT7	I

18.3 CCM Clock Tree

The following figures (Part 1 and Part 2) show the clock tree configuration and clock roots for CCM.

For detailed sub-block information, see:

- Clock Switcher
- Clock Root Generator

- Low Power Clock Gating module (LPCG)
- System Clocks

NOTE

The default frequency values (in MHz) for the PLLs and PFDs are shown in the Clock Tree diagram that follows. The PLLs and PFDs control registers may be reprogrammed according to the speed grade of the SoC being used, but should not exceed that maximum setting for that speed grade.

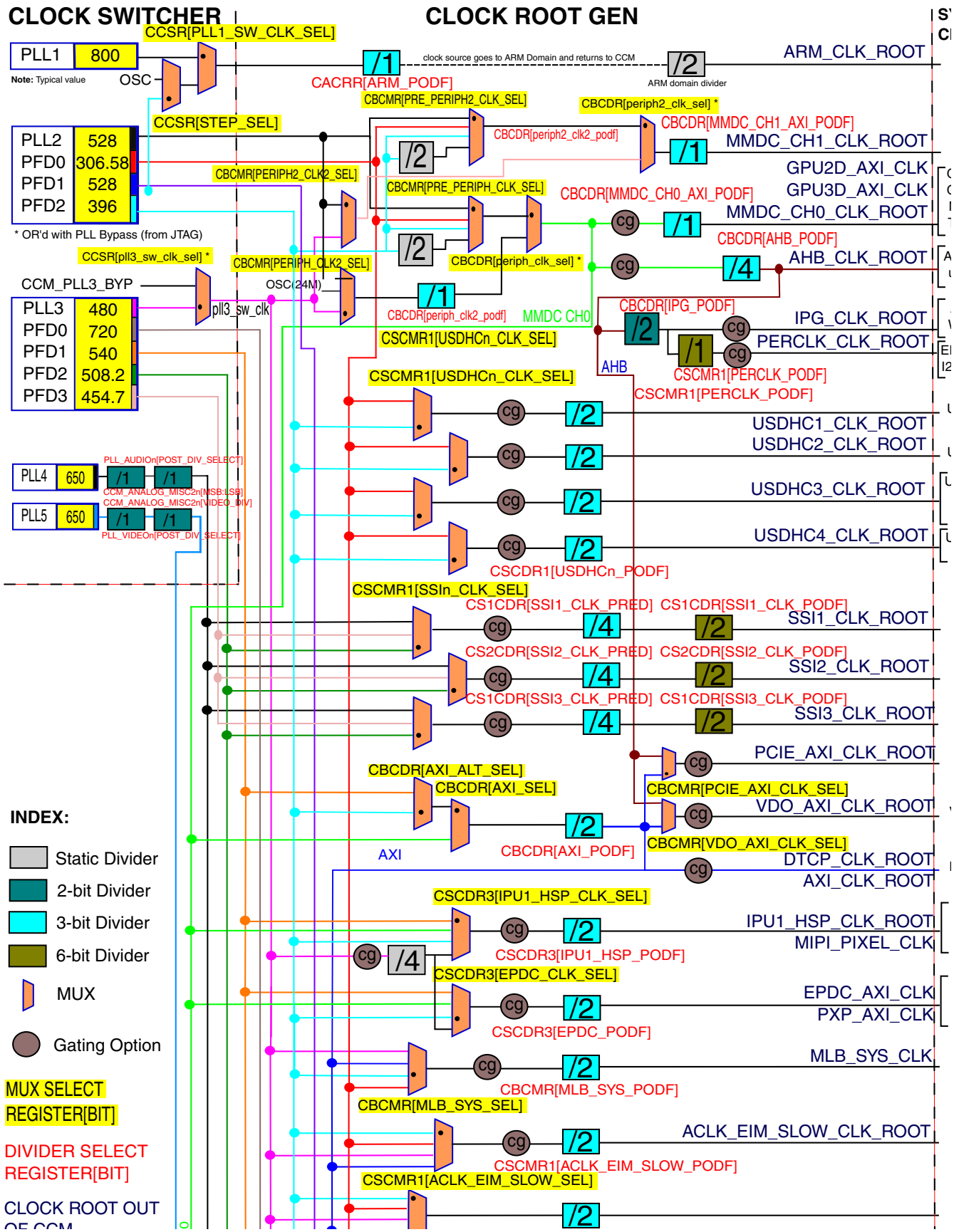
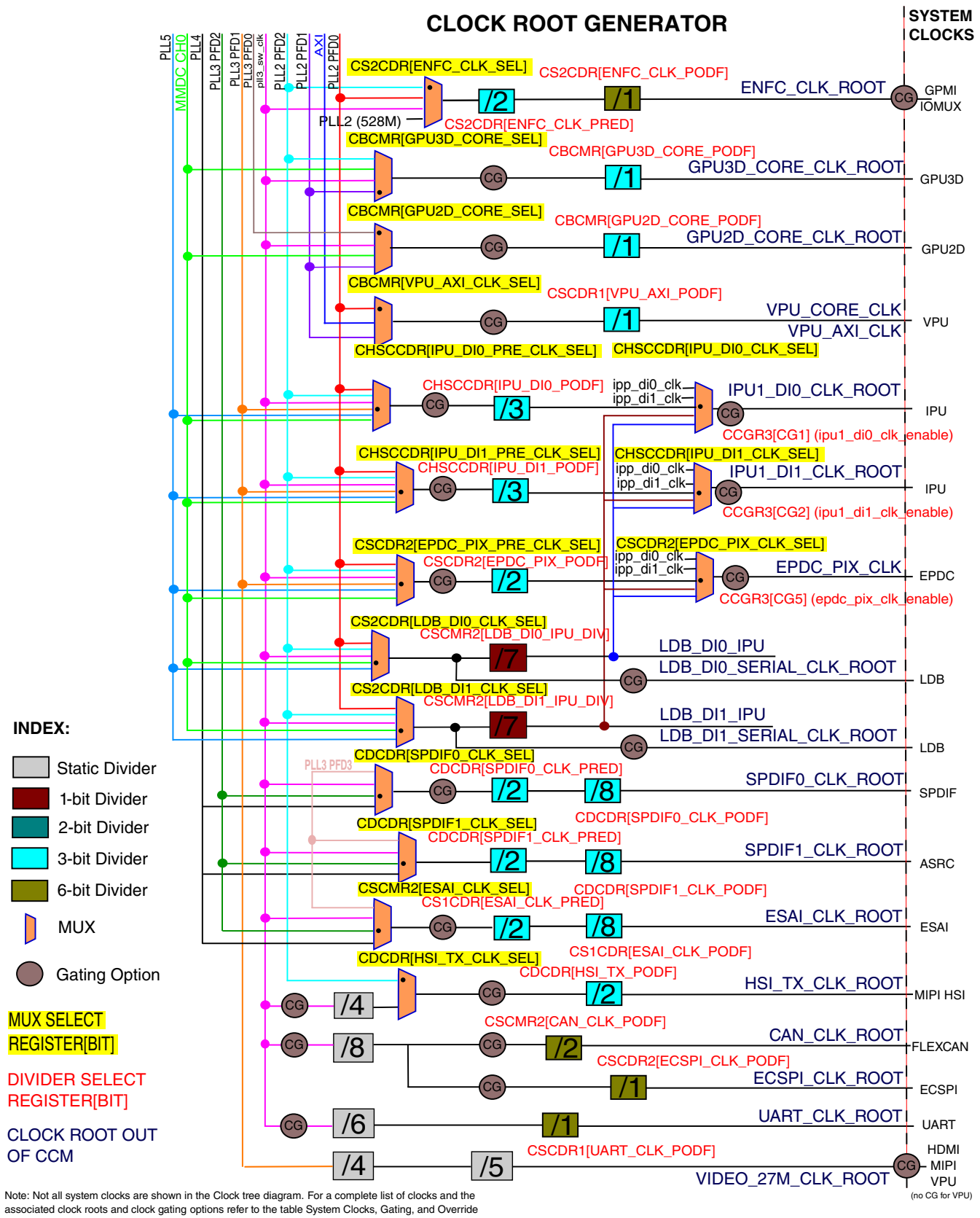


Figure 18-2. Clock Tree - Part 1



Note: Not all system clocks are shown in the Clock tree diagram. For a complete list of clocks and the associated clock roots and clock gating options refer to the table System Clocks, Gating, and Override

Figure 18-3. Clock Tree - Part 2

18.4 System Clocks

The table found here shows the CCM output clocks' system-level connectivity.

The gating option in the table can either be CGR bit or clock enable from the block itself. Applicable override bits are also shown.

Table 18-3. System Clocks, Gating, and Override

Module	Module Clock	Clock Root	Module Clock Gating Enable[1]	Module Override Enable
AIPSTZ*n*	hclk	ahb_clk_root	CCGR0[CG0] (aips_tz1_clk_enable) CCGR0[CG1] (aips_tz2_clk_enable)	
APBH	sec_mst_hclk	usdhc3_clk_root	CCGR0[CG2] (apbhdma_hclk_enable)	
	hclk	usdhc3_clk_root	CCGR0[CG2] (apbhdma_hclk_enable)	
Arm	clk_ahb	ahb_clk_root	CCGR0[CG11] (arm_dbg_clk_enable)	
	clk_apb_dbg	ahb_clk_root	CCGR0[CG11] (arm_dbg_clk_enable)	
	clk_atb	ahb_clk_root	CCGR0[CG11] (arm_dbg_clk_enable)	
	trace_clk_in	ahb_clk_root	CCGR0[CG11] (arm_dbg_clk_enable)	
	ipg_clk	ipg_clk_root	CCGR0[CG11] (arm_dbg_clk_enable)	
ASRC	ipg_clk	ahb_clk_root	CCGR0[CG3] (asrc_clk_enable)	
	mem_clk	ahb_clk_root	CCGR0[CG3] (asrc_clk_enable)	
	asrck_clock_d	spdif1_clk_root[2]		
AUDMUX	ipg_clk_s	ipg_clk_root		
CAAM	secure_mem_clk	ahb_clk_root	CCGR0[CG4] (caam_secure_mem_clk_enable)	
	exsc_aclk_exsc	ahb_clk_root	CCGR0[CG5] (caam_wrapper_aclk_enable)	
	aclk	ahb_clk_root	CCGR0[CG5] (caam_wrapper_aclk_enable)	
	ckil	ckil_sync_clk_root		

Table continues on the next page...

Table 18-3. System Clocks, Gating, and Override (continued)

Module	Module Clock	Clock Root	Module Clock Gating Enable[1]	Module Override Enable
	ipg_clk	ipg_clk_root	CCGR0[CG5] (caam_wrapper_ipg_enable)	
	ipg_clk_s	ipg_clk_root		
CCM	ipg_clk	ipg_clk_root		
	analog_apb_clk	ipg_clk_root		
	analog_ipg_clk_s	ipg_clk_root		
CSU	ap_ckil_clk	ckil_sync_clk_root		
	ipg_clk_s	ipg_clk_root		
DBGMON	clk	ipg_clk_root		
DCIC*n*	ipg_master_clk	ipg_clk_root	CCGR0[CG12] (dcic1_clk_enable) CCGR0[CG13] (dcic2_clk_enable)	
	master_clk	ipg_clk_root	CCGR0[CG12] (dcic1_clk_enable) CCGR0[CG13] (dcic2_clk_enable)	
	hsp_clk	ipu1_hsp_clk_root	CCGR0[CG12] (dcic1_clk_enable) CCGR0[CG13] (dcic2_clk_enable)	
	ipg_clk_s	ipu1_hsp_clk_root	CCGR0[CG12] (dcic1_clk_enable) CCGR0[CG13] (dcic2_clk_enable)	
	ipg_slave_clk	ipu1_hsp_clk_root	CCGR0[CG12] (dcic1_clk_enable) CCGR0[CG13] (dcic2_clk_enable)	
	slave_clk	ipu1_hsp_clk_root	CCGR0[CG12] (dcic1_clk_enable) CCGR0[CG13] (dcic2_clk_enable)	
DTCP	dtcp_clk	dtcp_clk_root	CCGR0[CG14] (dtcp_clk_enable)	
	mem_ram_CLK	dtcp_clk_root	CCGR0[CG14] (dtcp_clk_enable)	
	mem_rom_CLK	dtcp_clk_root	CCGR0[CG14] (dtcp_clk_enable)	
	sec_mst_hclk	dtcp_clk_root	CCGR0[CG14] (dtcp_clk_enable)	
ECSPI*n*	ipg_clk_32k	ckil_sync_clk_root		

Table continues on the next page...

Table 18-3. System Clocks, Gating, and Override (continued)

Module	Module Clock	Clock Root	Module Clock Gating Enable[1]	Module Override Enable
	ipg_clk_per	ecspi_clk_root	CCGR1[CG0] (ecspi1_clk_enable) CCGR1[CG1] (ecspi2_clk_enable) CCGR1[CG2] (ecspi3_clk_enable) CCGR1[CG3] (ecspi4_clk_enable)	
	ipg_clk	ipg_clk_root	CCGR1[CG0] (ecspi1_clk_enable) CCGR1[CG1] (ecspi2_clk_enable) CCGR1[CG2] (ecspi3_clk_enable) CCGR1[CG3] (ecspi4_clk_enable)	
	ipg_clk_s	ipg_clk_root		
EIM	ipg_clk_s	ipg_clk_root		
	aclk_exsc	aclk_eim_slow_clk_root	CCGR6[CG5] (emi_slow_clk_enable)	
	aclk	aclk_eim_slow_clk_root	CCGR6[CG5] (emi_slow_clk_enable)	
	aclk_slow	aclk_eim_slow_clk_root	CCGR6[CG5] (emi_slow_clk_enable)	
ENET	ipg_clk	ahb_clk_root	CCGR1[CG5] (enet_clk_enable)	
	ipg_clk_mac0	ahb_clk_root	CCGR1[CG5] (enet_clk_enable)	
	mem_mac0_rxmem_clk	ahb_clk_root	CCGR1[CG5] (enet_clk_enable)	
	mem_mac0_txmem_clk	ahb_clk_root	CCGR1[CG5] (enet_clk_enable)	
	ipg_clk_mac0_s	ipg_clk_root	CCGR1[CG5] (enet_clk_enable)	
	ipg_clk_s	ipg_clk_root		
	ipg_clk_time	ipg_clk_root	CCGR1[CG5] (enet_clk_enable)	
EPDC	pixclk	epdc_pix_clk_root	CCGR3[CG5] (epdc_pix_clk_enable)	
	aclk	epdc_axi_clk_root pxp_axi_clk_root	CCGR3[CG3] (epdc_axi_clk_enable)	
EPIT*n*	ipg_clk_32k	ckil_sync_clk_root		
	ipg_clk	ipg_clk_root	CCGR1[CG6] (epit1_clk_enable)	CMEOR[mod_en_ov_epit]

Table continues on the next page...

Table 18-3. System Clocks, Gating, and Override (continued)

Module	Module Clock	Clock Root	Module Clock Gating Enable[1]	Module Override Enable
			CCGR1[CG7] (epit2_clk_enable)	
	ipg_clk_s	ipg_clk_root		
	ipg_clk_highfreq	perclk_clk_root	CCGR1[CG6] (epit1_clk_enable) CCGR1[CG7] (epit2_clk_enable)	
ESAI*n*	ipg_clk_esai	ahb_clk_root	CCGR1[CG8] (esai_clk_enable)	
	mem_clk	ahb_clk_root	CCGR1[CG8] (esai_clk_enable)	
	extal_clk	esai_clk_root	CCGR1[CG8] (esai_clk_enable)	
	ipg_clk_s	ipg_clk_root		
FLEXCAN*n*	ipg_clk_pe	can_clk_root	CCGR0[CG8] (can1_serial_clk_enable) CCGR0[CG10] (can2_serial_clk_enable)	CMEOR[mod_en_ov_can1_cpi]
	ipg_clk_pe_nogate	can_clk_root	CCGR0[CG8] (can1_serial_clk_enable) CCGR0[CG10] (can2_serial_clk_enable)	
	ipg_clk	ipg_clk_root	CCGR0[CG7] (can1_clk_enable) CCGR0[CG9] (can2_clk_enable)	
	ipg_clk_chi	ipg_clk_root	CCGR0[CG7] (can1_clk_enable) CCGR0[CG9] (can2_clk_enable)	
	ipg_clk_s	ipg_clk_root		
	mem_ram_CLK	ipg_clk_root	CCGR0[CG7] (can1_clk_enable) CCGR0[CG9] (can2_clk_enable)	
GPC	ipg_clk	ipg_clk_root		
	ipg_clk_s	ipg_clk_root		
	pgc_clk	ipg_clk_root		
	sys_clk	ipg_clk_root		
GPIO*n*	ipg_clk_s	ipg_clk_root		
GPMI	gpmi_bch_input_gpmi_io_clk	enfc_clk_root	CCGR4[CG14] (gpmi_bch_input_gpmi_io_clk_enable)	

Table continues on the next page...

Table 18-3. System Clocks, Gating, and Override (continued)

Module	Module Clock	Clock Root	Module Clock Gating Enable[1]	Module Override Enable
	gpmi_input_apb_clk	usdhc3_clk_root	CCGR4[CG15] (gpmi_input_apb_clk_enable)	
	gpmi_bch_input_bch_clk	usdhc4_clk_root	CCGR4[CG13] (gpmi_bch_input_bch_clk_enable)	
	bch_input_apb_clk	usdhc3_clk_root	CCGR4[CG12] (bch_input_apb_clk_enable)	
GPT	ipg_clk_24m	xtal_clkout	CCGR1[CG10] (gpt_clk_enable)	CMEOR[mod_en_ov_gpt]
	ipg_clk_32k	ckil_sync_clk_root		
	ipg_clk	ipg_clk_root	CCGR1[CG10] (gpt_clk_enable)	CMEOR[mod_en_ov_gpt]
	ipg_clk_s	ipg_clk_root		
	ipg_clk_highfreq	perclk_clk_root	CCGR1[CG11] (gpt_serial_clk_enable)	
GPU2D	sec_mst_hclk	ahb_clk_root	CCGR1[CG12] (gpu2d_clk_enable)	CMEOR[mod_en_ov_gpu2d]
	HCLK	ahb_clk_root	CCGR1[CG12] (gpu2d_clk_enable)	CMEOR[mod_en_ov_gpu2d]
	clk2x	gpu2d_core_clk_root	CCGR1[CG12] (gpu2d_clk_enable)	CMEOR[mod_en_ov_gpu2d]
	ACLK0	mmdc_ch0_clk_root	CCGR1[CG12] (gpu2d_clk_enable)	CMEOR[mod_en_ov_gpu2d]
	ACLK1	mmdc_ch0_clk_root		
GPU3D	sec_mst_hclk	ahb_clk_root	CCGR1[CG13] (gpu3d_clk_enable)	CMEOR[mod_en_ov_gpu3d]
	HCLK	ahb_clk_root	CCGR1[CG13] (gpu3d_clk_enable)	CMEOR[mod_en_ov_gpu3d]
	clk2x	gpu3d_core_clk_root	CCGR1[CG13] (gpu3d_clk_enable)	CMEOR[mod_en_ov_gpu3d]
	ACLK0	mmdc_ch0_clk_root	CCGR1[CG13] (gpu3d_clk_enable)	CMEOR[mod_en_ov_gpu3d]
HDMI	sec_mst_hclk	ahb_clk_root	CCGR2[CG0] (hdmi_tx_enable)	
	ihclk	ahb_clk_root	CCGR2[CG0] (hdmi_tx_enable)	
	iahbclk	ahb_clk_root	CCGR2[CG0] (hdmi_tx_enable)	
	icecclk	ckil_sync_clk_root		
	isfrclk	video_27m_clk_root	CCGR2[CG2] (hdmi_tx_isfrclk_enable)	
I2C*n*	ipg_clk_s	ipg_clk_root		

Table continues on the next page...

Table 18-3. System Clocks, Gating, and Override (continued)

Module	Module Clock	Clock Root	Module Clock Gating Enable[1]	Module Override Enable
	ipg_clk_patref	perclk_clk_root	CCGR2[CG3] (i2c1_serial_clk_enable) CCGR2[CG4] (i2c2_serial_clk_enable) CCGR2[CG5] (i2c3_serial_clk_enable)	
IOMUX	ipt_clk_io	enfc_clk_root	CCGR2[CG7] (iomux_ipt_clk_io_enable)	
	ipg_clk_s	ipg_clk_root		
IPU	ipu_master_hclk	ahb_clk_root	CCGR3[CG0] (ipu1_clk_enable)	
	ipp_di_0_ext_clk	ipu1_di0_clk_root	CCGR3[CG1] (ipu1_di0_clk_enable)	
	ipp_di_1_ext_clk	ipu1_di1_clk_root	CCGR3[CG2] (ipu1_di1_clk_enable)	
	hsp_clk	ipu1_hsp_clk_root	CCGR3[CG0] (ipu1_clk_enable)	
	sec_mst_hclk	ahb_clk_root	CCGR3[CG0] (ipu1_clk_enable)	
KPP	ipg_clk_32k	ckil_sync_clk_root		
	ipg_clk_s	ipg_clk_root		
LDB	ch_0_serial_clk	ldb_di0_serial_clk_root	CCGR3[CG6] (ldb_di0_clk_enable)	
	di_0_clk_nc	ldb_di0_serial_clk_root	CCGR3[CG6] (ldb_di0_clk_enable)	
	ch_1_serial_clk	ldb_di1_serial_clk_root	CCGR3[CG7] (ldb_di1_clk_enable)	
	di_1_clk_nc	ldb_di1_serial_clk_root	CCGR3[CG7] (ldb_di1_clk_enable)	
MIPI_CORE	ccm_pixel_clk	ipu1_hsp_clk_root		
	ccm_ac_clk_125m	ahb_clk_root	CCGR3[CG8] (mipi_core_cfg_clk_enable)	
	ips_clk	ipg_clk_root	CCGR3[CG8] (mipi_core_cfg_clk_enable)	
	ips_clk_s	ipg_clk_root		
	cfg_clk	video_27m_clk_root	CCGR3[CG8] (mipi_core_cfg_clk_enable)	
	pll_refclk	video_27m_clk_root	CCGR3[CG8] (mipi_core_cfg_clk_enable)	

Table continues on the next page...

Table 18-3. System Clocks, Gating, and Override (continued)

Module	Module Clock	Clock Root	Module Clock Gating Enable[1]	Module Override Enable
MIPI_HSI	sec_mst_hclk	ahb_clk_root	CCGR3[CG8] (mipi_core_cfg_clk_enable)	
	h_clk	ahb_clk_root	CCGR3[CG8] (mipi_core_cfg_clk_enable)	
	tx_ref_clk	hsi_tx_clk_root	CCGR3[CG8] (mipi_core_cfg_clk_enable)	
MLB	sys_clk	mlb_sys_clk	CCGR3[CG9] (mlb_clk_enable)	
	mem_db_CLK	mlb_sys_clk	CCGR3[CG9] (mlb_clk_enable)	
	mem_ct_CLK	mlb_sys_clk	CCGR3[CG9] (mlb_clk_enable)	
	ipg_clk_s	ipg_clk_root	CCGR3[CG9] (mlb_clk_enable)	
	hclk	ahb_clk_root	CCGR3[CG9] (mlb_clk_enable)	
MMDC	ipg_clk_p0	ipg_clk_root	CCGR3[CG12] (mmdc_core_ipg_clk_p0_enable)	
	ipg_clk_p1	ipg_clk_root		
	aclk_fast_core_p0	mmdc_ch0_clk_root	CCGR3[CG10] (mmdc_core_aclk_fast_core_p0_enable)	
	aclk_fast_phy_p0	mmdc_ch0_clk_root	CCGR3[CG10] (mmdc_core_aclk_fast_core_p0_enable)	
	aclk_fast_core_p1	GND		
	aclk_fast_phy_p1	GND		
OCOTP	ipg_clk	ipg_clk_root	CCGR2[CG6] (iim_clk_enable)	
	ipg_clk_s	ipg_clk_root		
OCRAM	clk	ahb_clk_root	CCGR3[CG14] (ocram_clk_enable)	
	exsc_aclk_exsc	ahb_clk_root	CCGR3[CG14] (ocram_clk_enable)	
	mem_clk	ahb_clk_root	CCGR3[CG14] (ocram_clk_enable)	
PCIE	rst_aux_clk	ipg_clk_root	CCGR4[CG0] (pcie_root_enable)	
	mstr_aclk	pcie_axi_clk_root	CCGR4[CG0] (pcie_root_enable)	
	slv_aclk	pcie_axi_clk_root	CCGR4[CG0] (pcie_root_enable)	

Table continues on the next page...

Table 18-3. System Clocks, Gating, and Override (continued)

Module	Module Clock	Clock Root	Module Clock Gating Enable[1]	Module Override Enable
	ram_slv_axi_clk	pcie_axi_clk_root	CCGR4[CG0] (pcie_root_enable)	
	exsc_aclk_exsc	pcie_axi_clk_root	CCGR4[CG0] (pcie_root_enable)	
	ram_mstr_axi_clk	pcie_axi_clk_root	CCGR4[CG0] (pcie_root_enable)	
	rst_mstr_axi_clk	pcie_axi_clk_root	CCGR4[CG0] (pcie_root_enable)	
	rst_slv_axi_clk	pcie_axi_clk_root	CCGR4[CG0] (pcie_root_enable)	
	rst_dbi_axi_clk	pcie_axi_clk_root	CCGR4[CG0] (pcie_root_enable)	
PWM*n*	ipg_clk_32k	ckil_sync_clk_root		
	ipg_clk	ipg_clk_root	CCGR4[CG8] (pwm1_clk_enable) CCGR4[CG9] (pwm2_clk_enable) CCGR4[CG10] (pwm3_clk_enable) CCGR4[CG11] (pwm4_clk_enable)	
	ipg_clk_s	ipg_clk_root		
	ipg_clk_highfreq	perclk_clk_root	CCGR4[CG8] (pwm1_clk_enable) CCGR4[CG9] (pwm2_clk_enable) CCGR4[CG10] (pwm3_clk_enable) CCGR4[CG11] (pwm4_clk_enable)	
PXP	clk	epdc_axi_clk_root pxp_axi_clk_root	CCGR3[CG3] (epdc_axi_clk_enable)	
ROMCP	sec_mst_hclk	ahb_clk_root	CCGR5[CG0] (rom_clk_enable)	
	rom_96k_CLK	ahb_clk_root	CCGR5[CG0] (rom_clk_enable)	
	hclk	ahb_clk_root	CCGR5[CG0] (rom_clk_enable)	
	hclk_reg	ipg_clk_root	CCGR5[CG0] (rom_clk_enable)	
SDMA	events_sync_clk	ahb_clk_root	CCGR5[CG3] (sdma_clk_enable)	
	ap_ahb_clk	ahb_clk_root	CCGR5[CG3] (sdma_clk_enable)	

Table continues on the next page...

Table 18-3. System Clocks, Gating, and Override (continued)

Module	Module Clock	Clock Root	Module Clock Gating Enable[1]	Module Override Enable
	ips_hostctrl_clk	ipg_clk_root	CCGR5[CG3] (sdma_clk_enable)	
	core_clk	ipg_clk_root	CCGR5[CG3] (sdma_clk_enable)	
	tck	sjc_tck_fixme		
SNVS	ipg_hp_rtc_clk	ckil_sync_clk_root		
	hp_ipg_clk	ipg_clk_root		
	hp_ipg_clk_s	ipg_clk_root		
	lp_ipg_clk	ipg_clk_root		
	lp_ipg_clk_s	ipg_clk_root		
SPBA	ipg_clk	ipg_clk_root	CCGR5[CG6] (spba_clk_enable)	
	ipg_clk_s	ipg_clk_root	CCGR5[CG6] (spba_clk_enable)	
SPDIF	gclkw_t0	ipg_clk_root	CCGR5[CG7] (spdif_clk_enable)	
	ipg_clk_s	ipg_clk_root		
	tx_clk	spdif0_clk_root	CCGR5[CG7] (spdif_clk_enable)	
SRC	ipg_clk	ipg_clk_root		
	ipg_clk_s	ipg_clk_root		
SSI*n*	ipg_clk	ipg_clk_root	CCGR5[CG9] (ssi1_clk_enable) CCGR5[CG10] (ssi2_clk_enable) CCGR5[CG11] (ssi3_clk_enable)	
	ipg_clk_s	ipg_clk_root		
	ccm_ssi_clk	ssi_clk_root	CCGR5[CG9] (ssi1_clk_enable) CCGR5[CG10] (ssi2_clk_enable) CCGR5[CG11] (ssi3_clk_enable)	
TZASC*n*	aclk	mmdc_ch0_clk_root	CCGR2[CG11] (ipsync_ip2apb_tzasc1_ipg_master_clk_enable)	
UART*n*	ipg_clk	ipg_clk_root	CCGR5[CG12] (uart_clk_enable)	
	ipg_clk_s	ipg_clk_root		
	ipg_perclk	uart_clk_root	CCGR5[CG13] (uart_serial_clk_enable)	

Table continues on the next page...

Table 18-3. System Clocks, Gating, and Override (continued)

Module	Module Clock	Clock Root	Module Clock Gating Enable[1]	Module Override Enable
USB	ipg_ahb_clk	ahb_clk_root	CCGR0[CG11] (usboh3_clk_enable)	
	ipg_clk_32khz	ckil_sync_clk_root		
	ipg_clk_s	ipg_clk_root		
	ipg_clk_s_pl301	ipg_clk_root	CCGR6[CG0] (usboh3_clk_enable)	
	test_clk_240m	ipg_clk_root	CCGR6[CG0] (usboh3_clk_enable)	
	test_clk_480m	ipg_clk_root	CCGR6[CG0] (usboh3_clk_enable)	
	test_clk_60m	ipg_clk_root	CCGR6[CG0] (usboh3_clk_enable)	
USDHC*n*	hclk	ahb_clk_root	CCGR5[CG1] (usdhc1_clk_enable) CCGR5[CG2] (usdhc2_clk_enable) CCGR5[CG3] (usdhc3_clk_enable) CCGR5[CG4] (usdhc4_clk_enable)	CMEOR[mod_en_ov_usdhc]
	ipg_clk	ipg_clk_root	CCGR5[CG1] (usdhc1_clk_enable) CCGR5[CG2] (usdhc2_clk_enable) CCGR5[CG3] (usdhc3_clk_enable) CCGR5[CG4] (usdhc4_clk_enable)	CMEOR[mod_en_ov_usdhc]
	ipg_clk_s	ipg_clk_root		
	ipg_clk_perclk	usdhc_clk_root	CCGR5[CG1] (usdhc1_clk_enable) CCGR5[CG2] (usdhc2_clk_enable) CCGR5[CG3] (usdhc3_clk_enable) CCGR5[CG4] (usdhc4_clk_enable)	CMEOR[mod_en_ov_usdhc]
VDOA	vdoa_clk	vdo_axi_clk_root	CCGR6[CG6] (vdoaxi_clk_enable)	CMEOR[mod_en_ov_vdoa]
	ipg_clk_s	vdo_axi_clk_root	CCGR6[CG6] (vdoaxi_clk_enable)	
	mem_vdoa_clk	vdo_axi_clk_root	CCGR6[CG6] (vdoaxi_clk_enable)	CMEOR[mod_en_ov_vdoa]
VPU	ipg_clk_s	ipg_clk_root		

Table continues on the next page...

Table 18-3. System Clocks, Gating, and Override (continued)

Module	Module Clock	Clock Root	Module Clock Gating Enable[1]	Module Override Enable
	rclk	video_27m_clk_root		CMEOR[mod_en_ov_vpu)
	aclk	vpu_axi_clk_root	CCGR6[CG7] (vpu_clk_enable)	CMEOR[mod_en_ov_vpu]
	cclk	vpu_axi_clk_root	CCGR6[CG7] (vpu_clk_enable)	CMEOR[mod_en_ov_vpu]
WDOG*n*	ipg_clk_32k	ckil_sync_clk_root		
	ipg_clk	ipg_clk_root		
	ipg_clk_s	ipg_clk_root		

1. System or module clocks that cannot be clock gated by software control are shaded gray in the Module Clock Gating Enable column.
2. SPDIF1_CLK_ROOT cannot be gated by software.

Table 18-4. System Clock Frequency Values

Clock Root	Default Frequency (MHz)	Maximum Frequency (MHz)
ARM_CLK_ROOT	792	
MMDC_CH0_CLK_ROOT GPU3D_AXI_CLK GPU2D_AXI_CLK	396	528
AHB_CLK_ROOT	132	133
IPG_CLK_ROOT	66	66
PERCLK_CLK_ROOT	66	66
USDHCn_CLK_ROOT	198	208
SSIn_CLK_ROOT	63.5	66
DTCP_CLK_ROOT	270	540
PCIE_AXI_CLK_ROOT	270	540
VDO_AXI_CLK_ROOT	270	540
IPU1_HSP_CLK_ROOT MIPI_PIXEL_CLK	270	270
EPDC_AXI_CLK PXP_AXI_CLK	198	200
MLB_SYS_CLK	153	528
ACLK_EIM_SLOW_CLK_ROOT	132	540
ENFC_CLK_ROOT	198	208
GPU3D_CORE_CLK_ROOT	528	533
GPU2D_CORE_CLK_ROOT	528	533
VPU_CORE_CLK_ROOT VPU_AXI_CLK	307	333
IPU1_DI0_CLK_ROOT	180	270
IPU1_DI1_CLK_ROOT	180	270

Table continues on the next page...

Table 18-4. System Clock Frequency Values (continued)

Clock Root	Default Frequency (MHz)	Maximum Frequency (MHz)
EPDC_PIX_CLK	270	540
LDB_DI0_SERIAL_CLK_ROOT	540	595
LDB_DI0_IPU	77	77
LDB_DI1_SERIAL_CLK_ROOT	540	595
LDB_DI1_IPU	77	85
SPDIF0_CLK_ROOT	31.8	66
SPDIF1_CLK_ROOT	31.8	66
ESAI_CLK_ROOT	31.8	66
HSI_TX_CLK_ROOT	198	200
CAN_CLK_ROOT	30	66.5
ECSPI_CLK_ROOT	60	60
UART_CLK_ROOT	80	80
VIDEO_27M_CLK_ROOT	27	27

18.5 Functional Description

This section provides a complete functional description of the block.

18.5.1 Clock Generation

18.5.1.1 External Low Frequency Clock - CKIL

The chip can use a 32 kHz or 32.768 kHz crystal as the external low-frequency source (XTALOSC). Throughout this chapter, the low-frequency crystal is referred to as the 32 kHz crystal.

This clock source should always be active when the chip is powered on. The 32 kHz entering the CCM are referred to as CKIL. CKIL is synchronized to IPG_CLK and supplied to modules that need it.

18.5.1.1.1 CKIL synchronizing to IPG_CLK

CKIL is synchronized to ipg_clk when the system is in functional mode. When the system is in STOP mode (when there is no IPG_CLK) the CKIL synchronizer is bypassed, and raw CKIL is supplied to the system.

18.5.1.2 External High Frequency Clock - CKIH and internal oscillator

The chip uses an internal oscillator to generate the reference clock (OSC). The internal oscillator is connected to the external crystal (XTALOSC) which generates the 24 MHz reference clock.

18.5.1.3 PLL reference clock

There are several PLLs in this chip.

PLL1 - ARM PLL (typical functional frequency 800 MHz)

PLL2 - System PLL (functional frequency 528 MHz)

PLL3 - USB1 PLL (functional frequency 480 MHz)

PLL4 - Audio PLL

PLL5 - Video PLL

PLL6 - ENET PLL

PLL7 - USB2 PLL (functional frequency 480 MHz)

PLL8 - MLB PLL

Some of the PLLs are described in the sections below. See CCM Analog Memory Map/ Register Definition for register information.

18.5.1.3.1 ARM PLL (PLL1)

This PLL synthesizes a low jitter clock from a 24 MHz reference clock. The clock output frequency for this PLL ranges from 650 MHz to 1.3 GHz. The output frequency is selected by a 7-bit register field CCM_ANALOG_PLL_ARM[DIV_SELECT].

PLL output frequency = Fref * DIV_SEL/2

NOTE

The upper frequency range may exceed the maximum frequency supported. Please see the datasheet for more information.

18.5.1.3.2 System PLL (PLL2)

This PLL synthesizes a low jitter clock from the 24 MHz reference clock. The PLL has one output clock, plus 3 PFD outputs. The System PLL supports spread spectrum modulation for use in applications to minimize radiated emissions. The spread spectrum PLL output clock is frequency modulated so that the energy is spread over a wider bandwidth, thereby reducing peak radiated emissions. Due to this feature support, the associated lock time of this PLL is longer than other PLLs in the SoC that do not support spread spectrum modulation.

Spread spectrum operation is controlled by configuring the CCM_ANALOG_PLL_SYS_SS register. When enabled, the PLL output frequency will decrease by the amount defined in the STEP field, until it reaches the limiting frequency in the STOP field. The frequency will then similarly return to the original nominal frequency. The following equations control the spread-spectrum operation:

$$\text{Spread spectrum range} = \text{Fref} \times \frac{\text{CCM_ANALOG_PLL_SYS_SS[STOP]}}{\text{CCM_ANALOG_PLL_SYS_DENOM[B]}}$$

$$\text{Modulation frequency} = \text{Fref} \times \frac{\text{CCM_ANALOG_PLL_SYS_SS[STEP]}}{2 \times \text{CCM_ANALOG_PLL_SYS_SS[STOP]}}$$

Although this PLL does have a DIV_SELECT register field, it is intended that this PLL will only be run at the default frequency of 528 MHz.

This PLL also supports a Fractional-N synthesizer.

18.5.1.3.3 USB1 PLL (PLL3)

These PLLs synthesize a low jitter clock from the 24 MHz reference clock. USB1 PLL has 4 frequency-programmable PFD (phase fractional divider) outputs.

The output frequency of USB1 PLL is 480 MHz. Even though USB1 PLL has a DIV_SELECT register field, this PLL should always be set to 480 MHz in normal operation.

18.5.1.3.4 Audio PLL (PLL4)

The audio PLL synthesize a low jitter clock from a 24 MHz reference clock. The clock output frequency range for this PLL is from 650 MHz to 1.3 GHz. It has a Fractional-N synthesizer.

There are /1, /2, /4 , /8, /16 post dividers for the Audio PLL. The output frequency can be set by programming the fields in the CCM_ANALOG_PLL_AUDIO, and CCM_ANALOG_MISC2 register sets according to the following equation.

PLL output frequency = Fref * (DIV_SELECT + NUM/DENOM)

18.5.1.3.5 Video PLL (PLL5)

The video PLL synthesize a low jitter clock from a 24 MHz reference clock. The clock output frequency range for this PLL is from 650 MHz to 1.3 GHz. It has a Fractional-N synthesizer.

There are /1, /2, /4, /8, /16 post dividers for the Video PLL. The output frequency can be set by programming the fields in the CCM_ANALOG_PLL_VIDEO, and CCM_ANALOG_MISC2 register sets according to the following equation.

PLL output frequency = Fref * (DIV_SELECT + NUM/DENOM)

18.5.1.3.6 MLB PLL

The MediaLB PLL is necessary in the MediaLB 6-Pin implementation to phase align the internal and external clock edges, effectively tuning out the delay of the differential clock receiver. The PLL is also responsible for generating the higher speed internal clock when the internal-to-external clock ratio is not 1:1.

18.5.1.3.7 Ethernet PLL (PLL6)

This PLL synthesizes a low jitter clock from the 24 MHz reference clock.

The PLL outputs a 500 MHz clock. The reference clocks generated by this PLL are:

- Ref_PCIe =125 MHz
- Ref_ethernet, which is configurable based on the PLL_ENET[DIV_SELECT] register field.

18.5.1.3.8 USB2 PLL (PLL7)

USB2 PLL is only used by the USB UTM interface through a direct connection.

18.5.1.4 Phase Fractional Dividers (PFD)

There are several PFD outputs from the System PLL and USB1 PLL.

Each PFD output generates a fractional multiplication of the associated PLL's VCO frequency. Where the output frequency is equal to Fvco*18/N, N can range from 12-35. The PFDs allow for clock frequency changes without forcing the relock of the root PLL. This feature is useful in support of dynamic voltage and frequency scaling (DVFS). See CCM Analog Memory Map/Register Definition.

When the related PLL is powered up from the power down state or made to go through a relock cycle due to PLL regrogramming, it is required that the related PFDx_CLKGATE bit in CCM_ANALOG_PFD_480n or CCM_ANALOG_PFD_528n, be cycled on and off (1 to 0) after PLL lock. The PFDs can be in the clock gated state during PLL relock but must be un-clock gated only after lock is achieved. See the engineering bulletin, *Configuration of Phase Fractional Dividers (EB790)* at www.nxp.com for procedure details.

18.5.1.5 CCM internal clock generation

The clock generation is comprised of two sub-modules:

CCM_CLK_SWITCHER

CCM_CLK_ROOT_GEN

18.5.1.5.1 Clock Switcher

The Clock Switcher (CCM_CLK_SWITCHER) sub-module receives the PLL output clocks and the PLL bypass clocks.

Figure 18-4 describes the generation of the three switcher clocks.

The figure also includes the Frequency Switch Control sub-module responsible for frequency change.

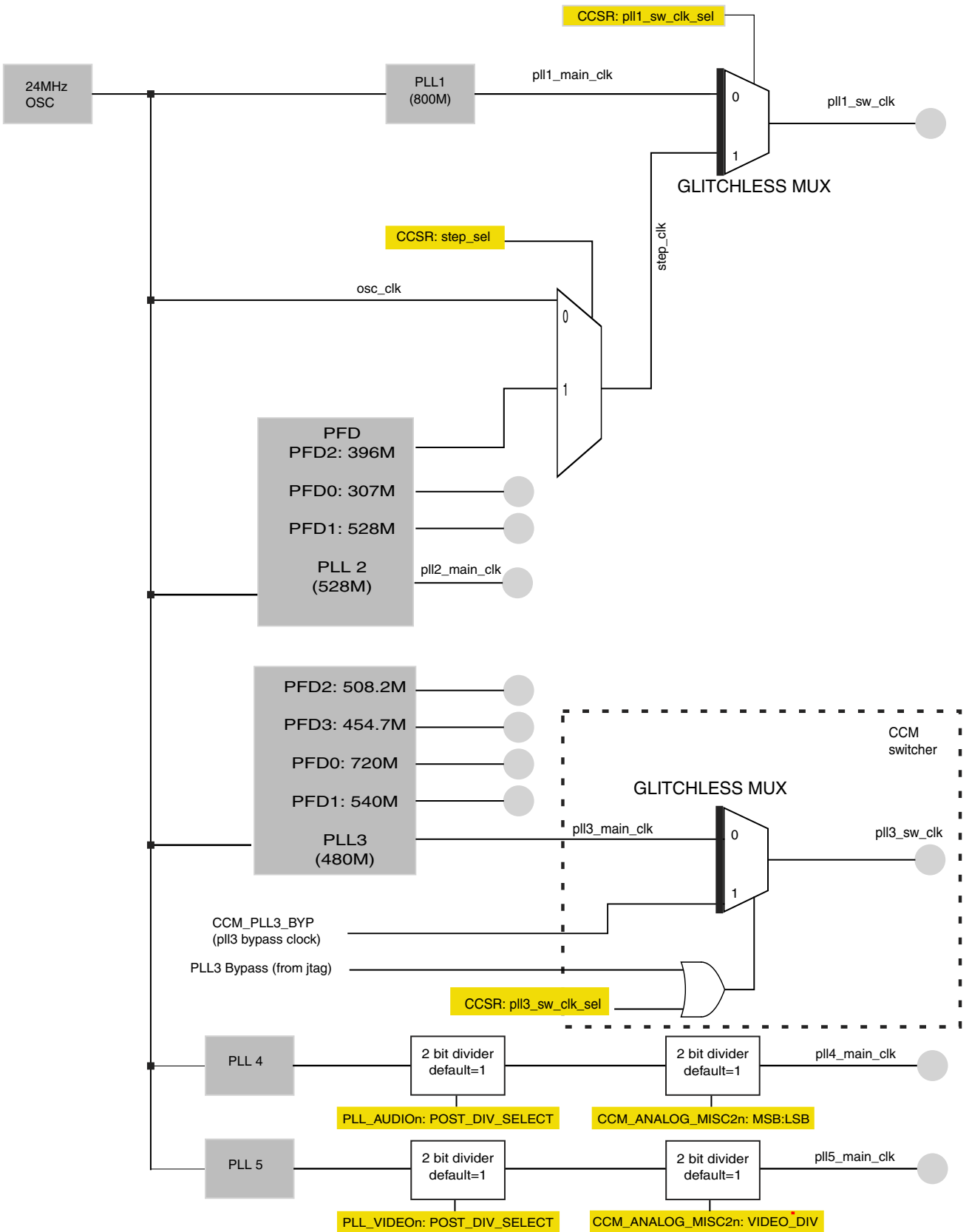


Figure 18-4. Switcher clock generation

18.5.1.5.2 PLL bypass procedure

In addition to PLL bypass options in CCM_ANALOG module, switcher and clk_root_gen sub-modules includes capability for each of the PLL clocks to be bypassed with an external bypass clock.

18.5.1.5.3 PLL clock change

In order to modify or stop the clock output of a specific PLL, all the clocks generated from the current PLL must be transitioned to the new PLL whose frequency is not being modified.

For clocks which can't be stopped (core and bus clocks), this should be done via the glitchless mux. Before changing the PLL setting, power it down. Power up the PLL after the change. See Disabling / Enabling PLLs for more information.

18.5.1.5.4 Clock Root Generator

The Clock Root Generator (CCM_CLK_ROOT_GEN) sub-module generates the root clocks to be delivered to LPCG.

The following figures describe clock generation. The frequencies in parantheses are the default typical frequencies.

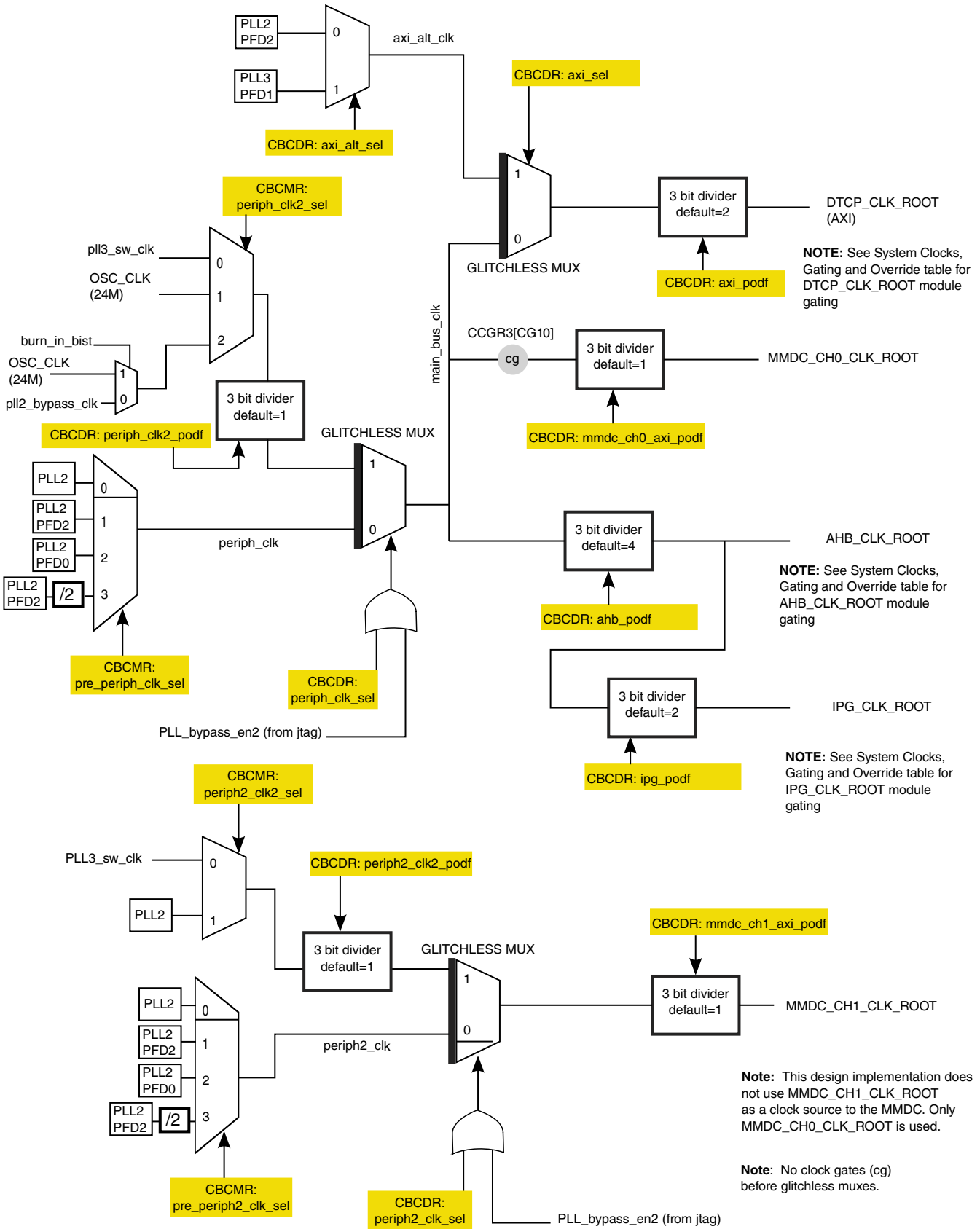
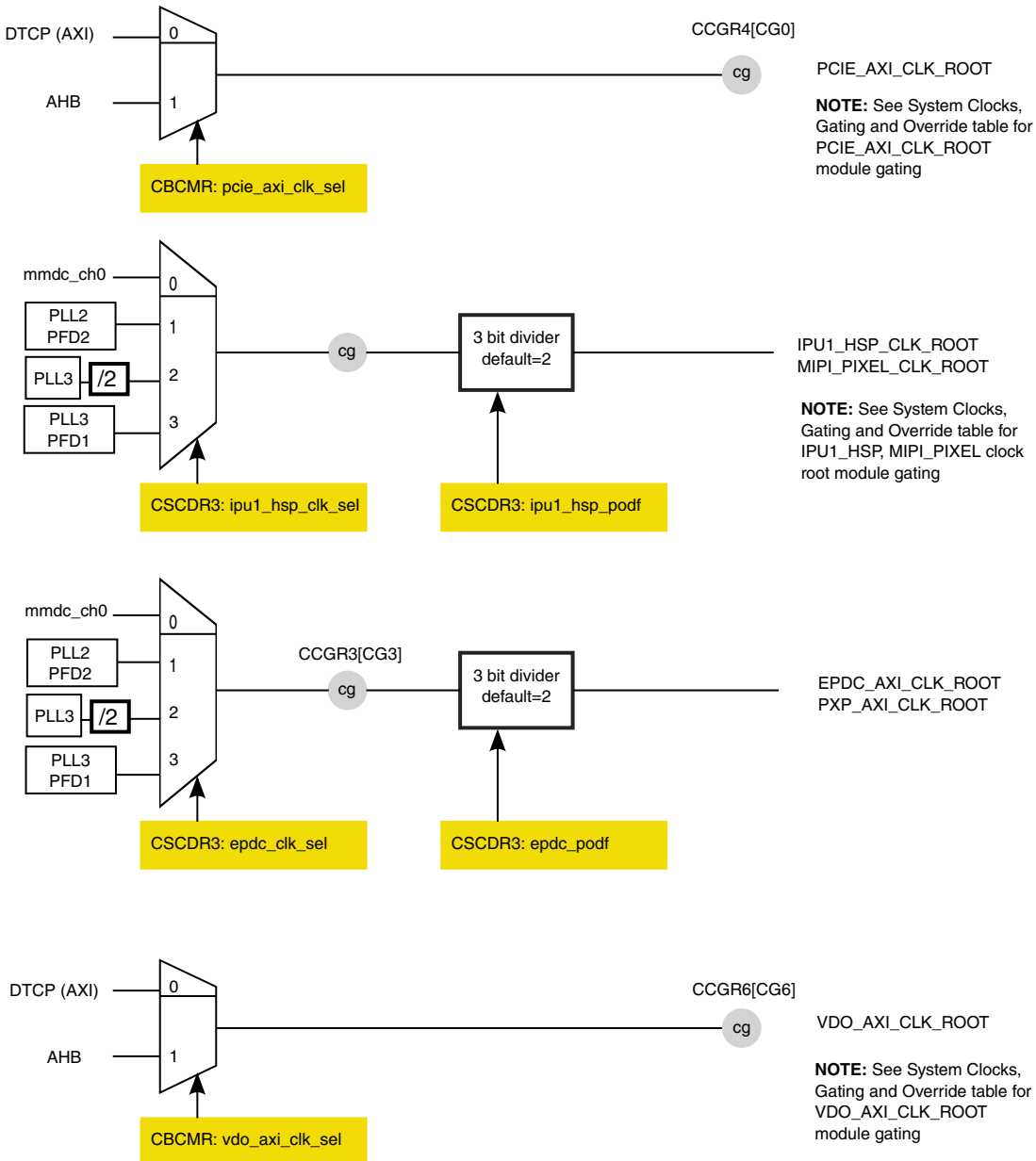


Figure 18-5. BUS clock generation


DTCP (AXI)
AHB
0
1
CBCMR: pcie_axi_clk_sel
CCGR4[CG0]
cg
PCIE_AXI_CLK_ROOT
NOTE: See System Clocks, Gating and Override table for PCIE_AXI_CLK_ROOT module gating
mmdc_ch0
PLL2 PFD2
PLL3
/2
PLL3 PFD1
0
1
2
3
CSCDR3: ipu1_hsp_clk_sel
cg
3 bit divider default=2
CSCDR3: ipu1_hsp_podf
IPU1_HSP_CLK_ROOT
MIPI_PIXEL_CLK_ROOT
NOTE: See System Clocks, Gating and Override table for IPU1_HSP, MIPI_PIXEL clock root module gating
mmdc_ch0
PLL2 PFD2
PLL3
/2
PLL3 PFD1
0
1
2
3
CSCDR3: epdc_clk_sel
CCGR3[CG3]
cg
3 bit divider default=2
CSCDR3: epdc_podf
EPDC_AXI_CLK_ROOT
PXP_AXI_CLK_ROOT
DTCP (AXI)
AHB
0
1
CBCMR: vdo_axi_clk_sel
CCGR6[CG6]
cg
VDO_AXI_CLK_ROOT
NOTE: See System Clocks, Gating and Override table for VDO_AXI_CLK_ROOT module gating

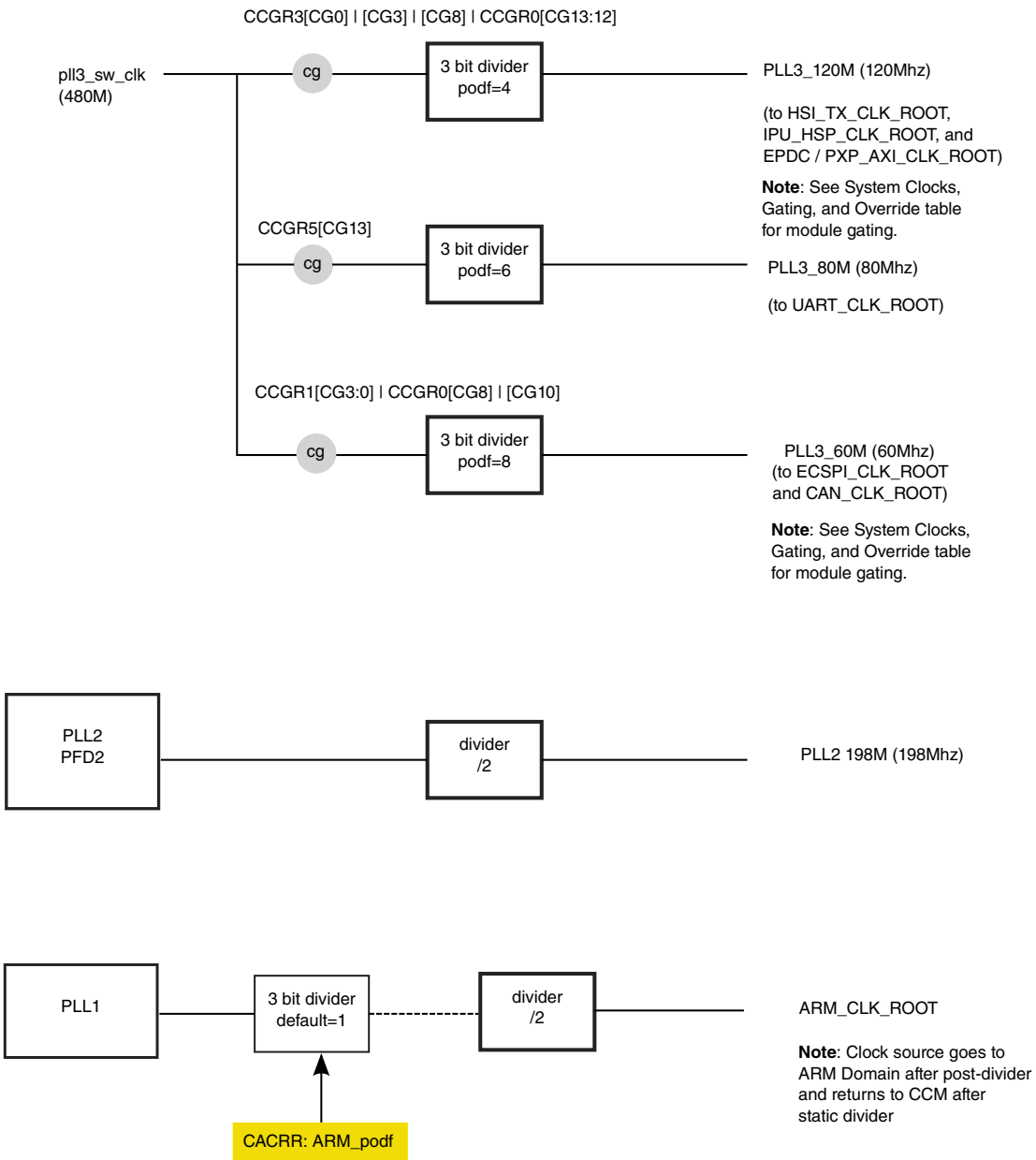
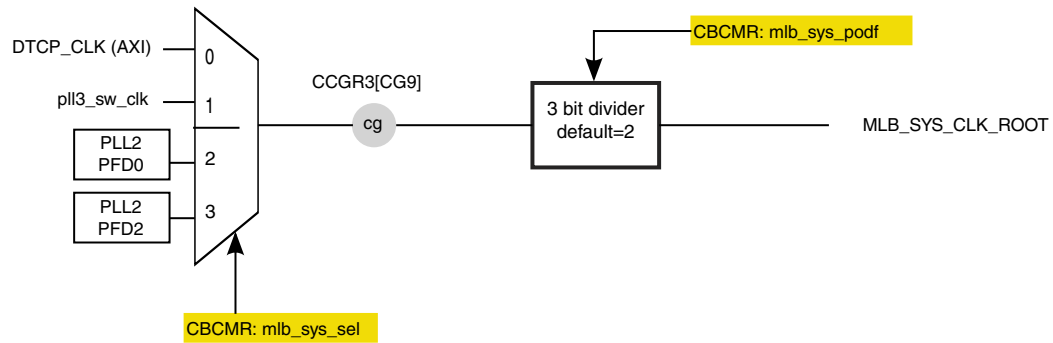
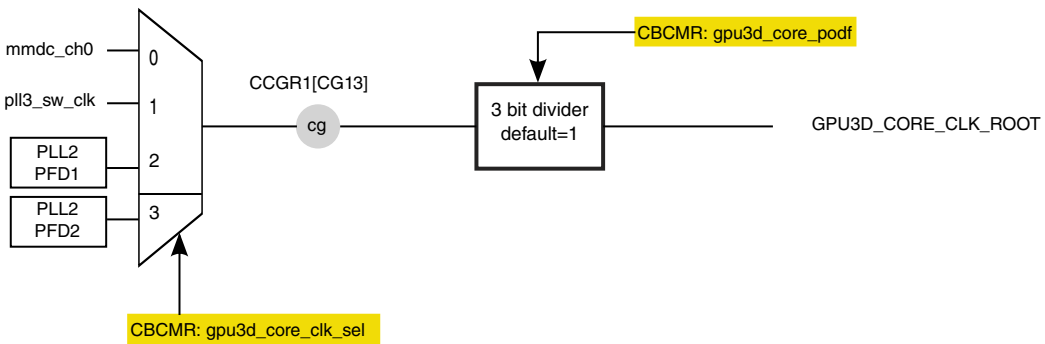
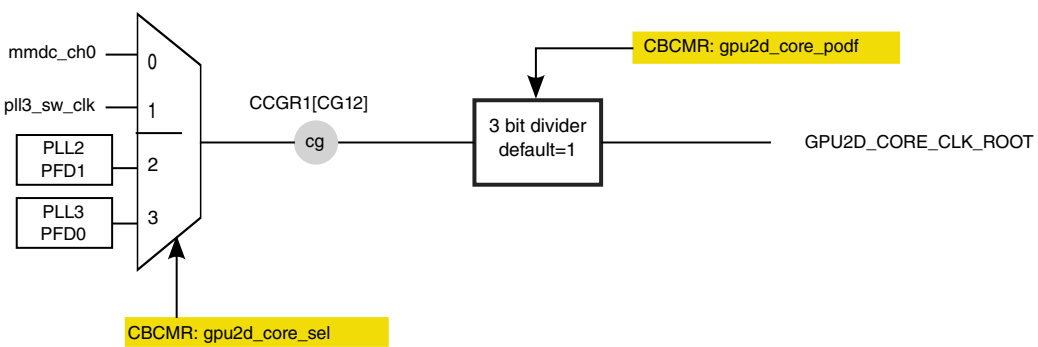
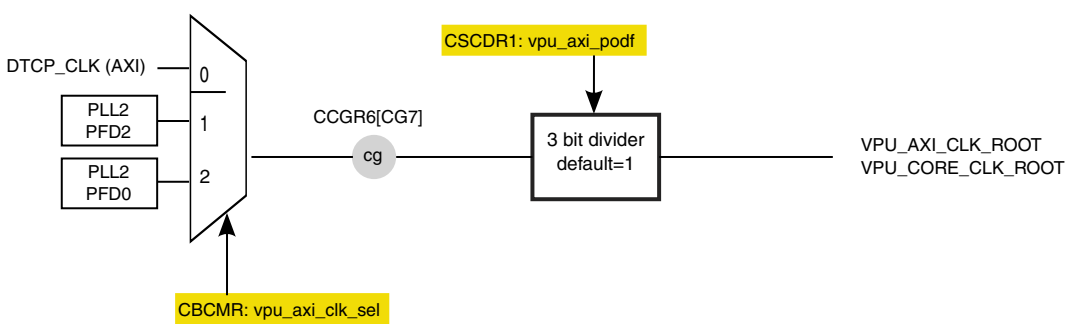


Figure 18-6. AXI clocks generation


DTCP_CLK (AXI)
pll3_sw_clk
PLL2 PFD0
PLL2 PFD2
0
1
2
3
CCGR3[CG9]
cg
3 bit divider default=2
CBCMR: mlb_sys_podf
MLB_SYS_CLK_ROOT
CBCMR: mlb_sys_sel


mmdc_ch0
pll3_sw_clk
PLL2 PFD1
PLL2 PFD2
0
1
2
3
CCGR1[CG13]
cg
3 bit divider default=1
CBCMR: gpu3d_core_podf
GPU3D_CORE_CLK_ROOT
CBCMR: gpu3d_core_clk_sel


mmdc_ch0
pll3_sw_clk
PLL2 PFD1
PLL3 PFD0
0
1
2
3
CCGR1[CG12]
cg
3 bit divider default=1
CBCMR: gpu2d_core_podf
GPU2D_CORE_CLK_ROOT
CBCMR: gpu2d_core_sel


DTCP_CLK (AXI)
PLL2 PFD2
PLL2 PFD0
0
1
2
CCGR6[CG7]
cg
3 bit divider default=1
CSCDR1: vpu_axi_podf
VPU_AXI_CLK_ROOT
VPU_CORE_CLK_ROOT
CBCMR: vpu_axi_clk_sel

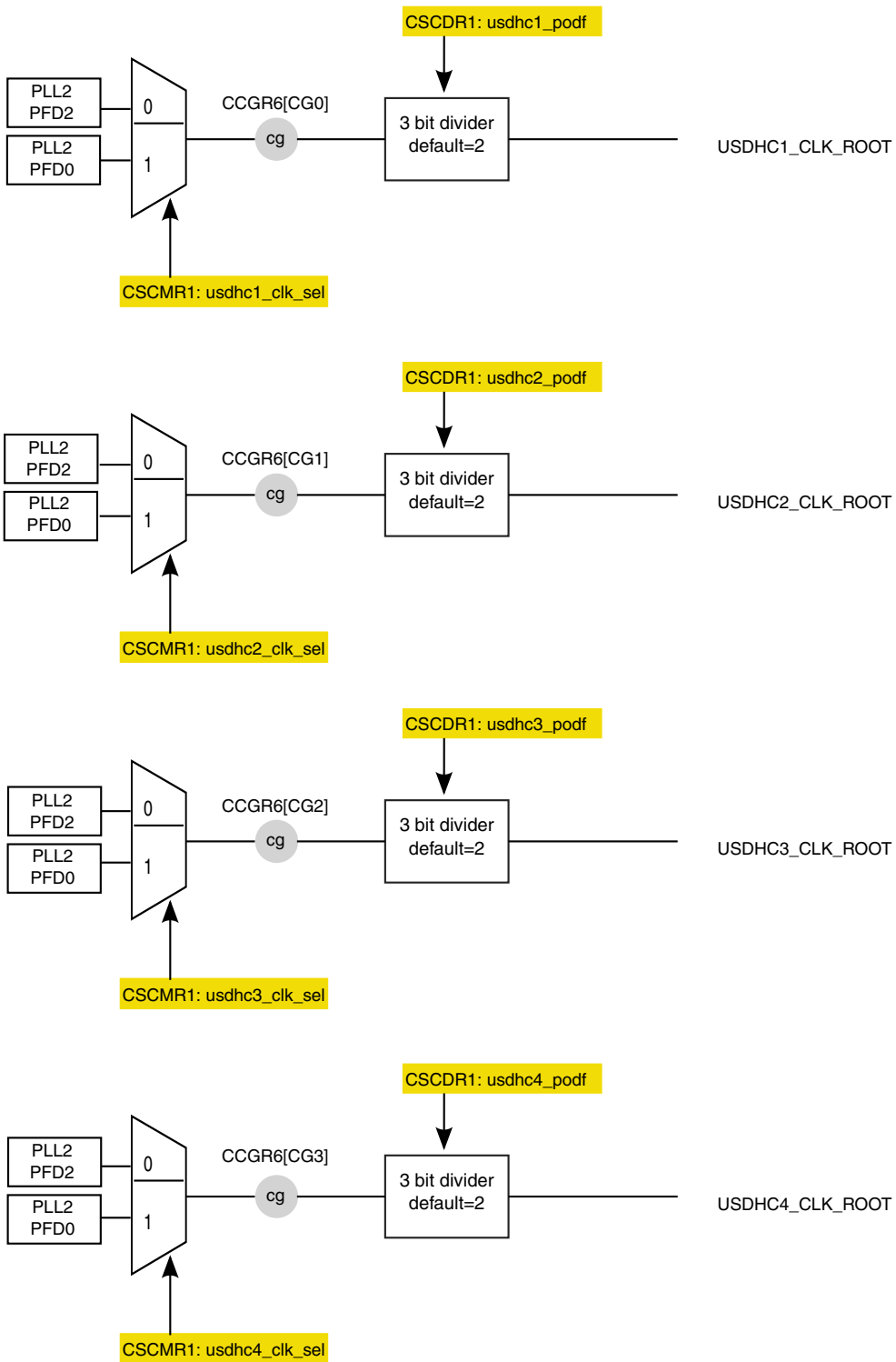
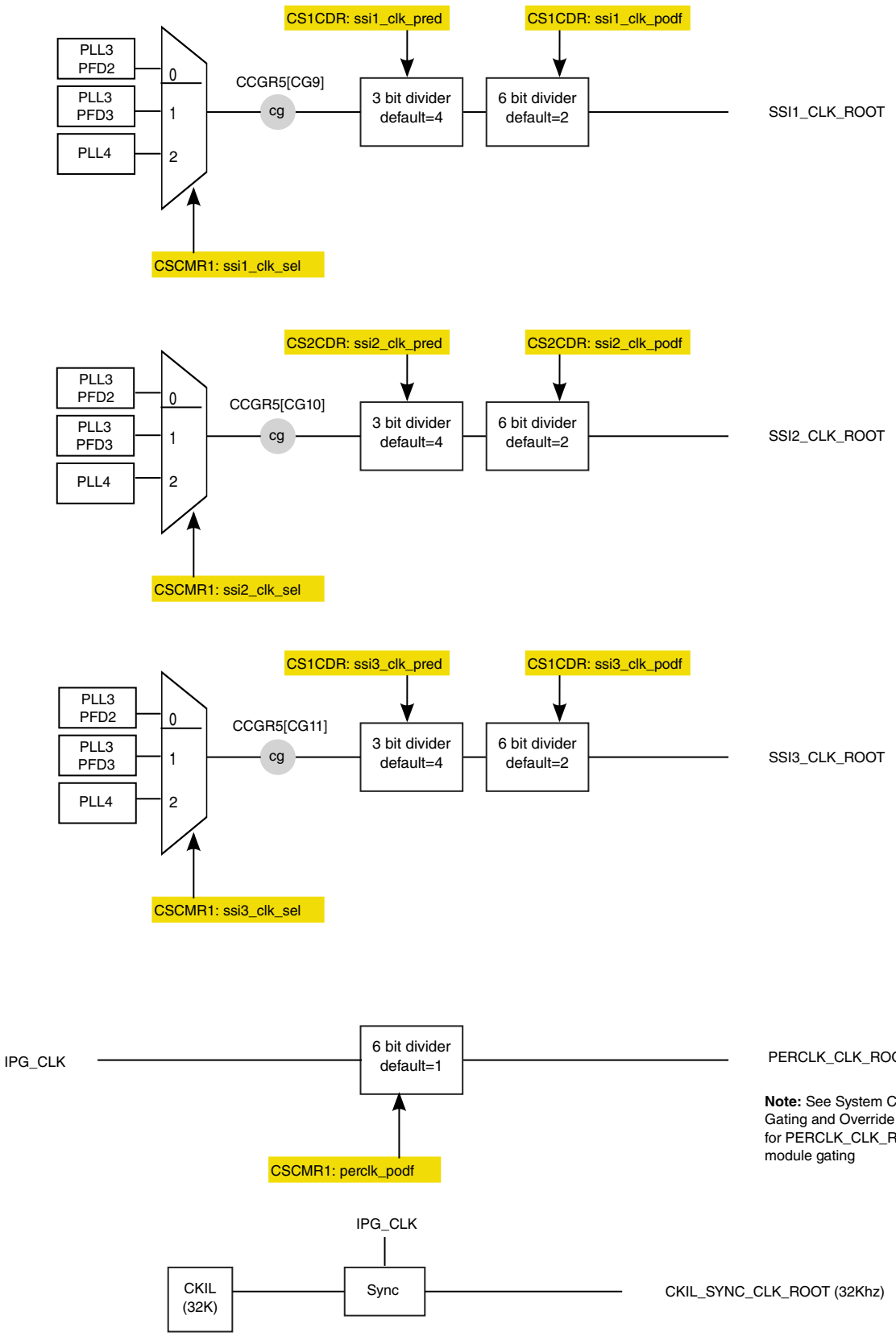
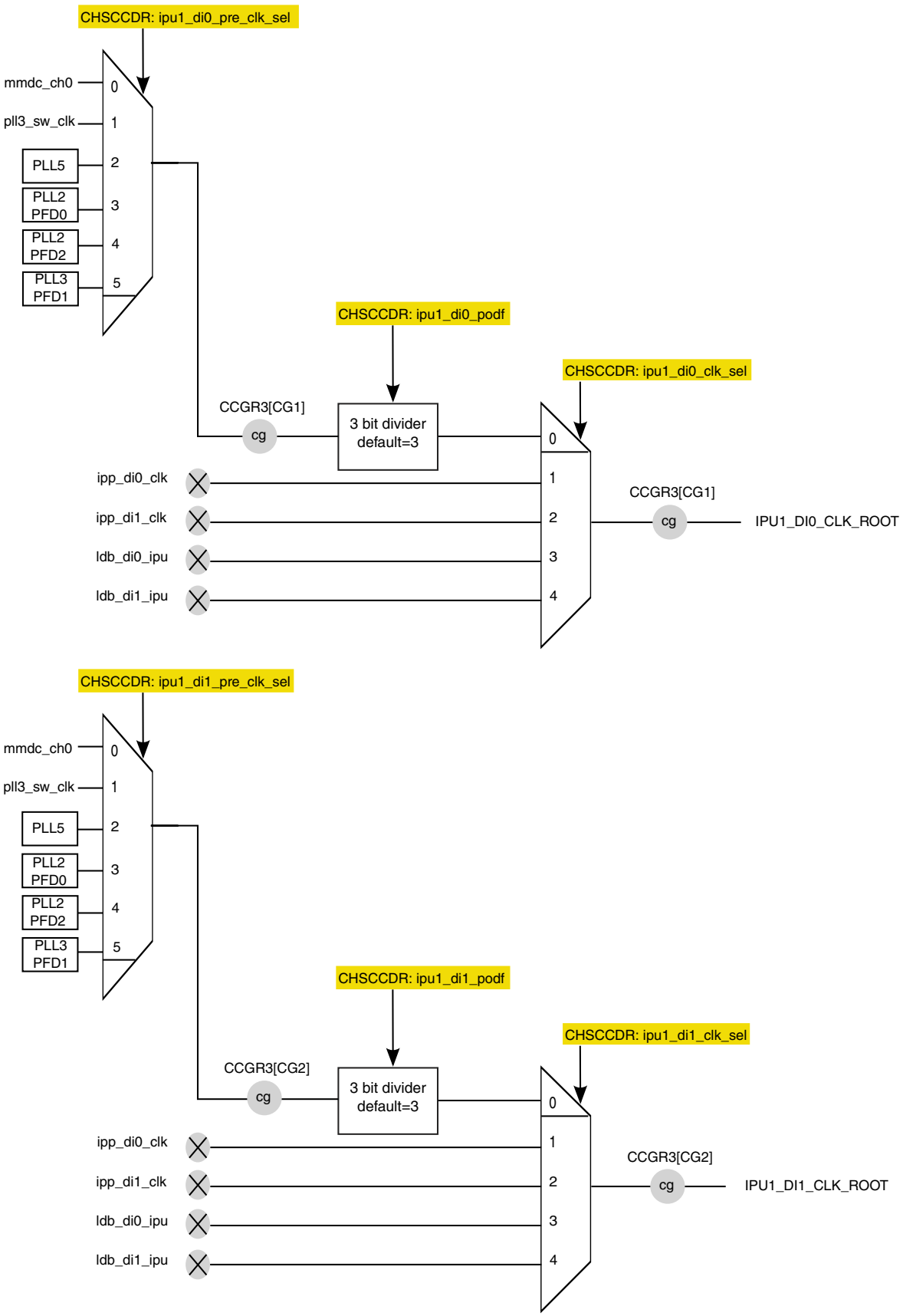
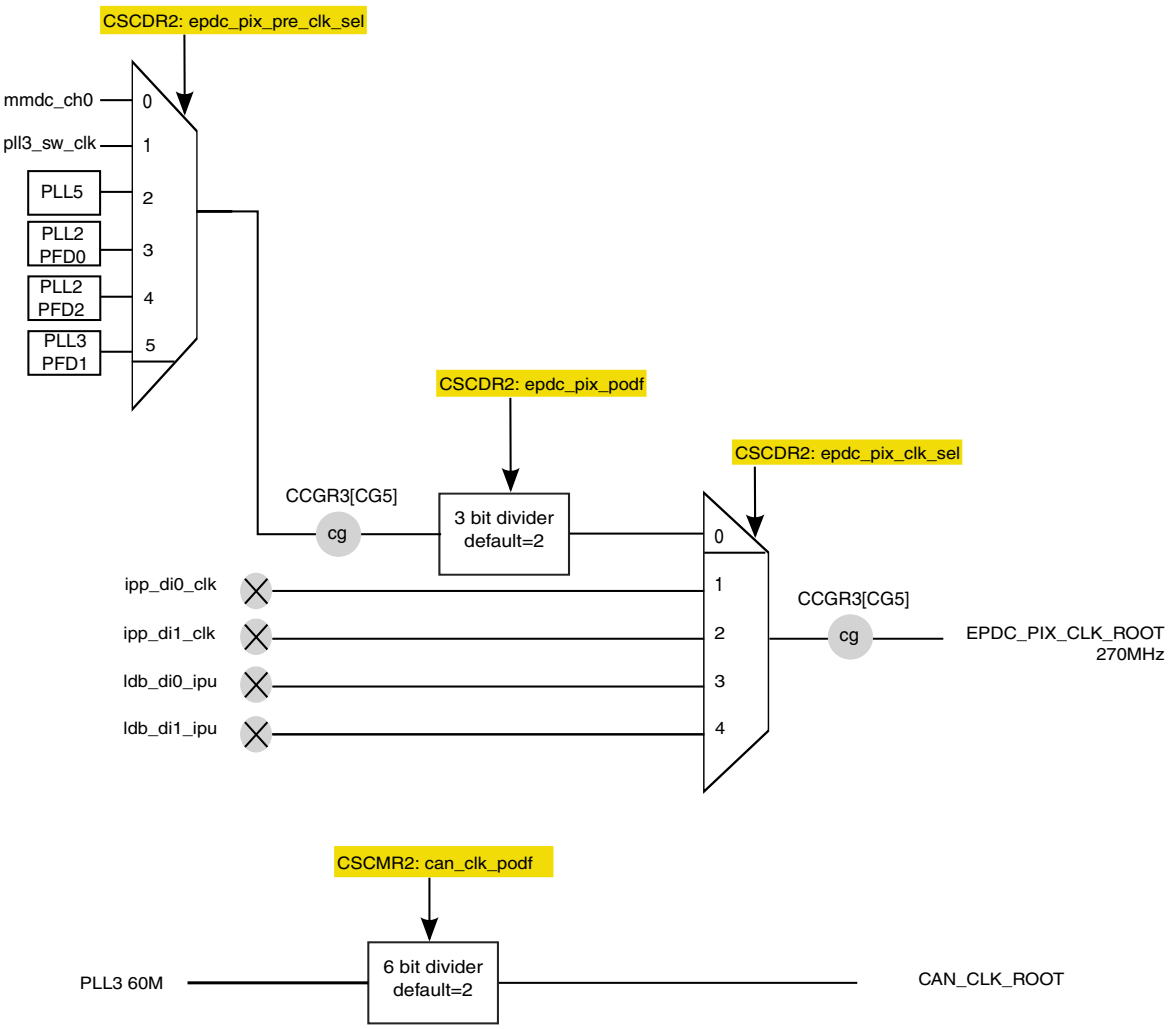


Figure 18-7. Serial clock generation



Note: See System Clocks, Gating and Override Table for PERCLK_CLK_ROOT module gating


CHSCCDR: ipu1_di0_pre_clk_sel
mmdc_ch0
pll3_sw_clk
PLL5
PLL2 PFD0
PLL2 PFD2
PLL3 PFD1
0
1
2
3
4
5
CHSCCDR: ipu1_di0_podf
CCGR3[CG1]
cg
3 bit divider default=3
CHSCCDR: ipu1_di0_clk_sel
ipp_di0_clk
ipp_di1_clk
ldb_di0_ipu
ldb_di1_ipu
CCGR3[CG1]
IPU1_DI0_CLK_ROOT
CHSCCDR: ipu1_di1_pre_clk_sel
CHSCCDR: ipu1_di1_podf
CCGR3[CG2]
CHSCCDR: ipu1_di1_clk_sel
CCGR3[CG2]
IPU1_DI1_CLK_ROOT


CSCDR2: epdc_pix_pre_clk_sel
mmdc_ch0
pll3_sw_clk
PLL5
PLL2 PFD0
PLL2 PFD2
PLL3 PFD1
0
1
2
3
4
5
CSCDR2: epdc_pix_podf
CSCDR2: epdc_pix_clk_sel
CCGR3[CG5]
cg
3 bit divider default=2
ipp_di0_clk
ipp_di1_clk
ldb_di0_ipu
ldb_di1_ipu
CCGR3[CG5]
EPDC_PIX_CLK_ROOT 270MHz
CSCMR2: can_clk_podf
PLL3 60M
6 bit divider default=2
CAN_CLK_ROOT

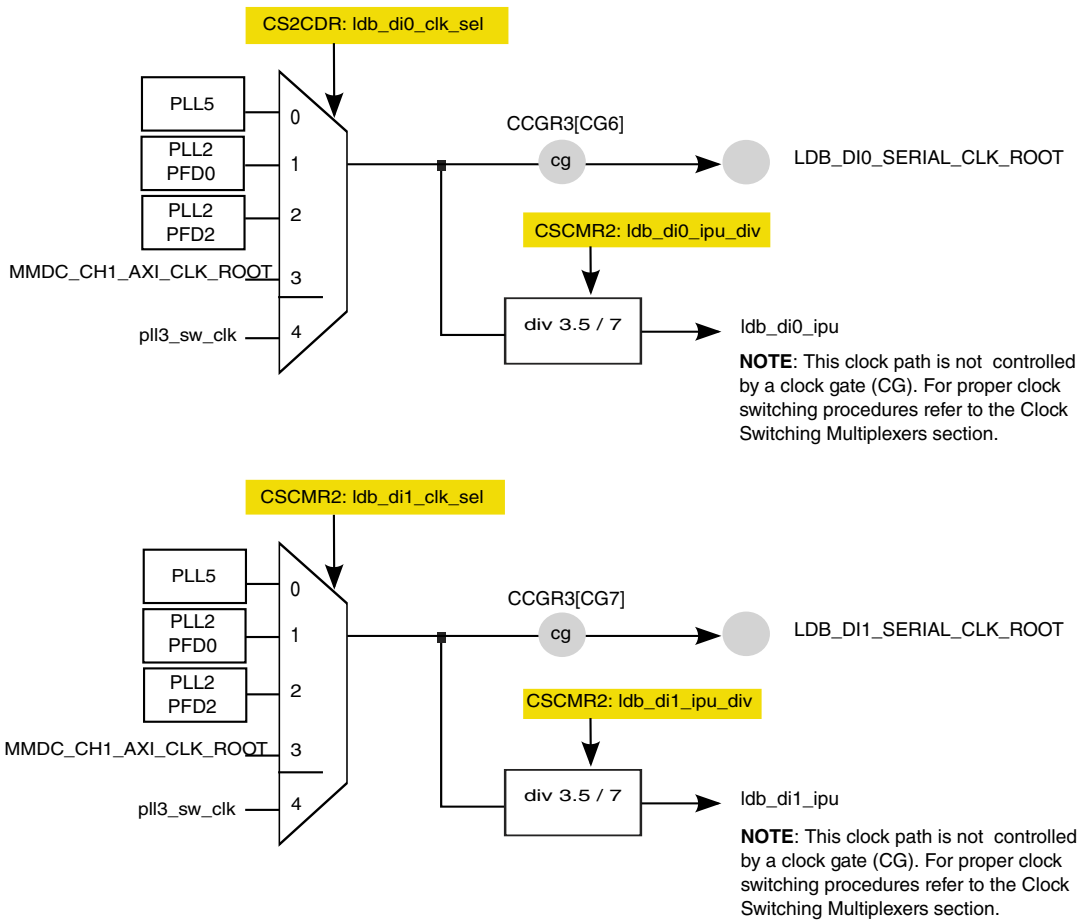
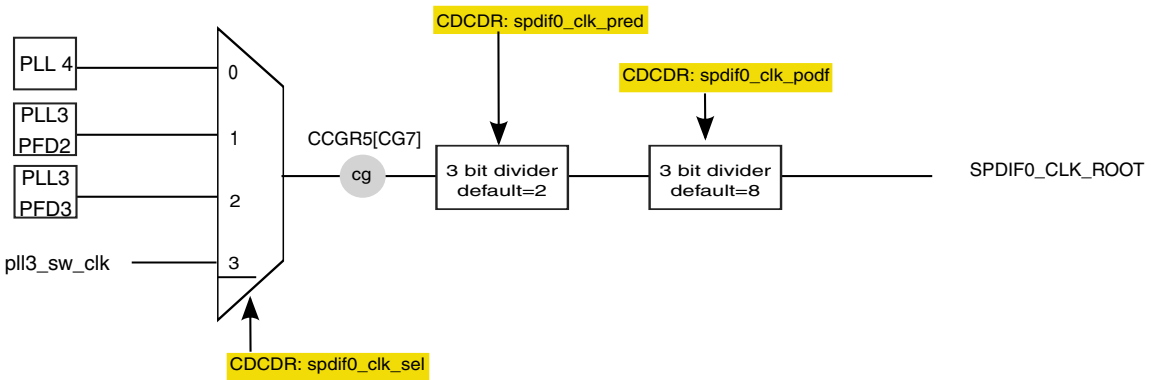
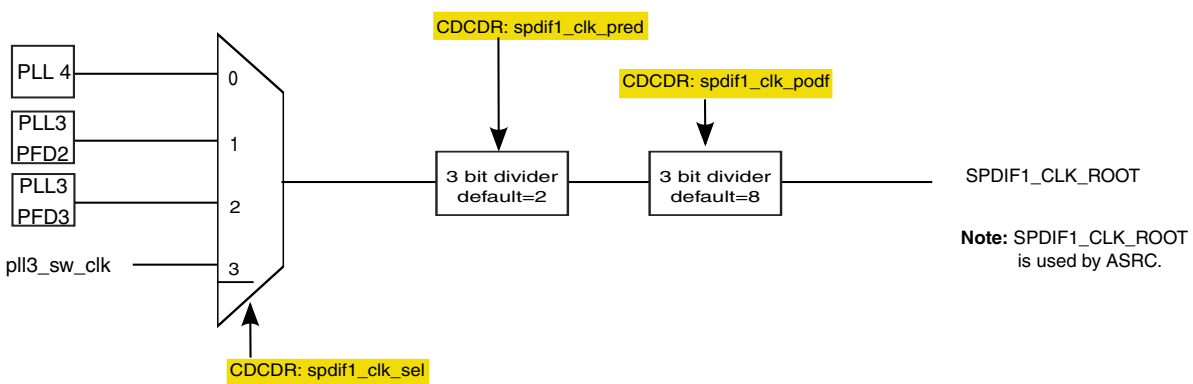
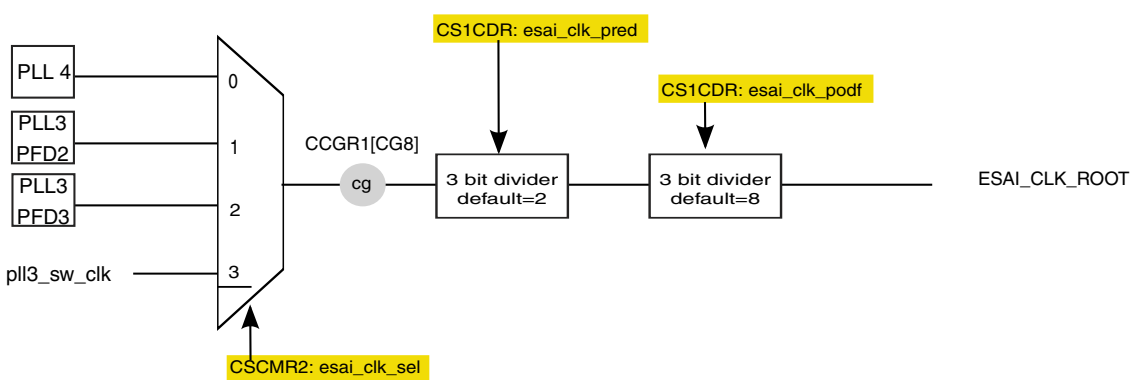


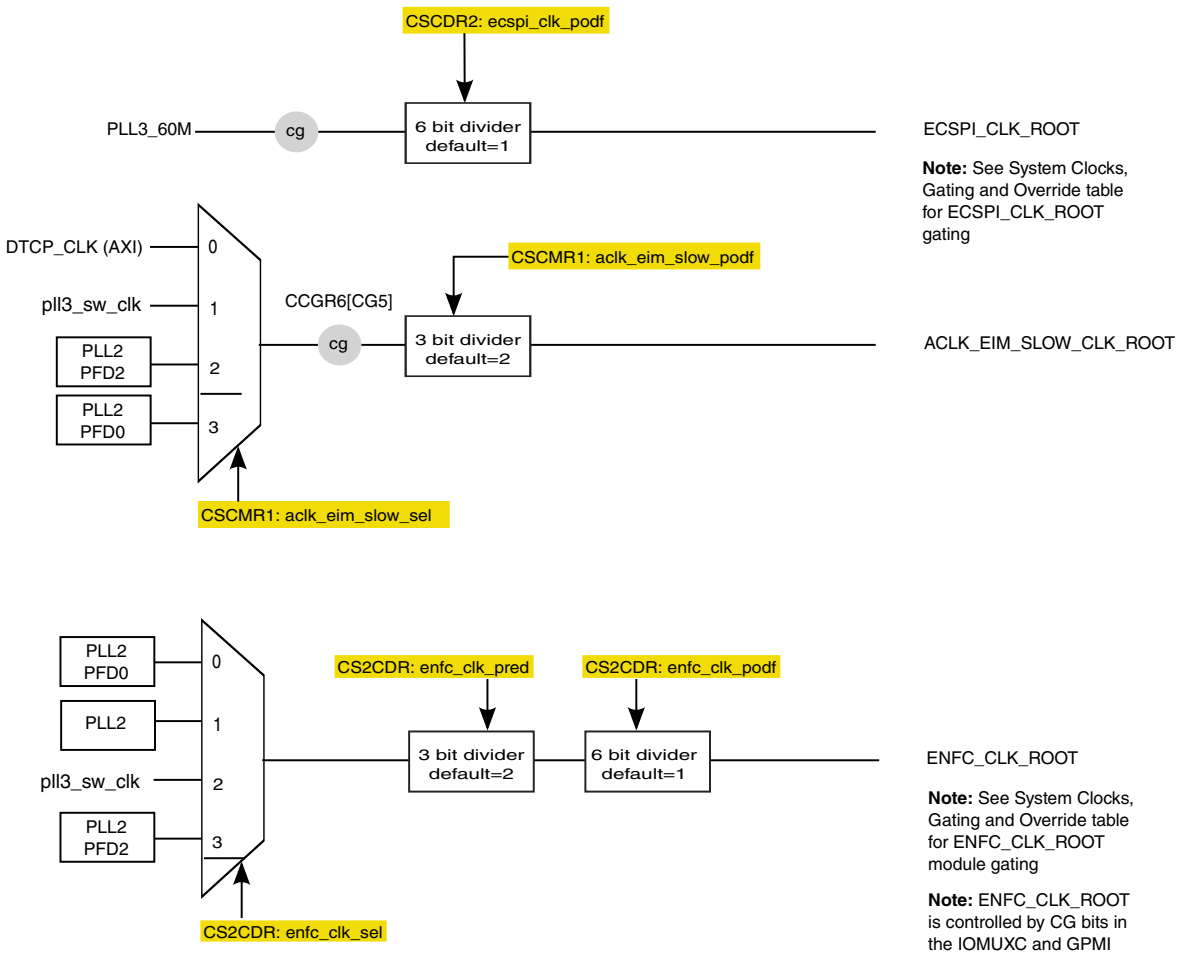
Figure 18-8. Serial clock generation





Note: SPDIF1_CLK_ROOT is used by ASRC.




CSCDR2: ecspi_clk_podf
PLL3_60M
cg
6 bit divider
default=1
ECSPI_CLK_ROOT
Note: See System Clocks, Gating and Override table for ECSPI_CLK_ROOT gating
DTCP_CLK (AXI)
pll3_sw_clk
PLL2 PFD2
PLL2 PFD0
0
1
2
3
CCGR6[CG5]
CSCMR1: aclk_eim_slow_podf
cg
3 bit divider
default=2
ACLK_EIM_SLOW_CLK_ROOT
CSCMR1: aclk_eim_slow_sel
PLL2 PFD0
PLL2
pll3_sw_clk
PLL2 PFD2
0
1
2
3
CS2CDR: enfc_clk_pred
CS2CDR: enfc_clk_podf
3 bit divider
default=2
6 bit divider
default=1
ENFC_CLK_ROOT
Note: See System Clocks, Gating and Override table for ENFC_CLK_ROOT module gating
Note: ENFC_CLK_ROOT is controlled by CG bits in the IOMUXC and GPMI Modules
CS2CDR: enfc_clk_sel

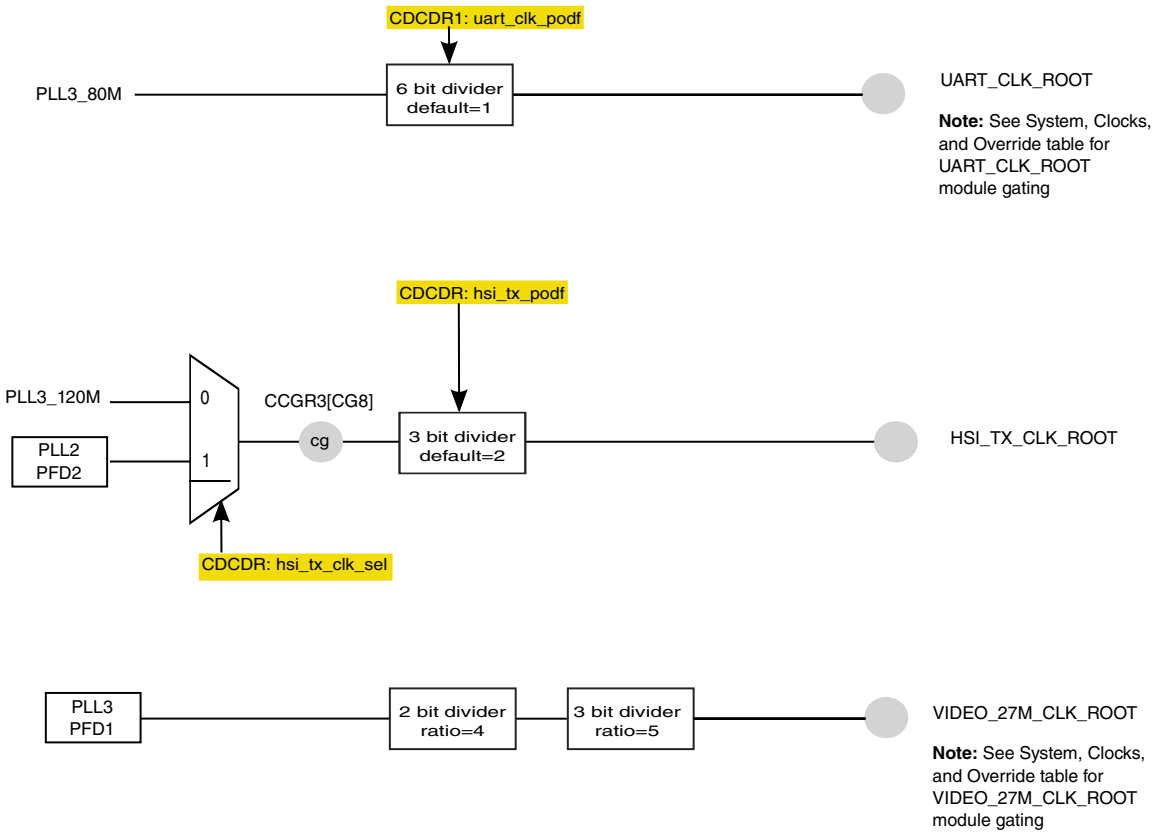


Figure 18-9. UART, HSI, Video clock generation

NOTE

All 6-bit PODF dividers found in the diagrams above can operate on low frequency.

18.5.1.5.5 Initial values controlled by the System JTAG Controller (SJC)

The initial values of the following dividers and muxes can be controlled by SJC.

In regular functional mode, the SJC will drive the reset values stated in the CCM register memory map. If SJC is programmed to change those values, then the reset value for those dividers/muxes will be taken from the SJC programability.

Software can update the changed reset value after reset sequence. The control signals and the dividers/muxes are listed below:

- [2:0] init_mmdc_ch1_axi_podf
- [2:0] init_periph2_clk2_podf
- [1:0] init_ipg_podf
- [2:0] init_ahb_podf

- [2:0] init_axi_podf
- [2:0] init_mmdc_ch0_axi_podf
- [2:0] init_periph_clk2_podf
- init_periph_clk_sel
- init_periph2_clk_sel

18.5.1.5.6 Divider change handshake

Modifying the following dividers will start the handshake with MMDC CH1 and/or CH0.

- mmdc_ch0_axi_podf
- mmdc_ch1_axi_podf
- periph_clk_sel
- periph2_clk_sel
- arm_podf
- axi_podf
- ahb_podf

The dividers listed above are designed with a handshake. For dividers without a handshake design, the following sequence must be performed when updating PODF value:

1. Gate the output clock off before updating PODF value.
2. Gate the output clock on after the PODF value is updated and stable.

To update the PODF value without gating the output clock off will cause unpredictable results such as no clock output.

18.5.1.6 Disabling / Enabling PLLs

PLL disabling and enabling is done via analog module.

Before disabling a PLL using the analog registers, software should first move all the clocks generated from that specific PLL to another source. This alternate source could be another PLL, or a PFD driven by another PLL. Alternatively, software can bypass the PLL and use the PLL reference clock (usually 24 MHz) as the output clock. Bypassing the PLL is done by setting the analog BYPASS bit in the control register for that PLL.

18.5.1.7 Clock Switching Multiplexers

There are a multitude of multiplexers available throughout the clock generation logic that provide alternate clock sources for the system clocks controlled by the CCM. The CCM uses several synchronous glitchless clock multiplexers as well as asynchronous glitchy clock multiplexers.

Synchronous muxes ensure there are no glitches between the transition of two asynchronous clocks and that there will be no pulses that are of a frequency higher than either input clock. For the synchronous multiplexer to work properly, both the current clock and the clock to be selected must remain active during the entire selection process.

There are several glitchless (synchronous) muxes used in the CCM. The table below lists the muxes and the respective control bits.

Table 18-5. Glitchless Multiplexers

Glitchless Mux	Mux Select Bit	Handshake Bit
periph_clk_mux	CBCDR[periph_clk_sel]	CDHIPR[periph_clk_sel_busy]
periph2_clk_mux	CBCDR[periph2_clk_sel]	CDHIPR[periph2_clk_sel_busy]
axi_alt_clk_mux	CBCDR[axi_sel]	
pll3_sw_clk_mux	CCSR[pll3_sw_clk_sel]	
pll1_sw_clk_mux	CCSR[pll1_sw_clk_sel]	

NOTE

Any change of the periph_clk_sel and periph2_clk_sel sync mux select will involve handshake with the MMDC. Refer to the CCDR and CDHIPR registers for the handshake bypass and busy bits.

Table 18-6. Multiplexers

Signal	Mux Select Bit	Glitchless
pll1_sw_clk	CCSR[pll1_sw_clk_sel]	Yes
step_clk	CCSR[step_sel]	No
pll3_sw_clk	CCSR[pll3_sw_clk_sel]	Yes
axi_alt	CBCDR[axi_alt_sel]	No
AXI_CLK_ROOT	CBCDR[axi_sel]	Yes
periph_clk2	CBCMR[periph_clk2_sel]	No
periph_clk	CBCDR[periph_clk_sel]	Yes
pre_periph_clk	CBCMR[pre_periph_clk_sel]	No
periph2_clk2	CBCMR[periph2_clk2_sel]	No
pre_periph2_clk	CBCMR[pre_periph2_clk_sel]	No
periph2_clk	CBCDR[periph2_clk_sel]	Yes
PCIE_AXI_CLK_ROOT	CBCMR[pcie_axi_clk_sel]	No
IPU1_HSP_CLK_ROOT	CSCDR3[ipu1_hsp_clk_sel]	No

Table continues on the next page...

Table 18-6. Multiplexers (continued)

Signal	Mux Select Bit	Glitchless
EPDC_AXI_CLK	CSCDR3[epdc_clk_sel]	No
VDO_AXI_CLK	CBCMR[vdo_axi_clk_sel]	No
GPU2D_CORE_CLK_ROOT	CBCMR[gpu2d_core_clk_sel]	No
GPU3D_CORE_CLK_ROOT	CBCMR[gpu3d_core_clk_sel]	No
MLB_SYS_CLK	CBCMR[mlb_sys_sel]	No
VPU_AXI_CLK_ROOT	CBCMR[vpu_axi_clk_sel]	No
USDHC1_CLK_ROOT	CSCMR1[usdhc1_clk_sel]	No
USDHC2_CLK_ROOT	CSCMR1[usdhc2_clk_sel]	No
USDHC3_CLK_ROOT	CSCMR1[usdhc3_clk_sel]	No
USDHC4_CLK_ROOT	CSCMR1[usdhc4_clk_sel]	No
SSI1_CLK_ROOT	CSCMR1[ssi1_clk_sel]	No
SSI2_CLK_ROOT	CSCMR1[ssi2_clk_sel]	No
SSI3_CLK_ROOT	CSCMR1[ssi3_clk_sel]	No
IPU1_DI0_CLK_ROOT	CHSCCDR[ipu1_di0_clk_sel]	No
IPU1_DI1_CLK_ROOT	CHSCCDR[ipu1_di1_clk_sel]	No
EPDC_PIX_CLK	CSCDR2[epdc_pix_clk_sel]	No
LDB_DI0_SERIAL_CLK_ROOT	CS2CDR[ldb_di0_clk_sel]	No
LDB_DI1_SERIAL_CLK_ROOT	CS2CDR[ldb_di1_clk_sel]	No
SPDIF0_CLK_ROOT	CDCDR[spdif0_clk_sel]	No
SPDIF1_CLK_ROOT	CDCDR[spdif1_clk_sel]	No
ESAI_CLK_ROOT	CSCMR2[esai_clk_sel]	No
ACLK_EIM_SLOW_CLK_ROOT	CSCMR1[aclk_eim_slow_sel]	No
ACLK_CLK_ROOT	CSCMR1[aclk_sel]	No
ENFC_CLK_ROOT	CS2CDR[enfc_clk_sel]	No
HSI_TX_CLK_ROOT	CDCDR[hsi_tx_clk_sel]	No

For critical system bus clocks, changing the clock source can be done in the CCM using the glitchless clock muxes in Figure 18-5. In the figure, the thick bar on the input side indicates the glitchless muxes. Those without the thick bar are regular muxes (not glitchless).

For example, before disabling PLL2, software can switch the FABRIC_CLK_ROOT away from the PLL2 or one of its PFDs by programming CBCMR[PERIPH2_CLK2_SEL] and CBCDR[PERIPH2_CLK2_PODF] to provide an appropriate frequency clock, then glitchlessly switch to it by programming CBCDR[PERIPH2_CLK_SEL].

Asynchronous multiplexers or glitchy multiplexers, allow the clock to switch immediately after the multiplexer select changed. This immediate switch of two asynchronous clock domains can cause the output clock to glitch. Since both clock sources to the mux are asynchronous, switching the clocks from one source to the other can cause a glitch to be generated, regardless of the input clock source.

The output clocks to the mux are required to be gated before switching the source clock in the CCM clock mux. If the output clocks are not gated, clock glitches can propagate to the logic that follows the clock mux, causing the logic to behave unpredictably.

For serial clocks, software should first disable the module, then gate its clock in the LPCG. Then it should move the mux controlling the source of the clocks to another PLL, and reset the module and its clocks. Only then is it safe to disable the PLL. The mux for the serial clocks is not glitchless.

18.5.1.8 Low Power Clock Gating module (LPCG)

The LPCG module receives the root clocks and splits them to clock branches for each module. The clock branches are gated clocks.

The enables for those gates can come from four sources:

1. Clock enable signal from CCM - this signal is generated by configuring of the CGR bits in the CCM. It is based on the low power mode.
2. Clock enable signal from the module - this signal is generated by the module based on internal logic of the module. Not every enable signal from the module is used. For used clock enable signals from the module, CCM will generate an override signal based on a programable bit in CCM (CMEOR).
3. Clock enable signal from Reset controller (SRC) - this signal will enable the clock during the reset procedure. Please see the SRC chapter for details on the clock enable signal during reset procedure.
4. Hard-coded enable from fuse box.

These enable signals are ANDed to generate the enable signal for the gating cell.

The enable signal for the gating cell is synchronized with the clock it needs to gate in order to prevent glitches on the gated clock.

Notifications are generated for CCM to indicate when clock roots should be opened and closed. All notifications that correspond to the same clock root will be ORed to generate one notification signal to CCM for clock root gating.

The following figure describes the clock split inside the LPCG module. It describes the case of two modules; one module is without an enable signal and one is shown with an enable signal. SRC enable signals and sync flip flops are omitted from this figure.

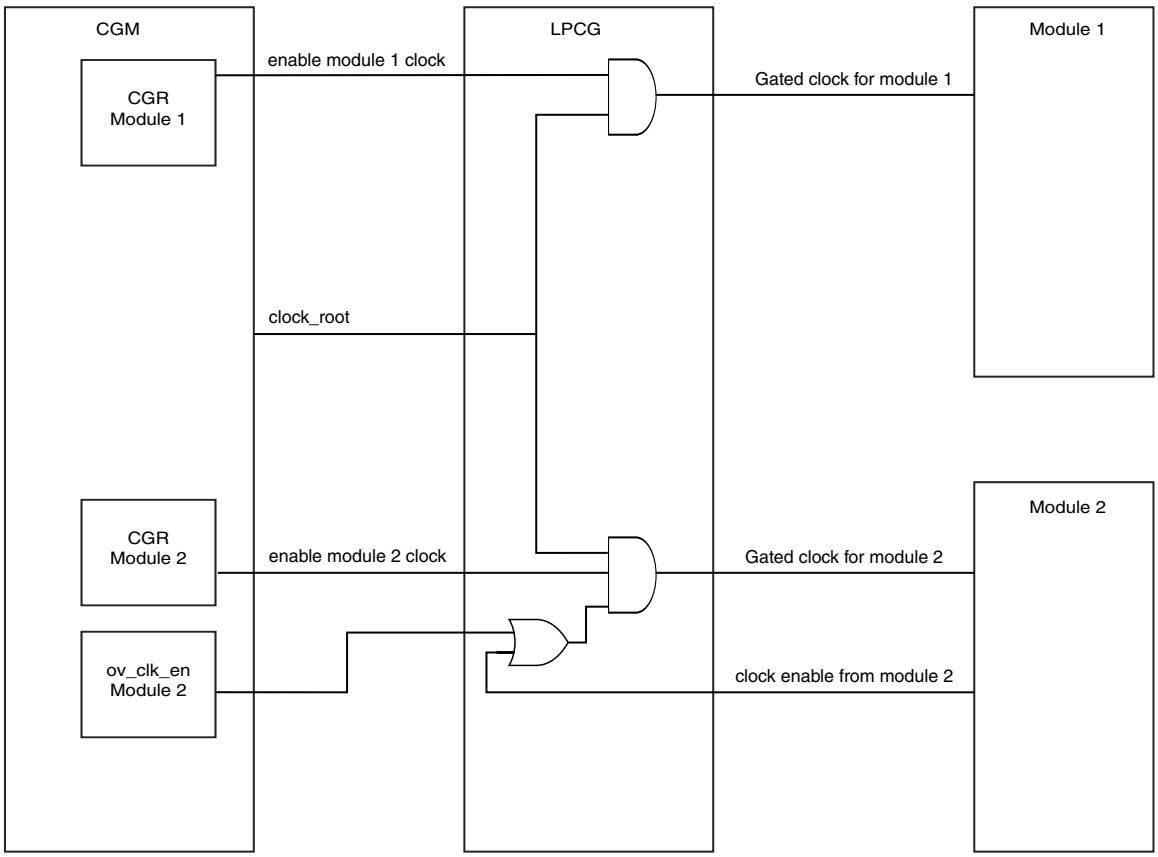


Figure 18-10. Clock split in LPCG

18.5.1.9 MMDC handshake

CCM will assert the mmdc_freq_change_req signal.

MMDC will assert the mmdc0_freq_change_ack and mmdc1_freq_change_ack signals to acknowledge that the frequency change request has been received and that the frequency can now be changed safely.

CCM will commence the actual change of division ratio of mmdc0 and/or mmdc1 dividers or apply mux change on root clocks after both of the non-masked acknowledges are asserted.

NOTE

MMDC handshakes can be masked.

18.5.2 DVFS support

When performing DVFS, the frequency shift procedure for the Arm core clock domain can be performed by software.

CPU PLL frequency and CCM ARM clock divider is controlled by CCM and CPU power domain supply voltage value is controlled by CCM_ANALOG module.

NOTE

The frequency should be shifted down first and then voltage value reduced, and vice-versa, when shifting the frequency up.

NOTE

CCM_ANALOG will not control the voltage value in Bypass mode

18.5.3 Power modes

The chip supports 3 low power modes: RUN mode, WAIT mode, STOP mode.

18.5.3.1 RUN mode

This is the normal/functional operating mode. In this mode, the CPU runs in its normal operational mode. Clocks to the modules can be gated by configuring the corresponding CCGRx bits.

18.5.3.2 WAIT mode

In this mode the CPU clock is gated. All other clocks are functional and can be gated by programming their CGR bits when all Arm cores are in WFI, and L2 cache and SCU are idle.

18.5.3.2.1 Entering WAIT mode

If the CLPCR[LPM] bit is set by software to WAIT mode, when CPU executes the next wait for interrupt (WFI) instruction, WAIT mode sequence will start.

As part of the WFI routine, alternative interrupt controller in GPC should be updated; the CPU platform interrupt controller will be disabled first by software and will be not functional, due to clock gating. Interrupts during WAIT mode are monitored by alternative interrupt controller.

After execution of the WFI routine, the CPU platform will assert idle signals for each component of the platform and CCM will gate clock to the platform.

The next actions can be programmed during WAIT mode:

1. CCM requests an acknowledge to close clocks to MMDC if its CGR bits indicate to close its clocks on WAIT mode, and if those clocks are not already closed in run mode. The request will be issued if the handshake is not bypassed by programing the CLPCR register. If the corresponding bits are set, the request signal will not be issued to the corresponding module and CCM will not wait for its acknowledge in the process of entering low power mode. After CCM receives all the acknowledge signals needed, then it will enter WAIT mode.
2. Close the clocks to the modules which were defined to be shut at WAIT mode in the CCGR bits.
3. Observability to indicate WAIT mode.

NOTE

Setting MMDC CGR bits to 01 can hang the entire system since the MMDC clock and fabric clock share the same clock root.

Any enabled interrupt assertion will start the exit from WAIT mode.

18.5.3.2.2 Exiting WAIT mode

As soon as enabled interrupt is asserted, CPU supply will be restored if CPU SRPG was applied and clocks are enabled to CPU and other modules.

18.5.3.3 STOP mode

In this mode all system clocks are stopped, along with the CPU, system buses and all PLLs. Power gating can be applied for Arm platform, GPU3D, and VPU. External supply voltage can be reduced to decrease leakage.

18.5.3.3.1 Entering STOP mode

Procedure entering STOP mode is the same, as entering WAIT mode until the moment of disabling clocks to modules. (LPM bit should be configured to STOP mode.)

After clocks to modules are gated, the following actions will be taken:

- PLLs are disabled
- CCM_PMIC_STBY_REQ asserted, if vstby bit is set
- osc_en signal is negated
- osc_pwrdn is asserted, if sbyos bit is set

Counter will be triggered after CCM_PMIC_STBY_REQ assertion to allow to external regulator or PMIC to decrease voltage until valid voltage range. On counter completion, stop_mode signal will be asserted, that will trigger disabling analog elements in anatop.

CCM's low power state machine will remain in state STOP_GPC until STOP mode is exited.

18.5.3.3.2 Exiting STOP mode

As soon as an enabled interrupt is asserted, the CCM will begin the process of exiting STOP mode.

The following will take place:

1. If vstby bit was set, deassert PMIC_STBY_REQ to notify power management IC to change voltage from standby voltage to functional voltage.
2. If well bias was enabled, deasserted well bias controls.
3. If sbyos was set, and CCM closed either external oscillator or on board oscillator, then CCM will start oscillator by asserting ref_en_b signal and deasserting cosc_pwrdown signal respectively.
4. After the number of cycles of CKILs defined in stby_count bits, wait until PMIC functional voltage is ready. This is the notification from power management IC that the voltage is ready at its functional value. Only then will CCM continue the steps.
5. Start osc. If oscillator was started, wait until oscnt has finished its counting to make sure that oscillator is ready.
6. Start PLLs. Only the PLLs that were configured to be on prior to the entrance to STOP mode will be started.
7. CCM will request GPC to restore Arm power by GPC_PUP_REQ. If power was removed from the Arm platform, GPC will notify CCM by asserting signal GPC_PUP_ACK that power to Arm is back on, and its safe to exit from STOP mode. Only then will the CCM progress to the next step.
8. After assertion of notification from src that the resets for the power gated modules has been finished, (src_power_gating_reset_done is set) negate the low power request signals to all modules and enable all module clocks including Arm clocks and CKIL sync, and return to run mode. (Clocks whose CCGR bits are not to be opened in RUN mode will not be opened; they will continued to be gated.)

After the system is in run mode, the ccm_ipg_stop and system_in_stop_mode signals negate.

18.6 CCM Memory Map/Register Definition

NOTE

The register reset values for CCM change depending on the boot configuration. See Clocks at boot time for more information.

CCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_4000	CCM Control Register (CCM_CCR)	32	R/W	0401_16FFh	18.6.1/822
20C_4004	CCM Control Divider Register (CCM_CCDR)	32	R/W	0000_0000h	18.6.2/824
20C_4008	CCM Status Register (CCM_CSR)	32	R	0000_0010h	18.6.3/825
20C_400C	CCM Clock Switcher Register (CCM_CCSR)	32	R/W	0000_0100h	18.6.4/826
20C_4010	CCM Arm Clock Root Register (CCM_CACRR)	32	R/W	0000_0000h	18.6.5/828
20C_4014	CCM Bus Clock Divider Register (CCM_CBCDR)	32	R/W	0001_8D00h	18.6.6/829
20C_4018	CCM Bus Clock Multiplexer Register (CCM_CBCMR)	32	R/W	0082_0324h	18.6.7/832
20C_401C	CCM Serial Clock Multiplexer Register 1 (CCM_CSCMR1)	32	R/W	00F0_0000h	18.6.8/834
20C_4020	CCM Serial Clock Multiplexer Register 2 (CCM_CSCMR2)	32	R/W	02B9_2F06h	18.6.9/837
20C_4024	CCM Serial Clock Divider Register 1 (CCM_CSCDR1)	32	R/W	0049_0B00h	18.6.10/ 838
20C_4028	CCM SSI1 Clock Divider Register (CCM_CS1CDR)	32	R/W	0EC1_02C1h	18.6.11/ 841
20C_402C	CCM SSI2 Clock Divider Register (CCM_CS2CDR)	32	R/W	0007_36C1h	18.6.12/ 842
20C_4030	CCM D1 Clock Divider Register (CCM_CDCDR)	32	R/W	33F7_1F92h	18.6.13/ 844
20C_4034	CCM HSC Clock Divider Register (CCM_CHSCCDR)	32	R/W	0002_A150h	18.6.14/ 846
20C_4038	CCM Serial Clock Divider Register 2 (CCM_CSCDR2)	32	R/W	0002_9B48h	18.6.15/ 849
20C_403C	CCM Serial Clock Divider Register 3 (CCM_CSCDR3)	32	R/W	0001_0841h	18.6.16/ 851
20C_4048	CCM Divider Handshake In-Process Register (CCM_CDHIPR)	32	R	0000_0000h	18.6.17/ 853
20C_4054	CCM Low Power Control Register (CCM_CLPCR)	32	R/W	0000_0079h	18.6.18/ 856

Table continues on the next page...

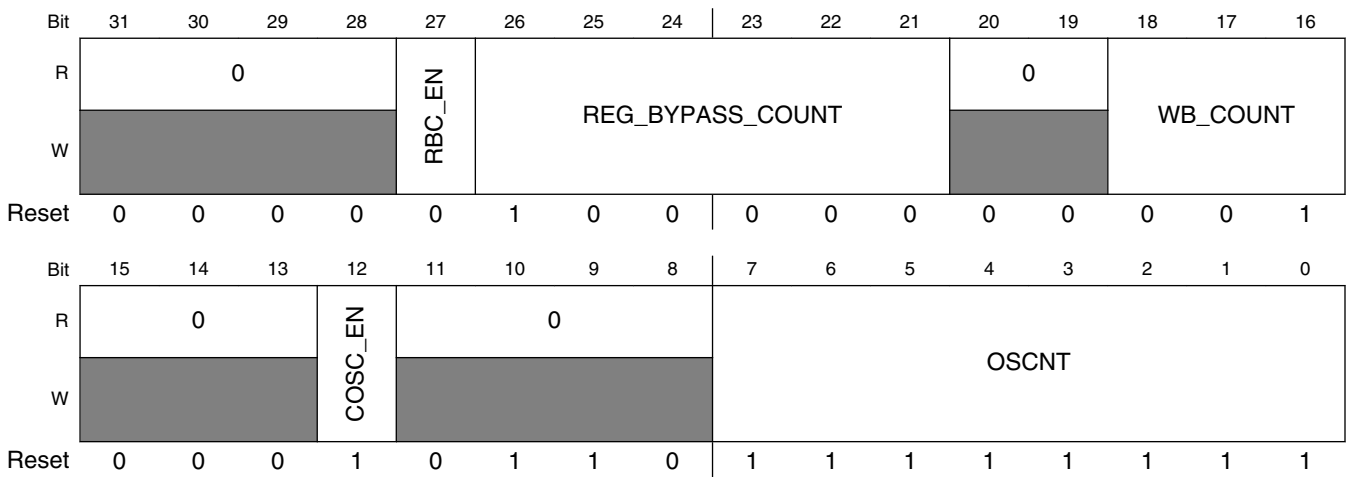
CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_4058	CCM Interrupt Status Register (CCM_CISR)	32	w1c	0000_0000h	18.6.19/ 859
20C_405C	CCM Interrupt Mask Register (CCM_CIMR)	32	R/W	FFFF_FFFFh	18.6.20/ 862
20C_4060	CCM Clock Output Source Register (CCM_CCOSR)	32	R/W	000A_0001h	18.6.21/ 865
20C_4064	CCM General Purpose Register (CCM_CGPR)	32	R/W	0000_FE62h	18.6.22/ 868
20C_4068	CCM Clock Gating Register 0 (CCM_CCGR0)	32	R/W	FFFF_FFFFh	18.6.23/ 869
20C_406C	CCM Clock Gating Register 1 (CCM_CCGR1)	32	R/W	FFFF_FFFFh	18.6.24/ 871
20C_4070	CCM Clock Gating Register 2 (CCM_CCGR2)	32	R/W	FC3F_FFFFh	18.6.25/ 872
20C_4074	CCM Clock Gating Register 3 (CCM_CCGR3)	32	R/W	FFFF_FFFFh	18.6.26/ 874
20C_4078	CCM Clock Gating Register 4 (CCM_CCGR4)	32	R/W	FFFF_FFFFh	18.6.27/ 875
20C_407C	CCM Clock Gating Register 5 (CCM_CCGR5)	32	R/W	FFFF_FFFFh	18.6.28/ 876
20C_4080	CCM Clock Gating Register 6 (CCM_CCGR6)	32	R/W	FFFF_FFFFh	18.6.29/ 878
20C_4088	CCM Module Enable Overide Register (CCM_CMEOR)	32	R/W	FFFF_FFFFh	18.6.30/ 879

18.6.1 CCM Control Register (CCM_CCR)

The figure below represents the CCM Control Register (CCR), which contains bits to control general operation of CCM. The table below provides its field descriptions.

Address: 20C_4000h base + 0h offset = 20C_4000h



CCM_CCR field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27 RBC_EN	Enable for REG_BYPASS_COUNTER. If enabled, analog_reg_bypass signal will be asserted after REG_BYPASS_COUNT clocks of CKIL, after standby voltage is requested. If standby voltage is not requested analog_reg_bypass won't be asserted, event if counter is enabled. 1 REG_BYPASS_COUNTER enabled. 0 REG_BYPASS_COUNTER disabled
26–21 REG_BYPASS_ COUNT	Counter for analog_reg_bypass signal assertion after standby voltage request by PMIC_STBY_REQ. Should be zeroed and reconfigured after exit from low power mode. REG_BYPASS_COUNT can also be used for holding off interrupts when the PGC unit is sending signals to power gate the core. 000000 no delay 000001 1 CKIL clock period delay 111111 63 CKIL clock periods delay
20–19 Reserved	This read-only field is reserved and always has the value 0.
18–16 WB_COUNT	Well Bias counter. Delay, defined by this value, counted by CKIL clock will be applied till well bias is enabled at exit from wait or stop low power mode. Counter will be used if wb_core_at_lpm or wb_per_at_lpm bits are set. Should be zeroed and reconfigured after exit from low power mode. 000 no delay

Table continues on the next page...

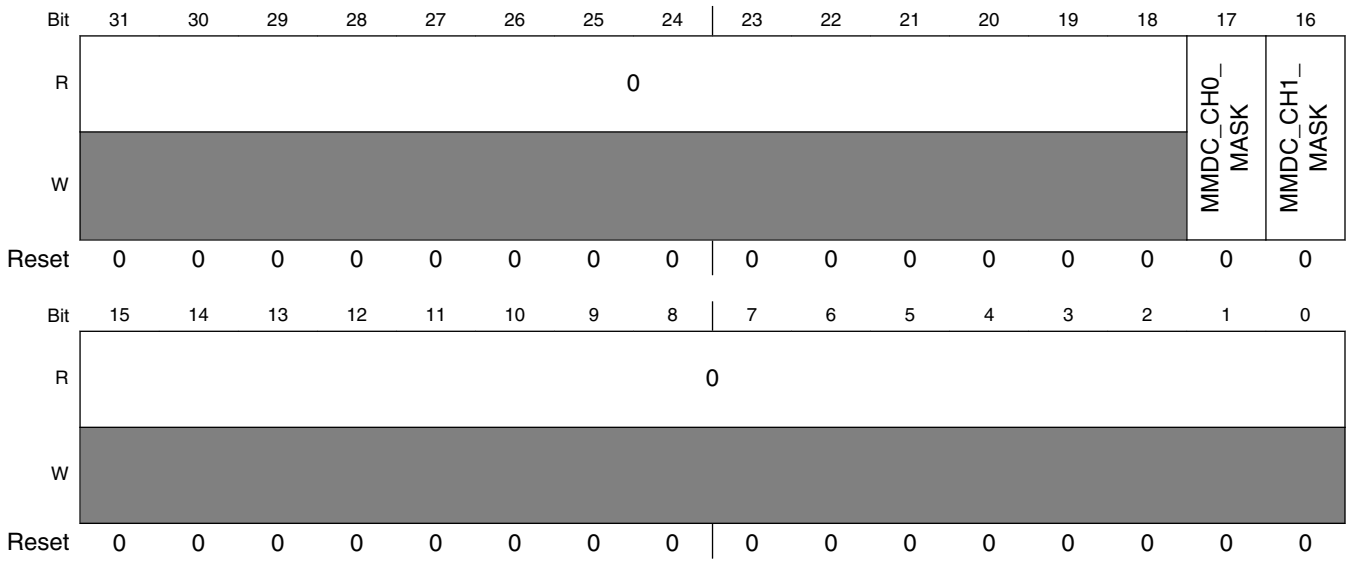
CCM_CCR field descriptions (continued)

Field	Description
	001 1 CKIL clock delay 111 7 CKIL clocks delay
15–13 Reserved	This read-only field is reserved and always has the value 0.
12 COSC_EN	On chip oscillator enable bit - this bit value is reflected on the output cosc_en. The system will start with on chip oscillator enabled to supply source for the PLLs. Software can change this bit if a transition to the bypass PLL clocks was performed for all the PLLs. In cases that this bit is changed from '0' to '1' then CCM will enable the on chip oscillator and after counting oscnt ckil clock cycles it will notify that on chip oscillator is ready by an interrupt cosc_ready and by status bit cosc_ready. The cosc_en bit should be changed only when on chip oscillator is not chosen as the clock source. 0 disable on chip oscillator 1 enable on chip oscillator
11–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
OSCNT	Oscillator ready counter value. These bits define value of 32KHz counter, that serve as counter for oscillator lock time. This is used for oscillator lock time. Current estimation is ~5ms. This counter will be used in ignition sequence and in wake from stop sequence if sbyos bit was defined, to notify that on chip oscillator output is ready for the dpll_ip to use and only then the gate in dpll_ip can be opened. 00000000 count 1 ckil 11111111 count 256 ckil's

18.6.2 CCM Control Divider Register (CCM_CCDR)

The figure below represents the CCM Control Divider Register (CCDR), which contains bits that control the loading of the dividers that need handshake with the modules they affect. The table below provides its field descriptions.

Address: 20C_4000h base + 4h offset = 20C_4004h



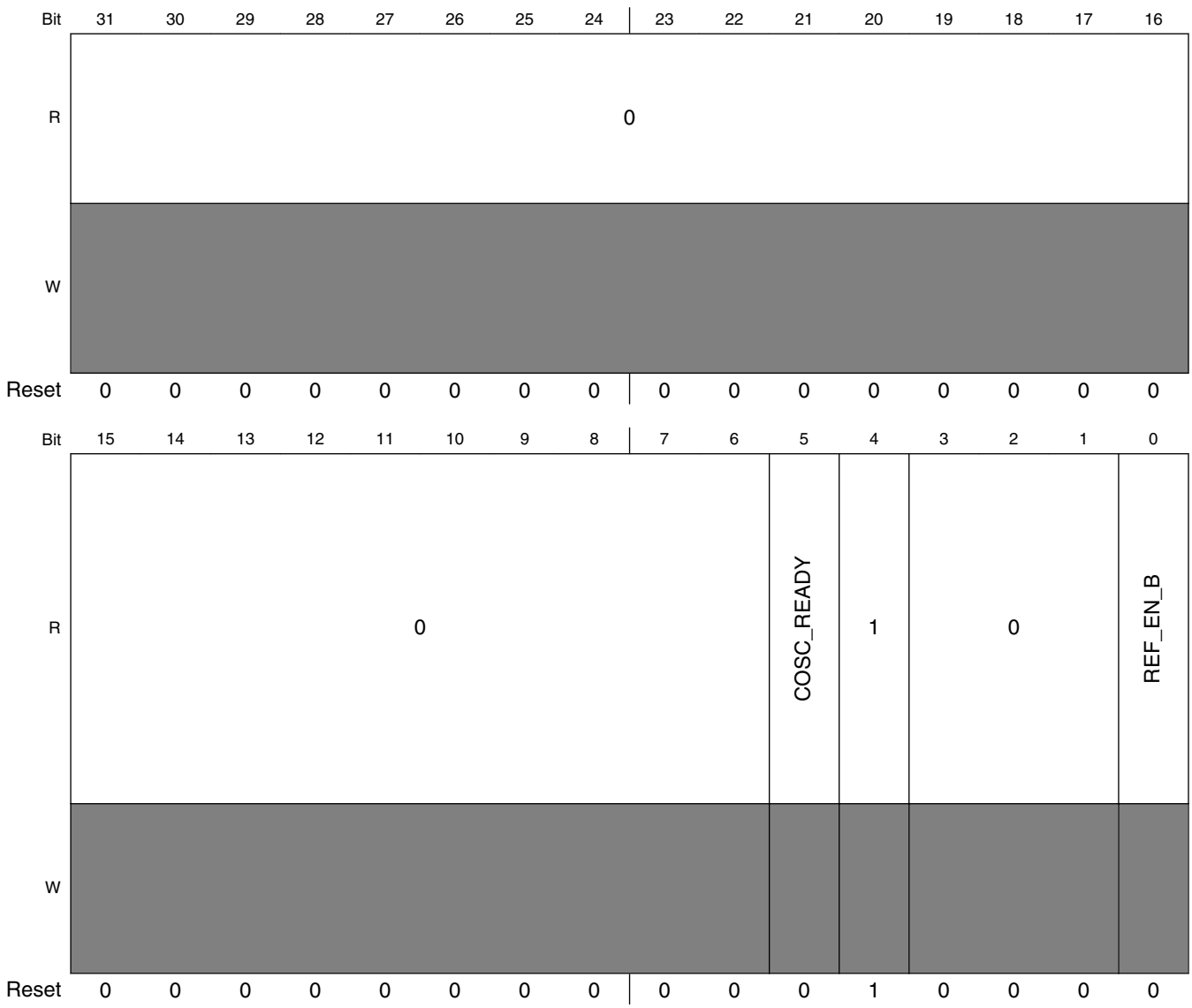
CCM_CCDR field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value 0.
17 MMDC_CH0_MASK	During divider ratio mmdc_ch0_axi_podf change or sync mux periph_clk_sel change (but not jtag) or SRC request during warm reset, mask handshake with mmdc_ch0 module. This field value should be set to 1 for normal operation. 0 allow handshake with mmdc_ch0 module 1 mask handshake with mmdc_ch0. Request signal will not be generated.
16 MMDC_CH1_MASK	During divider ratio mmdc_ch1_axi_podf change or sync mux periph2_clk_sel change (but not jtag) or SRC request during warm reset, mask handshake with mmdc_ch1 module. 0 Allow handshake with mmdc_ch1 module. 1 Mask handshake with mmdc_ch1. Request signal will not be generated.
Reserved	This read-only field is reserved and always has the value 0.

18.6.3 CCM Status Register (CCM_CSR)

The figure below represents the CCM status Register (CSR). The status bits are read-only bits. The table below provides its field descriptions.

Address: 20C_4000h base + 8h offset = 20C_4008h



CCM_CSR field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

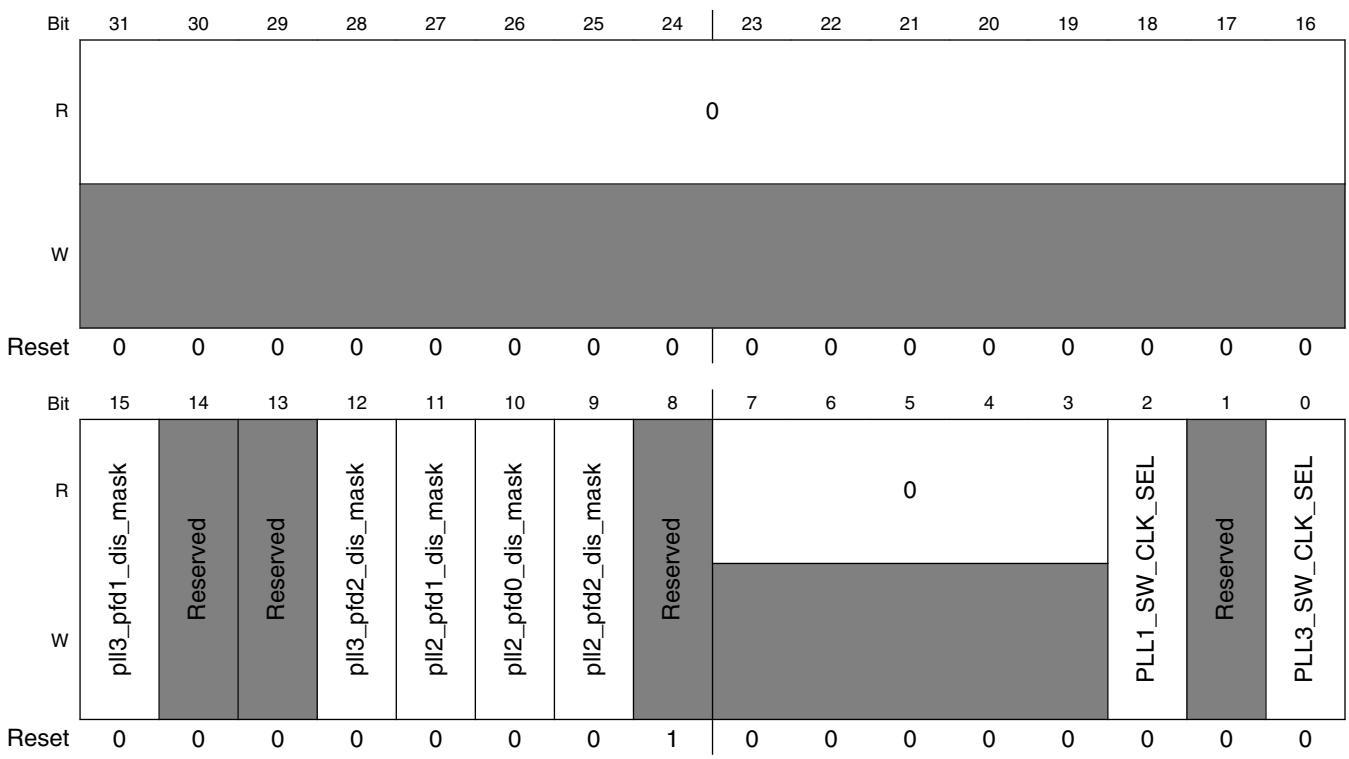
CCM_CSR field descriptions (continued)

Field	Description
5 COSC_READY	Status indication of on board oscillator. This bit will be asserted if on chip oscillator is enabled and on chip oscillator is not powered down, and if oscnt counter has finished counting. 0 on board oscillator is not ready. 1 on board oscillator is ready.
4 Reserved	This read-only field is reserved and always has the value 1.
3–1 Reserved	This read-only field is reserved and always has the value 0.
0 REF_EN_B	Status of the value of CCM_REF_EN_B output of ccm 0 value of CCM_REF_EN_B is '0' 1 value of CCM_REF_EN_B is '1'

18.6.4 CCM Clock Switcher Register (CCM_CCSR)

The figure below represents the CCM Clock Switcher register (CCSR). The CCSR register contains bits to control the switcher sub-module dividers and multiplexers. The table below provides its field descriptions.

Address: 20C_4000h base + Ch offset = 20C_400Ch



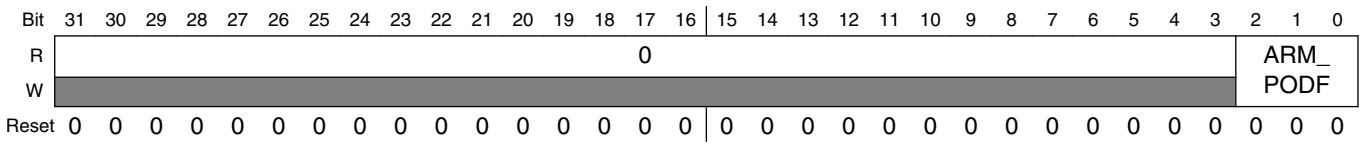
CCM_CCSR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 pll3_pfd1_dis_mask	Mask of PLL3 PFD1 auto-disable. 0 - PLL3 PFD1 disable=0 (PFD always on) 1 PLL3 PFD1 disable is managed by associated dividers disable. If all PLL3 PFD1-driven dividers are closed, PFD is disabled.
14 -	This field is reserved. Reserved
13 -	This field is reserved. Reserved
12 pll3_pfd2_dis_mask	Mask of PLL3 PFD2 auto-disable. 0 PLL3 PFD2 disable=0 (PFD always on) 1 PLL3 PFD2 disable is managed by associated dividers disable. If all PLL3 PFD2-driven dividers are closed, PFD is disabled.
11 pll2_pfd1_dis_mask	Mask of PLL2 PFD1 auto-disable. 0 PLL2 PFD1 disable=0 (PFD always on) 1 PLL2 PFD1 disable is managed by associated dividers disable. If all PLL2 PFD1-driven dividers are closed, PFD is disabled.
10 pll2_pfd0_dis_mask	Mask of PLL2 PFD0 auto-disable. 0 PLL2 PFD0 disable=0 (PFD always on) 1 PLL2 PFD0 disable is managed by associated dividers disable. If all PLL2 PFD0-driven dividers are closed, PFD is disabled.
9 pll2_pfd2_dis_mask	Mask of PLL2 PFD2 auto-disable. 0 PLL2 PFD2 disable=0 (PFD always on) 1 PLL2 PFD2 disable is managed by associated dividers disable. If all PLL2 PFD2-driven dividers are closed, PFD is disabled.
8 -	This field is reserved. Reserved
7–3 Reserved	This read-only field is reserved and always has the value 0.
2 PLL1_SW_CLK_SEL	Selects source to generate pll1_sw_clk. 0 pll1_main_clk 1 step_clk
1 -	This field is reserved. Reserved
0 PLL3_SW_CLK_SEL	Selects source to generate pll3_sw_clk. This bit should only be used for testing purposes. 0 pll3_main_clk 1 pll3 bypass clock

18.6.5 CCM Arm Clock Root Register (CCM_CACRR)

The figure below represents the CCM Arm Clock Root register (CACRR). The CACRR register contains bits to control the Arm clock root generation. The table below provides its field descriptions.

Address: 20C_4000h base + 10h offset = 20C_4010h



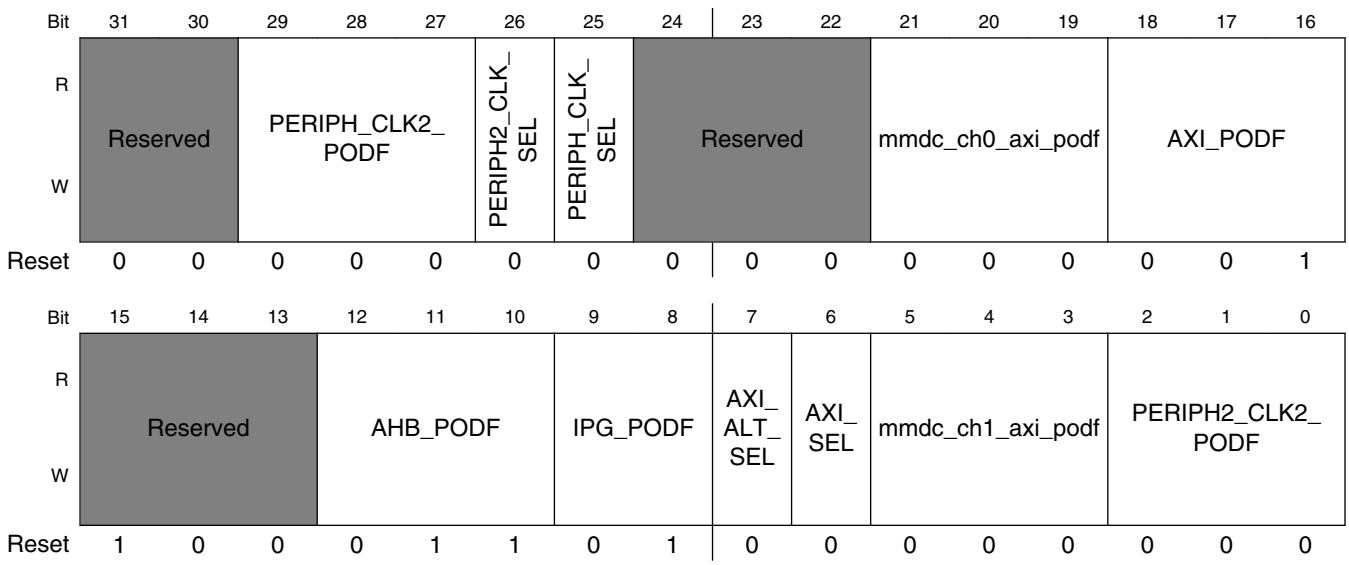
CCM_CACRR field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
ARM_PODF	Divider for Arm clock root. **NOTE:** If arm_freq_shift_divider is set to '1' then any new write to arm_podf will be held until arm_clk_switch_req signal is asserted. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8

18.6.6 CCM Bus Clock Divider Register (CCM_CBCDR)

The figure below represents the CCM Bus Clock Divider Register (CBCDR). The CBCDR register contains bits to control the clock generation sub module dividers. The table below provides its field descriptions.

Address: 20C_4000h base + 14h offset = 20C_4014h



CCM_CBCDR field descriptions

Field	Description
31–30 -	This field is reserved. Reserved
29–27 PERIPH_CLK2_ PODF	Divider for periph_clk2_podf. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
26 PERIPH2_CLK_ SEL	Selector for peripheral2 main clock (source of mmdc_ch1_clk_root). **NOTE:** Any change of this mux select will involve handshake with the MMDC. Refer to the CCDR and CDHIPR registers for the handshake bypass and busy bits.

Table continues on the next page...

CCM_CBCDR field descriptions (continued)

Field	Description
	0 PLL2 (pll2_main_clk) 1 derive clock from periph2_clk2_clk clock source.
25 PERIPH_CLK_SEL	Selector for peripheral main clock. **NOTE:** Alternative clock source should be used when PLL is relocked. For PLL relock procedure pls refer to the PLL chapter. **NOTE:** Any change of this sync mux select will involve handshake with the MMDC. Refer to the CCDR and CDHIPR registers for the handshake bypass and busy bits. 0 PLL2 (pll2_main_clk) 1 derive clock from periph_clk2_clk clock source.
24–22 -	This field is reserved. Reserved
21–19 mmdc_ch0_axi_podf	Divider for mmdc_ch0_axi podf. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
18–16 AXI_PODF	Divider for axi podf. **NOTE:** Any change of this divider might involve handshake with EMIand IPU. See CDHIPR register for the handshake busy bits. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
15–13 -	This field is reserved. Reserved
12–10 AHB_PODF	Divider for AHB PODF. **NOTE:** Any change of this divider might involve handshake with EMIand IPU. See CDHIPR register for the handshake busy bits. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6

Table continues on the next page...

CCM_CBCDR field descriptions (continued)

Field	Description
	110 divide by 7 111 divide by 8
9–8 IPG_PODF	Divider for ipg podf. **NOTE:** SDMA module will not support ratio of 1:3 and 1:4 for ahb_clk:ipg_clk. In case SDMA is used, then those ratios should not be used. 00 divide by 1 01 divide by 2 10 divide by 3 11 divide by 4
7 AXI_ALT_SEL	AXI alternative clock select 0 PLL2 PFD2 will be selected as alternative clock for AXI root clock 1 PLL3 PFD1 will be selected as alternative clock for AXI root clock
6 AXI_SEL	AXI clock source select 0 Periph_clk output will be used as AXI clock root 1 AXI alternative clock will be used as AXI clock root
5–3 mmdc_ch1_axi_podf	Divider for mmdc_ch1_axi podf. **NOTE:** This design implementation does not use MMDC_CH1_CLK_ROOT as a clock source to the MMDC. Only MMDC_CH0_CLK_ROOT is used. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
PERIPH2_CLK2_PODF	Divider for periph2_clk2 podf. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8

18.6.7 CCM Bus Clock Multiplexer Register (CCM_CBCMR)

The figure below represents the CCM Bus Clock Multiplexer Register (CBCMR). The CBCMR register contains bits to control the multiplexers that generate the bus clocks. The table below provides its field descriptions.

NOTE

Any change on the above multiplexer will have to be done while the module that its clock is affected is not functional and the respective clock is gated in LPCG. If the change will be done during operation of the module, then it is not guaranteed that the modules operation will not be harmed.

The change for arm_clk_sel should be done through sdma so that Arm will not use this clock during the change and the clock will be gated in LPCG.

Address: 20C_4000h base + 18h offset = 20C_4018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	gpu2d_core_podf			gpu3d_core_podf			mlb_sys_clk_podf			PRE_ PERIPH2_ CLK_SEL		PERIPH2_CLK2_ SEL	PRE_ PERIPH_ CLK_SEL		mlb_sys_sel	
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VPU_AXI_ CLK_SEL		PERIPH_ CLK2_SEL		vdoaxi_clk_sel	pcie_axi_clk_sel	gpu2d_core_ sel		Reserved		gpu3d_core_ clk_sel		Reserved			
Reset	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0

CCM_CBCMR field descriptions

Field	Description
31–29 gpu2d_core_podf	Divider for gpu2d_core clock. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6

Table continues on the next page...

CCM_CBCMR field descriptions (continued)

Field	Description
	110 divide by 7 111 divide by 8
28–26 gpu3d_core_podf	Divider for gpu3d_core clock. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
25–23 mlb_sys_clk_podf	Divider for mlb_sys clock. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
22–21 PRE_PERIPH2_ CLK_SEL	Selector for pre_periph2 clock multiplexer 00 derive clock from PLL2 01 derive clock from PLL2 PFD2 10 derive clock from PLL2 PFD0 11 derive clock from divided (/2) PLL2 PFD2
20 PERIPH2_CLK2_ SEL	Selector for periph2_clk2 clock multiplexer 0 derive clock from pll3_sw_clk 1 derive clock from PLL2 Main
19–18 PRE_PERIPH_ CLK_SEL	Selector for pre_periph clock multiplexer 00 derive clock from PLL2 01 derive clock from PLL2 PFD2 10 derive clock from PLL2 PFD0 11 derive clock from divided (/2) PLL2 PFD2
17–16 mlb_sys_sel	Selector for MLB clock multiplexer 00 derive clock from AXI 01 derive clock from pll3_sw_clk 10 derive clock from PLL2 PFD0 11 derive clock from PLL2 PFD2

Table continues on the next page...

CCM_CBCMR field descriptions (continued)

Field	Description
15–14 VPU_AXI_CLK_SEL	Selector for VPU axi clock multiplexer 00 derive clock from AXI 01 derive clock from PLL2 PFD2 10 derive clock from PLL2 PFD0 11 Reserved
13–12 PERIPH_CLK2_SEL	Selector for peripheral clk2 clock multiplexer 00 derive clock from pll3_sw_clk 01 derive clock from osc_clk (pll1_ref_clk) 10 derive clock from pll2_bypass_clk 11 reserved
11 vdoaxi_clk_sel	Selector for vdoaxi clock multiplexer 0 derive clock from AXI clock 1 derive clock from AHB clock
10 pcie_axi_clk_sel	Selector for pcie_axi clock multiplexer 0 derive clock from AXI clock 1 derive clock from AHB clock
9–8 gpu2d_core_sel	Selector for gpu2d_core clock multiplexer 00 derive clock from MMDC_CH0 01 derive clock from PLL3_SW_CLK 10 derive clock from PLL2 PFD1 11 derive clock from PLL3 PFD0
7–6 -	This field is reserved. Reserved
5–4 gpu3d_core_clk_sel	Selector for gpu3d_core clock multiplexer 00 derive clock from mmdc_ch0 01 derive clock from pll3_sw_clk 10 derive clock from PLL2 PFD1 11 derive clock from PLL2 PFD2
-	This field is reserved. Reserved

18.6.8 CCM Serial Clock Multiplexer Register 1 (CCM_CSCMR1)

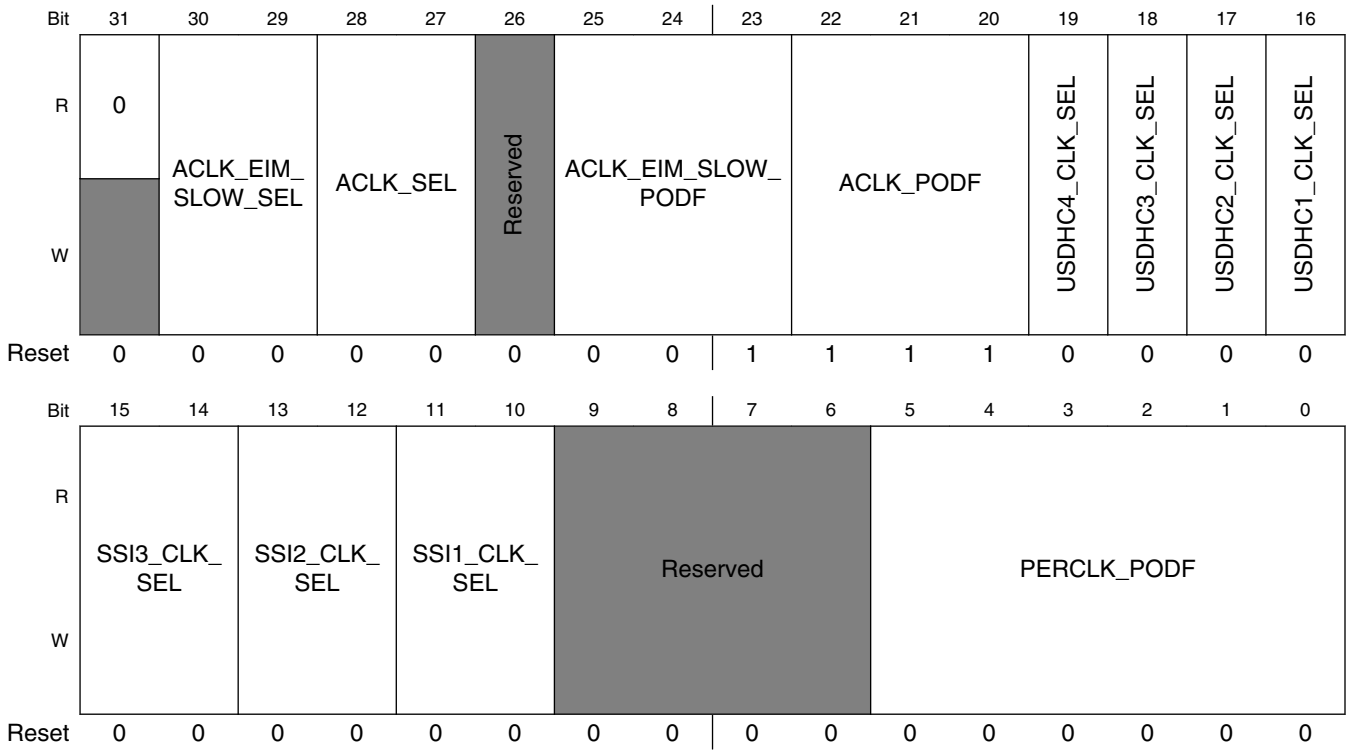
The figure below represents the CCM Serial Clock Multiplexer Register 1 (CSCMR1). The CSCMR1 register contains bits to control the multiplexers that generate the serial clocks. The table below provides its field descriptions.

NOTE

Any change on the above multiplexer will have to be done while the module that its clock is affected is not functional and

the clock is gated. If the change will be done during operation of the module, then it is not guaranteed that the modules operation will not be harmed.

Address: 20C_4000h base + 1Ch offset = 20C_401Ch



CCM_CSCMR1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 ACLK_EIM_SLOW_SEL	Selector for aclk_eim_slow root clock multiplexer 00 derive clock from AXI 01 derive clock from pll3_sw_clk 10 derive clock from PLL2 PFD2 11 derive clock from PLL2 PFD0
28–27 ACLK_SEL	Selector for aclk root clock multiplexer 00 derive clock from PLL2 PFD2 01 derive clock from pll3_sw_clk 10 derive clock from AXI 11 derive clock from PLL2 PFD0
26 -	This field is reserved. Reserved
25–23 ACLK_EIM_SLOW_PODF	Divider for aclk_eim_slow clock root. 000 divide by 1

Table continues on the next page...

CCM_CSCMR1 field descriptions (continued)

Field	Description
	001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
22–20 ACLK_PODF	Divider for aclk clock root. **NOTE:** These bits are inverted between R/W and are not sequential. 000 divide by 7 (Read value 110) 001 divide by 8 (Read value 111) 010 divide by 5 (Read value 100) 011 divide by 6 (Read value 101) 100 divide by 3 (Read value 010) 101 divide by 4 (Read value 011) 110 divide by 1 (Read value 000) 111 divide by 2 (Read value 001)
19 USDHC4_CLK_SEL	Selector for usdhc4 clock multiplexer 0 derive clock from PLL2 PFD2 1 derive clock from PLL2 PFD0
18 USDHC3_CLK_SEL	Selector for usdhc3 clock multiplexer 0 derive clock from PLL2 PFD2 1 derive clock from PLL2 PFD0
17 USDHC2_CLK_SEL	Selector for usdhc2 clock multiplexer 0 derive clock from PLL2 PFD2 1 derive clock from PLL2 PFD0
16 USDHC1_CLK_SEL	Selector for usdhc1 clock multiplexer 0 derive clock from PLL2 PFD2 1 derive clock from PLL2 PFD0
15–14 SSI3_CLK_SEL	Selector for ssi3 clock multiplexer 00 derive clock from PLL3 PFD2 10 derive clock from PLL4 11 Reserved
13–12 SSI2_CLK_SEL	Selector for ssi2 clock multiplexer 00 derive clock from PLL3 PFD2 10 derive clock from PLL4 11 Reserved
11–10 SSI1_CLK_SEL	Selector for ssi1 clock multiplexer 00 derive clock from PLL3 PFD2

Table continues on the next page...

CCM_CSCMR1 field descriptions (continued)

Field	Description
	10 derive clock from PLL4 11 Reserved
9–6 -	This field is reserved. Reserved
PERCLK_PODF	Divider for perclk podf. 000000 divide by 1 000001 divide by 2 000010 divide by 3 000011 divide by 4 000100 divide by 5 000101 divide by 6 000110 divide by 7 111111 divide by 64

18.6.9 CCM Serial Clock Multiplexer Register 2 (CCM_CSCMR2)

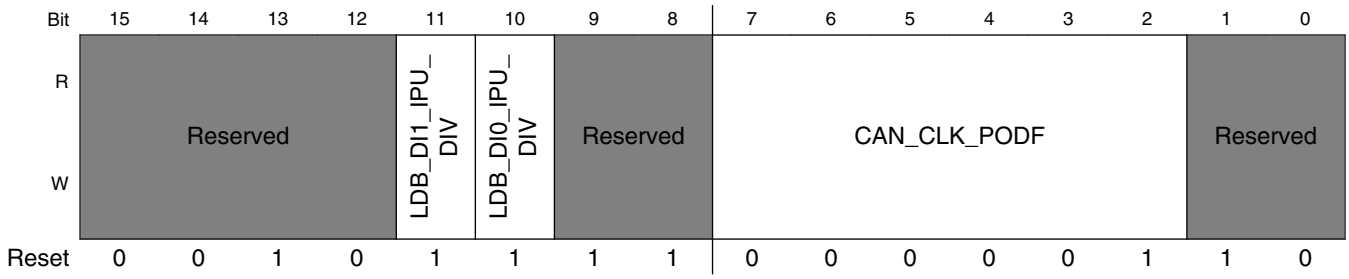
The figure below represents the CCM Serial Clock Multiplexer Register 2 (CSCMR2). The CSCMR2 register contains bits to control the multiplexers that generate the serial clocks. The table below provides its field descriptions.

NOTE

Any change on the above multiplexer will have to be done while the module that its clock is affected is not functional and the clock is gated. If the change will be done during operation of the module, then it is not guaranteed that the modules operation will not be harmed.

Address: 20C_4000h base + 20h offset = 20C_4020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											ESAI_CLK_SEL		Reserved		
Reset	0	0	0	0	0	0	1	0	1	0	1	1	1	0	0	1



CCM_CSCMR2 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20–19 ESAI_CLK_SEL	Selector for esai clock multiplexer 00 derive clock from PLL4 divided clock 01 derive clock from PLL3 PFD2 clock 01 derive clock from PLL3 PFD3 clock 11 derive clock from pll3_sw_clk
18–12 -	This field is reserved. Reserved
11 LDB_DI1_IPU_DIV	Control for divider of ldb clock for IPU di1 0 divide by 3.5 1 divide by 7
10 LDB_DI0_IPU_DIV	Control for divider of ldb clock for IPU di0 0 divide by 3.5 1 divide by 7
9–8 -	This field is reserved. Reserved
7–2 CAN_CLK_PODF	Divider for CAN clock podf. 000000 divide by 1 ... 000111 divide by 8 ... 111111 divide by 2^6
-	This field is reserved. Reserved

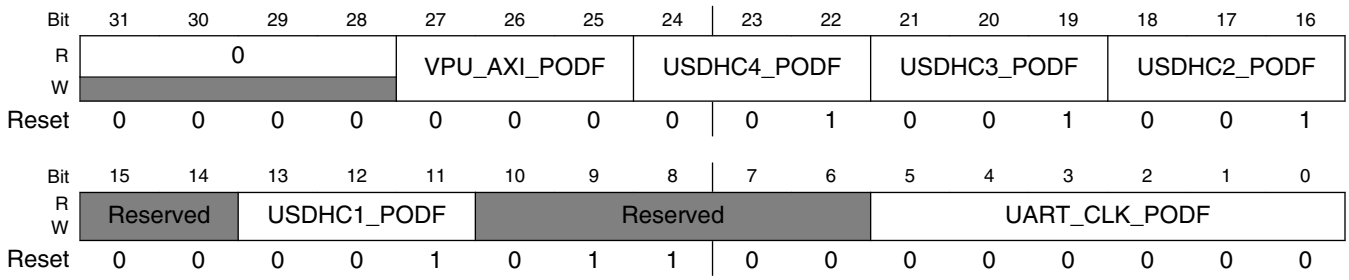
18.6.10 CCM Serial Clock Divider Register 1 (CCM_CSCDR1)

The figure below represents the CCM Serial Clock Divider Register 1 (CSCDR1). The CSCDR1 register contains bits to control the clock generation sub-module dividers. The table below provides its field descriptions.

NOTE

Any change on the above dividers will have to be done while the module that its clock is affected is not functional and the affected clock is gated. If the change will be done during operation of the module, then it is not guaranteed that the modules operation will not be harmed.

Address: 20C_4000h base + 24h offset = 20C_4024h



CCM_CSCDR1 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–25 VPU_AXI_PODF	Divider for vpu axi clock podf. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
24–22 USDHC4_PODF	Divider for usdhc4 clock pred. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
21–19 USDHC3_PODF	Divider for usdhc3 clock podf. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1

Table continues on the next page...

CCM_CSCDR1 field descriptions (continued)

Field	Description
	001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
18–16 USDHC2_PODF	Divider for usdhc2 clock. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
15–14 -	This field is reserved. Reserved
13–11 USDHC1_PODF	Divider for usdhc1 clock podf. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
10–6 -	This field is reserved. Reserved.
UART_CLK_ PODF	Divider for uart clock podf. 000000 divide by 1 111111 divide by 2^6

18.6.11 CCM SSI1 Clock Divider Register (CCM_CS1CDR)

The figure below represents the CCM ESAI, SSI1, and SSI3 Clock Divider Register (CS1CDR). The CS1CDR register contains bits to control the SSI1 and SSI3 clock generation dividers. The table below provides its field descriptions.

Address: 20C_4000h base + 28h offset = 20C_4028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				ESAI_CLK_PODF			SSI3_CLK_PRED			SSI3_CLK_PODF						0				ESAI_CLK_PRED			SSI1_CLK_PRED			SSI1_CLK_PODF					
W																																
Reset	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	1

CCM_CS1CDR field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–25 ESAI_CLK_PODF	Divider for esai clock podf. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
24–22 SSI3_CLK_PRED	Divider for ssi3 clock pred. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
21–16 SSI3_CLK_PODF	Divider for ssi3 clock podf. The input clock to this divider should be lower than 300Mhz, the predivider can be used to achieve this. 000000 divide by 1 111111 divide by 2^6
15–12 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

CCM_CS1CDR field descriptions (continued)

Field	Description
11–9 ESAI_CLK_ PRED	Divider for esai clock pred. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
8–6 SSI1_CLK_ PRED	Divider for ssi1 clock pred. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
SSI1_CLK_ PODF	Divider for ssi1 clock podf. The input clock to this divider should be lower than 300Mhz, the predivider can be used to achieve this. 000000 divide by 1 111111 divide by 2^6

18.6.12 CCM SSI2 Clock Divider Register (CCM_CS2CDR)

The figure below represents the CCM SSI2, LDB Clock Divider Register (CS2CDR). The CS2CDR register contains bits to control the SSI2 clock generation dividers, and ldb serial clocks select. The table below provides its field descriptions.

Address: 20C_4000h base + 2Ch offset = 20C_402Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					ENFC_CLK_PODF						ENFC_CLK_PRED			ENFC_CLK_SEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	LDB_DI1_CLK_SEL			LDB_DI0_CLK_SEL			SSI2_CLK_PRED			SSI2_CLK_PODF					
W																
Reset	0	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1

CCM_CS2CDR field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26–21 ENFC_CLK_ PODF	Divider for enfc clock divider. 000000 divide by 1 000001 divide by 2 111111 divide by 2^6
20–18 ENFC_CLK_ PRED	Divider for enfc clock pred divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
17–16 ENFC_CLK_SEL	Selector for enfc clock multiplexer **NOTE:** Multiplexor should be updated when output clock is gated. 00 derive clock from PLL2 PFD0 01 derive clock from PLL2 10 derive clock from pll3_sw_clk 11 derive clock from PLL2 PFD2
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 LDB_DI1_CLK_ SEL	Selector for ldb_di1 clock multiplexer **NOTE:** Multiplexor should be updated when both input and output clocks are gated. 000 pll5 clock 001 derive clock from PLL2 PFD0 010 derive clock from PLL2 PFD2 011 derive clock from mmdc_ch1 clock 100 derive clock from pll3_sw_clk 101-111 Reserved
11–9 LDB_DI0_CLK_ SEL	Selector for ldb_di1 clock multiplexer **NOTE:** Multiplexor should be updated when both input and output clocks are gated. 000 PLL5 clock 001 derive clock from PLL2 PFD0 010 derive clock from PLL2 PFD2 011 derive clock from mmdc_ch1 clock

Table continues on the next page...

CCM_CS2CDR field descriptions (continued)

Field	Description
	100 derive clock from pll3_sw_clk 101-111 Reserved
8–6 SSI2_CLK_ PRED	Divider for ssi2 clock pred. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
SSI2_CLK_ PODF	Divider for ssi2 clock podf. The input clock to this divider should be lower than 300Mhz, the predivider can be used to achieve this. **NOTE:** Divider should be updated when output clock is gated. 000000 divide by 1 111111 divide by 2^6

18.6.13 CCM D1 Clock Divider Register (CCM_CDCDR)

The figure below represents the CCM DI Clock Divider Register (CDCDR). The table below provides its field descriptions.

Address: 20C_4000h base + 30h offset = 20C_4030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HSI_TX_PODF			HSI_TX_CLK_SEL	SPDIF0_CLK_PRED			SPDIF0_CLK_PODF			SPDIF0_CLK_SEL		Reserved			
Reset	0	0	1	1	0	0	1	1	1	1	1	1	0	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	SPDIF1_CLK_PRED			SPDIF1_CLK_PODF			SPDIF1_CLK_SEL	Reserved							
Reset	0	0	0	1	1	1	1	1	1	0	0	1	0	0	1	0

CCM_CDCDR field descriptions

Field	Description
31–29 HSI_TX_PODF	Divider for hsi_tx clock podf. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
28 HSI_TX_CLK_SEL	Selector for hsi_tx clock multiplexer 0 derive from pll3_120M clock 1 derive clock from PLL2 PFD2
27–25 SPDIF0_CLK_PRED	Divider for spdif0 clock pred. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 (do not use with high input frequencies) 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
24–22 SPDIF0_CLK_PODF	Divider for spdif0 clock podf. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
21–20 SPDIF0_CLK_SEL	Selector for spdif0 clock multiplexer 00 derive clock from PLL4 divided clock 01 derive clock from PLL3 PFD2 10 derive clock from PLL3 PFD3 clock 11 derive clock from pll3_sw_clk
19–15 -	This field is reserved. Reserved

Table continues on the next page...

CCM_CDCDR field descriptions (continued)

Field	Description
14–12 SPDIF1_CLK_PRED	Divider for spdif1 clock pred. **NOTE:** Divider should be updated when output clock is gated. **NOTE:** Used for ASRC clock, not related to SPDIF module 000 divide by 1 (do not use with high input frequencies) 001 divide by 2 010 divide by 3 111 divide by 8
11–9 SPDIF1_CLK_PODF	Divider for spdif1 clock podf. **NOTE:** Divider should be updated when output clock is gated. **NOTE:** Used for ASRC clock, not related to SPDIF module 000 divide by 1 111 divide by 8
8–7 SPDIF1_CLK_SEL	Selector for spdif1 clock multiplexer **NOTE:** The CDCDR is an asynchronous glitchy mux, and a clock gate is not implemented on the output of this mux. Users must ensure input clocks to the CDCDR mux are disabled prior to changing the ASRC parent clock (SPDIF1_CLK_ROOT). 00 derive clock from PLL4 divided clock 01 derive clock from PLL3 PFD2 10 derive clock from PLL3 PFD3 clock 11 derive clock from pll3_sw_clk
-	This field is reserved. Reserved

18.6.14 CCM HSC Clock Divider Register (CCM_CHSCCDR)

The figure below represents the CCM HSC Clock Divider Register (CHSCCDR). The CHSCCDR register contains bits to control the ipu di clock generation dividers. The table below provides its field descriptions.

Address: 20C_4000h base + 34h offset = 20C_4034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														ipu1_di1_pre_clk_sel		ipu1_di1_podf				ipu1_di1_clk_sel			ipu1_di0_pre_clk_sel			ipu1_di0_podf			ipu1_di0_clk_sel		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0

CCM_CHSCCDR field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value 0.
17–15 ipu1_di1_pre_clk_sel	Selector for ipu1 di1 root clock pre-multiplexer **NOTE:** Multiplexor should only be updated when the output clock is gated. Please refer to System Clocks for the respective output clock gating bits. 000 derive clock from mmdc_ch0 clock 001 derive clock from pll3_sw_clk 010 derive clock from pll5 011 derive clock from PLL2 PFD0 100 derive clock from PLL2 PFD2 101 derive clock from PLL3 PFD1 110-111 Reserved
14–12 ipu1_di1_podf	Divider for ipu1_di clock divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
11–9 ipu1_di1_clk_sel	Selector for ipu1 di1 root clock multiplexer **NOTE:** Multiplexor should only be updated when the output clock is gated. Please refer to System Clocks for the respective output clock gating bits. 000 derive clock from divided pre-muxed ipu1 di1 clock 001 derive clock from ipp_di0_clk 010 derive clock from ipp_di1_clk 011 derive clock from ldb_di0_clk 100 derive clock from ldb_di1_clk 101-111 Reserved
8–6 ipu1_di0_pre_clk_sel	Selector for ipu1 di0 root clock pre-multiplexer **NOTE:** Multiplexor should only be updated when the output clock is gated. Please refer to System Clocks for the respective output clock gating bits. 000 derive clock from mmdc_ch0 clock 001 derive clock from pll3_sw_clk 010 derive clock from pll5 011 derive clock from PLL2 PFD0 100 derive clock from PLL2 PFD2

Table continues on the next page...

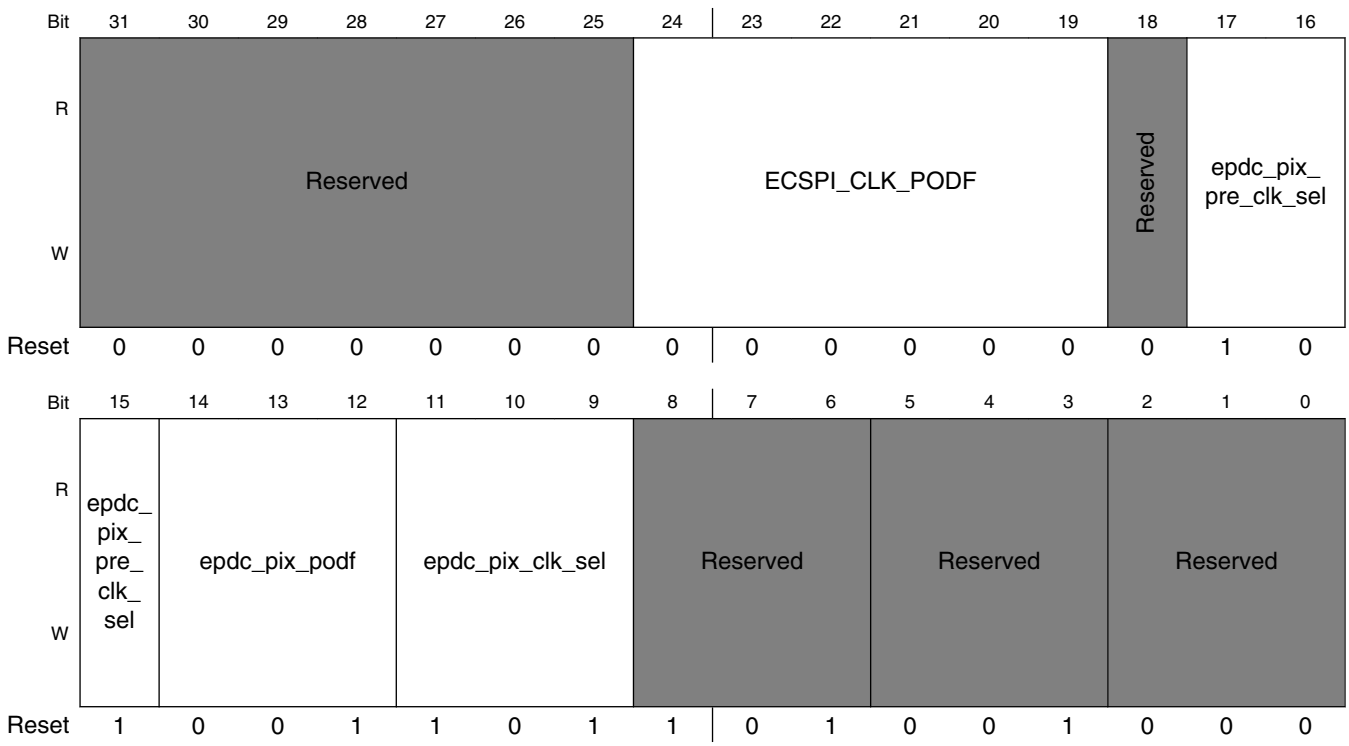
CCM_CHSCCDR field descriptions (continued)

Field	Description
	101 derive clock from PLL3 PFD1 110-111 Reserved
5–3 ipu1_di0_podf	Divider for ipu1_di0 clock divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
ipu1_di0_clk_sel	Selector for ipu1 di0 root clock multiplexer **NOTE:** Multiplexor should only be updated when the output clock is gated. Please refer to System Clocks for the respective output clock gating bits. 000 derive clock from divided pre-muxed ipu1 di0 clock 001 derive clock from ipp_di0_clk 010 derive clock from ipp_di1_clk 011 derive clock from ldb_di0_clk 100 derive clock from ldb_di1_clk 101-111 Reserved

18.6.15 CCM Serial Clock Divider Register 2 (CCM_CSCDR2)

The figure below represents the CCM Serial Clock Divider Register 2(CSCDR2). The CSCDR2 register contains bits to control the clock generation sub-module dividers. The table below provides its field descriptions.

Address: 20C_4000h base + 38h offset = 20C_4038h



CCM_CSCDR2 field descriptions

Field	Description
31–25 -	This field is reserved. Reserved
24–19 ECSPI_CLK_ PODF	Divider for ecspi clock podf. **NOTE:** Divider should be updated when output clock is gated. **NOTE:** The input clock to this divider should be lower than 300Mhz, the predivider can be used to achieve this. 000000 divide by 1 111111 divide by 2^6
18 -	This field is reserved. Reserved

Table continues on the next page...

CCM_CSCDR2 field descriptions (continued)

Field	Description
17–15 epdc_pix_pre_clk_sel	Selector for epdc_pix root clock pre-multiplexer 000 derive clock from mmdc_ch0 clock 001 derive clock from pll3_sw_clk 010 derive clock from PLL5 011 derive clock from PLL2 PFD0 100 derive clock from PLL2 PFD2 101 derive clock from PLL3 PFD1 110-111 Reserved
14–12 epdc_pix_podf	Divider for epdc_pix clock divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
11–9 epdc_pix_clk_sel	Selector for epdc_pix root clock multiplexer 000 derive clock from divided pre-muxed epdc_pix clock 001 derive clock from ipp_di0_clk 010 derive clock from ipp_di1_clk 011 derive clock from ldb_di0_clk 100 derive clock from ldb_di1_clk 101-111 Reserved
8–6 -	This field is reserved. Reserved.
5–3 -	This field is reserved. Reserved
-	This field is reserved. Reserved

18.6.16 CCM Serial Clock Divider Register 3 (CCM_CSCDR3)

The figure below represents the CCM Serial Clock Divider Register 3(CSCDR3). The CSCDR3 register contains bits to control the clock generation sub-module dividers. The table below provides its field descriptions.

Address: 20C_4000h base + 3Ch offset = 20C_403Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													epdc_podf		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	epdc_clk_sel		ipu1_hsp_podf			ipu1_hsp_ clk_sel		Reserved								
W																
Reset	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1

CCM_CSCDR3 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value 0.
18–16 epdc_podf	Divider for epdc_axi, and pxp_axi clock. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
15–14 epdc_clk_sel	Selector for epdc_axi and pxp_axi clock multiplexer 00 derive clock from mmdc_ch0 clock 01 derive clock from PLL2 PFD2 10 derive clock from pll3_120M 11 derive clock from PLL3 PFD1
13–11 ipu1_hsp_podf	Divider for ipu1_hsp clock. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3

Table continues on the next page...

CCM_CSCDR3 field descriptions (continued)

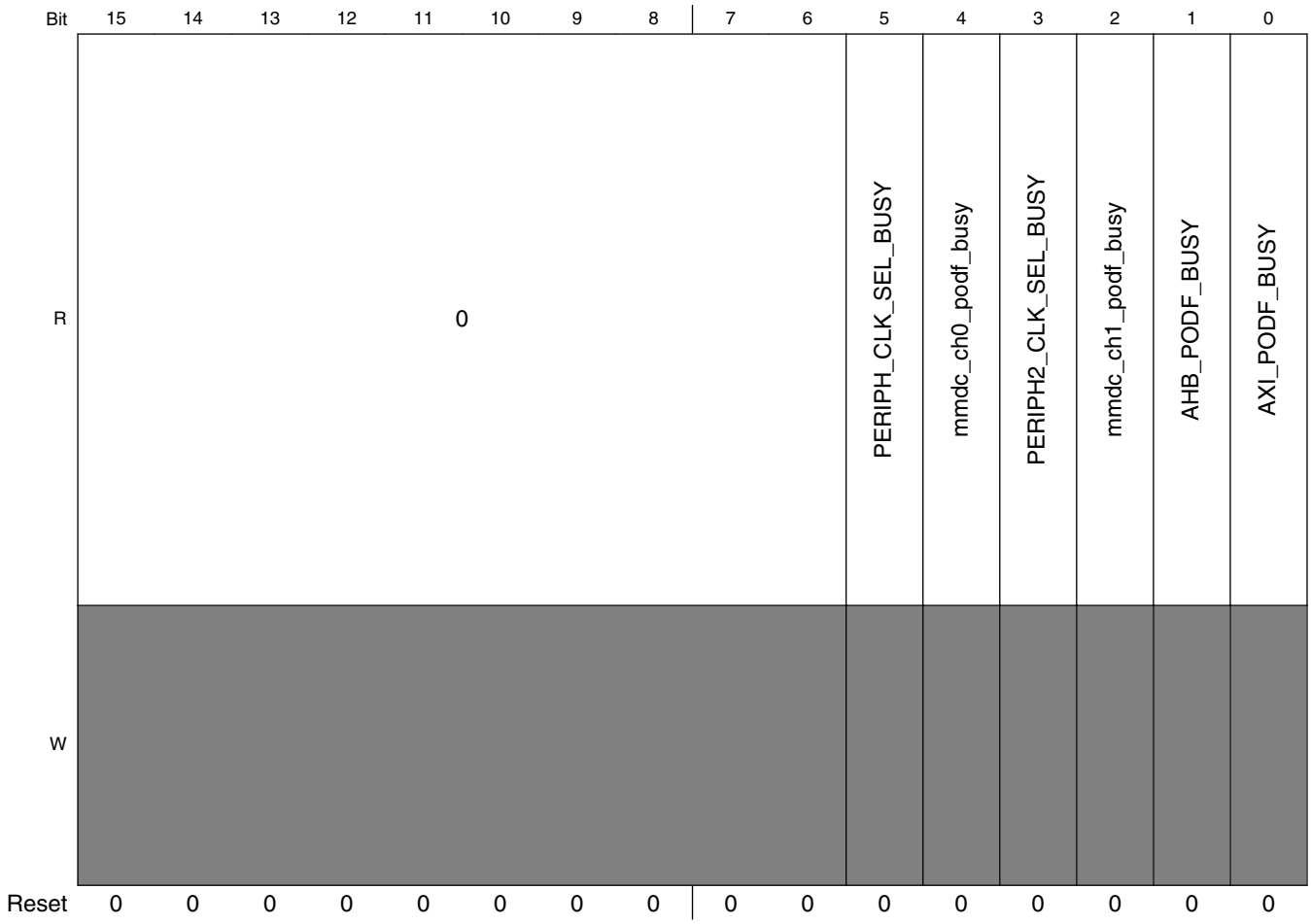
Field	Description
	011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
10–9 ipu1_hsp_clk_sel	Selector for ipu1_hsp clock multiplexer 00 derive clock from mmdc_ch0 clock 01 derive clock from PLL2 PFD2 10 derive clock from pll3_120M 11 derive clock from PLL3 PFD1
-	This field is reserved. Reserved

18.6.17 CCM Divider Handshake In-Process Register (CCM_CDHIPR)

The figure below represents the CCM Divider Handshake In-Process Register (CDHIPR). The CDHIPR register contains read-only bits that indicate that CCM is in the process of updating dividers or muxes that might need handshake with modules.

Address: 20C_4000h base + 48h offset = 20C_4048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															ARM_PODF_BUSY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



CCM_CDHIPR field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 ARM_PODF_ BUSY	Busy indicator for arm_podf. 0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process with module. The value read in the divider represents the previous value of the division factor, and after the handshake the written value of the arm_podf will be applied.
15–6 Reserved	This read-only field is reserved and always has the value 0.
5 PERIPH_CLK_ SEL_BUSY	Busy indicator for periph_clk_sel mux control. 0 mux is not busy and its value represents the actual division. 1 mux is busy with handshake process with module. The value read in the periph_clk_sel represents the previous value of select, and after the handshake periph_clk_sel value will be applied.
4 mmdc_ch0_ podf_busy	Busy indicator for mmdc_ch0_axi_podf.

Table continues on the next page...

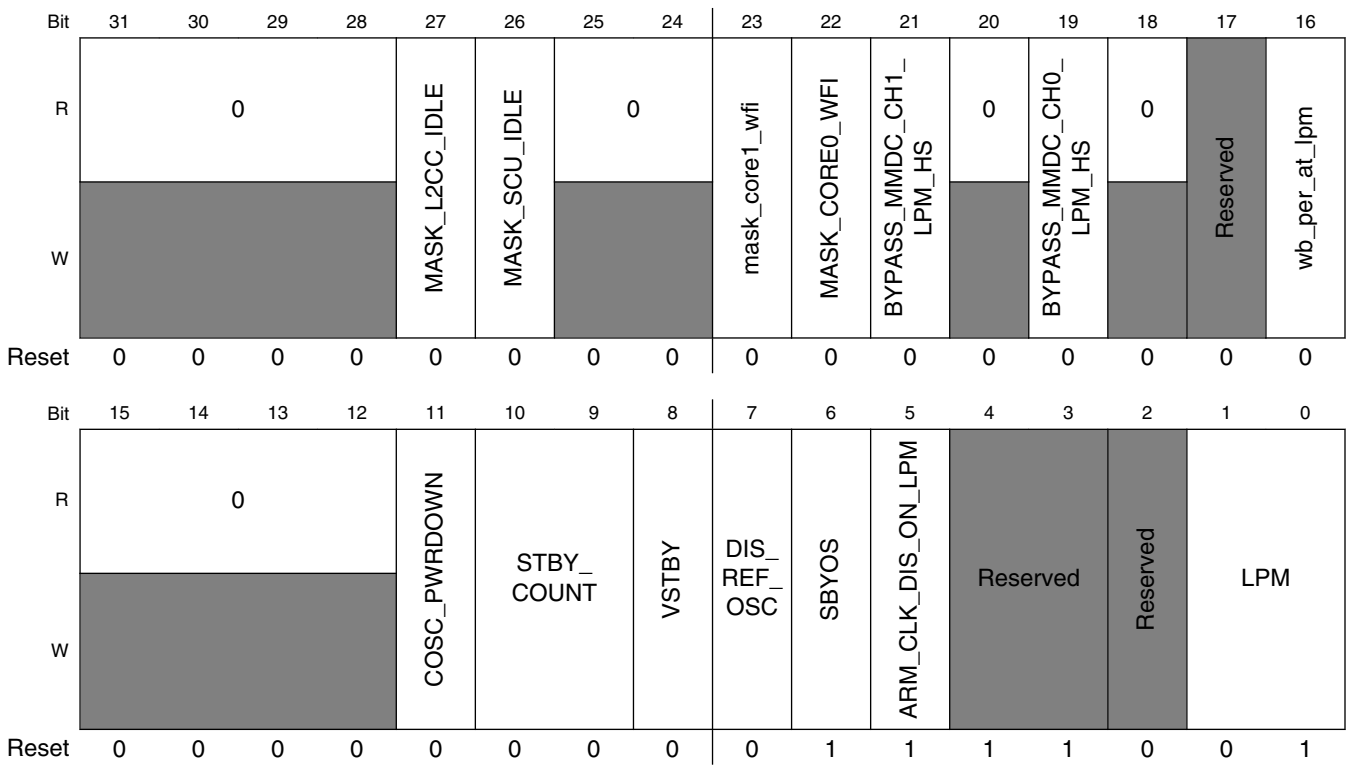
CCM_CDHIPR field descriptions (continued)

Field	Description
	0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process with module. The value read in the divider represents the previous value of the division factor, and after the handshake the written value of the mmdc_ch0_axi_podf will be applied.
3 PERIPH2_CLK_SEL_BUSY	Busy indicator for periph2_clk_sel mux control. 0 mux is not busy and its value represents the actual division. 1 mux is busy with handshake process with module. The value read in the periph2_clk_sel represents the previous value of select, and after the handshake periph2_clk_sel value will be applied.
2 mmdc_ch1_podf_busy	Busy indicator for mmdc_ch1_axi_podf. 0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process with module. The value read in the divider represents the previous value of the division factor, and after the handshake the written value of the mmdc_ch1_axi_podf will be applied.
1 AHB_PODF_BUSY	Busy indicator for ahb_podf. 0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process with module. The value read in the divider represents the previous value of the division factor, and after the handshake the written value of the ahb_podf will be applied.
0 AXI_PODF_BUSY	Busy indicator for axi_podf. 0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process with module. The value read in the divider represents the previous value of the division factor, and after the handshake the written value of the axi_podf will be applied.

18.6.18 CCM Low Power Control Register (CCM_CLPCR)

The figure below represents the CCM Low Power Control Register (CLPCR). The CLPCR register contains bits to control the low power modes operation.The table below provides its field descriptions.

Address: 20C_4000h base + 54h offset = 20C_4054h



CCM_CLPCR field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27 MASK_L2CC_IDLE	Mask L2CC IDLE for entering low power mode. **NOTE:** Assertion of all bits[27:22] will generate low power mode request 1 L2CC IDLE is masked 0 L2CC IDLE is not masked
26 MASK_SCU_IDLE	Mask SCU IDLE for entering low power mode **NOTE:** Assertion of all bits[27:22] will generate low power mode request 1 SCU IDLE is masked 0 SCU IDLE is not masked

Table continues on the next page...

CCM_CLPCR field descriptions (continued)

Field	Description
25–24 Reserved	This read-only field is reserved and always has the value 0.
23 mask_core1_wfi	Mask WFI of core1 for entering low power mode **NOTE:** Assertion of all bits[27:22] will generate low power mode request 1 WFI of core1 is masked 0 WFI of core1 is not masked
22 MASK_CORE0_WFI	Mask WFI of core0 for entering low power mode **NOTE:** Assertion of all bits[27:22] will generate low power mode request 0 WFI of core0 is not masked 1 WFI of core0 is masked
21 BYPASS_MMDC_CH1_LPM_HS	Bypass handshake with mmdc_ch1 on next entrance to low power mode (STOP or WAIT). CCM doesn't wait for the module's acknowledge. 0 Handshake with mmdc_ch1 on next entrance to low power mode will be performed. 1 Handshake with mmdc_ch1 on next entrance to low power mode will be bypassed.
20 Reserved	This read-only field is reserved and always has the value 0.
19 BYPASS_MMDC_CH0_LPM_HS	Bypass handshake with mmdc_ch0 on next entrance to low power mode (STOP or WAIT). CCM doesn't wait for the module's acknowledge. Handshake will also be bypassed, if CGR3 CG10 is set to gate fast mmdc_ch0 clock. **NOTE:** MMDC_CH0 clock is not used. This bit should always be programmed HIGH to let CCM know not to wait for the acknowledge from MMDC_CH0, due to no connection. 0 Handshake with mmdc_ch0 on next entrance to low power mode will be performed. 1 Handshake with mmdc_ch0 on next entrance to low power mode will be bypassed.
18 Reserved	This read-only field is reserved and always has the value 0.
17 -	This field is reserved. Reserved
16 wb_per_at_lpm	Enable periphery charge pump for well biasing at low power mode (stop or wait) 0 Periphery charge pump won't be enabled at STOP or WAIT low power modes 1 Periphery charge pump will be enabled at STOP or WAIT low power modes
15–12 Reserved	This read-only field is reserved and always has the value 0.
11 COSC_PWRDOWN	In run mode, software can manually control powering down of on chip oscillator, i.e. generating '1' on cosc_pwrdown signal. If software manually powered down the on chip oscillator, then sbyos functionality for on chip oscillator will be bypassed. The manual closing of onchip oscillator should be performed only in case the reference oscilator is not the source of all the clocks generation. 0 On chip oscillator will not be powered down, i.e. cosc_pwrdown = '0'. 1 On chip oscillator will be powered down, i.e. cosc_pwrdown = '1'.

Table continues on the next page...

CCM_CLPCR field descriptions (continued)

Field	Description
10–9 STBY_COUNT	Standby counter definition. These two bits define, in the case of stop exit (if VSTBY bit was set). **NOTE:** Clock cycles ratio depends on pmic_delay_scaler, defined by CGPR[0] bit. 00 CCM will wait (1*pmic_delay_scaler)+1 ckil clock cycles 01 CCM will wait (3*pmic_delay_scaler)+1 ckil clock cycles 10 CCM will wait (7*pmic_delay_scaler)+1 ckil clock cycles 11 CCM will wait (15*pmic_delay_scaler)+1 ckil clock cycles
8 VSTBY	Voltage standby request bit. This bit defines if PMIC_STBY_REQ pin, which notifies external power management IC to move from functional voltage to standby voltage, will be asserted in STOP mode. 0 Voltage will not be changed to standby voltage after next entrance to STOP mode. (PMIC_STBY_REQ will remain negated - '0') 1 Voltage will be requested to change to standby voltage after next entrance to stop mode. (PMIC_STBY_REQ will be asserted - '1').
7 DIS_REF_OSC	dis_ref_osc - in run mode, software can manually control closing of external reference oscillator clock, i.e. generating '1' on CCM_REF_EN_B signal. If software closed manually the external reference clock, then sbyos functionality will be bypassed. The manual closing of external reference oscilator should be performed only in case the reference oscilator is not the source of any clock generation. **NOTE:** When returning from stop mode, the PMIC_STBY_REQ will be deasserted (if it was asserted when entering stop mode). See stby_count bits. 0 external high frequency oscillator will be enabled, i.e. CCM_REF_EN_B = '0'. 1 external high frequency oscillator will be disabled, i.e. CCM_REF_EN_B = '1'
6 SBYOS	Standby clock oscillator bit. This bit defines if cosc_pwrdown, which power down the on chip oscillator, will be asserted in STOP mode. This bit is discarded if cosc_pwrdown='1' for the on chip oscillator. 0 On-chip oscillator will not be powered down, after next entrance to STOP mode. (CCM_REF_EN_B will remain asserted - '0' and cosc_pwrdown will remain de asserted - '0') 1 On-chip oscillator will be powered down, after next entrance to STOP mode. (CCM_REF_EN_B will be deasserted - '1' and cosc_pwrdown will be asserted - '1'). When returning from STOP mode, external oscillator will be enabled again, on-chip oscillator will return to oscillator mode, and after oscnt count, CCM will continue with the exit from the STOP mode process.
5 ARM_CLK_DIS_ON_LPM	Define if Arm clocks (arm_clk, soc_mxclk, soc_pclk, soc_dbg_pclk, vl_wrck) will be disabled on wait mode. This is useful for debug mode, when the user still wants to simulate entering wait mode and still keep Arm clock functioning. **NOTE:** Software should not enable Arm power gating in wait mode if this bit is cleared. 0 Arm clock enabled on wait mode. 1 Arm clock disabled on wait mode. .
4–3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
LPM	Setting the low power mode that system will enter on next assertion of dsm_request signal.

Table continues on the next page...

CCM_CLPCR field descriptions (continued)

Field	Description
	NOTE: Set CCM_CGPR[INT_MEM_CLK_LPM] and CCM_CGPR[1] bits to 1 when setting CCM_CLPCR[LPM] bits to 01 (WAIT Mode) or 10 (STOP mode) without power gating. CCM_CGPR[INT_MEM_CLK_LPM] and CCM_CGPR[1] bits do not have to be set for STOP mode entry. 00 Remain in run mode 01 Transfer to wait mode 10 Transfer to stop mode 11 Reserved

18.6.19 CCM Interrupt Status Register (CCM_CISR)

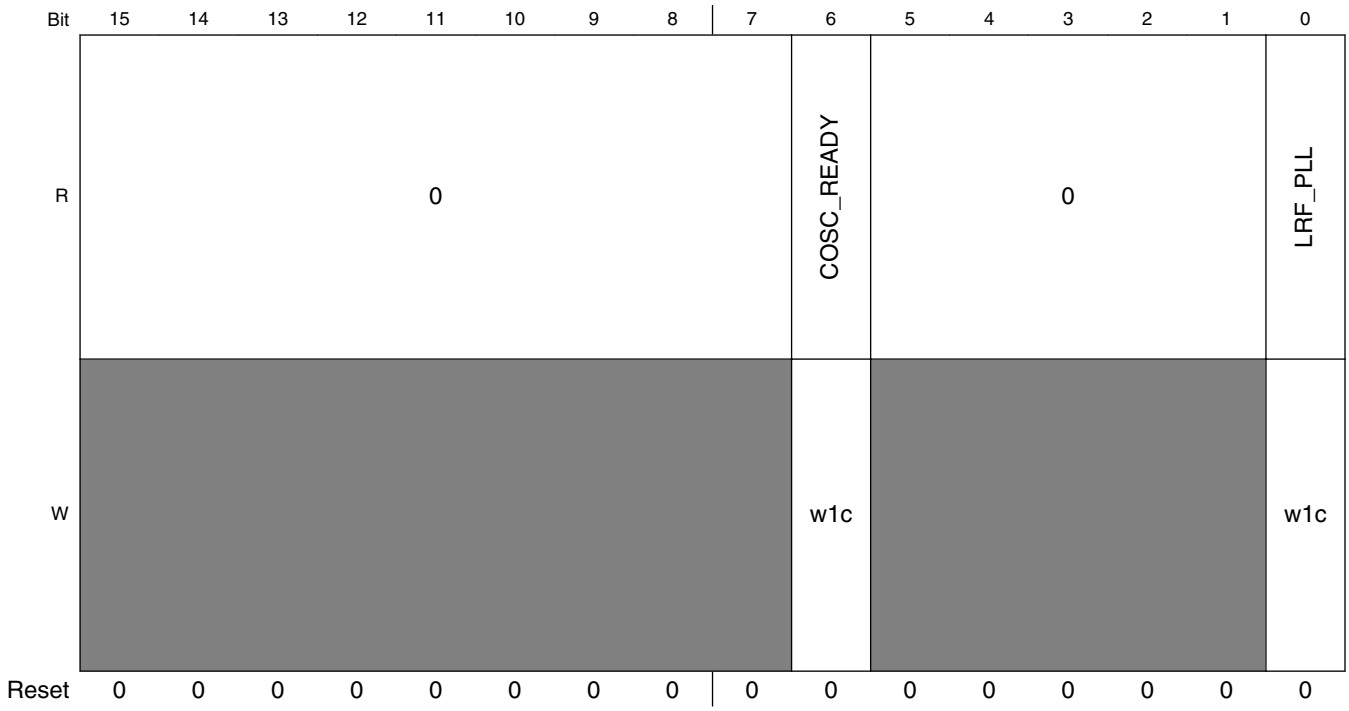
The figure below represents the CCM Interrupt Status Register (CISR). This is a write one to clear register. After an interrupt is generated, software should write one to clear it. The table below provides its field descriptions.

NOTE

CCM interrupt request 1 can be masked by CCM interrupt request 1 mask bit. CCM interrupt request 2 can be masked by CCM interrupt request 2 mask bit.

Address: 20C_4000h base + 58h offset = 20C_4058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					ARM_PODF_LOADED	0		mmdc_ch0_podf_loaded	PERIPH_CLK_SEL_LOADED	mmdc_ch1_podf_loaded	AHB_PODF_LOADED	PERIPH2_CLK_SEL_LOADED	0	AXI_PODF_LOADED	0
W						w1c			w1c	w1c	w1c	w1c	w1c		w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



CCM_CISR field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26 ARM_PODF_ LOADED	CCM interrupt request 1 generated due to frequency change of arm_podf. The interrupt will commence only if arm_podf is loaded during a DVFS operation. 0 interrupt is not generated due to frequency change of arm_podf 1 interrupt generated due to frequency change of arm_podf
25–24 Reserved	This read-only field is reserved and always has the value 0.
23 mmdc_ch0_ podf_loaded	CCM interrupt request 1 generated due to update of mmdc_ch0_axi_podf. 0 interrupt is not generated due to update of mmdc_ch0_axi_podf. 1 interrupt generated due to update of mmdc_ch0_axi_podf*
22 PERIPH_CLK_ SEL_LOADED	CCM interrupt request 1 generated due to update of periph_clk_sel. 0 interrupt is not generated due to update of periph_clk_sel. 1 interrupt generated due to update of periph_clk_sel.
21 mmdc_ch1_ podf_loaded	CCM interrupt request 1 generated due to frequency change of mmdc_ch0_podf_ **loaded** 0 interrupt is not generated due to frequency change of mmdc_ch0_podf_ **loaded** 1 interrupt generated due to frequency change of mmdc_ch0_podf_ **loaded**
20 AHB_PODF_ LOADED	CCM interrupt request 1 generated due to frequency change of ahb_podf 0 interrupt is not generated due to frequency change of ahb_podf 1 interrupt generated due to frequency change of ahb_podf

Table continues on the next page...

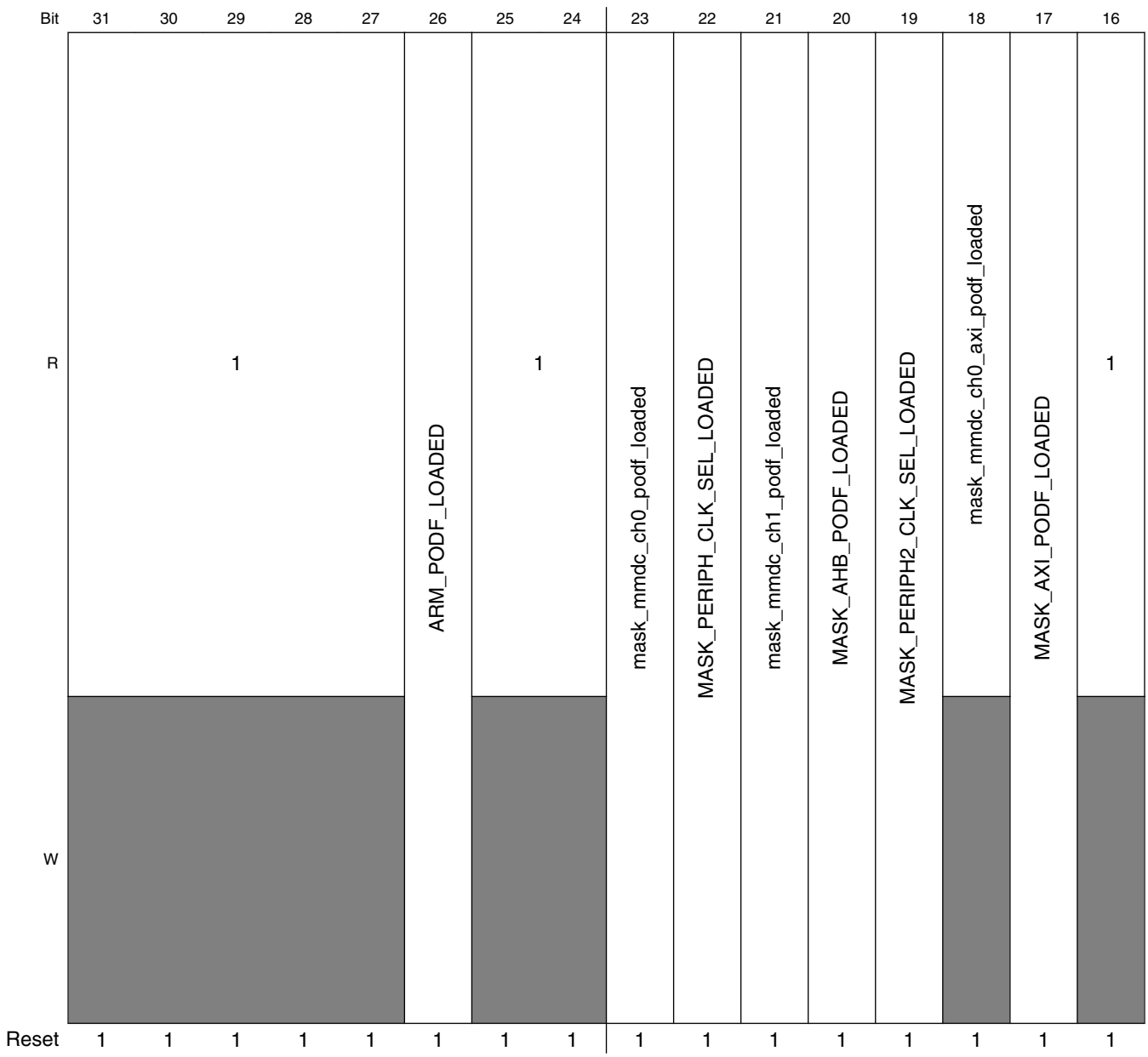
CCM_CISR field descriptions (continued)

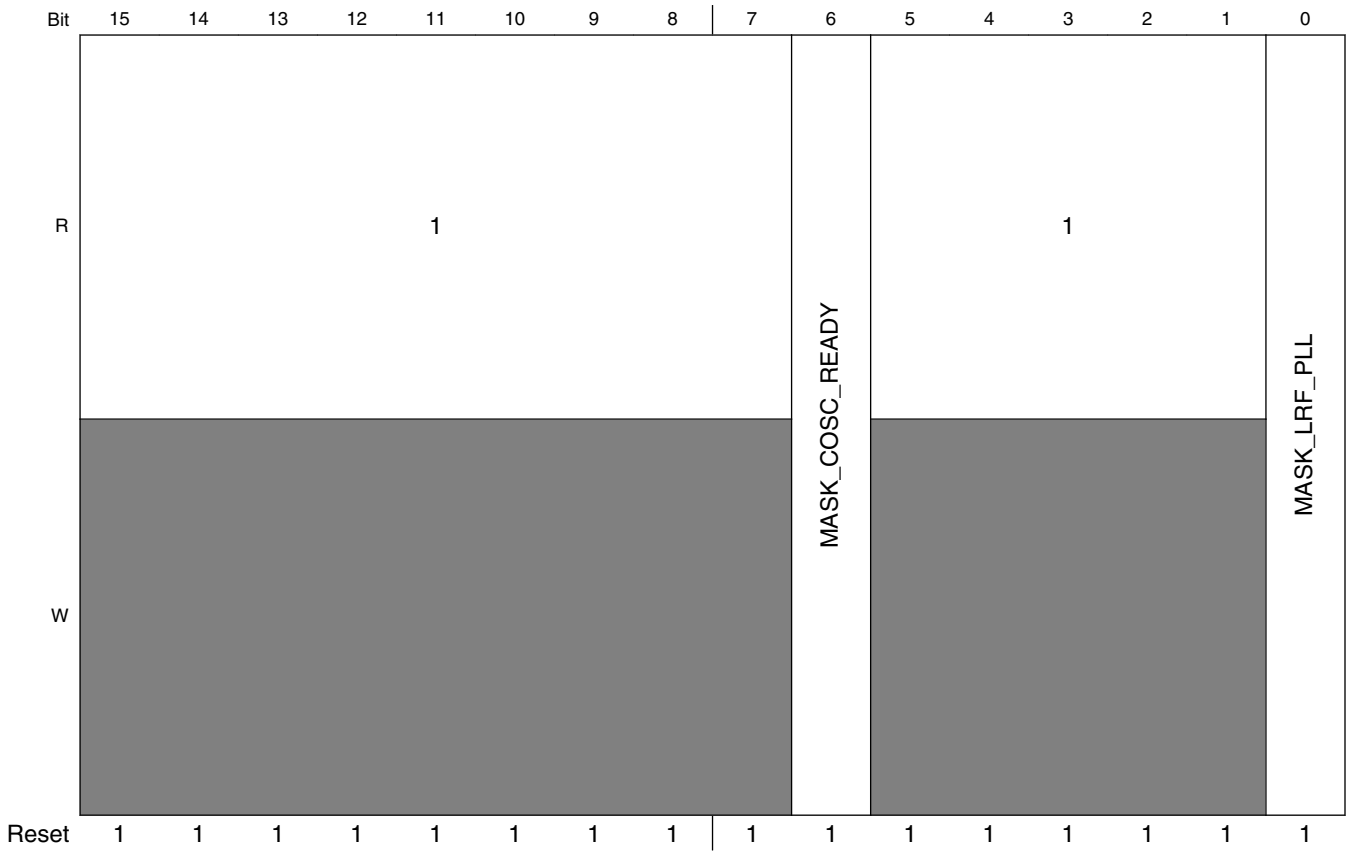
Field	Description
19 PERIPH2_CLK_SEL_LOADED	CCM interrupt request 1 generated due to frequency change of periph2_clk_sel 0 interrupt is not generated due to frequency change of periph2_clk_sel 1 interrupt generated due to frequency change of periph2_clk_sel
18 Reserved	This read-only field is reserved and always has the value 0. 0 interrupt is not generated due to frequency change of mmdc_ch0_axi_podf 1 interrupt generated due to frequency change of mmdc_ch0_axi_podf
17 AXI_PODF_LOADED	CCM interrupt request 1 generated due to frequency change of axi_podf 0 interrupt is not generated due to frequency change of axi_podf 1 interrupt generated due to frequency change of axi_podf
16–7 Reserved	This read-only field is reserved and always has the value 0.
6 COSC_READY	CCM interrupt request 2 generated due to on board oscillator ready, i.e. oscnt has finished counting. 0 interrupt is not generated due to on board oscillator ready 1 interrupt generated due to on board oscillator ready
5–1 Reserved	This read-only field is reserved and always has the value 0.
0 LRF_PLL	CCM interrupt request 2 generated due to lock of all enabled and not bypaseed PLLs 0 interrupt is not generated due to lock ready of all enabled and not bypaseed PLLs 1 interrupt generated due to lock ready of all enabled and not bypaseed PLLs

18.6.20 CCM Interrupt Mask Register (CCM_CIMR)

The figure below represents the CCM Interrupt Mask Register (CIMR). The table below provides its field descriptions.

Address: 20C_4000h base + 5Ch offset = 20C_405Ch





CCM_CIMR field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 1.
26 ARM_PODF_ LOADED	mask interrupt generation due to frequency change of arm_podf 0 don't mask interrupt due to frequency change of arm_podf - interrupt will be created 1 mask interrupt due to frequency change of arm_podf
25–24 Reserved	This read-only field is reserved and always has the value 1.
23 mask_mmdc_ ch0_podf_loaded	mask interrupt generation due to update of mask_mmdc_ch0_podf 0 don't mask interrupt due to update of mask_mmdc_ch0_podf - interrupt will be created 1 mask interrupt due to update of mask_mmdc_ch0_podf
22 MASK_PERIPH_ CLK_SEL_ LOADED	mask interrupt generation due to update of periph_clk_sel. 0 don't mask interrupt due to update of periph_clk_sel - interrupt will be created 1 mask interrupt due to update of periph_clk_sel
21 mask_mmdc_ ch1_podf_loaded	mask interrupt generation due to update of mask_mmdc_ch1_podf 0 don't mask interrupt due to update of mask_mmdc_ch1_podf - interrupt will be created 1 mask interrupt due to update of mask_mmdc_ch1_podf

Table continues on the next page...

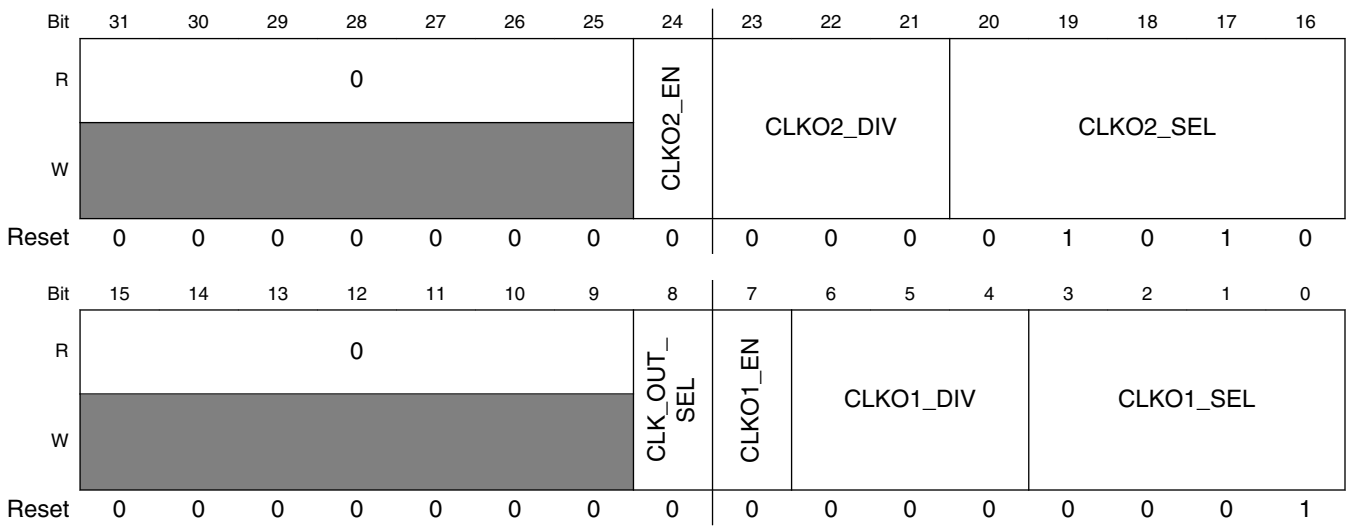
CCM_CIMR field descriptions (continued)

Field	Description
20 MASK_AHB_ PODF_LOADED	mask interrupt generation due to frequency change of ahb_podf 0 don't mask interrupt due to frequency change of ahb_podf - interrupt will be created 1 mask interrupt due to frequency change of ahb_podf
19 MASK_ PERIPH2_CLK_ SEL_LOADED	mask interrupt generation due to update of periph2_clk_sel. 0 don't mask interrupt due to update of periph2_clk_sel - interrupt will be created 1 mask interrupt due to update of periph2_clk_sel
18 mask_mmdc_ ch0_axi_podf_ loaded	mask interrupt generation due to frequency change of mmdc_ch0_axi_podf 0 don't mask interrupt due to frequency change of mmdc_ch0_axi_podf - interrupt will be created 1 mask interrupt due to frequency change of mmdc_ch0_axi_podf
17 MASK_AXI_ PODF_LOADED	mask interrupt generation due to frequency change of axi_podf 0 don't mask interrupt due to frequency change of axi_podf - interrupt will be created 1 mask interrupt due to frequency change of axi_podf
16–7 Reserved	This read-only field is reserved and always has the value 1.
6 MASK_COSC_ READY	mask interrupt generation due to on board oscillator ready 0 don't mask interrupt due to on board oscillator ready - interrupt will be created 1 mask interrupt due to on board oscillator ready
5–1 Reserved	This read-only field is reserved and always has the value 1.
0 MASK_LRF_PLL	mask interrupt generation due to lrf of PLLs 0 don't mask interrupt due to lrf of PLLs - interrupt will be created 1 mask interrupt due to lrf of PLLs

18.6.21 CCM Clock Output Source Register (CCM_CCOSR)

The figure below represents the CCM Clock Output Source Register (CCOSR). The CCOSR register contains bits to control the clocks that will be generated on the output ipp_do_clko1 (CCM_CLKO1) and ipp_do_clko2 (CCM_CLKO2).The table below provides its field descriptions.

Address: 20C_4000h base + 60h offset = 20C_4060h



CCM_CCOSR field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24 CLKO2_EN	Enable of CCM_CLKO2 clock 0 CCM_CLKO2 disabled. 1 CCM_CLKO2 enabled.
23–21 CLKO2_DIV	Setting the divider of CCM_CLKO2 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
20–16 CLKO2_SEL	Selection of the clock to be generated on CCM_CLKO2

Table continues on the next page...

CCM_CCOSR field descriptions (continued)

Field	Description
	00000 mmdc_ch0_clk_root 00001 mmdc_ch1_clk_root 00010 usdhc4_clk_root 00011 usdhc1_clk_root 00101 wrck_clk_root 00110 ecspi_clk_root 01000 usdhc3_clk_root 01001 pcie_clk_root 01010 arm_clk_root 01011 ipu1_hsp_clk_root 01100 epdc_axi_clk_root, pxp_axi_clk_root 01101 vdo_axi_clk_root 01110 osc_clk 01111 mlb_sys_clk_root 10000 gpu3d_core_clk_root 10001 usdhc2_clk_root 10010 ssi1_clk_root 10011 ssi2_clk_root 10100 ssi3_clk_root 10101 gpu2d_core_clk_root 10110 vpu_axi_clk_root 10111 can_clk_root 11000 ldb_di0_serial_clk_root 11001 ldb_di1_serial_clk_root 11010 esai_clk_root 11011 aclk_eim_slow_clk_root 11100 uart_clk_root 11101 spdif0_clk_root 11110 spdif1_clk_root 11111 hsi_tx_clk_root
15–9 Reserved	This read-only field is reserved and always has the value 0.
8 CLK_OUT_SEL	CCM_CLKO1 output to reflect CCM_CLKO1 or CCM_CLKO2 clocks 0 CCM_CLKO1 output drives CCM_CLKO1 clock 1 CCM_CLKO1 output drives CCM_CLKO2 clock
7 CLKO1_EN	Enable of CCM_CLKO1 clock 0 CCM_CLKO1 disabled. 1 CCM_CLKO1 enabled.
6–4 CLKO1_DIV	Setting the divider of CCM_CLKO1 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5

Table continues on the next page...

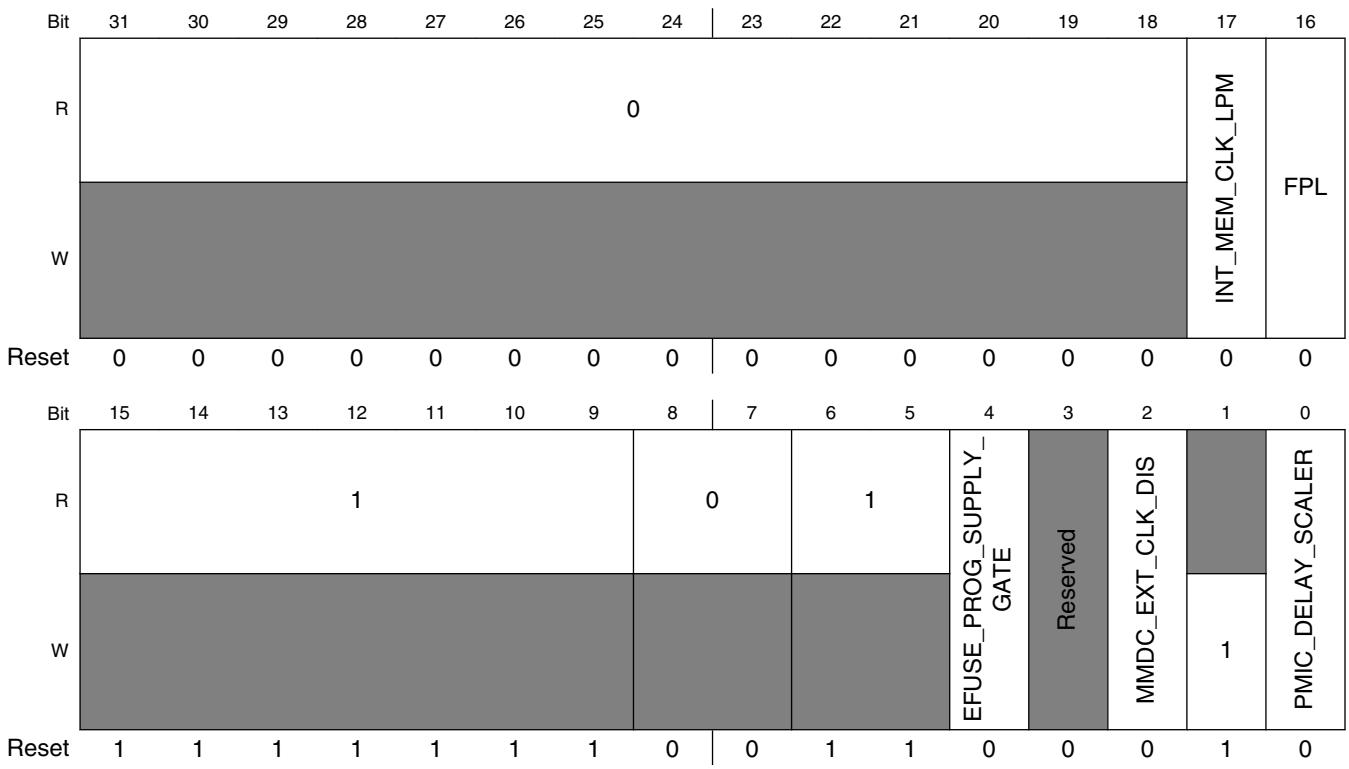
CCM_CCOSR field descriptions (continued)

Field	Description
	101 divide by 6 110 divide by 7 111 divide by 8
CLKO1_SEL	Selection of the clock to be generated on CCM_CLKO1 0000 pll3_sw_clk (this inputs has additional constant division /2) 0001 pll2_main_clk (this inputs has additional constant division /2) 0010 pll1_main_clk (this inputs has additional constant division /2) 0011 pll5_main_clk (this inputs has additional constant division /2) 0100 video_27M_clk_root 0101 dtcp_clk_root 0110 enfc_clk_root 0111 ipu1_di0_clk_root 1000 ipu1_di1_clk_root 1001 Reserved 1010 epdc_pix_clk_root 1011 ahb_clk_root 1100 ipg_clk_root 1101 perclk_root 1110 ckil_sync_clk_root 1111 pll4_main_clk

18.6.22 CCM General Purpose Register (CCM_CGPR)

Fast PLL enable. Can be used to engage PLL faster after STOP mode, if 24MHz OSC was active

Address: 20C_4000h base + 64h offset = 20C_4064h



CCM_CGPR field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value 0.
17 INT_MEM_CLK_LPM	Control for the Deep Sleep signal to the Arm Platform memories with additional control logic based on the Arm WFI signal. Used to keep the Arm Platform memory clocks enabled if an interrupt is pending when entering low power mode. **NOTE:** This bit should always bet set when the CCM_CLPCR_LPM bits are set to 01(WAIT Mode) or 10 (STOP mode) without power gating. This bit does not have to be set for STOP mode entry. 0 Disable the clock to the Arm platform memories when entering Low Power Mode 1 Keep the clocks to the Arm platform memories enabled only if an interrupt is pending when entering Low Power Modes (WAIT and STOP without power gating)
16 FPL	Fast PLL enable.

Table continues on the next page...

CCM_CGPR field descriptions (continued)

Field	Description
	0 Engage PLL enable default way. 1 Engage PLL enable 3 CKIL clocks earlier at exiting low power mode (STOP). Should be used only if 24MHz OSC was active in low power mode.
15–9 Reserved	This read-only field is reserved and always has the value 1.
8–7 Reserved	This read-only field is reserved and always has the value 0.
6–5 Reserved	This read-only field is reserved and always has the value 1.
4 EFUSE_PROG_SUPPLY_GATE	Defines the value of the output signal cgpr_dout[4]. Gate of program supply for efuse programing 0 fuse programing supply voltage is gated off to the efuse module 1 allow fuse programing.
3 -	This field is reserved. Reserved
2 MMDC_EXT_CLK_DIS	Disable external clock driver of MMDC during STOP mode 1 disable during stop mode 0 don't disable during stop mode.
1 -	Reserved. Keep default value set to '1' for proper operation.
0 PMIC_DELAY_SCALER	Defines clock dividion of clock for stby_count (pmic delay counter) 0 clock is not divided 1 clock is divided /8

18.6.23 CCM Clock Gating Register 0 (CCM_CCGR0)

CG(i) bits CCGR 0-6

These bits are used to turn on/off the clock to each module independently. The following table details the possible clock activity conditions for each module.

CGR value	Clock Activity Description
00	Clock is off during all modes. Stop enter hardware handshake is disabled.
01	Clock is on in run mode, but off in WAIT and STOP modes
10	Not applicable (Reserved).
11	Clock is on during all modes, except STOP mode.

Module should be stopped, before set its bits to "0"; clocks to the module will be stopped immediately.

The tables above show the register mappings for the different CGRs. The clock connectivity table should be used to match the "CCM output affected" to the actual clocks going into the modules.

The figure below represents the CCM Clock Gating Register 0 (CCM_CCGR0). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 7 CGR registers. The number of registers required is according to the number of peripherals in the system.

Address: 20C_4000h base + 68h offset = 20C_4068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR0 field descriptions

Field	Description
31–30 CG15	Reserved
29–28 CG14	dtcp clocks (dtcp_clk_enable)
27–26 CG13	dcic2 clocks (dcic2_clk_enable)
25–24 CG12	dcic1 clocks (dcic1_clk_enable)
23–22 CG11	CPU debug clocks (arm_dbg_clk_enable)
21–20 CG10	can2_serial clock (can2_serial_clk_enable)
19–18 CG9	can2 clock (can2_clk_enable)
17–16 CG8	can1_serial clock (can1_serial_clk_enable)
15–14 CG7	can1 clock (can1_clk_enable)
13–12 CG6	caam_wrapper_ipg clock (caam_wrapper_ipg_enable)
11–10 CG5	caam_wrapper_aclk clock (caam_wrapper_aclk_enable)
9–8 CG4	caam_secure_mem clock (caam_secure_mem_clk_enable)
7–6 CG3	asrc clock (asrc_clk_enable)

Table continues on the next page...

CCM_CCGR0 field descriptions (continued)

Field	Description
5–4 CG2	apbhdma hclk clock (apbhdma_hclk_enable)
3–2 CG1	aips_tz2 clocks (aips_tz2_clk_enable)
CG0	aips_tz1 clocks (aips_tz1_clk_enable)

18.6.24 CCM Clock Gating Register 1 (CCM_CCGR1)

The figure below represents the CCM Clock Gating Register 1(CCM_CCGR1). The clock gating registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is determined by the number of peripherals in the system.

Address: 20C_4000h base + 6Ch offset = 20C_406Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR1 field descriptions

Field	Description
31–30 CG15	Reserved
29–28 CG14	Reserved
27–26 CG13	gpu3d clock (gpu3d_clk_enable)
25–24 CG12	gpu2d clock (gpu2d_clk_enable) **NOTE:** GPU2D clock cannot be gated without gating OPENVG clock as well. Configure the CG bit (CCM_CCGR1[CG12]), to gate GPU2D.
23–22 CG11	gpt serial clock (gpt_serial_clk_enable)
21–20 CG10	gpt bus clock (gpt_clk_enable)
19–18 CG9	Reserved

Table continues on the next page...

CCM_CCGR1 field descriptions (continued)

Field	Description
17–16 CG8	esai clocks (esai_clk_enable)
15–14 CG7	epit2 clocks (epit2_clk_enable)
13–12 CG6	epit1 clocks (epit1_clk_enable)
11–10 CG5	enet clock (enet_clk_enable)
9–8 CG4	i2c4 clocks (i2c4_clk_enable)
7–6 CG3	ecspi4 clocks (ecspi4_clk_enable)
5–4 CG2	ecspi3 clocks (ecspi3_clk_enable)
3–2 CG1	ecspi2 clocks (ecspi2_clk_enable)
CG0	ecspi1 clocks (ecspi1_clk_enable)

18.6.25 CCM Clock Gating Register 2 (CCM_CCGR2)

The figure below represents the CCM Clock Gating Register 2 (CCM_CCGR2). The clock gating registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is determined by the number of peripherals in the system.

Address: 20C_4000h base + 70h offset = 20C_4070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8	
Reset	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR2 field descriptions

Field	Description
31–30 CG15	Reserved
29–28 CG14	Reserved

Table continues on the next page...

CCM_CCGR2 field descriptions (continued)

Field	Description
27–26 CG13	ipsync_vdoa_ipg clocks (ipsync_vdoa_ipg_master_clk_enable)
25–24 CG12	ipsync_ip2apb_tzasc2_ipg clocks (ipsync_ip2apb_tzasc2_ipg_master_clk_enable)
23–22 CG11	ipsync_ip2apb_tzasc1_ipg clocks (ipsync_ip2apb_tzasc1_ipg_master_clk_enable)
21–20 CG10	ipmux3 clock (ipmux3_clk_enable)
19–18 CG9	ipmux2 clock (ipmux2_clk_enable)
17–16 CG8	ipmux1 clock (ipmux1_clk_enable)
15–14 CG7	iomux_ipt_clk_io clock (iomux_ipt_clk_io_enable)
13–12 CG6	OCOTP_CTRL clock (ocotp_clk_enable)
11–10 CG5	i2c3_serial clock (i2c3_serial_clk_enable)
9–8 CG4	i2c2_serial clock (i2c2_serial_clk_enable)
7–6 CG3	i2c1_serial clock (i2c1_serial_clk_enable)
5–4 CG2	hdmi_tx_isfrclk clock (hdmi_tx_isfrclk_enable)
3–2 CG1	Reserved
CG0	hdmi_tx_iahbclk, hdmi_tx_ihclk clock (hdmi_tx_enable)

18.6.26 CCM Clock Gating Register 3 (CCM_CCGR3)

The figure below represents the CCM Clock Gating Register 3 (CCM_CCGR3). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is determined by the number of peripherals in the system.

Address: 20C_4000h base + 74h offset = 20C_4074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR3 field descriptions

Field	Description
31–30 CG15	Reserved
29–28 CG14	The OCRAM clock cannot be turned off when the CM cache is running on this device. ocram clock(ocram_clk_enable)
27–26 CG13	Reserved
25–24 CG12	mmdc_core_ipg_clk_p0 clock (mmdc_core_ipg_clk_p0_enable)
23–22 CG11	Reserved
21–20 CG10	mmdc_core_aclk_fast_core_p0 clock (mmdc_core_aclk_fast_core_p0_enable)
19–18 CG9	mlb clock (mlb_clk_enable)
17–16 CG8	mipi_core_cfg clock (mipi_core_cfg_clk_enable)
15–14 CG7	ldb_di1 clock (ldb_di1_clk_enable)
13–12 CG6	ldb_di0 clock (ldb_di0_clk_enable)
11–10 CG5	epdc_pix clock (epdc_pix_clk_enable)
9–8 CG4	Reserved

Table continues on the next page...

CCM_CCGR3 field descriptions (continued)

Field	Description
7–6 CG3	epdc/pxp clock (epdc_axi_clk_enable)
5–4 CG2	ipu1_di1 clock and pre-clock (ipu1_ipu_di1_clk_enable)
3–2 CG1	ipu1_di0 clock and pre-clock (ipu1_ipu_di0_clk_enable)
CG0	ipu1_ipu clock (ipu1_ipu_clk_enable)

18.6.27 CCM Clock Gating Register 4 (CCM_CCGR4)

The figure below represents the CCM Clock Gating Register 4 (CCM_CCGR4). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is determined by the number of peripherals in the system.

Address: 20C_4000h base + 78h offset = 20C_4078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR4 field descriptions

Field	Description
31–30 CG15	rawnand_u_gpmi_input_apb clock (rawnand_u_gpmi_input_apb_clk_enable)
29–28 CG14	rawnand_u_gpmi_bch_input_gpmi_io clock (rawnand_u_gpmi_bch_input_gpmi_io_clk_enable)
27–26 CG13	rawnand_u_gpmi_bch_input_bch clock (rawnand_u_gpmi_bch_input_bch_clk_enable)
25–24 CG12	rawnand_u_bch_input_apb clock (rawnand_u_bch_input_apb_clk_enable)
23–22 CG11	pwm4 clocks (pwm4_clk_enable)
21–20 CG10	pwm3 clocks (pwm3_clk_enable)
19–18 CG9	pwm2 clocks (pwm2_clk_enable)

Table continues on the next page...

CCM_CCGR4 field descriptions (continued)

Field	Description
17–16 CG8	pwm1 clocks (pwm1_clk_enable)
15–14 CG7	pl301_mx6qper2_mainclk_enable (pl301_mx6qper2_mainclk_enable)
13–12 CG6	pl301_mx6qper1_bch clocks (pl301_mx6qper1_bchclk_enable) **NOTE:** This gates bch_clk_root to sim_m fabric.
11–10 CG5	Reserved
9–8 CG4	pl301_mx6qfast1_s133 clock (pl301_mx6qfast1_s133clk_enable)
7–6 CG3	Reserved
5–4 CG2	Reserved.
3–2 CG1	Reserved.
CG0	pcie clock (pcie_root_enable)

18.6.28 CCM Clock Gating Register 5 (CCM_CCGR5)

The figure below represents the CCM Clock Gating Register 5 (CCM_CCGR5). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is determined by the number of peripherals in the system.

Address: 20C_4000h base + 7Ch offset = 20C_407Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR5 field descriptions

Field	Description
31–30 CG15	Reserved

Table continues on the next page...

CCM_CCGR5 field descriptions (continued)

Field	Description
29–28 CG14	Reserved
27–26 CG13	uart_serial clock (uart_serial_clk_enable)
25–24 CG12	uart clock (uart_clk_enable)
23–22 CG11	ssi3 clocks (ssi3_clk_enable)
21–20 CG10	ssi2 clocks (ssi2_clk_enable)
19–18 CG9	ssi1 clocks (ssi1_clk_enable)
17–16 CG8	Reserved
15–14 CG7	spdif clock (spdif_clk_enable)
13–12 CG6	spba clock (spba_clk_enable)
11–10 CG5	Reserved
9–8 CG4	Reserved
7–6 CG3	sdma clock (sdma_clk_enable)
5–4 CG2	Reserved
3–2 CG1	Reserved
CG0	rom clock (rom_clk_enable)

18.6.29 CCM Clock Gating Register 6 (CCM_CCGR6)

The figure below represents the CCM Clock Gating Register 6 (CCM_CCGR6). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is determined by the number of peripherals in the system.

Address: 20C_4000h base + 80h offset = 20C_4080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR6 field descriptions

Field	Description
31–30 CG15	Reserved
29–28 CG14	Reserved
27–26 CG13	Reserved
25–24 CG12	Reserved
23–22 CG11	Reserved
21–20 CG10	Reserved
19–18 CG9	Reserved
17–16 CG8	Reserved
15–14 CG7	vpu clocks (vpu_clk_enable)
13–12 CG6	vdoaxiclk root clock (vdoaxiclk_clk_enable)
11–10 CG5	eim_slow clocks (eim_slow_clk_enable)
9–8 CG4	usdhc4 clocks (usdhc4_clk_enable)

Table continues on the next page...

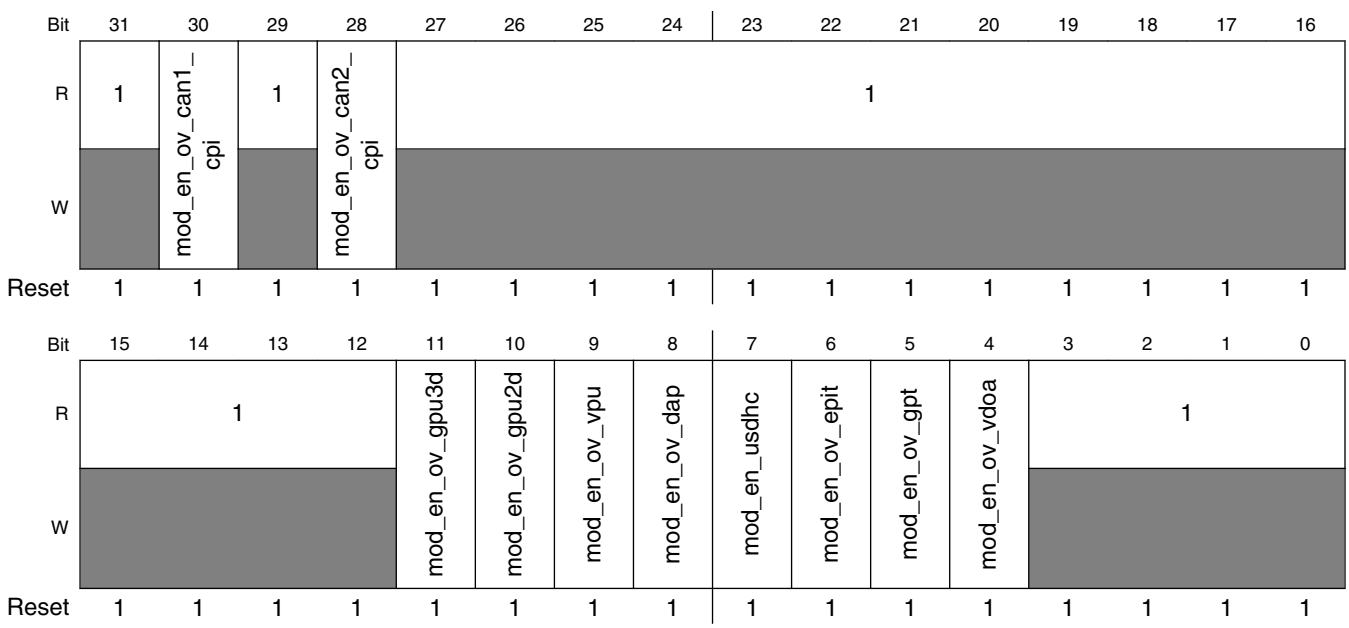
CCM_CCGR6 field descriptions (continued)

Field	Description
7–6 CG3	usdhc3 clocks (usdhc3_clk_enable)
5–4 CG2	usdhc2 clocks (usdhc2_clk_enable)
3–2 CG1	usdhc1 clocks (usdhc1_clk_enable)
CG0	usboh3 clock (usboh3_clk_enable)

18.6.30 CCM Module Enable Overide Register (CCM_CMEOR)

The following figure represents the CCM Module Enable Override Register (CMEOR). The CMEOR register contains bits to override the clock enable signal from the module. This bit is applicable only for modules whose clock enable signals are used. The following table provides its field descriptions.

Address: 20C_4000h base + 88h offset = 20C_4088h



CCM_CMEOR field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 1.

Table continues on the next page...

CCM_CMEOR field descriptions (continued)

Field	Description
30 mod_en_ov_ can1_cpi	Overide clock enable signal from CAN1 - clock will not be gated based on CAN's signal 'enable_clk_cpi'. 0 don't overide module enable signal 1 overide module enable signal
29 Reserved	This read-only field is reserved and always has the value 1.
28 mod_en_ov_ can2_cpi	Overide clock enable signal from CAN2 - clock will not be gated based on CAN's signal 'enable_clk_cpi'. 0 don't override module enable signal 1 override module enable signal
27–12 Reserved	This read-only field is reserved and always has the value 1.
11 mod_en_ov_ gpu3d	Overide clock enable signal from GPU3D - clock will not be gated based on GPU3D's signal. 0 don't override module enable signal 1 override module enable signal
10 mod_en_ov_ gpu2d	Overide clock enable signal from GPU2D - clock will not be gated based on GPU's signal 'gpu2d_busy' . 0 don't override module enable signal 1 override module enable signal
9 mod_en_ov_vpu	Overide clock enable signal from VPU- clock will not be gated based on VPU's signal 'vpu_idle' . 0 don't override module enable signal 1 override module enable signal
8 mod_en_ov_dap	Overide clock enable signal from DAP- clock will not be gated based on DAP's signal 'dap_dbgen' . 0 don't override module enable signal 1 override module enable signal
7 mod_en_usdhc	Overide clock enable signal from USDHC. 0 don't override module enable signal 1 override module enable signal
6 mod_en_ov_epit	Overide clock enable signal from EPIT - clock will not be gated based on EPIT's signal 'ipg_enable_clk' . 0 don't override module enable signal 1 override module enable signal
5 mod_en_ov_gpt	Overide clock enable signal from GPT - clock will not be gated based on GPT's signal 'ipg_enable_clk' . 0 don't override module enable signal 1 override module enable signal
4 mod_en_ov_ vdoa	Overide clock enable signal from vdoa - clock will not be gated based on vdoa signal. 0 don't override module enable signal 1 override module enable signal
Reserved	This read-only field is reserved and always has the value 1.

18.7 CCM Analog Memory Map/Register Definition

This section describes the registers for the analog PLLs. The registers which have the same description are grouped within {}. The register offsets for the various PLLs are:

- ARM PLL: {0h000, 0h004, 0h008, 0h00C}.
- USB1 PLL: {0h010, 0h014, 0h018, 0h01C}, {0h0F0, 0h0F4, 0h0F8, 0h0FC}.
- System PLL: {0h030, 0h034, 0h038, 0h03C}, 0h040, 0h050, 0h060, {0h100, 0h104, 0h108, 0h10C}.
- Audio / Video PLL: {0h070, 0h074, 0h078, 0h07C}, 0h080, 0h090, {0h0A0, 0h0A4, 0h0A8, 0h0AC}, 0h0B0, 0h0C0

CCM_ANALOG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_8000	Analog ARM PLL control Register (CCM_ANALOG_PLL_ARM)	32	R/W	0001_3042h	18.7.1/884
20C_8004	Analog ARM PLL control Register (CCM_ANALOG_PLL_ARM_SET)	32	R/W	0001_3042h	18.7.1/884
20C_8008	Analog ARM PLL control Register (CCM_ANALOG_PLL_ARM_CLR)	32	R/W	0001_3042h	18.7.1/884
20C_800C	Analog ARM PLL control Register (CCM_ANALOG_PLL_ARM_TOG)	32	R/W	0001_3042h	18.7.1/884
20C_8010	Analog USB1 480MHz PLL Control Register (CCM_ANALOG_PLL_USB1)	32	R/W	0001_2000h	18.7.2/886
20C_8014	Analog USB1 480MHz PLL Control Register (CCM_ANALOG_PLL_USB1_SET)	32	R/W	0001_2000h	18.7.2/886
20C_8018	Analog USB1 480MHz PLL Control Register (CCM_ANALOG_PLL_USB1_CLR)	32	R/W	0001_2000h	18.7.2/886
20C_801C	Analog USB1 480MHz PLL Control Register (CCM_ANALOG_PLL_USB1_TOG)	32	R/W	0001_2000h	18.7.2/886
20C_8020	Analog USB2 480MHz PLL Control Register (CCM_ANALOG_PLL_USB2)	32	R/W	0001_2000h	18.7.3/889
20C_8024	Analog USB2 480MHz PLL Control Register (CCM_ANALOG_PLL_USB2_SET)	32	R/W	0001_2000h	18.7.3/889
20C_8028	Analog USB2 480MHz PLL Control Register (CCM_ANALOG_PLL_USB2_CLR)	32	R/W	0001_2000h	18.7.3/889
20C_802C	Analog USB2 480MHz PLL Control Register (CCM_ANALOG_PLL_USB2_TOG)	32	R/W	0001_2000h	18.7.3/889
20C_8030	Analog System PLL Control Register (CCM_ANALOG_PLL_SYS)	32	R/W	0001_3001h	18.7.4/891
20C_8034	Analog System PLL Control Register (CCM_ANALOG_PLL_SYS_SET)	32	R/W	0001_3001h	18.7.4/891
20C_8038	Analog System PLL Control Register (CCM_ANALOG_PLL_SYS_CLR)	32	R/W	0001_3001h	18.7.4/891

Table continues on the next page...

CCM_ANALOG memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_803C	Analog System PLL Control Register (CCM_ANALOG_PLL_SYS_TOG)	32	R/W	0001_3001h	18.7.4/891
20C_8040	528MHz System PLL Spread Spectrum Register (CCM_ANALOG_PLL_SYS_SS)	32	R/W	0000_0000h	18.7.5/893
20C_8050	Numerator of 528MHz System PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_SYS_NUM)	32	R/W	0000_0000h	18.7.6/893
20C_8060	Denominator of 528MHz System PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_SYS_DENOM)	32	R/W	0000_0012h	18.7.7/894
20C_8070	Analog Audio PLL control Register (CCM_ANALOG_PLL_AUDIO)	32	R/W	0001_1006h	18.7.8/895
20C_8074	Analog Audio PLL control Register (CCM_ANALOG_PLL_AUDIO_SET)	32	R/W	0001_1006h	18.7.8/895
20C_8078	Analog Audio PLL control Register (CCM_ANALOG_PLL_AUDIO_CLR)	32	R/W	0001_1006h	18.7.8/895
20C_807C	Analog Audio PLL control Register (CCM_ANALOG_PLL_AUDIO_TOG)	32	R/W	0001_1006h	18.7.8/895
20C_8080	Numerator of Audio PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_AUDIO_NUM)	32	R/W	05F5_E100h	18.7.9/897
20C_8090	Denominator of Audio PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_AUDIO_DENOM)	32	R/W	2964_619Ch	18.7.10/ 897
20C_80A0	Analog Video PLL control Register (CCM_ANALOG_PLL_VIDEO)	32	R/W	0001_100Ch	18.7.11/ 899
20C_80A4	Analog Video PLL control Register (CCM_ANALOG_PLL_VIDEO_SET)	32	R/W	0001_100Ch	18.7.11/ 899
20C_80A8	Analog Video PLL control Register (CCM_ANALOG_PLL_VIDEO_CLR)	32	R/W	0001_100Ch	18.7.11/ 899
20C_80AC	Analog Video PLL control Register (CCM_ANALOG_PLL_VIDEO_TOG)	32	R/W	0001_100Ch	18.7.11/ 899
20C_80B0	Numerator of Video PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_VIDEO_NUM)	32	R/W	05F5_E100h	18.7.12/ 901
20C_80C0	Denominator of Video PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_VIDEO_DENOM)	32	R/W	10A2_4447h	18.7.13/ 901
20C_80D0	MLB PLL Control Register (CCM_ANALOG_PLL_MLB)	32	R/W	0001_0000h	18.7.14/ 902
20C_80D4	MLB PLL Control Register (CCM_ANALOG_PLL_MLB_SET)	32	R/W	0001_0000h	18.7.14/ 902
20C_80D8	MLB PLL Control Register (CCM_ANALOG_PLL_MLB_CLR)	32	R/W	0001_0000h	18.7.14/ 902
20C_80DC	MLB PLL Control Register (CCM_ANALOG_PLL_MLB_TOG)	32	R/W	0001_0000h	18.7.14/ 902
20C_80E0	Analog ENET PLL Control Register (CCM_ANALOG_PLL_ENET)	32	R/W	0001_1001h	18.7.15/ 904
20C_80E4	Analog ENET PLL Control Register (CCM_ANALOG_PLL_ENET_SET)	32	R/W	0001_1001h	18.7.15/ 904

Table continues on the next page...

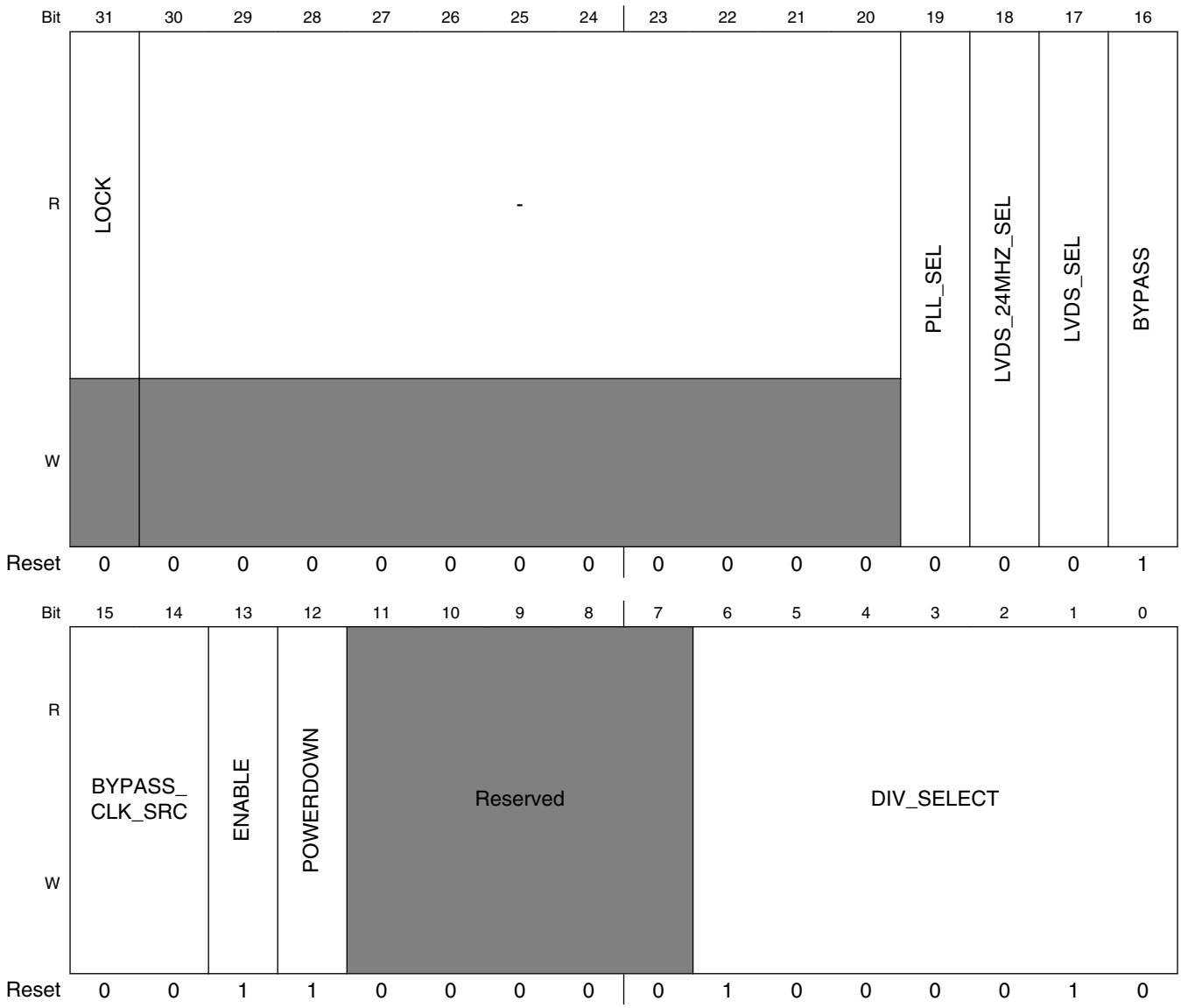
CCM_ANALOG memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_80E8	Analog ENET PLL Control Register (CCM_ANALOG_PLL_ENET_CLR)	32	R/W	0001_1001h	18.7.15/ 904
20C_80EC	Analog ENET PLL Control Register (CCM_ANALOG_PLL_ENET_TOG)	32	R/W	0001_1001h	18.7.15/ 904
20C_80F0	480MHz Clock (PLL3) Phase Fractional Divider Control Register (CCM_ANALOG_PFD_480)	32	R/W	1311_100Ch	18.7.16/ 905
20C_80F4	480MHz Clock (PLL3) Phase Fractional Divider Control Register (CCM_ANALOG_PFD_480_SET)	32	R/W	1311_100Ch	18.7.16/ 905
20C_80F8	480MHz Clock (PLL3) Phase Fractional Divider Control Register (CCM_ANALOG_PFD_480_CLR)	32	R/W	1311_100Ch	18.7.16/ 905
20C_80FC	480MHz Clock (PLL3) Phase Fractional Divider Control Register (CCM_ANALOG_PFD_480_TOG)	32	R/W	1311_100Ch	18.7.16/ 905
20C_8100	528MHz Clock (PLL2) Phase Fractional Divider Control Register (CCM_ANALOG_PFD_528)	32	R/W	1018_101Bh	18.7.17/ 907
20C_8104	528MHz Clock (PLL2) Phase Fractional Divider Control Register (CCM_ANALOG_PFD_528_SET)	32	R/W	1018_101Bh	18.7.17/ 907
20C_8108	528MHz Clock (PLL2) Phase Fractional Divider Control Register (CCM_ANALOG_PFD_528_CLR)	32	R/W	1018_101Bh	18.7.17/ 907
20C_810C	528MHz Clock (PLL2) Phase Fractional Divider Control Register (CCM_ANALOG_PFD_528_TOG)	32	R/W	1018_101Bh	18.7.17/ 907
20C_8150	Miscellaneous Register 0 (CCM_ANALOG_MISC0)	32	R/W	0400_0000h	18.7.18/ 910
20C_8154	Miscellaneous Register 0 (CCM_ANALOG_MISC0_SET)	32	R/W	0400_0000h	18.7.18/ 910
20C_8158	Miscellaneous Register 0 (CCM_ANALOG_MISC0_CLR)	32	R/W	0400_0000h	18.7.18/ 910
20C_815C	Miscellaneous Register 0 (CCM_ANALOG_MISC0_TOG)	32	R/W	0400_0000h	18.7.18/ 910
20C_8160	Miscellaneous Register 1 (CCM_ANALOG_MISC1)	32	R/W	0000_0000h	18.7.19/ 914
20C_8164	Miscellaneous Register 1 (CCM_ANALOG_MISC1_SET)	32	R/W	0000_0000h	18.7.19/ 914
20C_8168	Miscellaneous Register 1 (CCM_ANALOG_MISC1_CLR)	32	R/W	0000_0000h	18.7.19/ 914
20C_816C	Miscellaneous Register 1 (CCM_ANALOG_MISC1_TOG)	32	R/W	0000_0000h	18.7.19/ 914
20C_8170	Miscellaneous Register 2 (CCM_ANALOG_MISC2)	32	R/W	0027_2727h	18.7.20/ 917
20C_8174	Miscellaneous Register 2 (CCM_ANALOG_MISC2_SET)	32	R/W	0027_2727h	18.7.20/ 917
20C_8178	Miscellaneous Register 2 (CCM_ANALOG_MISC2_CLR)	32	R/W	0027_2727h	18.7.20/ 917
20C_817C	Miscellaneous Register 2 (CCM_ANALOG_MISC2_TOG)	32	R/W	0027_2727h	18.7.20/ 917

18.7.1 Analog ARM PLL control Register (CCM_ANALOG_PLL_ARM*n*)

The control register provides control for the system PLL.

Address: 20C_8000h base + 0h offset + (4d × i), where i=0d to 3d



CCM_ANALOG_PLL_ARM*n* field descriptions

Field	Description
31 LOCK	1 - PLL is currently locked. 0 - PLL is not currently locked.

Table continues on the next page...

CCM_ANALOG_PLL_ARM*n* field descriptions (continued)

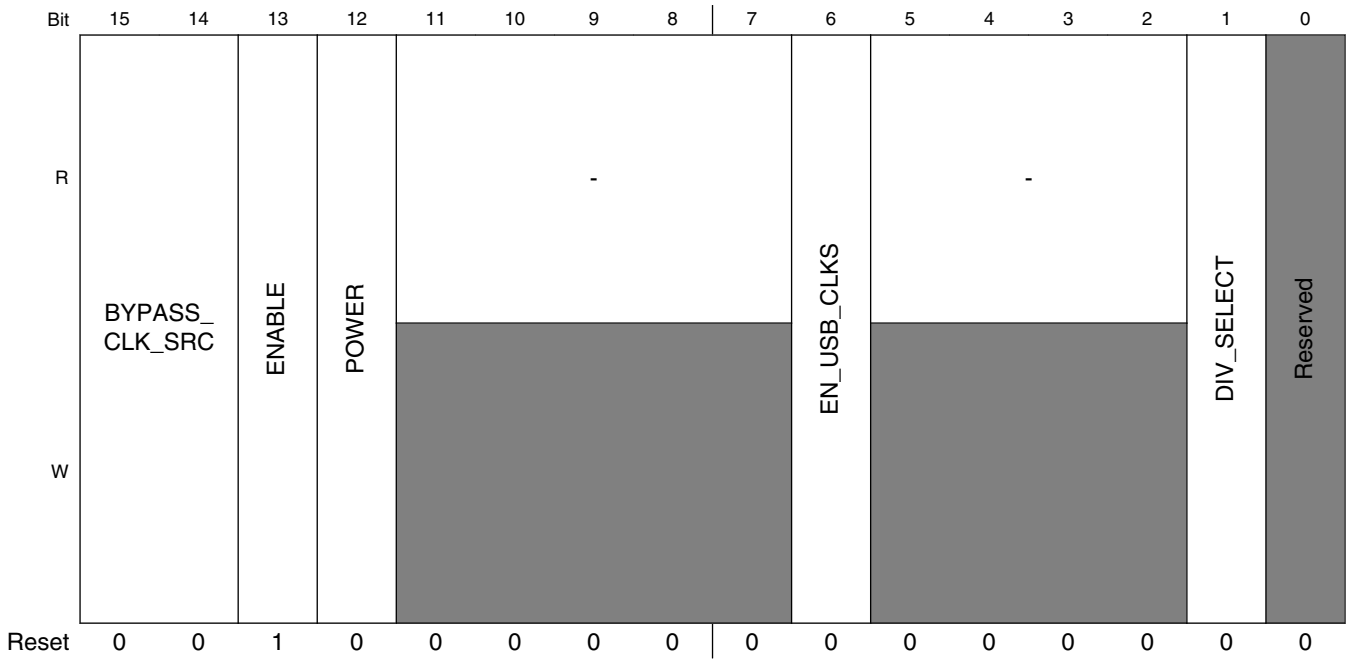
Field	Description
30–20 -	Always set to zero (0).
19 PLL_SEL	Reserved
18 LVDS_24MHZ_SEL	Analog Debug Bit
17 LVDS_SEL	Analog Debug Bit
16 BYPASS	Bypass the PLL.
15–14 BYPASS_CLK_SRC	Determines the bypass source. **NOTE:** Changing the Bypass clock source also changes the PLL reference clock source. 0x0 **REF_CLK_24M** — Select the 24MHz oscillator as source. 0x1 **CLK1** — Select the CLK1_N / CLK1_P as source. 0x2 **CLK2** — Select the CLK2_N / CLK2_P as source. 0x3 **XOR** — Select the XOR of CLK1_N / CLK1_P and CLK2_N / CLK2_P as source.
13 ENABLE	Enable the clock output.
12 POWERDOWN	Powers down the PLL.
11–7 -	This field is reserved. Reserved.
DIV_SELECT	This field controls the PLL loop divider. Valid range for divider value: 54-108. Fout = Fin * div_select/2.0.

18.7.2 Analog USB1 480MHz PLL Control Register (CCM_ANALOG_PLL_USB1*n*)

The control register provides control for USBPHY0 480MHz PLL.

Address: 20C_8000h base + 10h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	-														BYPASS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1



CCM_ANALOG_PLL_USB1*n* field descriptions

Field	Description
31 LOCK	1 - PLL is currently locked. 0 - PLL is not currently locked.
30–17 -	Always set to zero (0).
16 BYPASS	Bypass the PLL.
15–14 BYPASS_CLK_SRC	Determines the bypass source. 0x0 **REF_CLK_24M** — Select the 24MHz oscillator as source. 0x1 **CLK1** — Select the CLK1_N / CLK1_P as source. 0x2 **CLK2** — Select the CLK2_N / CLK2_P as source. 0x3 **XOR** — Select the XOR of CLK1_N / CLK1_P and CLK2_N / CLK2_P as source.
13 ENABLE	Enable the PLL clock output.
12 POWER	Powers up the PLL. This bit will be set automatically when USBPHY0 remote wakeup event happens.
11–7 -	Always set to zero (0).
6 EN_USB_CLKS	Powers the 9-phase PLL outputs for USBPHYn. Additionally, the UTMI clock gate must be deasserted in the USBPHYn to enable USBn operation (clear CLKGATE bit in USBPHYn_CTRL). This bit will be set automatically when USBPHYn remote wakeup event occurs. 0 PLL outputs for USBPHYn off. 1 PLL outputs for USBPHYn on.
5–2 -	Always set to zero (0).

Table continues on the next page...

CCM_ANALOG_PLL_USB1*n* field descriptions (continued)

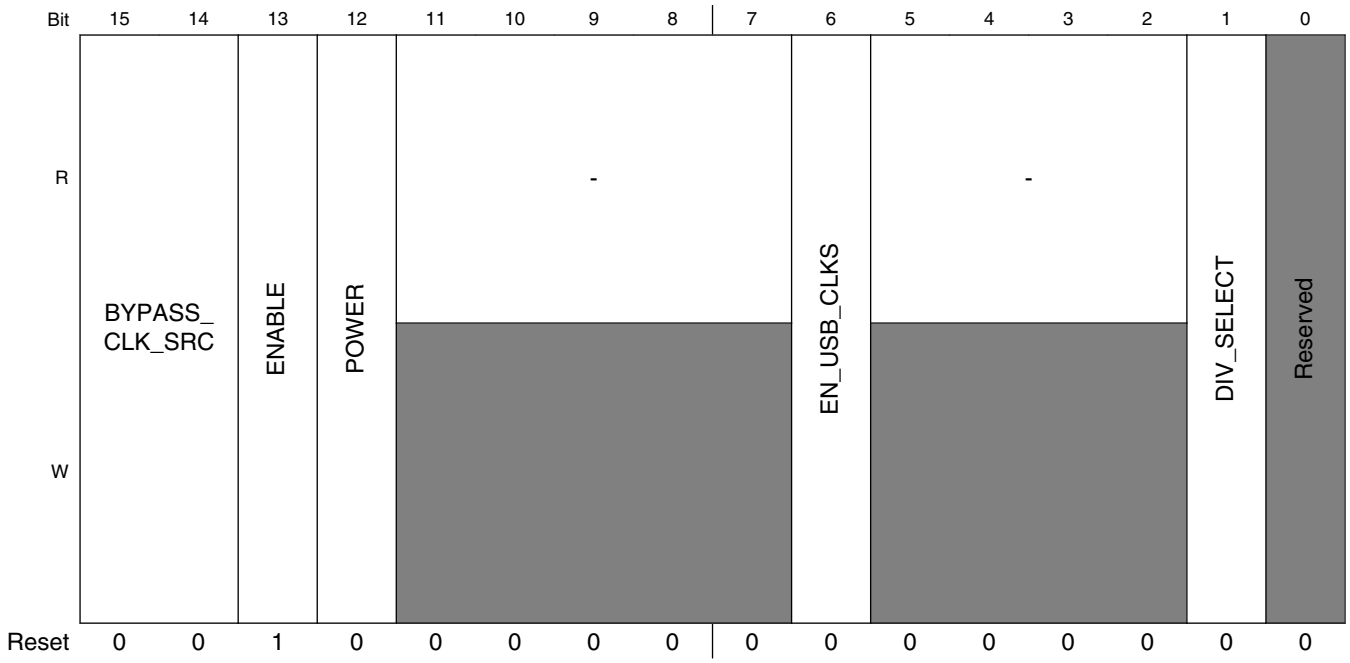
Field	Description
1 DIV_SELECT	This field controls the PLL loop divider. 0 - Fout=Fref*20; 1 - Fout=Fref*22.
0 -	This field is reserved. Reserved.

18.7.3 Analog USB2 480MHz PLL Control Register (CCM_ANALOG_PLL_USB2*n*)

The control register provides control for USBPHY1 480MHz PLL.

Address: 20C_8000h base + 20h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	-														BYPASS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1



CCM_ANALOG_PLL_USB2*n* field descriptions

Field	Description
31 LOCK	1 - PLL is currently locked. 0 - PLL is not currently locked.
30–17 -	Always set to zero (0).
16 BYPASS	Bypass the PLL.
15–14 BYPASS_CLK_SRC	Determines the bypass source. 0x0 **REF_CLK_24M** — Select the 24MHz oscillator as source. 0x1 **CLK1** — Select the CLK1_N / CLK1_P as source. 0x2 **CLK2** — Select the CLK2_N / CLK2_P as source. 0x3 **XOR** — Select the XOR of CLK1_N / CLK1_P and CLK2_N / CLK2_P as source.
13 ENABLE	Enable the PLL clock output.
12 POWER	Powers up the PLL. This bit will be set automatically when USBPHY1 remote wakeup event happens.
11–7 -	Always set to zero (0).
6 EN_USB_CLKS	0: 8-phase PLL outputs for USBPHY1 are powered down. If set to 1, 8-phase PLL outputs for USBPHY1 are powered up. Additionally, the utmi clock gate must be deasserted in the USBPHY1 to enable USB0 operation (clear CLKGATE bit in USBPHY1_CTRL).This bit will be set automatically when USBPHY1 remote wakeup event happens.
5–2 -	Always set to zero (0).
1 DIV_SELECT	This field controls the PLL loop divider. 0 - Fout=Fref*20; 1 - Fout=Fref*22.

Table continues on the next page...

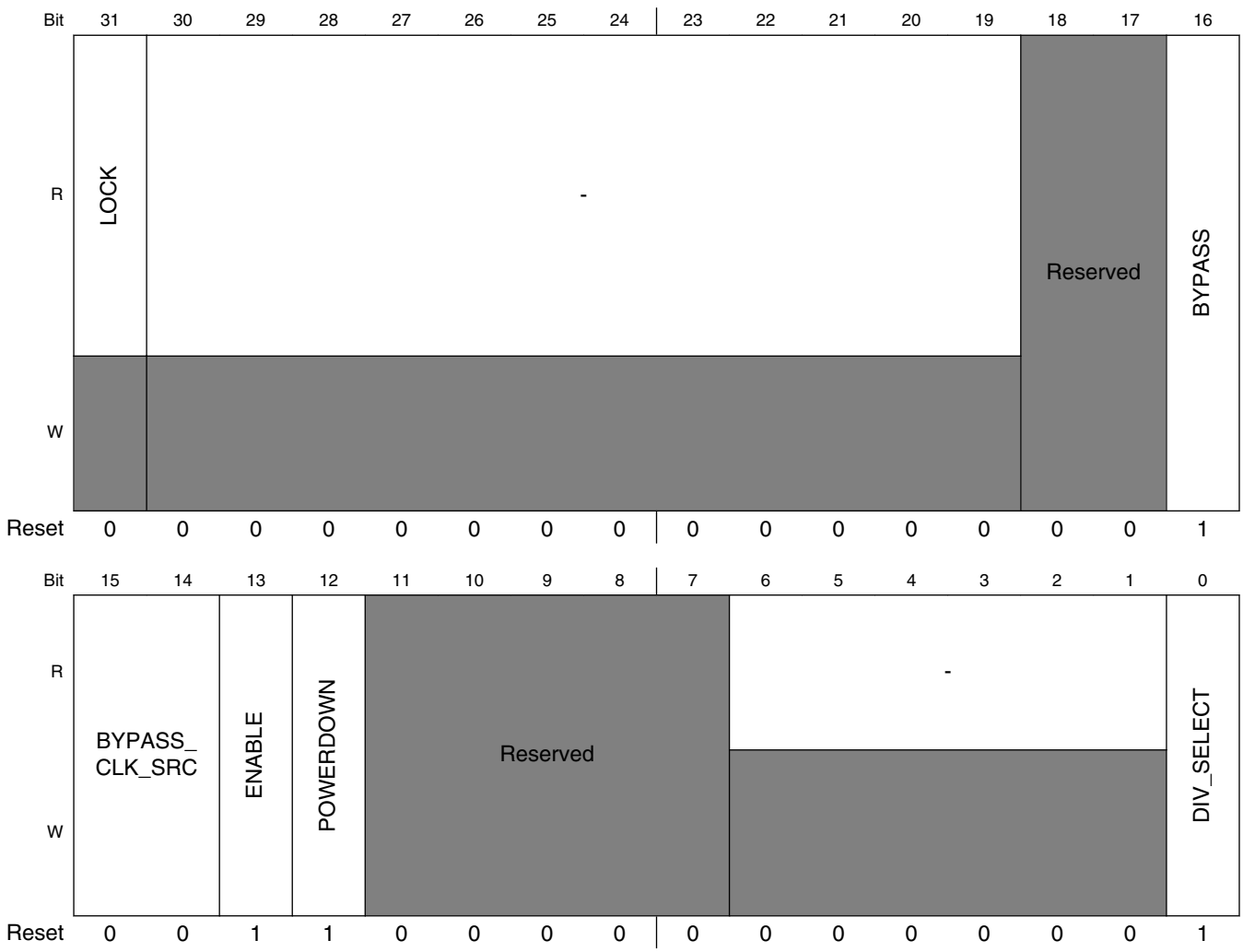
CCM_ANALOG_PLL_USB2*n* field descriptions (continued)

Field	Description
0 -	This field is reserved. Reserved.

18.7.4 Analog System PLL Control Register (CCM_ANALOG_PLL_SYS*n*)

The control register provides control for the 528MHz PLL.

Address: 20C_8000h base + 30h offset + (4d × i), where i=0d to 3d



CCM_ANALOG_PLL_SYS*n* field descriptions

Field	Description
31 LOCK	1 - PLL is currently locked; 0 - PLL is not currently locked.
30–19 -	Always set to zero (0).
18–17 -	This field is reserved. Reserved
16 BYPASS	Bypass the PLL.
15–14 BYPASS_CLK_SRC	Determines the bypass source. 0x0 **REF_CLK_24M** — Select the 24MHz oscillator as source. 0x1 **CLK1** — Select the CLK1_N / CLK1_P as source. 0x2 **CLK2** — Select the CLK2_N / CLK2_P as source. 0x3 **XOR** — Select the XOR of CLK1_N / CLK1_P and CLK2_N / CLK2_P as source.
13 ENABLE	Enable PLL output
12 POWERDOWN	Powers down the PLL.
11–7 -	This field is reserved. Reserved.
6–1 -	Always set to zero (0).
0 DIV_SELECT	This field controls the PLL loop divider. 0 - Fout=Fref*20; 1 - Fout=Fref*22.

18.7.5 528MHz System PLL Spread Spectrum Register (CCM_ANALOG_PLL_SYS_SS)

This register contains the 528MHz PLL spread spectrum controls.

Address: 20C_8000h base + 40h offset = 20C_8040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STOP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ENABLE	STEP														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_ANALOG_PLL_SYS_SS field descriptions

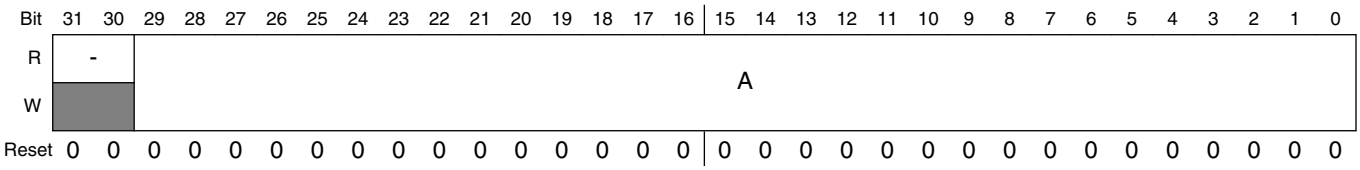
Field	Description
31–16 STOP	Frequency change = stop/CCM_ANALOG_PLL_SYS_DENOM[B]*24MHz.
15 ENABLE	Enable bit 0 — Spread spectrum modulation disabled 1 — Soread spectrum modulation enabled
STEP	Frequency change step = step/CCM_ANALOG_PLL_SYS_DENOM[B]*24MHz.

18.7.6 Numerator of 528MHz System PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_SYS_NUM)

This register contains the numerator of 528MHz PLL fractional loop divider (signed number).

Absolute value should be less than denominator.

Address: 20C_8000h base + 50h offset = 20C_8050h



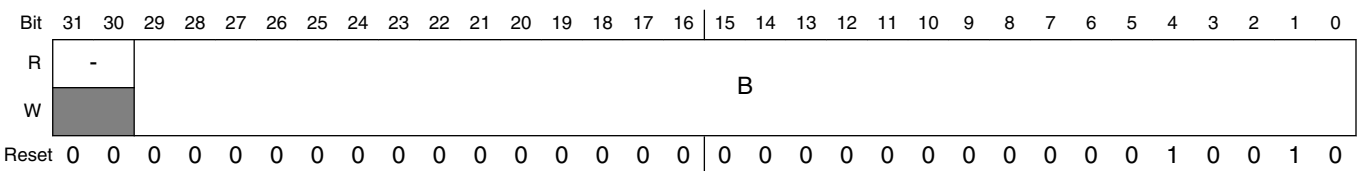
CCM_ANALOG_PLL_SYS_NUM field descriptions

Field	Description
31–30 -	Always set to zero (0).
A	30 bit numerator (A) of fractional loop divider (signed integer).

18.7.7 Denominator of 528MHz System PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_SYS_DENOM)

This register contains the denominator of 528MHz PLL fractional loop divider.

Address: 20C_8000h base + 60h offset = 20C_8060h



CCM_ANALOG_PLL_SYS_DENOM field descriptions

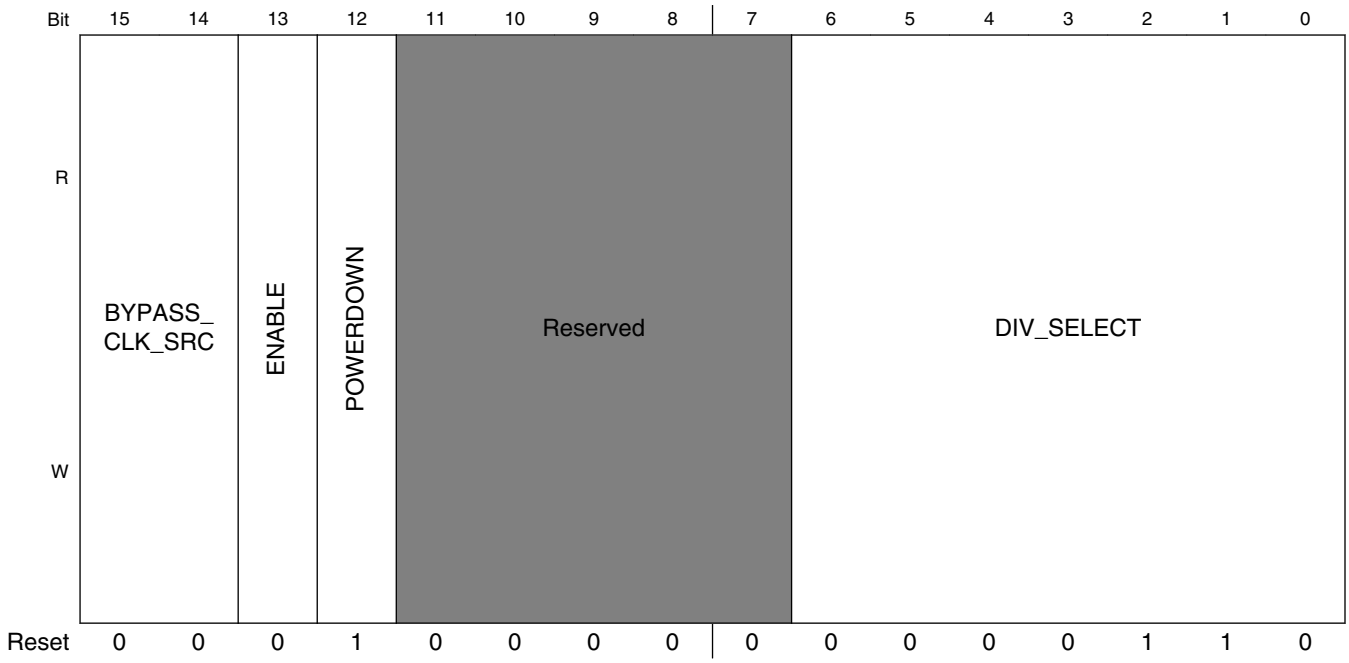
Field	Description
31–30 -	Always set to zero (0).
B	30 bit denominator (B) of fractional loop divider (unsigned integer).

18.7.8 Analog Audio PLL control Register (CCM_ANALOG_PLL_AUDIO*n*)

The control register provides control for the audio PLL.

Address: 20C_8000h base + 70h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	-									Reserved	POST_DIV_SELECT		Reserved		BYPASS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1



CCM_ANALOG_PLL_AUDIO*n* field descriptions

Field	Description
31 LOCK	1 - PLL is currently locked. 0 - PLL is not currently locked.
30–22 -	Always set to zero (0).
21 -	This field is reserved. Reserved
20–19 POST_DIV_ SELECT	These bits implement a divider after the PLL, but before the enable and bypass mux. 00 — Divide by 4. 01 — Divide by 2. 10 — Divide by 1. 11 — Reserved
18–17 -	This field is reserved. Revsered
16 BYPASS	Bypass the PLL.
15–14 BYPASS_CLK_ SRC	Determines the bypass source. 0x0 **REF_CLK_24M** — Select the 24MHz oscillator as source. 0x1 **CLK1** — Select the CLK1_N / CLK1_P as source. 0x2 **CLK2** — Select the CLK2_N / CLK2_P as source. 0x3 **XOR** — Select the XOR of CLK1_N / CLK1_P and CLK2_N / CLK2_P as source.
13 ENABLE	Enable PLL output
12 POWERDOWN	Powers down the PLL.

Table continues on the next page...

CCM_ANALOG_PLL_AUDIO*n* field descriptions (continued)

Field	Description
11–7 -	This field is reserved. Reserved.
DIV_SELECT	This field controls the PLL loop divider. Valid range for DIV_SELECT divider value: 27~54.

18.7.9 Numerator of Audio PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_AUDIO_NUM)

This register contains the numerator (A) of Audio PLL fractional loop divider (Signed number).

Absolute value should be less than denominator

Address: 20C_8000h base + 80h offset = 20C_8080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																A															
W																																
Reset	0	0	0	0	0	1	0	1	1	1	1	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0

CCM_ANALOG_PLL_AUDIO_NUM field descriptions

Field	Description
31–30 -	Always set to zero (0).
A	30 bit numerator of fractional loop divider.

18.7.10 Denominator of Audio PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_AUDIO_DENOM)

This register contains the denominator (B) of Audio PLL fractional loop divider (unsigned number).

Address: 20C_8000h base + 90h offset = 20C_8090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																B															
W																																
Reset	0	0	1	0	1	0	0	1	0	1	1	0	0	1	0	0	0	1	1	0	0	0	0	1	1	0	0	1	1	1	0	0

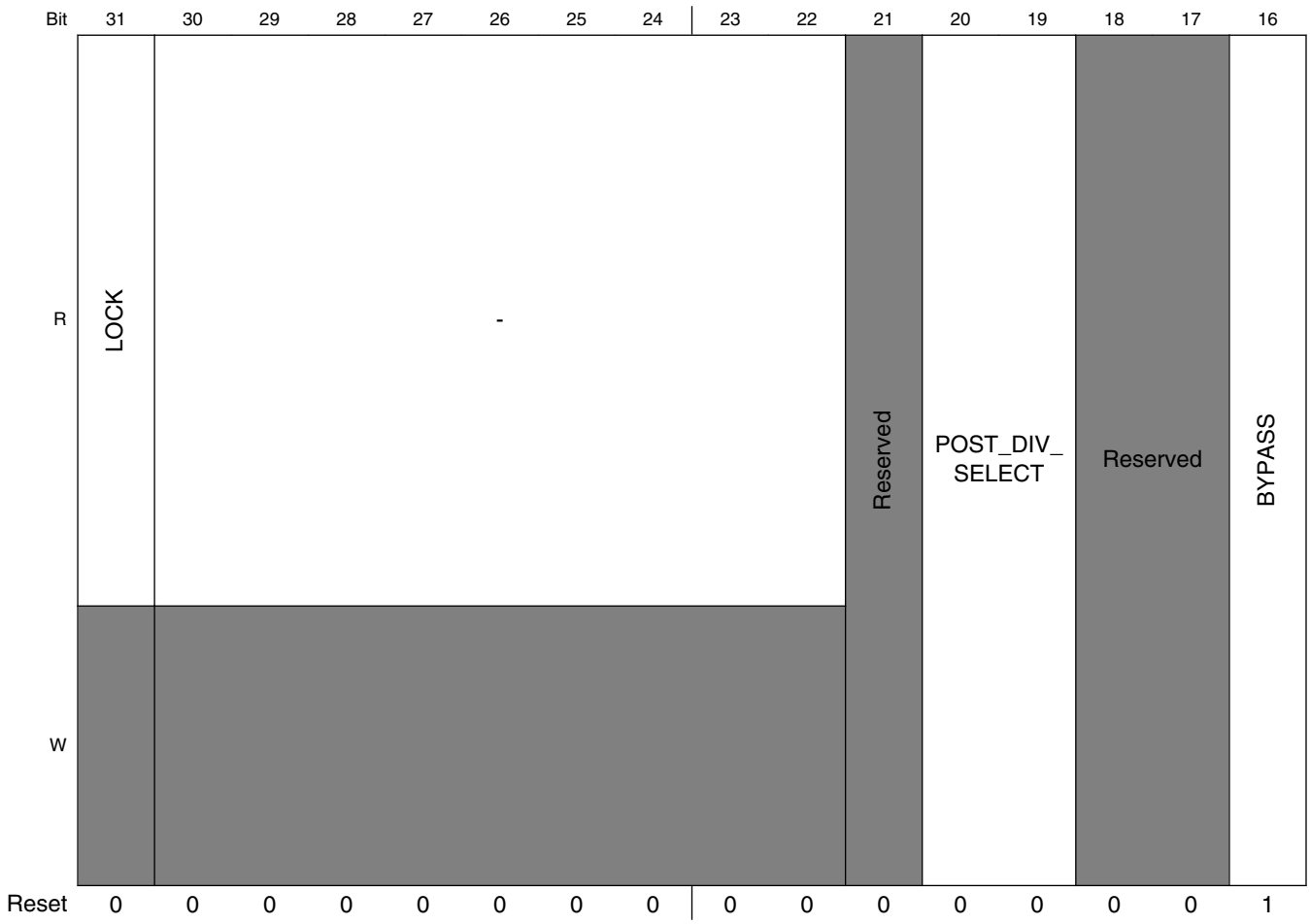
CCM_ANALOG_PLL_AUDIO_DENOM field descriptions

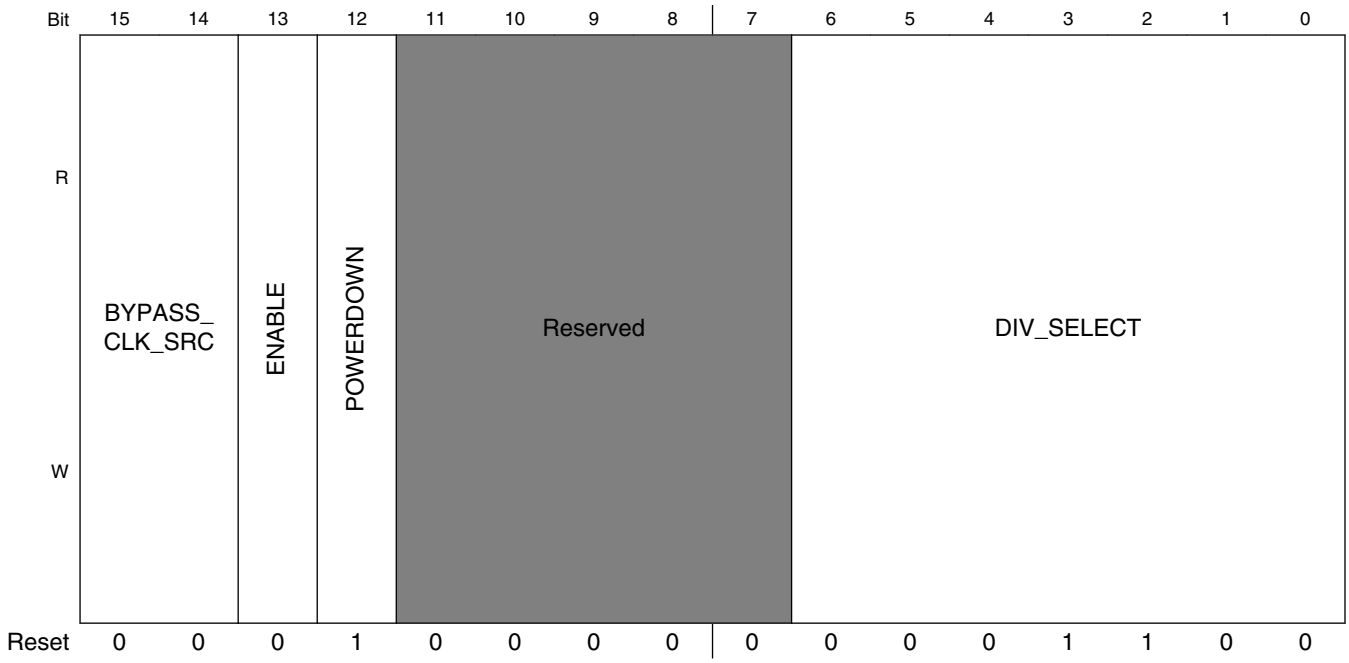
Field	Description
31–30 -	Always set to zero (0).
B	30 bit denominator of fractional loop divider.

18.7.11 Analog Video PLL control Register (CCM_ANALOG_PLL_VIDEO*n*)

The control register provides control for the Video PLL.

Address: 20C_8000h base + A0h offset + (4d × i), where i=0d to 3d





CCM_ANALOG_PLL_VIDEO*n* field descriptions

Field	Description
31 LOCK	1 - PLL is currently locked; 0 - PLL is not currently locked.
30–22 -	Always set to zero (0).
21 -	This field is reserved. Revserved
20–19 POST_DIV_ SELECT	These bits implement a divider after the PLL, but before the enable and bypass mux. 00 — Divide by 4. 01 — Divide by 2. 10 — Divide by 1. 11 — Reserved
18–17 -	This field is reserved. Reserved
16 BYPASS	Bypass the PLL.
15–14 BYPASS_CLK_ SRC	Determines the bypass source. 0x0 **REF_CLK_24M** — Select the 24MHz oscillator as source. 0x1 **CLK1** — Select the CLK1_N / CLK1_P as source. 0x2 **CLK2** — Select the CLK2_N / CLK2_P as source. 0x3 **XOR** — Select the XOR of CLK1_N / CLK1_P and CLK2_N / CLK2_P as source.
13 ENABLE	Enable PLL output
12 POWERDOWN	Powers down the PLL.

Table continues on the next page...

CCM_ANALOG_PLL_VIDEO*n* field descriptions (continued)

Field	Description
11–7 -	This field is reserved. Reserved.
DIV_SELECT	This field controls the PLL loop divider. Valid range for DIV_SELECT divider value: 27~54.

18.7.12 Numerator of Video PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_VIDEO_NUM)

This register contains the numerator (A) of Video PLL fractional loop divider (signed number).

Absolute value should be less than denominator

Address: 20C_8000h base + B0h offset = 20C_80B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																A															
W																																
Reset	0	0	0	0	0	1	0	1	1	1	1	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0

CCM_ANALOG_PLL_VIDEO_NUM field descriptions

Field	Description
31–30 -	Always set to zero (0).
A	30 bit numerator of fractional loop divider (signed number). The absolute value should be less than denominator.

18.7.13 Denominator of Video PLL Fractional Loop Divider Register (CCM_ANALOG_PLL_VIDEO_DENOM)

This register contains the denominator (B) of Video PLL fractional loop divider (unsigned number).

Address: 20C_8000h base + C0h offset = 20C_80C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																B															
W																																
Reset	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1

CCM_ANALOG_PLL_VIDEO_DENOM field descriptions

Field	Description
31–30 -	Always set to zero (0).
B	30 bit denominator of fractional loop divider.

18.7.14 MLB PLL Control Register (CCM_ANALOG_PLL_MLB*n*)

This register defines the control bits for the MLB PLL.

Address: 20C_8000h base + D0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	Reserved		MLB_FLT_RES_CFG			RX_CLK_DELAY_CFG		VDDD_DELAY_CFG				VDDA_DELAY_CFG			BYPASS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		PHASE_SEL		HOLD_RING_OFF	Reserved										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

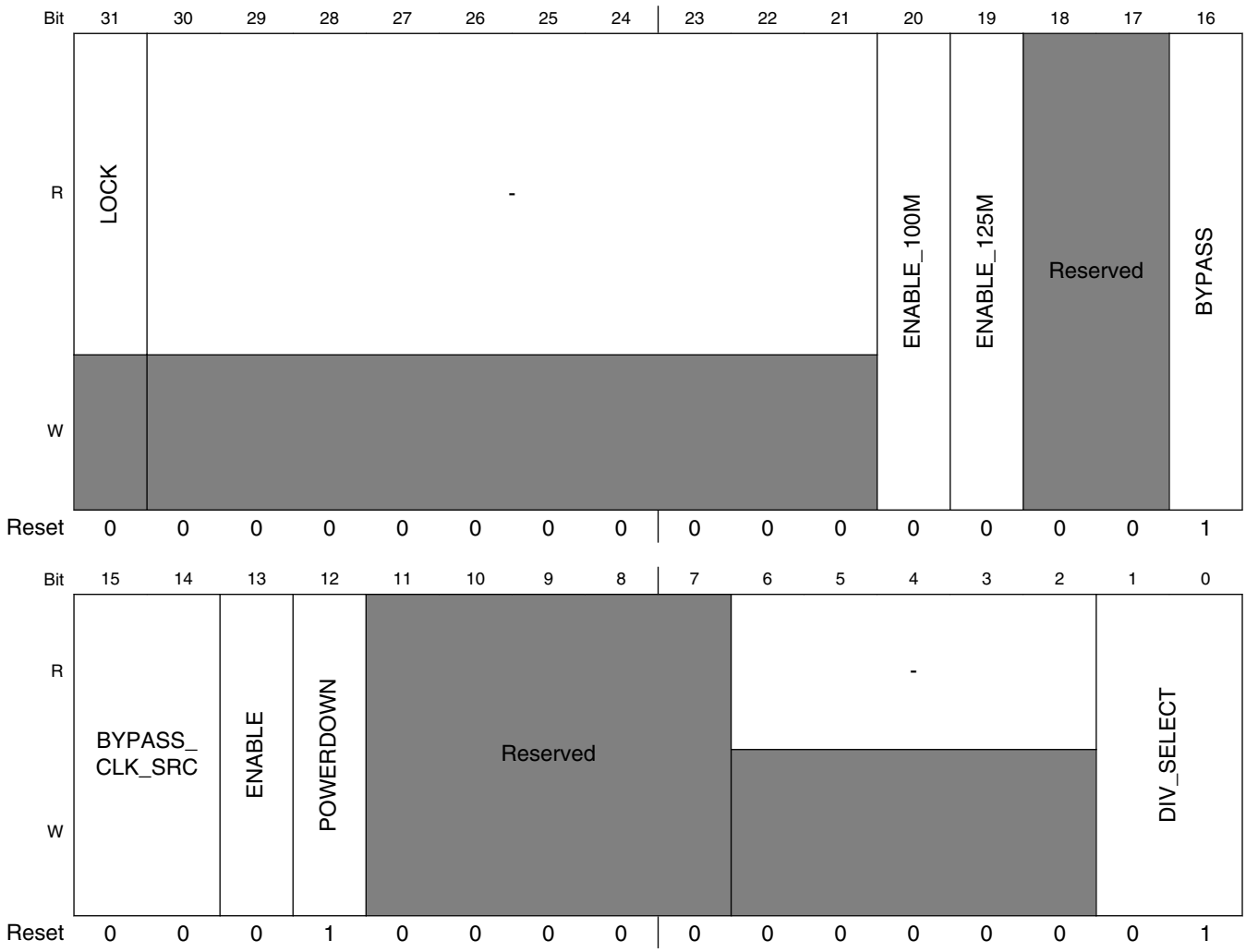
CCM_ANALOG_PLL_MLB*n* field descriptions

Field	Description
31 LOCK	Lock bit 0 PLL is not currently locked 1 PLL is currently locked.
30–29 -	This field is reserved. Reserved.
28–26 MLB_FLT_RES_CFG	Configure the filter resistor for different divider ratio of MLB PLL.
25–23 RX_CLK_DELAY_CFG	Configure the phase delay of the MLB PLL RX Clock.
22–20 VDDD_DELAY_CFG	Configure the phase delay of the MLB PLL by adjusting the delay line in core Vdd poser domain.
19–17 VDDA_DELAY_CFG	Configure the phase delay of the MLB PLL by adjusting the delay line in Vddio power domain.
16 BYPASS	Bypass the PLL.
15–14 -	This field is reserved. Reserved.
13–12 PHASE_SEL	Analog debut bit.
11 HOLD_RING_OFF	Analog debug bit.
-	This field is reserved. Reserved.

18.7.15 Analog ENET PLL Control Register (CCM_ANALOG_PLL_ENET*n*)

The control register provides control for the ENET PLL.

Address: 20C_8000h base + E0h offset + (4d × i), where i=0d to 3d



CCM_ANALOG_PLL_ENET*n* field descriptions

Field	Description
31 LOCK	1 - PLL is currently locked; 0 - PLL is not currently locked.
30–21 -	Always set to zero (0).
20 ENABLE_100M	Enables an offset in the phase frequency detector.

Table continues on the next page...

CCM_ANALOG_PLL_ENET*n* field descriptions (continued)

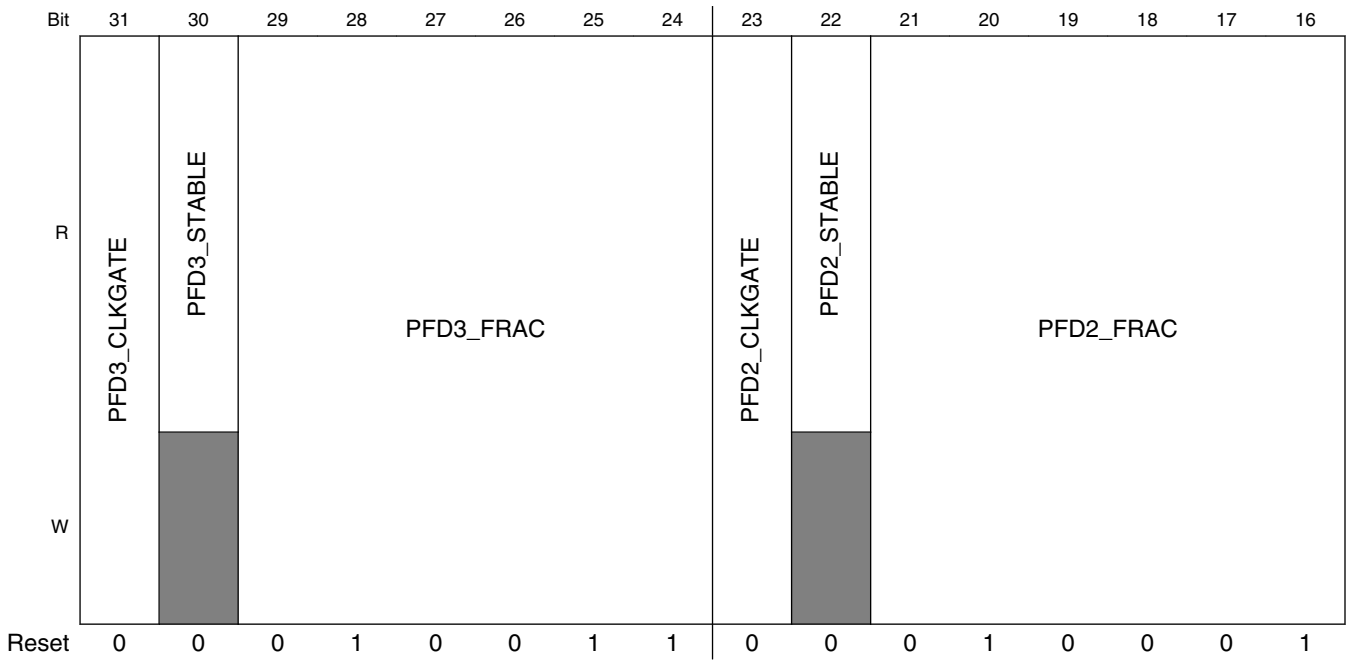
Field	Description
19 ENABLE_125M	Enables an offset in the phase frequency detector.
18–17 -	This field is reserved. Reserved
16 BYPASS	Bypass the PLL.
15–14 BYPASS_CLK_SRC	Determines the bypass source. 0x0 **REF_CLK_24M** — Select the 24MHz oscillator as source. 0x1 **CLK1** — Select the CLK1_N / CLK1_P as source. 0x2 **CLK2** — Select the CLK2_N / CLK2_P as source. 0x3 **XOR** — Select the XOR of CLK1_N / CLK1_P and CLK2_N / CLK2_P as source.
13 ENABLE	Enable the ethernet clock output.
12 POWERDOWN	Powers down the PLL.
11–7 -	This field is reserved. Reserved.
6–2 -	Always set to zero (0).
DIV_SELECT	Controls the frequency of the ethernet reference clock.00 - 25MHz; 01 - 50MHz; 10 - 100MHz (not 50% duty cycle); 11 - 125MHz;

18.7.16 480MHz Clock (PLL3) Phase Fractional Divider Control Register (CCM_ANALOG_PFD_480*n*)

The PFD_480 control register provides control for PFD clock generation.

This register controls the 4-phase fractional clock dividers. The fractional clock frequencies are a product of the values in these registers.

Address: 20C_8000h base + F0h offset + (4d × i), where i=0d to 3d



CCM_ANALOG_PFD_480*n* field descriptions

Field	Description
31 PFD3_CLKGATE	IO Clock Gate. If set to 1, the 3rd fractional divider clock (reference ref_pfd3) is off (power savings). 0: ref_pfd3 fractional divider clock is enabled. Need to assert this bit before PLL is powered down
30 PFD3_STABLE	This read-only bitfield is for DIAGNOSTIC PURPOSES ONLY since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit,

Table continues on the next page...

CCM_ANALOG_PFD_480*n* field descriptions (continued)

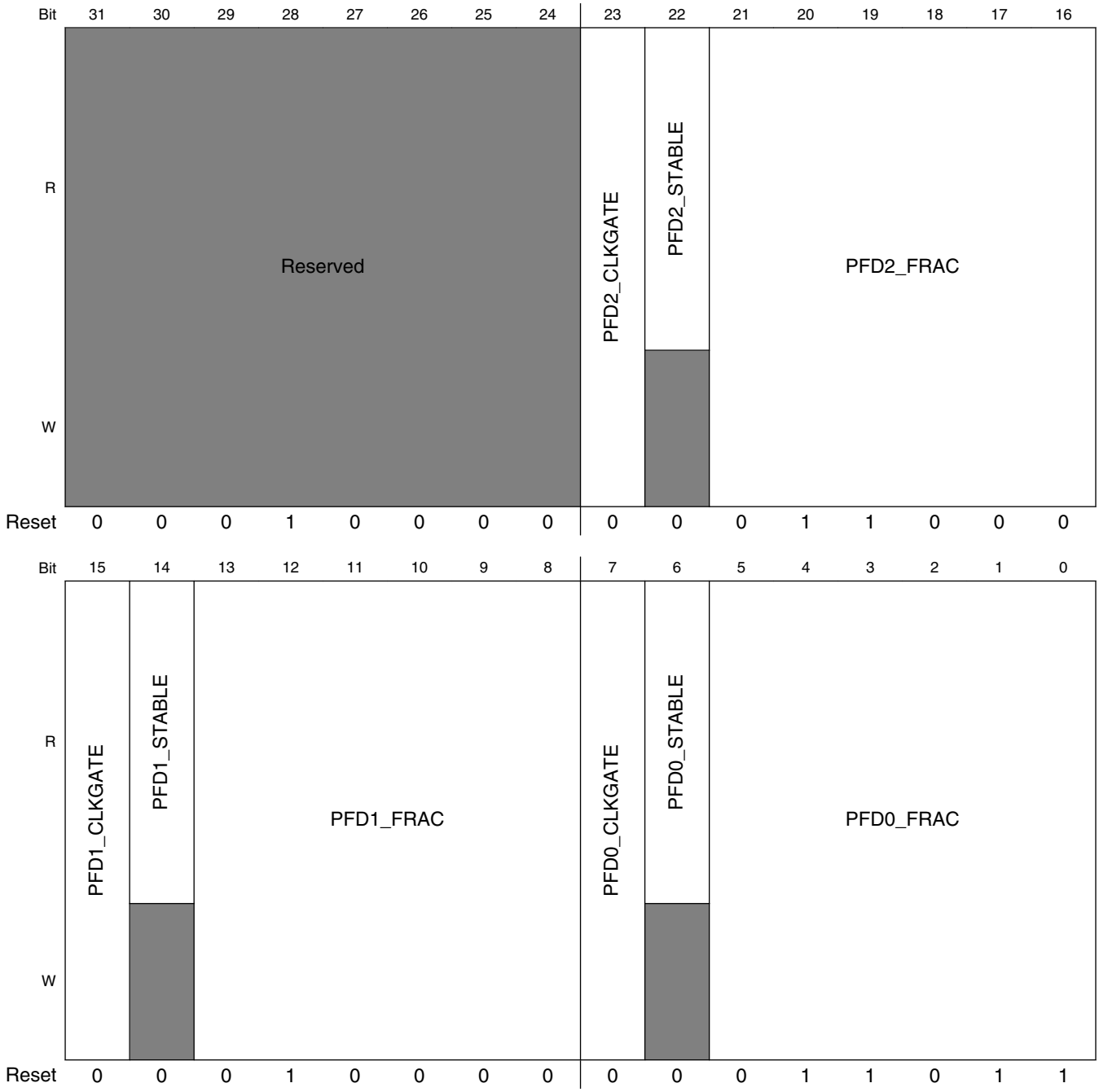
Field	Description
	program the new value, and when this bit inverts, the phase divider clock output is stable. Note that the value will not invert when the fractional divider is taken out of or placed into clock-gated state.
29–24 PFD3_FRAC	This field controls the fractional divide value. The resulting frequency shall be 480*18/PFD3_FRAC where PFD3_FRAC is in the range 12-35.
23 PFD2_CLKGATE	IO Clock Gate. If set to 1, the IO fractional divider clock (reference ref_pfd2) is off (power savings). 0: ref_pfd2 fractional divider clock is enabled. Need to assert this bit before PLL is powered down
22 PFD2_STABLE	This read-only bitfield is for DIAGNOSTIC PURPOSES ONLY since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bit inverts, the phase divider clock output is stable. Note that the value will not invert when the fractional divider is taken out of or placed into clock-gated state.
21–16 PFD2_FRAC	This field controls the fractional divide value. The resulting frequency shall be 480*18/PFD2_FRAC where PFD2_FRAC is in the range 12-35.
15 PFD1_CLKGATE	IO Clock Gate. If set to 1, the IO fractional divider clock (reference ref_pfd1) is off (power savings). 0: ref_pfd1 fractional divider clock is enabled. Need to assert this bit before PLL is powered down
14 PFD1_STABLE	This read-only bitfield is for DIAGNOSTIC PURPOSES ONLY since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bit inverts, the phase divider clock output is stable. Note that the value will not invert when the fractional divider is taken out of or placed into clock-gated state.
13–8 PFD1_FRAC	This field controls the fractional divide value. The resulting frequency shall be 480*18/PFD1_FRAC where PFD1_FRAC is in the range 12-35.
7 PFD0_CLKGATE	If set to 1, the IO fractional divider clock (reference ref_pfd0) is off (power savings). 0: ref_pfd0 fractional divider clock is enabled. Need to assert this bit before PLL is powered down
6 PFD0_STABLE	This read-only bitfield is for DIAGNOSTIC PURPOSES ONLY since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bit inverts, the phase divider clock output is stable. Note that the value will not invert when the fractional divider is taken out of or placed into clock-gated state.
PFD0_FRAC	This field controls the fractional divide value. The resulting frequency shall be 480*18/PFD0_FRAC where PFD0_FRAC is in the range 12-35.

18.7.17 528MHz Clock (PLL2) Phase Fractional Divider Control Register (CCM_ANALOG_PFD_528*n*)

The PFD_528 control register provides control for PFD clock generation.

This register controls the 3-phase fractional clock dividers. The fractional clock frequencies are a product of the values in these registers.

Address: 20C_8000h base + 100h offset + (4d × i), where i=0d to 3d



CCM_ANALOG_PFD_528*n* field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 PFD2_CLKGATE	IO Clock Gate. If set to 1, the IO fractional divider clock (reference ref_pfd2) is off (power savings). 0: ref_pfd2 fractional divider clock is enabled. Need to assert this bit before PLL powered down

Table continues on the next page...

CCM_ANALOG_PFD_528*n* field descriptions (continued)

Field	Description
22 PFD2_STABLE	This read-only bitfield is for DIAGNOSTIC PURPOSES ONLY since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bit inverts, the phase divider clock output is stable. Note that the value will not invert when the fractional divider is taken out of or placed into clock-gated state.
21–16 PFD2_FRAC	This field controls the fractional divide value. The resulting frequency shall be 528*18/PFD2_FRAC where PFD2_FRAC is in the range 12-35.
15 PFD1_CLKGATE	IO Clock Gate. If set to 1, the IO fractional divider clock (reference ref_pfd1) is off (power savings). 0: ref_pfd1 fractional divider clock is enabled. Need to assert this bit before PLL powered down
14 PFD1_STABLE	This read-only bitfield is for DIAGNOSTIC PURPOSES ONLY since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bit inverts, the phase divider clock output is stable. Note that the value will not invert when the fractional divider is taken out of or placed into clock-gated state.
13–8 PFD1_FRAC	This field controls the fractional divide value. The resulting frequency shall be 528*18/PFD1_FRAC where PFD1_FRAC is in the range 12-35.
7 PFD0_CLKGATE	If set to 1, the IO fractional divider clock (reference ref_pfd0) is off (power savings). 0: ref_pfd0 fractional divider clock is enabled. Need to assert this bit before PLL powered down
6 PFD0_STABLE	This read-only bitfield is for DIAGNOSTIC PURPOSES ONLY since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bit inverts, the phase divider clock output is stable. Note that the value will not invert when the fractional divider is taken out of or placed into clock-gated state.
PFD0_FRAC	This field controls the fractional divide value. The resulting frequency shall be 528*18/PFD0_FRAC where PFD0_FRAC is in the range 12-35.

18.7.18 Miscellaneous Register 0 (CCM_ANALOG_MISC0*n*)

This register defines the control and status bits for miscellaneous analog blocks.

Address: 20C_8000h base + 150h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			CLKGATE_DELAY			CLKGATE_CTRL		Reserved				WBCP_VPW_THRESH		OSC_XTALOK_EN	OSC_XTALOK
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OSC_I		discon_high_snvs	STOP_MODE_CONFIG	Reserved				REFTOP_VBGUP	REFTOP_VBGADJ			REFTOP_SELFBIASOFF	Reserved		REFTOP_PWD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_ANALOG_MISC0*n* field descriptions

Field	Description
31–29 -	This field is reserved.
28–26 CLKGATE_ DELAY	This field specifies the delay between powering up the XTAL 24MHz clock and releasing the clock to the digital logic inside the analog block. **NOTE:** Do not change the field during a low power event. This is not a field that the user would normally need to modify. **NOTE:** Not related to CCM. See Crystal Oscillator (XTALOSC) 000 0.5ms 001 1.0ms 010 2.0ms 011 3.0ms 100 4.0ms 101 5.0ms 110 6.0ms 111 7.0ms
25 CLKGATE_CTRL	This bit allows disabling the clock gate (always ungated) for the xtal 24MHz clock that clocks the digital logic in the analog block. **NOTE:** Do not change the field during a low power event. This is not a field that the user would normally need to modify. **NOTE:** Not related to CCM. See Crystal Oscillator (XTALOSC) 0 **ALLOW_AUTO_GATE** — Allow the logic to automatically gate the clock when the XTAL is powered down. 1 **NO_AUTO_GATE** — Prevent the logic from ever gating off the clock.
24–20 -	This field is reserved. Always set to zero.
19–18 WBCP_VPW_ THRESH	This signal alters the voltage that the pwell is charged pumped to. **NOTE:** Not related to CCM. See Power Management Unit (PMU) 00 **NOMINAL_BIAS** — Nominal output pwell bias voltage. 01 **PLUS_25MV** — Increase pwell output voltage by 25mV. 10 **MINUS_25MV** — Decrease pwell output pwell voltage by 25mV. 11 **MINUS_50MV** — Decrease pwell output pwell voltage by 50mV.
17 OSC_XTALOK_ EN	This bit enables the detector that signals when the 24MHz crystal oscillator is stable. **NOTE:** Not related to CCM. See Crystal Oscillator (XTALOSC)
16 OSC_XTALOK	Status bit that signals that the output of the 24-MHz crystal oscillator is stable. Generated from a timer and active detection of the actual frequency. **NOTE:** Not related to CCM. See Crystal Oscillator (XTALOSC)

Table continues on the next page...

CCM_ANALOG_MISC0*n* field descriptions (continued)

Field	Description
15–14 OSC_I	This field determines the bias current in the 24MHz oscillator. The aim is to start up with the highest bias current, which can be decreased after startup if it is determined to be acceptable. **NOTE:** Not related to CCM. See Crystal Oscillator (XTALOSC) 00 **NOMINAL** — Nominal 01 **MINUS_12_5_PERCENT** — Decrease current by 12.5% 10 **MINUS_25_PERCENT** — Decrease current by 25.0% 11 **MINUS_37_5_PERCENT** — Decrease current by 37.5%
13 discon_high_ snvs	This bit forces the short between VDDHIGH_IN and VSNVS_IN to open when asserted. This is useful in power cases where SNVS_IN > VDDHIGH_IN. **NOTE:** Not related to CCM. See Power Management Unit (PMU)
12 STOP_MODE_ CONFIG	Configure the analog behavior in stop mode. 0x0 **DEEP** — Deep Stop Mode - 0x0 All analog except RTC powered down on Stop mode assertion 0x1 **LIGHT** — Light Stop Mode - 0x1 All the analog domain except the LDO_1P1, LDO_2P5, and PLL3 is powered down on STOP mode assertion. If required the CCM can be configured not to power down the oscillator (XTALOSC). PLL3 can be disabled with register settings if desired.
11–8 -	This field is reserved. Reserved
7 REFTOP_ VBGUP	Status bit that signals the analog bandgap voltage is up and stable. 1 - Stable. **NOTE:** Not related to CCM. See Power Management Unit (PMU)
6–4 REFTOP_ VBGADJ	**NOTE:** Not related to CCM. See Power Management Unit (PMU) 000 Nominal VBG 001 VBG+0.78% 010 VBG+1.56% 011 VBG+2.34% 100 VBG-0.78% 101 VBG-1.56% 110 VBG-2.34% 111 VBG-3.12%
3 REFTOP_ SELFBIASOFF	Control bit to disable the self-bias circuit in the analog bandgap. The self-bias circuit is used by the bandgap during startup. This bit should be set after the bandgap has stabilized and is necessary for best noise performance of analog blocks using the outputs of the bandgap. **NOTE:** Value should be returned to zero before removing vddhigh_in or asserting bit 0 of this register (REFTOP_PWD) to assure proper restart of the circuit. **NOTE:** Not related to CCM. See Power Management Unit (PMU) 0 Uses coarse bias currents for startup 1 Uses bandgap-based bias currents for best performance.
2–1 -	This field is reserved.

Table continues on the next page...

CCM_ANALOG_MISC0*n* field descriptions (continued)

Field	Description
0 REFTOP_PWD	Control bit to power-down the analog bandgap reference circuitry. **NOTE:** A note of caution, the bandgap is necessary for correct operation of most of the LDO, PLL, and other analog functions on the die. **NOTE:** Not related to CCM. See Power Management Unit (PMU)

18.7.19 Miscellaneous Register 1 (CCM_ANALOG_MISC1*n*)

This register defines the control and status bits for miscellaneous analog blocks. The LVDS1 and LVDS2 controls below control the behavior of the anaclk1/1b and anaclk2/2b LVDS IO's.

Address: 20C_8000h base + 160h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_DIG_BO	IRQ_ANA_BO	IRQ_TEMPSENSE	Reserved												
W	w1c	w1c	w1c													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		LVDSCLK2_IBEN	LVDSCLK1_IBEN	LVDSCLK2_OBEN	LVDSCLK1_OBEN	LVDS2_CLK_SEL					LVDS1_CLK_SEL				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_ANALOG_MISC1*n* field descriptions

Field	Description
31 IRQ_DIG_BO	This status bit is set to one when when any of the digital regulator brownout interrupts assert. Check the regulator status bits to discover which regulator interrupt asserted. **NOTE:** Not related to CCM. See Power Management Unit (PMU)

Table continues on the next page...

CCM_ANALOG_MISC1*n* field descriptions (continued)

Field	Description
30 IRQ_ANA_BO	This status bit is set to one when when any of the analog regulator brownout interrupts assert. Check the regulator status bits to discover which regulator interrupt asserted. **NOTE:** Not related to CCM. See Power Management Unit (PMU)
29 IRQ_ TEMPSENSE	This status bit is set to one when when the temperature sensor interrupt asserts. **NOTE:** Not related to CCM. See Temperature Monitor (TEMPMON)
28–14 -	This field is reserved.
13 LVDSCLK2_ IBEN	This enables the LVDS input buffer for anaclk2/2b. Do not enable input and output buffers simultaneously.
12 LVDSCLK1_ IBEN	This enables the LVDS input buffer for anaclk1/1b. Do not enable input and output buffers simultaneously.
11 LVDSCLK2_ OBEN	This enables the LVDS output buffer for anaclk2/2b. Do not enable input and output buffers simultaneously.
10 LVDSCLK1_ OBEN	This enables the LVDS output buffer for anaclk1/1b. Do not enable input and output buffers simultaneously.
9–5 LVDS2_CLK_ SEL	This field selects the clk to be routed to anaclk2/2b. 00000 **ARM_PLL** — Arm PLL 00001 **SYS_PLL** — System PLL 00010 **PFD4** — ref_pfd4_clk == pll2_pfd0_clk 00011 **PFD5** — ref_pfd5_clk == pll2_pfd1_clk 00100 **PFD6** — ref_pfd6_clk == pll2_pfd2_clk 00101 **PFD7** — ref_pfd7_clk == pll2_pfd3_clk 00110 **AUDIO_PLL** — Audio PLL 00111 **VIDEO_PLL** — Video PLL 01000 **MLB_PLL** — MLB PLL 01001 **ETHERNET_REF** — ethernet ref clock (ENET_PLL) 01010 **PCIE_REF** — PCIe ref clock (125M) 01011 **REF_100M** — REF_100M clock 01100 **USB1_PLL** — USB1 PLL clock 01101 **USB2_PLL** — USB2 PLL clock 01110 **PFD0** — ref_pfd0_clk == pll3_pfd0_clk 01111 **PFD1** — ref_pfd1_clk == pll3_pfd1_clk 10000 **PFD2** — ref_pfd2_clk == pll3_pfd2_clk 10001 **PFD3** — ref_pfd3_clk == pll3_pfd3_clk 10010 **XTAL** — xtal (24M) 10011 **LVDS1** — LVDS1 (loopback) 10100 **LVDS2** — LVDS2 (not useful) 10101 to 11111 ref_pfd7_clk == pll2_pfd3_clk
LVDS1_CLK_ SEL	This field selects the clk to be routed to anaclk1/1b.

Table continues on the next page...

CCM_ANALOG_MISC1*n* field descriptions (continued)

Field	Description	
	00000	**ARM_PLL** — Arm PLL
	00001	**SYS_PLL** — System PLL
	00010	**PFD4** — ref_pfd4_clk == pll2_pfd0_clk
	00011	**PFD5** — ref_pfd5_clk == pll2_pfd1_clk
	00100	**PFD6** — ref_pfd6_clk == pll2_pfd2_clk
	00101	**PFD7** — ref_pfd7_clk == pll2_pfd3_clk
	00110	**AUDIO_PLL** — Audio PLL
	00111	**VIDEO_PLL** — Video PLL
	01000	**MLB_PLL** — MLB PLL
	01001	**ETHERNET_REF** — ethernet ref clock (ENET_PLL)
	01010	**PCIE_REF** — PCIe ref clock (125M)
	01100	**USB1_PLL** — USB1 PLL clock
	01101	**USB2_PLL** — USB2 PLL clock
	01110	**PFD0** — ref_pfd0_clk == pll3_pfd0_clk
	01111	**PFD1** — ref_pfd1_clk == pll3_pfd1_clk
	10000	**PFD2** — ref_pfd2_clk == pll3_pfd2_clk
	10001	**PFD3** — ref_pfd3_clk == pll3_pfd3_clk
	10010	**XTAL** — xtal (24M)
	10011	**LVDS1** — LVDS1 (loopback)
	10100	**LVDS2** — LVDS2 (not useful)
	10101 to 11111	ref_pfd7_clk == pll2_pfd3_clk

18.7.20 Miscellaneous Register 2 (CCM_ANALOG_MISC2*n*)

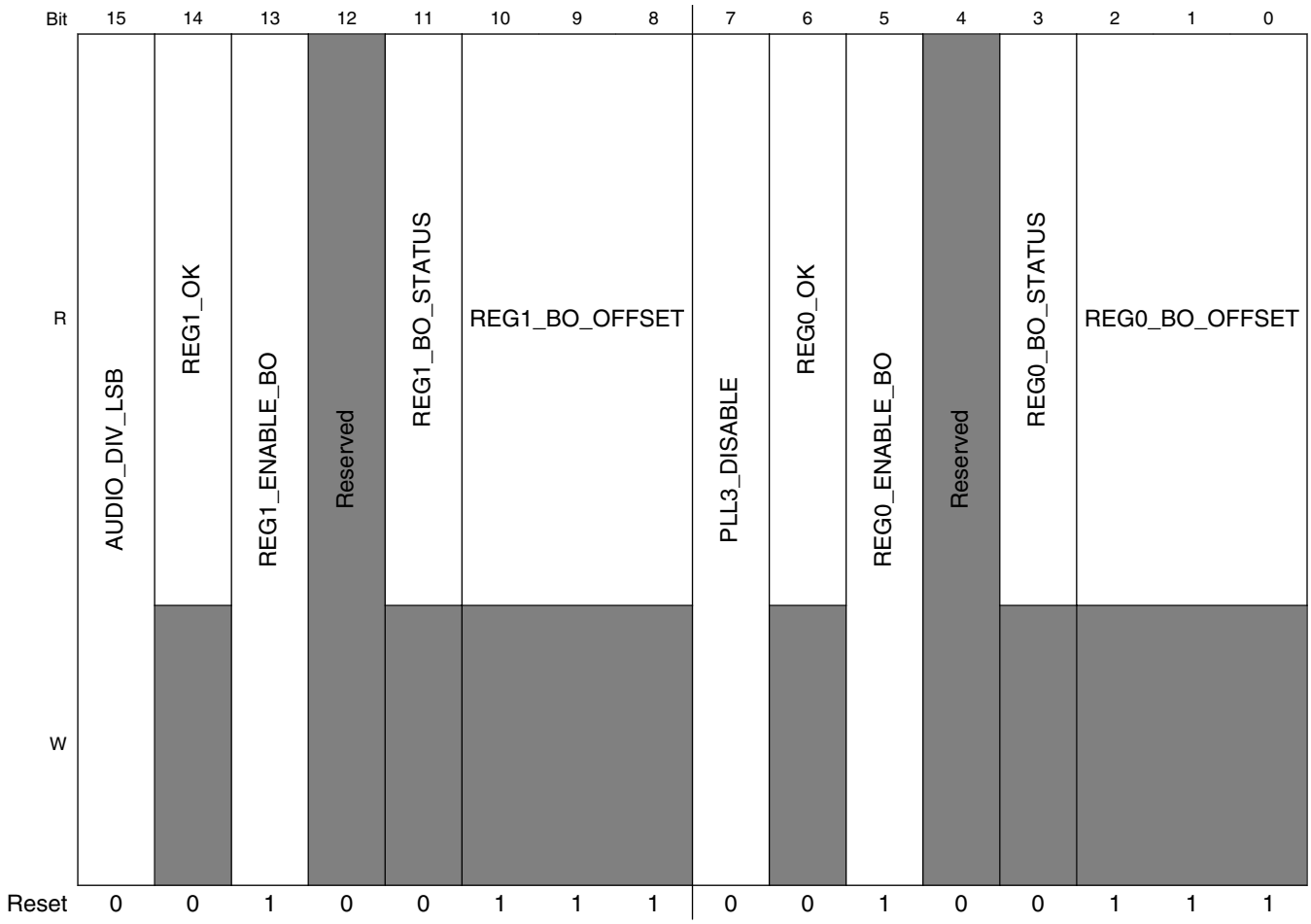
This register defines the control for miscellaneous analog blocks.

NOTE

This register is shared with PMU.

Address: 20C_8000h base + 170h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VIDEO_DIV		REG2_STEP_TIME		REG1_STEP_TIME		REG0_STEP_TIME		AUDIO_DIV_MSB	REG2_OK	REG2_ENABLE_BO	Reserved	REG2_BO_STATUS	REG2_BO_OFFSET		
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1



CCM_ANALOG_MISC2*n* field descriptions

Field	Description
31–30 VIDEO_DIV	Post-divider for video. The output clock of the video PLL should be gated prior to changing this divider to prevent glitches. This divider is feed by PLL_VIDEOn[POST_DIV_SELECT] to achieve division ratios of /1, /2, /4. 00 divide by 1 (Default) 01 divide by 2 10 divide by 1 11 divide by 4
29–28 REG2_STEP_ TIME	Number of clock periods (24MHz clock). **NOTE:** Not related to CCM. See Power Management Unit (PMU) 00 **64_CLOCKS** — 64 01 **128_CLOCKS** — 128 10 **256_CLOCKS** — 256 11 **512_CLOCKS** — 512
27–26 REG1_STEP_ TIME	Number of clock periods (24MHz clock).

Table continues on the next page...

CCM_ANALOG_MISC2*n* field descriptions (continued)

Field	Description
	NOTE: Not related to CCM. See Power Management Unit (PMU) 00 **64_CLOCKS** — 64 01 **128_CLOCKS** — 128 10 **256_CLOCKS** — 256 11 **512_CLOCKS** — 512
25–24 REG0_STEP_TIME	Number of clock periods (24MHz clock). **NOTE:** Not related to CCM. See Power Management Unit (PMU) 00 **64_CLOCKS** — 64 01 **128_CLOCKS** — 128 10 **256_CLOCKS** — 256 11 **512_CLOCKS** — 512
23 AUDIO_DIV_MSB	MSB of Post-divider for Audio PLL. The output clock of the video PLL should be gated prior to changing this divider to prevent glitches. This divider is feed by PLL_AUDIOn[POST_DIV_SELECT] to achieve division ratios of /1, /2, /4. **NOTE:** MSB bit value pertains to the first bit, please program the LSB bit (bit 15) as well to change divider value for more information. 00 divide by 1 (Default) 01 divide by 2 10 divide by 1 11 divide by 4
22 REG2_OK	Signals that the voltage is above the brownout level for the SOC supply. 1 = regulator output > brownout_target **NOTE:** Not related to CCM. See Power Management Unit (PMU)
21 REG2_ENABLE_BO	Enables the brownout detection. **NOTE:** Not related to CCM. See Power Management Unit (PMU)
20 -	This field is reserved.
19 REG2_BO_STATUS	Reg2 brownout status bit. **NOTE:** Not related to CCM. See Power Management Unit (PMU)
18–16 REG2_BO_OFFSET	This field defines the brown out voltage offset for the xPU power domain. IRQ_DIG_BO is also asserted. Single-bit increments reflect 25mV brownout voltage steps. The reset brown-offset is 175mV below the programmed target code. Brownout target = OUTPUT_TRG - BO_OFFSET. Some steps may be irrelevant because of input supply limitations or load operation. **NOTE:** Not related to CCM. See Power Management Unit (PMU) 100 Brownout offset = 0.100V 111 Brownout offset = 0.175V

Table continues on the next page...

CCM_ANALOG_MISC2*n* field descriptions (continued)

Field	Description
15 AUDIO_DIV_LSB	LSB of Post-divider for Audio PLL. The output clock of the video PLL should be gated prior to changing this divider to prevent glitches. This divider is feed by PLL_AUDIOn[POST_DIV_SELECT] to achieve division ratios of /1, /2, /4. **NOTE:** LSB bit value pertains to the last bit, please program the MSB bit (bit 23) as well, to change divider value for more information. 00 divide by 1 (Default) 01 divide by 2 10 divide by 1 11 divide by 4
14 REG1_OK	GPU/VPU supply **NOTE:** Not related to CCM. See Power Management Unit (PMU)
13 REG1_ENABLE_BO	Enables the brownout detection. **NOTE:** Not related to CCM. See Power Management Unit (PMU)
12 -	This field is reserved.
11 REG1_BO_STATUS	Reg1 brownout status bit. **NOTE:** Not related to CCM. See Power Management Unit (PMU) 1 Brownout, supply is below target minus brownout offset.
10–8 REG1_BO_OFFSET	This field defines the brown out voltage offset for the xPU power domain. IRQ_DIG_BO is also asserted. Single-bit increments reflect 25mV brownout voltage steps. The reset brown-offset is 175mV below the programmed target code. Brownout target = OUTPUT_TRG - BO_OFFSET. Some steps may be irrelevant because of input supply limitations or load operation. **NOTE:** Not related to CCM. See Power Management Unit (PMU) 100 Brownout offset = 0.100V 111 Brownout offset = 0.175V
7 PLL3_DISABLE	When USB is in low power suspend mode this Control bit is used to indicate if other system peripherals require the USB PLL3 clock when the SoC is not in low power mode. A user needs to set this bit if they want to optionally disable PLL3 while the SoC is not in any low power mode to save power. When the system does go into low power mode this bit setting would not have any affect. **NOTE:** When USB is in low power suspend mode users would need to ensure PLL3 is not being used before setting this bit in RUN mode. Please refer to the correct PLL disabling procedure in Disabling / Enabling PLLs 0 PLL3 is being used by peripherals and is enabled when SoC is not in any low power mode 1 PLL3 can be disabled when the SoC is not in any low power mode
6 REG0_OK	Arm supply **NOTE:** Not related to CCM. See Power Management Unit (PMU)

Table continues on the next page...

CCM_ANALOG_MISC2*n* field descriptions (continued)

Field	Description
5 REG0_ENABLE_BO	Enables the brownout detection. **NOTE:** Not related to CCM. See Power Management Unit (PMU)
4 -	This field is reserved.
3 REG0_BO_STATUS	Reg0 brownout status bit. **NOTE:** Not related to CCM. See Power Management Unit (PMU) 1 Brownout, supply is below target minus brownout offset.
REG0_BO_OFFSET	This field defines the brown out voltage offset for the CORE power domain. IRQ_DIG_BO is also asserted. Single-bit increments reflect 25mV brownout voltage steps. Some steps may be irrelevant because of input supply limitations or load operation. **NOTE:** Not related to CCM. See Power Management Unit (PMU) 100 Brownout offset = 0.100V 111 Brownout offset = 0.175V

Chapter 19
MIPI CSI to IPU Gasket (CSI2IPU)

19.1 Overview

The CSI2IPU gasket is a digital core that functions as a gasket interface between the MIPI CSI-2 host controller and the IPU system. This facilitates communication between a MIPI CSI-2 compliant camera sensor and IPU (the image processing unit). The gasket's main functions are to synchronize the CSI-2 input 32-bit data bus with the 16-bit data bus and to separate the four virtual channels.

The following block diagram shows the CSI2IPU gasket's position in the system as part of the CSI to IPU connectivity.

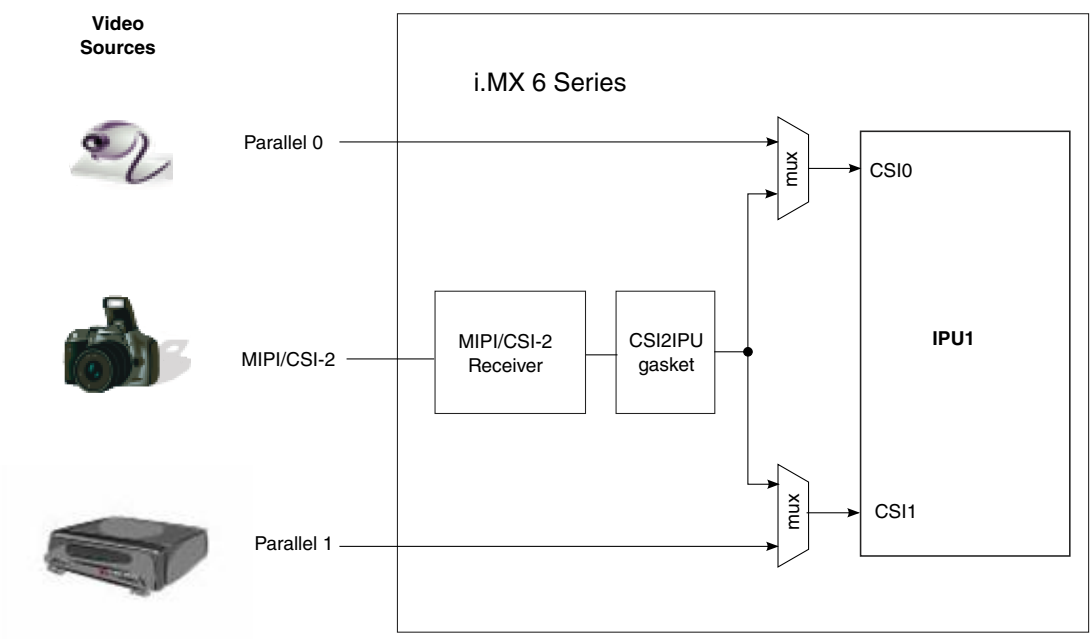


Figure 19-1. CSI2IPU gasket connectivity

19.1.1 CSI2IPU feature summary

The CSI2IPU gasket supports:

- Up to 4 virtual channels of MIPI CSI-2 host controller
- All data types of the MIPI Alliance Standard for Camera Serial Interface (CSI)
- Dynamically configurable pixel clock gating or non-gating for the IPU module
- Dynamically configurable RGB444 and YUV422 data format for the IPU module
- A software reset to reset the program during operation

19.1.2 CSI2IPU architectural description

The following figure shows the overall architecture of the CSI2IPU gasket.

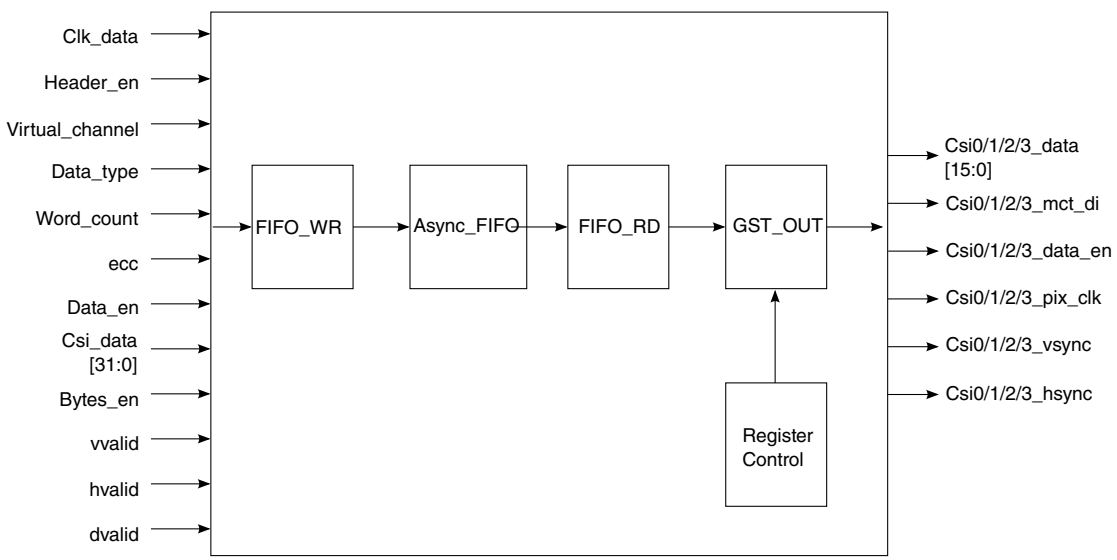


Figure 19-2. CSI2IPU gasket architecture

The main blocks are:

- FIFO WRITE—Responsible for receiving the CSI-2 host input signals and integrating them into a wr_data[46:0] bus
- ASYNC FIFO—Used to synchronize the wr_data to the read clock domain. It also generates FIFO full and empty signals.
- FIFO READ—Responsible for reading FIFO output and changing the 32-bit data bus to the 16-bit data bus.

- GST OUT—Generates output data according to the format controlled by the register bank.
- REGISTER CONTROL—register bank used to control the GST OUT block.

19.2 CSI2IPU signals

The following tables describe the CSI2IPU gasket's input and output signals.

Table 19-1. Input signals to CSI2IPU

Signal	Description
clk_data	Clock input for the CSI host controller
header_en	Shows that the header data at the input (signals virtual_channel, data_type, word_count, and ecc) is valid for the packet being transferred. This signal stays HIGH during the complete packet transfer.
virtual_channel[1:0]	Virtual channel identifier value • 00— VC0 • 01—VC1 • 10—VC2 • 11—VC3
data_type[5:0]	• 00h–07h—Synchronization short packet data types • 08h–0Fh—Generic short packet data types • 10h–17h—Generic long packet data types • 18h–1Fh—YUV data • 20h–27h—RGB data • 28h–2Fh—RAW data • 30h– 37h—User defined byte-based data • 38h–3Fh—Reserved
word_count[15:0]	16-bit word count information from the packet header
ecc[7:0]	8-bit error correction code for the packet header
data_en	Shows that new payload data is present at the input (signals csi_data and bytes_en). This signal is only asserted when receiving long packets. Note that the signal header_en is also asserted when data_en is asserted.
bytes_en[1:0]	Shows how may bytes are in the csi_data output signal • 00 = 1 valid byte in csi_data[7:0] • 01 = 2 valid bytes in csi_data[15:0] • 10 = 3 valid bytes in csi_data[23:0] • 11 = 4 valid bytes in csi_data[31:0]
vvalid[3:0]	This signal is asserted when a Frame Start is detected and de-asserted when a Frame End is detected. Each virtual channel has one valid signal. • vvalid[0]—vvalid for Virtual Channel 0 • vvalid[1]—vvalid for Virtual Channel 1 • vvalid[2]—vvalid for Virtual Channel 2 • vvalid[3]—vvalid for Virtual Channel 3
hvalid[3:0]	This signal is asserted when a Line Start is detected and de-asserted when a Line End is detected. Line Start and Line End are optional; when they are not available, hvalid has the same behavior as dvalid. Each virtual channel has one valid signal. • hvalid[0]—hvalid for Virtual Channel 0

Table continues on the next page...

Table 19-1. Input signals to CSI2IPU (continued)

Signal	Description
	• hvalid[1]—hvalid for Virtual Channel 1 • hvalid[2]—hvalid for Virtual Channel 2 • hvalid[3]—hvalid for Virtual Channel 3
dvalid[3:0]	Used to signal when valid data is available
csi_data_out[31:0]	Payload data
hw_resetn	System hardware reset
sft_resetn	System software reset from top module
gst_clk_sel	• 0—Gating clock mode • 1—Non-gating clock mode
gst_rgb444_fm	Rgb444 output format select • 0—{4'h0,r4g4b4} • 1—{r4,1'b0,g4,1'b0,b4,1'b0}
gst_yuv422_8bit_fm	Yuv422 output format select • 0—output YUYV • 1—output UYVY
ccm_pixel_clk	Reading clock from CCM

Table 19-2. CSI2IPU output signals to IPU

Signal	Description
csi*n*[1] _data[15:0]	Virtual channel *n*[1] CSI data
csi*n*[1] _mct_di[7:0]	Virtual channel *n*[1] {Channel_ID[1:0],Data type[5:0]}
csi*n*[1] _data_en	virtual channel *n*[1] data enable signal
csi*n*[1] _pix_clk	virtual channel *n*[1] pixel clock signal clock
csi*n*[1] _vsync	virtual channel *n*[1] vertical synchronism signal(Frame start)
csi*n*[1] _hsync	virtual channel *n*[1] horizontal synchronism signal
csi*n*[1] _Byte_en	virtual channel *n*[1] byte enable • 0–[7:0] valid • 1:[15:0] valid

1. n = 0 - 3

19.3 Timing interface

The following figure shows the timing for the image data interface.

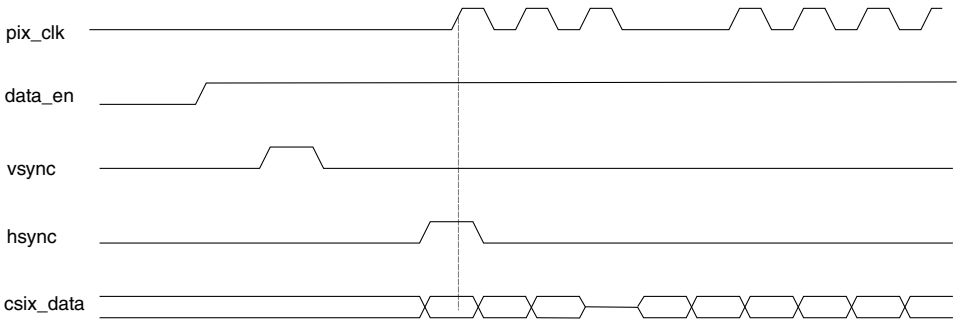


Figure 19-3. Image data interface

19.4 Payload data output formats

The following figures illustrate the formats for the payload data output.

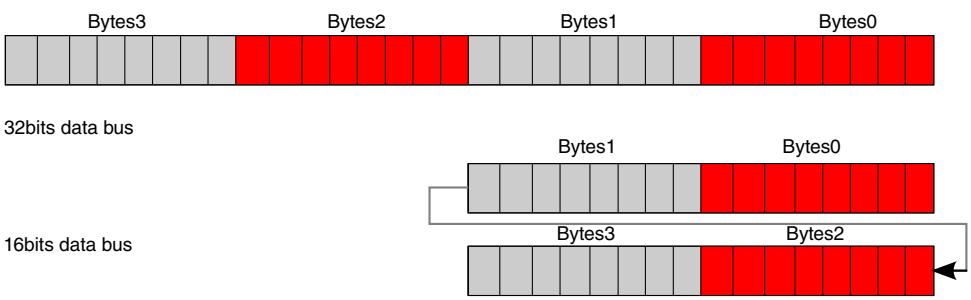


Figure 19-4. General/arbitrary data reception

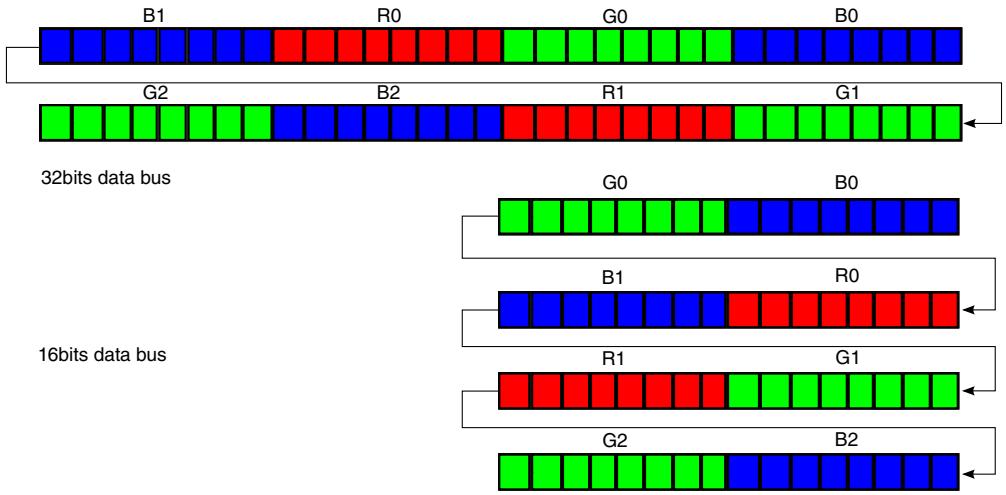


Figure 19-5. RGB888 data reception

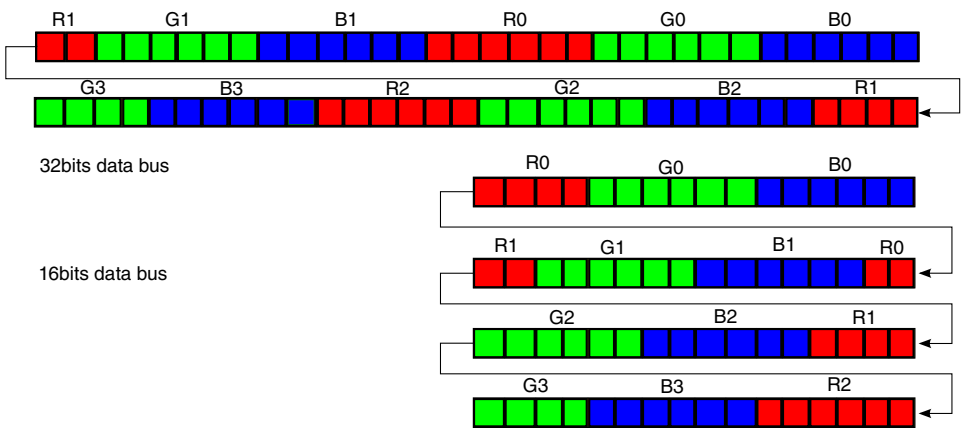


Figure 19-6. RGB666 data reception

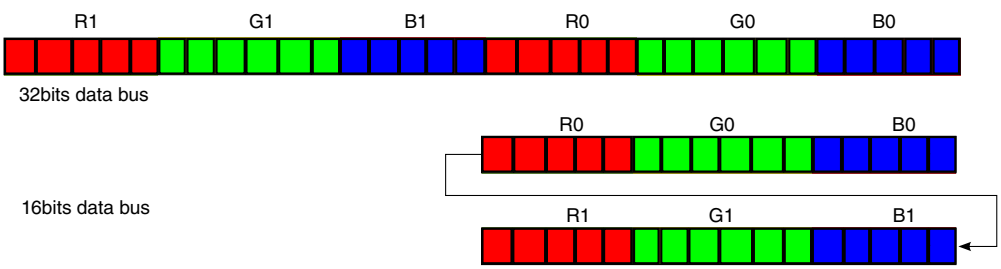


Figure 19-7. RGB565 data reception

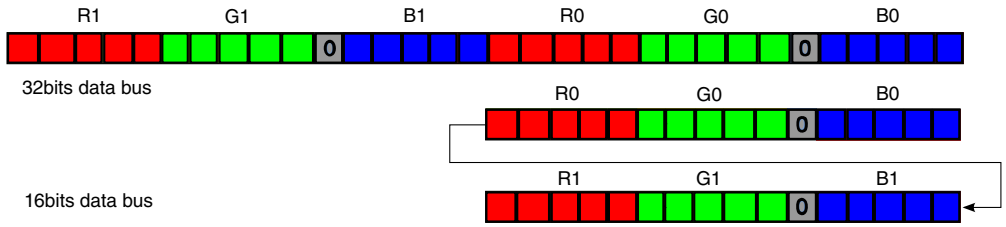


Figure 19-8. RGB555 data reception

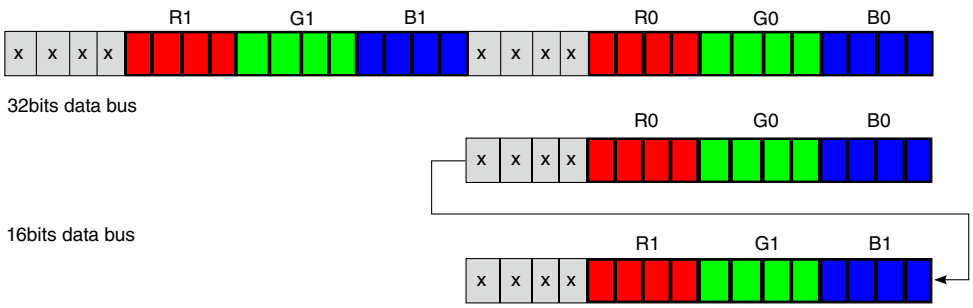


Figure 19-9. RGB444 data reception

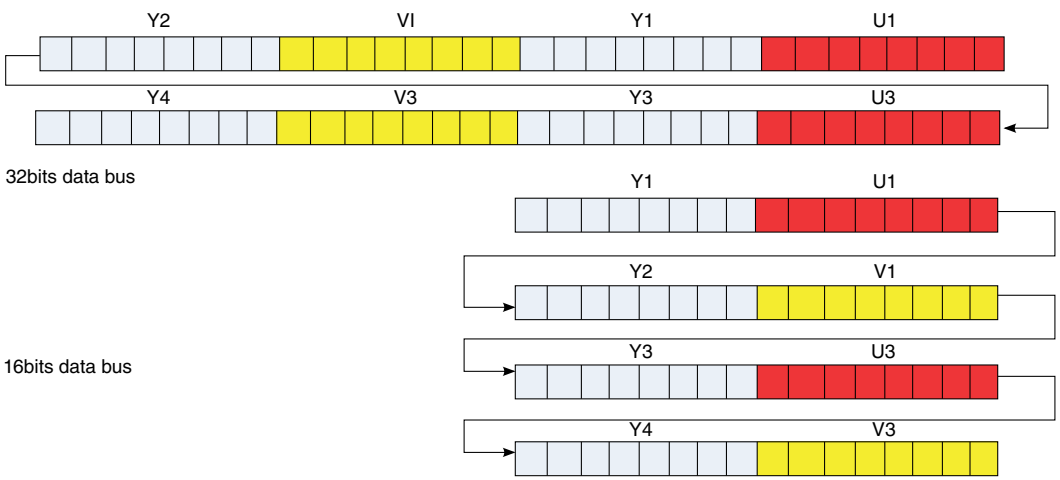


Figure 19-10. YUV422-8 data reception

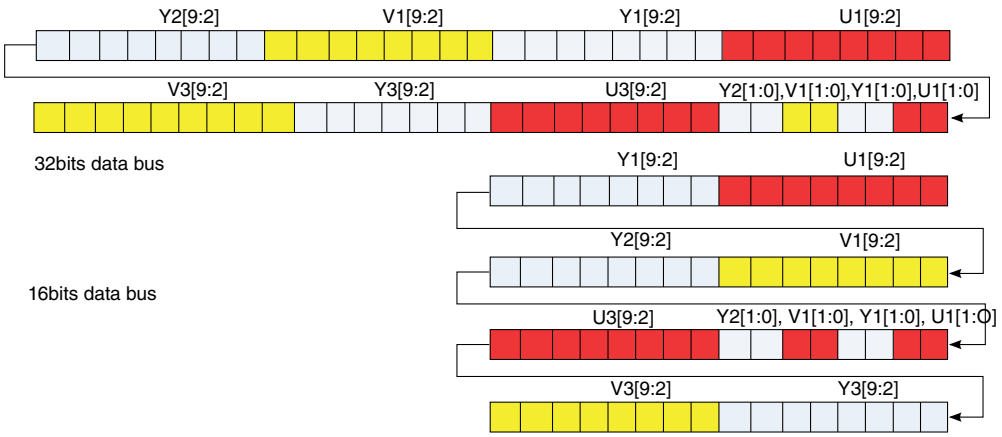


Figure 19-11. YUV422-10 data reception

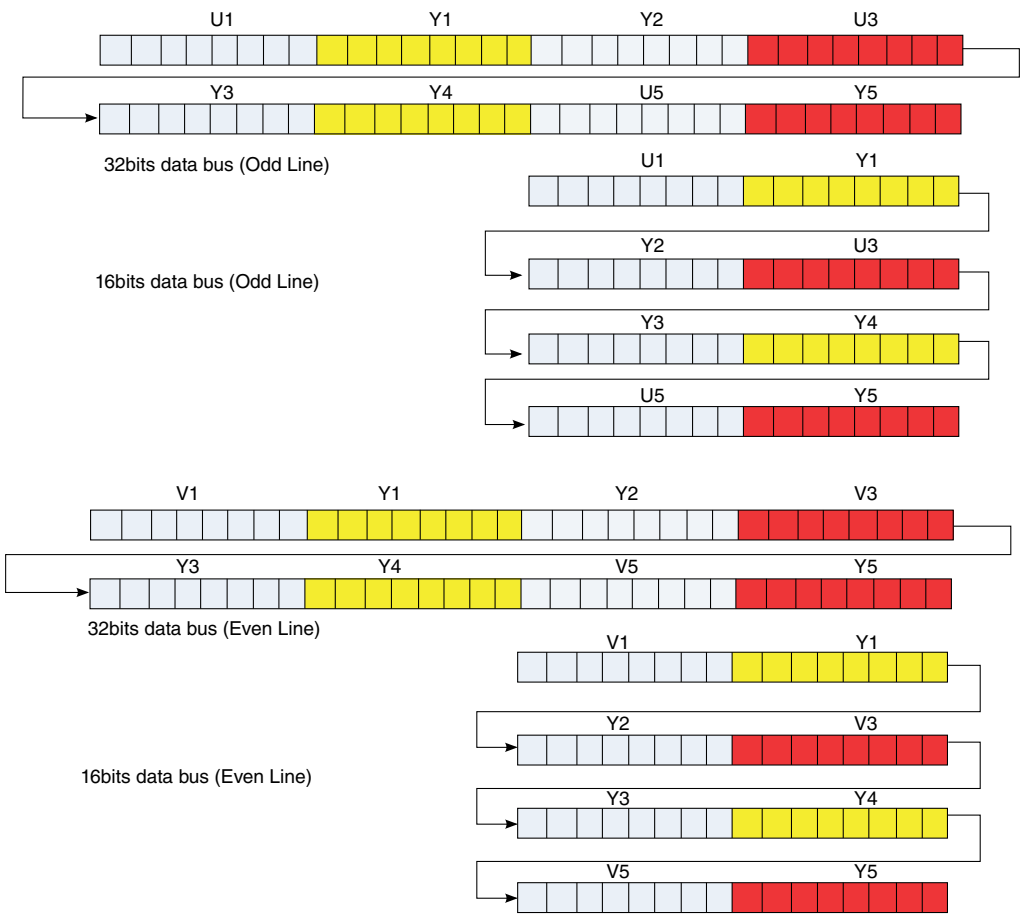


Figure 19-12. YUV420-8 (legacy) data reception

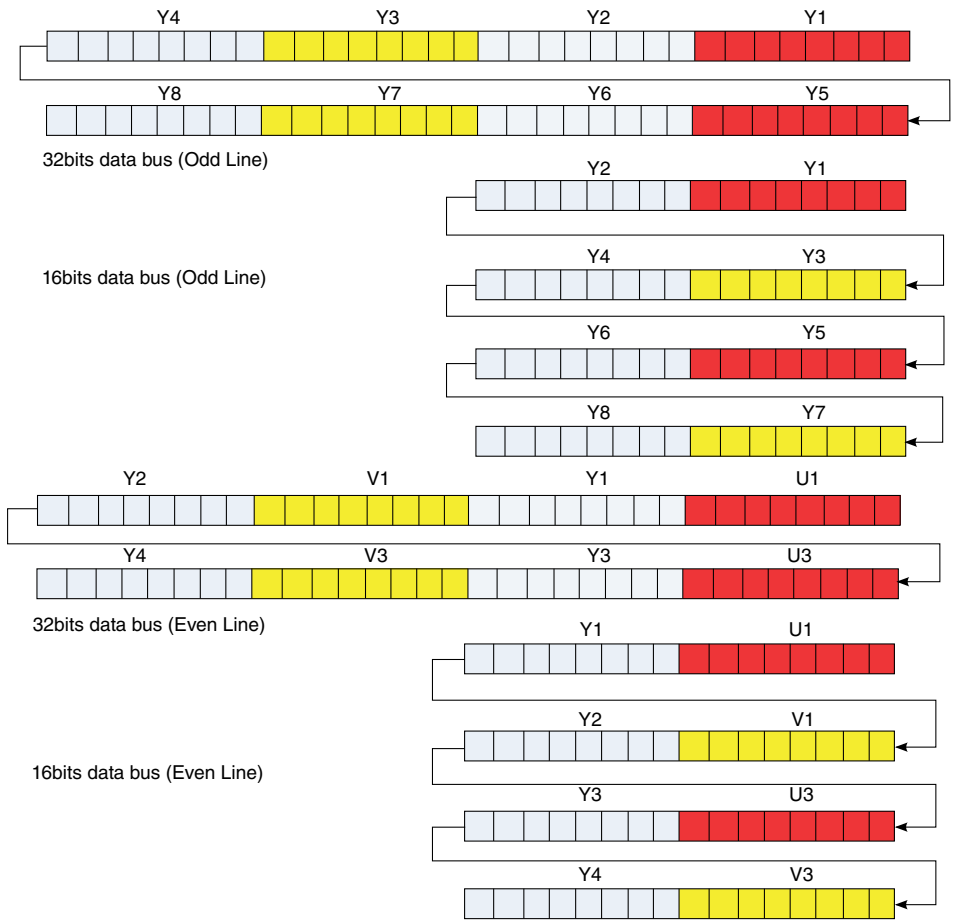


Figure 19-13. YUV420-8 data reception

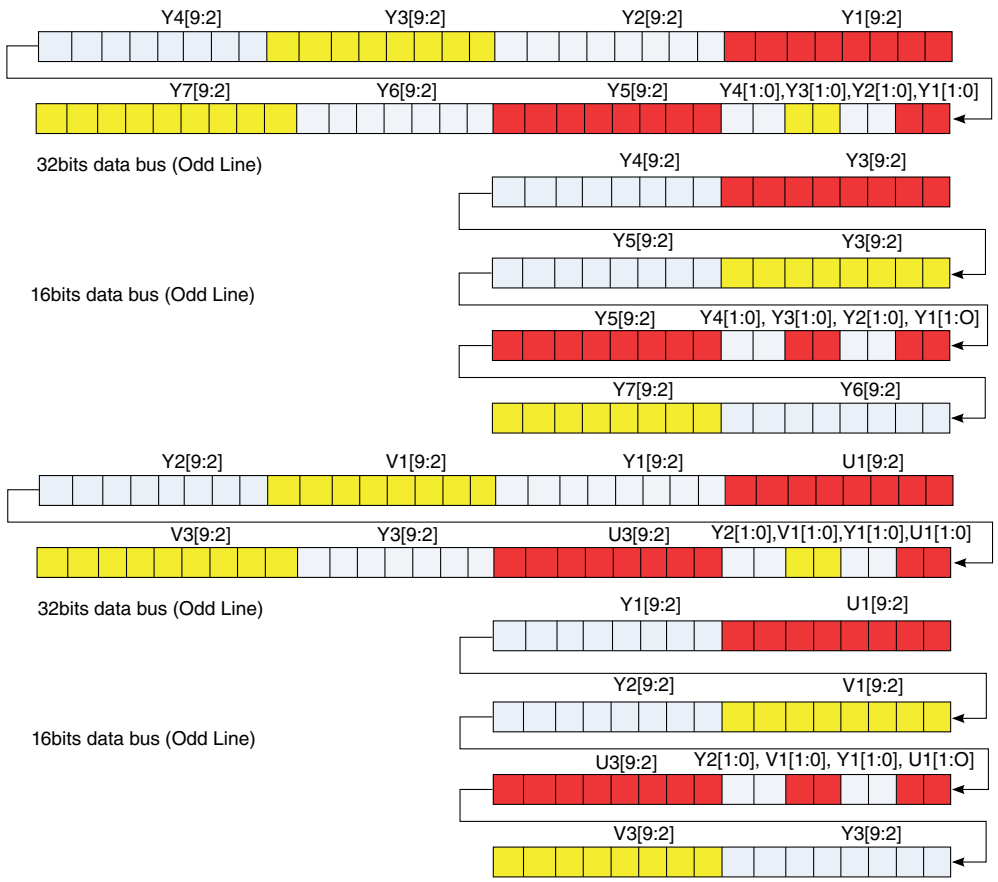


Figure 19-14. YUV420-10 data reception

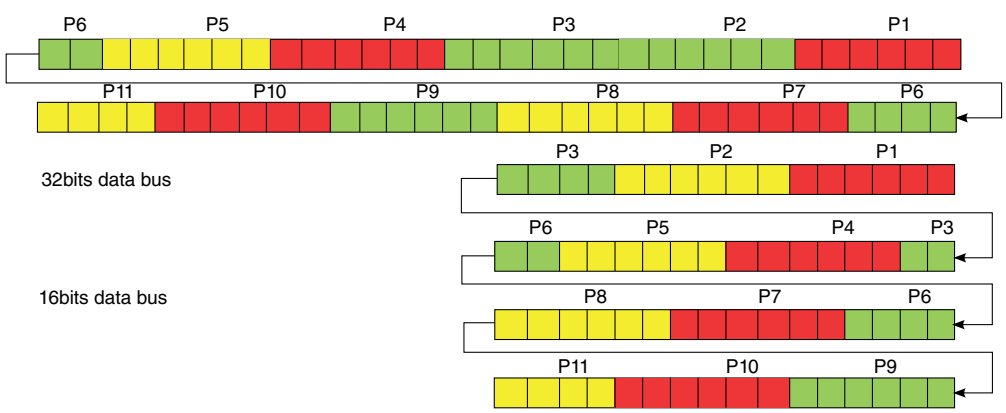


Figure 19-15. RAW-6 data reception

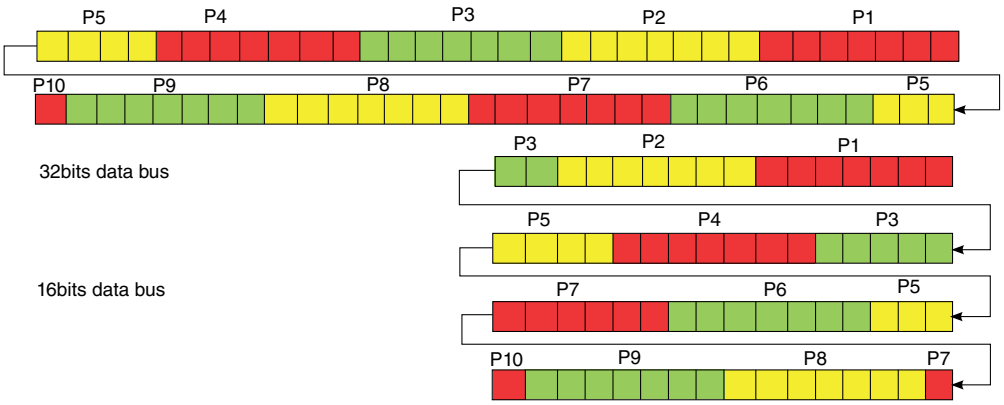


Figure 19-16. RAW-7 data reception

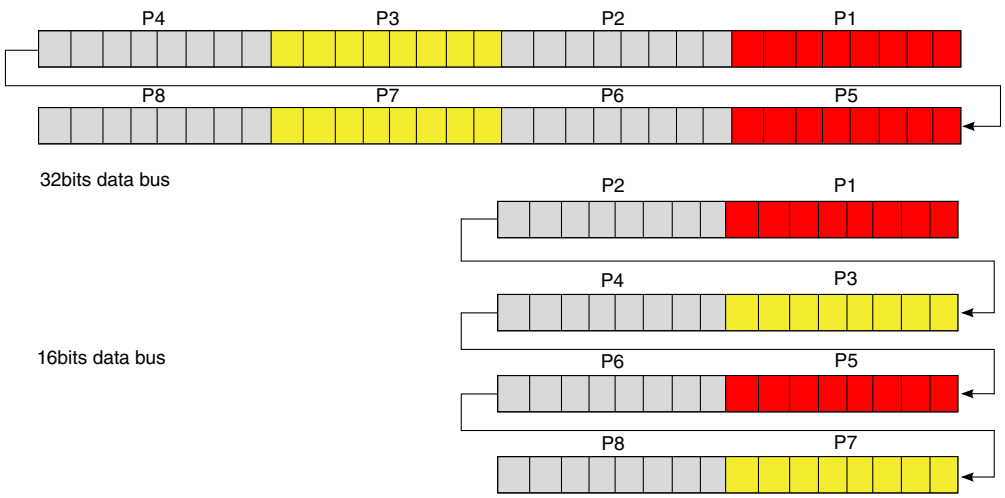


Figure 19-17. RAW-8 data reception

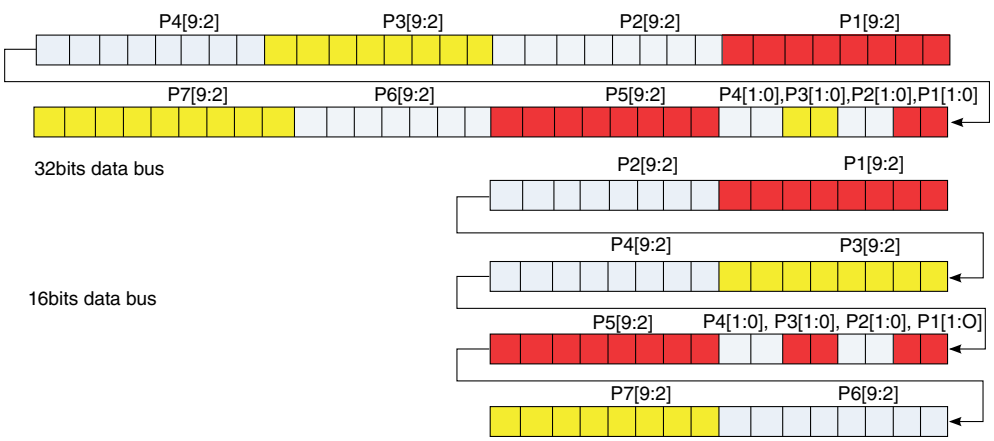


Figure 19-18. RAW-10 data reception

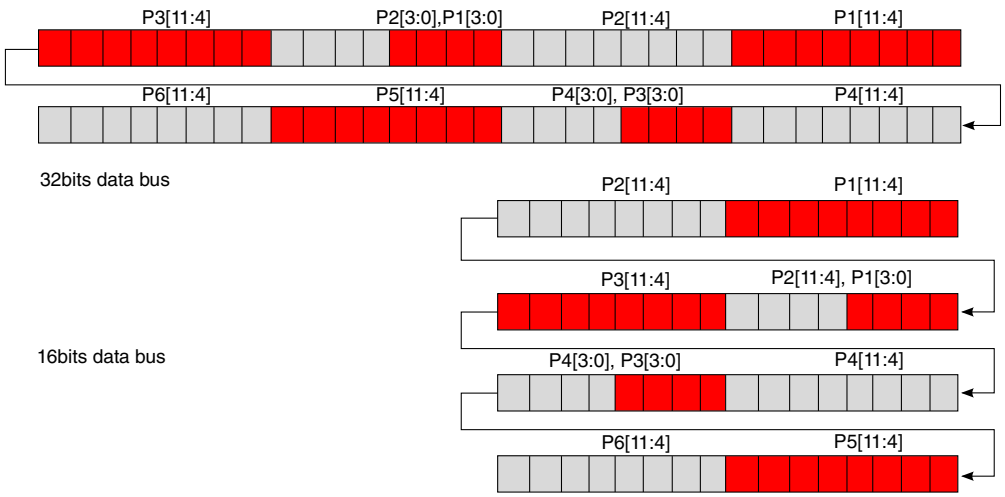


Figure 19-19. RAW-12 data reception

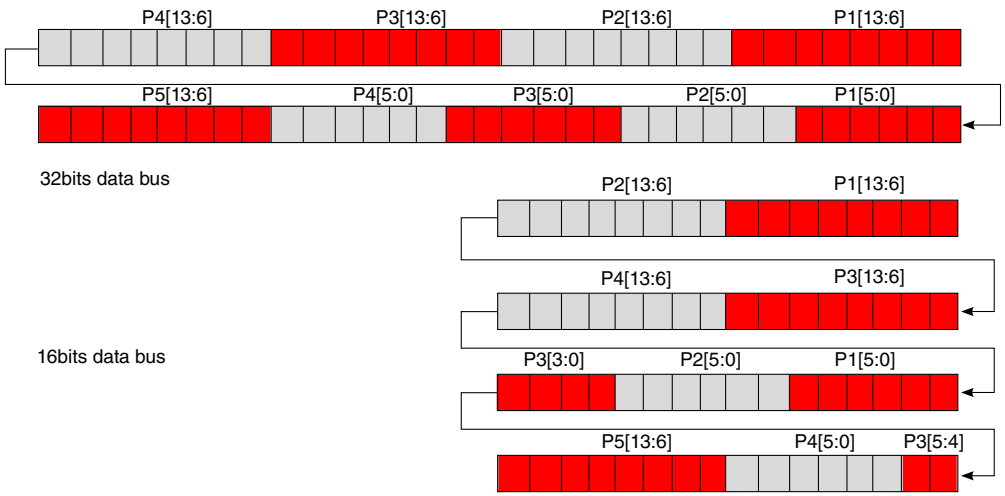


Figure 19-20. RAW-14 data reception

19.5 CSI2IPU Memory Map/Register Definition

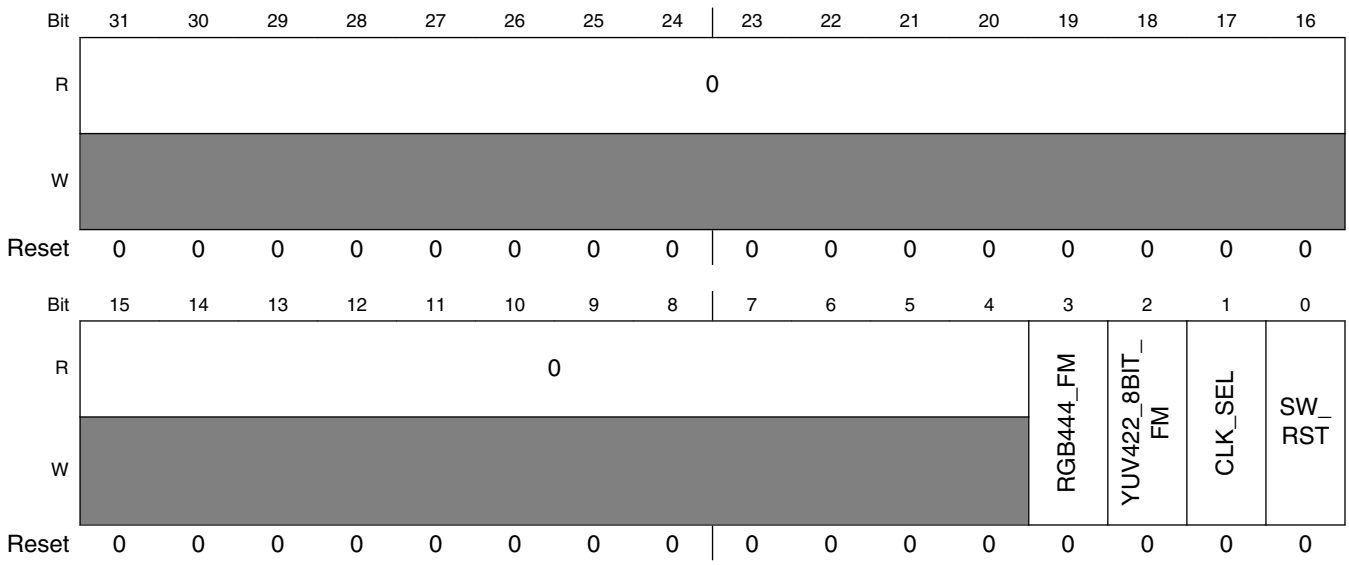
CSI2IPU memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21D_CF00	CSI 2 IPU Gasket Software Reset (CSI2IPU_SW_RST)	32	R/W	0000_0000h	19.5.1/935

19.5.1 CSI 2 IPU Gasket Software Reset (CSI2IPU_SW_RST)

This register describes the IPU interface signals.

Address: 21D_C000h base + F00h offset = 21D_CF00h



CSI2IPU_SW_RST field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value 0.
3 RGB444_FM	rgb444 mode selection 0 {4'h0,r4b4g4} 1 {r4,1'b0,g4,2'b00,b4,1'b0}
2 YUV422_8BIT_FM	YUV422 8-bit mode selection 0 YUYV 1 UYVY
1 CLK_SEL	Clock mode selection 0 Gated Mode 1 Non-Gated Mode
0 SW_RST	Software Reset 0 Software Reset Disable 1 Software Reset Enable

Chapter 20
Display Content Integrity Checker (DCIC)

20.1 Overview

The goal of the DCIC is to verify that a safety-critical information sent to a display is not corrupted.

Such a verification is mandatory for warning icons in the instrument cluster of a car, to comply with the ASIL B (Automotive Safety Integrity Level B) specification. It is also required in other safety-sensitive systems.

Using external muxing DCIC can monitor either one of the IPU display port outputs or feedback signals going from IO pads of Parallel display interface. The figure below shows DCIC integration in system with one IPU port.

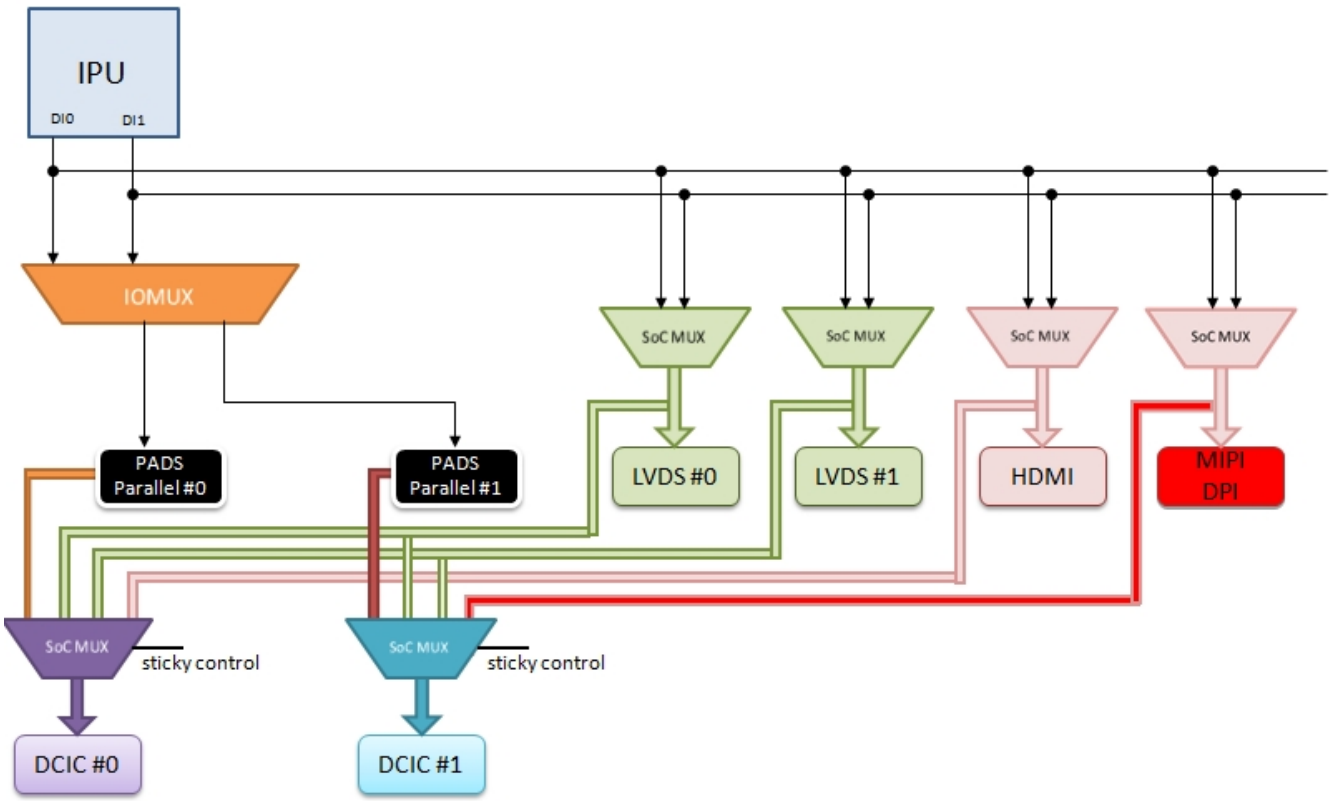


Figure 20-1. DCIC system integration example

20.1.1 Block Diagram

The figure below shows DCIC top level block diagram.

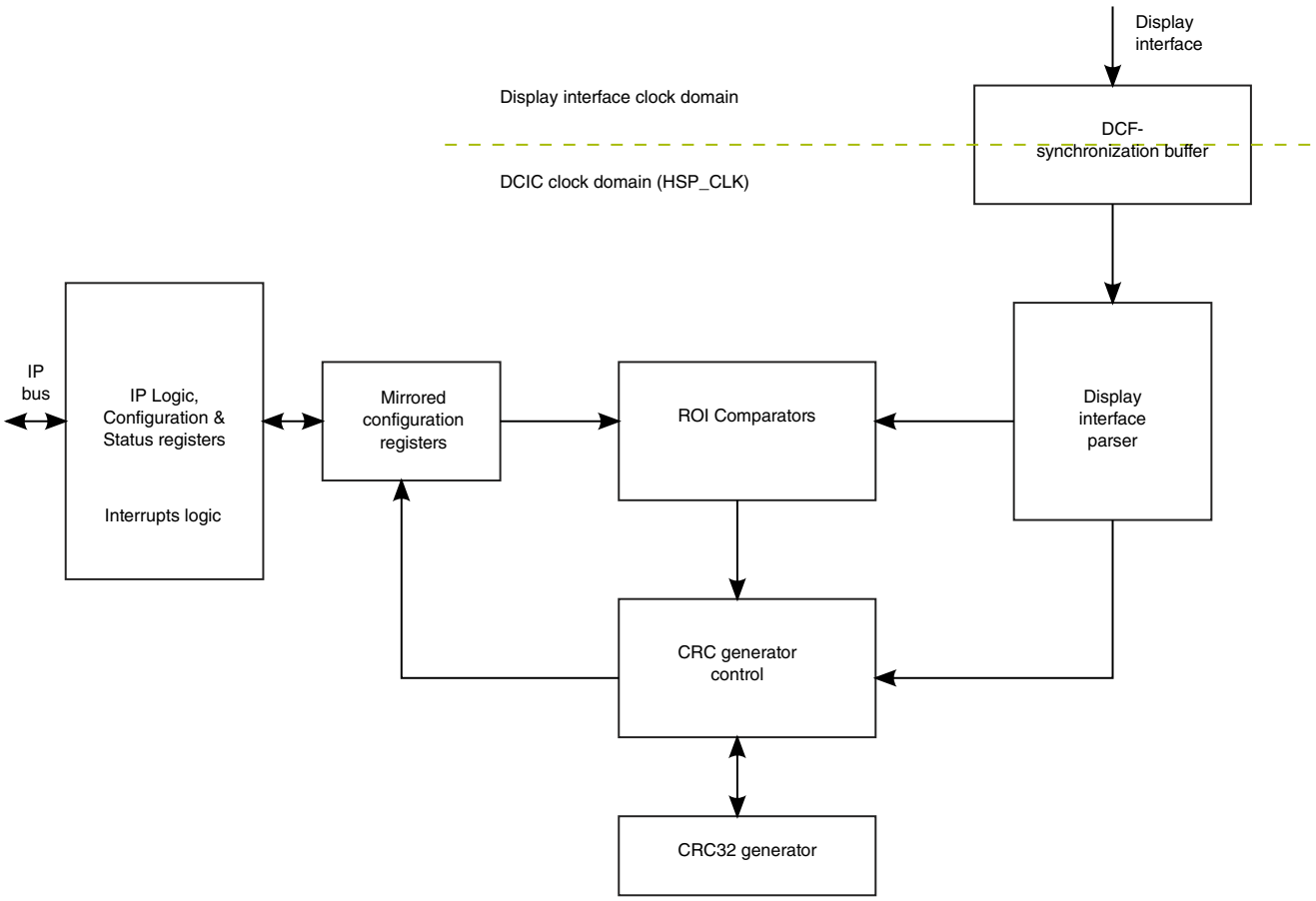


Figure 20-2. DCIC Block Diagram

20.1.2 Features

- Pixel clock up to 266 MHz
- Configurable polarity of Display Interface control signals
- 24-bit pixel data bus
- Up to 16 rectangular ROIs with a configurable location and size
- Independent CRC32 signature calculation for each ROI
- External controller mismatch indication signal

20.2 Functional Description

This section provides a complete functional description of the block.

20.2.1 Generic synchronous parallel display interface

The figure below depicts the LCD interface timing for a generic active matrix color TFT panel. In this figure signals are shown with negative polarity, but both polarity modes are supported by DCIC through configuration of appropriate bits of DCICCR register.

The sequence of events for active matrix interface timing is:

- IPP_DISP_CLK latches data into the panel on its negative edge (when positive polarity is selected). In active mode, IPP_DISP_CLK runs continuously.
- HSYNC causes the panel to start a new line.
- VSYNC causes the panel to start a new frame. It always encompasses at least one HSYNC pulse.
- DRDY acts like an output enable signal. This output enables the data to be shifted onto the display. When disabled, the data is invalid and the trace is off.

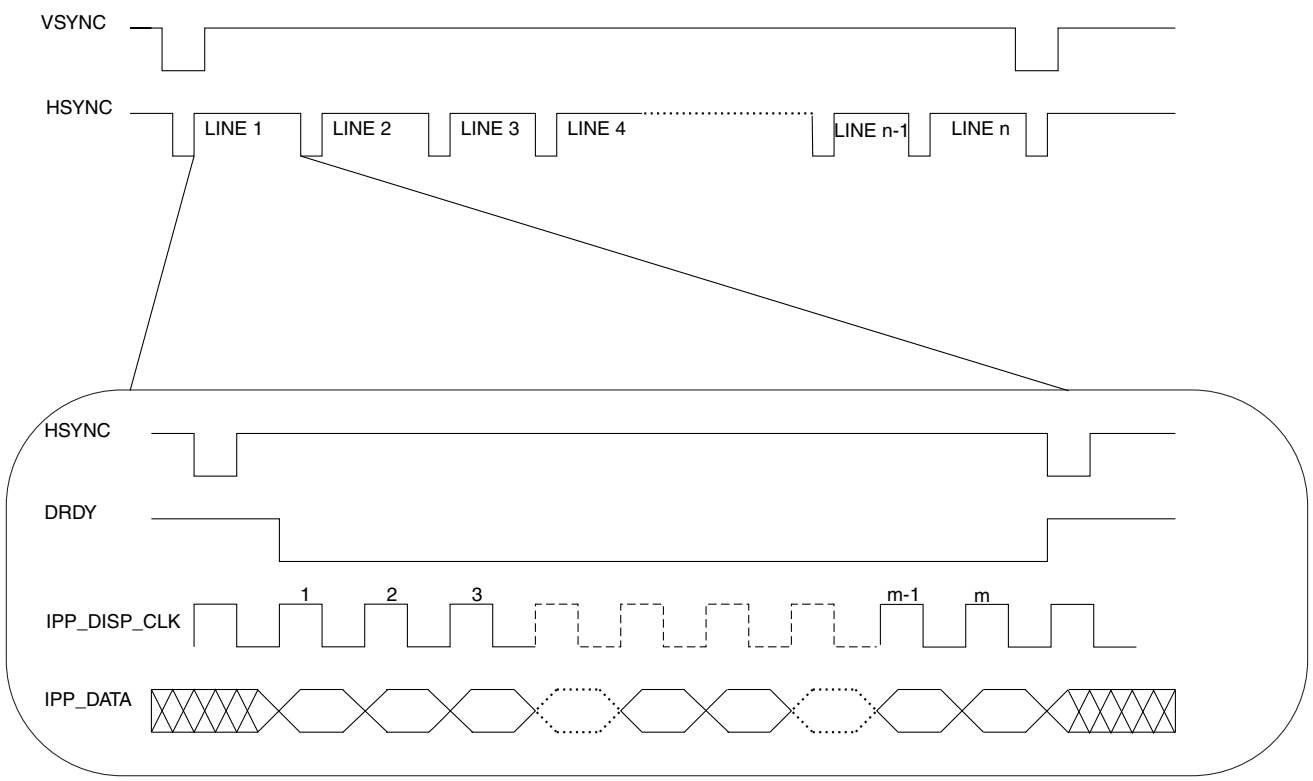


Figure 20-3. Interface Timing Diagram for TFT (Active Matrix) Panels

20.2.2 CRC Polynomial

DCIC uses the CRC32 polynomial to calculate the data signature.

Initial value for CRC calculation is 0x00000000.

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

CRC generator uses XOR-ed combination of input data bits and previous result bits.

20.2.3 Mode of operation

After DCIC is configured and enabled, providing IPU display interface is already enabled, first VSYNC is being watched. VSYNC assertion zeroizes main position counters and initializes CRC storage registers.

Each HSYNC assertion after first line in frame that holds valid pixel data (non blanking) will signal new line, i.e. increment Y-counter and zeroize X-counter. Each valid data cycle (DATA_EN asserted) will increment X-counter. Thus X & Y values used to configure the module should be the coordinates on the actual visual part of the screen without blanking intervals. X & Y position counters are constantly compared to ROIs configuration. When current position matches one of the enabled ROIs, it triggers the enablement of CRC generator, which will use the current pixel data and previous pixel CRC result for this ROI. The result of the new calculation stored separately for each ROI. Next VSYNC, apart of signalling new frame start, will cause sampling calculated CRC values into configuration registers domain. These values then will be compared to expected signatures and appropriate status bits / interrupts will be asserted.

20.2.4 Interrupts

There are two maskable interrupts:

- Functional - Asserted when match results ready.
- Error - Asserted when there is a signature mismatch.

Both interrupts are generated immediately after completing the signature match check. Software should clean the interrupts by writing "1" to appropriate status bits (FI_STAT, ROI_MATCH_STAT). EI_STAT bit is a result of OR between all ROI_MATCH_STAT bits, hence it will be cleared automatically when these bits are clear.

Interrupt masks can be set/reset only while FREEZE_MASK bit isn't set. This bit can not be set back to zero.

20.2.5 Software reset

Disabling and enabling the module operation (IC_EN bit, DCIC Control Register (DCIC_DCICC)) will reset all the logic apart of configuration registers domain. Status bits of DCICS (DCIC Status Register (DCIC_DCICS)) register will also be cleared.

20.2.6 Clock domains

Asynchronous FIFO is being used to sample raw display interface signals and transfer them into DCIC fast clock domain (hsp_clk).

Display interface clock is always slower than DCIC fast clock.

IP registers are driven by the same fast clock and will be synchronized to system IP Bus clock by external IP_SYNC module.

20.2.7 External controller mismatch indication signal

Besides of interrupts to SoC core, DCIC provides an additional mismatch indication for external controller. The signal, if enabled, continuously oscillating at a rate which is an integer division of the main clock (hsp clock).

- When the status bit of the mismatch interrupt is set - i.e. from mismatch detection until the CPU clears the bit: division x16
- Otherwise: division x4

The indication signal is idle while integrity check is disabled.

20.2.8 Power saving

Disabling the module by clearing IC_EN bit will stop all module activities, including sampling of input signals into the FIFO.

20.2.9 System Considerations

- The DCIC always assumes a 24-bit pixel. For proper functionality with lower color depth, one must ensure that:
  - When the CPU calculates the reference signature, it applies the same mapping of a pixel to a 24-bit field as that performed by the IPU.

    - The display is connected to the appropriate pins.
    - The simplest mapping would be
        - Map the pixel to the data bus in the same way as for 24 bpp
        - Set the values at the extra LSBs to zero.
- Parameter updates
    - The reference signature can be freely updated from frame to frame, in coordination with the changing content (since it is used only once per frame, just before the interrupt)
    - Each ROI can be freely and independently enabled/disabled between frames (since the enable bit is double buffered)
    - The size and location of a ROI can be modified while it is disabled.
    - Display interface signals polarity can be changed only when the module is disabled.
- ROI (regions of interest) don't overlap, i.e. each pixel belongs to single ROI at most.
- DCIC operation is intended for monitoring relatively static portions of graphic interface, otherwise it will be complicated software task to keep up with frame synchronization and update expected signature for each frame.
- There should be no processing done at IPU on monitored ROIs, which can't be taken in consideration by software when calculating the expected signature.

20.3 DCIC Memory Map/Register Definition

Important: All write accesses have to be full word (32-bit) accesses. No error/abort will be responded in case of different access size, but no data will be written. Similarly, there will be no error response in case of access to undefined memory space.

Read access will return 32-bit data, which may be truncated on system level in case it wasn't full word acces.

DCIC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_4000	DCIC Control Register (DCIC1_DCICC)	32	R/W	0000_0070h	20.3.1/944
20E_4004	DCIC Interrupt Control Register (DCIC1_DCICIC)	32	R/W	0000_0003h	20.3.2/945
20E_4008	DCIC Status Register (DCIC1_DCICS)	32	w1c	0000_0000h	20.3.3/946
20E_4010	DCIC ROI Config Register m (DCIC1_DCICRC)	32	R/W	0000_0000h	20.3.4/947
20E_4014	DCIC ROI Size Register m (DCIC1_DCICRS)	32	R/W	0000_0000h	20.3.5/948

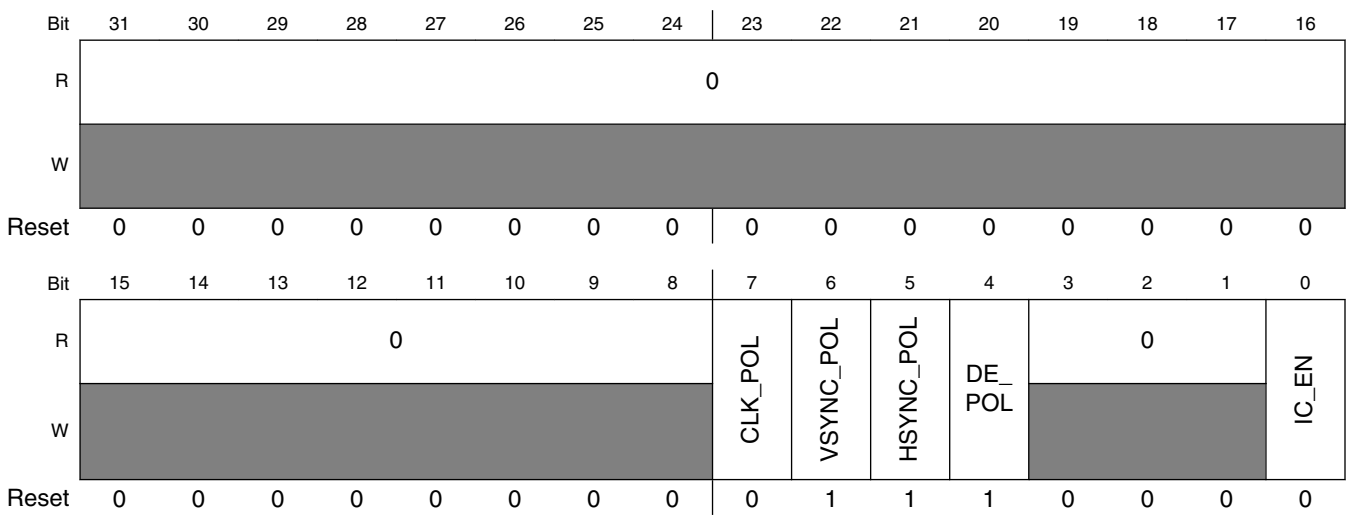
Table continues on the next page...

DCIC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_4018	DCIC ROI Reference Signature Register m (DCIC1_DCICRRS)	32	R/W	0000_0000h	20.3.6/949
20E_401C	DCIC ROI Calculated Signature m (DCIC1_DCICRCS)	32	R	0000_0000h	20.3.7/949
20E_8000	DCIC Control Register (DCIC2_DCICC)	32	R/W	0000_0070h	20.3.1/944
20E_8004	DCIC Interrupt Control Register (DCIC2_DCICIC)	32	R/W	0000_0003h	20.3.2/945
20E_8008	DCIC Status Register (DCIC2_DCICS)	32	w1c	0000_0000h	20.3.3/946
20E_8010	DCIC ROI Config Register m (DCIC2_DCICRC)	32	R/W	0000_0000h	20.3.4/947
20E_8014	DCIC ROI Size Register m (DCIC2_DCICRS)	32	R/W	0000_0000h	20.3.5/948
20E_8018	DCIC ROI Reference Signature Register m (DCIC2_DCICRRS)	32	R/W	0000_0000h	20.3.6/949
20E_801C	DCIC ROI Calculated Signature m (DCIC2_DCICRCS)	32	R	0000_0000h	20.3.7/949

20.3.1 DCIC Control Register (DCICx_DCICC)

Address: Base address + 0h offset



DCICx_DCICC field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.
7 CLK_POL	DISP_CLK signal polarity. 0 Not inverted (default). 1 Inverted.
6 VSYNC_POL	VSYNC_IN signal polarity.

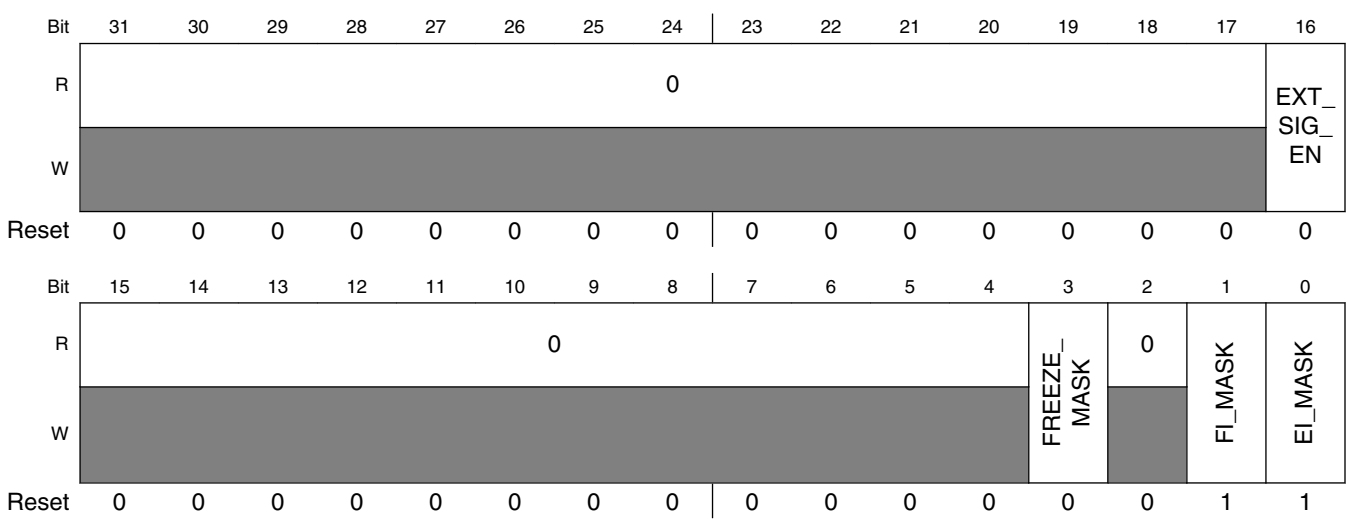
Table continues on the next page...

DCICx_DCICC field descriptions (continued)

Field	Description
	0 Active High. 1 Active Low (default).
5 HSYNC_POL	HSYNC_IN signal polarity. 0 Active High. 1 Active Low (default).
4 DE_POL	DATA_EN_IN signal polarity. 0 Active High. 1 Active Low (default).
3–1 Reserved	This read-only field is reserved and always has the value 0.
0 IC_EN	Integrity Check enable. Main enable switch. 0 Disabled 1 Enabled

20.3.2 DCIC Interrupt Control Register (DCICx_DCICIC)

Address: Base address + 4h offset



DCICx_DCICIC field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 EXT_SIG_EN	External controller mismatch indication signal. 0 Disabled (default) 1 Enabled

Table continues on the next page...

DCICx_DCICIC field descriptions (continued)

Field	Description
15–4 Reserved	This read-only field is reserved and always has the value 0.
3 FREEZE_MASK	Disable change of interrupt masks. "Sticky" bit which can be set once and cleared by reset only. 0 Masks change allowed (default) 1 Masks are frozen
2 Reserved	This read-only field is reserved and always has the value 0.
1 FI_MASK	Functional Interrupt mask. Can be changed only while FREEZE_MASK = 0. 0 Mask disabled - Interrupt assertion enabled 1 Mask enabled - Interrupt assertion disabled (default)
0 EI_MASK	Error Interrupt mask. Can be changed only while FREEZE_MASK = 0. 0 Mask disabled - Interrupt assertion enabled 1 Mask enabled - Interrupt assertion disabled (default)

20.3.3 DCIC Status Register (DCICx_DCICS)

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														FI_STAT	EI_STAT
W															w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

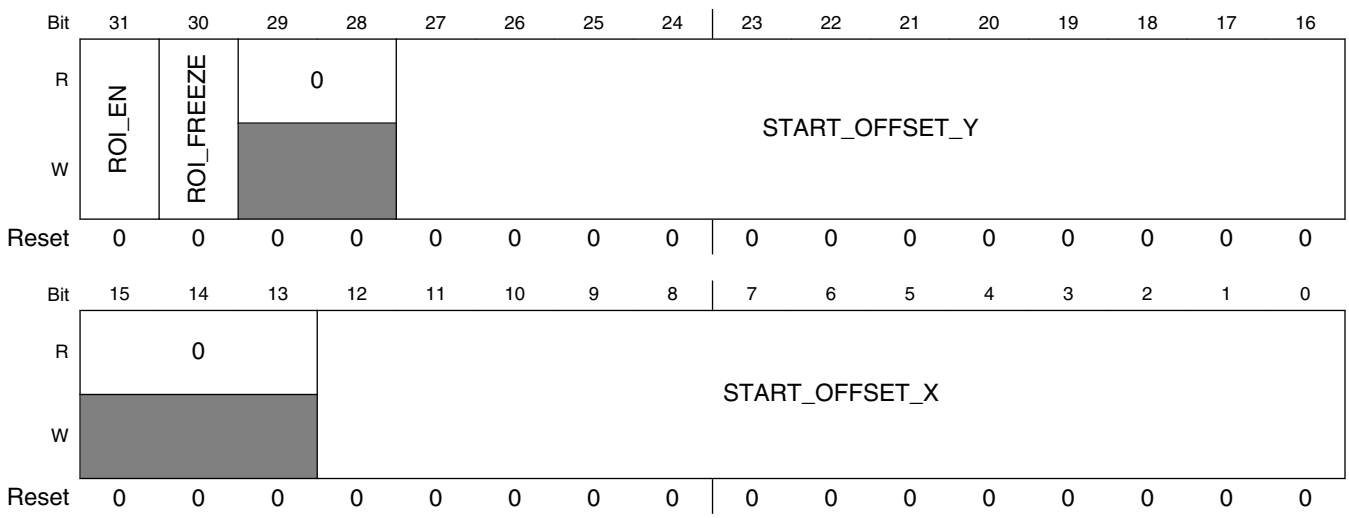
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ROI_MATCH_STAT															
W	w1c															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCIC*x*_DCICS field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value 0.
17 FI_STAT	Functional Interrupt status. Write "1" to clear. 0 No pending Interrupt 1 Pending Interrupt
16 EI_STAT	Error Interrupt status. Result of "OR" operation on ROI_MATCH_STAT[15:0] bits. Cleared when these bits are clear. 0 No pending Interrupt 1 Pending Interrupt
ROI_MATCH_ STAT	Each set bit of this field indicates there was a mismatch at appropriate ROIs signature during the last frame. Valid only for active ROIs. Write "1" to clear. 0 ROI calculated CRC matches expected signature 1 Mismatch at ROI calculated CRC

20.3.4 DCIC ROI Config Register m (DCIC*x*_DCICRC)

Address: Base address + 10h offset



DCIC*x*_DCICRC field descriptions

Field	Description
31 ROI_EN	ROI #m tracking enable 0 Disabled 1 Enabled

Table continues on the next page...

DCICx_DCICRC field descriptions (continued)

Field	Description
30 ROI_FREEZE	When set, the only parameter of ROI #m that can be changed is reference signature. "Sticky" bit - can be set once and cleared by reset only. 0 ROI configuration can be changed 1 ROI configuration is frozen
29–28 Reserved	This read-only field is reserved and always has the value 0.
27–16 START_ OFFSET_Y	Row number of ROIs upper-left corner (Y coordinate) Range: 0 to 2^12-1
15–13 Reserved	This read-only field is reserved and always has the value 0.
START_ OFFSET_X	Column number of ROIs upper-left corner (X coordinate) Range: 0 to 2^13-1

20.3.5 DCIC ROI Size Register m (DCICx_DCICRS)

Address: Base address + 14h offset

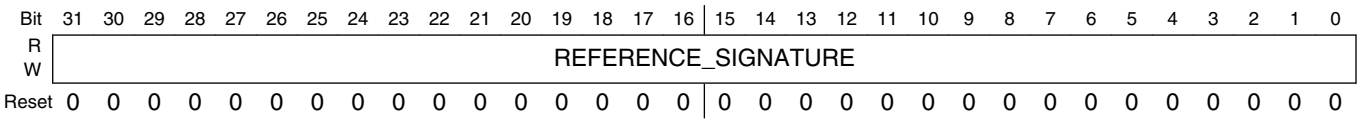
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				END_OFFSET_Y												0			END_OFFSET_X											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCICx_DCICRS field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–16 END_OFFSET_Y	Row number of ROIs lower-right corner (Y coordinate) Range: 1 to 2^12-1
15–13 Reserved	This read-only field is reserved and always has the value 0.
END_OFFSET_X	Column number of ROIs lower-right corner (X coordinate) Range: 1 to 2^13-1

20.3.6 DCIC ROI Reference Signature Register m (DCICx_DCICRRS)

Address: Base address + 18h offset

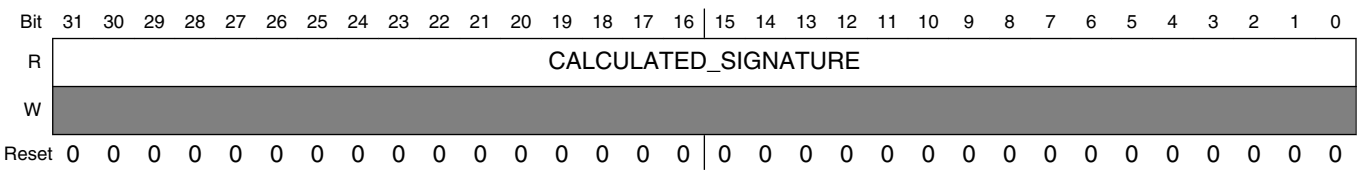


DCICx_DCICRRS field descriptions

Field	Description
REFERENCE_ SIGNATURE	32-bit expected signature (CRC calculation result) for ROI #m

20.3.7 DCIC ROI Calculated Signature m (DCICx_DCICRCS)

Address: Base address + 1Ch offset



DCICx_DCICRCS field descriptions

Field	Description
CALCULATED_ SIGNATURE	32-bit actual signature (CRC calculation result) for ROI #m during the last frame. Updated automatically at the beginning of a next frame.

Chapter 21
Enhanced Configurable SPI (ECSPI)

21.1 Overview

The Enhanced Configurable Serial Peripheral Interface (ECSPI) is a full-duplex, synchronous, four-wire serial communication block.

The ECSPI contains a 64 x 32 receive buffer (RXFIFO) and a 64 x 32 transmit buffer (TXFIFO). With data FIFOs, the ECSPI allows rapid data communication with fewer software interrupts. The figure below shows a block diagram of the ECSPI.

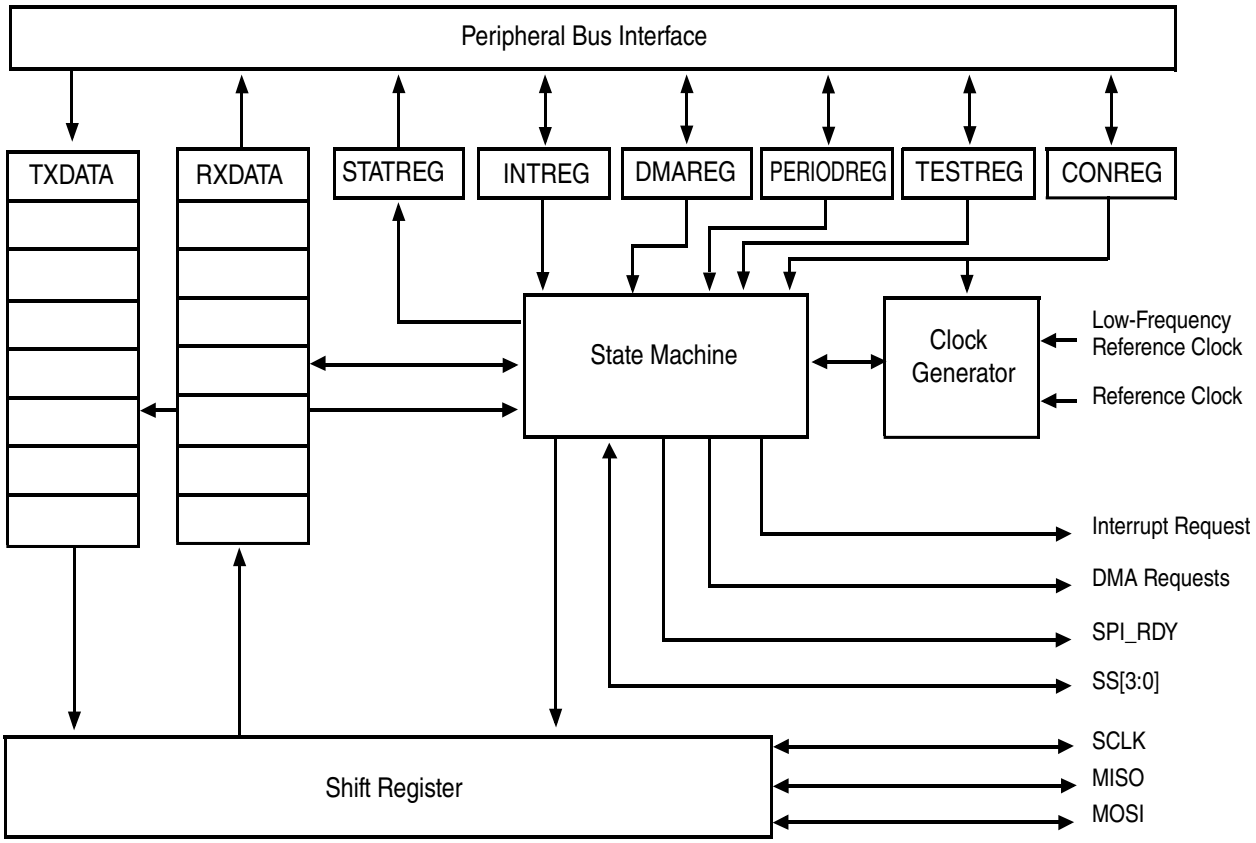


Figure 21-1. ECSPI Block Diagram

21.1.1 Features

Key features of the ECSPI include:

- Full-duplex synchronous serial interface
- Master/Slave configurable
- Four Chip Select (SS) signals to support multiple peripherals
- Transfer continuation function allows unlimited length data transfers
- 32-bit wide by 64-entry FIFO for both transmit and receive data
- Polarity and phase of the Chip Select (SS) and SPI Clock (SCLK) are configurable
- Direct Memory Access (DMA) support
- Refer to the product data sheet for the maximum operating frequency

21.1.2 Modes and Operations

The ECSPI supports the modes described in the indicated sections:

- Master Mode
- Slave Mode
- Low Power Modes

As described in Operations, the ECSPI supports the operations described in the indicated sections:

- Typical Master Mode
  - Master Mode with SPI_RDY
  - Master Mode with Wait States
  - Master Mode with SS_CTL[3:0] Control
  - Master Mode with Phase Control
- Typical Slave Mode

21.2 External Signals

The following table describes the external signals of ECSPI:

Table 21-1. ECSPI1 External Signals

Signal	Description	Pad	Mode	Direction
ECSPI1_MISO (MISO)	Master data in; slave data out	CSI0_DAT6	ALT2	IO
		DISP0_DAT22	ALT2	
		EIM_D17	ALT1	
		KEY_COL1	ALT0	
ECSPI1_MOSI (MOSI)	Master data out; slave data in	CSI0_DAT5	ALT2	IO
		DISP0_DAT21	ALT2	
		EIM_D18	ALT1	
		KEY_ROW0	ALT0	
ECSPI1_RDY (RDY)	SPI data ready signal	GPIO_19	ALT4	I
ECSPI1_SCLK (SCLK)	SPI clock signal	CSI0_DAT4	ALT2	IO
		DISP0_DAT20	ALT2	
		EIM_D16	ALT1	
		KEY_COL0	ALT0	
ECSPI1_SS0 (SS0)	Chip select signal	CSI0_DAT7	ALT2	IO
		DISP0_DAT23	ALT2	
		EIM_EB2	ALT1	
		KEY_ROW1	ALT0	

Table continues on the next page...

Table 21-1. ECSPI1 External Signals (continued)

Signal	Description	Pad	Mode	Direction
ECSPI1_SS1 (SS1)	Chip select signal	DISP0_DAT15	ALT2	IO
		EIM_D19	ALT1	
		KEY_COL2	ALT0	
ECSPI1_SS2 (SS2)	Chip select signal	EIM_D24	ALT3	IO
		KEY_ROW2	ALT0	
ECSPI1_SS3 (SS3)	Chip select signal	EIM_D25	ALT3	IO
		KEY_COL3	ALT0	

Table 21-2. ECSPI2 External Signals

Signal	Description	Pad	Mode	Direction
ECSPI2_MISO (MISO)	Master data in; slave data out	CSI0_DAT10	ALT2	IO
		DISP0_DAT17	ALT2	
		EIM_OE	ALT2	
ECSPI2_MOSI (MOSI)	Master data out; slave data in	CSI0_DAT9	ALT2	IO
		DISP0_DAT16	ALT2	
		EIM_CS1	ALT2	
ECSPI2_RDY (RDY)	SPI data ready signal	EIM_A25	ALT2	I
ECSPI2_SCLK (SCLK)	SPI clock signal	CSI0_DAT8	ALT2	IO
		DISP0_DAT19	ALT2	
		EIM_CS0	ALT2	
ECSPI2_SS0 (SS0)	Chip select signal	CSI0_DAT11	ALT2	IO
		DISP0_DAT18	ALT2	
		EIM_RW	ALT2	
ECSPI2_SS1 (SS1)	Chip select signal	DISP0_DAT15	ALT3	IO
		EIM_LBA	ALT2	
ECSPI2_SS2 (SS2)	Chip select signal	EIM_D24	ALT4	IO
ECSPI2_SS3 (SS3)	Chip select signal	EIM_D25	ALT4	IO

Table 21-3. ECSPI3 External Signals

Signal	Description	Pad	Mode	Direction
ECSPI3_MISO (MISO)	Master data in; slave data out	DISP0_DAT2	ALT2	IO
ECSPI3_MOSI (MOSI)	Master data out; slave data in	DISP0_DAT1	ALT2	IO
ECSPI3_RDY (RDY)	SPI data ready signal	DISP0_DAT7	ALT2	I
ECSPI3_SCLK (SCLK)	SPI clock signal	DISP0_DAT0	ALT2	IO
ECSPI3_SS0 (SS0)	Chip select signal	DISP0_DAT3	ALT2	IO
ECSPI3_SS1 (SS1)	Chip select signal	DISP0_DAT4	ALT2	IO

Table continues on the next page...

Table 21-3. ECSPI3 External Signals (continued)

Signal	Description	Pad	Mode	Direction
ECSPI3_SS2 (SS2)	Chip select signal	DISP0_DAT5	ALT2	IO
ECSPI3_SS3 (SS3)	Chip select signal	DISP0_DAT6	ALT2	IO

Table 21-4. ECSPI4 External Signals

Signal	Description	Pad	Mode	Direction
ECSPI4_MISO (MISO)	Master data in; slave data out	EIM_D22	ALT1	IO
ECSPI4_MOSI (MOSI)	Master data out; slave data in	EIM_D28	ALT2	IO
ECSPI4_RDY (RDY)	SPI data ready signal	EIM_EB3	ALT1	I
ECSPI4_SCLK (SCLK)	SPI clock signal	EIM_D21	ALT1	IO
ECSPI4_SS0 (SS0)	Chip select signal	EIM_D20	ALT1	IO
		EIM_D29	ALT2	
ECSPI4_SS1 (SS1)	Chip select signal	EIM_A25	ALT1	IO
ECSPI4_SS2 (SS2)	Chip select signal	EIM_D24	ALT1	IO
ECSPI4_SS3 (SS3)	Chip select signal	EIM_D25	ALT1	IO

Figure 21-2 shows the ECSPI in master mode connected to four external devices in a one-way communication link.

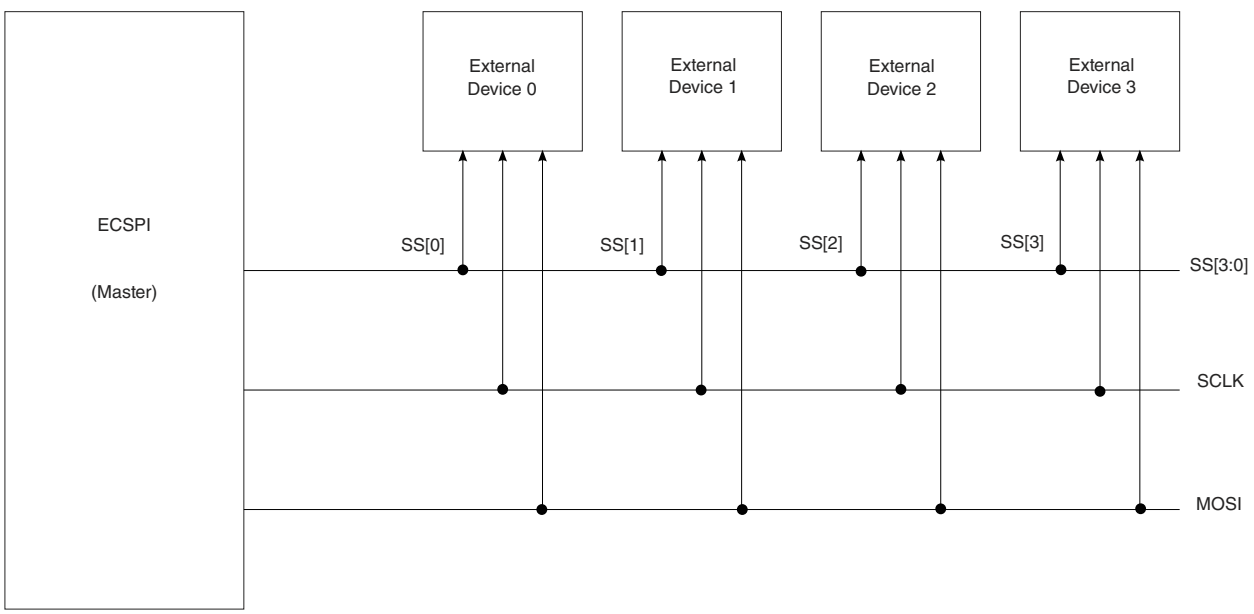


Figure 21-2. Example Connection Diagram

21.3 Clocks

The following table describes the clock sources for eCSPI. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 21-5. eCSPI Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_32k	ckil_sync_clk_root	Low-frequency reference clock (32kHz)
ipg_clk_per	ecspi_clk_root	eCSPI module clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

21.4 Functional Description

This section provides a complete functional description of the ECSPI. The figure found here shows the relationship of SCLK and data lines while ECSPI has been configured with different POL and PHA settings.

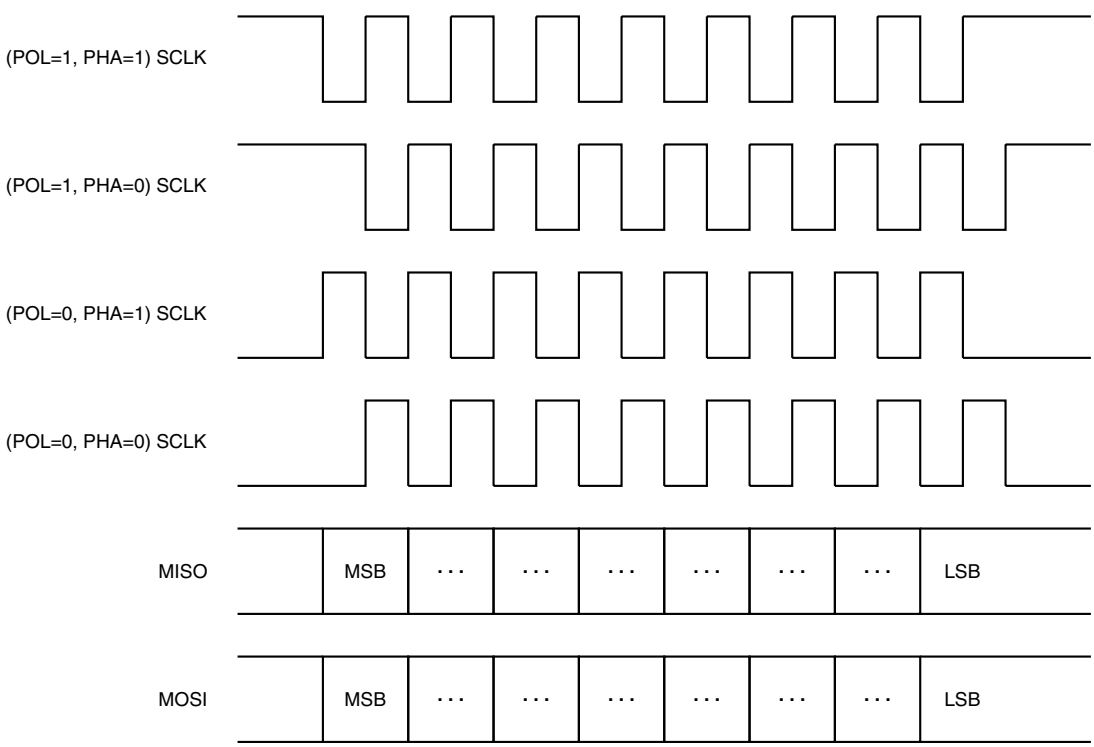


Figure 21-3. ECSPI SCLK, MISO, and MOSI Relationship

21.4.1 Master Mode

When the ECSPI is configured as a master, it uses a serial link to transfer data between the ECSPI and an external slave device.

One of the Chip Select (SS) signals and the clock signal (SCLK) are used to transfer data between two devices. If the external device is a transmit-only device, the ECSPI master's output port can be ignored and used for other purposes. In order to use the internal TXFIFO and RXFIFO, two auxiliary output signals, Chip Select (SS) and SPI_RDY, are used for data transfer rate control. Software can also configure the sample period control register to a fixed data transfer rate.

21.4.2 Slave Mode

When the ECSPI is configured as a slave, software can configure the ECSPI Control register to match the external SPI master's timing.

In this configuration, Chip Select (SS) becomes an input signal, and is used to control data transfers through the Shift register, as well as to load/store the data FIFO.

Slave mode only supports the case when ECSPIx_CONFIGREG[SS_CTL] is cleared. The accurate burst length should always be specified using the BURST_LENGTH parameter. ECSPIx_CONFIGREG[SS_CTL] set to 1 is not supported in slave mode.

21.4.3 Low Power Modes

The ECSPI does not operate under low power mode.

It holds its operation when its clock is gated off in master mode. In slave mode, the ECSPI does not respond when its clock is gated off.

21.4.4 Operations

The information found here describes the ECSPI's operations.

21.4.4.1 Typical Master Mode

The ECSPI master uses the Chip Select (SS) signal to enable an external SPI device, and uses the SCLK signal to transfer data in and out of the Shift register.

The SPI_RDY enables fast data communication with fewer software interrupts. By programming the ECSPI_PERIODREG register accordingly, the ECSPI can be used for a fixed data transfer rate.

When the ECSPI is in Master mode the SS, SCLK, and MOSI are output signals, and the MISO signal is an input.

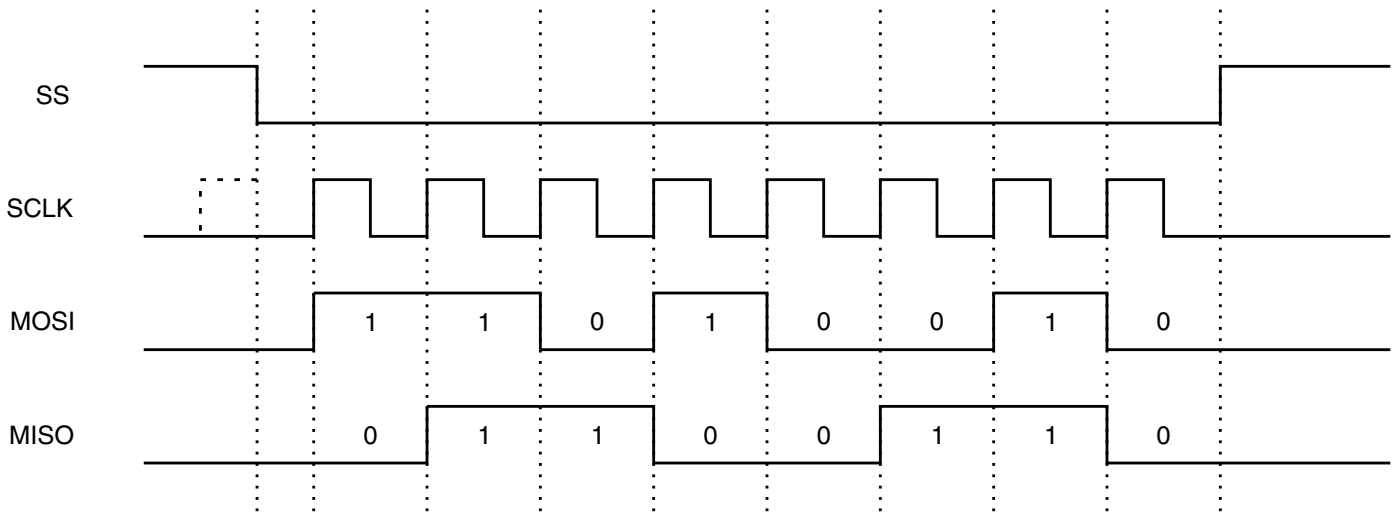


Figure 21-4. Typical SPI Burst (8-bit Transfer)

In the above figure, the Chip Select (SS) signal enables the selected external SPI device, and the SCLK synchronizes the data transfer. The MOSI and MISO signals change on rising edge of SCLK and the MISO signal is latched on the falling edge of the SCLK. The figure above shows a data of 0xD2 is shifted out, and a data of 0x66 is shifted in.

21.4.4.1.1 Master Mode with SPI_RDY

By default, the ECSPI does not use the SPI_RDY signal in master mode (MODE =1).

A SPI burst begins when the following events happen:

- The ECSPI is enabled, TXFIFO has data in it, and ECSPI_CONREG[XCH] bit or the ECSPI_CONREG[SMC] bit is set.
- When the SPI Data Ready Control (ECSPI_CONREG[DRCTL]) bits contains either 01 or 10, the SPI_RDY signal controls when a SPI burst starts.

A SPI burst is defined as a bus transaction that starts when the slave select is asserted and ends when the slave select is negated. The Chip Select (SS) signal will remain asserted until all the bits in a SPI burst are shifted out.

If ECSPI_CONREG[DRCTL] is set to 01, the SPI burst can be triggered only if a falling edge of the SPI_RDY signal has been detected.

The following figure shows the relationship between a SPI burst and the falling edge of SPI_RDY signal.

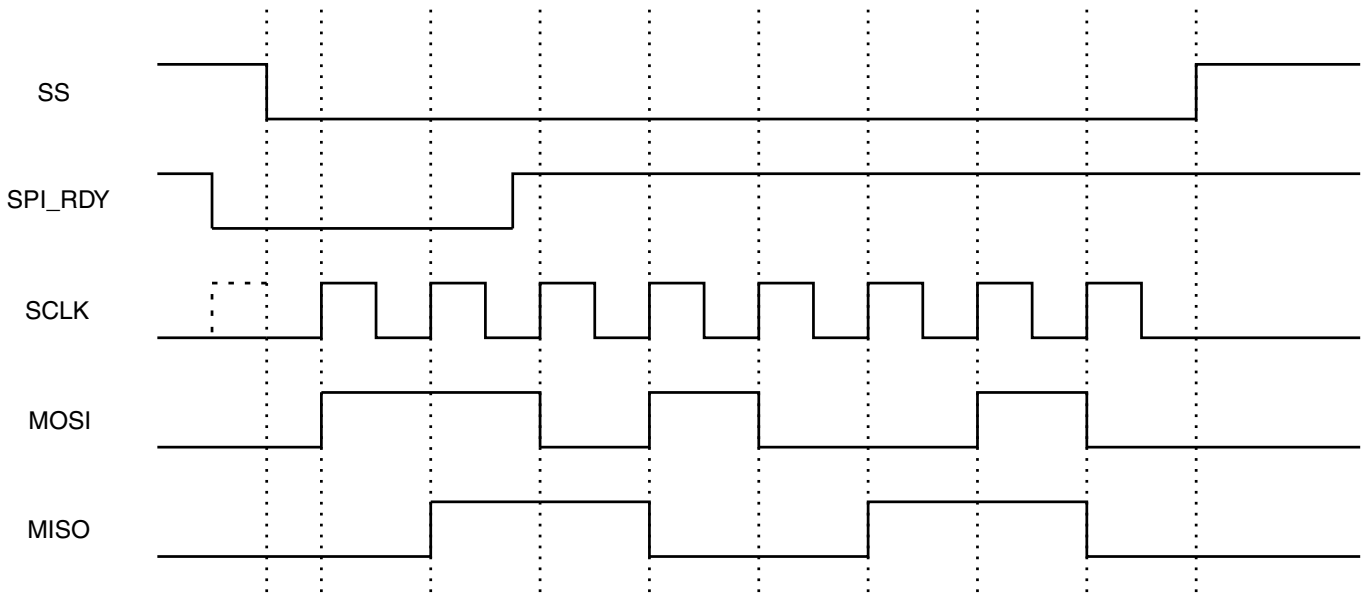


Figure 21-5. Relationship Between a SPI Burst and SPI_RDY: Falling-Edge Triggered

A SPI burst does not start until the falling edge of the SPI_RDY signal is detected. The next SPI burst starts when the next SPI_RDY falling edge is detected, after the last burst has finished.

If SPI Data Ready Control (ECSPI_CONREG[DRCTL]) is set to 10, the SPI burst can be triggered only if the SPI_RDY signal is low.

The following figure shows the relationship between a SPI burst and the SPI_RDY signal. The SPI burst does not begin until the SPI_RDY signal goes low. The ECSPI will keep transmitting SPI burst if the SPI_RDY signal remains low.

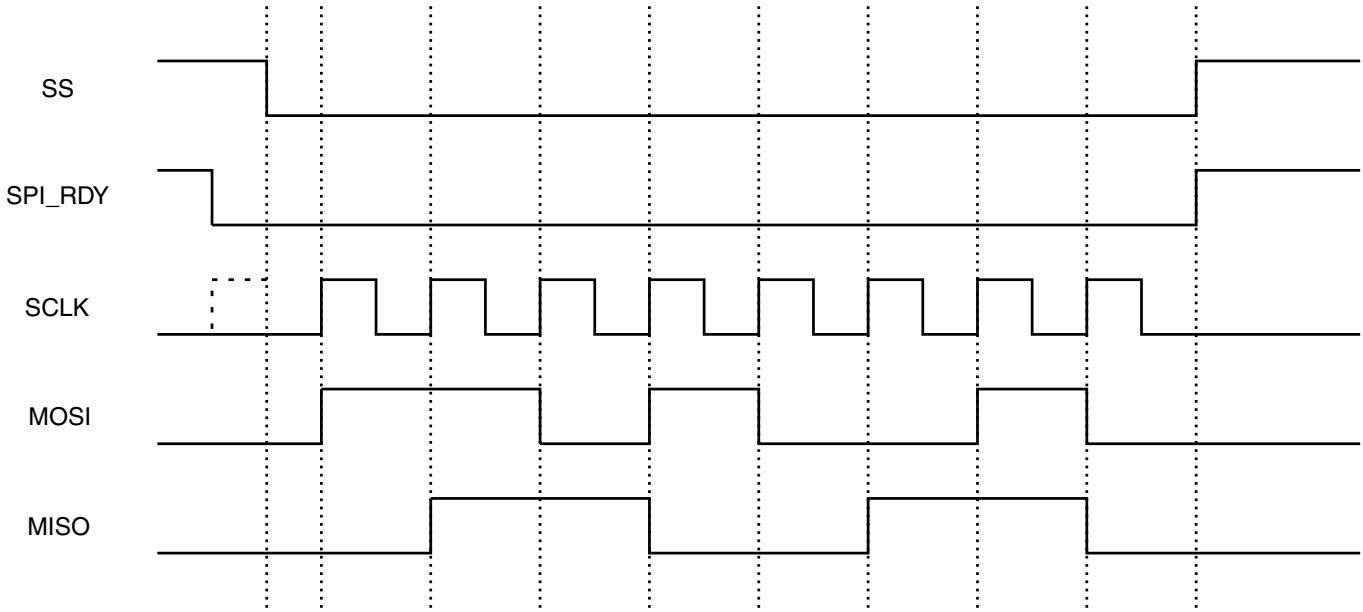


Figure 21-6. Relationship Between a SPI Burst and SPI_RDY: Low-Level Triggered

21.4.4.1.2 Master Mode with Wait States

Wait states can be inserted between SPI bursts. This provides a way for software to slow down the SPI burst to meet the timing requirements of a slower SPI device.

The following figure shows wait states inserted between SPI bursts.

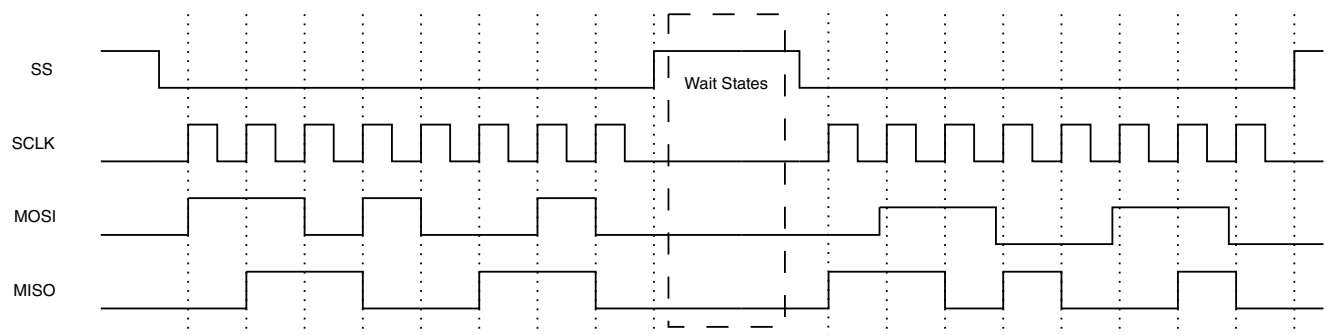


Figure 21-7. SPI Bursts with Wait States

In this case, the number of wait states is controlled by ECSPI_PERIODREG[SAMPLE PERIOD] and the wait states' clock source is selected by ECSPI_PERIODREG[CSRC].

21.4.4.1.3 Master Mode with SS_CTL[3:0] Control

The SPI SS Control (SS_CTL[3:0]) controls whether the current operation is single burst or multiple bursts.

When the SPI SS Wave Form Select (SS_CTL[3:0]) is set, the current operation is multiple bursts transfer. When the SPI SS Wave Form Select (SS_CTL[3:0]) bit is cleared, the current operation is single burst transfer. A SPI burst can contains multiple words as defined in the BURST LENGTH field of the ECSPI_CONREG register.

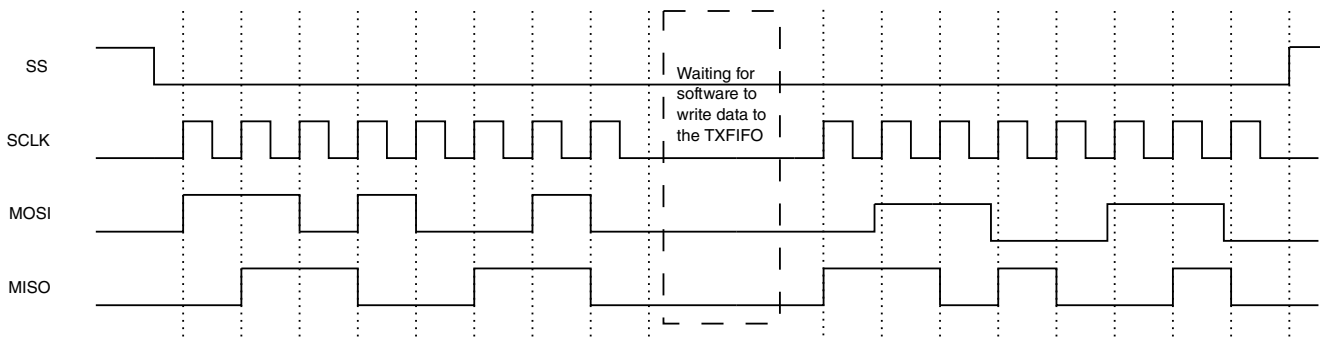


Figure 21-8. SPI Burst While SS_CTL[3:0] is Clear

In Figure 21-8, two 8-bit bursts in the TXFIFO have been combined and transmitted in one SPI burst. The maximum length of a single SPI burst is defined by the BURST LENGTH and limited by the FIFO size. (Figure 21-8 corresponds to a BURST LENGTH of 8.) This provides a way for transferring a longer SPI burst by writing data into TXFIFO while the ECSPI is transmitting.

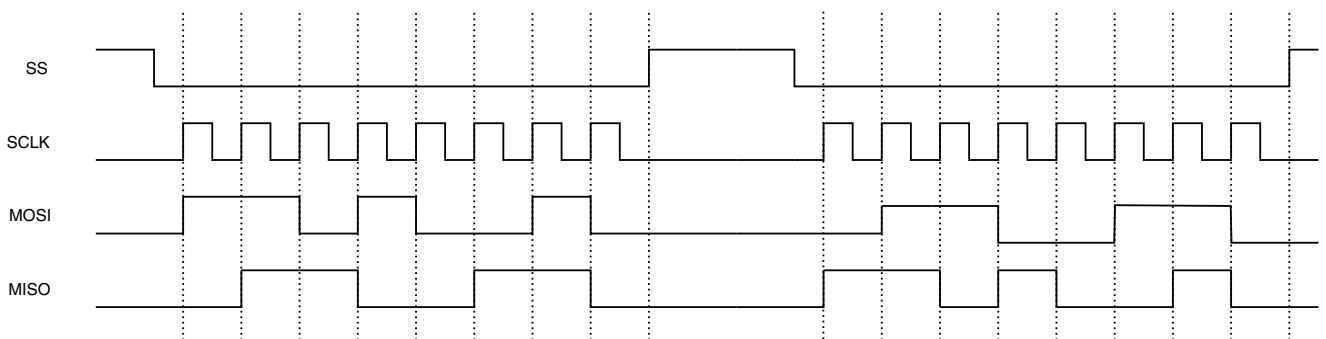


Figure 21-9. SPI Bursts While SS_CTL[3:0] is Set

In Figure 21-9, two FIFO entries are transmitted, one entry with each SPI burst. The ECSPI will continue to transmit SPI bursts until the TXFIFO is empty. When wait states can be inserted between SPI bursts, the SS will negate between SPI bursts until the wait states finish.

21.4.4.1.4 Master Mode with Phase Control

The Phase Control (ECSPI_CONREG[PHA]) bit controls how the transmit data shifts out and the receive data shifts in.

When the Phase control (ECSPI_CONREG[PHA]) bit is set, the transmit data will shift out on the rising edge of SCLK, and the receive data is latched on the falling edge of SCLK. The most-significant bit is output on the first rising SCLK edge.

When ECSPI_CONREG[PHA] is cleared, the transmit data is shifted out on the falling edge of SCLK and the receive data is latched on the rising edge of SCLK. The MSB is output when the host processor loads the transmitted data.

Inverting the SCLK polarity does not impact the edge-triggered operations because they are internal to the serial peripheral interface master. Figure 21-10 shows how SPI burst works with different POL and PHA configuration.

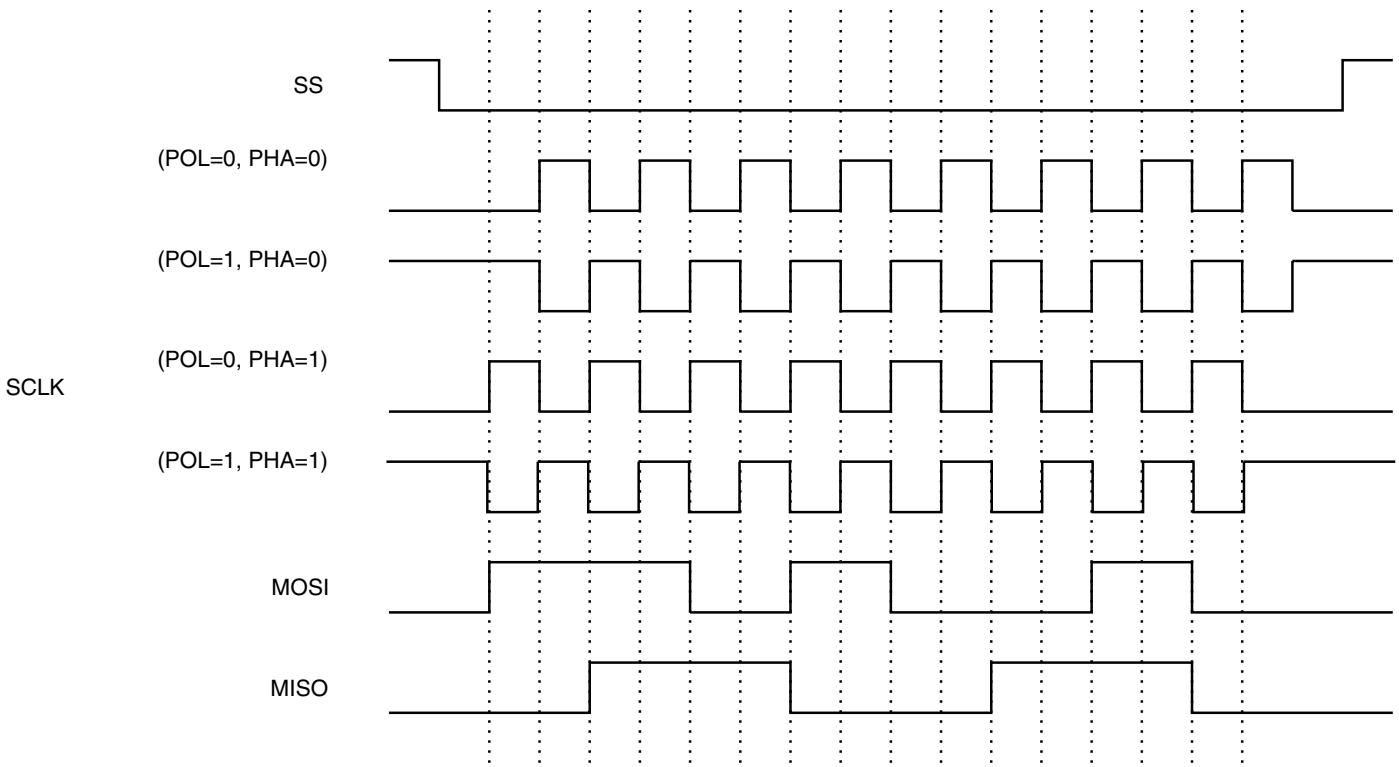


Figure 21-10. SPI Burst with Different POL and PHA Configurations

21.4.4.2 Typical Slave Mode

When the ECSPI is configured as a slave (Mode = 0), software can configure the ECSPI Control register to match the external SPI master's timing. In this configuration, SS becomes an input signal, and is used to latch data in and out of the internal data Shift registers, as well as to advance the data FIFO.

The SS, SCLK, and MOSI are inputs and MISO is output. Most of the timing diagrams are similar to the diagrams shown previously for the SPI in Master mode (Mode = 1), because the inputs come from a SPI master device.

However, the timing is different when SS is used to advance the data FIFO. When the SS_POL=0 is set while the ECSPI is configured in Slave mode, the data FIFO will advance on the rising edge of the SS signal. When the polarity is reversed (SS_POL = 1), the data FIFO will advance on the falling edge of the SS signal.

The figure below shows a SPI burst in which the data FIFO is advanced by the rising edge of the SS signal.

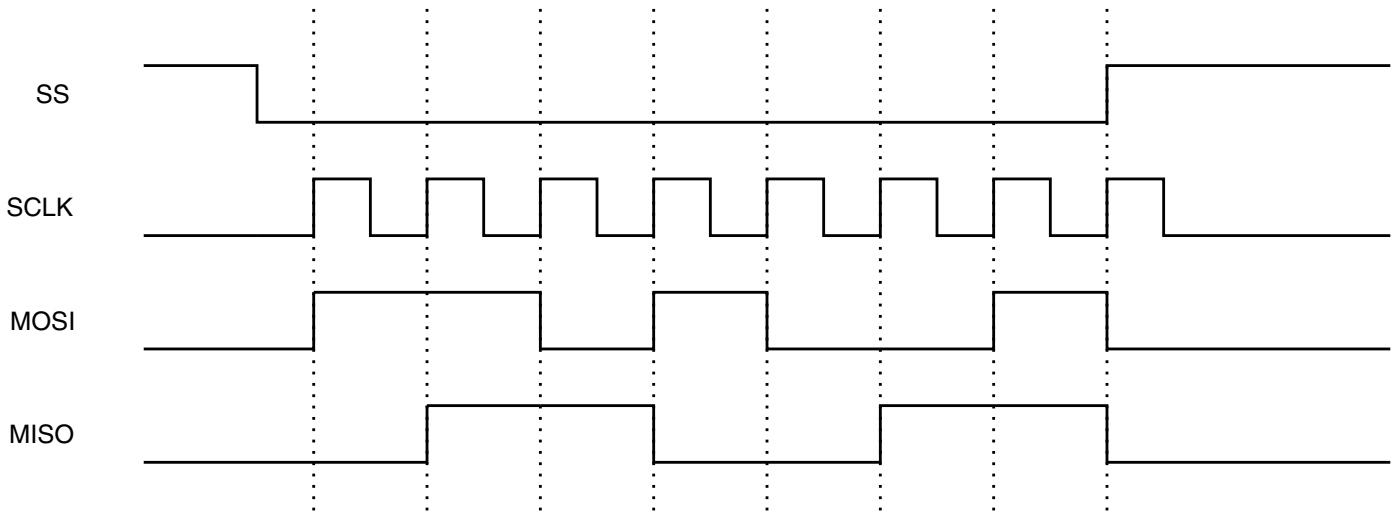


Figure 21-11. Advancing the Data FIFO on the Rising Edge of SS

In the above case, only the most significant 7 bits are loaded to the RXFIFO.

21.4.5 Reset

Whenever a device reset occurs, a reset is performed on the ECSPI, resetting all registers to their default values.

Software can reset the block using the CONREG[EN] bit; see ECSPI.

21.4.6 Interrupts

Interrupt control provides a way to manage the ECSPI FIFOs:

- For transmitting data, software can enable the TXFIFO empty, TXFIFO data request, and TXFIFO full interrupts to maintain the TXFIFO using an interrupt service routine.
- For receiving data, software can enable the RXFIFO ready, RXFIFO data request, and RXFIFO full interrupts to retrieve data from the RXFIFO using an interrupt service routine.

Other interrupt sources can be used to control or debug the SPI bursts:

- The transfer-completed interrupt means that there is no data left in the TXFIFO and that the data in the Shift register has been shifted out.
- The RXFIFO overflow interrupt means that the RXFIFO received more than 64 words and will not accept any other words.

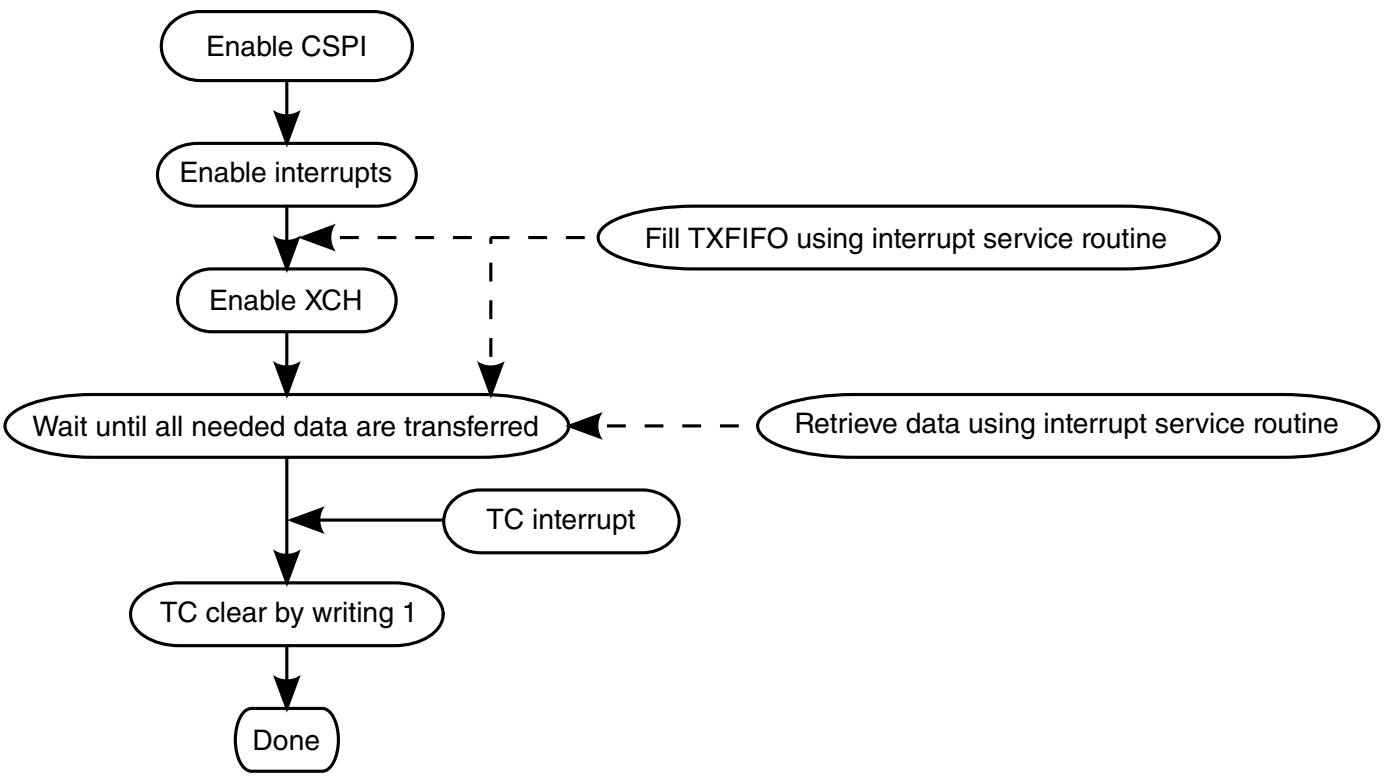


Figure 21-12. Program Sequence of SPI Burst Using Interrupt Control

NOTE

The TC bit does not provide a reliable indication that the transfer is complete. Under some conditions, the TC interrupt can occur before the transfer is completed. If the TC bit is used as an interrupt source, the XCH bit should be polled after the TC interrupt occurs to accurately confirm that the transfer is complete.

21.4.7 DMA

DMA control provides another method to utilize the FIFOs in the ECSPI. By using DMA request and acknowledge signals, larger amounts of data can be transferred, and will reduce interrupts and host processor loading. When the appropriate conditions are matched, the block will send out a DMA request.

The DMA can deal with the following conditions:

- TXFIFO empty
- TXFIFO data request
- RXFIFO data request
- RXFIFO full

The figure below shows a program sequence of SPI bursts using DMA control.

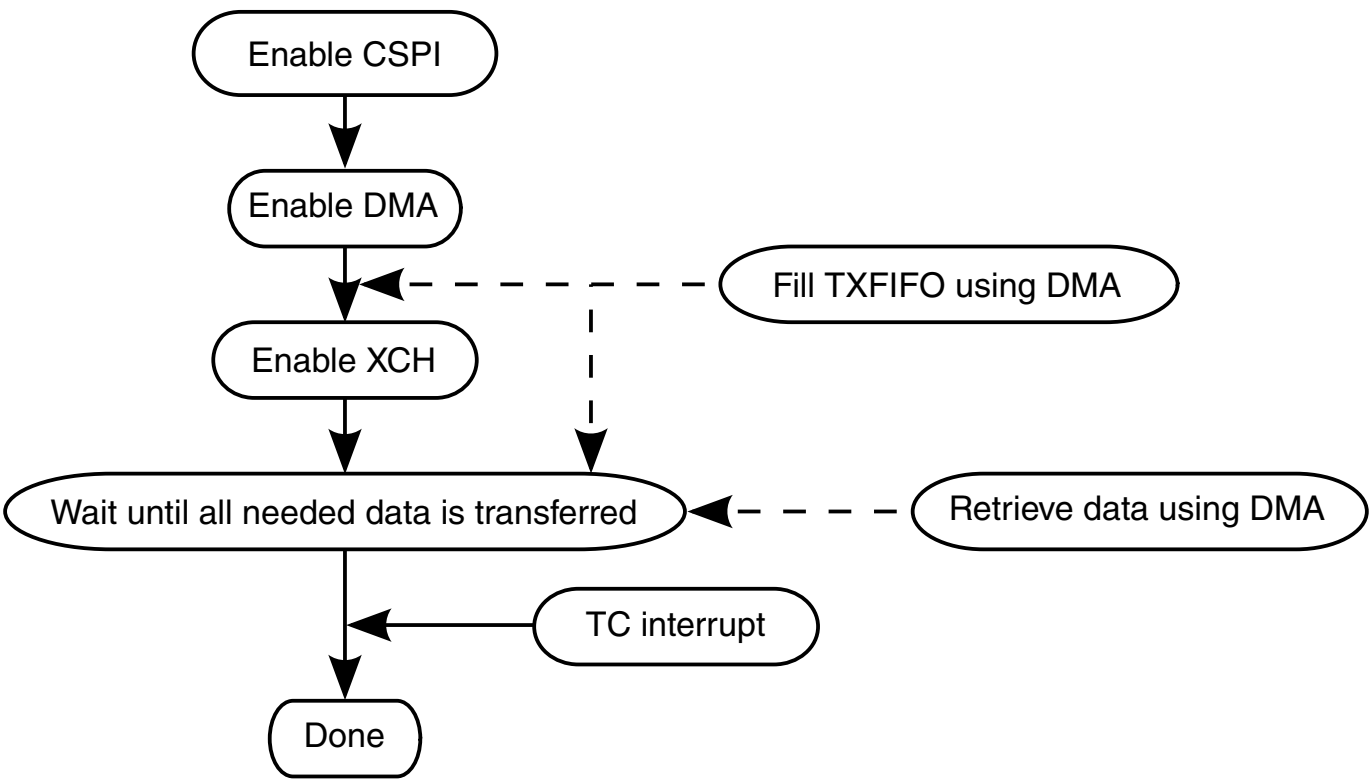


Figure 21-13. Program Sequence of SPI Burst Using DMA

NOTE

The TC bit does not provide a reliable indication that the transfer is complete. Under some conditions, the TC interrupt can occur before the transfer is completed. If the TC bit is used

as an interrupt source, the XCH bit should be polled after the TC interrupt occurs to accurately confirm that the transfer is complete.

21.4.8 Byte Order

The ECSPI does not support byte re-ordering in hardware.

21.5 Initialization

This section provides initialization information for ECSPI.

To initialize the block:

1. Clear the EN bit in ECSPI_CONREG to reset the block.
2. Enable the clocks for ECSPI within the CCM.
3. Configure the Control Register and then set the EN bit in the ECSPI_CONREG to put ECSPI out of reset.
4. Configure corresponding IOMUX for ECSPI external signals.
5. Configure registers of ECSPI properly according to the specifications of the external SPI device.

21.6 Applications

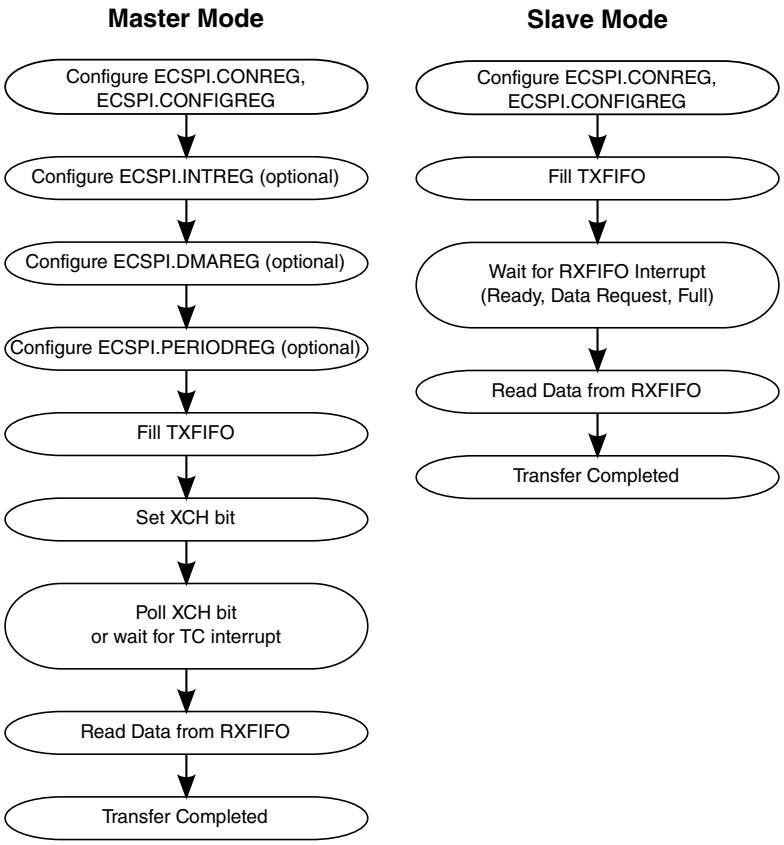


Figure 21-14. Flowchart of the ECSPI Operation

NOTE

The TC bit does not provide a reliable indication that the transfer is complete. Under some conditions, the TC interrupt can occur before the transfer is completed. If the TC bit is used as an interrupt source, the XCH bit should be polled after the TC interrupt occurs to accurately confirm that the transfer is complete.

21.7 ECSPI Memory Map/Register Definition

This section includes the block memory map and detailed descriptions of all registers. For the base address of a particular block instantiation, see the system memory map.

ECSPI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
200_8000	Receive Data Register (ECSPI1_RXDATA)	32	R	0000_0000h	21.7.1/969
200_8004	Transmit Data Register (ECSPI1_TXDATA)	32	W	0000_0000h	21.7.2/970
200_8008	Control Register (ECSPI1_CONREG)	32	R/W	0000_0000h	21.7.3/970
200_800C	Config Register (ECSPI1_CONFIGREG)	32	R/W	0000_0000h	21.7.4/973
200_8010	Interrupt Control Register (ECSPI1_INTREG)	32	R/W	0000_0000h	21.7.5/975
200_8014	DMA Control Register (ECSPI1_DMAREG)	32	R/W	0000_0000h	21.7.6/976
200_8018	Status Register (ECSPI1_STATREG)	32	R/W	0000_0003h	21.7.7/978
200_801C	Sample Period Control Register (ECSPI1_PERIODREG)	32	R/W	0000_0000h	21.7.8/979
200_8020	Test Control Register (ECSPI1_TESTREG)	32	R/W	0000_0000h	21.7.9/981
200_8040	Message Data Register (ECSPI1_MSGDATA)	32	W	0000_0000h	21.7.10/ 982
200_C000	Receive Data Register (ECSPI2_RXDATA)	32	R	0000_0000h	21.7.1/969
200_C004	Transmit Data Register (ECSPI2_TXDATA)	32	W	0000_0000h	21.7.2/970
200_C008	Control Register (ECSPI2_CONREG)	32	R/W	0000_0000h	21.7.3/970
200_C00C	Config Register (ECSPI2_CONFIGREG)	32	R/W	0000_0000h	21.7.4/973
200_C010	Interrupt Control Register (ECSPI2_INTREG)	32	R/W	0000_0000h	21.7.5/975
200_C014	DMA Control Register (ECSPI2_DMAREG)	32	R/W	0000_0000h	21.7.6/976
200_C018	Status Register (ECSPI2_STATREG)	32	R/W	0000_0003h	21.7.7/978
200_C01C	Sample Period Control Register (ECSPI2_PERIODREG)	32	R/W	0000_0000h	21.7.8/979
200_C020	Test Control Register (ECSPI2_TESTREG)	32	R/W	0000_0000h	21.7.9/981
200_C040	Message Data Register (ECSPI2_MSGDATA)	32	W	0000_0000h	21.7.10/ 982
201_0000	Receive Data Register (ECSPI3_RXDATA)	32	R	0000_0000h	21.7.1/969
201_0004	Transmit Data Register (ECSPI3_TXDATA)	32	W	0000_0000h	21.7.2/970
201_0008	Control Register (ECSPI3_CONREG)	32	R/W	0000_0000h	21.7.3/970
201_000C	Config Register (ECSPI3_CONFIGREG)	32	R/W	0000_0000h	21.7.4/973
201_0010	Interrupt Control Register (ECSPI3_INTREG)	32	R/W	0000_0000h	21.7.5/975
201_0014	DMA Control Register (ECSPI3_DMAREG)	32	R/W	0000_0000h	21.7.6/976
201_0018	Status Register (ECSPI3_STATREG)	32	R/W	0000_0003h	21.7.7/978
201_001C	Sample Period Control Register (ECSPI3_PERIODREG)	32	R/W	0000_0000h	21.7.8/979
201_0020	Test Control Register (ECSPI3_TESTREG)	32	R/W	0000_0000h	21.7.9/981
201_0040	Message Data Register (ECSPI3_MSGDATA)	32	W	0000_0000h	21.7.10/ 982
201_4000	Receive Data Register (ECSPI4_RXDATA)	32	R	0000_0000h	21.7.1/969
201_4004	Transmit Data Register (ECSPI4_TXDATA)	32	W	0000_0000h	21.7.2/970
201_4008	Control Register (ECSPI4_CONREG)	32	R/W	0000_0000h	21.7.3/970
201_400C	Config Register (ECSPI4_CONFIGREG)	32	R/W	0000_0000h	21.7.4/973
201_4010	Interrupt Control Register (ECSPI4_INTREG)	32	R/W	0000_0000h	21.7.5/975
201_4014	DMA Control Register (ECSPI4_DMAREG)	32	R/W	0000_0000h	21.7.6/976

Table continues on the next page...

ECSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
201_4018	Status Register (ECSPI4_STATREG)	32	R/W	0000_0003h	21.7.7/978
201_401C	Sample Period Control Register (ECSPI4_PERIODREG)	32	R/W	0000_0000h	21.7.8/979
201_4020	Test Control Register (ECSPI4_TESTREG)	32	R/W	0000_0000h	21.7.9/981
201_4040	Message Data Register (ECSPI4_MSGDATA)	32	W	0000_0000h	21.7.10/ 982

21.7.1 Receive Data Register (ECSPIx_RXDATA)

The Receive Data register (ECSPI_RXDATA) is a read-only register that forms the top word of the 64 x 32 receive FIFO. This register holds the data received from an external SPI device during a data transaction. Only word-sized read operations are allowed.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ECSPI_RXDATA																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_RXDATA field descriptions

Field	Description
ECSPI_RXDATA	Receive Data. This register holds the top word of the receive data FIFO. The FIFO is advanced for each read of this register. The data read is undefined when the Receive Data Ready (RR) bit in the Interrupt Control/Status register is cleared. Zeros are read when ECSPI is disabled.

21.7.2 Transmit Data Register (ECSPIx_TXDATA)

The Transmit Data (ECSPI_TXDATA) register is a write-only data register that forms the bottom word of the 64 x 32 TXFIFO. The TXFIFO can be written to as long as it is not full, even when the SPI Exchange bit (XCH) in ECSPI_CONREG is set. This allows software to write to the TXFIFO during a SPI data exchange process. Writes to this register are ignored when the ECSPI is disabled (ECSPI_CONREG[EN] bit is cleared).

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																																
W	ECSPI_TXDATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_TXDATA field descriptions

Field	Description
ECSPI_TXDATA	Transmit Data. This register holds the top word of data loaded into the FIFO. Data written to this register must be a word operation. The number of bits actually transmitted is determined by the BURST_LENGTH field of the corresponding SPI Control register. If this field contains more bits than the number specified by BURST_LENGTH, the extra bits are ignored. For example, to transfer 10 bits of data, a 32-bit word must be written to this register. Bits 9-0 are shifted out and bits 31-10 are ignored. When the ECSPI is operating in Slave mode, zeros are shifted out when the FIFO is empty. Zeros are read when ECSPI is disabled.

21.7.3 Control Register (ECSPIx_CONREG)

The Control Register (ECSPI_CONREG) allows software to enable the ECSPI , configure its operating modes, specify the divider value, and SPI_RDY control signal, and define the transfer length.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BURST_LENGTH												CHANNEL_ SELECT		DRCTL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PRE_DIVIDER				POST_DIVIDER				CHANNEL_MODE				SMC	XCH	HT	EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_CONREG field descriptions

Field	Description
31–20 BURST_ LENGTH	Burst Length. This field defines the length of a SPI burst to be transferred. The Chip Select (SS) will remain asserted until all bits in a SPI burst are shifted out. A maximum of 2^12 bits can be transferred in a single SPI burst. In master mode, it controls the number of bits per SPI burst. Since the shift register always loads 32-bit data from transmit FIFO, only the n least-significant (n = BURST LENGTH + 1) will be shifted out. The remaining bits will be ignored. Number of Valid Bits in a SPI burst. 0x000 A SPI burst contains the 1 LSB in a word. 0x001 A SPI burst contains the 2 LSB in a word. 0x002 A SPI burst contains the 3 LSB in a word. ... 0x01F A SPI burst contains all 32 bits in a word. 0x020 A SPI burst contains the 1 LSB in first word and all 32 bits in second word. 0x021 A SPI burst contains the 2 LSB in first word and all 32 bits in second word. ... 0xFFE A SPI burst contains the 31 LSB in first word and 2^7 -1 words. 0xFFF A SPI burst contains 2^7 words.
19–18 CHANNEL_ SELECT	SPI CHANNEL SELECT bits. Select one of four external SPI Master/Slave Devices. In master mode, these two bits select the external slave devices by asserting the Chip Select (SSn) outputs. Only the selected Chip Select (SSn) signal can be active at a given time; the remaining three signals will be negated. 00 Channel 0 is selected. Chip Select 0 (SS0) will be asserted. 01 Channel 1 is selected. Chip Select 1 (SS1) will be asserted. 10 Channel 2 is selected. Chip Select 2 (SS2) will be asserted. 11 Channel 3 is selected. Chip Select 3 (SS3) will be asserted.
17–16 DRCTL	SPI Data Ready Control. This field selects the utilization of the SPI_RDY signal in master mode. ECSPI checks this field before it starts an SPI burst. 00 The SPI_RDY signal is a don't care. 01 Burst will be triggered by the falling edge of the SPI_RDY signal (edge-triggered). 10 Burst will be triggered by a low level of the SPI_RDY signal (level-triggered). 11 Reserved.
15–12 PRE_DIVIDER	SPI Pre Divider. ECSPI uses a two-stage divider to generate the SPI clock. This field defines the pre-divider of the reference clock. 0000 Divide by 1. 0001 Divide by 2. 0010 Divide by 3. ... 1101 Divide by 14. 1110 Divide by 15. 1111 Divide by 16.
11–8 POST_DIVIDER	SPI Post Divider. ECSPI uses a two-stage divider to generate the SPI clock. This field defines the post-divider of the reference clock using the equation: 2^{n} . 0000 Divide by 1. 0001 Divide by 2.

Table continues on the next page...

ECSPIx_CONREG field descriptions (continued)

Field	Description
	0010 Divide by 4. ... 1110 Divide by 2^{14}. 1111 Divide by 2^{15}.
7–4 CHANNEL_ MODE	SPI CHANNEL MODE selects the mode for each SPI channel. CHANNEL MODE[3] is for SPI channel 3. CHANNEL MODE[2] is for SPI channel 2. CHANNEL MODE[1] is for SPI channel 1. CHANNEL MODE[0] is for SPI channel 0. 0 Slave mode. 1 Master mode.
3 SMC	Start Mode Control. This bit applies only to channels configured in Master mode (CHANNEL MODE = 1). It controls how the ECSPI starts a SPI burst, either through the SPI exchange bit, or immediately when the TXFIFO is written to. 0 SPI Exchange Bit (XCH) controls when a SPI burst can start. Setting the XCH bit will start a SPI burst or multiple bursts. This is controlled by the SPI SS Wave Form Select (SS_CTL). Refer to XCH and SS_CTL descriptions. 1 Immediately starts a SPI burst when data is written in TXFIFO.
2 XCH	SPI Exchange Bit. This bit applies only to channels configured in Master mode (CHANNEL MODE = 1). If the Start Mode Control (SMC) bit is cleared, writing a 1 to this bit starts one SPI burst or multiple SPI bursts according to the SPI SS Wave Form Select (SS_CTL). The XCH bit remains set while either the data exchange is in progress, or when the ECSPI is waiting for an active input if SPIRDY is enabled through DRCTL. This bit is cleared automatically when all data in the TXFIFO and the shift register has been shifted out. 0 Idle. 1 Initiates exchange (write) or busy (read).
1 HT	Hardware Trigger Enable. This bit is used in master mode only. It enables hardware trigger (HT) mode. Note, HT mode is not supported by this product. 0 Disable HT mode. 1 Enable HT mode.
0 EN	SPI Block Enable Control. This bit enables the ECSPI. This bit must be set before writing to other registers or initiating an exchange. Writing zero to this bit disables the block and resets the internal logic with the exception of the ECSPI_CONREG. The block's internal clocks are gated off whenever the block is disabled. 0 Disable the block. 1 Enable the block.

21.7.4 Config Register (ECSPIx_CONFIGREG)

The Config Register (ECSPI_CONFIGREG) allows software to configure each SPI channel, configure its operating modes, specify the phase and polarity of the clock, configure the Chip Select (SS), and define the HT transfer length. Note, HT mode is not supported by this product.

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			HT_LENGTH					SCLK_CTL				DATA_CTL				SS_POL				SS_CTL				SCLK_POL				SCLK_PHA			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_CONFIGREG field descriptions

Field	Description
31–29 -	This field is reserved. Reserved
28–24 HT_LENGTH	HT LENGTH. This field defines the message length in HT Mode. Note, HT mode is not supported by this product. The length in bits of one message is (HT LENGTH + 1).
23–20 SCLK_CTL	SCLK CTL. This field controls the inactive state of SCLK for each SPI channel. SCLK CTL[3] is for SPI channel 3. SCLK CTL[2] is for SPI channel 2. SCLK CTL[1] is for SPI channel 1. SCLK CTL[0] is for SPI channel 0. 0 Stay low. 1 Stay high.
19–16 DATA_CTL	DATA CTL. This field controls inactive state of the data line for each SPI channel. DATA CTL[3] is for SPI channel 3. DATA CTL[2] is for SPI channel 2. DATA CTL[1] is for SPI channel 1. DATA CTL[0] is for SPI channel 0. 0 Stay high. 1 Stay low.
15–12 SS_POL	SPI SS Polarity Select. In both Master and Slave modes, this field selects the polarity of the Chip Select (SS) signal. SS POL[3] is for SPI channel 3. SS POL[2] is for SPI channel 2. SS POL[1] is for SPI channel 1. SS POL[0] is for SPI channel 0.

Table continues on the next page...

ECSPIx_CONFIGREG field descriptions (continued)

Field	Description
	0 Active low. 1 Active high.
11–8 SS_CTL	SPI SS Wave Form Select. In master mode, this field controls the output wave form of the Chip Select (SS) signal when the SMC (Start Mode Control) bit is cleared. The SS_CTL bits are ignored if the SMC bit is set. SS CTL[3] is for SPI channel 3. SS CTL[2] is for SPI channel 2. SS CTL[1] is for SPI channel 1. SS CTL[0] is for SPI channel 0. In slave mode, this bit controls when the SPI burst is completed. An SPI burst is completed by the Chip Select (SS) signal edges. (SSPOL = 0: rising edge; SSPOL = 1: falling edge) The RXFIFO is advanced whenever a Chip Select (SS) signal edge is detected or the shift register contains 32-bits of valid data. 0 In master mode - only one SPI burst will be transmitted. 1 In master mode - Negate Chip Select (SS) signal between SPI bursts. Multiple SPI bursts will be transmitted. The SPI transfer will automatically stop when the TXFIFO is empty. 0 In slave mode - an SPI burst is completed when the number of bits received in the shift register is equal to (BURST LENGTH + 1). Only the n least-significant bits (n = BURST LENGTH[4:0] + 1) of the first received word are valid. All bits subsequent to the first received word in RXFIFO are valid. 1 Reserved
7–4 SCLK_POL	SPI Clock Polarity Control. This field controls the polarity of the SCLK signal. See Figure 21-10 for more information. SCLK_POL[3] is for SPI channel 3. SCLK_POL[2] is for SPI channel 2. SCLK_POL[1] is for SPI channel 1. SCLK_POL[0] is for SPI channel 0. 0 Active high polarity (0 = Idle). 1 Active low polarity (1 = Idle).
SCLK_PHA	SPI Clock/Data Phase Control. This field controls the clock/data phase relationship. See Figure 21-10 for more information. SCLK PHA[3] is for SPI channel 3. SCLK PHA[2] is for SPI channel 2. SCLK PHA[1] is for SPI channel 1. SCLK PHA[0] is for SPI channel 0. 0 Phase 0 operation. 1 Phase 1 operation.

21.7.5 Interrupt Control Register (ECSPIx_INTREG)

The Interrupt Control Register (ECSPI_INTREG) enables the generation of interrupts to the host processor. If the ECSPI is disabled, this register reads zero.

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TCEN	ROEN	RFEN	RDREN	RREN	TFEN	TDREN	TEEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_INTREG field descriptions

Field	Description
31–8 -	This field is reserved. Reserved
7 TCEN	Transfer Completed Interrupt enable. This bit enables the Transfer Completed Interrupt. 0 Disable 1 Enable
6 ROEN	RXFIFO Overflow Interrupt enable. This bit enables the RXFIFO Overflow Interrupt. 0 Disable 1 Enable
5 RFEN	RXFIFO Full Interrupt enable. This bit enables the RXFIFO Full Interrupt. 0 Disable 1 Enable
4 RDREN	RXFIFO Data Request Interrupt enable. This bit enables the RXFIFO Data Request Interrupt when the number of data entries in the RXFIFO is greater than RX_THRESHOLD. 0 Disable 1 Enable
3 RREN	RXFIFO Ready Interrupt enable. This bit enables the RXFIFO Ready Interrupt. 0 Disable 1 Enable

Table continues on the next page...

ECSPIx_INTREG field descriptions (continued)

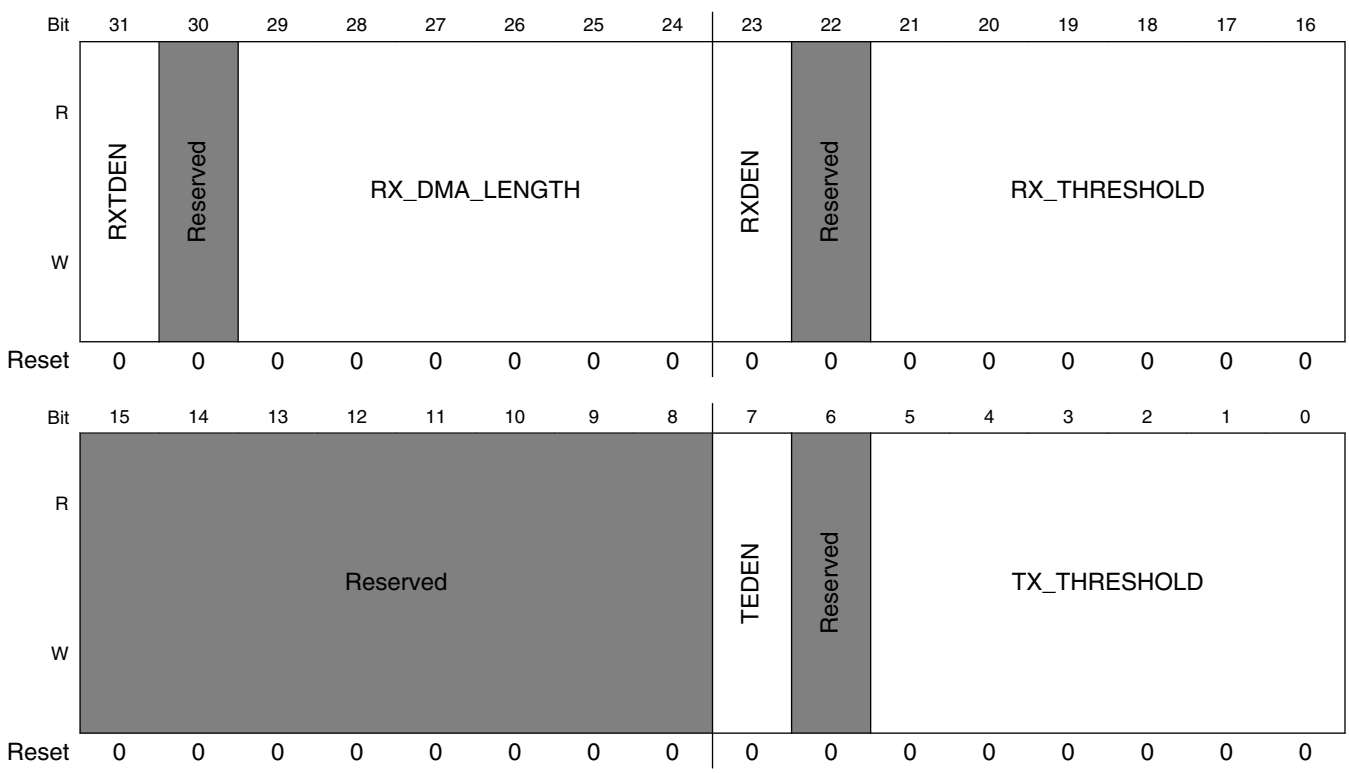
Field	Description
2 TFEN	TXFIFO Full Interrupt enable. This bit enables the TXFIFO Full Interrupt. 0 Disable 1 Enable
1 TDREN	TXFIFO Data Request Interrupt enable. This bit enables the TXFIFO Data Request Interrupt when the number of data entries in the TXFIFO is less than or equal to TX_THRESHOLD. 0 Disable 1 Enable
0 TEEN	TXFIFO Empty Interrupt enable. This bit enables the TXFIFO Empty Interrupt. 0 Disable 1 Enable

21.7.6 DMA Control Register (ECSPIx_DMAREG)

The Direct Memory Access Control Register (ECSPI_DMAREG) provides software a way to use an on-chip DMA controller for ECSPI data. Internal DMA request signals enable direct data transfers between the ECSPI FIFOs and system memory. The ECSPI sends out DMA requests when the appropriate FIFO conditions are matched.

If the ECSPI is disabled, this register is read as 0.

Address: Base address + 14h offset



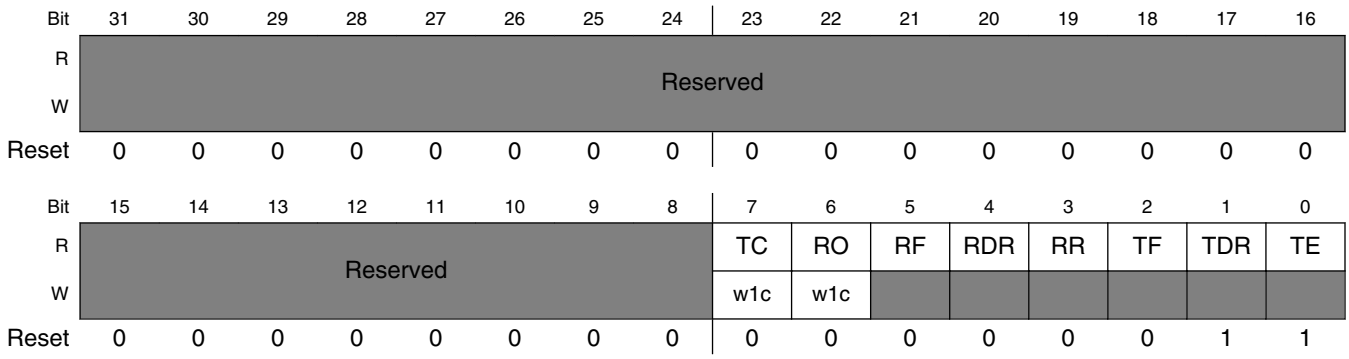
ECSPIx_DMAREG field descriptions

Field	Description
31 RXTDEN	RXFIFO TAIL DMA Request/Interrupt Enable. This bit enables an internal counter that is increased at each read of the RXFIFO. This counter is cleared automatically when it reaches RX DMA LENGTH. If the number of words remaining in the RXFIFO is greater than or equal to RX DMA LENGTH, a DMA request/ interrupt is generated even if it is less than or equal to RX_THRESHOLD. 0 Disable 1 Enable
30 -	This field is reserved. Reserved
29–24 RX_DMA_ LENGTH	RX DMA LENGTH. This field defines the burst length of a DMA operation. Applies only when RXTDEN is set.
23 RXDEN	RXFIFO DMA Request Enable. This bit enables/disables the RXFIFO DMA Request. 0 Disable 1 Enable
22 -	This field is reserved. Reserved
21–16 RX_ THRESHOLD	RX THRESHOLD. This field defines the FIFO threshold that triggers a RX DMA/INT request. A RX DMA/INT request is issued when the number of data entries in the RXFIFO is greater than RX_THRESHOLD.
15–8 -	This field is reserved. Reserved
7 TEDEN	TXFIFO Empty DMA Request Enable. This bit enables/disables the TXFIFO Empty DMA Request. 0 Disable 1 Enable
6 -	This field is reserved. Reserved
TX_ THRESHOLD	TX THRESHOLD. This field defines the FIFO threshold that triggers a TX DMA/INT request. A TX DMA/INT request is issued when the number of data entries in the TXFIFO is not greater than TX_THRESHOLD.

21.7.7 Status Register (ECSPIx_STATREG)

The ECSPI Status Register (ECSPI_STATREG) reflects the status of the ECSPI's operating condition. If the ECSPI is disabled, this register reads 0x0000_0003.

Address: Base address + 18h offset



ECSPIx_STATREG field descriptions

Field	Description
31–8 -	This field is reserved. Reserved
7 TC	Transfer Completed Status bit. Writing 1 to this bit clears it. **NOTE:** The TC bit does not provide a reliable indication that the transfer is complete. Under some conditions, the TC interrupt can occur before the transfer is completed. If the TC bit is used as an interrupt source, the XCH bit should be polled after the TC interrupt occurs to accurately confirm that the transfer is complete. 0 Transfer in progress. 1 Transfer completed.
6 RO	RXFIFO Overflow. When set, this bit indicates that RXFIFO has overflowed. Writing 1 to this bit clears it. 0 RXFIFO has no overflow. 1 RXFIFO has overflowed.
5 RF	RXFIFO Full. This bit is set when the RXFIFO is full. 0 Not Full. 1 Full.
4 RDR	RXFIFO Data Request. 0 When RXTDE is set - Number of data entries in the RXFIFO is not greater than RX_THRESHOLD. 1 When RXTDE is set - Number of data entries in the RXFIFO is greater than RX_THRESHOLD or a DMA TAIL DMA condition exists. 0 When RXTDE is clear - Number of data entries in the RXFIFO is not greater than RX_THRESHOLD. 1 When RXTDE is clear - Number of data entries in the RXFIFO is greater than RX_THRESHOLD.

Table continues on the next page...

ECSPIx_STATREG field descriptions (continued)

Field	Description
3 RR	RXFIFO Ready. This bit is set when one or more words are stored in the RXFIFO. 0 No valid data in RXFIFO. 1 More than 1 word in RXFIFO.
2 TF	TXFIFO Full. This bit is set when if the TXFIFO is full. 0 TXFIFO is not Full. 1 TXFIFO is Full.
1 TDR	TXFIFO Data Request. 0 Number of valid data slots in TXFIFO is greater than TX_THRESHOLD. 1 Number of valid data slots in TXFIFO is not greater than TX_THRESHOLD.
0 TE	TXFIFO Empty. This bit is set if the TXFIFO is empty. 0 TXFIFO contains one or more words. 1 TXFIFO is empty.

21.7.8 Sample Period Control Register (ECSPIx_PERIODREG)

The Sample Period Control Register (ECSPI_PERIODREG) provides software a way to insert delays (wait states) between consecutive SPI transfers. Control bits in this register select the clock source for the sample period counter and the delay count indicating the number of wait states to be inserted between data transfers.

The delay counts apply only when the current channel is operating in Master mode (ECSPI_CONREG[CHANNEL MODE] = 1).ECSPI_PERIODREG also contains the CSD CTRL field used to insert a delay between the Chip Select's active edge and the first SPI Clock edge.

Address: Base address + 1Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved										CSD_CTL					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSRC	SAMPLE_PERIOD														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

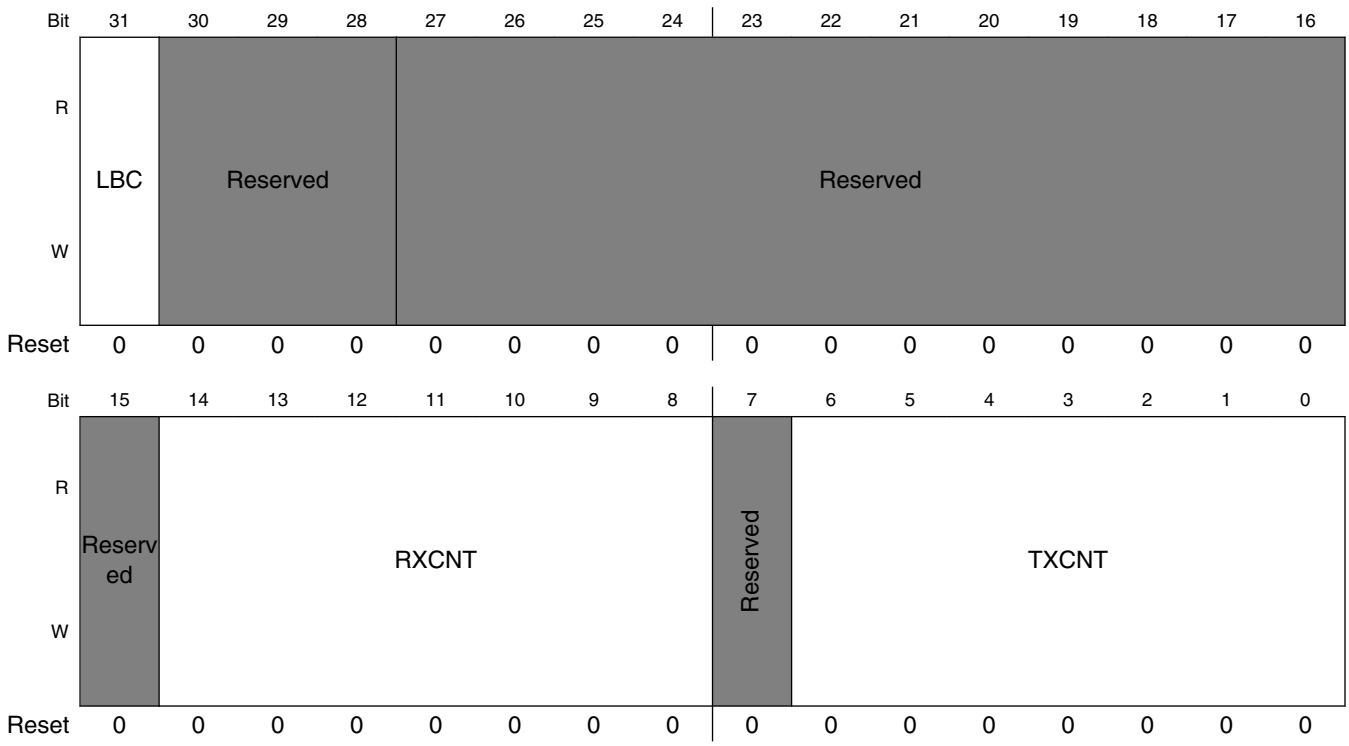
ECSPIx_PERIODREG field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–16 CSD_CTL	Chip Select Delay Control bits. This field defines how many SPI clocks will be inserted between the chip select's active edge and the first SPI clock edge. The range is from 0 to 63.
15 CSRC	Clock Source Control. This bit selects the clock source for the sample period counter. 0 SPI Clock (SCLK) 1 Low-Frequency Reference Clock (32.768 KHz)
SAMPLE_ PERIOD	Sample Period Control. These bits control the number of wait states to be inserted in data transfers. During the idle clocks, the state of the SS output will operate according to the SS_CTL control field in the ECSPI_CONREG register. 0x0000 0 wait states inserted 0x0001 1 wait state inserted ... ... 0x7FFE 32766 wait states inserted 0x7FFF 32767 wait states inserted

21.7.9 Test Control Register (ECSPIx_TESTREG)

The Test Control Register (ECSPI_TESTREG) provides software a mechanism to internally connect the receive and transmit devices of the ECSPI, and monitor the contents of the receive and transmit FIFOs.

Address: Base address + 20h offset



ECSPIx_TESTREG field descriptions

Field	Description
31 LBC	Loop Back Control. This bit is used in Master mode only. When this bit is set, the ECSPI connects the transmitter and receiver sections internally, and the data shifted out from the most-significant bit of the shift register is looped back into the least-significant bit of the Shift register. In this way, a self-test of the complete transmit/receive path can be made. The output pins continue to operate normally, however the input pins are ignored. 0 Not connected. 1 Transmitter and receiver sections internally connected for Loopback.
30–28 -	This field is reserved. Reserved, all bits should be ignored.
27–15 -	This field is reserved. Reserved
14–8 RXCNT	RXFIFO Counter. This field indicates the number of words in the RXFIFO.

Table continues on the next page...

ECSPIx_TESTREG field descriptions (continued)

Field	Description
7 -	This field is reserved. Reserved
TXCNT	TXFIFO Counter. This field indicates the number of words in the TXFIFO.

21.7.10 Message Data Register (ECSPIx_MSGDATA)

The Message Data Register (ECSPI_MSGDATA) forms the top word of the 16 x 32 MSG Data FIFO. Only word-size accesses are allowed for this register. Reading from this register returns zero, and writing to this register stores data in the MSG Data FIFO.

Address: Base address + 40h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																																
W	ECSPI_MSGDATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_MSGDATA field descriptions

Field	Description
ECSPI_ MSGDATA	ECSPI_MSGDATA holds the top word of MSG Data FIFO. The MSG Data FIFO is advanced for each write of this register. The data read is zero. The data written to this register is stored in the MSG Data FIFO.

Chapter 22
External Interface Module (EIM)

22.1 Overview

The EIM handles the interface to devices external to the chip, including generation of chip selects, clock and control for external peripherals and memory. It provides asynchronous access to devices with SRAM-like interface and synchronous access to devices with NOR-Flash-like or PSRAM-like interface.

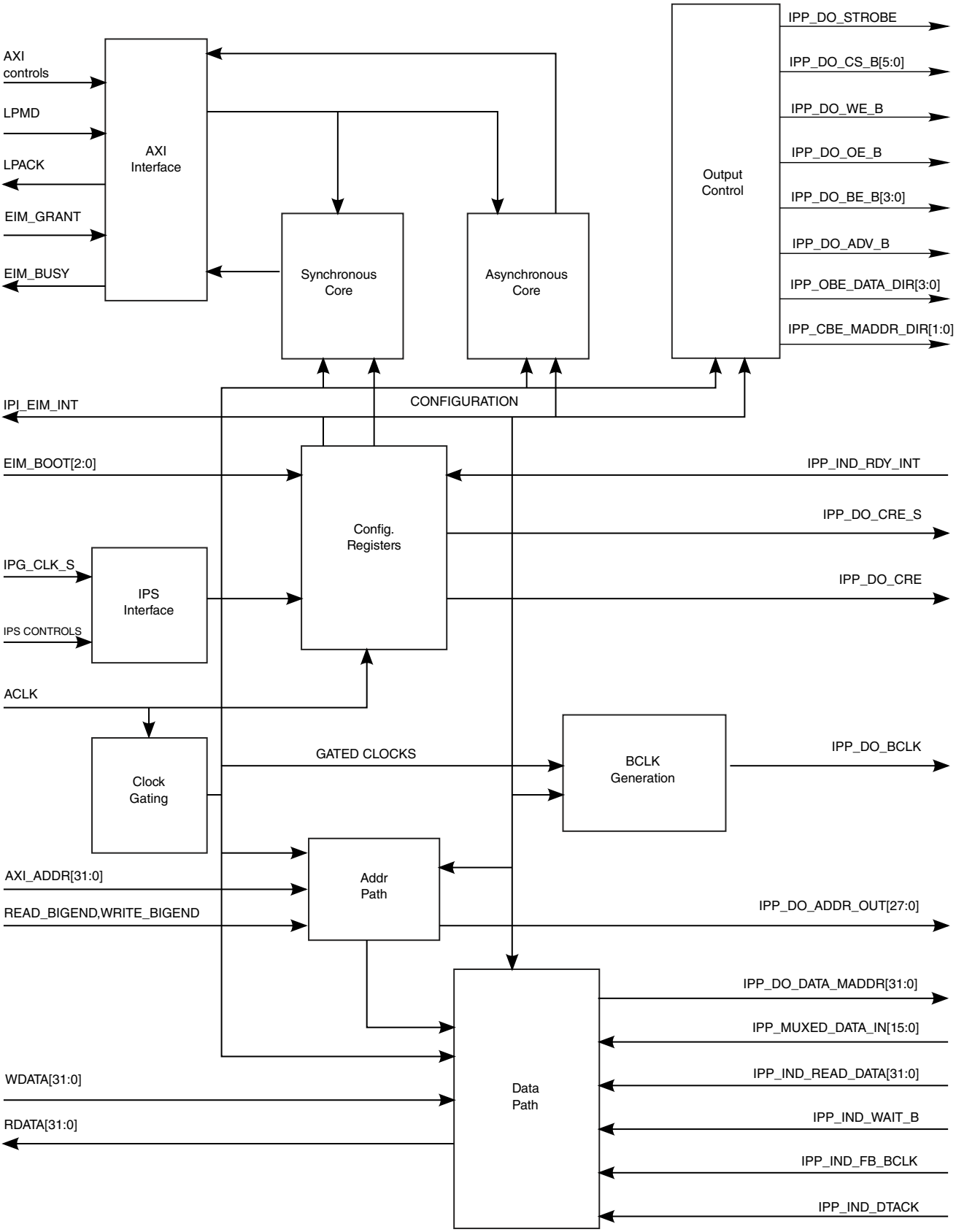


Figure 22-1. EIM Diagram

22.1.1 Features

- Four chip selects for external devices
  - Flexible address decoding. Each chip select memory space determined separately, according to VIA port configuration (see Chip Select Memory Map). Configurable Chip Select 0 base address (by VIA)
  - Individual select signal for each one of the memory space defined. Up to 6 memory spaces may be defined and programmed individually.
  - 128 MByte maximum supported density by default (AUS bit is cleared). When the AUS bit is set, maximum supported density is 32 MBytes.
- Selectable Write Protection for each Chip Select
- Support for multiplexed address / data bus operation x16 and x32 port size
- Programmable Data Port Size for each Chip Select (x8, x16 and x32)
- Programmable Wait-State generator for each Chip Select, for write and read accesses separately
- Asynchronous accesses with programmable setup and hold times for control signals
- Support for Asynchronous page mode accesses (x16 and x32 port size)
- Support continuous Burst Clock which can be used as reference clock for FPGA
- Independent synchronous Memory Burst Read Mode support for NOR-Flash and PSRAM memories (x16 and x32 port size)
- Independent synchronous Memory Burst Write Mode support for PSRAM and NOR-Flash like memories (CellularRAM™ from Micron, Infineon, and Cypress, OneNAND™ and utRAM™ from Samsung, and COSMORAM™ from Toshiba)
- Support of NAND-Flash devices with NOR-Flash like interface - MDOC™ (M-Systems), OneNAND™ (Samsung)
- Independent programmable variable/fix Latency support for read and write synchronous (burst) mode
- Support for Big Endian and Little Endian operation modes per access
- Arm AXI slave interface. One ID at a time support.
- External Interrupt support, RDY_INT signal function as external interrupt
- Boot from external device support according to boot signals, using RDY_INT signal
  - RDY signal support assertion after reset for MDOC™ (M-Systems) device
  - INT signal support assertion after reset for OneNAND™ (Samsung) device

22.1.2 Modes of Operation

The EIM has the following modes of operation:

- Asynchronous Mode
- Asynchronous Page Mode

- Multiplexed Address/Data mode
- Burst Clock Mode
- Low Power Modes
- Boot Mode

See details in the EIM Operational Modes.

22.1.2.1 Asynchronous Mode

This is a non-burst mode that is used for SRAM access. In this mode, a single data is read/written with each access (asserted address).

All controls' timings are controlled by preset values in Chip Select Configuration Registers.

22.1.2.2 Asynchronous Page Read Mode

Setting the APR bit causes the EIM to perform memory burst accesses by emulating page mode operation.

The external address asserts for each piece of data. The initial access timing is according to RWSC field, and the next address assertions timing is according to PAT field. When APR bit is set, RCSN OEN, RADVN and RBEN fields are ignored for burst access to the external device.

The page size can be set via the BL field to 2, 4, 8, 16, or 32 words (the word size is determined by the DSZ field).

22.1.2.3 Multiplexed Address/Data Mode

In this mode, multiplexing addresses and data bits on the same pins is supported for synchronous/asynchronous accesses to x8/x16/x32 data width memory devices.

For more information about the pins that drive data/address in 8/16/32 non-muxed mode and 16/32 muxed mode, see the following table.

Table 22-1. EIM multiplexing

Setup	Non Multiplexed Address/Data Mode							Multiplexed Address/ Data mode	
	8 Bit				16 Bit		32 Bit	16 Bit	32 Bit
	MUM = 0, DSZ = 100	MUM = 0, DSZ = 101	MUM = 0, DSZ = 110	MUM = 0, DSZ = 111	MUM = 0, DSZ = 001	MUM = 0, DSZ = 010	MUM = 0, DSZ = 011	MUM = 1, DSZ = 001	MUM = 1, DSZ = 011
A[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]
A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_D[9:0]
D[7:0], EIM_EB0	EIM_D[7:0]	-	-	-	EIM_D[7:0]	-	EIM_D[7:0]	EIM_DA[7:0]	EIM_DA[7:0]
D[15:8], EIM_EB1	-	EIM_D[15:8]	-	-	EIM_D[15:8]	-	EIM_D[15:8]	EIM_DA[15:8]	EIM_DA[15:8]
D[23:16], EIM_EB2	-	-	EIM_D[23:16]	-	-	EIM_D[23:16]	EIM_D[23:16]	-	EIM_D[7:0]
D[31:24], EIM_EB3	-	-	-	EIM_D[31:24]	-	EIM_D[31:24]	EIM_D[31:24]	-	EIM_D[15:8]

22.1.2.4 Burst Clock Mode

The controller has the ability to support burst synchronous operations in various frequencies, depending on the frequency of the input clock supplied by the system (EIM clock).

The EIM clock can be divided by one, two, three or four, and its frequency can be changed according to the requirements. Variable and fix latency are supported for this mode, according to the external device requirements.

- Synchronous read mode. This is a burst mode, which is used for reading from Flash/ PSRAM memory devices. In this mode, after address assertion a burst of sequential data can be read. Data exchange is carried out according to BCLK being generated by EIM. An access is delayed according to external WAIT_B signal assertion (signal from the memory device).
- Synchronous write mode. A burst mode used for accessing external devices, which support synchronous write type of access (PSRAM protocol). In this mode, after address assertion a burst of sequential data can be written to the external device. Access may be delayed according to WAIT_B signal assertion (signal from the memory device) before first piece of data arrived to the external device.

NOTE

Maximum frequency of the EIM main clock is 133 MHz. It may be reduced by the system for special cases of external

devices, which demand a different frequency then integer division of the 133 MHz clock.

22.1.2.5 Low Power Modes

The input clock is gated by ACT_CS bits. When all the ACT_CS are negated (all CS disable) the internal clock is turned off; awready/wready & arready signal are de-asserted and the master can't access the EIM.

22.1.2.6 Boot Mode

It is possible to perform a boot operation from external device located on CS0. The configuration of the relevant bits are done with boot mode signals according to the external device parameters (for example, port size and protocol assertion).

See System Boot for more details.

22.2 External Signals

The following table describes the external signals of EIM:

Table 22-2. EIM External Signals

Signal	Description	Pad	Mode	Direction
EIM_ACLK_FREERUN	AXI clock signal	EIM_A25	ALT9	I
		EIM_D31	ALT9	
		EIM_EB3	ALT9	
EIM_AD00	LSB multiplexed Address/Data Bus signal	EIM_DA0	ALT0	IO
EIM_AD01	LSB multiplexed Address/Data Bus signal	EIM_DA1	ALT0	IO
EIM_AD02	LSB multiplexed Address/Data Bus signal	EIM_DA2	ALT0	IO
EIM_AD03	LSB multiplexed Address/Data Bus signal	EIM_DA3	ALT0	IO
EIM_AD04	LSB multiplexed Address/Data Bus signal	EIM_DA4	ALT0	IO
EIM_AD05	LSB multiplexed Address/Data Bus signal	EIM_DA5	ALT0	IO
EIM_AD06	LSB multiplexed Address/Data Bus signal	EIM_DA6	ALT0	IO

Table continues on the next page...

Table 22-2. EIM External Signals (continued)

Signal	Description	Pad	Mode	Direction
EIM_AD07	LSB multiplexed Address/Data Bus signal	EIM_DA7	ALT0	IO
EIM_AD08	LSB multiplexed Address/Data Bus signal	EIM_DA8	ALT0	IO
EIM_AD09	LSB multiplexed Address/Data Bus signal	EIM_DA9	ALT0	IO
EIM_AD10	LSB multiplexed Address/Data Bus signal	EIM_DA10	ALT0	IO
EIM_AD11	LSB multiplexed Address/Data Bus signal	EIM_DA11	ALT0	IO
EIM_AD12	LSB multiplexed Address/Data Bus signal	EIM_DA12	ALT0	IO
EIM_AD13	LSB multiplexed Address/Data Bus signal	EIM_DA13	ALT0	IO
EIM_AD14	LSB multiplexed Address/Data Bus signal	EIM_DA14	ALT0	IO
EIM_AD15	LSB multiplexed Address/Data Bus signal	EIM_DA15	ALT0	IO
EIM_ADDR16	MSB Address Bus signal	EIM_A16	ALT0	O
EIM_ADDR17	MSB Address Bus signal	EIM_A17	ALT0	O
EIM_ADDR18	MSB Address Bus signal	EIM_A18	ALT0	O
EIM_ADDR19	MSB Address Bus signal	EIM_A19	ALT0	O
EIM_ADDR20	MSB Address Bus signal	EIM_A20	ALT0	O
EIM_ADDR21	MSB Address Bus signal	EIM_A21	ALT0	O
EIM_ADDR22	MSB Address Bus signal	EIM_A22	ALT0	O
EIM_ADDR23	MSB Address Bus signal	EIM_A23	ALT0	O
EIM_ADDR24	MSB Address Bus signal	EIM_A24	ALT0	O
EIM_ADDR25	MSB Address Bus signal	EIM_A25	ALT0	O
EIM_ADDR26	MSB Address Bus signal	NANDF_CS3	ALT3	O
EIM_BCLK	Burst Clock (BCLK). This active-high output signal is used to clock external burstcapable devices to synchronize the loading and incrementing of addresses and delivery of burst read and write data to/from the EIM. Its behavior is affected by the BCM field in the EIM_WCR and the SWR, SRD, BCD, and BCS fields of the EIM_CSxGCR1.	EIM_BCLK	ALT0	O
EIM_CRE	Used as CRE/PS for CellularRam memory. It is used for the Mode Register Set command. This signal can be configured as active low or active high. See CRE and CREP field descriptions of the EIM_CSxGCR1 registers.	NANDF_CS2	ALT3	O

Table continues on the next page...

Table 22-2. EIM External Signals (continued)

Signal	Description	Pad	Mode	Direction
EIM_CS0	Chip Selects. These signals are active-low. Behavior is affected by the RCSA and RCSN fields of the EIM_CSxRCR1 registers and the WCSA and WCSN fields of the EIM_CSxWCR1registers.	EIM_CS0	ALT0	O
EIM_CS1	Chip Selects. These signals are active-low. Behavior is affected by the RCSA and RCSN fields of the EIM_CSxRCR1 registers and the WCSA and WCSN fields of the EIM_CSxWCR1registers.	EIM_CS1	ALT0	O
EIM_CS2	Chip Selects. These signals are active-low. Behavior is affected by the RCSA and RCSN fields of the EIM_CSxRCR1 registers and the WCSA and WCSN fields of the EIM_CSxWCR1registers.	DISP0_DAT18	ALT7	O
		SD2_DAT1	ALT2	
EIM_CS3	Chip Selects. These signals are active-low. Behavior is affected by the RCSA and RCSN fields of the EIM_CSxRCR1 registers and the WCSA and WCSN fields of the EIM_CSxWCR1registers.	DISP0_DAT19	ALT7	O
		SD2_DAT2	ALT2	
EIM_DATA00	MSB Data Bus signal	CSI0_DATA_EN	ALT1	IO
EIM_DATA01	MSB Data Bus signal	CSI0_VSYNC	ALT1	IO
EIM_DATA02	MSB Data Bus signal	CSI0_DAT4	ALT1	IO
EIM_DATA03	MSB Data Bus signal	CSI0_DAT5	ALT1	IO
EIM_DATA04	MSB Data Bus signal	CSI0_DAT6	ALT1	IO
EIM_DATA05	MSB Data Bus signal	CSI0_DAT7	ALT1	IO
EIM_DATA06	MSB Data Bus signal	CSI0_DAT8	ALT1	IO
EIM_DATA07	MSB Data Bus signal	CSI0_DAT9	ALT1	IO
EIM_DATA08	MSB Data Bus signal	CSI0_DAT12	ALT1	IO
EIM_DATA09	MSB Data Bus signal	CSI0_DAT13	ALT1	IO
EIM_DATA10	MSB Data Bus signal	CSI0_DAT14	ALT1	IO
EIM_DATA11	MSB Data Bus signal	CSI0_DAT15	ALT1	IO
EIM_DATA12	MSB Data Bus signal	CSI0_DAT16	ALT1	IO
EIM_DATA13	MSB Data Bus signal	CSI0_DAT17	ALT1	IO
EIM_DATA14	MSB Data Bus signal	CSI0_DAT18	ALT1	IO
EIM_DATA15	MSB Data Bus signal	CSI0_DAT19	ALT1	IO
EIM_DATA16	MSB Data Bus signal	EIM_D16	ALT0	IO
EIM_DATA17	MSB Data Bus signal	EIM_D17	ALT0	IO
EIM_DATA18	MSB Data Bus signal	EIM_D18	ALT0	IO
EIM_DATA19	MSB Data Bus signal	EIM_D19	ALT0	IO
EIM_DATA20	MSB Data Bus signal	EIM_D20	ALT0	IO

Table continues on the next page...

Table 22-2. EIM External Signals (continued)

Signal	Description	Pad	Mode	Direction
EIM_DATA21	MSB Data Bus signal	EIM_D21	ALT0	IO
EIM_DATA22	MSB Data Bus signal	EIM_D22	ALT0	IO
EIM_DATA23	MSB Data Bus signal	EIM_D23	ALT0	IO
EIM_DATA24	MSB Data Bus signal	EIM_D24	ALT0	IO
EIM_DATA25	MSB Data Bus signal	EIM_D25	ALT0	IO
EIM_DATA26	MSB Data Bus signal	EIM_D26	ALT0	IO
EIM_DATA27	MSB Data Bus signal	EIM_D27	ALT0	IO
EIM_DATA28	MSB Data Bus signal	EIM_D28	ALT0	IO
EIM_DATA29	MSB Data Bus signal	EIM_D29	ALT0	IO
EIM_DATA30	MSB Data Bus signal	EIM_D30	ALT0	IO
EIM_DATA31	MSB Data Bus signal	EIM_D31	ALT0	IO
EIM_DTACK_B	Data Acknowledge, asynchronous access. This input is used as a data acknowledge signal for single asynchronous accesses.	EIM_WAIT	ALT1	I
EIM_EB0	Byte Enable. These active-low output signals indicate valid data bytes for the current access. They may be configured to assert for write cycles only. EIM_EB[0] corresponds to DATA_OUT[7:0] For asynchronous write accesses, behavior is affected by the WBEA and WBEN fields of the EIM_CS1WCR1-EIM_CS5WCR1 Registers. On synchronous or asynchronous read accesses, these signals are always asserted at the start of the access and negated at end of the access.	EIM_EB0	ALT0	O
EIM_EB1	Byte Enable. These active-low output signals indicate valid data bytes for the current access. They may be configured to assert for write cycles only. EIM_EB[1] corresponds to DATA_OUT[15:8] For asynchronous write accesses, behavior is affected by the WBEA and WBEN fields of the EIM_CS1WCR1-EIM_CS5WCR1 Registers. On synchronous or asynchronous read accesses, these signals are always asserted at the start of the access and negated at end of the access.	EIM_EB1	ALT0	O

Table continues on the next page...

Table 22-2. EIM External Signals (continued)

Signal	Description	Pad	Mode	Direction
EIM_EB2	Byte Enable. These active-low output signals indicate valid data bytes for the current access. They may be configured to assert for write cycles only. EIM_EB[2] corresponds to DATA_OUT[23:16] For asynchronous write accesses, behavior is affected by the WBEA and WBEN fields of the EIM_CS1WCR1-EIM_CS5WCR1 Registers. On synchronous or asynchronous read accesses, these signals are always asserted at the start of the access and negated at end of the access.	EIM_EB2	ALT0	O
EIM_EB3	Byte Enable. These active-low output signals indicate valid data bytes for the current access. They may be configured to assert for write cycles only. EIM_EB[3] corresponds to DATA_OUT[31:24]. For asynchronous write accesses, behavior is affected by the WBEA and WBEN fields of the EIM_CS1WCR1-EIM_CS5WCR1 Registers. On synchronous or asynchronous read accesses, these signals are always asserted at the start of the access and negated at end of the access.	EIM_EB3	ALT0	O
EIM_LBA	Address Valid. This active-low output signal is asserted during burst mode accesses to cause the external burst capable device to load a new starting burst address. Assertion of LBA indicates that a valid address is present on the address bus. Its behavior is affected by the SWR, SRD, BCD, and BCS fields of the EIM_CSxGCR1 registers, the RADVA and RADVN fields of the EIM_CSxRCR1 registers, and the WADVA and WADVN fields of the EIM_CSxWCR1 registers. In asynchronous mode, LBA length is affected by the RADVA, WADVA, RADVN, and WADVN fields. Minimum length of LBA signal in all modes is one EIM clock cycle.	EIM_LBA	ALT0	O

Table continues on the next page...

Table 22-2. EIM External Signals (continued)

Signal	Description	Pad	Mode	Direction
EIM_OE	Output Enable. This active-low output signal indicates the bus access is a read and enables external devices to drive the data bus with read data. Its behavior is affected by the OEA and OEN bit fields in the Chip Select Configuration Registers.	EIM_OE	ALT0	O
EIM_RW	Memory Write Enable. This active-low output signal indicates the bus access is a write and enables external devices to sample the data bus. Its behavior is affected by the WEA and WEN bit fields in the Chip Select Configuration Registers.	EIM_RW	ALT0	O
EIM_WAIT	Ready/Busy/Wait signal. This active-low input signal is asserted by external burst capable devices which support fixed or variable latency of data. It is serviced in synchronous mode only (EIM_CSxGCR1[SWR, SRD] =1). WAIT will have a pull up resistor in pad. The signal indicates whether the External device is ready for data transaction or not. Busy cycles (or wait cycles) of the external device can occur at the start of a Burst access or at page boundary crossover. **NOTE:** For burst devices, WAIT output should be configured to change one cycle before data is ready (before delay). **NOTE:** Some External devices may not use this input signal for ready state indication (fix latency without WAIT signal monitoring). For these devices EIM should be configured accordingly (see RFL, WFL, and PSZ field descriptions). **NOTE:** This is same as what is shown in IP_IND_WAIT_B	EIM_WAIT	ALT0	I

22.2.1 Other Important Block I/O Signals Internal to the SoC

The following table provides a description of other signals internal to the chip that are important in understanding the function of EIM.

Table 22-3. EIM Important Internal Signals

Name	I/O	Description
EIM_FB_BCLK	Input	Burst Clock Feedback. This block input is used to sample read data during high transfer speeds. The signal provides feedback from the I/O pad of the BCLK output pin and tends to align more closely with data from the external memory device.
EIM_BOOT	Input	EIM Boot Configuration. These block inputs determine the reset state of DSZ[1:0] and MUM. A more detailed description can be found in Boot Fusemap.
ACLK	Input	AXI clock, maximum frequency 133 Mhz
IPG_CLK_S	Input	EIM module IPG clock
RST_B	Input	Active low HW reset
EIM_WARM_RESET	Input	Warm Reset. If this signal is asserted the rst_b will reset only the internal FF and state machine while S/W registers will keep their current state. This signal is active high signal.

22.3 Clocks

The following table describes the clock sources for EIM. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 22-4. EIM Clocks

Clock name	Clock Root	Description
aclk	aclk_eim_slow_clk_root	EIM clock (main)
aclk_slow	aclk_eim_slow_clk_root	EIM clock (slow)
ipg_clk_s	ipg_clk_root	Peripheral access clock
aclk_exsc	aclk_eim_slow_clk_root	EIM clock (external device)

- ACLK: EIM clock (main clock, AXI clock) with a Max frequency of 133 MHz. Can be gated externally when there is no active AXI access.
- ACLK_SLOW: EIM all time running ACLK. Used for flip-flops that must be active even when EIM is in low power down mode to provide clock for lpack/lpmd registers, IP registers and IP to AXI sync registers.
- IPG_CLK_S: IPG clock for IP accesses. IP registers are activated by ACLK_SLOW clock.
- ACLK_EXSC: Clock created from EIM clock for External device usage. Integer division by 1, 2, 3 and 4 of the clock can be use with BCD bit field configuration, according to external devices demands. EIM clock frequency may be reduced for lower frequency support which cannot be achieved via BCD bit field.

22.4 Chip Select Memory Map

The EIM memory space is mapped into 128 MB total memory space in the processor memory. For addresses, see the System memory map table and the CM4 memory map table. The total 128 MB of memory can be divided among the EIM four chip selects. The memory configuration across the chip selects is controlled by the IOMUXC_GPR1 register. The ADDRSn[10] fields control how much memory is allotted to each chip select.

The following four configurations are supported:

- CS0 (128 MB), CS1 (0 MB), CS2 (0 MB), CS3 (0 MB) [default configuration]
- CS0 (64 MB), CS1 (64 MB), CS2 (0 MB), CS3 (0 MB)
- CS0 (64 MB), CS1 (32 MB), CS2 (32 MB), CS3 (0 MB)
- CS0 (32 MB), CS1 (32 MB), CS2 (32 MB), CS3 (32 MB)

22.5 Functional Description

This section provides the functional description for the EIM.

22.5.1 Continuous BCLK

EIM module can be programmed to provide a continuous BCLK in synchronous mode as the reference clock of PPGA. In this case, an internal DLL(Delay-locked Loop) is used to synchronize the internal clock which is used to sample read data from the I/O pad, with the feedback BCLK from PAD. The DLL compares the phase difference of internal clock to the feedback BCLK and will lock up when the delay of Reference Delay Line equals to the phase difference. After the lock up of Reference Delay Chain , DLL will copy the Master Delay Chain's value to Slave Delay Chain automatically. The Slave Delay Chain is used to delay the internal clock. If continuous BCLK is not selected, DLL should not be enabled.

When using an continuous BCLK in synchronous mode, BCD MUST be set to 0, otherwise will cause malfunction of EIM.

To let EIM work properly under continuous BCLK MODE, the initialization must follow below procedure. The initialize procedure should be done before any access to EIM. If it is not possible to do so, e.g. device booted up from EIM, the initialize flow should be done after performming a reset of EIM.

The recommended initialize flow is as follow

1. Disable EIM clock by clearing bit 4 of EIM_WIAR Register.
2. Select Continuous BCLK by setting bit 3 of EIM_WCR Register.
3. Enable DLL by setting bit 0 of EIM_DCR Register.
4. Enable EIM clock by setting bit 4 of EIM_WIAR Register.
5. Reset DLL by toggling bit 1 of EIM_DCR Register(1->0->1).
6. Wait for DLL lock (Both bit 0 and bit 1 of EIM_DSR Register are asserted).

After initialization, EIM can be programed to various access timing, but BCD MUST be kept to 0.

DLL Upadte Interval and inital value can be set by write EIM_DCR Register in corresponding bit field.

Also DLL provides manual adjustment to delay. An extra offset can be added to internal delay line, which can be used when DLL have some kind of predicable error. OVERRIDE bit field of EIM_DCR register will override reference DLL locked value and use a direct value for delay.

Note: During initialization, BCLK frequence and duty cycle are not guaranteed.

22.5.2 Bus Sizing Configuration

The EIM supports byte, half word and word operands allowing access to x8, x16, x32 ports. It can be address/data multiplexed in x16, x32 ports. The port size is programmable via the DSZ bit field in the corresponding Chip Select Configuration Register. An 8-bit port can reside in each one of the bytes of the data bus. A 16-bit port can reside on the lower 16 bits of the data bus, DATA_IN/OUT[15:0] or on the higher 16 bits of the data bus, DATA_IN/OUT[31:16].

In the case of a multi-cycle transfer, the lower two address bits (ADDR[1:0]) are incremented appropriately. The EIM address bus is configured according to DSZ bit field and AUS bits. There is either one bit (for x16 port size) or two bits (for x32 port size) right shift of the address bits (only when AUS=0) and no bit shift when AUS = 1 or DSZ[2] = 1.

The EIM has a data multiplexer which takes the four bytes of the AXI data bus and routes them to their required positions to properly interface to memory.

NOTE

A word access to or from a x16 port requires two external bus cycles to complete the transfer.

A word access to or from a x8 port requires four external bus cycles to complete the transfer.

22.5.2.1 8 BIT PORT SUPPORT

EIM has limited support for mot68000 & intel 386 protocols.

22.5.2.1.1 MOTOROLA 68000

EIM has limited support for mot68000 protocol. Only basic read or write asynchronous operations are supported.

The following operations are not supported:

- Read modify write
- Sync access
- All special accesses (Arm platform space, bus arbitration, bus control, bus error & reset operations)
- FC outputs

22.5.2.1.2 INTEL 386

EIM has limited support for intel 386 protocol. Only basic read or write async non-pipelined operations are supported.

The following operations are not supported:

- Other bus cycles (interrupt, halt & refresh)
- Bus lock
- M/IO, DC, LBA, NA, REFRESH & BS8 signals

22.5.3 EIM Operational Modes

Listed here are the main operational modes for EIM selected by control bit fields settings.

For details, see the bit field descriptions of SWR / SRD / MUM. All modes are supported in with 8-, 16- or 32-bit port configuration, according to DSZ bit field.

Table 22-5. EIM Operation Modes Field Settings

Control bit fields			Brief mode description
MUM	SRD	SWR	
0	0	0	Asynchronous write / Asynchronous read for APR=0 / Asynchronous page read for APR=1, none multiplexed
		1	Synchronous write/ Asynchronous read or APR=0 / Asynchronous page read for APR=1,none multiplexed
	1	0	Asynchronous write/Synchronous read none multiplexed
		1	Synchronous write/read none multiplexed
1	0	0	Asynchronous write/read multiplexed
		1	Synchronous write/ Asynchronous read multiplexed
	1	0	Asynchronous write/Synchronous read multiplexed
		1	Synchronous write/read multiplexed

22.5.4 Burst Mode (Synchronous) Memory Operation

This mode is enabled for read or write access. Bit SWR sets the burst mode for write operations at the corresponding chip select and bit SRD sets it for read operation.

When this mode is set, the controller attempts to translate the Master burst accesses to memory burst accesses, being limited by the memory burst length, predefined by BL value, or memory and Master WRAP/INCR boundary crossing non-matching. Only the first address accessed is put by the controller on the external address bus in a memory burst sequence.

EIM may translate from some Master sequential accesses to one or several memory bursts, but not from two Master individual accesses to one memory burst.

For the first access in a memory burst sequence, the EIM asserts $\overline{\text{ADV}}$, causing the external burst device to latch the starting burst address; then toggle the burst clock (BCLK) for a predefined number of cycles in order to latch the first unit of data. Subsequent accessed data units can then be burst in fewer clock cycles, realizing an overall increase in bus bandwidth.

NOTE

The BCLK signal toggles only when burst access is executed toward the external device (BCM=1'b0 for normal mode use). It runs with a 50% duty cycle until the end of access is reached. When access is terminated, BCLK stops toggling.

Memory burst accesses are terminated by the EIM whenever it detects the following:

- The specific burst length has executed completely (end of access)
- Write access - missing data in write buffer (Master is delaying the data transfer toward the EIM)
- Next sequential access crosses boundary with unequal condition (wrap/increment, burst length) on the Master and memory
- Current memory burst length reached

22.5.5 Burst Clock Divisor (BCD)

In some cases, it may be necessary to slow the external bus in relation to the internal bus to allow accesses to burst devices that have a maximum operating frequency less than the operating frequency of the internal bus.

The internal bus frequency can be divided by one, two, three or four for presentation on the external bus in burst mode operation.

BCLK can only be set to integer divisions of the incoming clock frequency. To get a specific frequency on BCLK, configure the divider to change the incoming EIM clock accordingly.

By programming the BCD bit field to various values, two signals on the external bus are affected; $\overline{\text{ADV}}$ and BCLK. The $\overline{\text{ADV}}$ signal is asserted according to RADVA or WADVA bit fields programming, and is negated according to the formula mentioned in RADVN and WADVN bit fields description. The BCLK signal runs with a 50% duty cycle until the end of access is reached.

If BCM = 1, the BCLK runs at frequency according to GBCD bit field settings on every async memory access, regardless of the SWR and SRD bits configuration. Caution should be exercised when using BCM bit; GBCD bit field should be updated once and should not change when BCLK is toggling. The BCM bit is used mainly for system debug mode. It has no functional use of the EIM in normal mode.

22.5.6 Burst Clock Start (BCS)

In an effort to allow greater flexibility in achieving the minimum number of wait states on burst accesses, you can determine when you want the BCLK to start toggling after the start of access. This allows the BCLK to be skewed from point of data capture on the EIM clock by any number of EIM clock cycles.

Care must be exercised when setting BCS bit field in conjunction with the BCD and RWSC/WWSC bit fields. See the external timing diagrams in Burst (Synchronous Mode) Read Memory Accesses Timing Diagram - BCD=1 and Burst (Synchronous Mode) Read Memory Accesses Timing Diagram - BCD=0 for examples of how to use the BCS, BCD and RWSC/WWSC bit fields together.

22.5.7 Multiplexed Address/Data Mode Support

The control bit MUM allows support memory with multiplexed address/data bus both in asynchronous and in synchronous modes.

Caution should be exercised for using OEA/WEA & ADH bit fields. They should be configured according to the external device requirements, as it determines the time point of end of address phase and start of data phase.

22.5.8 Mixed Master/Memory Burst Modes Support

To provide mixed sequential/wrap accesses with different length, EIM interprets burst signal and generate additional $\overline{\text{ADV}}$ signals whenever there appear unequal address or burst boundary crossing condition.

BL bit field is used to notify EIM about current memory burst and wrap condition for properly external address generation. In case of non-matching boundaries in both the memory and Master access, EIM starts a new memory burst access by updating address from Master on address bus and generating $\overline{\text{ADV}}$ signal.

22.5.9 AXI (Master) Bus Cycles Support

The EIM uses an Arm AXI slave interface. It has a 32-bit bus and supports one access (one ID) at a time. No out of order or parallel accesses are supported.

The following AXI protocol signals are not supported:

- AWLOCK
- AWCACHE
- ARLOCK
- ARCACHE

ARID bus is sampled when:

- new read access is valid on the read address channel and is reflected on the RID bus output toward the master.

AWID bus is sampled when:

- new write access is valid on the write address channel and is reflected on the WID/BID bus output toward the master.

ARPROT and AWPROT signal are partially used. ARPROT[0] and AWPROT[0] bits are used for normal/privileged access detection. ARPROT[2:1] and AWPROT[2:1] are not used.

When sampling a valid access on both of the address channels, the read access will be performed first while write access is pending. After last data transfer completed, the pending write will be executed.

A new access may be executed one cycle after sampling a valid access on the read or write address channels, assuming there is no current access (back to back) which can cause a recovery or end of access penalty cycles, for write access, also assuming data is in write buffer for fast execution.

NOTE

- Only 32-bit word size accesses are supported for burst mode accesses.
- Only 8-bit (1 byte), 16-bit (2 byte) and 32-bit (4 byte) word size supported for single access.
- Maximum number of burst length is 16.
- According to AXI protocol, burst access should not cross 4 KB blocks. In case EIM gets an access that crosses the 4 KB, memory address calculation is invalid.

AXI transfers shown in the table below are also supported. These AXI cycles will be translated into the necessary cycles on the memory side. For example, for optimal operation in case Arm cache is configured to 8 beat burst with wrap, a synchronous flash and cellular RAM memory should be configured in 16 word wrap burst mode when using a 16-bit data port, and in 8 word wrap burst mode when using a 32-bit data port. EIM uses BL bit field to support different memory configurations. The controller splits the transaction when needed in some cases. See Table 22-7.

Table 22-6. AXI Burst Cycles Supported

Burst Length - Number of data transfers	Burst size - Bytes in transfer	Burst type	Description
1	1	INCR	Single transfer

Table continues on the next page...

Table 22-6. AXI Burst Cycles Supported (continued)

Burst Length - Number of data transfers	Burst size - Bytes in transfer	Burst type	Description
1	2	INCR	Single transfer
1	4	INCR	Single transfer
2	4	WRAP	2-beat wrapping burst
4	4	WRAP	4-beat wrapping burst
8	4	WRAP	8-beat wrapping burst
16	4	WRAP	16-beat wrapping burst
2	4	INCR	2-beat incrementing burst
3	4	INCR	3-beat incrementing burst
4	4	INCR	4-beat incrementing burst
5	4	INCR	5-beat incrementing burst
6	4	INCR	6-beat incrementing burst
7	4	INCR	7-beat incrementing burst
8	4	INCR	8-beat incrementing burst
9	4	INCR	9-beat incrementing burst
10	4	INCR	10-beat incrementing burst
11	4	INCR	11-beat incrementing burst
12	4	INCR	12-beat incrementing burst
13	4	INCR	13-beat incrementing burst
14	4	INCR	14-beat incrementing burst
15	4	INCR	15-beat incrementing burst
16	4	INCR	16-beat incrementing burst

Table 22-7. AXI to Memory Burst Splits Number

AXI Burst Type	Memory Burst Type Config.	# of accesses to X8 Memory Port size	# of accesses to X16 Memory Port size	# of accesses to X32 Memory Port size
INC16 Aligned Addr.	WRAP4	16	8	4
	Cont.	1	1	1
INC16 Unaligned Addr.	WRAP4	17	9	5
	Cont.	1	1	1
WRAP16 Aligned Addr.	WRAP16	4	2	1
	Cont.	1	1	1
WRAP16 Unaligned Addr.	WRAP16	5	3	1
	Cont.	2	2	2
INC8 Aligned Addr.	WRAP8	4	2	1
	WRAP16	2	1	1
INC8	WRAP8	4 or 5	2 or 3	2

Table continues on the next page...

Table 22-7. AXI to Memory Burst Splits Number (continued)

AXI Burst Type	Memory Burst Type Config.	# of accesses to X8 Memory Port size	# of accesses to X16 Memory Port size	# of accesses to X32 Memory Port size
Unaligned Addr.	WRAP16	2 or 3	2	1 or 2
WRAP8	WRAP16	2	1	1
Aligned Addr.	Cont.	1	1	1
WRAP8 Unaligned Addr.	WRAP16	2 or 3	1	2
	Cont.	2	2	2

22.5.10 WAIT_B Signal, RWSC and WWSC bit fields Usage

Most of the external devices supporting burst mode for write or read accesses provide a signal which indicates data is valid on the memory bus (a.k.a. handshake mode). For this mode, RFL and WFL bits should be cleared and RWSC/ WWSC bit fields indicate when the controller should start sampling this signal from the external device or, in other words, how many BCLK cycles should be masked.

For devices which do not use this signal or have a fixed latency ability, the RFL and WFL bits may be set for internal calculation regarding BCLK cycles penalty until data is valid (memory initial access time). For this mode, RWSC/ WWSC indicates when the data is ready for sampling by the controller (read access) or the external device (write access). There is separation between read and write accesses wait-state control. For read access, RWSC bit field is valid and WWSC bit field is ignored; for write access, WWSC is valid and RWSC is ignored.

22.5.11 IPS Register Interface

Access to the registers of the EIM, read or write, is made with IPS protocol signals. The system should avoid changing the registers while master/memory transaction is valid, as this can cause an unknown behavior of the controller.

Register access size is 32-bit as the register size definition, other size of access (byte or half word) is not supported.

22.5.12 MRS Set for PSRAM

Memory registers of PSRAM devices can be configured according to external signal, which indicates whether the access is to a memory array or memory register domain.

When the CRE bit is set, the following transactions to the external device will assert the CRE signal. The polarity of this signal is determined by the CREP bit for active low or active high assertion of the signal.

22.5.13 EIM Access Termination

EIM is monitoring the corresponding CSx control signal every time variable latency access or dtack access is performed toward the external device.

In variable latency accesses, the Watchdog Timer (WDOG-1) counts BCLK cycles. If it reaches the wdog_limit (according to the WDOG_LIMIT bit field in the WCR) before the device signals can drive/sample new data, the controller will terminate the access and generate an error response transfer toward the Master.

In dtack access, WDOG-1 counts ACLK cycles instead of BCLK and it reaches the wdog_limit before the device asserts the dtack signal, the controller will terminate the access and generate an error response transfer toward the Master.

WDOG-1 can by disabled by WDOG_EN bit in the WCR.

22.5.14 Error Conditions

The following conditions cause an error (AXI error or IPS error) response signal:

- AXI errors
  - Access to a disabled chip select - access to a mapped chip select address space where the CSEN bit in the corresponding chip select Configuration Register is clear
  - Access to a non mapped address - access to an address that is not mapped to any CS.
  - User access to a supervisor-protected chip select address space (the SP bit in the corresponding chip select Configuration Register is set)
  - User access in fixed mode access
  - User performs write access to write protected chip select
  - First write data ID and write address ID do not match. (No data is written to the memory.)

    - First Write Data ID and write address ID match but one or more of the other Write data IDs does not match the First Write data ID (data is written to memory according)
    - Access duration to external device from CSx signal assertion is 128/256/512/1024 cycles (access is terminated by the controller) - This error can be disabled be software.
- IPS errors
    - User read or write access to a reserved/non-valid address in the EIM Configuration Register

22.5.15 DTACK Mode

In DTACK mode, the EIM uses DTACK signal as an indication of when to end the access.

DTACK is an asynchronous edge/level sensitive signal. DTACK polarity is configurable by the DAP bit in CsxGCR2 (default value is 0).

In this case, EIM begins the access and after a few cycles (according DAPS field) and waits until DTACK (after synchronization) becomes asserted, then samples the data in read access and completes the current data access (see Figure 22-15, Figure 22-16 & Figure 22-17).

If more than one data is needed, CS will be negated between access (CSREC field is not zero) and the AXI burst access will be split into single accesses (see Figure 22-19).

22.5.16 RDY_INT Signal as Interrupt

The EIM has an external interrupt support. When INTEN bit in the WCR is set, signal RDY_INT is used as interrupt; its status is being reflected by INT bit and output signal.

It is cleared by writing one to the INT bit. When INTEN is cleared, the interrupt is disabled. This interrupt is a level interrupt and its polarity can be configured by the INTPOL bit in the WCR.

22.5.17 RDY_INT Signal as Ready After Reset Indication

This feature is used for boot propose from external devices based on NANDFlash array memory with NORFlash interface.

When ERRST bit is set, RDY_INT signal is monitored to determine ready after reset of the external device located on CS0.

The monitoring is taking place when CS0 is accessed for the first time. The access will be pending until assertion of the signal is detected. When detection occurs, ERRST bit is self-cleared and pending access is executed to the external device on CS0.

22.5.18 EIM_GRANT / EIM_BUSY Handshake Description

Prior to executing command to one of the external device (chip select), EIM assert EIM_BUSY signal (1'b1) and checks the EIM_GRANT signal status.

If EIM_GRANT signal is high, it indicates external data bus is not used by other slaves (NAND Flash Controller) and EIM may start to execute the access. If EIM_GRANT is low, EIM waits until it is set (1'b1) before executing the access.

EIM keeps EIM_BUSY signal set until it completes the access toward the external device.

Once EIM_GRANT signal is set, it can not be reset until EIM_BUSY signal is cleared by EIM.

NOTE

In 16-bit Muxed EIM doesn't use the data bus, therefore there is no sharing of the data bus with NFC. EIM doesn't wait for EIM_GRANT signal from NFC and doesn't assert the EIM_BUSY signal.

22.5.19 LPMD / LPACK Handshake Description

These signals are used for frequency and/or voltage change, and for entering low power mode during normal operation of the EIM. Before any change can take place, the controller and all the relevant external devices should be in idle state, which means no access or data transfer is in process.

LPMD input signal is asserted once EIM detects the assertion of LPMD, all ready signals of the AXI channels are negated, and EIM is not sampling new accesses. It finishes all the ongoing accesses and already pending ones. When EIM is in idle state, the LPACK output signal is asserted. EIM will stay in idle state and the LPACK signal will stay asserted until the LPMD signal is negated.

22.5.20 Endianness

Big and Little endianness are supported by the controller according to the following table.

Table 22-8. EIM Out/in Data in Case AXI Out/in Data is 0xB3B2B1B0

Endian mode	AXI access	AXI address [1:0]	Port size and used bits								
			Word port				Half word port			Byte port	
			[31:24]	[23:16]	[15:8]	[7:0]	External address [0]	[31:24] ([15:8])	[23:16] ([7:0])	External address [1:0]	[31:24] ([23:16]) ([15:8]) ([7:0])
Big	Word	0	0xB3	0xB2	0xB1	0xB0	0	0xB3	0xB2	0	0xB3
										1	0xB2
							1	0xB1	0xB0	2	0xB1
										3	0xB0
	Half Word	0			0xB1	0xB0	0	0xB3	0xB2	0	0xB3
										1	0xB2
		2	0xB3	0xB2			1	0xB1	0xB0	2	0xB1
										3	0xB0
	Byte	0				0xB0	0		0xB3	0	0xB3
		1			0xB1			0xB2		1	0xB2
		2		0xB2			1		0xB1	2	0xB1
		3	0xB3					0xB0		3	0xB0
Little	Word	0	0xB3	0xB2	0xB1	0xB0	0	0xB1	0xB0	0	0xB0
										1	0xB1
							1	0xB3	0xB2	2	0xB2
										3	0xB3
	Half Word	0			0xB1	0xB0	0	0xB1	0xB0	0	0xB0
										1	0xB1
		2	0xB3	0xB2			1	0xB3	0xB2	2	0xB2
										3	0xB3
	Byte	0				0xB0	0		0xB0	0	0xB0
		1			0xB1			0xB1		1	0xB1
		2		0xB2			1		0xB2	2	0xB2
		3	0xB3					0xB3		3	0xB3

22.5.21 Strobe Signal Use

The strobe signal is toggling according to address/data valid condition on the external bus for read and write accesses, and for both synchronous and asynchronous modes.

At any time point when address/data is valid on the external bus, the strobe signal will generate a positive edge, which can be used to sample the external data and control signal.

NOTE

Strobe signal for read data is active (RL + 1) cycles after data on external bus is valid.

22.6 Initialization Information

22.6.1 Booting from EIM

EIM is ready to work with CS0 after the hardware reset, but it has been configured for very slow access (for boot purposes), with additional setup and hold time.

Other CSs are disabled by hardware reset. Therefore, all CSs must be properly initialized before use in writing values to the corresponding chip select configuration registers.

DSZ[1:0] and MUM fields are set according to EIM_BOOT [2:0] block inputs.

22.7 Typical Application

Application note uses following functions to illustrate EIM and memory accesses:

- WR16(address, data) is a 16 bit write access
- WR32(address, data) is a 32 bit write access
- RD16(address, data) is a 16 bit read access
- RD32(address, data) is a 32 bit read access
- WR_I(address, data, delta, counter) is a write data sequence, there data(i+1) = data(i) + delta
- COMMAND_SEQUENCE
- CHECK_STATUS

NOTE

COMMAND_SEQUENCE and CHECK_STATUS are described in AMD Flash Utility, Intel Sibley Flash Utility, MDOC Device Utility, Samsung OneNAND Utility, and Spansion Flash Utility.

All addresses are byte addresses. "CS0" is a Chip Select 0 base address. "EIM_" is a prefix of EIM's registers. 'h is a prefix of hexadecimal constant. "//" is a comment beginning. csba[cs] is a dimension of CS base addresses. "addr" means an address offset in current CS address space. Examples use CS0 address space, but it may apply to any CS except for boot mode functionality.

Configuration examples were verfied with the memory models listed below and may require some adjustments for other family members.

22.7.1 Access to Intel Sibley Flash

The following configurations are intended to Sibley family muxed and non-muxed devices.

22.7.1.1 Intel Sibley Flash Asynchronous Mode Configuration

- WR32('EIM_CS0GCR1,'h00210081);
- WR32('EIM_CS0RCR1,'h0e020000);
- WR32('EIM_CS0RCR2,'h00000000);
- WR32('EIM_CS0WCR1,'h0704a040);

22.7.1.2 Intel Sibley Flash Synchronous Mode Configuration

Configuration used for 133 MHz synchronous access to flash:

```
// Set memory to synchronous read mode
     WR16('CS0+('h5903<<1),'h0060);
     WR16('CS0+('h5903<<1),'h0003);
     WR16('CS0+('h0000<<1),'h00ff);
// Set EIM configuration to synchronous timing
     WR32('EIM_CS0GCR1,'h50214225);                    // 133 MHz
     WR32('EIM_CS0RCR1,'h0c000000);                    // 12 cycles on memory
```

Configuration used for 66 MHz synchronous access to muxed flash:

```
// Set memory to synchronous read mode
     WR16('CS0+('h3103<<1),'h0060);
     WR16('CS0+('h3103<<1),'h0003);
     WR16('CS0+('h0000<<1),'h00ff);
//-------------------------------------------------------------------------
// Set EIM configuration to synchronous timing
     WR32('EIM_CS0GCR1,'h5021122d);                    // 66 MHz
     WR32('EIM_CS0RCR1,'h07000000);                    // 7cycles on memory
```

22.7.1.3 Intel Sibley Flash Utility

```
// Single data word programming to addr
     WR16('CS0+addr,'h0060);                 // Unlock
     WR16('CS0+addr,'h00d0);
     WR16('CS0+addr, 'h0041);
     WR16('CS0+addr, data);
     WR16('CS0+caddr, 'h0070);                 // Read Status command
     while('CS0+data[7] == 0)                                  // Wait / Polling
         RD16('CS0+addr, data);              // Read status
     RD16('CS0+addr, data);                  // Read status
     WR16('CS0+'h0000,'h00ff);
// Write buffer programming
     WR16('CS0+addr,'h0060);                 // Unlock
     WR16('CS0+addr,'h00d0);
     data    = 0;
     WR16('CS0+addr, 'h0070);                // Read Status command
     while(data[7] == 0)                             // Wait
         RD16('CS0+addr, data);              // Read status
     WR16('CS0+'h0000,'h00ff);
     WR16('CS0+addr,'h00e9);           // Write Buffer command
     WR16('CS0+addr,255);              // Word counter (<256)
     for(i=0; i<'h200; i = i + 'h40)
        WR_I('CS0+addr+i, data+((i>2)*'h0010_0001), 'h0010_0001, 16); // Data
     WR16('CS0+addr,'h00d0);           // Write Confirm command
     data  = 0;
     while(data[7] == 0)                      // Wait
        RD16('CS0+addr, data);         // Read status
     RD16('CS0+addr, data);            // Read status
     WR16('CS0+'h0000,'h00ff);
```

22.7.2 Access to MDOC Device

The following configurations are intended to MDOC H3 device.

22.7.2.1 MDOC Device Boot

To boot from the MDOC device the ERRST bit should be configured to 1, so that EIM will hold the first read access to CS0 until the MDOC asserts the RDY signal.

22.7.2.2 MDOC Device Asynchronous Mode Configuration

```
// Non-muxed mode
WR32('EIM_CS0GCR1,'h00410081);
WR32('EIM_CS0RCR1,'h0e121010);
WR32('EIM_CS0RCR2,'h00000000);
WR32('EIM_CS0WCR1,'h12092492);
// Muxed mode
WR32('EIM_CS0GCR1,'h00410081);
WR32('EIM_CS0RCR1,'h0e121010);
WR32('EIM_CS0RCR2,'h00000000);
WR32('EIM_CS0WCR1,'h12092492);
```

22.7.2.3 MDOC Device Utility

```
// Read Manufacturer ID and Device ID
    RE16('CS0+'h9400,'h4833);
    RE16('CS0+'h9422,'hb7cc);
```

22.7.3 Access to Micron PSRAM

The following configurations are intended to mt45w4mw16bfb_706.

22.7.3.1 Micron PSRAM Asynchronous Mode Configuration

```
// 16 bit memory
    WR32('EIM_CS0GCR1,'h403104b1);
    WR32('EIM_CS0RCR1,'h0b010000);
    WR32('EIM_CS0RCR2,'h00000008);
    WR32('EIM_CS0WCR1,'h0b040040);
// 32 bit memory
    WR32('EIM_CS0GCR1,'h403304b1);
    WR32('EIM_CS0RCR1,'h0f010000);
    WR32('EIM_CS0RCR2,'h00000008);
    WR32('EIM_CS0WCR1,'h0f040040);
```

22.7.3.2 Micron PSRAM Synchronous Mode Configuration

```
// 16 bit memory
    WR32('EIM_CS0GCR1,'h403104b1);
    WR32('EIM_CS0WCR1,'h0b040000);
    WR16('CS0+('h85947<<1),'h0040); // memory configuration
    WR32('EIM_CS0GCR1,'h4021_5487);// fixed latency memory wrap 4
    WR32('EIM_CS0RCR1,'h04000000);
    WR32('EIM_CS0RCR2,'h00000008);
    WR32('EIM_CS0WCR1,'h04000000);
// 32 bit memory
    WR32('EIM_CS0GCR1,'h6003_04f1);
    WR32('EIM_CS0WCR1,'h0b04_0000);
    WR32('CS0+('h85947<<2),'h0040); // memory configuration
    WR32('EIM_CS0GCR1,'h4003_1487); // var latency memory inc. page size 128
    WR32('EIM_CS0RCR1,'h04000000);
    WR32('EIM_CS0RCR2,'h00000008);
    WR32('EIM_CS0WCR1,'h04000000);
```

22.7.4 Access to Samsung OneNAND

Mentioned below are the configurations intended for Samsung OneNAND muxed and non-muxed devices.

22.7.4.1 Samsung OneNAND Boot

There are two ways to boot from Samsung OneNAND. In the first way, the ERRST bit is set to 0 and the user has to poll the interrupt status in the OneNAND interrupt register (or set interrupt handler there). In the second way, the ERRST bit is set to 1 and the user should enable the device interrupt output before the first read from CS0 access is issued.

Load sectors 2,3 to DataRAM, page 0 done in the next example:

- WR16('CS0+('hF241<<1),'h0); // Clear interrupt status
- WR16('CS0+('hF100<<1),'h0); // block[8:0] address
- WR16('CS0+('hF107<<1),'h2); // sector[1:0] and page[7:2] addresses
- WR16('CS0+('hF200<<1),'h802); // buffer[11:8] address and counter[1:0]
- WR16('CS0+('hF101<<1),'h0); // DDP choose
- WR16('CS0+('hF220<<1),'h0); // Set command

22.7.4.2 Samsung OneNAND Asynchronous Mode Configuration

```
// Non-muxed memory
    WR32('EIM_CS0GCR1,'h00410081);
    WR32('EIM_CS0RCR1,'h0b010000);
    WR32('EIM_CS0RCR2,'h00000000);
    WR32('EIM_CS0WCR1,'h0c092480);
// Muxed memory
    WR32('EIM_CS0GCR1,'h00410089);
    WR32('EIM_CS0RCR1,'h0b010000);
    WR32('EIM_CS0RCR2,'h00000000);
    WR32('EIM_CS0WCR1,'h0c092480);
```

22.7.4.3 Samsung OneNAND Synchronous Mode Configuration

Set memory and EIM to synchronous read mode is shown in the next example:

```
    WR16('CS0+('hF221<<1),'hc0e0); // Synchronous read, 4 clk latency
    WR32('EIM_CS0GCR1,'h50412405); // 44 MHz (non-muxed)
    WR32('EIM_CS0RCR1,'h05010000);
```

The muxed Samsung OneNAND supports synchronous write, too:

```
// Set memory & EIM to synchronous read and write mode
    WR16('CS0+('hF221<<1),'hc0f2); // Sync. read and write, 4 clk latency
    WR32('EIM_CS0GCR1,'h5041240f); // 44 MHz
    WR32('EIM_CS0RCR1,'h05010000);
    WR32('EIM_CS0WCR1,'h05040000);
```

22.7.4.4 Samsung OneNAND Utility

The following utility algorithms are used on the Samsung OneNAND:

```
// Unlock Block command
    WR16('CS0+('hF100<<1),'h0);        // DFS
    WR16('CS0+('hF100<<1),'h0);        // DBS
    WR16('CS0+('hF24c<<1),'h2);        // SBA - block number (2)
    WR16('CS0+('hF241<<1),'h0);        // Clear interrupt status
    WR16('CS0+('hF220<<1),'h23);       // Unlock command
    data ='h0;
    while(!(data &'h0004))             // Polling
       RD32('WIAR, data); // Read status
// Erase block command
    WR16('CS0+('hF100<<1),'h2);        // DFS and block ([8:0]) address
    WR16('CS0+('hF101<<1),'h0);        // DBS
    WR16('CS0+('hF241<<1),'h0);        // Clear interrupt status
              WR16('CS0+('hF220<<1),'h94);            // Erase command
    data ='h0;
    while(!(data &'h0004))                    // Wait
        RD32('WIAR, data); // Read status
// Program page command
    WR16('CS0+('hF100<<1),'h2);             // DFS and block[8:0] address
    WR16('CS0+('hF107<<1),'h0);             // sector[1:0] and page[7:2] addresses
    WR16('CS0+('hF200<<1),'h800);           // buffer[11:8] address and counter[1:0]
    WR16('CS0+('hF241<<1),'h0);             // Clear interrupt status
    WR16('CS0+('hF220<<1),'h80);            // Program command
    data ='h0;
    while(!(data &'h0004))                  // Wait
        RD32('WIAR, data); // Read status
```

22.7.5 Access to Samsung UtRAM

Below mentioned configurations are intended for Samsung UtRAM.

22.7.5.1 Samsung UtRAM Asynchronous Mode Configuration

```
WR32('EIM_CS0GCR1,'h400104b1);
WR32('EIM_CS0RCR1,'h0a010000);
WR32('EIM_CS0RCR2,'h00000008);
WR32('EIM_CS0WCR1,'h0b040040);
```

22.7.5.2 Samsung UtRAM Synchronous Mode Configuration

```
RD16('CS0+('hff_ffff<<1),data);             // command sequence
RD16('CS0+('hff_ffff<<1),data);
RD16('CS0+('hff_ffff<<1),data);
RD16('CS0+('hff_feff<<1),data);
RD16('CS0+('h00_82a0<<1),data);             // memory sync. configuration
WR32('EIM_CS0GCR1,'h4021_53b7);            // fixed latency memory wrap 32
WR32('EIM_CS0RCR1,'h0500_0000);
WR32('EIM_CS0WCR1,'h0300_0000);
```

22.7.6 Access to Spansion Flash

Below mentioned configurations are intended for Spansion Flash.

22.7.6.1 Spansion Flash Asynchronous Mode Configuration

```
           WR32('EIM_CS0GCR1,'h00410081);
           WR32('EIM_CS0RCR1,'h0a018000);
           WR32('EIM_CS0RCR2,'h00000000);
           WR32('EIM_CS0WCR1,'h0704a240);
WR16('CS0+('hF220<<1),'h94);            // Erase command
           data ='h0;
           while(!(data &'h0004))                   // Wait
               RD32('WIAR, data); // Read status
      // Program page command
           WR16('CS0+('hF100<<1),'h2);              // DFS and block[8:0] address
           WR16('CS0+('hF107<<1),'h0);              // sector[1:0] and page[7:2] addresses
           WR16('CS0+('hF200<<1),'h800);            // buffer[11:8] address and counter[1:0]
           WR16('CS0+('hF241<<1),'h0);              // Clear interrupt status
           WR16('CS0+('hF220<<1),'h80);             // Program command
           data ='h0;
           while(!(data &'h0004))                   // Wait
               RD32('WIAR, data); // Read status
```

22.7.6.2 Spansion Flash Synchronous Mode Configuration

```
           WR16('CS0+('h0555<<1),'h00aa);   // command sequence
           WR16('CS0+('h02aa<<1),'h0055);
           WR16('CS0+('h0555<<1),'hd0);
           WR16('CS0+('h0000<<1),'h1ec4);   // memory sync. configuration
           WR32('EIM_CS0GCR1,'h50411325); // 66 MHz
           WR32('EIM_CS0RCR1,'h05000000); // 5 cycles on memory
```

22.7.6.3 Spansion Flash Utility

```
       // Single word programming
           COMMAND_SEQUENCE(cs,16,'ha0);            // single word programming
           WR16('CS0+addr, data);
           CHECK_STATUS('CS0+addr,data,16,1,errst);
       // Write buffer programming
           COMMAND_SEQUENCE(0,16,'h25);               // write buffer programming
           WR16('CS0+addr,'h001f);                      // counter-1
           WR_I('CS0+addr, data, 'h0010_0001, 16); // data
           WR16('CS0+addr,'h0029);                       // write buffer to flash
           CHECK_STATUS('CS0+addr+'h3e,data[31:16]+'h00f0,16,1,errst);
```

There COMMAND_SEQUENCE and CHECK_STATUS are next functions:

```
       task COMMAND_SEQUENCE;
           input[2:0]    cs;
           input[7:0]    port_size;
           input[31:0]   code;
       begin
           if(port_size == 16)
              begin
              WR16(csba[cs]+('h0555<<1),'h00aa);
              WR16(csba[cs]+('h02aa<<1),'h0055);
              WR16(csba[cs]+('h0555<<1),code);
              end
           else
              begin
              WR32(csba[cs]+('h0555<<2),'h00aa);
```

```
             WR32(csba[cs]+('h02aa<<2),'h0055);
             WR32(csba[cs]+('h0555<<2),code);
             end
       end
       endtask
       task     CHECK_STATUS;
           input[31:0] addr;
           input[31:0] edata;
           input[7:0]   port_size;
           input[7:0]   opcode;
           output[7:0] errst;
           reg[31:0]    data;
           reg[31:0]    data3;
       begin
           errst = 0;
           data  = 0;
           data3 = 0;
while(!(data == edata) && !errst) // Wait operation
         begin: BR_EN
         RD16(addr, data);             // Read status
         if(data[7] !== edata[7])
             begin
             if(data[5] == 1)
                  begin
                  RD16(addr, data3);
                  RD16(addr, data);
                  if(data[6] !== data3[6])
                       begin
                       $display("CHECK_STATUS: Error timout on single data program");
                       errst = 1;
                       disable BR_EN;
                       end
                  end
             else
                  begin
                  if(opcode == 2)
                       if(data[1] == 1)
                           begin
                           RD16(addr, data3);
                           if(port_size == 32)
                              RD32(addr, data);
                           else
                              RD16(addr, data);
                           if(data[1] == 1 && data !== edata)
                              begin
                              $display("CHECK_STATUS: Error on write buffer");
                              errst =3;
                              disable BR_EN;
                              end
                           end
                  end
             end
         else
             begin
             RD16(addr, data3);
             if(port_size == 32)
                  RD32(addr, data);
             else
                  begin
                  RD16(addr, data);
                  edata[31:16] = 16'h0;
                  end
             if(data !== edata)
                  begin
                  $display("CHECK_STATUS: Error in data write on single data program");
                  errst =2;
                  disable BR_EN;
                  end
             end
         end
```

```
end
endtask
```

22.7.7 8 bit support

This section details the pin connections for Intel mode and Motorola mode.

Intel Mode - For intel mode use the following connection:

Table 22-9. Intel Mode pin connections

Arm platform Pin	EIM Pin	Notes
ADS#	IPP_DO_ADV_B	WAL = 1,RAL = 1
W/R	IPP_DO_BE_B	WBED = 1
WR#	WE#	
RD#	OE#	

Mot. Mode - For intel mode use the following connection:

Table 22-10. Motorola Mode pin connections

Arm platform Pin	EIM Pin	Notes
AS#	IPP_DO_CS_B	
R/W#	WE#	
LDS#	BE#	

22.8 External Bus Timing Diagrams

The following timing diagrams show the timing of accesses to memory or a peripheral with different timing parameters. All examples done for CS0, but are valid for any others chip select. BE means one from current used BE[3:0].

22.8.1 Asynchronous Read Memory Accesses Timing Diagram

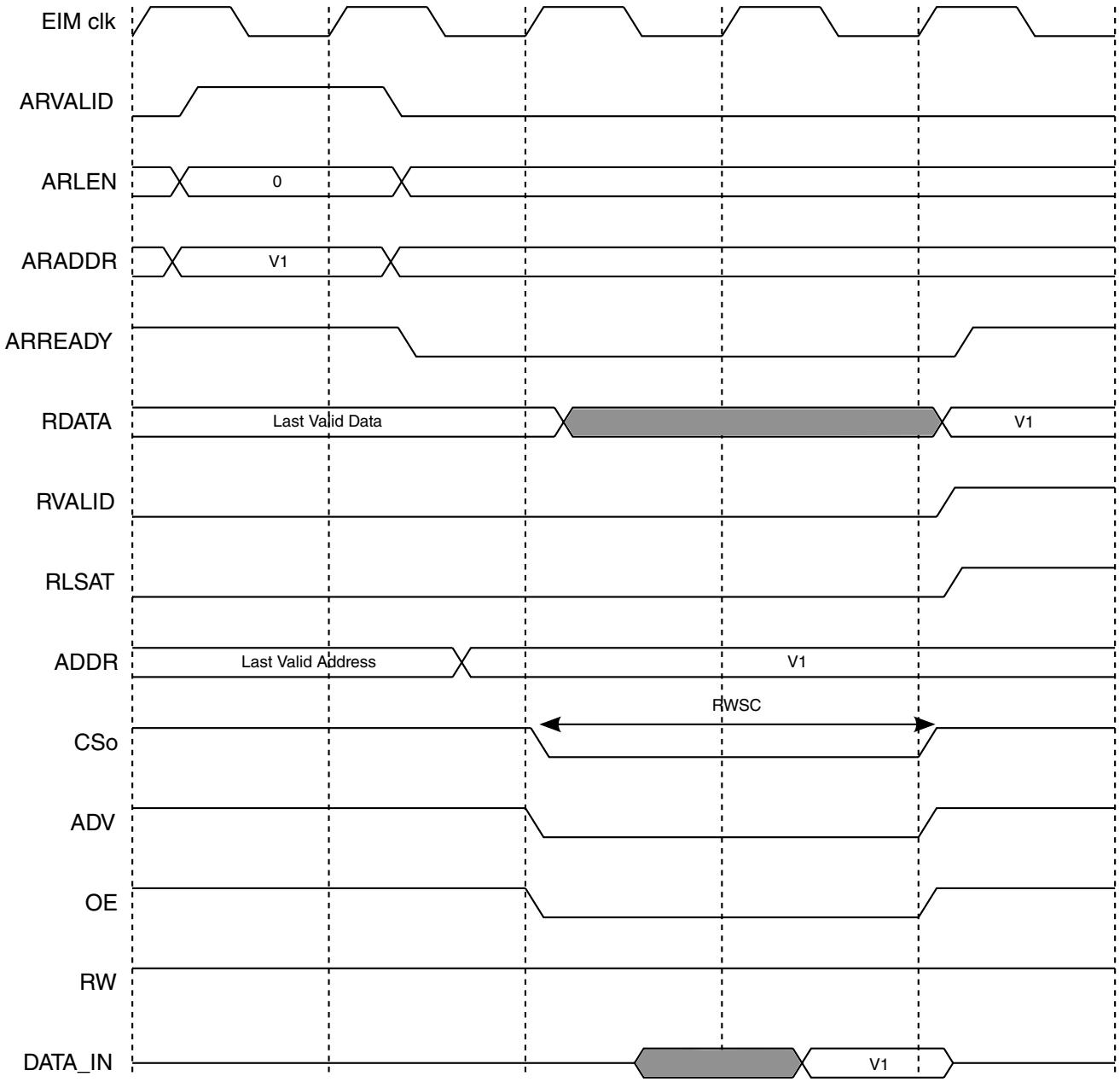


Figure 22-2. Read Access, RWSC=2,RCSA=0,OEA=0,RCSN=0,OEN=0, RAL=1

22.8.2 Asynchronous Write Memory Accesses Timing Diagram

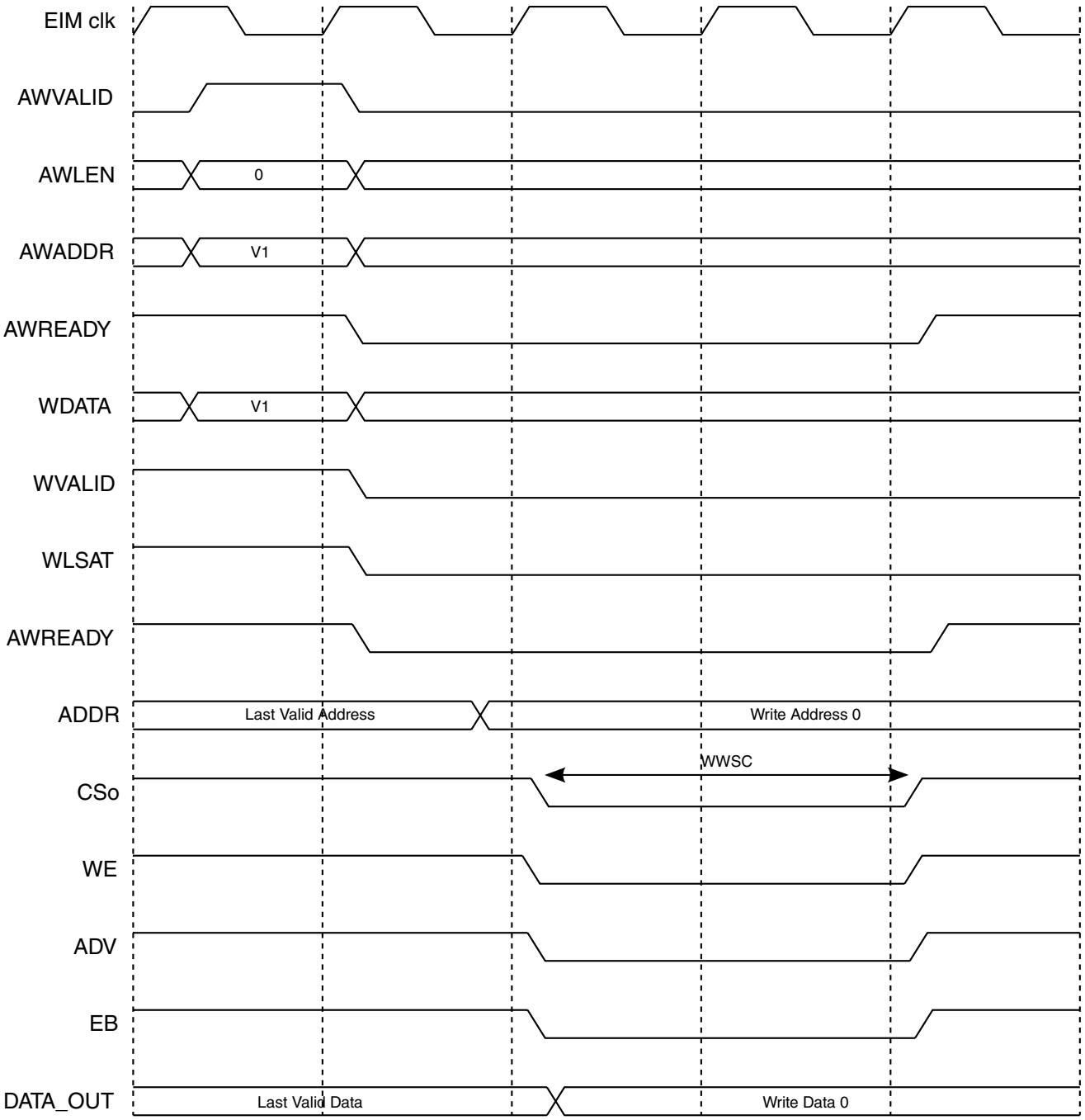


Figure 22-3. Write Access, WWSC=2,WCSA=0,WEA=0,WCSN=0,WEN=0,BEA=0, BEN=0, WAL=1

22.8.3 Asynchronous Read/Write Memory Accesses Timing Diagram

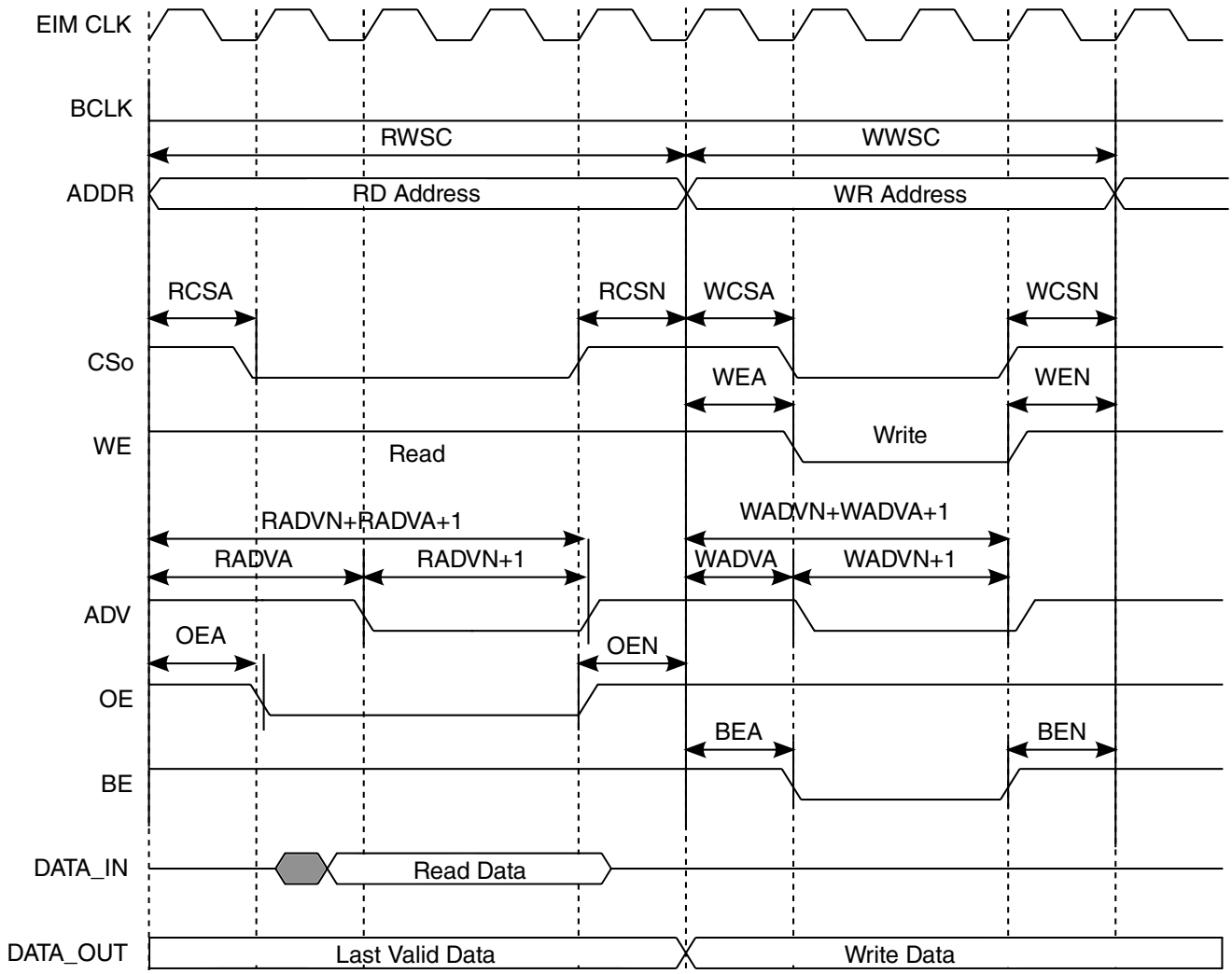


Figure 22-4.
RCSA=1,RADVA=2,OEA=1,RADVN=1,RCSN=1,OEN=1,WCSA=1,WEA=1,WADVA=1,BEA=1,WADVN=1,WCSN=1,WEN=1,BEN=1

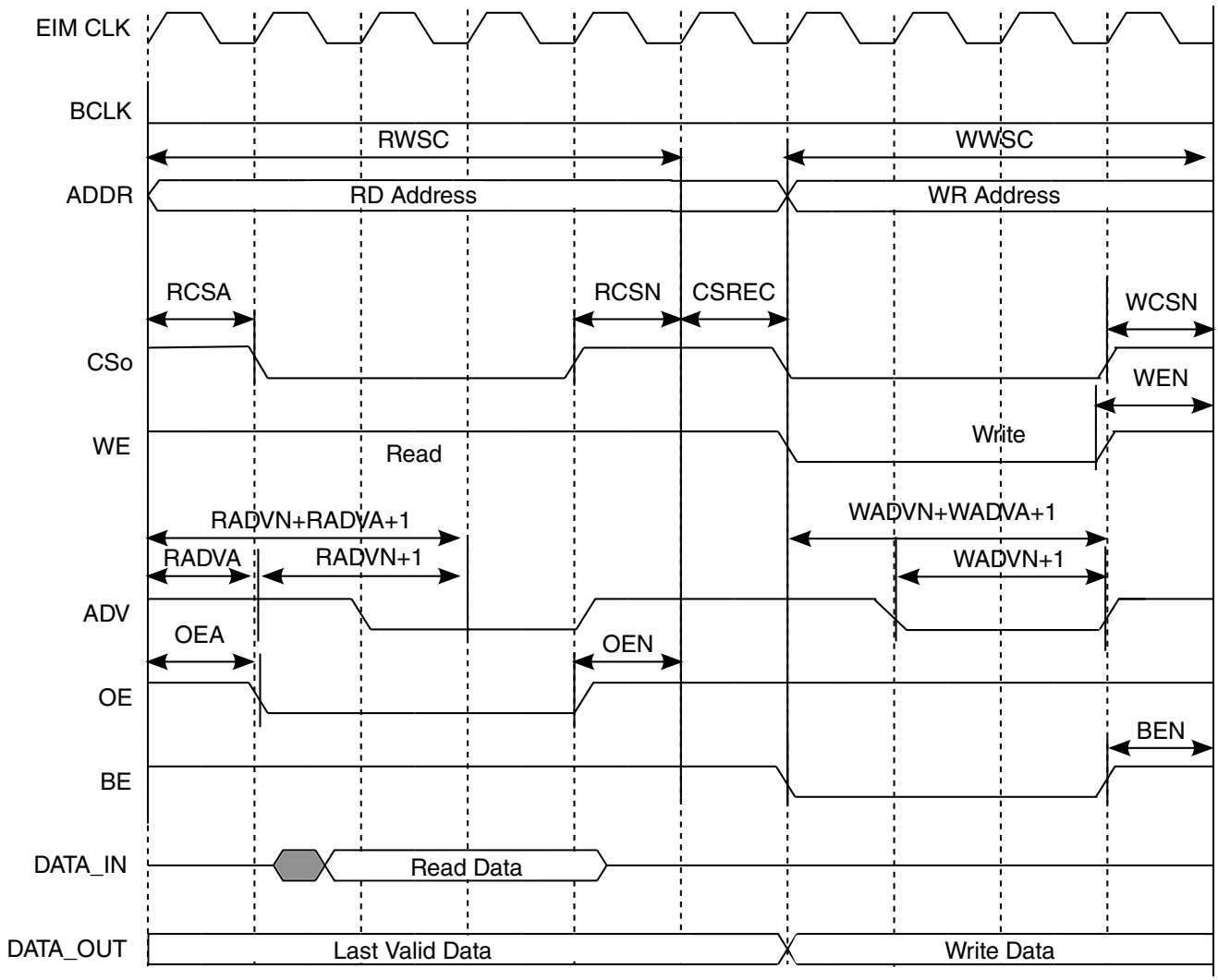


Figure 22-5.
RWSC=5,RCSA=1,RCSN=1,RADVA=1,RADVN=1,OEA=1,OEN=1,WWSC=4,WCSA=0,WCSN=1,WEA=0,WEN=1,WADVA=1,WADVN=1,BEA=0,BEN=1,CSREC=1

22.8.4 Asynchronous Read/Write Using RAL, WAL and CSREC

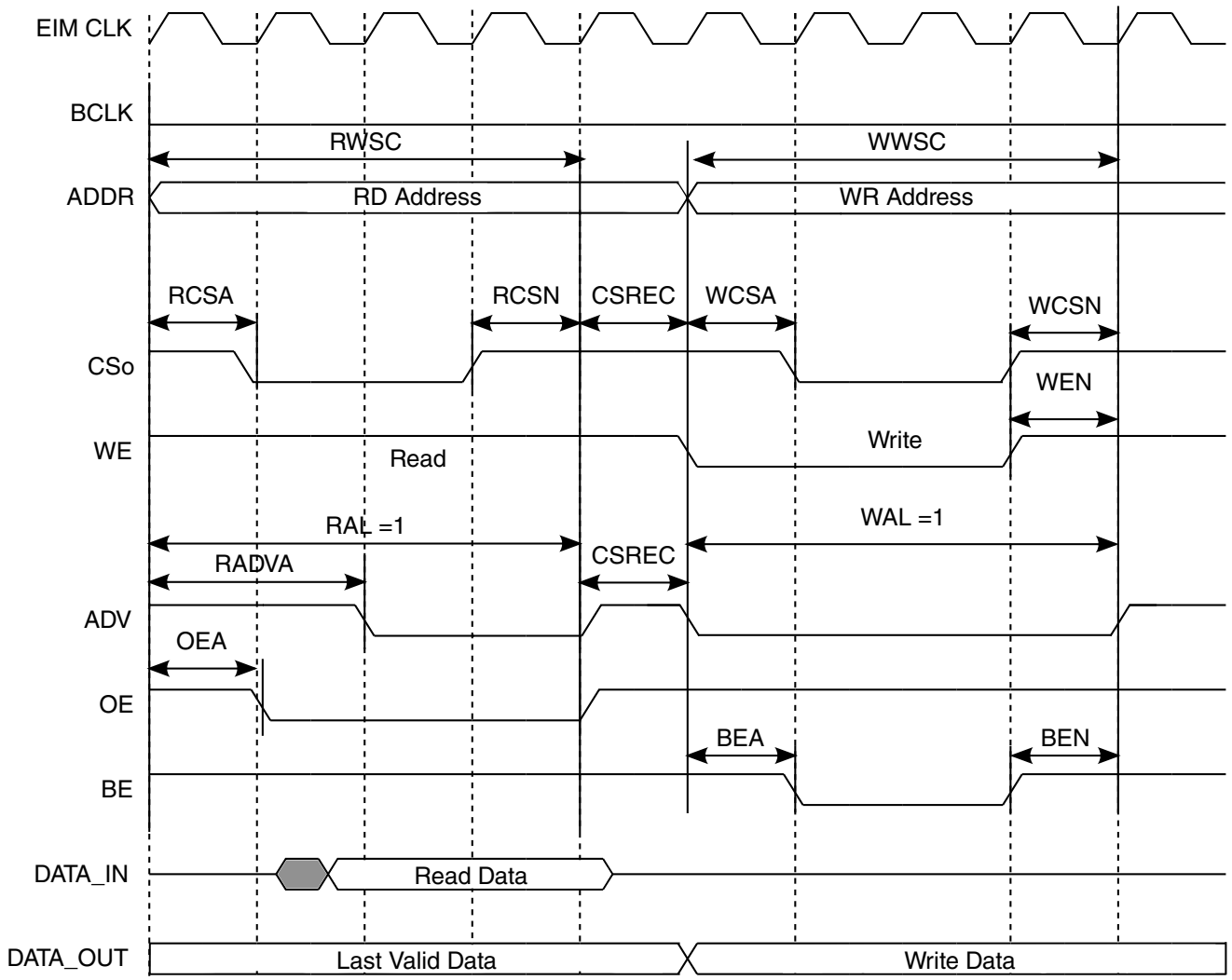


Figure 22-6.
RAL=1,RCSN=1,RADVA=2,OEA=1,RCSN=1,CSREC=1,WCSA=1,WEA=0,WADVA=0,BEA=1,WAL=1,WCSN=1,WEN=1,BEN=1

22.8.5 Consecutive Asynchronous Write Memory Accesses Timing Diagram

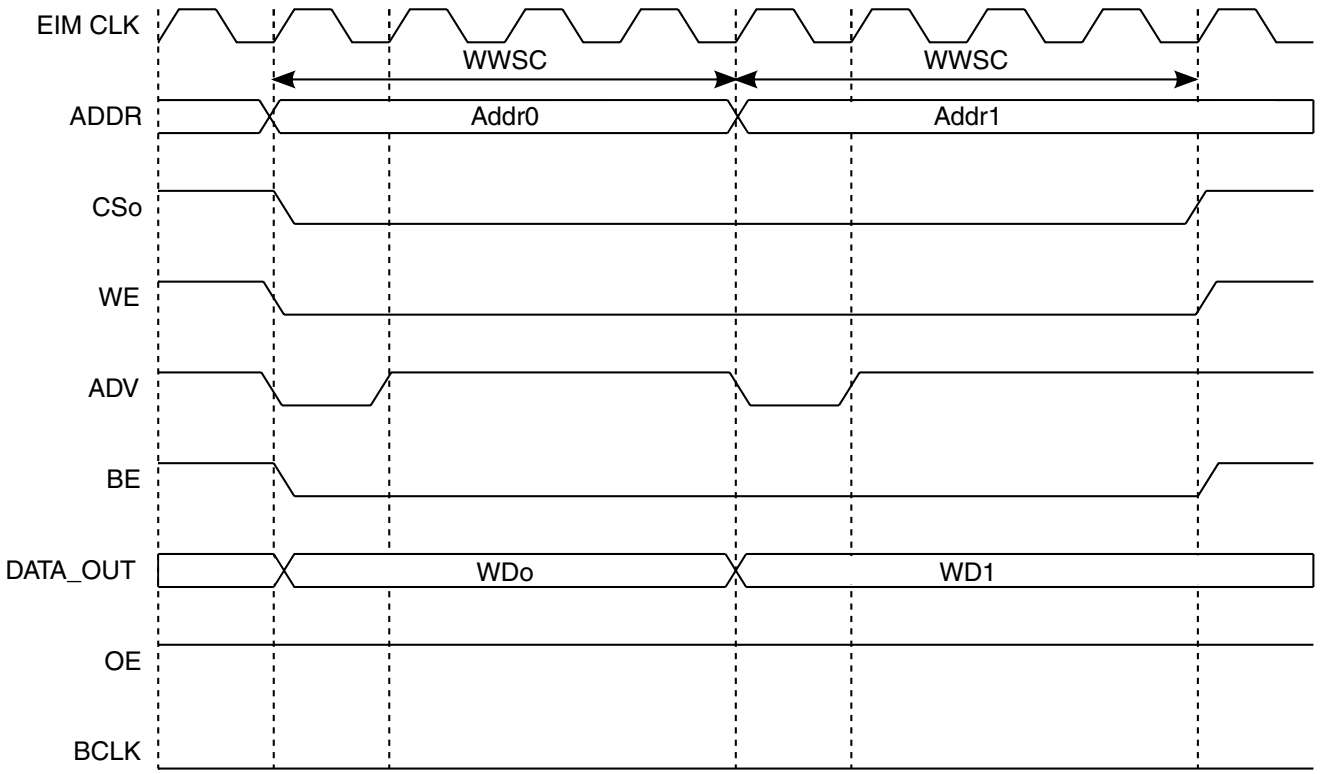


Figure 22-7. WWSC=4,WCSA=0,WEA=0,WADVA=0,BEA=0,WCSN=0,WEN=0,WADVN=0,BEN=0,CSREC=0

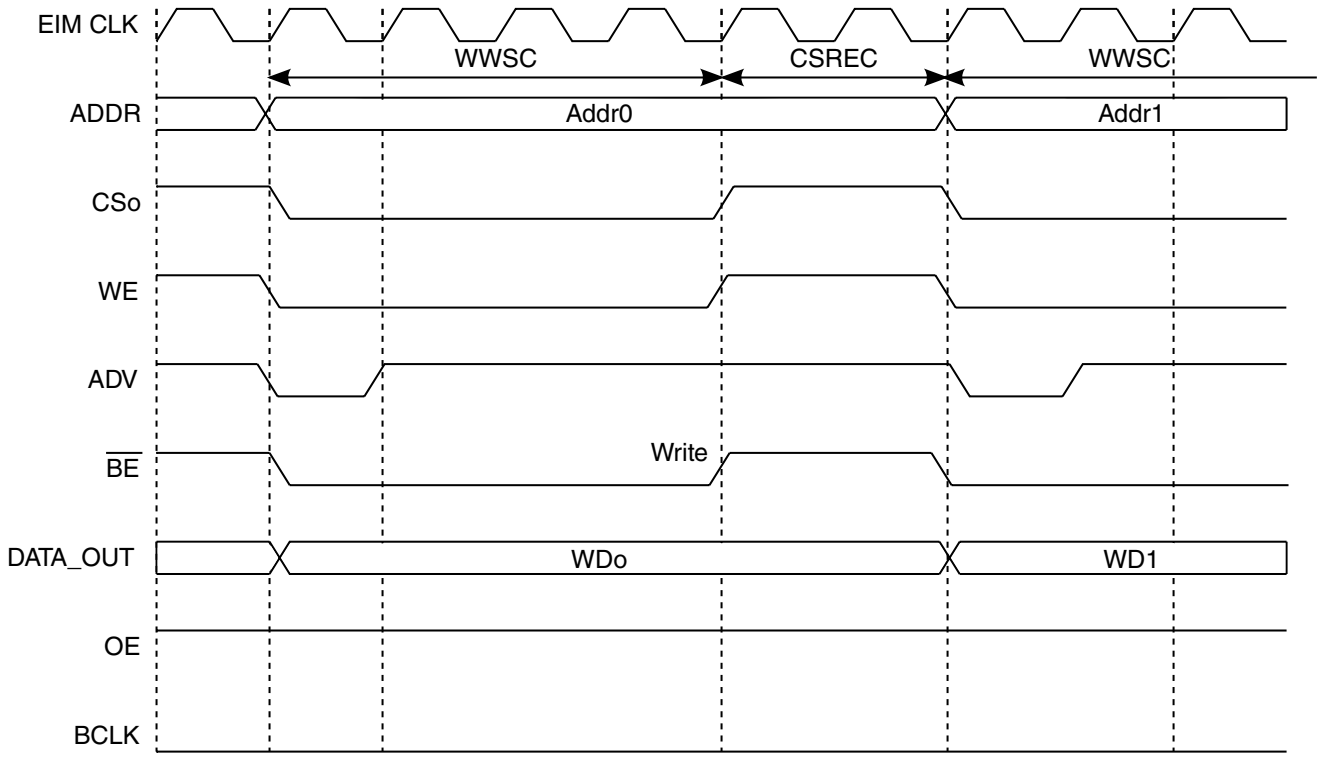


Figure 22-8. WWSC=4,WCSA=0,WEA=0,WADVA=0,BEA=0,WCSN=0,WEN=0,WADVN=0,BEN=0,CSREC=2

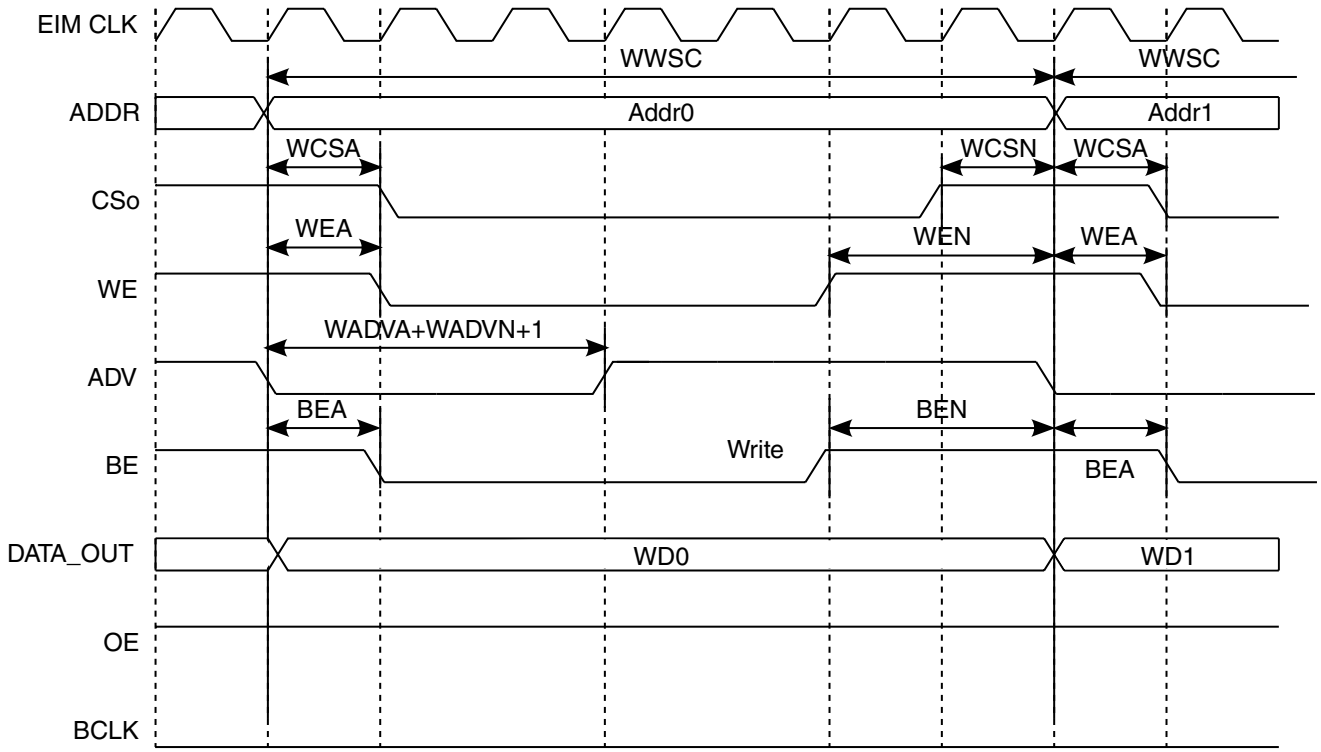


Figure 22-9.
WWSC=7,WCSA=1,WCSN=1,WEA=1,WEN=2,WADVA=0,WADVN=2,BEA=1,BEN=2

22.8.6 Consecutive Asynchronous Read Memory Accesses Timing Diagram

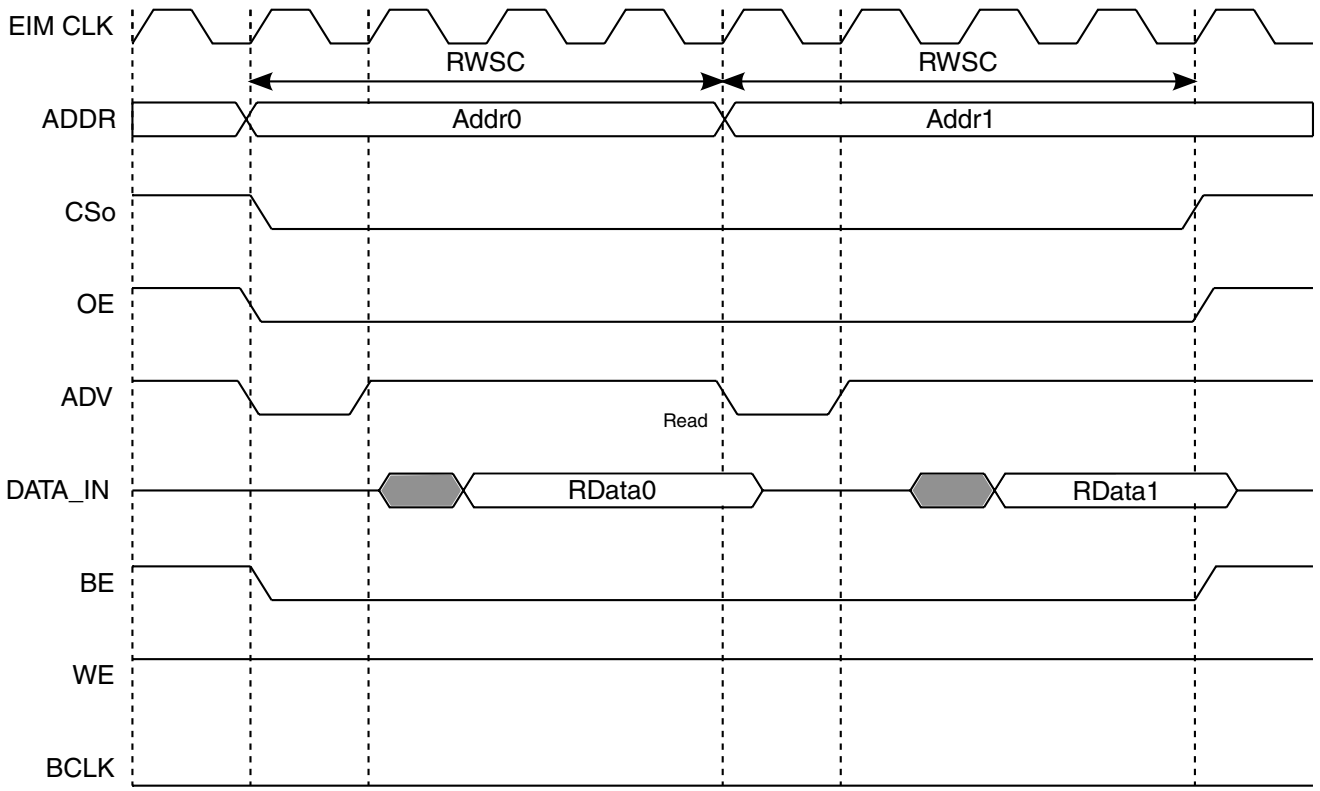


Figure 22-10. RWSC=4,RCSA=0,OEA=0,RADVA=0,RCSN=0,OEN=0,RADVN=0,CSREC=0

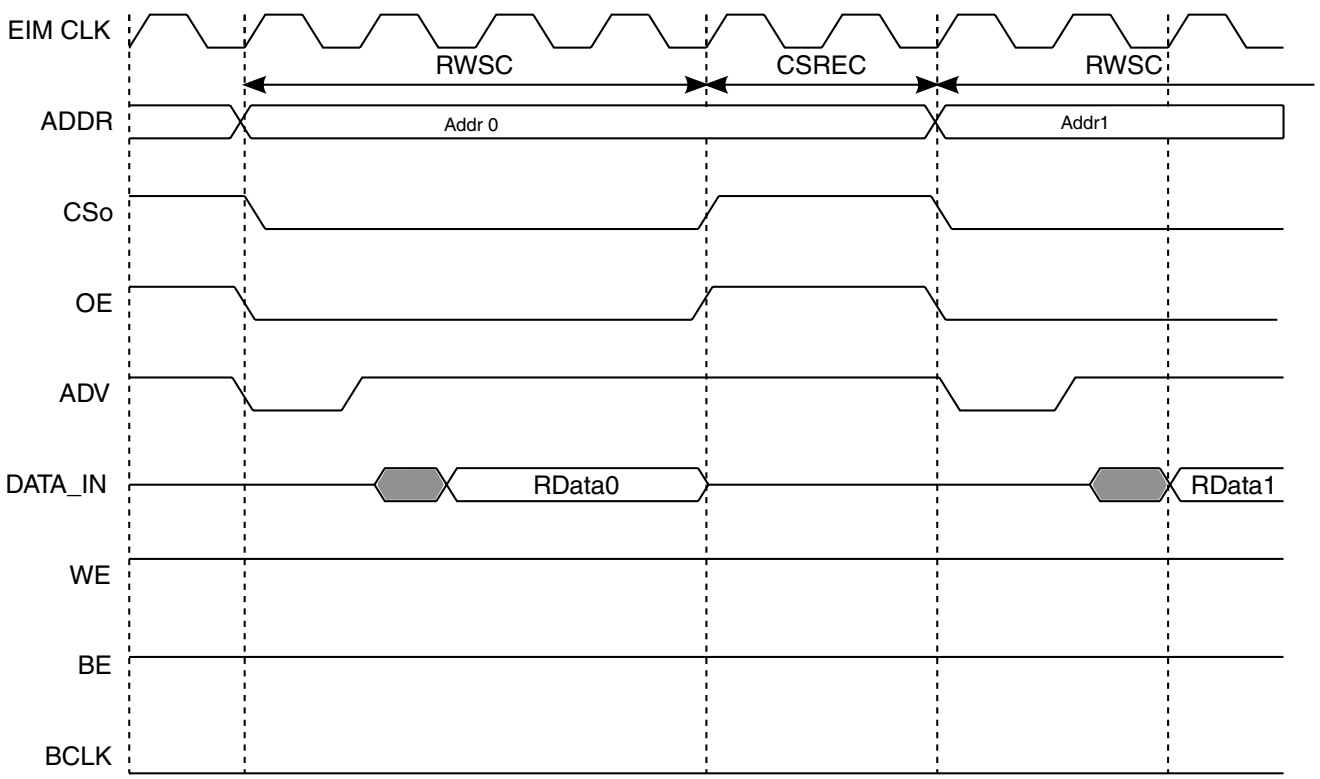


Figure 22-11. RWSC=4,RCSA=0,OEA=0,RADVA=0,RCSN=0,OEN=0,RADVN=0,CSREC=2

22.8.7 Burst (Synchronous Mode) Read Memory Accesses Timing Diagram - BCD=0

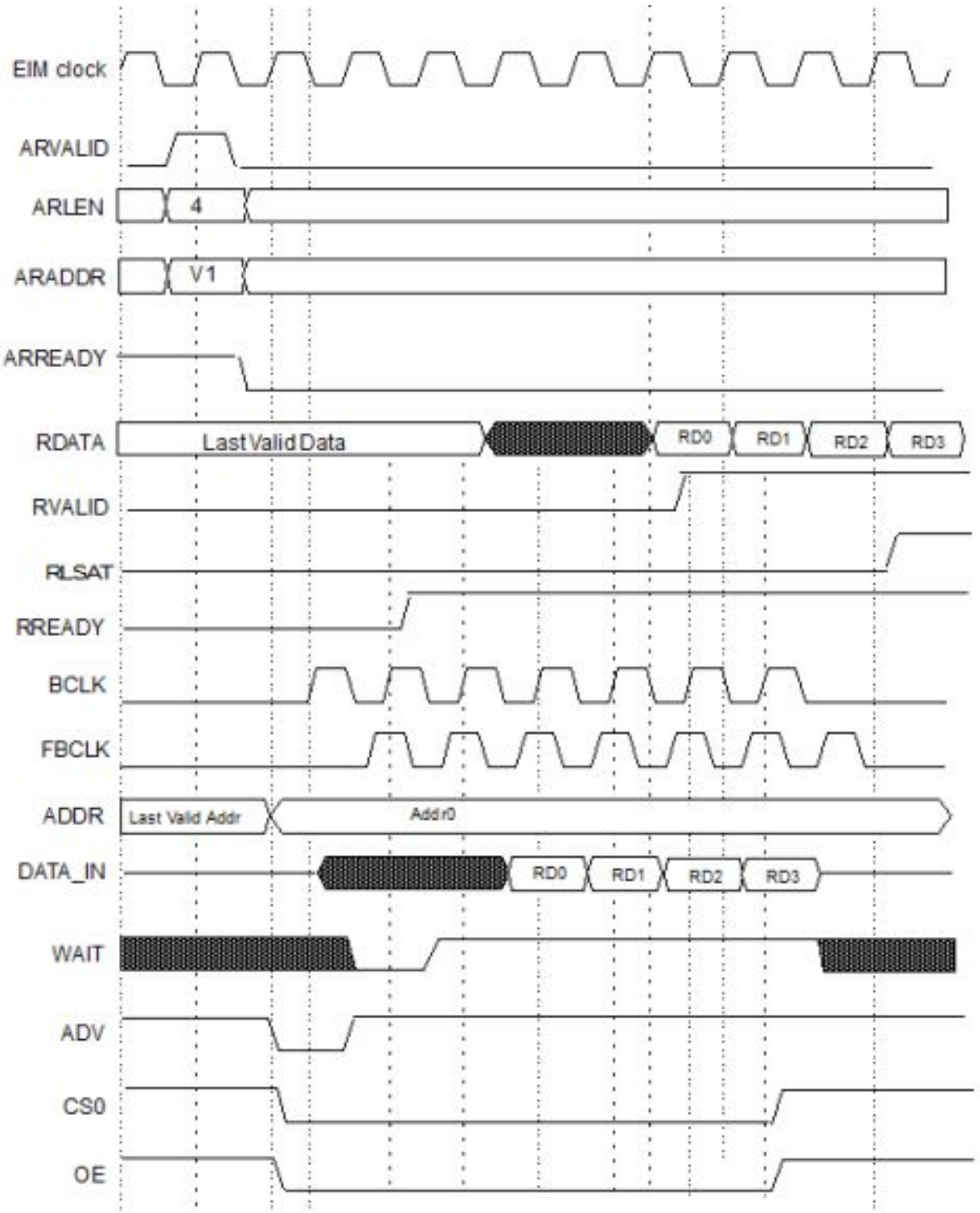


Figure 22-12. SRD=1,BCD=0,BCS=0,RWSC=1,RADVA=0,RADVN=0,RFL=0,RL=0

22.8.8 Burst (Synchronous Mode) Read Memory Accesses Timing Diagram - BCD=1

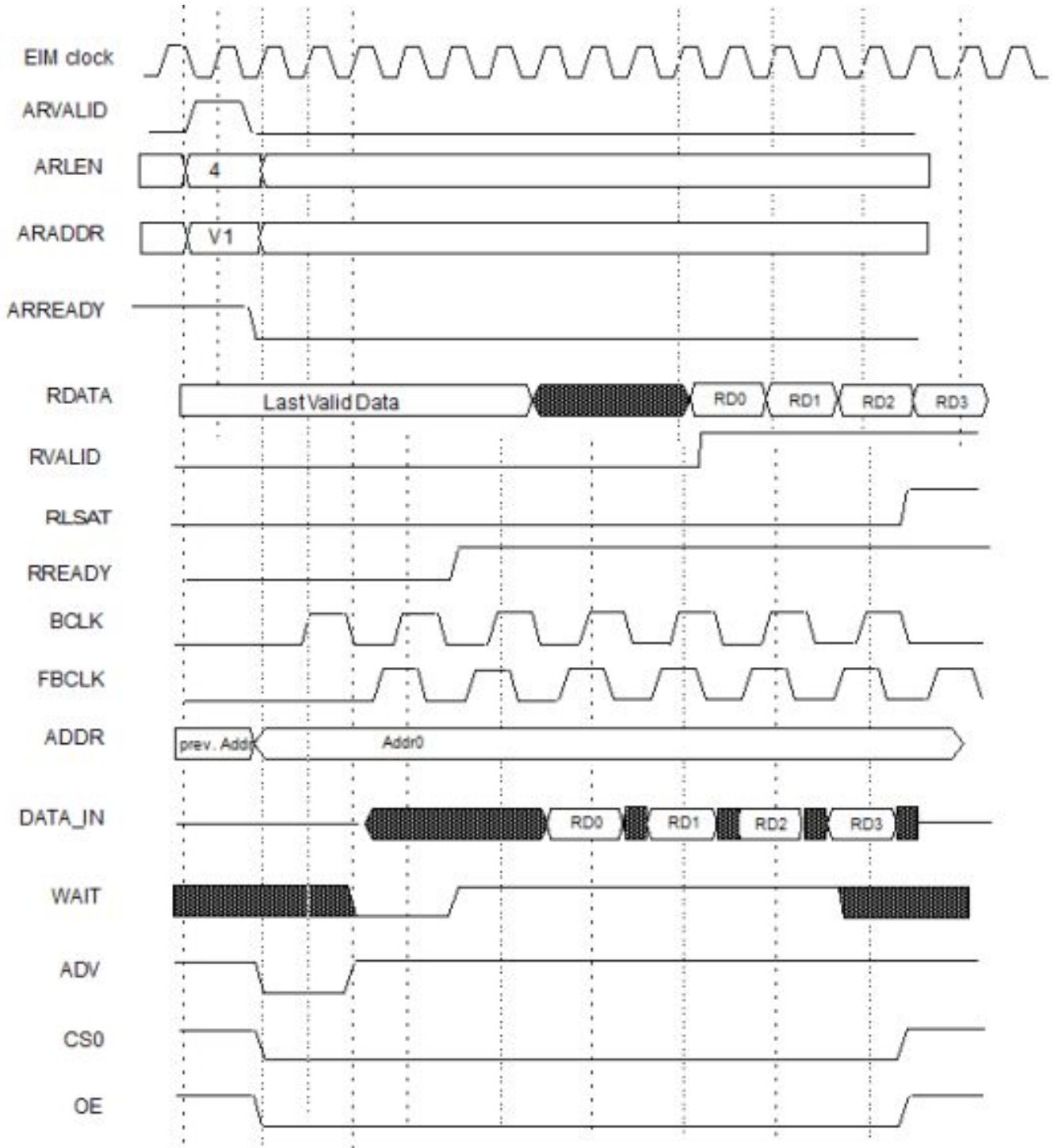
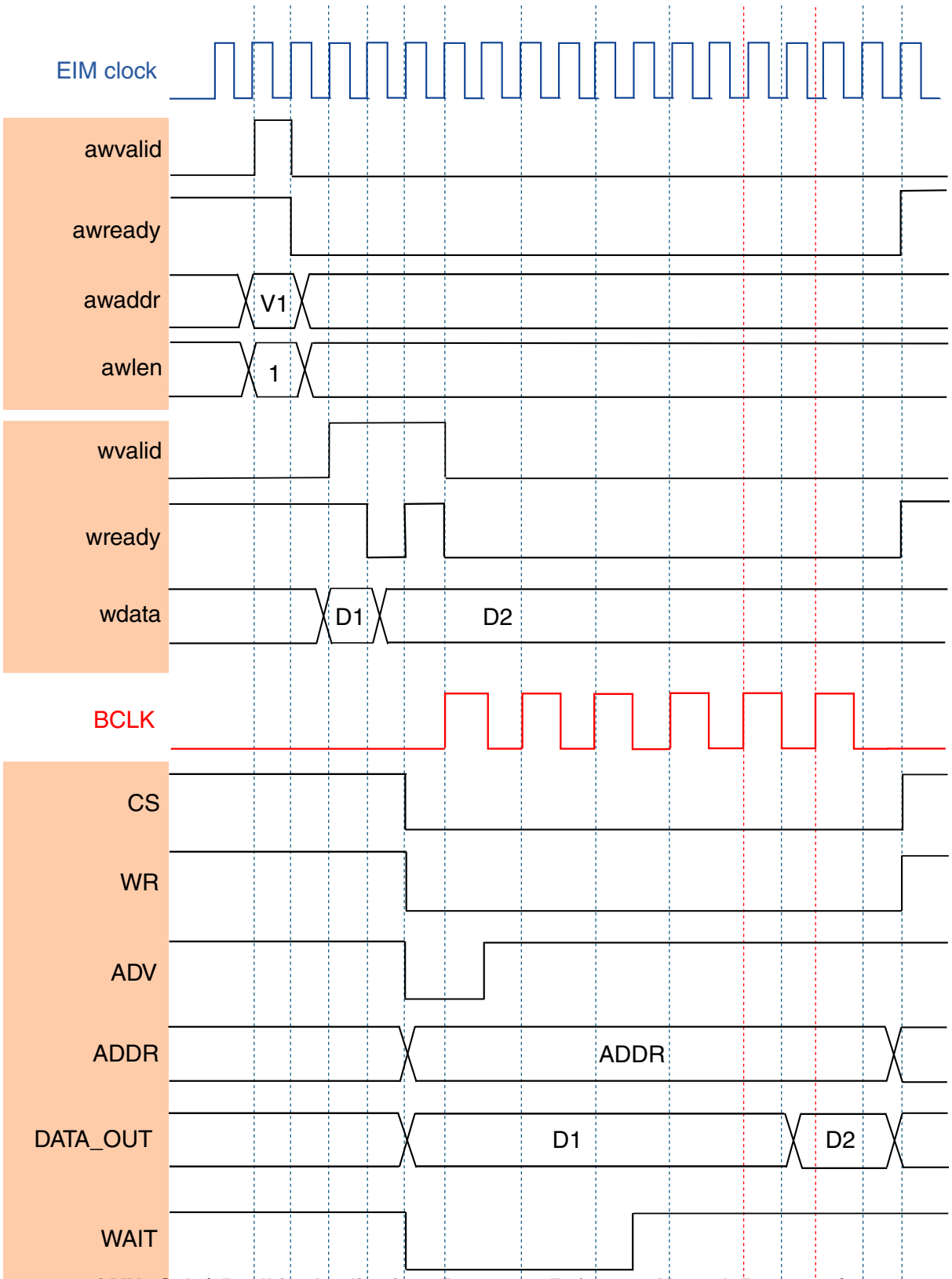


Figure 22-13. BCD=1, RL = 3

22.8.9 Burst (Synchronous Mode) Write Memory Access Timing - BCD=1



22.8.10 Asynchronous Page Mode Access

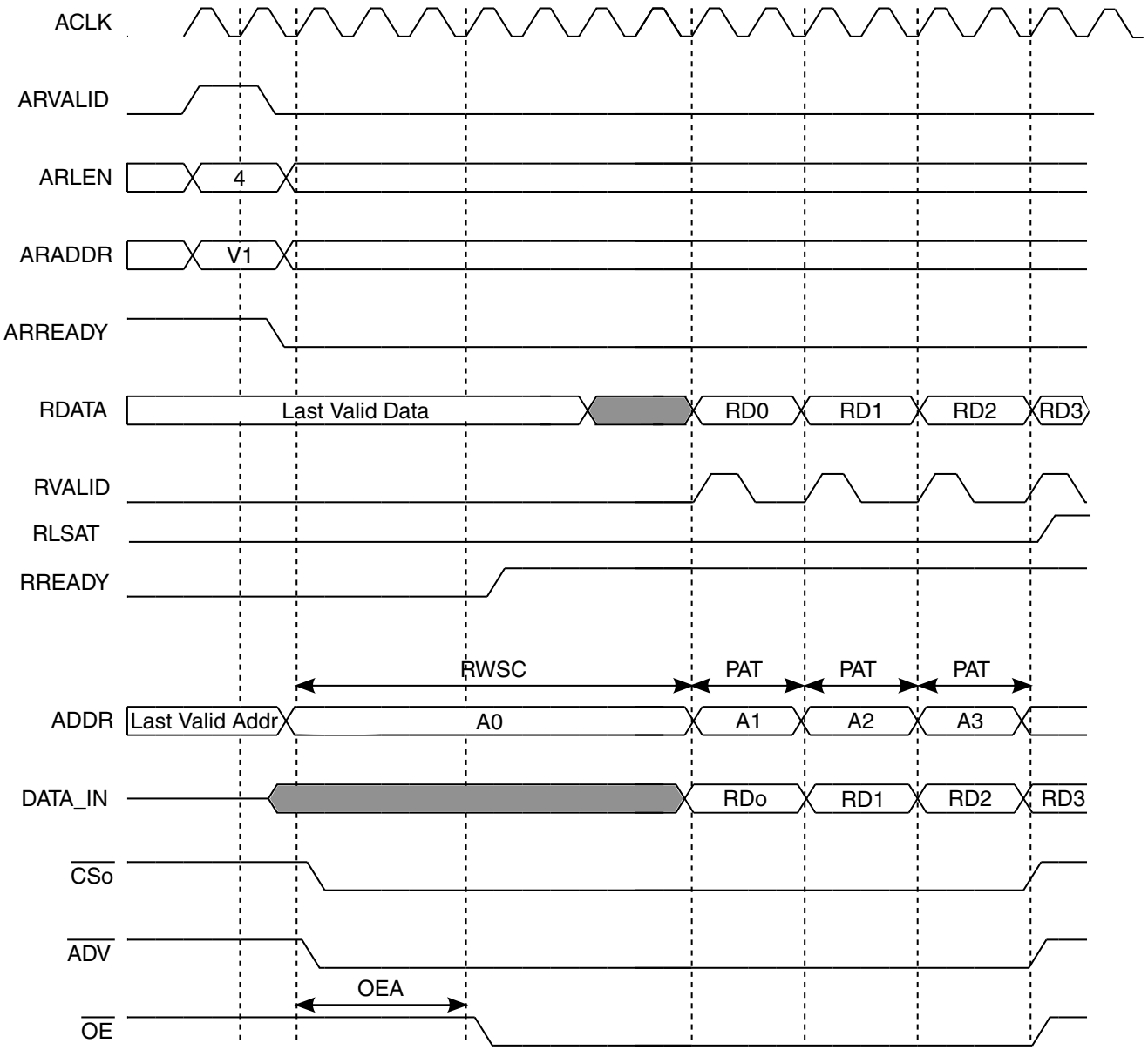


Figure 22-14. PAT = 2

22.8.11 DTACK Mode - AXI Single Access

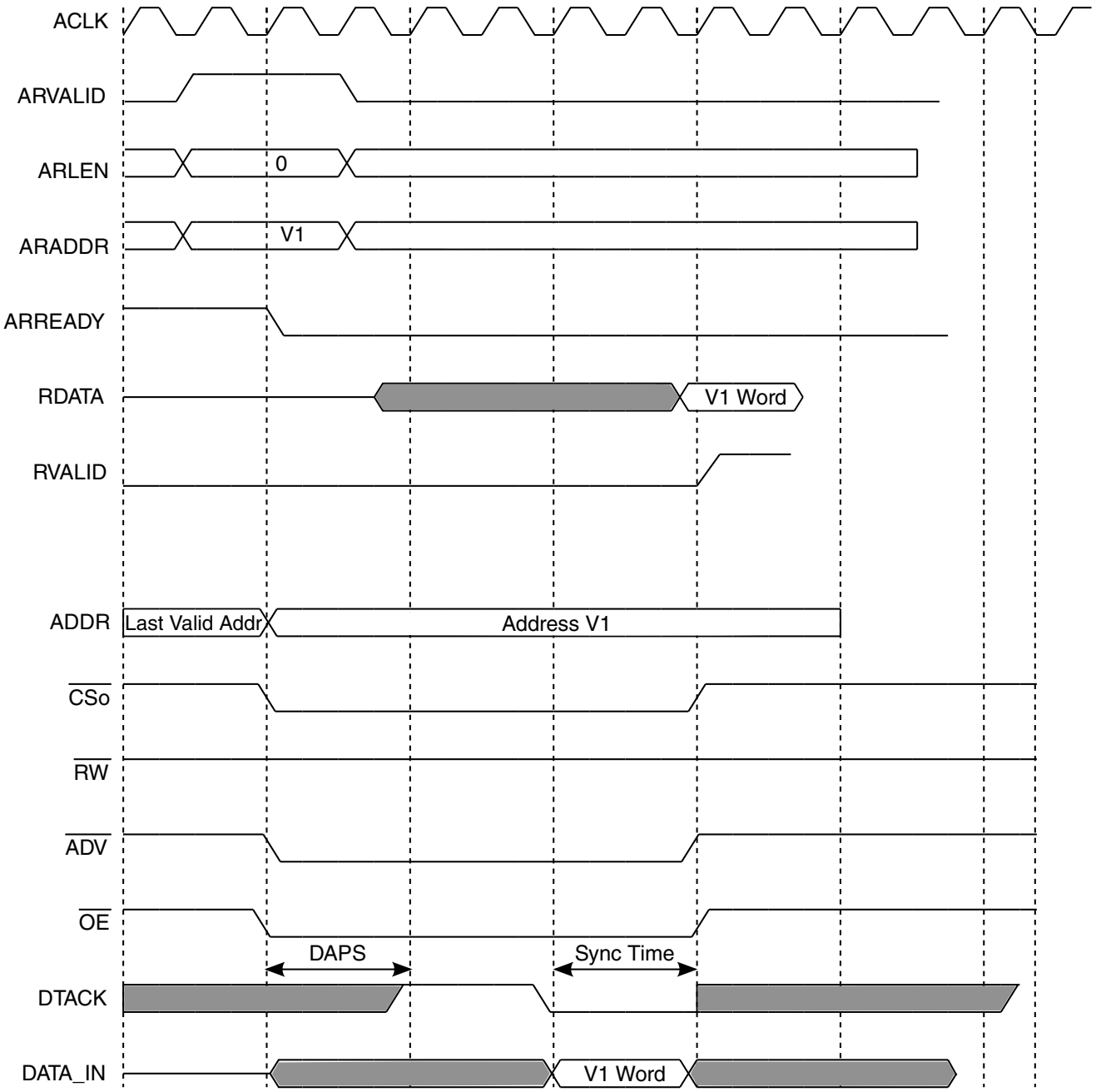


Figure 22-15. DAPS = 2

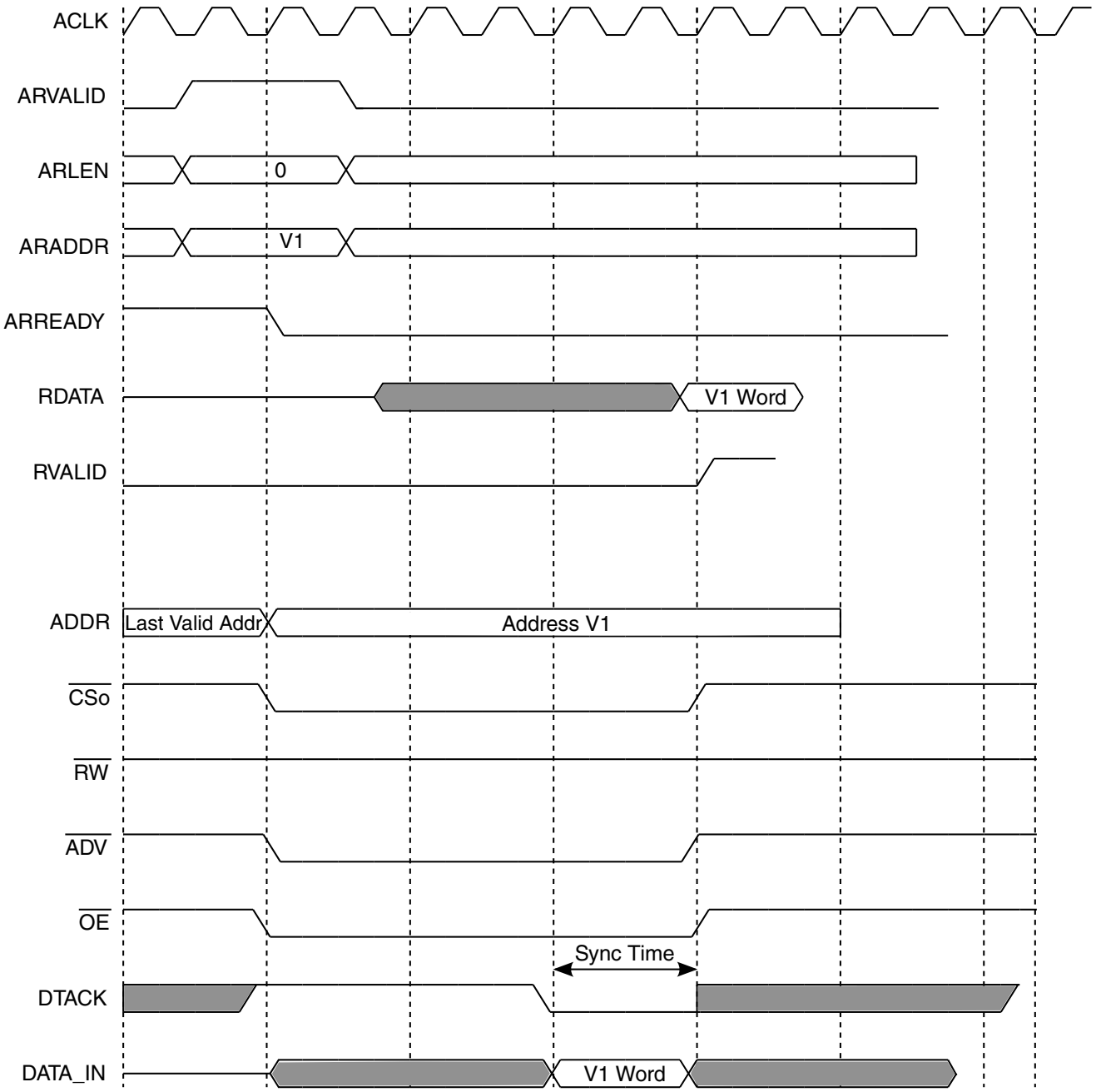


Figure 22-16. DAPS = 0

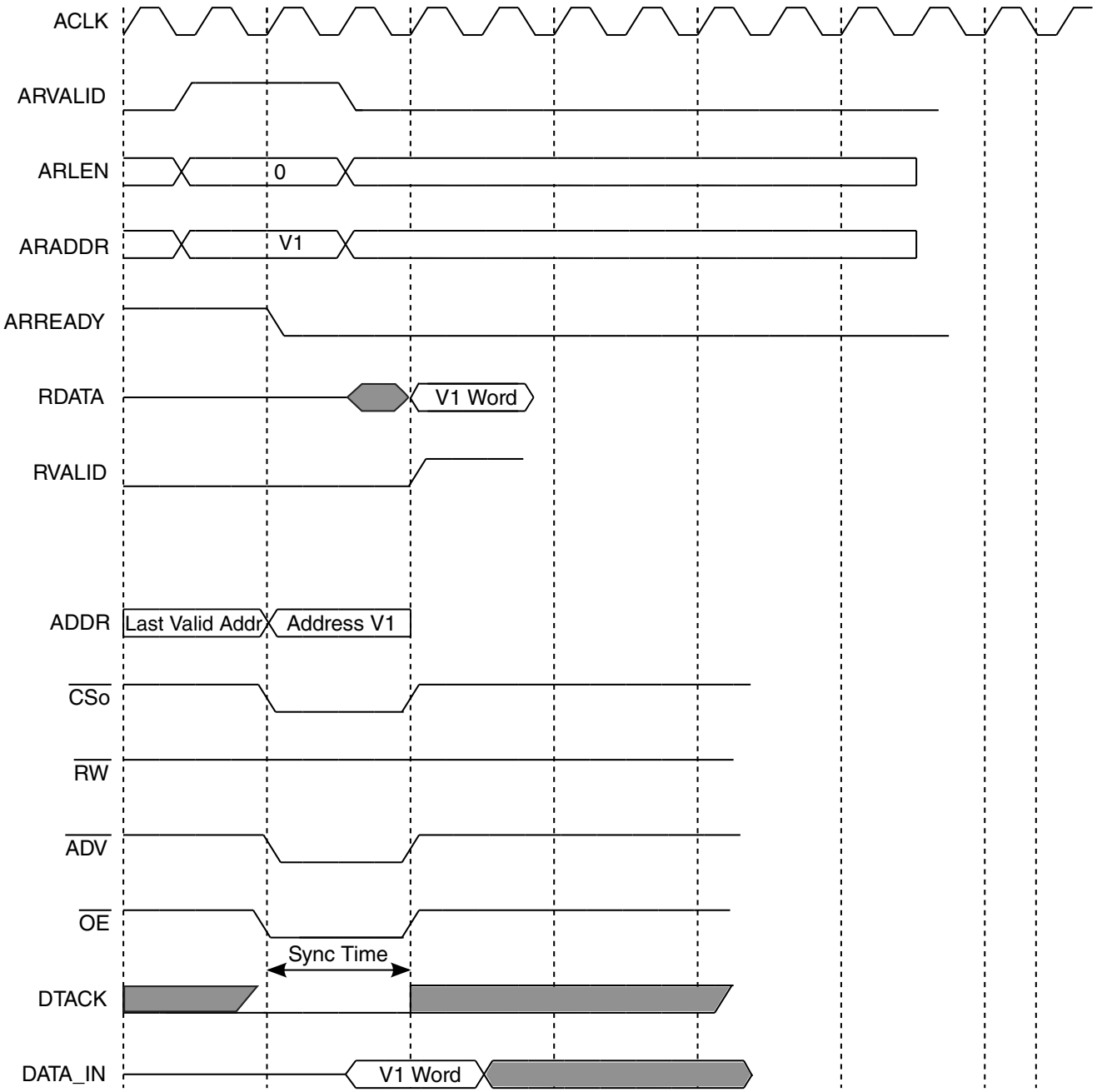


Figure 22-17. DAPS = 0

22.8.12 DTACK Mode - AXI Single Write Access

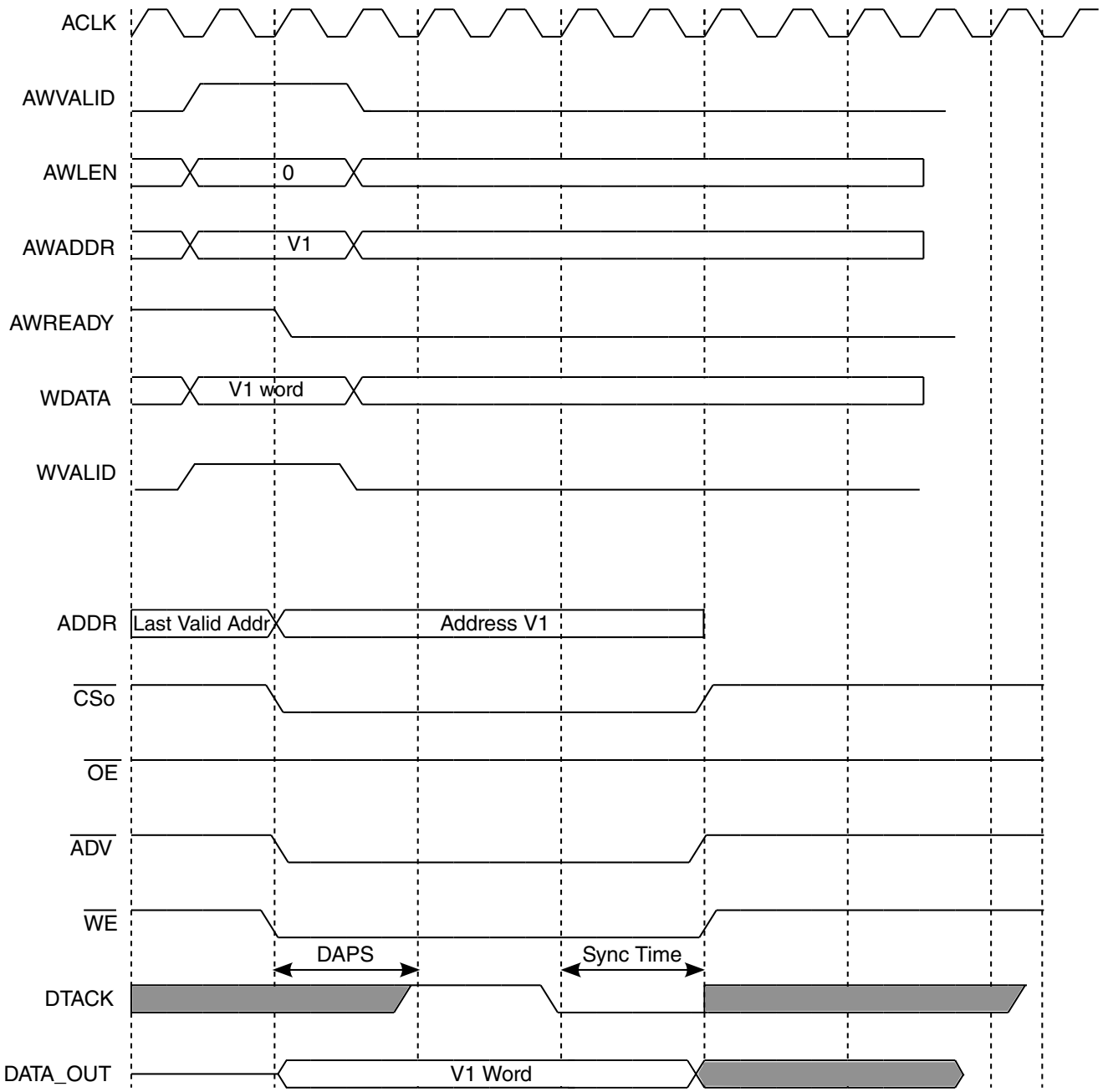


Figure 22-18. DAPS = 2

22.8.13 DTACK Mode - AXI Burst Access

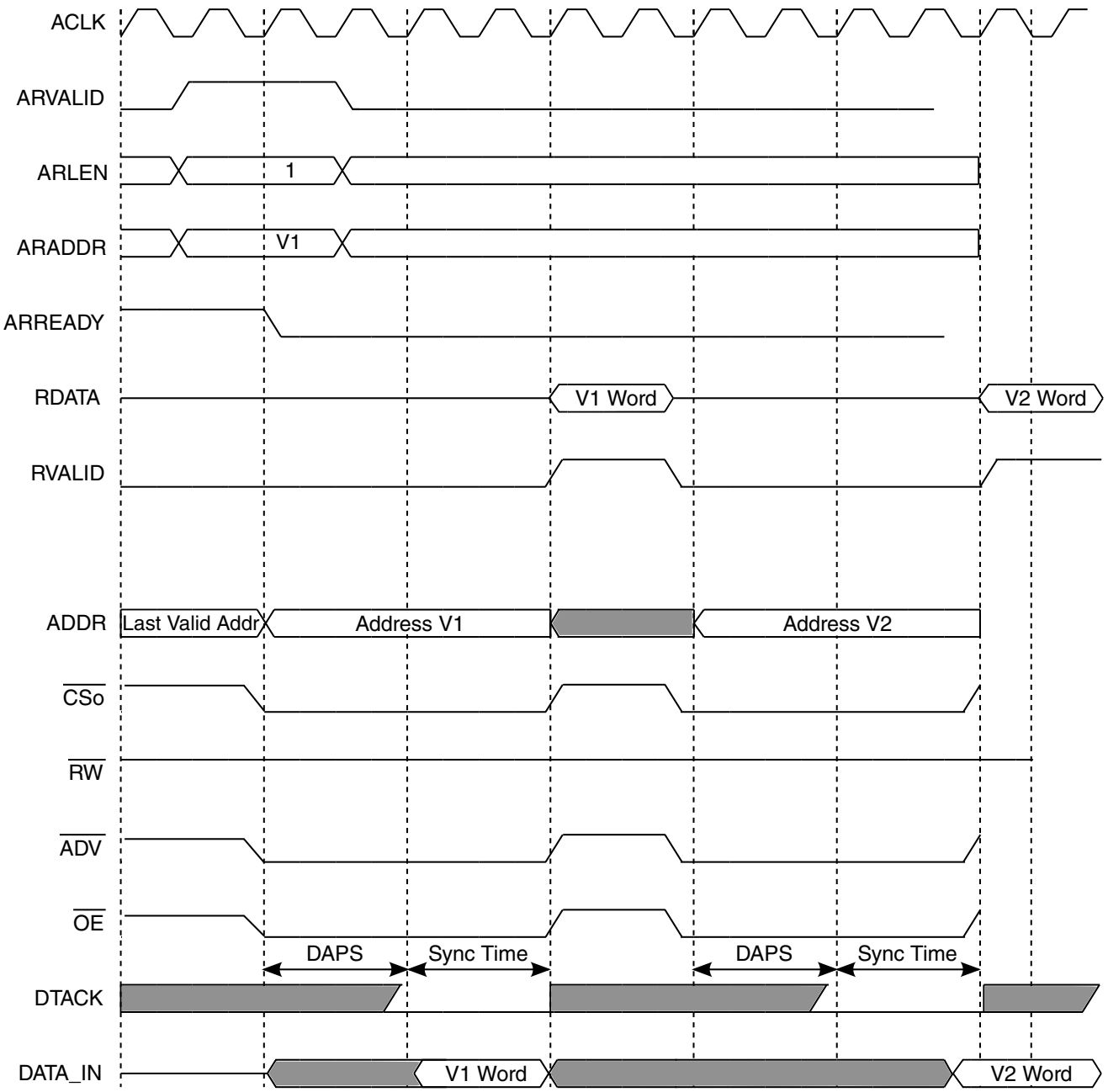


Figure 22-19. DAPS = 2 CSREC = 2

22.9 EIM Memory Map/Register Definition

The EIM includes 33 user-accessible 32-bit registers. The the EIM Configuration Register (EIM_WCR) contains control bits that configure the EIM for certain operation modes.

The 160 bits used to control Individual Chip Select are divided into five registers:

- Chip Select n General Configuration Register 1 (EIM_CSnGCR1)
- Chip Select n General Configuration Register 2 (EIM_CSnGCR2)
- Chip Select n Read Configuration Register 1 (EIM_CSnRCR1)
- Chip Select n Read Configuration Register 2 (EIM_CSnRCR2)
- Chip Select n Write Configuration Register (EIM_CSnWCR)

In addition there are 3 general registers: EIM_WCR, EIM_WIAR, and EIM_EAR.

NOTE

- All EIM registers are sampled by IPG_CLK_S, therefore IPG_CLK_S must be active when accessing through IP bus.
- Read access from all registers (except EIM_WIAR and EIM_EAR) will generate one IPG_XFR_WAIT cycle.
- Read access from EIM_WIAR and EIM_EAR will generate six IPG_XFR_WAIT cycles.
- Write access to all registers (except EIM_EAR) will generate three IPG_XFR_WAIT cycles.
- Write access to EIM_EAR will generate six IPG_XFR_WAIT cycles.

EIM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21B_8000	Chip Select n General Configuration Register 1 (EIM_CS0GCR1)	32	R/W	0061_0088h	22.9.1/1038
21B_8004	Chip Select n General Configuration Register 2 (EIM_CS0GCR2)	32	R/W	0000_1010h	22.9.2/1042
21B_8008	Chip Select n Read Configuration Register 1 (EIM_CS0RCR1)	32	R/W	1C00_2000h	22.9.3/1043
21B_800C	Chip Select n Read Configuration Register 2 (EIM_CS0RCR2)	32	R/W	0000_0000h	22.9.4/1046
21B_8010	Chip Select n Write Configuration Register 1 (EIM_CS0WCR1)	32	R/W	1C00_0000h	22.9.5/1047
21B_8014	Chip Select n Write Configuration Register 2 (EIM_CS0WCR2)	32	R/W	0000_0000h	22.9.6/1050
21B_8018	Chip Select n General Configuration Register 1 (EIM_CS1GCR1)	32	R/W	0061_0088h	22.9.1/1038

Table continues on the next page...

EIM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21B_801C	Chip Select n General Configuration Register 2 (EIM_CS1GCR2)	32	R/W	0000_1010h	22.9.2/1042
21B_8020	Chip Select n Read Configuration Register 1 (EIM_CS1RCR1)	32	R/W	1C00_2000h	22.9.3/1043
21B_8024	Chip Select n Read Configuration Register 2 (EIM_CS1RCR2)	32	R/W	0000_0000h	22.9.4/1046
21B_8028	Chip Select n Write Configuration Register 1 (EIM_CS1WCR1)	32	R/W	1C00_0000h	22.9.5/1047
21B_802C	Chip Select n Write Configuration Register 2 (EIM_CS1WCR2)	32	R/W	0000_0000h	22.9.6/1050
21B_8030	Chip Select n General Configuration Register 1 (EIM_CS2GCR1)	32	R/W	0061_0088h	22.9.1/1038
21B_8034	Chip Select n General Configuration Register 2 (EIM_CS2GCR2)	32	R/W	0000_1010h	22.9.2/1042
21B_8038	Chip Select n Read Configuration Register 1 (EIM_CS2RCR1)	32	R/W	1C00_2000h	22.9.3/1043
21B_803C	Chip Select n Read Configuration Register 2 (EIM_CS2RCR2)	32	R/W	0000_0000h	22.9.4/1046
21B_8040	Chip Select n Write Configuration Register 1 (EIM_CS2WCR1)	32	R/W	1C00_0000h	22.9.5/1047
21B_8044	Chip Select n Write Configuration Register 2 (EIM_CS2WCR2)	32	R/W	0000_0000h	22.9.6/1050
21B_8048	Chip Select n General Configuration Register 1 (EIM_CS3GCR1)	32	R/W	0061_0088h	22.9.1/1038
21B_804C	Chip Select n General Configuration Register 2 (EIM_CS3GCR2)	32	R/W	0000_1010h	22.9.2/1042
21B_8050	Chip Select n Read Configuration Register 1 (EIM_CS3RCR1)	32	R/W	1C00_2000h	22.9.3/1043
21B_8054	Chip Select n Read Configuration Register 2 (EIM_CS3RCR2)	32	R/W	0000_0000h	22.9.4/1046
21B_8058	Chip Select n Write Configuration Register 1 (EIM_CS3WCR1)	32	R/W	1C00_0000h	22.9.5/1047
21B_805C	Chip Select n Write Configuration Register 2 (EIM_CS3WCR2)	32	R/W	0000_0000h	22.9.6/1050
21B_8090	EIM Configuration Register (EIM_WCR)	32	R/W	See section	22.9.7/1051
21B_8094	DLL Control Register (EIM_DCR)	32	R/W	0014_0000h	22.9.8/1053
21B_8098	DLL Status Register (EIM_DSR)	32	R	0000_0000h	22.9.9/1054
21B_809C	EIM IP Access Register (EIM_WIAR)	32	R/W	0000_0010h	22.9.10/ 1055
21B_80A0	Error Address Register (EIM_EAR)	32	R/W	0000_0000h	22.9.11/ 1056

22.9.1 Chip Select n General Configuration Register 1 (EIM_CS*n*GCR1)

Address: 21B_8000h base + 0h offset + (24d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PSZ				WP	GBC			AUS	CSREC			SP	DSZ		
Reset	0	0	0	0	0	0	0	0	0				0			

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BCS		BCD		WC	BL			CREP	CRE	RFL	WFL	MUM	SRD	SWR	CSEN
Reset	0	0	0	0	0	0	0	0	1	0	0	0		0	0	0

EIM_CS*n*GCR1 field descriptions

Field	Description
31–28 PSZ	Page Size. This bit field indicates memory page size in words (word is defined by the DSZ field). PSZ is used when fix latency mode is applied, WFL=1 for sync. write accesses, RFL=1 for sync. Read accesses. When working in fix latency mode WAIT signal from the external device is not being monitored, PSZ is used to determine if page boundary is reached and renewal of access is preformed. This bit field is ignored when sync. Mode is disabled or fix latency mode is not being used for write or read access separately. It can be valid for both access type, read or write, or only for one type, according to configuration. PSZ is cleared by a hardware reset. 0000 8 words page size 0001 16 words page size 0010 32 words page size 0011 64 words page size 0100 128 words page size 0101 256 words page size 0110 512 words page size 0111 1024 (1k) words page size 1000 2048 (2k) words page size 1001 - 1111 Reserved
27 WP	Write Protect. This bit prevents writes to the address range defined by the corresponding chip select. WP is cleared by a hardware reset. 0 Writes are allowed in the memory range defined by chip. 1 Writes are prohibited. All attempts to write to an address mapped by this chip select result in a error response and no assertion of the chip select output.
26–24 GBC	Gap Between Chip Selects. This bit field, according to the settings shown below, determines the minimum time between end of access to the current chip select and start of access to different chip select. GBC is cleared by a hardware reset.

Table continues on the next page...

EIM_CS*n*GCR1 field descriptions (continued)

Field	Description
	Example settings: 000 minimum of 0 EIM clock cycles before next access from different chip select (async. mode only) 001 minimum of 1 EIM clock cycles before next access from different chip select 010 minimum of 2 EIM clock cycles before next access from different chip select 111 minimum of 7 EIM clock cycles before next access from different chip select
23 AUS	Address UnShifted. This bit indicates an unshifted mode for address assertion for the relevant chip select accesses. AUS bit is cleared by hardware reset. 0 Address shifted according to port size (DSZ config) (128 Mbyte maximum supported memory density). 1 Address unshifted (32 Mbyte maximum supported memory density).
22–20 CSREC	CS Recovery. This bit field, according to the settings shown below, determines the minimum pulse width of CS, OE, and WE control signals before executing a new back to back access to the same chip select. CSREC is cleared by a hardware reset. **NOTE:** The reset value for EIM_CS0GCR1, CSREC[2:0] is 0b110. For EIM_CS1GCR1 - EIM_CS5GCR, the reset value is 0b000. Example settings: 000 0 EIM clock cycles minimum width of CS, OE and WE signals (read async. mode only) 001 1 EIM clock cycles minimum width of CS, OE and WE signals 010 2 EIM clock cycles minimum width of CS, OE and WE signals 111 7 EIM clock cycles minimum width of CS, OE and WE signals
19 SP	Supervisor Protect. This bit prevents accesses to the address range defined by the corresponding chip select when the access is attempted in the User mode. SP is cleared by a hardware reset. 0 User mode accesses are allowed in the memory range defined by chip select. 1 User mode accesses are prohibited. All attempts to access an address mapped by this chip select in User mode results in an error response and no assertion of the chip select output.
18–16 DSZ	Data Port Size. This bit field defines the width of an external device's data port as shown below. **NOTE:** Only async. access supported for 8 bit port. **NOTE:** The reset value for EIM_CS0GCR1, DSZ[2] = 0, DSZ[1:0] = EIM_BOOT[1:0]. For EIM_CS1GCR1 - EIM_CS5GCR1, the reset value is 0b001. 000 Reserved. 001 16 bit port resides on DATA[15:0] 010 16 bit port resides on DATA[31:16] 011 32 bit port resides on DATA[31:0] 100 8 bit port resides on DATA[7:0] 101 8 bit port resides on DATA[15:8] 110 8 bit port resides on DATA[23:16] 111 8 bit port resides on DATA[31:24]
15–14 BCS	Burst Clock Start. When SRD=1 or SWR=1,this bit field determines the number of EIM clock cycles delay from start of access before the first rising edge of BCLK is generated. When BCD=0 value of BCS=0 results in a half clock delay after the start of access. For other values of BCD a one clock delay after the start of access is applied, not an immediate assertion. BCS is cleared by a hardware reset. 00 0 EIM clock cycle additional delay

Table continues on the next page...

EIM_CS*n*GCR1 field descriptions (continued)

Field	Description
	01 1 EIM clock cycle additional delay 10 2 EIM clock cycle additional delay 11 3 EIM clock cycle additional delay
13–12 BCD	Burst Clock Divisor. This bit field contains the value used to program the burst clock divisor for BCLK generation. It is used to divide the internal EIMbus frequency. BCD is cleared by a hardware reset. **NOTE:** For other then the mentioned below frequency such as 104 MHz, EIM clock (input clock) should be adjust accordingly. 00 Divide EIM clock by 1 01 Divide EIM clock by 2 10 Divide EIM clock by 3 11 Divide EIM clock by 4
11 WC	Write Continuous. The WI bit indicates that write access to the memory are always continuous accesses regardless of the BL field value. WI is cleared by hardware reset. 0 Write access burst length occurs according to BL value. 1 Write access burst length is continuous.
10–8 BL	Burst Length. The BL bit field indicates memory burst length in words (word is defined by the DSZ field) and should be properly initialized for mixed wrap/increment accesses support. Continuous BL value corresponds to continuous burst length setting of the external memory device. For fix memory burst size, type is always wrap. In case not matching wrap boundaries in both the memory (BL field) and Master access on the current address, EIM update address on the external device address bus and regenerates the access. BL is cleared by a hardware reset. When APR=1, Page Read Mode is applied, BL determine the number of words within the read page burst. BL is cleared by a hardware reset for EIM_CS0GCR1 - EIM_CS5GCR1. 000 4 words Memory wrap burst length (read page burst size when APR = 1) 001 8 words Memory wrap burst length (read page burst size when APR = 1) 010 16 words Memory wrap burst length (read page burst size when APR = 1) 011 32 words Memory wrap burst length (read page burst size when APR = 1) 100 Continuous burst length (2 words read page burst size when APR = 1) 101 Reserved 110 Reserved 111 Reserved
7 CREP	Configuration Register Enable Polarity. This bit indicates CRE memory pin assertion state, active-low or active-high, while executing a memory register set command to the external device (PSRAM memory type). CREP is set by a hardware reset. **NOTE:** Whenever PSRAM is connected the CREP value must be correct also for accesses where CRE is disabled. For Non-PSRAM memory CREP value should be 1. 0 CRE signal is active low 1 CRE signal is active high
6 CRE	Configuration Register Enable. This bit indicates CRE memory pin state while executing a memory register set command to PSRAM external device. CRE is cleared by a hardware reset.

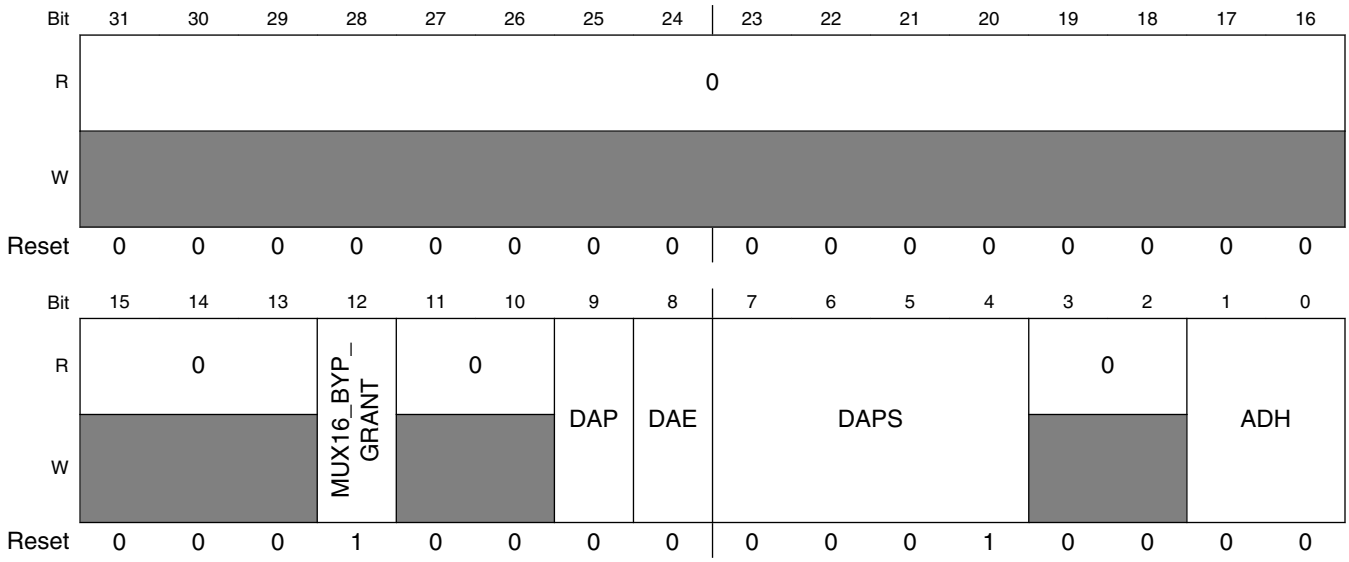
Table continues on the next page...

EIM_CS*n*GCR1 field descriptions (continued)

Field	Description
	0 CRE signal use is disable 1 CRE signal use is enable
5 RFL	Read Fix Latency. This bit field determine if the controller is monitoring the WAIT signal from the External device connected to the chip select (handshake mode - fix or variable data latency) or if it start sampling data according to RWSC field, it only valid in synchronous mode. RFL is cleared by a hardware reset. When RFL=1 Burst access is terminated on page boundary and resume on the following page according to BL bit field configuration, because WAIT signal is not monitored from the external device. 0 the External device WAIT signal is being monitored, and it reflect the external data bus state 1 the state of the External devices is determined internally (Fix latency mode only)
4 WFL	Write Fix Latency. This bit field determine if the controller is monitoring the WAIT signal from the External device connected to the chip select (handshake mode - fix or variable data latency) or if it start data transfer according to WWSC field, it only valid in synchronous mode. WFL is cleared by a hardware reset. When WFL=1 Burst access is terminated on page boundary and resume on the following page according to BL bit field configuration, because WAIT signal is not monitored from the external device 0 the External device WAIT signal is being monitored, and it reflect the external data bus state 1 the state of the External devices is determined internally (Fix latency mode only)
3 MUM	Multiplexed Mode. This bit determines the address/data multiplexed mode for asynchronous and synchronous accesses for 8 bit, 16 bit or 32 bit devices (DSZ config. dependent). **NOTE:** The reset value for EIM_CS0GCR1[MUM] = EIM_BOOT[2]. For EIM_CS1GCR1 - EIM_CS5GCR1 the reset value is 0. 0 Multiplexed Mode disable 1 Multiplexed Mode enable
2 SRD	Synchronous Read Data. This bit field determine the read accesses mode to the External device of the chip select. The External device should be configured to the same mode as this bit implicates. SRD is cleared by a hardware reset. **NOTE:** Sync. accesses supported only for 16/32 bit port. 0 read accesses are in Asynchronous mode 1 read accesses are in Synchronous mode
1 SWR	Synchronous Write Data. This bit field determine the write accesses mode to the External device of the chip select. The External device should be configured to the same mode as this bit implicates. SWR is cleared by a hardware reset. **NOTE:** Sync. accesses supported only for 16/32 bit port. 0 write accesses are in Asynchronous mode 1 write accesses are in Synchronous mode
0 CSEN	CS Enable. This bit controls the operation of the chip select pin. CSEN is set by a hardware reset for CSGCR0 to allow external boot operation. CSEN is cleared by a hardware reset to CSGCR1-CSGCR5. **NOTE:** Reset value for EIM_CS0GCR1 for CSEN is 1. For EIM_CS1GCR1-CS1GCR5 reset value is 0. 0 Chip select function is disabled; attempts to access an address mapped by this chip select results in an error respond and no assertion of the chip select output 1 Chip select is enabled, and is asserted when presented with a valid access.

22.9.2 Chip Select n General Configuration Register 2 (EIM_CS*n*GCR2)

Address: 21B_8000h base + 4h offset + (24d × i), where i=0d to 3d



EIM_CS*n*GCR2 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value 0.
12 MUX16_BYP_GRANT	Muxed 16 bypass grant. This bit when asserted causes EIM to bypass the grant/ack. arbitration with NFC (only for 16 bit muxed mode accesses). 0 EIM waits for grant before driving a 16 bit muxed mode access to the memory. 1 EIM ignores the grant signal and immediately drives a 16 bit muxed mode access to the memory.
11–10 Reserved	This read-only field is reserved and always has the value 0.
9 DAP	Data Acknowledge Polarity. This bit indicates DTACK memory pin assertion state, active-low or active-high, while executing an async access using DTACK signal from the external device. DAP is cleared by a hardware reset. 0 DTACK signal is active high 1 DTACK signal is active low
8 DAE	Data Acknowledge Enable. This bit indicates external device is using DTACK pin as strobe/terminator of an async. access. DTACK signal may be used only in asynchronous single read (APR=0) or write accesses. DTACK poling start point is set by DAPS bit field. polarity of DTACK is set by DAP bit field. DAE is cleared by a hardware reset. 0 DTACK signal use is disable 1 DTACK signal use is enable
7–4 DAPS	Data Acknowledge Poling Start. This bit field determine the starting point of DTACK input signal polling. DAPS is used only in asynchronous single read or write accesses.

Table continues on the next page...

EIM_CS*n*GCR2 field descriptions (continued)

Field	Description
	NOTE: Since DTACK is an async. signal the start point of DTACK signal polling is at least 3 cycles after the start of access. DAPS is cleared by a hardware reset. Example settings: 0000 3 EIM clk cycle between start of access and first $\overline{\text{DTACK}}$ check 0001 4 EIM clk cycles between start of access and first $\overline{\text{DTACK}}$ check 0010 5 EIM clk cycles between start of access and first $\overline{\text{DTACK}}$ check 0111 10 EIM clk cycles between start of access and first $\overline{\text{DTACK}}$ check 1011 14 EIM clk cycles between start of access and first DTACK check 1111 18 EIM clk cycles between start of access and first DTACK check
3–2 Reserved	This read-only field is reserved and always has the value 0.
ADH	Address hold time - This bit field determine the address hold time after ADV negation when mum = 1 (muxed mode). When mum = 0 this bit has no effect. For read accesses the field determines when the pads direction will be switched. **NOTE:** Reset value for EIM_CS0GCR2 for ADH is 0b10. For EIM_CS1GCR2-EIM_CS5GCR2 reset value is 0b00. 00 0 cycle after ADV negation 01 1 cycle after ADV negation 10 2 cycle after ADV negation 11 Reserved

22.9.3 Chip Select n Read Configuration Register 1 (EIM_CS*n*RCR1)

Address: 21B_8000h base + 8h offset + (24d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		RWSC						0	RADVA			RAL	RADVN		
W																
Reset	0	0							0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	OEA			0	OEN			0	RCSA			0	RCSN		
W																
Reset	0				0	0	0	0	0	0	0	0	0	0	0	0

EIM_CS*n*RCR1 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

EIM_CS*n*RCR1 field descriptions (continued)

Field	Description
29–24 RWSC	Read Wait State Control. This bit field programs the number of wait-states, according to the settings shown below, for synchronous or asynchronous read access to the external device connected to the chip select. When SRD=1 and RFL=0, RWSC indicates the number of burst clock (BCLK) cycles from the start of an access, before the controller can start sample data.Since WAIT signal can be asserted one cycle before the first data can be sampled, the controller starts evaluating the WAIT signal state one cycle before, this is referred as handshake mode or variable latency mode. When SRD=1 and RFL=1, RWSC indicates the number of burst clock (BCLK) cycles from the start of an access, until the external device is ready for data transfer, this is referred as fix latency mode. When SRD=0, RFL bit is ignored, RWSC indicates the asynchronous access length and the number of EIM clock cycles from the start of access until the external device is ready for data transfer. RWSC is cleared by a hardware reset. **NOTE:** The reset value for EIM_CS0RCR1, RWSC[5:0] = 0b011100. For CG1RCR1 - CS1RCR5 the reset value is 0b000000. Example settings: 000000 Reserved 000001 RWSC value is 1 000010 RWSC value is 2 111101 RWSC value is 61 111110 RWSC value is 62 111111 RWSC value is 63
23 Reserved	This read-only field is reserved and always has the value 0.
22–20 RADVA	ADV Assertion. This bit field determines when ADV signal is asserted for synchronous or asynchronous read modes according to the settings shown below. RADVA is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between beginning of access and ADV assertion 001 1 EIM clock cycles between beginning of access and ADV assertion 010 2 EIM clock cycles between beginning of access and ADV assertion 111 7 EIM clock cycles between beginning of access and ADV assertion
19 RAL	Read ADV Low. This bit field determine ADV signal negation time. When RAL=1, RADVN bit field is ignored and ADV signal will stay asserted until end of access. When RAL=0 negation of ADV signal is according to RADVN bit field configuration.
18–16 RADVN	ADV Negation. This bit field determines when ADV signal to memory is negated during read accesses. When SRD=1 (synchronous read mode), ADV negation occurs according to the following formula: (RADVN + RADVA + BCD + BCS + 1) EIM clock cycles from start of access. When asynchronous read mode is applied (SRD=0) and RAL=0 ADV negation occurs according to the following formula: (RADVN + RADVA + 1) EIM clock cycles from start of access. RADVN is cleared by a hardware reset. **NOTE:** The reset value for EIM_CS0RCR1[RADVN] = 0b010. For EIM_CS1RCR1 - EIM_CS5RCR1, the reset value is 0b000. **NOTE:** This field should be configured so ADV negation will occur before the end of access. For ADV negation at the same time with the end of access user should RAL bit.

Table continues on the next page...

EIM_CS*n*RCR1 field descriptions (continued)

Field	Description
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 OEA	OE Assertion. This bit field determines when OE signal are asserted during read cycles (synchronous or asynchronous mode), according to the settings shown below. OEA is cleared by a hardware reset. In muxed mode OE assertion occurs (OEA + RADVN + RADVA + ADH +1) EIM clock cycles from start of access. **NOTE:** The reset value for EIM_CS0RCR1[OEA] is 0b000 if EIM_BOOT[2] = 0. If EIM_BOOT[2] is 1, the reset value for EIM_CS0RCR1[OEA] is 0b010. The reset value of this field for EIM_CS1RCR1 - EIM_CS5RCR1 is 0b000. Example settings: 000 0 EIM clock cycles between beginning of access and OE assertion 001 1 EIM clock cycles between beginning of access and OE assertion 010 2 EIM clock cycles between beginning of access and OE assertion 111 7 EIM clock cycles between beginning of access and OE assertion
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 OEN	OE Negation. This bit field determines when OE signal is negated during read cycles in asynchronous single mode only (SRD=0 and APR = 0), according to the settings shown below. This bit field is ignored when SRD=1. OEN is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between end of access and OE negation 001 1 EIM clock cycles between end of access and OE negation 010 2 EIM clock cycles between end of access and OE negation 111 7 EIM clock cycles between end of access and OE negation
7 Reserved	This read-only field is reserved and always has the value 0.
6–4 RCSA	Read CS Assertion. This bit field determines when CS signal is asserted during read cycles (synchronous or asynchronous mode), according to the settings shown below. RCSA is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between beginning of read access and CS assertion 001 1 EIM clock cycles between beginning of read access and CS assertion 010 2 EIM clock cycles between beginning of read access and CS assertion 111 7 EIM clock cycles between beginning of read access and CS assertion
3 Reserved	This read-only field is reserved and always has the value 0.
RCSN	Read CS Negation. This bit field determines when CS signal is negated during read cycles in asynchronous single mode only (SRD=0 and APR = 0), according to the settings shown below. This bit field is ignored when SRD=1. RCSN is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between end of read access and CS negation 001 1 EIM clock cycles between end of read access and CS negation 010 2 EIM clock cycles between end of read access and CS negation 111 7 EIM clock cycles between end of read access and CS negation

22.9.4 Chip Select n Read Configuration Register 2 (EIM_CS*n*RCR2)

Address: 21B_8000h base + Ch offset + (24d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	APR	PAT			0		RL		0	RBEA			RBE	RBEN		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EIM_CS*n*RCR2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 APR	Asynchronous Page Read. This bit field determine the asynchronous read mode to the external device. When APR=0, the async. read access is done as single word (where word is defined by the DSZ field). when APR=1, the async. read access executed as page read. page size is according to BL field config., RCSN,RBEN,OEN and RADVN are being ignored. APR is cleared by a hardware reset for EIM_CS1GCR1 - EIM_CS5GCR1. **NOTE:** SRD=0 and MUM=0 must apply when APR=1
14–12 PAT	Page Access Time. This bit field is used in Asynchronous Page Read mode only (APR=1). the initial access is set by RWSC as in regular asynchronous mode. the consecutive address assertions width determine by PAT field according to the settings shown below. when APR=0 this field is ignored. PAT is cleared by a hardware reset for EIM_CS1GCR1 - EIM_CS5GCR1. 000 Address width is 2 EIM clock cycles 001 Address width is 3 EIM clock cycles 010 Address width is 4 EIM clock cycles 011 Address width is 5 EIM clock cycles 100 Address width is 6 EIM clock cycles 101 Address width is 7 EIM clock cycles 110 Address width is 8 EIM clock cycles 111 Address width is 9 EIM clock cycles
11–10 Reserved	This read-only field is reserved and always has the value 0.
9–8 RL	Read Latency. This bit field indicates cycle latency when executing a synchronous read operation. The fields holds the feedback clock loop delay in aclk cycle units. This field is cleared by a hardware reset. 00 Feedback clock loop delay is up to 1 cycle for BCD = 0 or 1.5 cycles for BCD != 0 01 Feedback clock loop delay is up to 2 cycles for BCD = 0 or 2.5 cycles for BCD != 0

Table continues on the next page...

EIM_CS*n*RCR2 field descriptions (continued)

Field	Description
	10 Feedback clock loop delay is up to 3 cycles for BCD = 0 or 3.5 cycles for BCD != 0 11 Feedback clock loop delay is up to 4 cycles for BCD = 0 or 4.5 cycles for BCD != 0
7 Reserved	This read-only field is reserved and always has the value 0.
6–4 RBEA	Read BE Assertion. This bit field determines when BE signal is asserted during read cycles (synchronous or asynchronous mode), according to the settings shown below. RBEA is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between beginning of read access and BE assertion 001 1 EIM clock cycles between beginning of read access and BE assertion 010 2 EIM clock cycles between beginning of read access and BE assertion 111 7 EIM clock cycles between beginning of read access and BE assertion
3 RBE	Read BE enable. This bit field determines if BE will be asserted during read access. 0 BE are disabled during read access. 1 BE are enable during read access according to value of RBEA and RBEN bit fields.
RBEN	Read BE Negation. This bit field determines when BE signal is negated during read cycles in asynchronous single mode only (SRD=0 and APR=0), according to the settings shown below. This bit field is ignored when SRD=1. RBEN is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between end of read access and BE negation 001 1 EIM clock cycles between end of read access and BE negation 010 2 EIM clock cycles between end of read access and BE negation 111 7 EIM clock cycles between end of read access and BE negation

22.9.5 Chip Select n Write Configuration Register 1 (EIM_CS*n*WCR1)

Address: 21B_8000h base + 10h offset + (24d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WAL	WBED	WWSC						WADVA			WADVN			WBEA	
Reset	0	0							0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WBEA	WBEN			WEA				WEN			WCSA			WCSN	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EIM_CS*n*WCR1 field descriptions

Field	Description
31 WAL	Write ADV Low. This bit field determine ADV signal negation time in write accesses. When WAL=1, WADVN bit field is ignored and ADV signal will stay asserted until end of access. When WAL=0 negation of ADV signal is according to WADVN bit field configuration.
30 WBED	Write Byte Enable Disable. When asserted this bit prevent from IPP_DO_BE_B[x] to be asserted during write accesses.This bit is cleared by hardware reset.
29–24 WWSC	Write Wait State Control. This bit field programs the number of wait-states, according to the settings shown below, for synchronous or asynchronous write access to the external device connected to the chip select. When SWR=1 and WFL=0, WWSC indicates the number of burst clock (BCLK) cycles from the start of an access, before the memory can sample the first data.Since WAIT signal can be asserted one cycle before the first data can be sampled, the controller starts evaluating the WAIT signal state one cycle before, this is referred as handshake mode or variable latency mode. When SWR=1 and WFL=1, WWSC indicates the number of burst clock (BCLK) cycles from the start of an access, until the external device is ready for data transfer, this is referred as fix latency mode. When SWR=0, WFL bit is ignored, WWSC indicates the asynchronous access length and the number of EIM clock cycles from the start of access until the external device is ready for data transfer. WWSC is cleared by a hardware reset. **NOTE:** The reset value for EIM_CS0WCR1, WWSC[5:0] = 0b011100. For EIM_CS1WCR1 - EIM_CS5WCR1, the reset value of this field is 0b000000. Example settings: 000000 Reserved 000001 WWSC value is 1 000010 WWSC value is 2 000011 WWSC value is 3 111111 WWSC value is 63
23–21 WADVA	ADV Assertion. This bit field determines when ADV signal is asserted for synchronous or asynchronous write modes according to the settings shown below. WADVA is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between beginning of access and ADV assertion 001 1 EIM clock cycles between beginning of access and ADV assertion 010 2 EIM clock cycles between beginning of access and ADV assertion 111 7 EIM clock cycles between beginning of access and ADV assertion
20–18 WADVN	ADV Negation. This bit field determines when ADV signal to memory is negated during write accesses. When SWR=1 (synchronous write mode), ADV negation occurs according to the following formula: (WADVN + WADVA + BCD + BCS + 1) EIM clock cycles. When asynchronous read mode is applied (SWR=0) ADV negation occurs according to the following formula: (WADVN + WADVA + 1) EIM clock cycles. **NOTE:** Reset value for EIM_CS0WCR for WADVN is 0b010. For EIM_CS1WCR - EIM_CS5WCR reset value is 0b000. **NOTE:** This field should be configured so ADV negation will occur before the end of access. For ADV negation at the same time as the end of access, S/W should set the WAL bit.
17–15 WBEA	BE Assertion. This bit field determines when BE signal is asserted during write cycles in async. mode only (SWR=0), according to the settings shown below. BEA is cleared by a hardware reset.

Table continues on the next page...

EIM_CS*n*WCR1 field descriptions (continued)

Field	Description
	NOTE: Reset value for EIM_CS0WCR for WBEA is 0b010. For EIM_CS1WCR - EIM_CS5WCR reset value is 0b000. Example settings: 000 0 EIM clock cycles between beginning of access and BE assertion 001 1 EIM clock cycles between beginning of access and BE assertion 010 2 EIM clock cycles between beginning of access and BE assertion 111 7 EIM clock cycles between beginning of access and BE assertion
14–12 WBEN	BE[3:0] Negation. This bit field determines when BE[3:0] bus signal is negated during write cycles in asynchronous mode only (SWR=0), according to the settings shown below. This bit field is ignored when SWR=1. BEN is cleared by a hardware reset. **NOTE:** Reset value for EIM_CS0WCR for WBEN is 0b010. For EIM_CS1WCR - EIM_CS5WCR reset value is 0b000. Example settings: 000 0 EIM clock cycles between end of access and BE negation 001 1 EIM clock cycles between end of access and BE negation 010 2 EIM clock cycles between end of access and BE negation 111 7 EIM clock cycles between end of access and BE negation
11–9 WEA	WE Assertion. This bit field determines when WE signal is asserted during write cycles (synchronous or asynchronous mode), according to the settings shown below. This bit field is ignored when executing a read access to the external device. WEA is cleared by a hardware reset. **NOTE:** Reset value for EIM_CS0WCR for WEA is 0b010. For EIM_CS1WCR - EIM_CS5WCR reset value is 0b000. Example settings: 000 0 EIM clock cycles between beginning of access and WE assertion 001 1 EIM clock cycles between beginning of access and WE assertion 010 2 EIM clock cycles between beginning of access and WE assertion 111 7 EIMclock cycles between beginning of access and WE assertion
8–6 WEN	WE Negation. This bit field determines when WE signal is negated during write cycles in asynchronous mode only (SWR=0), according to the settings shown below. This bit field is ignored when SWR=1. WEN is cleared by a hardware reset. **NOTE:** Reset value for EIM_CS0WCR for WEN is 0b010. For EIM_CS1WCR - EIM_CS5WCR reset value is 0b000. Example settings: 000 0 EIM clock cycles between end of access and WE negation 001 1 EIM clock cycles between end of access and WE negation 010 2 EIM clock cycles between end of access and WE negation 111 7 EIM clock cycles between end of access and WE negation
5–3 WCSA	Write CS Assertion. This bit field determines when CS signal is asserted during write cycles (synchronous or asynchronous mode), according to the settings shown below.this bit field is ignored when executing a read access to the external device. WCSA is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between beginning of write access and CS assertion

Table continues on the next page...

EIM_CS*n*WCR1 field descriptions (continued)

Field	Description
	001 1 EIM clock cycles between beginning of write access and CS assertion 010 2 EIM clock cycles between beginning of write access and CS assertion 111 7 EIMclock cycles between beginning of write access and CS assertion
WCSN	Write CS Negation. This bit field determines when CS signal is negated during write cycles in asynchronous mode only (SWR=0), according to the settings shown below. This bit field is ignored when SWR=1. WCSN is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between end of read access and CS negation 001 1 EIM clock cycles between end of read access and CS negation 010 2 EIM clock cycles between end of read access and CS negation 111 7 EIM clock cycles between end of read access and CS negation

22.9.6 Chip Select n Write Configuration Register 2 (EIM_CS*n*WCR2)

Address: 21B_8000h base + 14h offset + (24d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

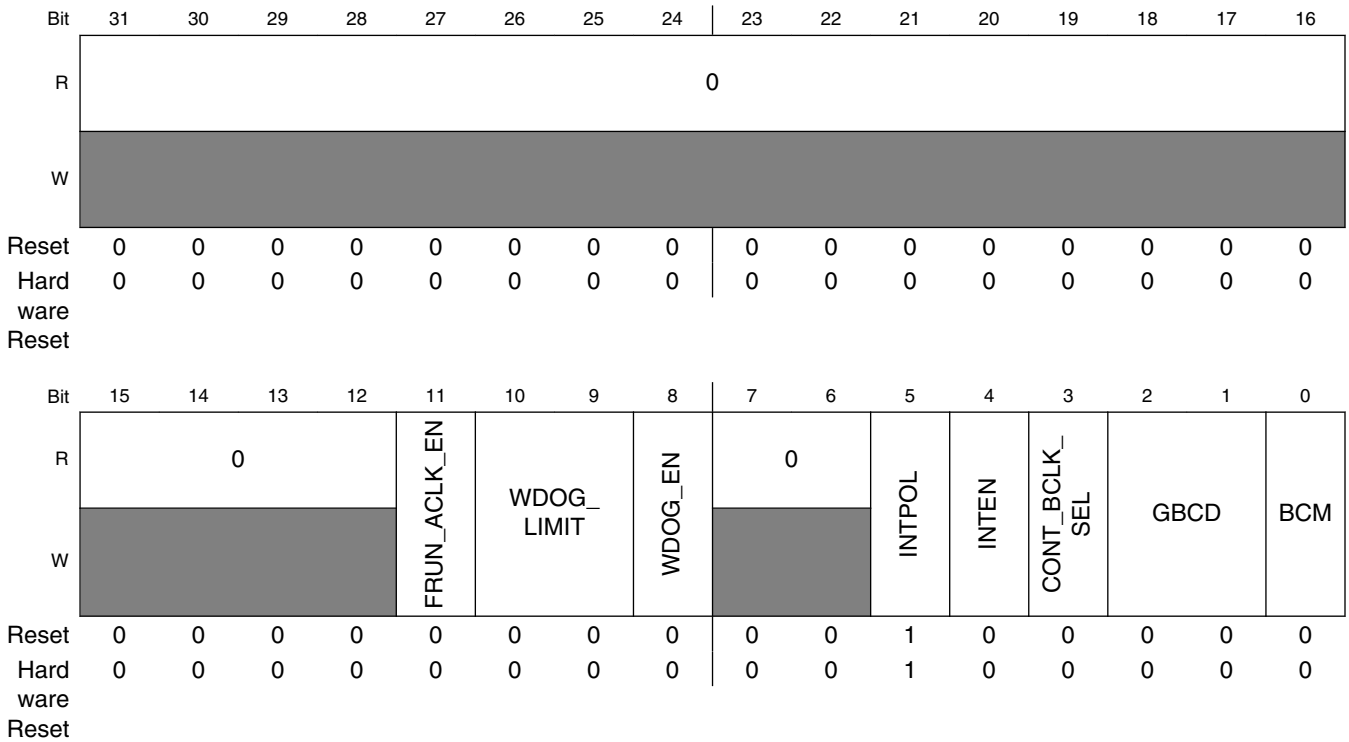
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															WBCDD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EIM_CS*n*WCR2 field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 WBCDD	Write Burst Clock Divisor Decrement. If this bit is asserted and BCD value is 0 sync. write access will be preformed as if BCD value is 1.When this bit is negated or BCD value is not 0 this bit has no affect. This bit is cleared by hardware reset.

22.9.7 EIM Configuration Register (EIM_WCR)

Address: 21B_8000h base + 90h offset = 21B_8090h



EIM_WCR field descriptions

Field	Description
31–12 Reserved	Reserved This read-only field is reserved and always has the value 0.
11 FRUN_ACLK_EN	Free run ACLK enable
10–9 WDOG_LIMIT	Memory Watchdog (WDOG) cycle limit. This bit field determines the number of BCLK cycles (ACLK cycles in dtack mode) before the WDOG counter terminates the access and send an error response to the master. 00 128 BCLK cycles 01 256 BCLK cycles 10 512 BCLK cycles 11 1024 BCLK cycles
8 WDOG_EN	Memory WDOG enable. This bit controls the operation of the wdog counter that terminates the EIM access. 0 Memory WDOG is Disabled 1 Memory WDOG is Enabled

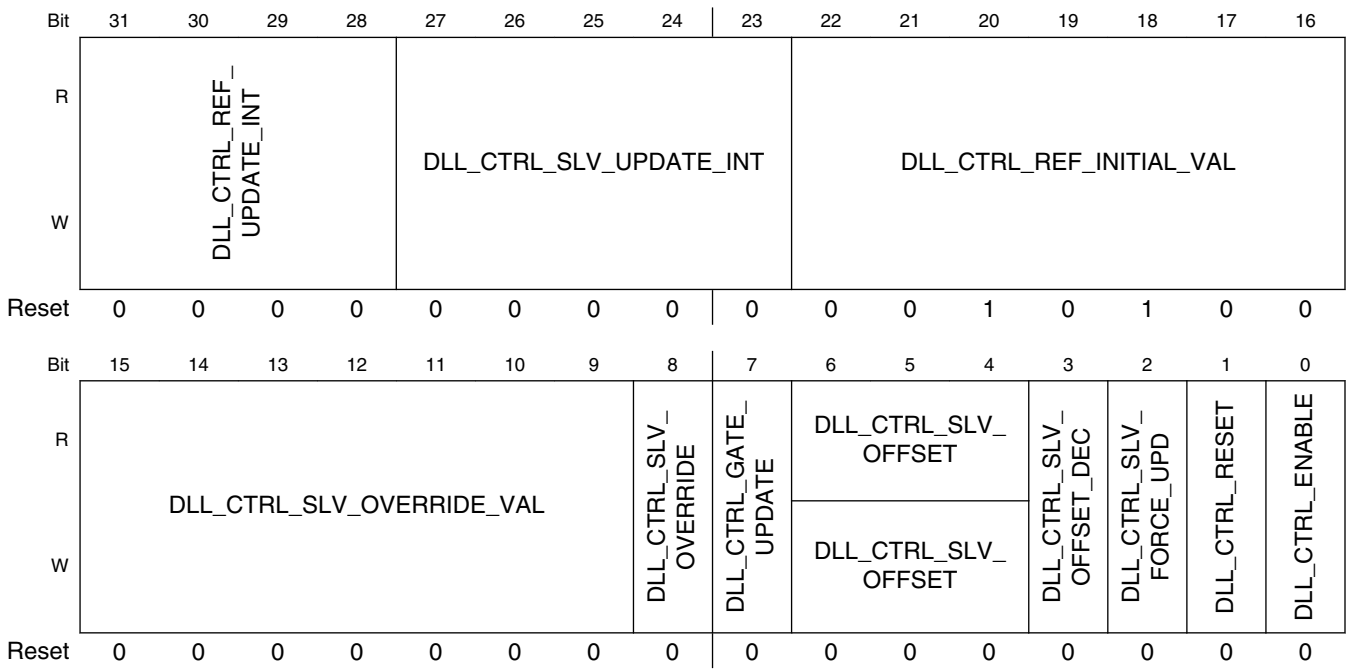
Table continues on the next page...

EIM_WCR field descriptions (continued)

Field	Description
7–6 Reserved	This read-only field is reserved and always has the value 0.
5 INTPOL	Interrupt Polarity. This bit field determines the polarity of the external device interrupt. 0 External interrupt polarity is active low 1 External interrupt polarity is active high
4 INTEN	Interrupt Enable. When this bit is set the External signal RDY_INT as active interrupt. When interrupt occurs, INT bit at the WCR will be set and t EIM_EXT_INT signal will be asserted correspondingly. This bit is cleared by a hardware reset. 0 External interrupt Disable 1 External interrupt Enable
3 CONT_BCLK_ SEL	Continuous BCLK select When this bit is set BCLK pin output continuous clock. Otherwize, BCLK will output clock only when nesserary. 0 BCLK When nesserary 1 BCLK Continuous
2–1 GBCD	General Burst Clock Divisor. When BCM bit is set, this bit field contains the value used to program the burst clock divisor for Continuous BCLK generation. The other BCD bit fields for each chip select are ignored. It is used to divide the internal AXI bus frequency. When BCM=0 GBCD bit field has no influence. GBCD is cleared by a hardware reset. 00 Divide EIM clock by 1 01 Divide EIM clock by 2 10 Divide EIM clock by 3 11 Divide EIM clock by 4
0 BCM	Burst Clock Mode. This bit selects the burst clock mode of operation. It is used for system debug mode. BCM is cleared by a hardware reset. **NOTE:** The BCLK frequency in this mode is according to GBCD bit field. **NOTE:** The BCLK phase is opposite to the EIM clock in this mode if GBCD is 0. **NOTE:** This bit should be used only in async. accesses. No sync access can be executed if this bit is set. **NOTE:** When this bit is set bcd field shouldn't be configured to 0. 0 The burst clock runs only when accessing a chip select range with the SWR/SRD bits set. When the burst clock is not running it remains in a logic 0 state. When the burst clock is running it is configured by the BCD and BCS bit fields in the chip select Configuration Register. 1 The burst clock runs whenever ACLK is active (independent of chip select configuration)

22.9.8 DLL Control Register (EIM_DCR)

Address: 21B_8000h base + 94h offset = 21B_8094h



EIM_DCR field descriptions

Field	Description
31–28 DLL_CTRL_ REF_UPDATE_ INT	Reference DLL Update Interval DLL control loop update interval. The interval cycle is (2 + REF_UPDATE_INT) * ref_clock. By default, the DLL control loop shall update every two ref_clock cycles. It should be noted that increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that may effect the delay (such as voltage and temperature)
27–23 DLL_CTRL_ SLV_UPDATE_ INT	Slave DLL Update Interval If default 0 is used, it means 256 cycles of ref_clock. A value of 0x0f results in 15 cycles and so on. Note that software can always cause an update of the slave-delay line using the SLV_FORCE_UPDATE register. Note that the slave delay line will also update automatically when the reference DLL transitions to a locked state (from an un-locked state).
22–16 DLL_CTRL_ REF_INITIAL_ VAL	This field is used to select the initial value of reference chain before DLL enabled. It's recommended to set the initial value close to the locked value to accelerate the locking.
15–9 DLL_CTRL_ SLV_ OVERRIDE_VAL	When SLV_OVERRIDE=1 This field is used to select 1 of 128 physical taps manually. A value of 0 selects tap 1, and a value of 0x7f selects tap 128.
8 DLL_CTRL_ SLV_OVERRIDE	Set this bit to 1 to Enable manual override for slave delay chain using SLV_OVERRIDE_VAL; to set 0 to disable manual override. This feature does not require the DLL to be enabled using the ENABLE bit. In fact to reduce power, if SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE=0

Table continues on the next page...

EIM_DCR field descriptions (continued)

Field	Description
7 DLL_CTRL_ GATE_UPDATE	Set this bit to 1 to force DLL not update from now on. Since when clock exists, glitches might appear during update. This bit is used by software if we met such kind of condition. Set it to 0 to let DLL update automatically
6–4 DLL_CTRL_ SLV_OFFSET	OFFSET value for DLL_CTRL_SLV_SEL
3 DLL_CTRL_ SLV_OFFSET_ DEC	Slave Chain Offset Decrease Decrease(or increase) the value defined by DLL_CTRL_SLV_OFFSET when calculating DLL_STS_SLV_SEL 0 DLL_STS_SLV_SEL = DLL_STS_REF_SEL + DLL_CTRL_SLV_OFFSET 1 DLL_STS_SLV_SEL = DLL_STS_REF_SEL - DLL_CTRL_SLV_OFFSET
2 DLL_CTRL_ SLV_FORCE_ UPD	Setting this bit to 1, forces the slave delay line to update to the DLL calibrated value immediately. The slave delay line shall update automatically based on the SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the slave-line update can only occur if SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered). Be sure to use it when EIM is idle.
1 DLL_CTRL_ RESET	DLL Reset Bit Setting this bit to 1 force a reset on DLL. This will cause the DLL to lose lock and re-lock. This signal is used by the DLL as edge-sensitive, so in order to create a subsequent reset, RESET must be taken low and then asserted again
0 DLL_CTRL_ ENABLE	Set this bit to 1 to enable the DLL and delay chain; otherwise; set to 0 to bypasses DLL. Note that using the slave delay line override feature with SLV_OVERRIDE and SLV_OVERRIDE VAL, the DLL does not need to be enabled

22.9.9 DLL Status Register (EIM_DSR)

Address: 21B_8000h base + 98h offset = 21B_8098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DLL_STS_REF_SEL								DLL_STS_SLV_SEL						DLL_STS_REF_LOCK	DLL_STS_SLV_LOCK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EIM_DSR field descriptions

Field	Description
31–16 Reserved	Reserved This read-only field is reserved and always has the value 0.
15–9 DLL_STS_REF_SEL	Reference delay line select taps. Be noted this is encoded by 7 bits for 127taps.
8–2 DLL_STS_SLV_SEL	Slave delay line select status. This is the instant value generated from reference chain. Since only when ref_lock is detected can the reference chain get updated, this value should be the right value next be update to the slave line when reference is locked.
1 DLL_STS_REF_LOCK	Reference DLL lock status. This signifies that the DLL has detected and locked to feedback BCLK, allowing the slave delay-line to perform programmed clock delays
0 DLL_STS_SLV_LOCK	Slave delay-line lock status. This signifies that a valid delay has been set to the slave-delay line and that the slave-delay line is implementing the programmed delay value

22.9.10 EIM IP Access Register (EIM_WIAR)

Address: 21B_8000h base + 9Ch offset = 21B_809Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											ACLK_EN	ERRST	INT	IPS_ACK	IPS_REQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

EIM_WIAR field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 ACLK_EN	ACLK enable. This bit gates the ACLK for the EIM except from FFs that get ipg_aclk_s. After reset ACLK is enabled. 0 ACLK is disabled 1 ACLK is enabled

Table continues on the next page...

EIM_WIAR field descriptions (continued)

Field	Description
3 ERRST	READY After Reset. This bit controls the initial ready/busy status for external devices on CS0 immediately after hardware reset. This is a sticky bit which is cleared once the RDY_INT signal is asserted by the external device. When ERRST = 1 the first fetch access from EIM to the external device located on CS0 will be pending until RDY_INT signal indicates that the external device is ready, then EIM will execute the access. **NOTE:** Reset value for ERRST is EIM_BOOT[4]. 0 RDY_INT After Reset Disable 1 RDY_INT After Reset Enable
2 INT	Interrupt. This bit indicates interrupt assertion by an external device according to RDY_INT signal. When polling this bit, INT=0 indicates interrupt not occurred and INT=1 indicates assertion of the external device interrupt. This bit is cleared by a hardware reset.
1 IPS_ACK	IPS ACK. The EIM is ready for ips access. There is no active AXI access and no new AXI access is accepted till this bit is cleared. This bit is cleared by the master after it completes the ips accesses. 0 Master cannot access ips. 1 Master can access ips.
0 IPS_REQ	IPS request. The Master requests to access one of the IPS registers.Dutring such access the EIM should not preform any AXI/memory accesses. The EIM finishes the AXI accesses that already starts and asserts the IPS_ACK bit. 0 No Master requests ips access 1 Master requests ips access

22.9.11 Error Address Register (EIM_EAR)

Address: 21B_8000h base + A0h offset = 21B_80A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Error_ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EIM_EAR field descriptions

Field	Description
Error_ADDR	Error Address. This bit field holds the AXI address of the last access that caused error. This register is read only register.

Chapter 23
10/100/1000-Mbps Ethernet MAC (ENET)

23.1 Introduction

The MAC-NET core, in conjunction with a 10/100/1000/1000[1]-Mbit/s MAC, implements layer 3 network acceleration functions. These functions are designed to accelerate the processing of various common networking protocols, such as IP, TCP, UDP, and ICMP, providing wire speed services to client applications.

23.2 Overview

The core implements a triple-speed 10/100/1000/1000[1]-Mbit/s Ethernet MAC compliant with the IEEE802.3-2002 standard. The MAC layer provides compatibility with half- or full-duplex 10/100-Mbit/s and full-duplex gigabit[1] Ethernet LANs.

The MAC operation is fully programmable and can be used in Network Interface Card (NIC), bridging, or switching applications. The core implements the remote network monitoring (RMON) counters according to IETF RFC 2819.

The core also implements a hardware acceleration block to optimize the performance of network controllers providing TCP/IP, UDP, and ICMP protocol services. The acceleration block performs critical functions in hardware, which are typically implemented with large software overhead.

The core implements programmable embedded FIFOs that can provide buffering on the receive path for lossless flow control.

Advanced power management features are available with magic packet detection and programmable power-down modes.

1. Actual throughput is greater than 100 Mbit/s but significantly less than 1000 Mbit/s.

A unified DMA (uDMA), internal to the ENET module, optimizes data transfer between the ENET core and the SoC, and supports an enhanced buffer descriptor programming model to support IEEE 1588 functionality.

The programmable Ethernet MAC with IEEE 1588 integrates a standard IEEE 802.3 Ethernet MAC with a time-stamping module. The IEEE 1588 standard provides accurate clock synchronization for distributed control nodes for industrial automation applications.

23.2.1 Features

The MAC-NET core includes the following features.

23.2.1.1 Ethernet MAC features

- Implements the full 802.3 specification with preamble/SFD generation, frame padding generation, CRC generation and checking
- Supports zero-length preamble
- Dynamically configurable to support 10/100-Mbit/s and gigabit[2] operation
- Supports 10/100 Mbit/s full-duplex and configurable half-duplex operation
- Supports gigabit[2] full-duplex operation
- Compliant with the AMD magic packet detection with interrupt for node remote power management
- Seamless interface to commercial ethernet PHY devices via one of the following:
  - a 4-bit Media Independent Interface (MII) operating at 2.5/25 MHz.
  - a 4-bit non-standard MII-Lite (MII without the CRS and COL signals) operating at 2.5/25 MHz.
  - a 2-bit Reduced MII (RMII) operating at 50 MHz.
  - a (double data rate) 4-bit Reduced GMII (RGMII) operating at 125 MHz.
- Simple 64-Bit FIFO user-application interface
- CRC-32 checking at full speed with optional forwarding of the frame check sequence (FCS) field to the client
- CRC-32 generation and append on transmit or forwarding of user application provided FCS selectable on a per-frame basis
- In full-duplex mode:
  - Implements automated pause frame (802.3 x31A) generation and termination, providing flow control without user application intervention
  - Pause quanta used to form pause frames — dynamically programmable
  - Pause frame generation additionally controllable by user application offering flexible traffic flow control

2. Actual throughput is greater than 100 Mbit/s but significantly less than 1000 Mbit/s.

    - Optional forwarding of received pause frames to the user application
    - Implements standard flow-control mechanism
- In half-duplex mode: provides full collision support, including jamming, backoff, and automatic retransmission
- Supports VLAN-tagged frames according to IEEE 802.1Q
- Programmable MAC address: Insertion on transmit; discards frames with mismatching destination address on receive (except broadcast and pause frames)
- Programmable promiscuous mode support to omit MAC destination address checking on receive
- Multicast and unicast address filtering on receive based on 64-entry hash table, reducing higher layer processing load
- Programmable frame maximum length providing support for any standard or proprietary frame length
- Statistics indicators for frame traffic and errors (alignment, CRC, length) and pause frames providing for IEEE 802.3 basic and mandatory management information database (MIB) package and remote network monitoring (RFC 2819)
- Simple handshake user application FIFO interface with fully programmable depth and threshold levels
- Provides separate status word for each received frame on the user interface providing information such as frame length, frame type, VLAN tag, and error information
- Multiple internal loopback options
- MDIO master interface for PHY device configuration and management supports two programmable MDIO base addresses, and standard (IEEE 802.3 Clause 22) and extended (Clause 45) MDIO frame formats
- Supports legacy FEC buffer descriptors

23.2.1.2 IP protocol performance optimization features

- Operates on TCP/IP and UDP/IP and ICMP/IP protocol data or IP header only
- Enables wire-speed processing
- Supports IPv4 and IPv6
- Transparent passing of frames of other types and protocols
- Supports VLAN tagged frames according to IEEE 802.1q with transparent forwarding of VLAN tag and control field
- Automatic IP-header and payload (protocol specific) checksum calculation and verification on receive

- Automatic IP-header and payload (protocol specific) checksum generation and automatic insertion on transmit configurable on a per-frame basis
- Supports IP and TCP, UDP, ICMP data for checksum generation and checking
- Supports full header options for IPv4 and TCP protocol headers
- Provides IPv6 support to datagrams with base header only — datagrams with extension headers are passed transparently unmodifed/unchecked
- Provides statistics information for received IP and protocol errors
- Configurable automatic discard of erroneous frames
- Configurable automatic host-to-network (RX) and network-to-host (TX) byte order conversion for IP and TCP/UDP/ICMP headers within the frame
- Configurable padding remove for short IP datagrams on receive
- Configurable Ethernet payload alignment to allow for 32-bit word-aligned header and payload processing
- Programmable store-and-forward operation with clock and rate decoupling FIFOs

23.2.1.3 IEEE 1588 features

- Supports all IEEE 1588 frames.
- Allows reference clock to be chosen independently of network speed.
- Software-programmable precise time-stamping of ingress and egress frames
- Timer monitoring capabilities for system calibration and timing accuracy management
- Precise time-stamping of external events with programmable interrupt generation
- Programmable event and interrupt generation for external system control
- Supports hardware- and software-controllable timer synchronization.
- Provides a 4-channel IEEE 1588 timer. Each channel supports input capture and output compare using the 1588 counter.

23.2.2 Block diagram

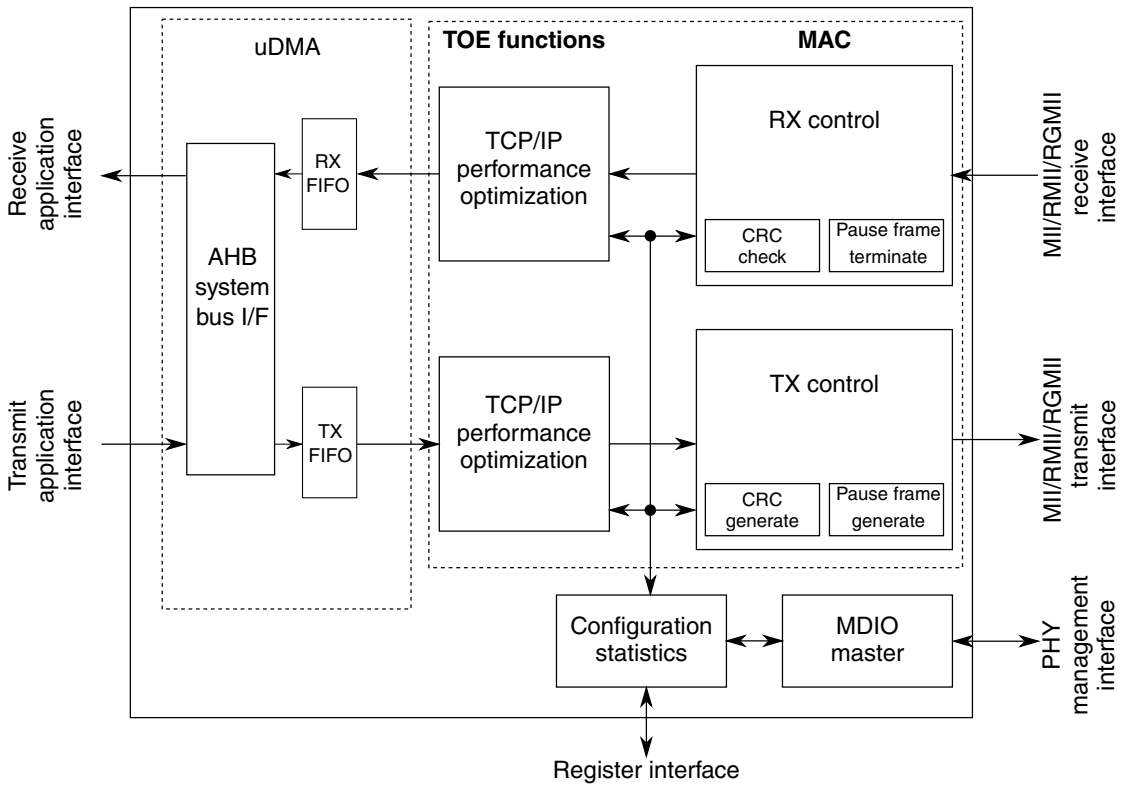


Figure 23-1. Ethernet MAC-NET core block diagram

23.3 External Signals

The table found here describes the external signals of ENET.

Table 23-1. ENET External Signals

Signal	Description	Mode	Pad	Alt Mode	Direction
ENET_1588_EVENT0_IN	Capture/compare block input/output event bus signal. When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the corresponding ENET_TCCRn register for inspection by software. When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the	MII/RMII/ RGMII	ENET_TXD1	ALT4	I

Table continues on the next page...

Table 23-1. ENET External Signals (continued)

Signal	Description	Mode	Pad	Alt Mode	Direction
	corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set.				
ENET_1588_EVENT0_OUT	Capture/compare block input/output event bus signal. When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the corresponding ENET_TCCRn register for inspection by software. When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set.	MII/RMII/ RGMII	GPIO_19	ALT1	O
ENET_1588_EVENT1_IN	Capture/compare block input/output event bus signal. When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the corresponding ENET_TCCRn register for inspection by software. When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set.	MII/RMII/ RGMII	ENET_MDC	ALT4	I
ENET_1588_EVENT1_OUT	Capture/compare block input/output event bus signal. When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the corresponding ENET_TCCRn register for inspection by software. When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set.	MII/RMII/ RGMII	ENET_MDIO	ALT4	O
ENET_1588_EVENT2_IN	Capture/compare block input/output event bus signal. When configured for capture and a rising edge is detected,	MII/RMII/ RGMII	GPIO_16	ALT1	I

Table continues on the next page...

Table 23-1. ENET External Signals (continued)

Signal	Description	Mode	Pad	Alt Mode	Direction
	the current timer value is latched and transferred into the corresponding ENET_TCCRn register for inspection by software. When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set.				
ENET_1588_EVENT2_OUT	Capture/compare block input/output event bus signal. When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the corresponding ENET_TCCRn register for inspection by software. When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set.	MII/RMII/ RGMII	ENET_RX_ER	ALT4	O
ENET_1588_EVENT3_IN	Capture/compare block input/output event bus signal. When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the corresponding ENET_TCCRn register for inspection by software. When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set.	MII/RMII/ RGMII	GPIO_17	ALT1	I
ENET_1588_EVENT3_OUT	Capture/compare block input/output event bus signal. When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the corresponding ENET_TCCRn register for inspection by software.	MII/RMII/ RGMII	ENET_RXD1	ALT4	O

Table continues on the next page...

Table 23-1. ENET External Signals (continued)

Signal	Description	Mode	Pad	Alt Mode	Direction
	When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set.				
ENET_COL	Asserted upon detection of a collision and remains asserted while the collision persists. This signal is not defined for full duplex mode.	MII	KEY_ROW1	ALT1	I
ENET_CRS	Carrier sense. When asserted, indicates transmit or receive medium is not idle. In RMII mode, this signal is present on the RMII_CRS_DV pin.	MII	KEY_COL3	ALT1	I
ENET_MDC	Output clock provides a timing reference to the PHY for data transfers on the MDIO signal.	MII/RMII/ RGMII	ENET_MDC	ALT1	O
			KEY_COL2	ALT4	
ENET_MDIO	Transfers control information between the external PHY and the media-access controller. Data is synchronous to MDC. This signal is an input after reset.	MII/RMII/ RGMII	ENET_MDIO	ALT1	IO
			KEY_COL1	ALT1	
ENET_REF_CLK	In RMII mode, this signal is the reference clock for receive, transmit, and the control interface.	RMII	GPIO_16	ALT2	I
			RGMII_TX_CTL	ALT7	
ENET_RX_CLK	In MII mode, provides a timing reference for RX_EN, RX_DATA[3:0], and RX_ER.	MII	GPIO_18	ALT1	I
ENET_RX_DATA0	Contains the Ethernet input data transferred from the PHY to the media-access controller when RX_EN is asserted.	MII/RMII	ENET_RXD0	ALT1	I
ENET_RX_DATA1	Contains the Ethernet input data transferred from the PHY to the media-access controller when RX_EN is asserted.	MII/RMII	ENET_RXD1	ALT1	I
ENET_RX_DATA2	Contains the Ethernet input data transferred from the PHY to the media-access controller when RX_EN is asserted.	MII	KEY_COL2	ALT1	I
ENET_RX_DATA3	Contains the Ethernet input data transferred from the PHY to the media-access controller when RX_EN is asserted.	MII	KEY_COL0	ALT1	I

Table continues on the next page...

Table 23-1. ENET External Signals (continued)

Signal	Description	Mode	Pad	Alt Mode	Direction
ENET_RX_EN	Asserting this input indicates the PHY has valid nibbles present on the MII. RX_EN must remain asserted from the first recovered nibble of the frame through to the last nibble. Asserting RX_EN must start no later than the SFD and exclude any EOF. In RMII mode, this pin also generates the CRS signal.	MII/RMII	ENET_CRS_DV	ALT1	I
ENET_RX_ER	When asserted with RXDV, indicates the PHY detects an error in the current frame.	MII/RMII	ENET_RX_ER	ALT1	I
ENET_TX_CLK	Input clock In MII mode, provides a timing reference for TX_EN, TX_DATA[3:0], and TX_ER. IN RGMII mode, provides the external 125 MHz reference clock input.	MII/RGMII	ENET_REF_CLK	ALT1	I
ENET_TX_DATA0	Serial output Ethernet data. Only valid during TX_EN assertion.	MII/RMII	ENET_TXD0	ALT1	O
ENET_TX_DATA1	Serial output Ethernet data. Only valid during TX_EN assertion.	MII/RMII	ENET_TXD1	ALT1	O
ENET_TX_DATA2	Serial output Ethernet data. Only valid during TX_EN assertion.	MII	KEY_ROW2	ALT1	O
ENET_TX_DATA3	Serial output Ethernet data. Only valid during TX_EN assertion.	MII	KEY_ROW0	ALT1	O
ENET_TX_EN	Indicates when valid nibbles are present on the MII. This signal is asserted with the first nibble of a preamble and is deasserted before the first TX_CLK following the final nibble of the frame.	MII/RMII	ENET_TX_EN	ALT1	O
ENET_TX_ER	When asserted for one or more clock cycles while TXEN is also asserted, PHY sends one or more illegal symbols.	MII/RMII	GPIO_19	ALT6	O
RGMII_RD0	Contains the Ethernet input data transferred from the PHY to the media-access controller when RX_EN is asserted.	RGMII	RGMII_RD0	ALT1	I
RGMII_RD1	Contains the Ethernet input data transferred from the PHY to the media-access controller when RX_EN is asserted.	RGMII	RGMII_RD1	ALT1	I
RGMII_RD2	Contains the Ethernet input data transferred from the PHY to the media-access controller when RX_EN is asserted.	RGMII	RGMII_RD2	ALT1	I

Table continues on the next page...

Table 23-1. ENET External Signals (continued)

Signal	Description	Mode	Pad	Alt Mode	Direction
RGMII_RD3	Contains the Ethernet input data transferred from the PHY to the media-access controller when RX_EN is asserted.	RGMII	RGMII_RD3	ALT1	I
RGMII_RXC	In RGMII mode, provides a timing reference for RX_DATA[3:0] and RX_CTL.	RGMII	RGMII_RXC	ALT1	I
RGMII_RX_CTL	In RGMII mode, contains RX_EN on the rising edge of RGMII_RXC, and a logical derivative of RX_EN and RX_ER (RX_EN XOR RX_ER) on the falling edge of RGMII_RXC.	RGMII	RGMII_RX_CTL	ALT1	I
RGMII_TD0	Serial output Ethernet data. Only valid during TX_EN assertion.	RGMII	RGMII_TD0	ALT1	O
RGMII_TD1	Serial output Ethernet data. Only valid during TX_EN assertion.	RGMII	RGMII_TD1	ALT1	O
RGMII_TD2	Serial output Ethernet data. Only valid during TX_EN assertion.	RGMII	RGMII_TD2	ALT1	O
RGMII_TD3	Serial output Ethernet data. Only valid during TX_EN assertion.	RGMII	RGMII_TD3	ALT1	O
RGMII_TXC	In RGMII mode, provides a timing reference for TX_DATA[3:0] and TX_CTL.	RGMII	RGMII_TXC	ALT1	O
RGMII_TX_CTL	In RGMII mode, contains TX_EN on the rising edge of RGMII_TXC, and a logical derivative of TX_EN and TX_ER (TX_EN XOR TX_ER) on the falling edge of RGMII_TXC.	RGMII	RGMII_TX_CTL	ALT1	O

23.4 Clocks

The table found here describes the clock sources for ENET. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 23-2. ENET Clocks

Clock name	Clock Root	Description
ipg_clk	ahb_clk_root	Module clock
ipg_clk_mac0	ahb_clk_root	MAC peripheral clock
ipg_clk_mac0_s	ipg_clk_root	MAC peripheral access clock
ipg_clk_s	ipg_clk_root	Peripheral access clock
ipg_clk_time	ipg_clk_root	Peripheral clock
mac0_rxmem_clk	ahb_clk_root	MAC receive memory clock
mac0_txmem_clk	ahb_clk_root	MAC transmit memory clock

NOTE

The ENET module requires ahb_clk_root to be 125 MHz or greater.

23.5 Memory map/register definition

ENET registers must be read or written with 32-bit accesses. Non-32 bit accesses will terminate with an error.

Reserved bits should be written with 0 and ignored on read. Unused registers read zero and a write has no effect.

This table shows Ethernet registers organization.

Table 23-3. Register map summary

Offset Address	Section	Description
0x0000 – 0x01FF	Configuration	Core control and status registers
0x0200 – 0x03FF	Statistics counters	MIB and Remote Network Monitoring (RFC 2819) registers
0x0400 – 0x0430	1588 control	1588 adjustable timer (TSM) and 1588 frame control
0x0600 – 0x07FC	Capture/Compare block	Registers for the Capture/Compare block

ENET memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_8004	Interrupt Event Register (ENET_EIR)	32	w1c	0000_0000h	23.5.1/1072
218_8008	Interrupt Mask Register (ENET_EIMR)	32	R/W	0000_0000h	23.5.2/1075
218_8010	Receive Descriptor Active Register (ENET_RDAR)	32	R/W	0000_0000h	23.5.3/1078
218_8014	Transmit Descriptor Active Register (ENET_TDAR)	32	R/W	0000_0000h	23.5.4/1078
218_8024	Ethernet Control Register (ENET_ECR)	32	R/W	F000_0000h	23.5.5/1080
218_8040	MII Management Frame Register (ENET_MMFR)	32	R/W	0000_0000h	23.5.6/1082
218_8044	MII Speed Control Register (ENET_MSCR)	32	R/W	0000_0000h	23.5.7/1083
218_8064	MIB Control Register (ENET_MIBC)	32	R/W	C000_0000h	23.5.8/1085
218_8084	Receive Control Register (ENET_RCR)	32	R/W	05EE_0001h	23.5.9/1087
218_80C4	Transmit Control Register (ENET_TCR)	32	R/W	0000_0000h	23.5.10/ 1090
218_80E4	Physical Address Lower Register (ENET_PALR)	32	R/W	0000_0000h	23.5.11/ 1092
218_80E8	Physical Address Upper Register (ENET_PAUR)	32	R/W	0000_8808h	23.5.12/ 1092

Table continues on the next page...

ENET memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_80EC	Opcode/Pause Duration Register (ENET_OPD)	32	R/W	0001_0000h	23.5.13/ 1093
218_8118	Descriptor Individual Upper Address Register (ENET_IAUR)	32	R/W	0000_0000h	23.5.14/ 1093
218_811C	Descriptor Individual Lower Address Register (ENET_IALR)	32	R/W	0000_0000h	23.5.15/ 1094
218_8120	Descriptor Group Upper Address Register (ENET_GAUR)	32	R/W	0000_0000h	23.5.16/ 1094
218_8124	Descriptor Group Lower Address Register (ENET_GALR)	32	R/W	0000_0000h	23.5.17/ 1095
218_8144	Transmit FIFO Watermark Register (ENET_TFWR)	32	R/W	0000_0000h	23.5.18/ 1095
218_8180	Receive Descriptor Ring Start Register (ENET_RDSR)	32	R/W	0000_0000h	23.5.19/ 1096
218_8184	Transmit Buffer Descriptor Ring Start Register (ENET_TDSR)	32	R/W	0000_0000h	23.5.20/ 1097
218_8188	Maximum Receive Buffer Size Register (ENET_MRBR)	32	R/W	0000_0000h	23.5.21/ 1098
218_8190	Receive FIFO Section Full Threshold (ENET_RSFL)	32	R/W	0000_0000h	23.5.22/ 1099
218_8194	Receive FIFO Section Empty Threshold (ENET_RSEM)	32	R/W	0000_0000h	23.5.23/ 1099
218_8198	Receive FIFO Almost Empty Threshold (ENET_RAEM)	32	R/W	0000_0004h	23.5.24/ 1100
218_819C	Receive FIFO Almost Full Threshold (ENET_RAFL)	32	R/W	0000_0004h	23.5.25/ 1100
218_81A0	Transmit FIFO Section Empty Threshold (ENET_TSEM)	32	R/W	0000_0000h	23.5.26/ 1101
218_81A4	Transmit FIFO Almost Empty Threshold (ENET_TAEM)	32	R/W	0000_0004h	23.5.27/ 1101
218_81A8	Transmit FIFO Almost Full Threshold (ENET_TAFL)	32	R/W	0000_0008h	23.5.28/ 1102
218_81AC	Transmit Inter-Packet Gap (ENET_TIPG)	32	R/W	0000_000Ch	23.5.29/ 1102
218_81B0	Frame Truncation Length (ENET_FTRL)	32	R/W	0000_07FFh	23.5.30/ 1103
218_81C0	Transmit Accelerator Function Configuration (ENET_TACC)	32	R/W	0000_0000h	23.5.31/ 1103
218_81C4	Receive Accelerator Function Configuration (ENET_RACC)	32	R/W	0000_0000h	23.5.32/ 1104
218_8200	Reserved Statistic Register (ENET_RMON_T_DROP)	32	R	0000_0000h	23.5.33/ 1105
218_8204	Tx Packet Count Statistic Register (ENET_RMON_T_PACKETS)	32	R	0000_0000h	23.5.34/ 1106

Table continues on the next page...

ENET memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_8208	Tx Broadcast Packets Statistic Register (ENET_RMON_T_BC_PKT)	32	R	0000_0000h	23.5.35/ 1106
218_820C	Tx Multicast Packets Statistic Register (ENET_RMON_T_MC_PKT)	32	R	0000_0000h	23.5.36/ 1107
218_8210	Tx Packets with CRC/Align Error Statistic Register (ENET_RMON_T_CRC_ALIGN)	32	R	0000_0000h	23.5.37/ 1107
218_8214	Tx Packets Less Than Bytes and Good CRC Statistic Register (ENET_RMON_T_UNDERSIZE)	32	R	0000_0000h	23.5.38/ 1108
218_8218	Tx Packets GT MAX_FL bytes and Good CRC Statistic Register (ENET_RMON_T_OVERSIZE)	32	R	0000_0000h	23.5.39/ 1108
218_821C	Tx Packets Less Than 64 Bytes and Bad CRC Statistic Register (ENET_RMON_T_FRAG)	32	R	0000_0000h	23.5.40/ 1109
218_8220	Tx Packets Greater Than MAX_FL bytes and Bad CRC Statistic Register (ENET_RMON_T_JAB)	32	R	0000_0000h	23.5.41/ 1109
218_8224	Tx Collision Count Statistic Register (ENET_RMON_T_COL)	32	R	0000_0000h	23.5.42/ 1110
218_8228	Tx 64-Byte Packets Statistic Register (ENET_RMON_T_P64)	32	R	0000_0000h	23.5.43/ 1110
218_822C	Tx 65- to 127-byte Packets Statistic Register (ENET_RMON_T_P65TO127)	32	R	0000_0000h	23.5.44/ 1111
218_8230	Tx 128- to 255-byte Packets Statistic Register (ENET_RMON_T_P128TO255)	32	R	0000_0000h	23.5.45/ 1111
218_8234	Tx 256- to 511-byte Packets Statistic Register (ENET_RMON_T_P256TO511)	32	R	0000_0000h	23.5.46/ 1112
218_8238	Tx 512- to 1023-byte Packets Statistic Register (ENET_RMON_T_P512TO1023)	32	R	0000_0000h	23.5.47/ 1112
218_823C	Tx 1024- to 2047-byte Packets Statistic Register (ENET_RMON_T_P1024TO2047)	32	R	0000_0000h	23.5.48/ 1113
218_8240	Tx Packets Greater Than 2048 Bytes Statistic Register (ENET_RMON_T_P_GTE2048)	32	R	0000_0000h	23.5.49/ 1113
218_8244	Tx Octets Statistic Register (ENET_RMON_T_OCTETS)	32	R	0000_0000h	23.5.50/ 1113
218_8248	Reserved Statistic Register (ENET_IEEE_T_DROP)	32	R	0000_0000h	23.5.51/ 1114
218_824C	Frames Transmitted OK Statistic Register (ENET_IEEE_T_FRAME_OK)	32	R	0000_0000h	23.5.52/ 1114
218_8250	Frames Transmitted with Single Collision Statistic Register (ENET_IEEE_T_1COL)	32	R	0000_0000h	23.5.53/ 1115
218_8254	Frames Transmitted with Multiple Collisions Statistic Register (ENET_IEEE_T_MCOL)	32	R	0000_0000h	23.5.54/ 1115
218_8258	Frames Transmitted after Deferral Delay Statistic Register (ENET_IEEE_T_DEF)	32	R	0000_0000h	23.5.55/ 1116
218_825C	Frames Transmitted with Late Collision Statistic Register (ENET_IEEE_T_LCOL)	32	R	0000_0000h	23.5.56/ 1116

Table continues on the next page...

ENET memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_8260	Frames Transmitted with Excessive Collisions Statistic Register (ENET_IEEE_T_EXCOL)	32	R	0000_0000h	23.5.57/ 1117
218_8264	Frames Transmitted with Tx FIFO Underrun Statistic Register (ENET_IEEE_T_MACERR)	32	R	0000_0000h	23.5.58/ 1117
218_8268	Frames Transmitted with Carrier Sense Error Statistic Register (ENET_IEEE_T_CSERR)	32	R	0000_0000h	23.5.59/ 1118
218_826C	Reserved Statistic Register (ENET_IEEE_T_SQE)	32	R (reads 0)	0000_0000h	23.5.60/ 1118
218_8270	Flow Control Pause Frames Transmitted Statistic Register (ENET_IEEE_T_FDXFC)	32	R	0000_0000h	23.5.61/ 1119
218_8274	Octet Count for Frames Transmitted w/o Error Statistic Register (ENET_IEEE_T_OCTETS_OK)	32	R	0000_0000h	23.5.62/ 1119
218_8284	Rx Packet Count Statistic Register (ENET_RMON_R_PACKETS)	32	R	0000_0000h	23.5.63/ 1120
218_8288	Rx Broadcast Packets Statistic Register (ENET_RMON_R_BC_PKT)	32	R	0000_0000h	23.5.64/ 1120
218_828C	Rx Multicast Packets Statistic Register (ENET_RMON_R_MC_PKT)	32	R	0000_0000h	23.5.65/ 1121
218_8290	Rx Packets with CRC/Align Error Statistic Register (ENET_RMON_R_CRC_ALIGN)	32	R	0000_0000h	23.5.66/ 1121
218_8294	Rx Packets with Less Than 64 Bytes and Good CRC Statistic Register (ENET_RMON_R_UNDERSIZE)	32	R	0000_0000h	23.5.67/ 1122
218_8298	Rx Packets Greater Than MAX_FL and Good CRC Statistic Register (ENET_RMON_R_OVERSIZE)	32	R	0000_0000h	23.5.68/ 1122
218_829C	Rx Packets Less Than 64 Bytes and Bad CRC Statistic Register (ENET_RMON_R_FRAG)	32	R	0000_0000h	23.5.69/ 1123
218_82A0	Rx Packets Greater Than MAX_FL Bytes and Bad CRC Statistic Register (ENET_RMON_R_JAB)	32	R	0000_0000h	23.5.70/ 1123
218_82A4	Reserved Statistic Register (ENET_RMON_R_RESVD_0)	32	R (reads 0)	0000_0000h	23.5.71/ 1123
218_82A8	Rx 64-Byte Packets Statistic Register (ENET_RMON_R_P64)	32	R	0000_0000h	23.5.72/ 1124
218_82AC	Rx 65- to 127-Byte Packets Statistic Register (ENET_RMON_R_P65TO127)	32	R	0000_0000h	23.5.73/ 1124
218_82B0	Rx 128- to 255-Byte Packets Statistic Register (ENET_RMON_R_P128TO255)	32	R	0000_0000h	23.5.74/ 1125
218_82B4	Rx 256- to 511-Byte Packets Statistic Register (ENET_RMON_R_P256TO511)	32	R	0000_0000h	23.5.75/ 1125
218_82B8	Rx 512- to 1023-Byte Packets Statistic Register (ENET_RMON_R_P512TO1023)	32	R	0000_0000h	23.5.76/ 1126
218_82BC	Rx 1024- to 2047-Byte Packets Statistic Register (ENET_RMON_R_P1024TO2047)	32	R	0000_0000h	23.5.77/ 1126
218_82C0	Rx Packets Greater than 2048 Bytes Statistic Register (ENET_RMON_R_P_GTE2048)	32	R	0000_0000h	23.5.78/ 1127

Table continues on the next page...

ENET memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_82C4	Rx Octets Statistic Register (ENET_RMON_R_OCTETS)	32	R	0000_0000h	23.5.79/ 1127
218_82C8	Frames not Counted Correctly Statistic Register (ENET_IEEE_R_DROP)	32	R	0000_0000h	23.5.80/ 1128
218_82CC	Frames Received OK Statistic Register (ENET_IEEE_R_FRAME_OK)	32	R	0000_0000h	23.5.81/ 1128
218_82D0	Frames Received with CRC Error Statistic Register (ENET_IEEE_R_CRC)	32	R	0000_0000h	23.5.82/ 1129
218_82D4	Frames Received with Alignment Error Statistic Register (ENET_IEEE_R_ALIGN)	32	R	0000_0000h	23.5.83/ 1129
218_82D8	Receive FIFO Overflow Count Statistic Register (ENET_IEEE_R_MACERR)	32	R	0000_0000h	23.5.84/ 1130
218_82DC	Flow Control Pause Frames Received Statistic Register (ENET_IEEE_R_FDXFC)	32	R	0000_0000h	23.5.85/ 1130
218_82E0	Octet Count for Frames Received without Error Statistic Register (ENET_IEEE_R_OCTETS_OK)	32	R	0000_0000h	23.5.86/ 1131
218_8400	Adjustable Timer Control Register (ENET_ATCR)	32	R/W	0000_0000h	23.5.87/ 1131
218_8404	Timer Value Register (ENET_ATVR)	32	R/W	0000_0000h	23.5.88/ 1133
218_8408	Timer Offset Register (ENET_ATOFF)	32	R/W	0000_0000h	23.5.89/ 1133
218_840C	Timer Period Register (ENET_ATPER)	32	R/W	3B9A_CA00h	23.5.90/ 1134
218_8410	Timer Correction Register (ENET_ATCOR)	32	R/W	0000_0000h	23.5.91/ 1134
218_8414	Time-Stamping Clock Period Register (ENET_ATINC)	32	R/W	0000_0000h	23.5.92/ 1135
218_8418	Timestamp of Last Transmitted Frame (ENET_ATSTMP)	32	R	0000_0000h	23.5.93/ 1135
218_8604	Timer Global Status Register (ENET_TGSR)	32	R/W	0000_0000h	23.5.94/ 1136
218_8608	Timer Control Status Register (ENET_TCSR0)	32	R/W	0000_0000h	23.5.95/ 1137
218_860C	Timer Compare Capture Register (ENET_TCCR0)	32	R/W	0000_0000h	23.5.96/ 1138
218_8610	Timer Control Status Register (ENET_TCSR1)	32	R/W	0000_0000h	23.5.95/ 1137
218_8614	Timer Compare Capture Register (ENET_TCCR1)	32	R/W	0000_0000h	23.5.96/ 1138
218_8618	Timer Control Status Register (ENET_TCSR2)	32	R/W	0000_0000h	23.5.95/ 1137
218_861C	Timer Compare Capture Register (ENET_TCCR2)	32	R/W	0000_0000h	23.5.96/ 1138

Table continues on the next page...

ENET memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_8620	Timer Control Status Register (ENET_TCSR3)	32	R/W	0000_0000h	23.5.95/ 1137
218_8624	Timer Compare Capture Register (ENET_TCCR3)	32	R/W	0000_0000h	23.5.96/ 1138

23.5.1 Interrupt Event Register (ENET_EIR)

When an event occurs that sets a bit in EIR, an interrupt occurs if the corresponding bit in the interrupt mask register (EIMR) is also set. Writing a 1 to an EIR bit clears it; writing 0 has no effect. This register is cleared upon hardware reset.

NOTE

TxBD[INT] and RxBD[INT] must be set to 1 to allow setting the corresponding EIR register flags in enhanced mode, ENET_ECR[EN1588] = 1. Legacy mode does not require these flags to be enabled.

Address: 218_8000h base + 4h offset = 218_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	BABR	BABT	GRA	TXF	TXB	RXF	RXB	MII	EBERR	LC	RL	UN	PLR	WAKEUP	TS_AVAIL
W		w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TS_TIMER															
W	w1c	0		0		0		0				0				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_EIR field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 BABR	Babbling Receive Error Indicates a frame was received with length in excess of RCR[MAX_FL] bytes.
29 BABT	Babbling Transmit Error Indicates the transmitted frame length exceeds RCR[MAX_FL] bytes. Usually this condition is caused when a frame that is too long is placed into the transmit data buffer(s). Truncation does not occur.
28 GRA	Graceful Stop Complete This interrupt is asserted after the transmitter is put into a pause state after completion of the frame currently being transmitted. See Graceful Transmit Stop (GTS) for conditions that lead to graceful stop. **NOTE:** The GRA interrupt is asserted only when the TX transitions into the stopped state. If this bit is cleared by writing 1 and the TX is still stopped, the bit is not set again.
27 TXF	Transmit Frame Interrupt Indicates a frame has been transmitted and the last corresponding buffer descriptor has been updated.
26 TXB	Transmit Buffer Interrupt Indicates a transmit buffer descriptor has been updated.
25 RXF	Receive Frame Interrupt Indicates a frame has been received and the last corresponding buffer descriptor has been updated.
24 RXB	Receive Buffer Interrupt Indicates a receive buffer descriptor is not the last in the frame has been updated.
23 MII	MII Interrupt. Indicates that the MII has completed the data transfer requested.
22 EBERR	Ethernet Bus Error Indicates a system bus error occurred when a uDMA transaction is underway. When this bit is set, ECR[ETHEREN] is cleared, halting frame processing by the MAC. When this occurs, software must ensure proper actions, possibly resetting the system, to resume normal operation.
21 LC	Late Collision Indicates a collision occurred beyond the collision window (slot time) in half-duplex mode. The frame truncates with a bad CRC and the remainder of the frame is discarded.
20 RL	Collision Retry Limit Indicates a collision occurred on each of 16 successive attempts to transmit the frame. The frame is discarded without being transmitted and transmission of the next frame commences. This error can only occur in half-duplex mode.
19 UN	Transmit FIFO Underrun Indicates the transmit FIFO became empty before the complete frame was transmitted. **NOTE:** In situations where the device has various masters generating high traffic, a FIFO underrun can occur on the transmit FIFO. To avoid transmit FIFO underrun, store and forward can be enabled

Table continues on the next page...

ENET_EIR field descriptions (continued)

Field	Description
	in ENET_TFWR[STRFWD]. See STRFWD. Also, a higher priority can be set for ENET traffic using available means on the central bus fabric connecting the ENET module.
18 PLR	Payload Receive Error Indicates a frame was received with a payload length error. See Frame Length/Type Verification: Payload Length Check for more information.
17 WAKEUP	Node Wakeup Request Indication Read-only status bit to indicate that a magic packet has been detected. Will act only if ECR[MAGICEN] is set.
16 TS_AVAIL	Transmit Timestamp Available Indicates that the timestamp of the last transmitted timing frame is available in the ATSTMP register.
15 TS_TIMER	Timestamp Timer The adjustable timer reached the period event. A period event interrupt can be generated if ATCR[PEREN] is set and the timer wraps according to the periodic setting in the ATPER register. Set the timer period value before setting ATCR[PEREN].
14–13 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
12 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
11–9 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
8 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.

23.5.2 Interrupt Mask Register (ENET_EIMR)

EIMR controls which interrupt events are allowed to generate actual interrupts. A hardware reset clears this register. If the corresponding bits in the EIR and EIMR registers are set, an interrupt is generated. The interrupt signal remains asserted until a 1 is written to the EIR field (write 1 to clear) or a 0 is written to the EIMR field.

Address: 218_8000h base + 8h offset = 218_8008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R		BABR	BABT	GRA	TXF	TXB	RXF	RXB	MII	EBERR	LC	RL	UN	PLR	WAKEUP	TS_AVAIL
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TS_TIMER															
W		0		0		0		0				0				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_EIMR field descriptions

Field	Description
31 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
30 BABR	BABR Interrupt Mask Corresponds to interrupt source EIR[BABR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR BABR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
29 BABT	BABT Interrupt Mask Corresponds to interrupt source EIR[BABT] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR BABT field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
28 GRA	GRA Interrupt Mask Corresponds to interrupt source EIR[GRA] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The

Table continues on the next page...

ENET_EIMR field descriptions (continued)

Field	Description
	corresponding EIR GRA field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
27 TXF	TXF Interrupt Mask Corresponds to interrupt source EIR[TXF] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
26 TXB	TXB Interrupt Mask Corresponds to interrupt source EIR[TXB] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
25 RXF	RXF Interrupt Mask Corresponds to interrupt source EIR[RXF] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
24 RXB	RXB Interrupt Mask Corresponds to interrupt source EIR[RXB] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RXB field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
23 MII	MII Interrupt Mask Corresponds to interrupt source EIR[MII] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR MII field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
22 EBERR	EBERR Interrupt Mask Corresponds to interrupt source EIR[EBERR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR EBERR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
21 LC	LC Interrupt Mask Corresponds to interrupt source EIR[LC] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR LC field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

Table continues on the next page...

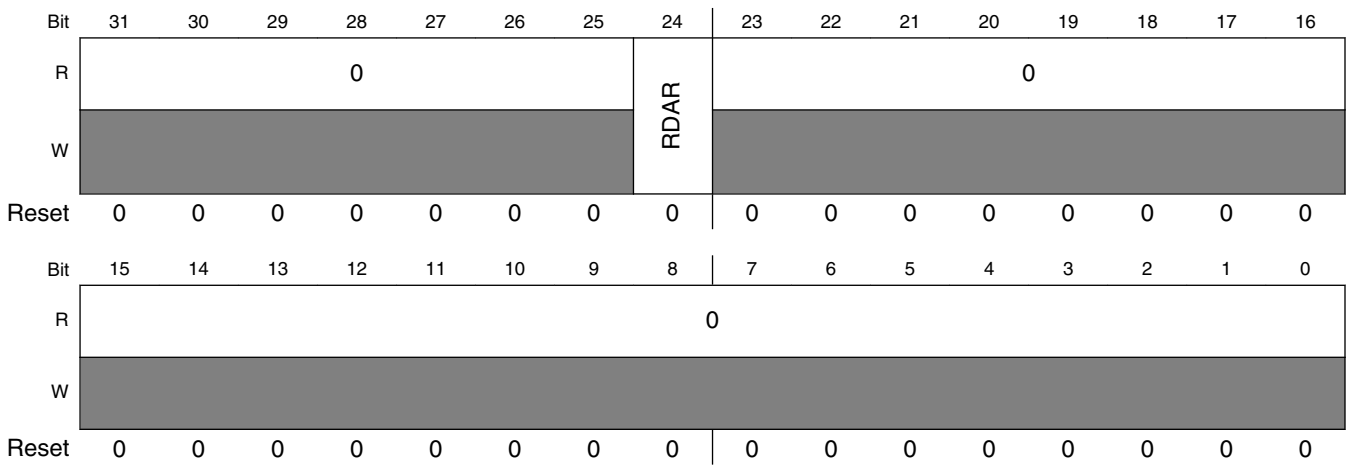
ENET_EIMR field descriptions (continued)

Field	Description
20 RL	RL Interrupt Mask Corresponds to interrupt source EIR[RL] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RL field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
19 UN	UN Interrupt Mask Corresponds to interrupt source EIR[UN] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR UN field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
18 PLR	PLR Interrupt Mask Corresponds to interrupt source EIR[PLR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR PLR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
17 WAKEUP	WAKEUP Interrupt Mask Corresponds to interrupt source EIR[WAKEUP] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR WAKEUP field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
16 TS_AVAIL	TS_AVAIL Interrupt Mask Corresponds to interrupt source EIR[TS_AVAIL] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TS_AVAIL field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
15 TS_TIMER	TS_TIMER Interrupt Mask Corresponds to interrupt source EIR[TS_TIMER] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TS_TIMER field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
14–13 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
12 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
11–9 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
8 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.

23.5.3 Receive Descriptor Active Register (ENET_RDAR)

RDAR is a command register, written by the user, to indicate that the receive descriptor ring has been updated, that is, that the driver produced empty receive buffers with the empty bit set.

Address: 218_8000h base + 10h offset = 218_8010h



ENET_RDAR field descriptions

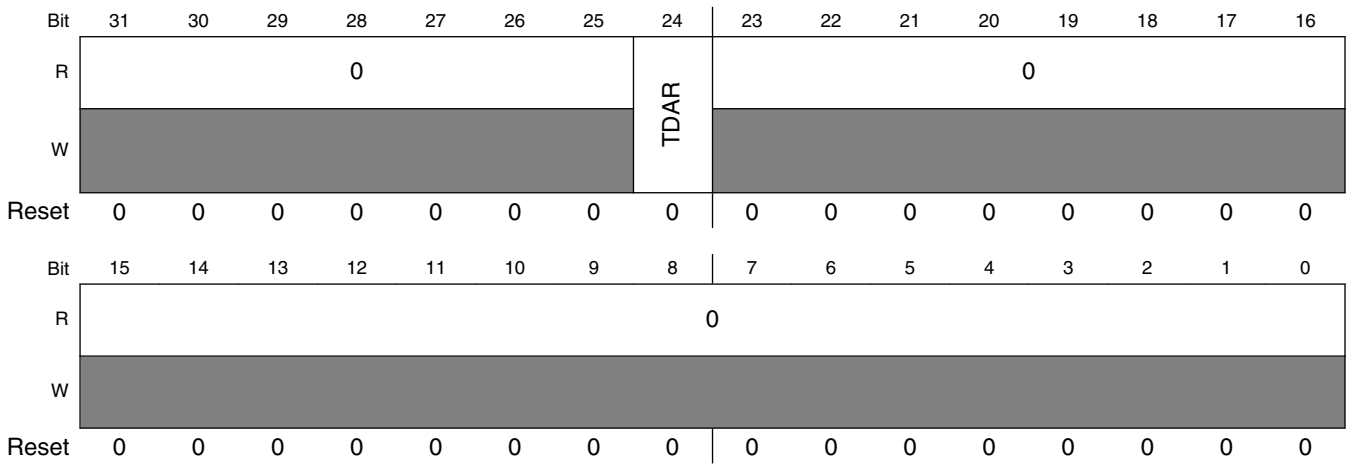
Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 RDAR	Receive Descriptor Active Always set to 1 when this register is written, regardless of the value written. This field is cleared by the MAC device when no additional empty descriptors remain in the receive ring. It is also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

23.5.4 Transmit Descriptor Active Register (ENET_TDAR)

The TDAR is a command register that the user writes to indicate that the transmit descriptor ring has been updated, that is, that transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor.

The TDAR register is cleared at reset, when ECR[ETHEREN] transitions from set to cleared, or when ECR[RESET] is set.

Address: 218_8000h base + 14h offset = 218_8014h



ENET_TDAR field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 TDAR	Transmit Descriptor Active Always set to 1 when this register is written, regardless of the value written. This bit is cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

23.5.5 Ethernet Control Register (ENET_ECR)

ECR is a read/write user register, though hardware may also alter fields in this register. It controls many of the high level features of the Ethernet MAC, including legacy FEC support through the EN1588 field.

Address: 218_8000h base + 24h offset = 218_8024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														Reserved	
Reset	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				Reserved	Reserved	Reserved	DBSWP	STOPEN	DBGEN	SPEED	EN1588	SLEEP	MAGICEN	ETHEREN	RESET
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_ECR field descriptions

Field	Description
31–18 Reserved	This field is reserved. Always write 11110000000000b to this field.
17–12 Reserved	This field is reserved. Always write 0 to this field.
11 Reserved	This field is reserved. Always write 0 to this field.
10 Reserved	This field is reserved. Always write 0 to this field.
9 Reserved	This field is reserved. Always write 0 to this field.
8 DBSWP	Descriptor Byte Swapping Enable Swaps the byte locations of the buffer descriptors. **NOTE:** This field must be written to 1 after reset. 0 The buffer descriptor bytes are not swapped to support big-endian devices. 1 The buffer descriptor bytes are swapped to support little-endian devices.
7 STOPEN	STOPEN Signal Control Controls device behavior in doze mode. In doze mode, if this field is set then all the clocks of the ENET assembly are disabled, except the RMII /MII clock. Doze mode is similar to a conditional stop mode entry for the ENET assembly depending on ECR[STOPEN].

Table continues on the next page...

ENET_ECR field descriptions (continued)

Field	Description
	NOTE: If module clocks are gated in this mode, the module can still wake the system after receiving a magic packet in stop mode. MAGICEN must be set prior to entering sleep/stop mode.
6 DBGEN	Debug Enable Enables the MAC to enter hardware freeze mode when the device enters debug mode. 0 MAC continues operation in debug mode. 1 MAC enters hardware freeze mode when the processor is in debug mode.
5 SPEED	Selects between 10/100-Mbit/s and 1000-Mbit/s modes of operation. 0 10/100-Mbit/s mode 1 1000-Mbit/s mode (Actual throughput is greater than 100 Mbit/s but significantly less than 1000 Mbit/s.)
4 EN1588	EN1588 Enable Enables enhanced functionality of the MAC. 0 Legacy FEC buffer descriptors and functions enabled. 1 Enhanced frame time-stamping functions enabled.
3 SLEEP	Sleep Mode Enable 0 Normal operating mode. 1 Sleep mode.
2 MAGICEN	Magic Packet Detection Enable Enables/disables magic packet detection. **NOTE:** MAGICEN is relevant only if the SLEEP field is set. If MAGICEN is set, changing the SLEEP field enables/disables sleep mode and magic packet detection. **NOTE:** EIMR[WAKEUP] must be written to one if Magic packet wakeup is programed to wake up the chip from low power mode. 0 Magic detection logic disabled. 1 The MAC core detects magic packets and asserts EIR[WAKEUP] when a frame is detected.
1 ETHEREN	Ethernet Enable Enables/disables the Ethernet MAC. When the MAC is disabled, the buffer descriptors for an aborted transmit frame are not updated. The uDMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers. Hardware clears this field under the following conditions: • RESET is set by software • An error condition causes the EBERR field to set. **NOTE:** • ETHEREN must be set at the very last step during ENET configuration/setup/initialization, only *after* all other ENET-related registers have been configured. • If ETHEREN is cleared to 0 by software then next time ETHEREN is set, the EIR interrupts must cleared to 0 due to previous pending interrupts. 0 Reception immediately stops and transmission stops after a bad CRC is appended to any currently transmitted frame. 1 MAC is enabled, and reception and transmission are possible.

Table continues on the next page...

ENET_ECR field descriptions (continued)

Field	Description
0 RESET	Ethernet MAC Reset When this field is set, it clears the ETHEREN field.

23.5.6 MII Management Frame Register (ENET_MMFR)

Writing to MMFR triggers a management frame transaction to the PHY device unless MSCR is programmed to zero.

If MSCR is changed from zero to non-zero during a write to MMFR, an MII frame is generated with the data previously written to the MMFR. This allows MMFR and MSCR to be programmed in either order if MSCR is currently zero.

If the MMFR register is written while frame generation is in progress, the frame contents are altered. Software must use the EIR[MII] interrupt indication to avoid writing to the MMFR register while frame generation is in progress.

Address: 218_8000h base + 40h offset = 218_8040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ST		OP		PA					RA					TA		DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_MMFR field descriptions

Field	Description
31–30 ST	Start Of Frame Delimiter See Table 23-41 (Clause 22) or Table 23-43 (Clause 45) for correct value.
29–28 OP	Operation Code See Table 23-41 (Clause 22) or Table 23-43 (Clause 45) for correct value.
27–23 PA	PHY Address See Table 23-41 (Clause 22) or Table 23-43 (Clause 45) for correct value.
22–18 RA	Register Address See Table 23-41 (Clause 22) or Table 23-43 (Clause 45) for correct value.
17–16 TA	Turn Around This field must be programmed to 10 to generate a valid MII management frame.
DATA	Management Frame Data This is the field for data to be written to or read from the PHY register.

23.5.7 MII Speed Control Register (ENET_MSCR)

MSCR provides control of the MII clock (MDC pin) frequency and allows a preamble drop on the MII management frame.

The MII_SPEED field must be programmed with a value to provide an MDC frequency of less than or equal to 2.5 MHz to be compliant with the IEEE 802.3 MII specification. The MII_SPEED must be set to a non-zero value to source a read or write management frame. After the management frame is complete, the MSCR register may optionally be cleared to turn off MDC. The MDC signal generated has a 50% duty cycle except when MII_SPEED changes during operation. This change takes effect following a rising or falling edge of MDC.

For example, if the internal module clock (that is, peripheral bus clock) is 25 MHz, programming MII_SPEED to 0x4 results in an MDC as given in the following equation:

MII clock frequency = 25 MHz / ((4 + 1) x 2) = 2.5 MHz

The following table shows the optimum values for MII_SPEED as a function of IPS bus clock frequency.

Table 23-4. Programming Examples for MSCR

Internal module clock frequency	MSCR [MII_SPEED]	MDC frequency
25 MHz	0x4	2.50 MHz
33 MHz	0x6	2.36 MHz
40 MHz	0x7	2.50 MHz
50 MHz	0x9	2.50 MHz
66 MHz	0xD	2.36 MHz

Address: 218_8000h base + 44h offset = 218_8044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					HOLDTIME			DIS_ PRE	MII_SPEED						0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

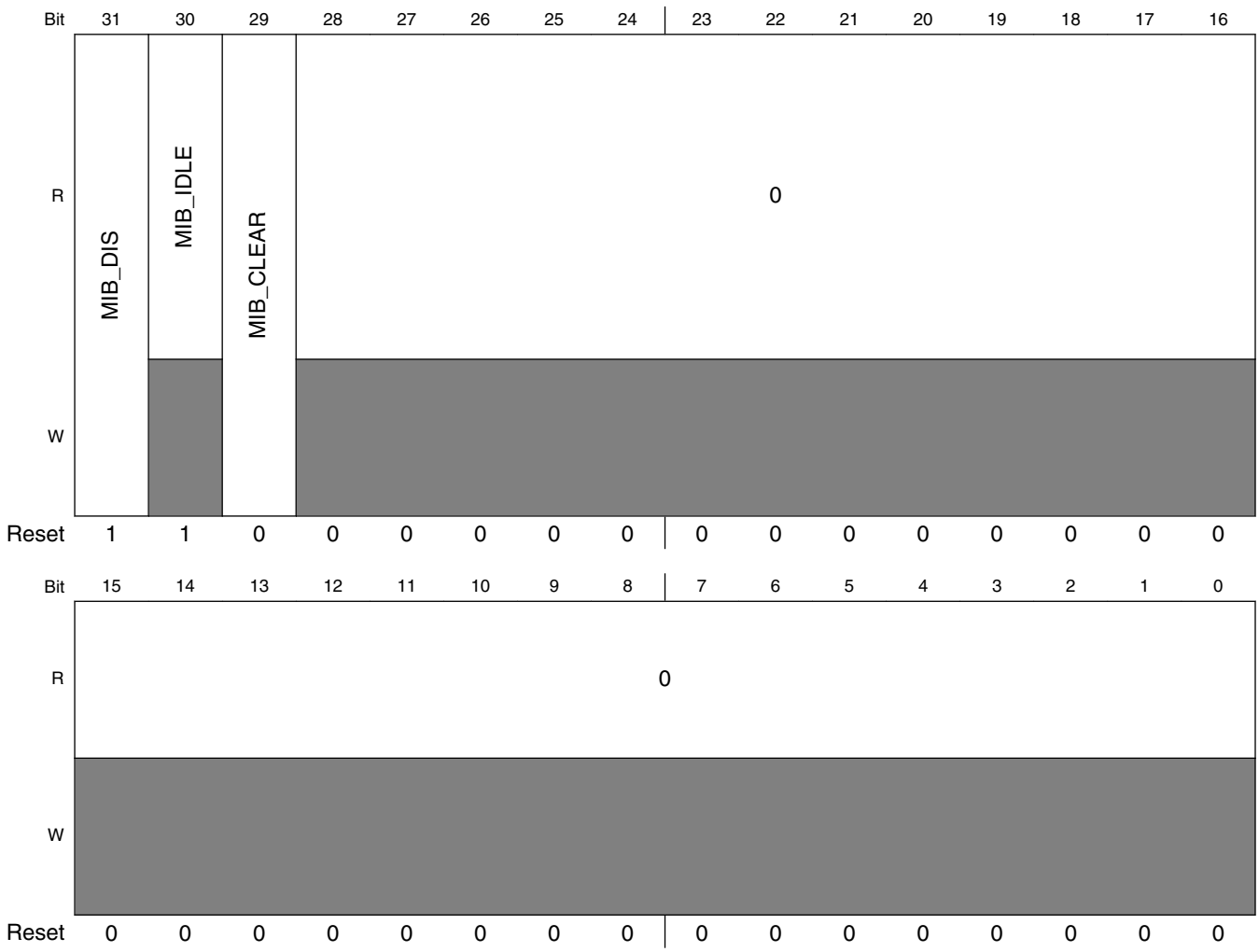
ENET_MSCR field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 HOLDTIME	Hold time On MDIO Output IEEE802.3 clause 22 defines a minimum of 10 ns for the hold time on the MDIO output. Depending on the host bus frequency, the setting may need to be increased. 000 1 internal module clock cycle 001 2 internal module clock cycles 010 3 internal module clock cycles 111 8 internal module clock cycles
7 DIS_PRE	Disable Preamble Enables/disables prepending a preamble to the MII management frame. The MII standard allows the preamble to be dropped if the attached PHY devices do not require it. 0 Preamble enabled. 1 Preamble (32 ones) is not prepended to the MII management frame.
6–1 MII_SPEED	MII Speed Controls the frequency of the MII management interface clock (MDC) relative to the internal module clock. A value of 0 in this field turns off MDC and leaves it in low voltage state. Any non-zero value results in the MDC frequency of: 1/((MII_SPEED + 1) x 2) of the internal module clock frequency
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

23.5.8 MIB Control Register (ENET_MIBC)

MIBC is a read/write register controlling and observing the state of the MIB block. Access this register to disable the MIB block operation or clear the MIB counters. The MIB_DIS field resets to 1.

Address: 218_8000h base + 64h offset = 218_8064h



ENET_MIBC field descriptions

Field	Description
31 MIB_DIS	Disable MIB Logic If this control field is set, 0 MIB logic is enabled. 1 MIB logic is disabled. The MIB logic halts and does not update any MIB counters.

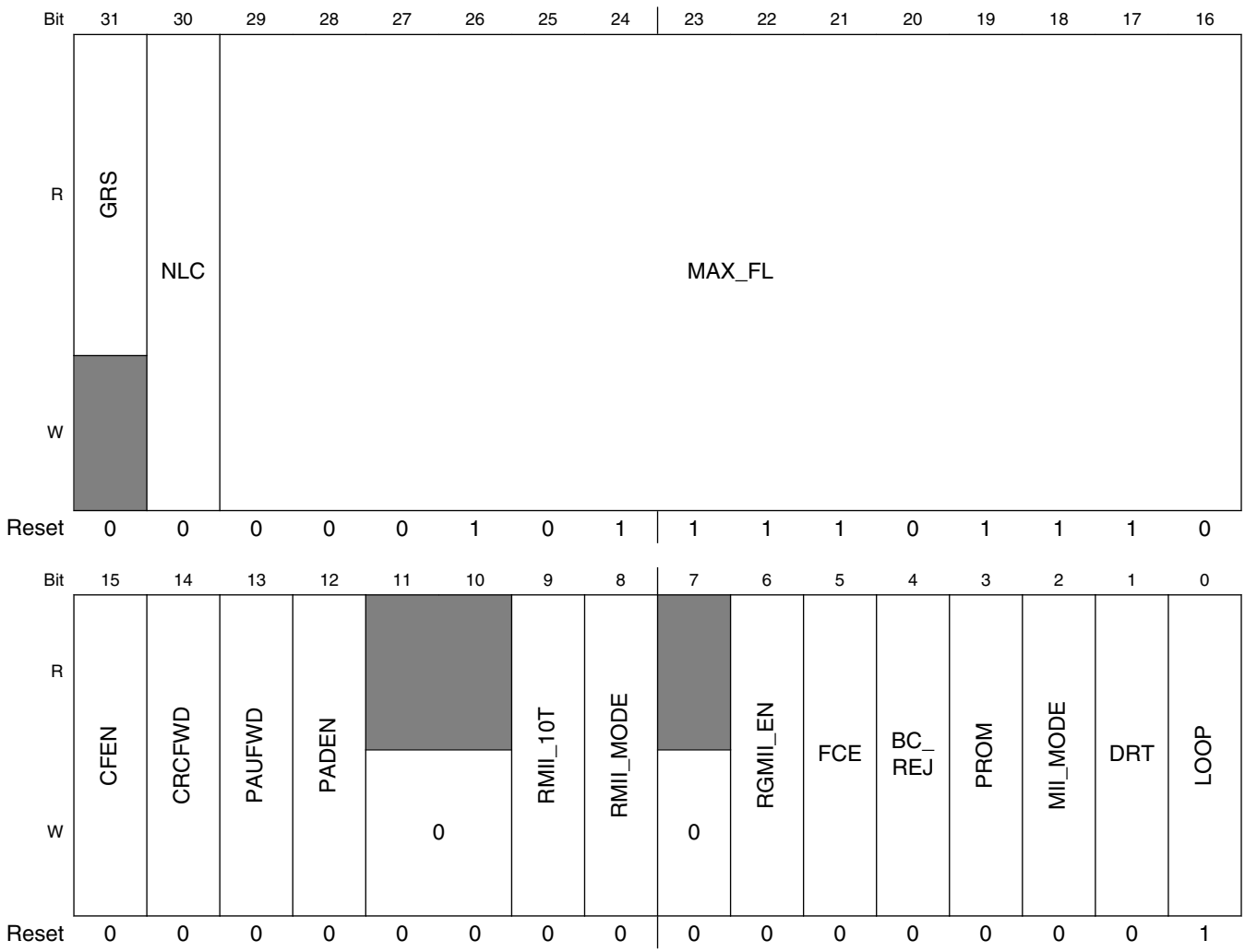
Table continues on the next page...

ENET_MIBC field descriptions (continued)

Field	Description
30 MIB_IDLE	MIB Idle 0 The MIB block is updating MIB counters. 1 The MIB block is not currently updating any MIB counters.
29 MIB_CLEAR	MIB Clear **NOTE:** This field is not self-clearing. To clear the MIB counters set and then clear this field. 0 See note above. 1 All statistics counters are reset to 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

23.5.9 Receive Control Register (ENET_RCR)

Address: 218_8000h base + 84h offset = 218_8084h



ENET_RCR field descriptions

Field	Description
31 GRS	Graceful Receive Stopped Read-only status indicating that the MAC receive datapath is stopped.
30 NLC	Payload Length Check Disable Enables/disables a payload length check. 0 The payload length check is disabled. 1 The core checks the frame's payload length with the frame length/type field. Errors are indicated in the EIR[PLR] field.
29–16 MAX_FL	Maximum Frame Length

Table continues on the next page...

ENET_RCR field descriptions (continued)

Field	Description
	Resets to decimal 1518. Length is measured starting at DA and includes the CRC at the end of the frame. Transmit frames longer than MAX_FL cause the BABT interrupt to occur. Receive frames longer than MAX_FL cause the BABR interrupt to occur and set the LG field in the end of frame receive buffer descriptor. The recommended default value to be programmed is 1518 or 1522 if VLAN tags are supported.
15 CFEN	MAC Control Frame Enable Enables/disables the MAC control frame. 0 MAC control frames with any opcode other than 0x0001 (pause frame) are accepted and forwarded to the client interface. 1 MAC control frames with any opcode other than 0x0001 (pause frame) are silently discarded.
14 CRCFWD	Terminate/Forward Received CRC Specifies whether the CRC field of received frames is transmitted or stripped. **NOTE:** If padding function is enabled (PADEN = 1), CRCFWD is ignored and the CRC field is checked and always terminated and removed. 0 The CRC field of received frames is transmitted to the user application. 1 The CRC field is stripped from the frame.
13 PAUFWD	Terminate/Forward Pause Frames Specifies whether pause frames are terminated or forwarded. 0 Pause frames are terminated and discarded in the MAC. 1 Pause frames are forwarded to the user application.
12 PADEN	Enable Frame Padding Remove On Receive Specifies whether the MAC removes padding from received frames. 0 No padding is removed on receive by the MAC. 1 Padding is removed from received frames.
11–10 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
9 RMII_10T	Enables 10-Mbit/s mode of the RMII or RGMII . 0 100-Mbit/s operation. 1 10-Mbit/s operation.
8 RMII_MODE	RMII Mode Enable Specifies whether the MAC is configured for MII mode or RMII operation , when ECR[SPEED] is cleared . **NOTE:** Do not set both RCR[RGMII_EN] and RCR[RMII_MODE]. 0 MAC configured for MII mode. 1 MAC configured for RMII operation.
7 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
6 RGMII_EN	RGMII Mode Enable **NOTE:** Do not set both RCR[RGMII_EN] and RCR[RMII_MODE].

Table continues on the next page...

ENET_RCR field descriptions (continued)

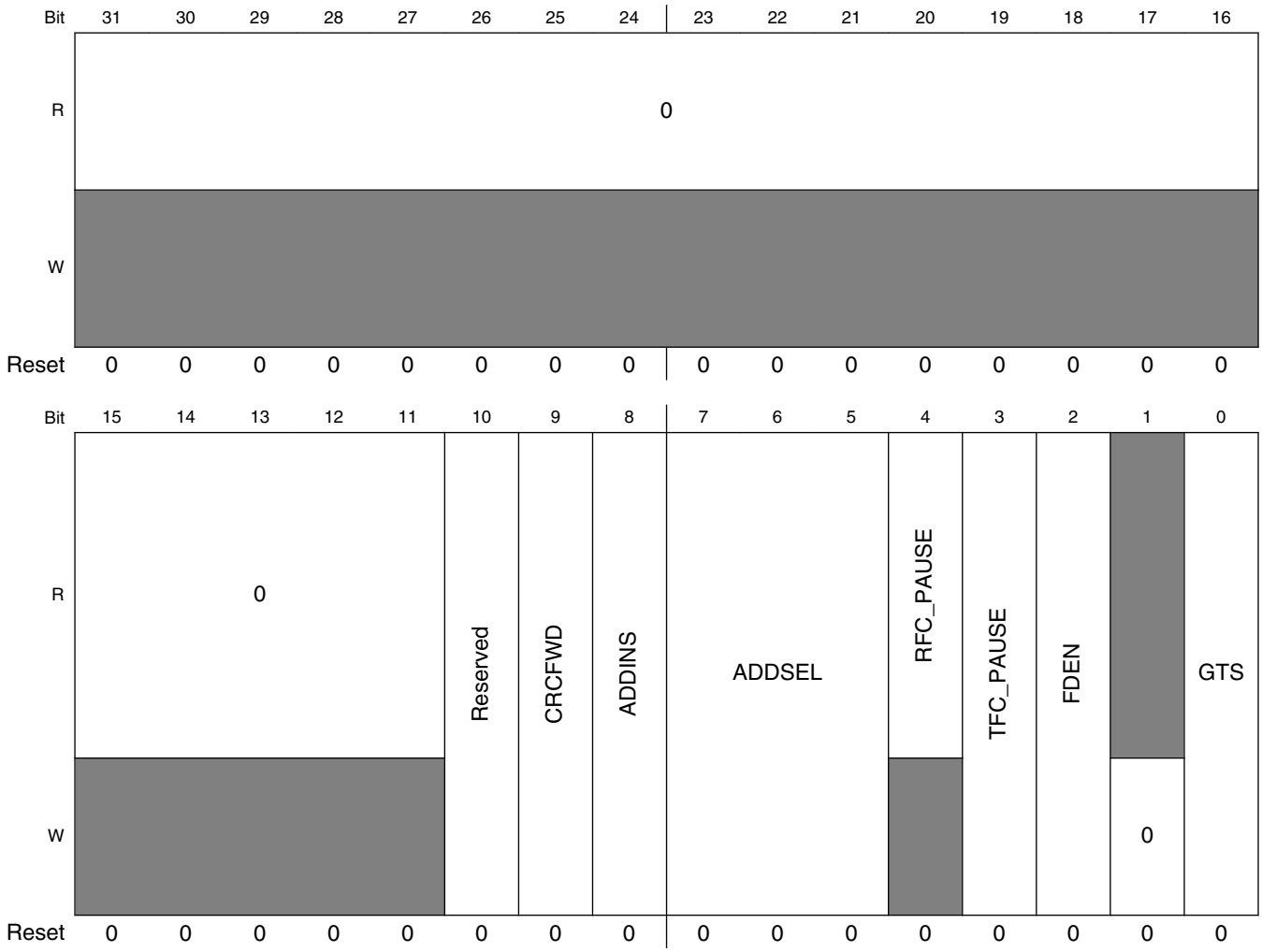
Field	Description
	0 MAC configured for non-RGMII operation 1 MAC configured for RGMII operation. If ECR[SPEED] is set, the MAC is in RGMII 1000-Mbit/s[1] mode. If ECR[SPEED] is cleared, the MAC is in RGMII 10/100-Mbit/s mode.
5 FCE	Flow Control Enable If set, the receiver detects PAUSE frames. Upon PAUSE frame detection, the transmitter stops transmitting data frames for a given duration.
4 BC_REJ	Broadcast Frame Reject If set, frames with destination address (DA) equal to 0xFFFF_FFFF_FFFF are rejected unless the PROM field is set. If BC_REJ and PROM are set, frames with broadcast DA are accepted and the MISS (M) is set in the receive buffer descriptor.
3 PROM	Promiscuous Mode All frames are accepted regardless of address matching. 0 Disabled. 1 Enabled.
2 MII_MODE	Media Independent Interface Mode This field must always be set. 0 Reserved. 1 MII or RMII mode, as indicated by the RMII_MODE field.
1 DRT	Disable Receive On Transmit 0 Receive path operates independently of transmit (i.e., full-duplex mode). Can also be used to monitor transmit activity in half-duplex mode. 1 Disable reception of frames while transmitting. (Normally used for half-duplex mode.)
0 LOOP	Internal Loopback This is an MII internal loopback, therefore MII_MODE must be written to 1 and RMII_MODE must be written to 0. 0 Loopback disabled. 1 Transmitted frames are looped back internal to the device and transmit MII output signals are not asserted. DRT must be cleared.

1. Actual throughput is greater than 100 Mbit/s but significantly less than 1000 Mbit/s.

23.5.10 Transmit Control Register (ENET_TCR)

TCR is read/write and configures the transmit block. This register is cleared at system reset. FDEN can only be modified when ECR[ETHEREN] is cleared.

Address: 218_8000h base + C4h offset = 218_80C4h



ENET_TCR field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 Reserved	This field is reserved. This field is read/write and must be set to 0.
9 CRCFWD	Forward Frame From Application With CRC

Table continues on the next page...

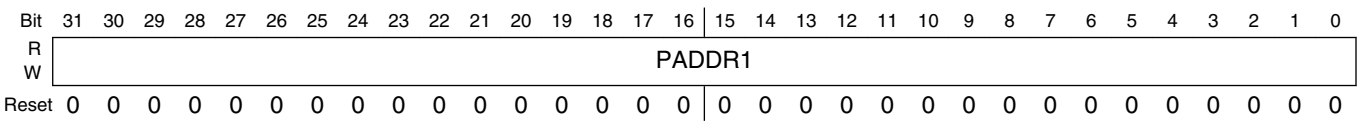
ENET_TCR field descriptions (continued)

Field	Description
	0 TxBD[TC] controls whether the frame has a CRC from the application. 1 The transmitter does not append any CRC to transmitted frames, as it is expecting a frame with CRC from the application.
8 ADDINS	Set MAC Address On Transmit 0 The source MAC address is not modified by the MAC. 1 The MAC overwrites the source MAC address with the programmed MAC address according to ADDSEL.
7–5 ADDSEL	Source MAC Address Select On Transmit If ADDINS is set, indicates the MAC address that overwrites the source MAC address. 000 Node MAC address programmed on PADDR1/2 registers. 100 Reserved. 101 Reserved. 110 Reserved.
4 RFC_PAUSE	Receive Frame Control Pause This status field is set when a full-duplex flow control pause frame is received and the transmitter pauses for the duration defined in this pause frame. This field automatically clears when the pause duration is complete.
3 TFC_PAUSE	Transmit Frame Control Pause Pauses frame transmission. When this field is set, EIR[GRA] is set. With transmission of data frames stopped, the MAC transmits a MAC control PAUSE frame. Next, the MAC clears TFC_PAUSE and resumes transmitting data frames. If the transmitter pauses due to user assertion of GTS or reception of a PAUSE frame, the MAC may continue transmitting a MAC control PAUSE frame. 0 No PAUSE frame transmitted. 1 The MAC stops transmission of data frames after the current transmission is complete.
2 FDEN	Full-Duplex Enable If this field is set, frames transmit independent of carrier sense and collision inputs. Only modify this bit when ECR[ETHEREN] is cleared.
1 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
0 GTS	Graceful Transmit Stop When this field is set, MAC stops transmission after any frame currently transmitted is complete and EIR[GRA] is set. If frame transmission is not currently underway, the GRA interrupt is asserted immediately. After transmission finishes, clear GTS to restart. The next frame in the transmit FIFO is then transmitted. If an early collision occurs during transmission when GTS is set, transmission stops after the collision. The frame is transmitted again after GTS is cleared. There may be old frames in the transmit FIFO that transmit when GTS is reasserted. To avoid this, clear ECR[ETHEREN] following the GRA interrupt.

23.5.11 Physical Address Lower Register (ENET_PALR)

PALR contains the lower 32 bits (bytes 0, 1, 2, 3) of the 48-bit address used in the address recognition process to compare with the destination address (DA) field of receive frames with an individual DA. In addition, this register is used in bytes 0 through 3 of the six-byte source address field when transmitting PAUSE frames.

Address: 218_8000h base + E4h offset = 218_80E4h



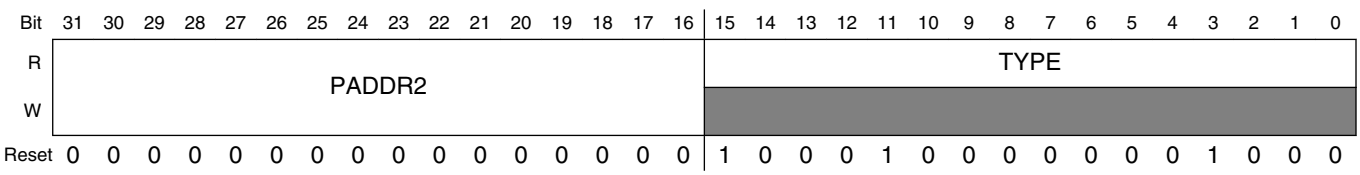
ENET_PALR field descriptions

Field	Description
PADDR1	Pause Address Bytes 0 (bits 31:24), 1 (bits 23:16), 2 (bits 15:8), and 3 (bits 7:0) of the 6-byte individual address are used for exact match and the source address field in PAUSE frames.

23.5.12 Physical Address Upper Register (ENET_PAUR)

PAUR contains the upper 16 bits (bytes 4 and 5) of the 48-bit address used in the address recognition process to compare with the destination address (DA) field of receive frames with an individual DA. In addition, this register is used in bytes 4 and 5 of the six-byte source address field when transmitting PAUSE frames. Bits 15:0 of PAUR contain a constant type field (0x8808) for transmission of PAUSE frames.

Address: 218_8000h base + E8h offset = 218_80E8h



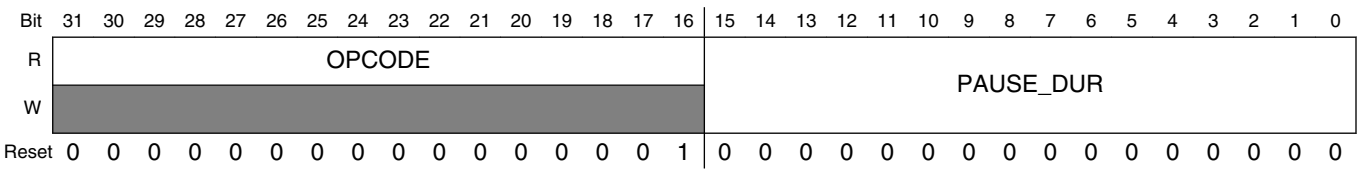
ENET_PAUR field descriptions

Field	Description
31–16 PADDR2	Bytes 4 (bits 31:24) and 5 (bits 23:16) of the 6-byte individual address used for exact match, and the source address field in PAUSE frames.
TYPE	Type Field In PAUSE Frames These fields have a constant value of 0x8808.

23.5.13 Opcode/Pause Duration Register (ENET_OPD)

OPD is read/write accessible. This register contains the 16-bit opcode and 16-bit pause duration fields used in transmission of a PAUSE frame. The opcode field is a constant value, 0x0001. When another node detects a PAUSE frame, that node pauses transmission for the duration specified in the pause duration field. The lower 16 bits of this register are not reset and you must initialize it.

Address: 218_8000h base + ECh offset = 218_80ECh



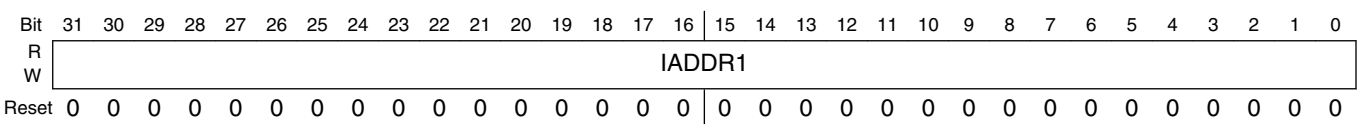
ENET_OPD field descriptions

Field	Description
31–16 OPCODE	Opcode Field In PAUSE Frames These fields have a constant value of 0x0001.
PAUSE_DUR	Pause Duration Pause duration field used in PAUSE frames.

23.5.14 Descriptor Individual Upper Address Register (ENET_IAUR)

IAUR contains the upper 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the destination address (DA) field of receive frames with an individual DA. This register is not reset and you must initialize it.

Address: 218_8000h base + 118h offset = 218_8118h



ENET_IAUR field descriptions

Field	Description
IADDR1	Contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR1 contains hash index bit 63. Bit 0 of IADDR1 contains hash index bit 32.

23.5.15 Descriptor Individual Lower Address Register (ENET_IALR)

IALR contains the lower 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the DA field of receive frames with an individual DA. This register is not reset and you must initialize it.

Address: 218_8000h base + 11Ch offset = 218_811Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IADDR2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IALR field descriptions

Field	Description
IADDR2	Contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR2 contains hash index bit 31. Bit 0 of IADDR2 contains hash index bit 0.

23.5.16 Descriptor Group Upper Address Register (ENET_GAUR)

GAUR contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. You must initialize this register.

Address: 218_8000h base + 120h offset = 218_8120h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GADDR1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_GAUR field descriptions

Field	Description
GADDR1	Contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR1 contains hash index bit 63. Bit 0 of GADDR1 contains hash index bit 32.

23.5.17 Descriptor Group Lower Address Register (ENET_GALR)

GALR contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. You must initialize this register.

Address: 218_8000h base + 124h offset = 218_8124h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GADDR2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_GALR field descriptions

Field	Description
GADDR2	Contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR2 contains hash index bit 31. Bit 0 of GADDR2 contains hash index bit 0.

23.5.18 Transmit FIFO Watermark Register (ENET_TFWR)

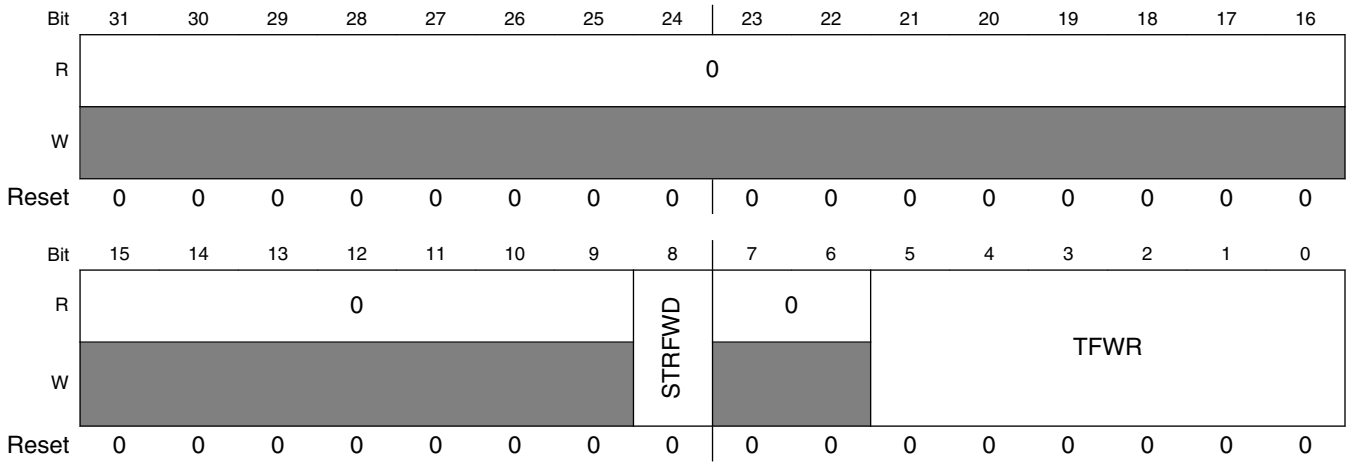
If TFWR[STRFWD] is cleared, TFWR[TFWR] controls the amount of data required in the transmit FIFO before transmission of a frame can begin. This allows you to minimize transmit latency (TFWR = 00 or 01) or allow for larger bus access latency (TFWR = 11) due to contention for the system bus. Setting the watermark to a high value minimizes the risk of transmit FIFO underrun due to contention for the system bus. The byte counts associated with the TFWR field may need to be modified to match a given system requirement, for example, worst-case bus access latency by the transmit data uDMA channel.

When the FIFO level reaches the value the TFWR field and when the STR_FWD is set to ‘0’, the MAC transmit control logic starts frame transmission even before the end-of-frame is available in the FIFO (cut-through operation).

If a complete frame has a size smaller than the threshold programmed with TFWR, the MAC also transmits the Frame to the line.

To enable store and forward on the Transmit path, set STR_FWD to ‘1’. In this case, the MAC starts to transmit data only when a complete frame is stored in the Transmit FIFO.

Address: 218_8000h base + 144h offset = 218_8144h



ENET_TFWR field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 STRFWD	Store And Forward Enable 0 Reset. The transmission start threshold is programmed in TFWR[TFWR]. 1 Enabled.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TFWR	Transmit FIFO Write If TFWR[STRFWD] is cleared, this field indicates the number of bytes, in steps of 64 bytes, written to the transmit FIFO before transmission of a frame begins. **NOTE:** If a frame with less than the threshold is written, it is still sent independently of this threshold setting. The threshold is relevant only if the frame is larger than the threshold given. 000000 64 bytes written. 000001 64 bytes written. 000010 128 bytes written. 000011 192 bytes written. ... ... 111111 4032 bytes written.

23.5.19 Receive Descriptor Ring Start Register (ENET_RDSR)

RDSR points to the beginning of the circular receive buffer descriptor queue in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, it is recommended to be 128-bit aligned, that is, evenly divisible by 16.

NOTE

This register must be initialized prior to operation

Address: 218_8000h base + 180h offset = 218_8180h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	R_DES_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	R_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RDSR field descriptions

Field	Description
31–3 R_DES_START	Pointer to the beginning of the receive buffer descriptor queue.
2 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

23.5.20 Transmit Buffer Descriptor Ring Start Register (ENET_TDSR)

TDSR provides a pointer to the beginning of the circular transmit buffer descriptor queue in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, it is recommended to be 128-bit aligned, that is, evenly divisible by 16.

NOTE

This register must be initialized prior to operation.

Address: 218_8000h base + 184h offset = 218_8184h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	X_DES_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	X_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_TDSR field descriptions

Field	Description
31–3 X_DES_START	Pointer to the beginning of the transmit buffer descriptor queue.

Table continues on the next page...

ENET_TDSR field descriptions (continued)

Field	Description
2 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

23.5.21 Maximum Receive Buffer Size Register (ENET_MRBR)

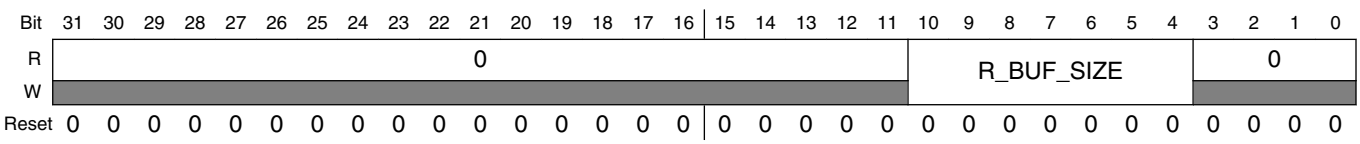
The MRBR is a user-programmable register that dictates the maximum size of all receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer.

- R_BUF_SIZE is concatentated with the four least-significant bits of this register and are used as the maximum receive buffer size.
- To allow one maximum size frame per buffer, MRBR must be set to RCR[MAX_FL] or larger.
- To properly align the buffer, MRBR must be evenly divisible by 16. To ensure this, the lower four bits are set to zero by the device.
- To minimize bus usage (descriptor fetches), set MRBR greater than or equal to 256 bytes.

NOTE

This register must be initialized before operation.

Address: 218_8000h base + 188h offset = 218_8188h

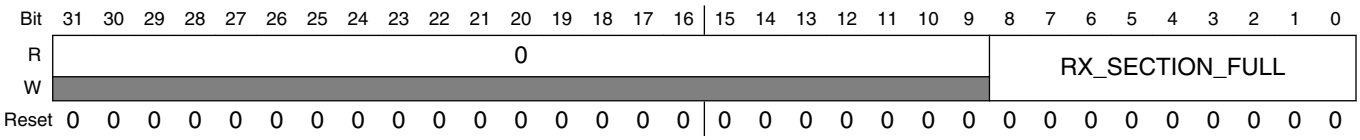


ENET_MRBR field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–4 R_BUF_SIZE	Receive buffer size in bytes. This value, concatenated with the four least-significant bits of this register (which are always zero), is the effective maximum receive buffer size.
Reserved	This field, which is always zero, is the four least-significant bits of the maximum receive buffer size. This field is reserved. This read-only field is reserved and always has the value 0.

23.5.22 Receive FIFO Section Full Threshold (ENET_RSFL)

Address: 218_8000h base + 190h offset = 218_8190h

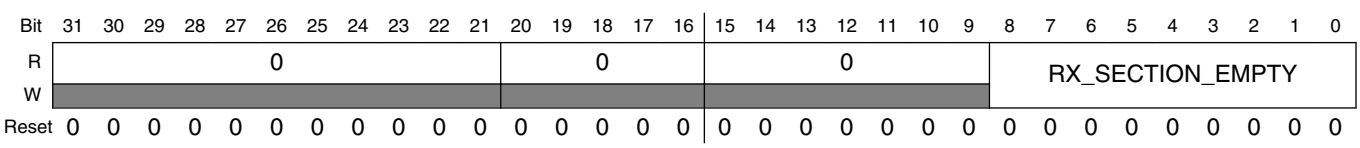


ENET_RSFL field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RX_SECTION_FULL	Value Of Receive FIFO Section Full Threshold Value, in 64-bit words, of the receive FIFO section full threshold. Clear this field to enable store and forward on the RX FIFO. When programming a value greater than 0 (cut-through operation), it must be greater than RAEM[RX_ALMOST_EMPTY]. When the FIFO level reaches the value in this field, data is available in the Receive FIFO (cut-through operation).

23.5.23 Receive FIFO Section Empty Threshold (ENET_RSEM)

Address: 218_8000h base + 194h offset = 218_8194h



ENET_RSEM field descriptions

Field	Description
31–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RX_SECTION_EMPTY	Value Of The Receive FIFO Section Empty Threshold Value, in 64-bit words, of the receive FIFO section empty threshold. When the FIFO has reached this level, a pause frame will be issued. A value of 0 disables automatic pause frame generation. When the FIFO level goes below the value programmed in this field, an XON pause frame is issued to indicate the FIFO congestion is cleared to the remote Ethernet client.

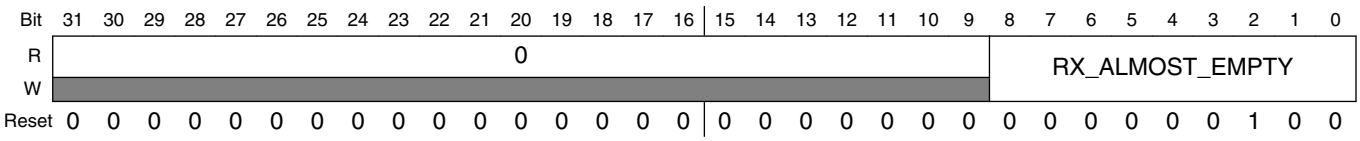
Table continues on the next page...

ENET_RSEM field descriptions (continued)

Field	Description
	NOTE: The section-empty threshold indications from both FIFOs are OR'ed to cause XOFF pause frame generation.

23.5.24 Receive FIFO Almost Empty Threshold (ENET_RAEM)

Address: 218_8000h base + 198h offset = 218_8198h

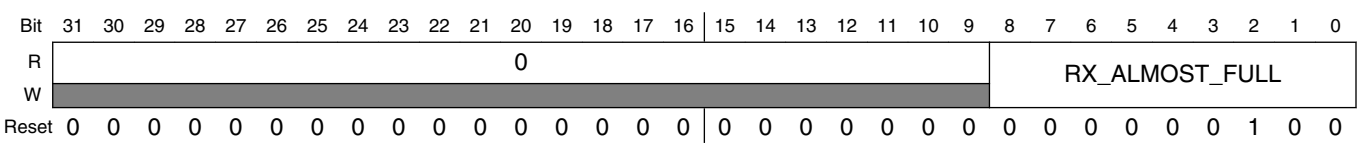


ENET_RAEM field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RX_ALMOST_EMPTY	Value Of The Receive FIFO Almost Empty Threshold Value, in 64-bit words, of the receive FIFO almost empty threshold. When the FIFO level reaches the value programmed in this field and the end-of-frame has not been received for the frame yet, the core receive read control stops FIFO read (and subsequently stops transferring data to the MAC client application). It continues to deliver the frame, if again more data than the threshold or the end-of-frame is available in the FIFO. A minimum value of 4 should be set.

23.5.25 Receive FIFO Almost Full Threshold (ENET_RAFL)

Address: 218_8000h base + 19Ch offset = 218_819Ch



ENET_RAFL field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RX_ALMOST_FULL	Value Of The Receive FIFO Almost Full Threshold Value, in 64-bit words, of the receive FIFO almost full threshold. When the FIFO level comes close to the maximum, so that there is no more space for at least RX_ALMOST_FULL number of words, the MAC stops writing data in the FIFO and truncates the received frame to avoid FIFO overflow. The corresponding error status will be set when the frame is delivered to the application. A minimum value of 4 should be set.

23.5.26 Transmit FIFO Section Empty Threshold (ENET_TSEM)

Address: 218_8000h base + 1A0h offset = 218_81A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TX_SECTION_EMPTY																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_TSEM field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TX_SECTION_ EMPTY	Value Of The Transmit FIFO Section Empty Threshold Value, in 64-bit words, of the transmit FIFO section empty threshold. See Transmit FIFO for more information.

23.5.27 Transmit FIFO Almost Empty Threshold (ENET_TAEM)

Address: 218_8000h base + 1A4h offset = 218_81A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TX_ALMOST_EMPTY																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

ENET_TAEM field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TX_ALMOST_ EMPTY	Value of Transmit FIFO Almost Empty Threshold Value, in 64-bit words, of the transmit FIFO almost empty threshold. When the FIFO level reaches the value programmed in this field, and no end-of-frame is available for the frame, the MAC transmit logic, to avoid FIFO underflow, stops reading the FIFO and transmits a frame with an MII error indication. See Transmit FIFO for more information. A minimum value of 4 should be set.

23.5.28 Transmit FIFO Almost Full Threshold (ENET_TAFL)

Address: 218_8000h base + 1A8h offset = 218_81A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TX_ALMOST_FULL															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

ENET_TAFL field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TX_ALMOST_FULL	Value Of The Transmit FIFO Almost Full Threshold Value, in 64-bit words, of the transmit FIFO almost full threshold. A minimum value of six is required . A recommended value of at least 8 should be set allowing a latency of two clock cycles to the application. If more latency is required the value can be increased as necessary (latency = TAFL - 5). When the FIFO level comes close to the maximum, so that there is no more space for at least TX_ALMOST_FULL number of words, the pin ff_tx_rdy is deasserted. If the application does not react on this signal, the FIFO write control logic, to avoid FIFO overflow, truncates the current frame and sets the error status. As a result, the frame will be transmitted with an GMII/MII error indication. See Transmit FIFO for more information. **NOTE:** A FIFO overflow is a fatal error and requires a global reset on the transmit datapath or at least deassertion of ETHEREN.

23.5.29 Transmit Inter-Packet Gap (ENET_TIPG)

Address: 218_8000h base + 1ACh offset = 218_81ACh

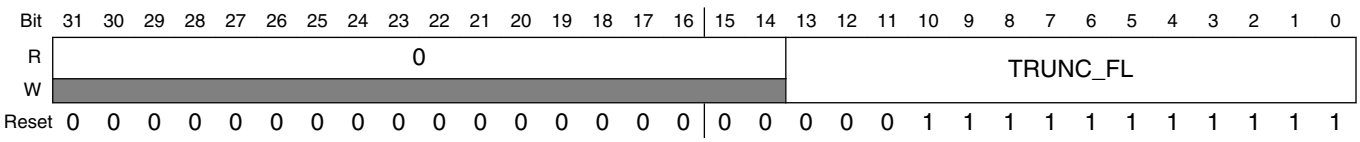
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											IPG				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

ENET_TIPG field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
IPG	Transmit Inter-Packet Gap Indicates the IPG, in bytes, between transmitted frames. Valid values range from 8 to 26. If the written value is less than 8 or greater than 26, the internal (effective) IPG is 12. **NOTE:** The IPG value read will be the value that was written, even if it is out of range.

23.5.30 Frame Truncation Length (ENET_FTRL)

Address: 218_8000h base + 1B0h offset = 218_81B0h



ENET_FTRL field descriptions

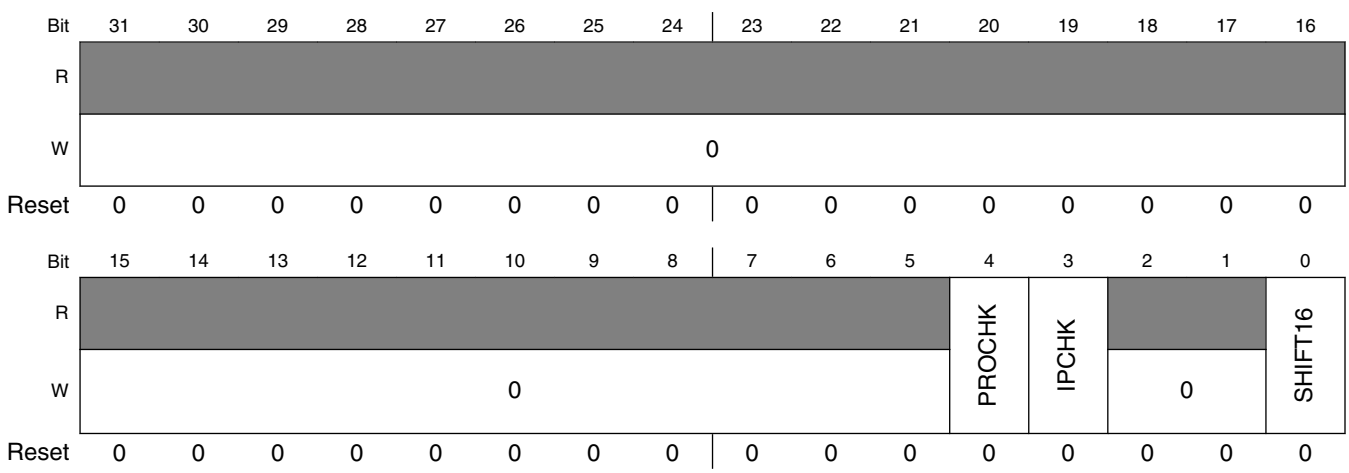
Field	Description
31–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TRUNC_FL	Frame Truncation Length Indicates the value a receive frame is truncated, if it is greater than this value. Must be greater than or equal to RCR[MAX_FL]. **NOTE:** Truncation happens at TRUNC_FL. However, when truncation occurs, the application (FIFO) may receive less data, guaranteeing that it never receives more than the set limit.

23.5.31 Transmit Accelerator Function Configuration (ENET_TACC)

TACC controls accelerator actions when sending frames. The register can be changed before or after each frame, but it must remain unmodified during frame writes into the transmit FIFO.

The TFWR[STRFWD] field must be set to use the checksum feature.

Address: 218_8000h base + 1C0h offset = 218_81C0h

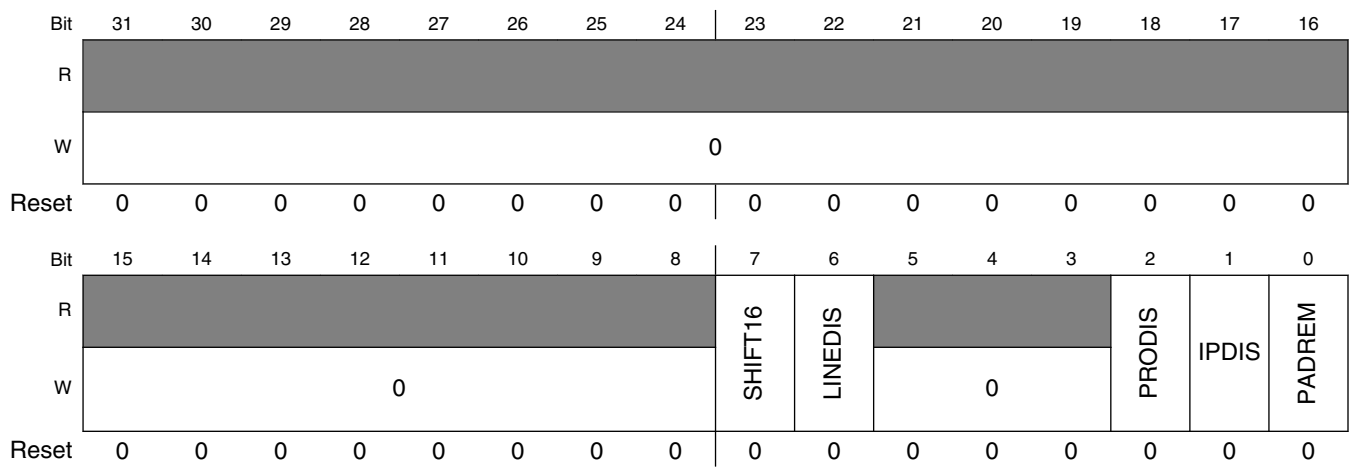


ENET_TACC field descriptions

Field	Description
31–5 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
4 PROCHK	Enables insertion of protocol checksum. 0 Checksum not inserted. 1 If an IP frame with a known protocol is transmitted, the checksum is inserted automatically into the frame. The checksum field must be cleared. The other frames are not modified.
3 IPCHK	Enables insertion of IP header checksum. 0 Checksum is not inserted. 1 If an IP frame is transmitted, the checksum is inserted automatically. The IP header checksum field must be cleared. If a non-IP frame is transmitted the frame is not modified.
2–1 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
0 SHIFT16	TX FIFO Shift-16 0 Disabled. 1 Indicates to the transmit data FIFO that the written frames contain two additional octets before the frame data. This means the actual frame begins at bit 16 of the first word written into the FIFO. This function allows putting the frame payload on a 32-bit boundary in memory, as the 14-byte Ethernet header is extended to a 16-byte header.

23.5.32 Receive Accelerator Function Configuration (ENET_RACC)

Address: 218_8000h base + 1C4h offset = 218_81C4h



ENET_RACC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.

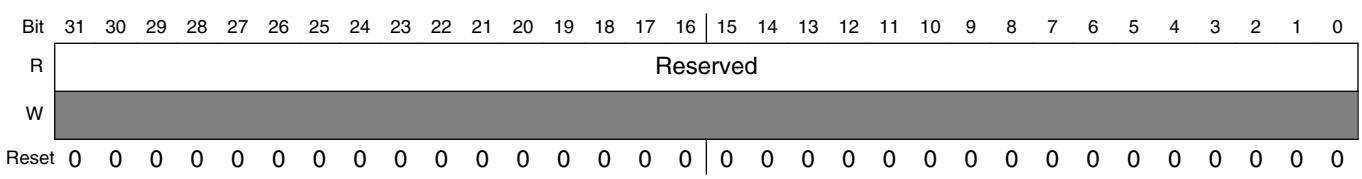
Table continues on the next page...

ENET_RACC field descriptions (continued)

Field	Description
7 SHIFT16	RX FIFO Shift-16 When this field is set, the actual frame data starts at bit 16 of the first word read from the RX FIFO aligning the Ethernet payload on a 32-bit boundary. **NOTE:** This function only affects the FIFO storage and has no influence on the statistics, which use the actual length of the frame received. 0 Disabled. 1 Instructs the MAC to write two additional bytes in front of each frame received into the RX FIFO.
6 LINEDIS	Enable Discard Of Frames With MAC Layer Errors 0 Frames with errors are not discarded. 1 Any frame received with a CRC, length, or PHY error is automatically discarded and not forwarded to the user application interface.
5–3 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
2 PRODIS	Enable Discard Of Frames With Wrong Protocol Checksum 0 Frames with wrong checksum are not discarded. 1 If a TCP/IP, UDP/IP, or ICMP/IP frame is received that has a wrong TCP, UDP, or ICMP checksum, the frame is discarded. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).
1 IPDIS	Enable Discard Of Frames With Wrong IPv4 Header Checksum 0 Frames with wrong IPv4 header checksum are not discarded. 1 If an IPv4 frame is received with a mismatching header checksum, the frame is discarded. IPv6 has no header checksum and is not affected by this setting. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).
0 PADREM	Enable Padding Removal For Short IP Frames 0 Padding not removed. 1 Any bytes following the IP payload section of the frame are removed from the frame.

23.5.33 Reserved Statistic Register (ENET_RMON_T_DROP)

Address: 218_8000h base + 200h offset = 218_8200h

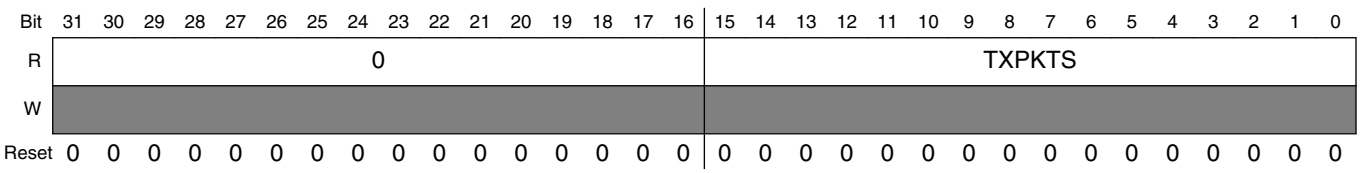


ENET_RMON_T_DROP field descriptions

Field	Description
Reserved	This read-only field always has the value 0. This field is reserved.

23.5.34 Tx Packet Count Statistic Register (ENET_RMON_T_PACKETS)

Address: 218_8000h base + 204h offset = 218_8204h



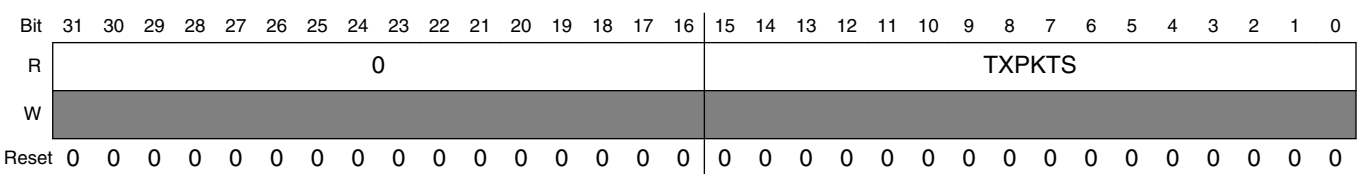
ENET_RMON_T_PACKETS field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Packet count Transmit packet count

23.5.35 Tx Broadcast Packets Statistic Register (ENET_RMON_T_BC_PKT)

RMON Tx Broadcast Packets

Address: 218_8000h base + 208h offset = 218_8208h



ENET_RMON_T_BC_PKT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Broadcast packets

23.5.36 Tx Multicast Packets Statistic Register (ENET_RMON_T_MC_PKT)

Address: 218_8000h base + 20Ch offset = 218_820Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_MC_PKT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Multicast packets

23.5.37 Tx Packets with CRC/Align Error Statistic Register (ENET_RMON_T_CRC_ALIGN)

Address: 218_8000h base + 210h offset = 218_8210h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_CRC_ALIGN field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Packets with CRC/align error

23.5.38 Tx Packets Less Than Bytes and Good CRC Statistic Register (ENET_RMON_T_UNDERSIZE)

Address: 218_8000h base + 214h offset = 218_8214h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_UNDERSIZE field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of transmit packets less than 64 bytes with good CRC

23.5.39 Tx Packets GT MAX_FL bytes and Good CRC Statistic Register (ENET_RMON_T_OVERSIZE)

Address: 218_8000h base + 218h offset = 218_8218h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_OVERSIZE field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of transmit packets greater than MAX_FL bytes with good CRC

23.5.40 Tx Packets Less Than 64 Bytes and Bad CRC Statistic Register (ENET_RMON_T_FRAG)

.

Address: 218_8000h base + 21Ch offset = 218_821Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_FRAG field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of packets less than 64 bytes with bad CRC

23.5.41 Tx Packets Greater Than MAX_FL bytes and Bad CRC Statistic Register (ENET_RMON_T_JAB)

Address: 218_8000h base + 220h offset = 218_8220h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_JAB field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of transmit packets greater than MAX_FL bytes and bad CRC

23.5.42 Tx Collision Count Statistic Register (ENET_RMON_T_COL)

Address: 218_8000h base + 224h offset = 218_8224h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_COL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of transmit collisions

23.5.43 Tx 64-Byte Packets Statistic Register (ENET_RMON_T_P64)

.

Address: 218_8000h base + 228h offset = 218_8228h

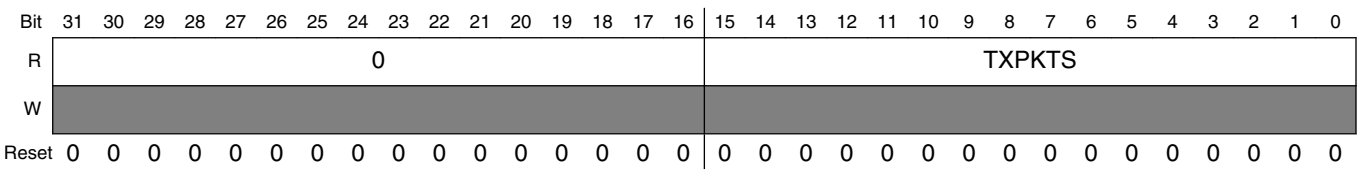
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_P64 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of 64-byte transmit packets

23.5.44 Tx 65- to 127-byte Packets Statistic Register (ENET_RMON_T_P65TO127)

Address: 218_8000h base + 22Ch offset = 218_822Ch

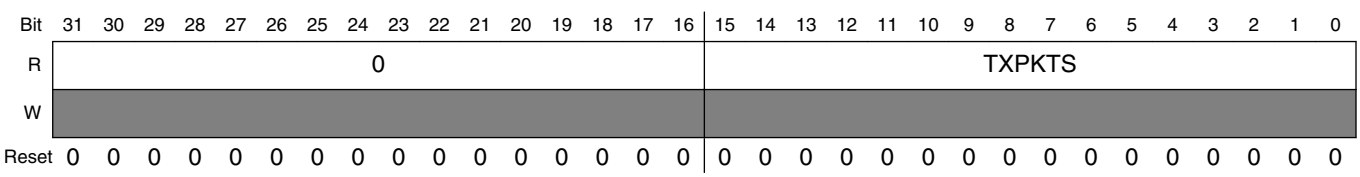


ENET_RMON_T_P65TO127 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of 65- to 127-byte transmit packets

23.5.45 Tx 128- to 255-byte Packets Statistic Register (ENET_RMON_T_P128TO255)

Address: 218_8000h base + 230h offset = 218_8230h



ENET_RMON_T_P128TO255 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of 128- to 255-byte transmit packets

23.5.46 Tx 256- to 511-byte Packets Statistic Register (ENET_RMON_T_P256TO511)

Address: 218_8000h base + 234h offset = 218_8234h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_P256TO511 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of 256- to 511-byte transmit packets

23.5.47 Tx 512- to 1023-byte Packets Statistic Register (ENET_RMON_T_P512TO1023)

.

Address: 218_8000h base + 238h offset = 218_8238h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_P512TO1023 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of 512- to 1023-byte transmit packets

23.5.48 Tx 1024- to 2047-byte Packets Statistic Register (ENET_RMON_T_P1024TO2047)

Address: 218_8000h base + 23Ch offset = 218_823Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_P1024TO2047 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of 1024- to 2047-byte transmit packets

23.5.49 Tx Packets Greater Than 2048 Bytes Statistic Register (ENET_RMON_T_P_GTE2048)

Address: 218_8000h base + 240h offset = 218_8240h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_P_GTE2048 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of transmit packets greater than 2048 bytes

23.5.50 Tx Octets Statistic Register (ENET_RMON_T_OCTETS)

Address: 218_8000h base + 244h offset = 218_8244h

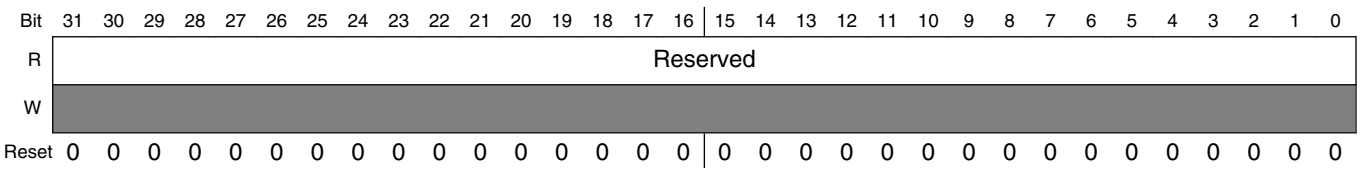
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXOCTS																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_OCTETS field descriptions

Field	Description
TXOCTS	Number of transmit octets

23.5.51 Reserved Statistic Register (ENET_IEEE_T_DROP)

Address: 218_8000h base + 248h offset = 218_8248h

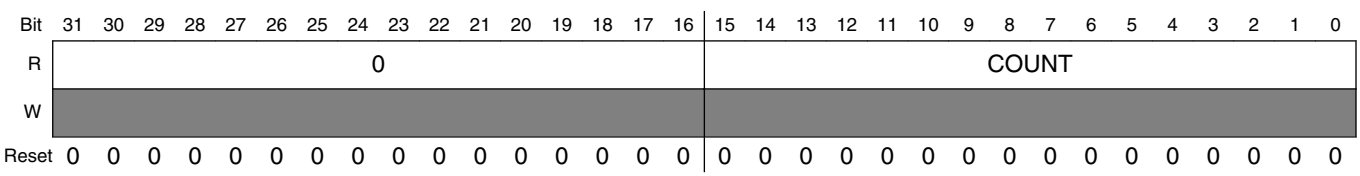


ENET_IEEE_T_DROP field descriptions

Field	Description
Reserved	This read-only field always has the value 0. This field is reserved.

23.5.52 Frames Transmitted OK Statistic Register (ENET_IEEE_T_FRAME_OK)

Address: 218_8000h base + 24Ch offset = 218_824Ch



ENET_IEEE_T_FRAME_OK field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted OK **NOTE:** Does not increment for the broadcast frames when broadcast reject is enabled and promiscuous mode is disabled within the receive control register (RCR).

23.5.53 Frames Transmitted with Single Collision Statistic Register (ENET_IEEE_T_1COL)

Address: 218_8000h base + 250h offset = 218_8250h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_T_1COL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted with one collision

23.5.54 Frames Transmitted with Multiple Collisions Statistic Register (ENET_IEEE_T_MCOL)

Address: 218_8000h base + 254h offset = 218_8254h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_T_MCOL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted with multiple collisions

23.5.55 Frames Transmitted after Deferral Delay Statistic Register (ENET_IEEE_T_DEF)

Address: 218_8000h base + 258h offset = 218_8258h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_T_DEF field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted with deferral delay

23.5.56 Frames Transmitted with Late Collision Statistic Register (ENET_IEEE_T_LCOL)

Address: 218_8000h base + 25Ch offset = 218_825Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_T_LCOL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted with late collision

23.5.57 Frames Transmitted with Excessive Collisions Statistic Register (ENET_IEEE_T_EXCOL)

Address: 218_8000h base + 260h offset = 218_8260h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_T_EXCOL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted with excessive collisions

23.5.58 Frames Transmitted with Tx FIFO Underrun Statistic Register (ENET_IEEE_T_MACERR)

Address: 218_8000h base + 264h offset = 218_8264h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_T_MACERR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted with transmit FIFO underrun

23.5.59 Frames Transmitted with Carrier Sense Error Statistic Register (ENET_IEEE_T_CSERR)

Address: 218_8000h base + 268h offset = 218_8268h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_T_CSERR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted with carrier sense error

23.5.60 Reserved Statistic Register (ENET_IEEE_T_SQE)

Address: 218_8000h base + 26Ch offset = 218_826Ch

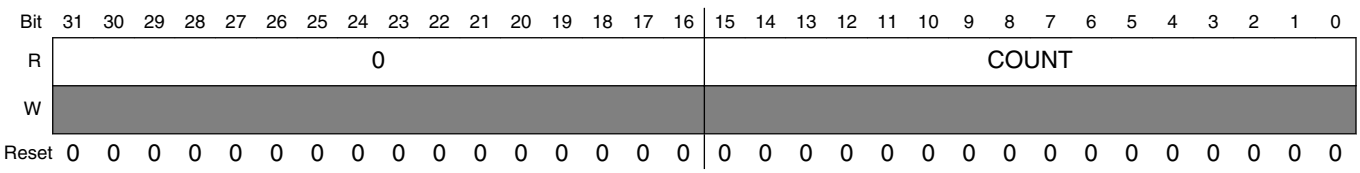
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_T_SQE field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	This read-only field is reserved and always has the value 0. **NOTE:** Counter not implemented as no SQE information is available.

23.5.61 Flow Control Pause Frames Transmitted Statistic Register (ENET_IEEE_T_FDXFC)

Address: 218_8000h base + 270h offset = 218_8270h

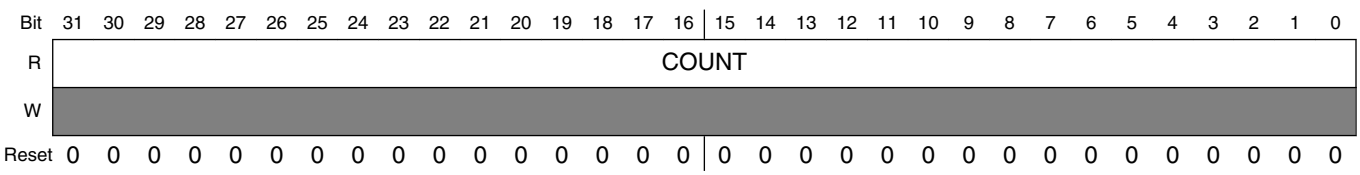


ENET_IEEE_T_FDXFC field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of flow-control pause frames transmitted

23.5.62 Octet Count for Frames Transmitted w/o Error Statistic Register (ENET_IEEE_T_OCTETS_OK)

Address: 218_8000h base + 274h offset = 218_8274h



ENET_IEEE_T_OCTETS_OK field descriptions

Field	Description
COUNT	Octet count for frames transmitted without error **NOTE** Counts total octets (includes header and FCS fields).

23.5.63 Rx Packet Count Statistic Register (ENET_RMON_R_PACKETS)

Address: 218_8000h base + 284h offset = 218_8284h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_PACKETS field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of packets received

23.5.64 Rx Broadcast Packets Statistic Register (ENET_RMON_R_BC_PKT)

Address: 218_8000h base + 288h offset = 218_8288h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_BC_PKT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of receive broadcast packets

23.5.65 Rx Multicast Packets Statistic Register (ENET_RMON_R_MC_PKT)

Address: 218_8000h base + 28Ch offset = 218_828Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_MC_PKT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of receive multicast packets

23.5.66 Rx Packets with CRC/Align Error Statistic Register (ENET_RMON_R_CRC_ALIGN)

Address: 218_8000h base + 290h offset = 218_8290h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_CRC_ALIGN field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of receive packets with CRC or align error

23.5.67 Rx Packets with Less Than 64 Bytes and Good CRC Statistic Register (ENET_RMON_R_UNDERSIZE)

Address: 218_8000h base + 294h offset = 218_8294h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_UNDERSIZE field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of receive packets with less than 64 bytes and good CRC

23.5.68 Rx Packets Greater Than MAX_FL and Good CRC Statistic Register (ENET_RMON_R_OVERSIZE)

Address: 218_8000h base + 298h offset = 218_8298h

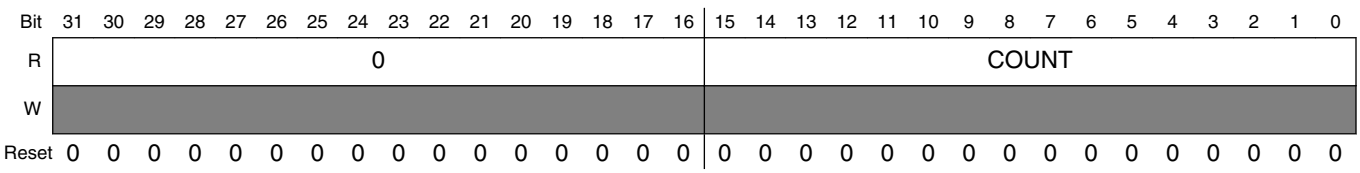
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_OVERSIZE field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of receive packets greater than MAX_FL and good CRC

23.5.69 Rx Packets Less Than 64 Bytes and Bad CRC Statistic Register (ENET_RMON_R_FRAG)

Address: 218_8000h base + 29Ch offset = 218_829Ch

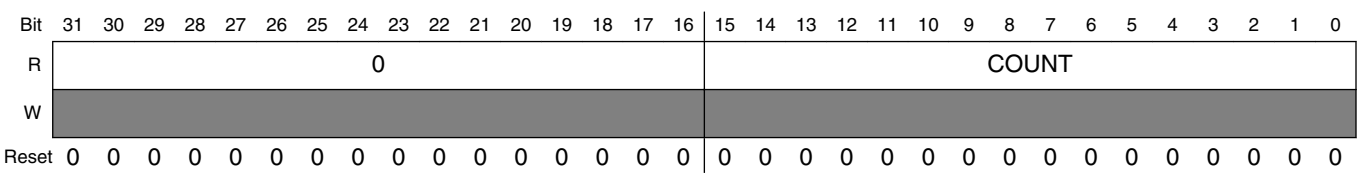


ENET_RMON_R_FRAG field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of receive packets with less than 64 bytes and bad CRC

23.5.70 Rx Packets Greater Than MAX_FL Bytes and Bad CRC Statistic Register (ENET_RMON_R_JAB)

Address: 218_8000h base + 2A0h offset = 218_82A0h

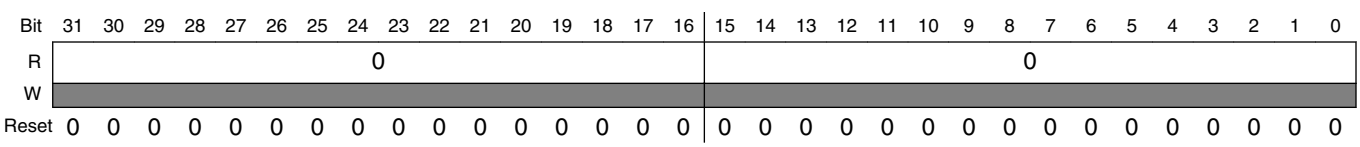


ENET_RMON_R_JAB field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of receive packets greater than MAX_FL and bad CRC

23.5.71 Reserved Statistic Register (ENET_RMON_R_RESVD_0)

Address: 218_8000h base + 2A4h offset = 218_82A4h



ENET_RMON_R_RESVD_0 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

23.5.72 Rx 64-Byte Packets Statistic Register (ENET_RMON_R_P64)

Address: 218_8000h base + 2A8h offset = 218_82A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_P64 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of 64-byte receive packets

23.5.73 Rx 65- to 127-Byte Packets Statistic Register (ENET_RMON_R_P65TO127)

Address: 218_8000h base + 2ACh offset = 218_82ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_P65TO127 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of 65- to 127-byte recieve packets

23.5.74 Rx 128- to 255-Byte Packets Statistic Register (ENET_RMON_R_P128TO255)

Address: 218_8000h base + 2B0h offset = 218_82B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_P128TO255 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of 128- to 255-byte recieve packets

23.5.75 Rx 256- to 511-Byte Packets Statistic Register (ENET_RMON_R_P256TO511)

Address: 218_8000h base + 2B4h offset = 218_82B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_P256TO511 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of 256- to 511-byte recieve packets

23.5.76 Rx 512- to 1023-Byte Packets Statistic Register (ENET_RMON_R_P512TO1023)

Address: 218_8000h base + 2B8h offset = 218_82B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_P512TO1023 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of 512- to 1023-byte recieve packets

23.5.77 Rx 1024- to 2047-Byte Packets Statistic Register (ENET_RMON_R_P1024TO2047)

Address: 218_8000h base + 2BCh offset = 218_82BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_P1024TO2047 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of 1024- to 2047-byte recieve packets

23.5.78 Rx Packets Greater than 2048 Bytes Statistic Register (ENET_RMON_R_P_GTE2048)

Address: 218_8000h base + 2C0h offset = 218_82C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_P_GTE2048 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of greater-than-2048-byte recieve packets

23.5.79 Rx Octets Statistic Register (ENET_RMON_R_OCTETS)

Address: 218_8000h base + 2C4h offset = 218_82C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_OCTETS field descriptions

Field	Description
COUNT	Number of receive octets

23.5.80 Frames not Counted Correctly Statistic Register (ENET_IEEE_R_DROP)

Counter increments if a frame with invalid or missing SFD character is detected and has been dropped. None of the other counters increments if this counter increments.

Address: 218_8000h base + 2C8h offset = 218_82C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_R_DROP field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Frame count

23.5.81 Frames Received OK Statistic Register (ENET_IEEE_R_FRAME_OK)

Address: 218_8000h base + 2CCh offset = 218_82CCh

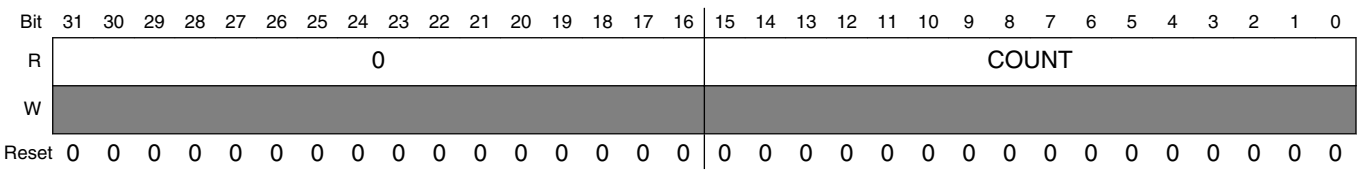
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_R_FRAME_OK field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames received OK

23.5.82 Frames Received with CRC Error Statistic Register (ENET_IEEE_R_CRC)

Address: 218_8000h base + 2D0h offset = 218_82D0h

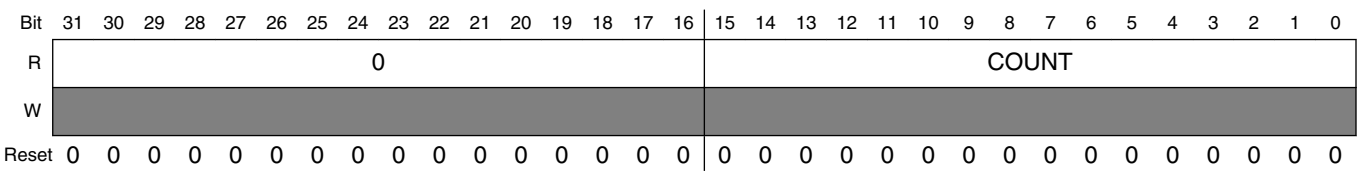


ENET_IEEE_R_CRC field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames received with CRC error

23.5.83 Frames Received with Alignment Error Statistic Register (ENET_IEEE_R_ALIGN)

Address: 218_8000h base + 2D4h offset = 218_82D4h



ENET_IEEE_R_ALIGN field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames received with alignment error

23.5.84 Receive FIFO Overflow Count Statistic Register (ENET_IEEE_R_MACERR)

Address: 218_8000h base + 2D8h offset = 218_82D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_R_MACERR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Receive FIFO overflow count

23.5.85 Flow Control Pause Frames Received Statistic Register (ENET_IEEE_R_FDXFC)

Address: 218_8000h base + 2DCh offset = 218_82DCh

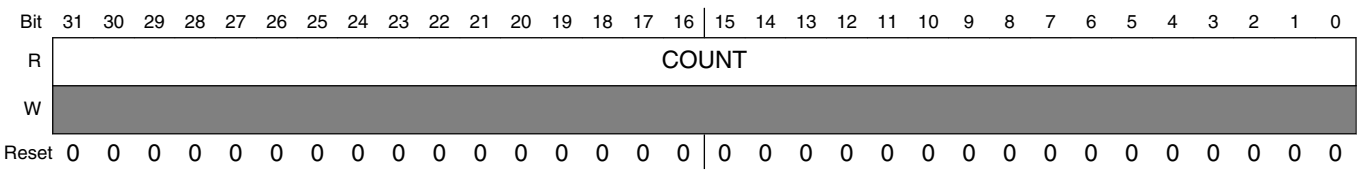
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_R_FDXFC field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of flow-control pause frames received

23.5.86 Octet Count for Frames Received without Error Statistic Register (ENET_IEEE_R_OCTETS_OK)

Address: 218_8000h base + 2E0h offset = 218_82E0h



ENET_IEEE_R_OCTETS_OK field descriptions

Field	Description
COUNT	Number of octets for frames received without error **NOTE:** Counts total octets (includes header and FCS fields). Does not increment for the broadcast frames when broadcast reject is enabled and promiscuous mode is disabled within the receive control register (RCR).

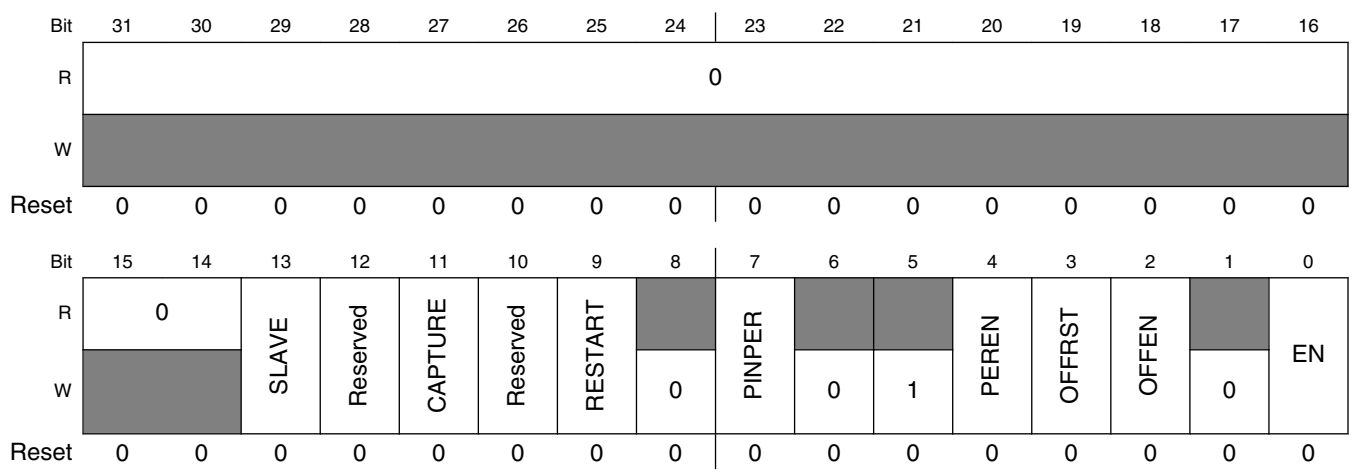
23.5.87 Adjustable Timer Control Register (ENET_ATCR)

ATCR command fields can trigger the corresponding events directly. It is not necessary to preserve any of the configuration fields when a command field is set in the register, that is, no read-modify-write is required.

NOTE

The CAPTURE and RESTART fields and bits 12 and 10 must be 0 in order to write to the other fields in this register.

Address: 218_8000h base + 400h offset = 218_8400h



ENET_ATCR field descriptions

Field	Description
31–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 SLAVE	Enable Timer Slave Mode 0 The timer is active and all configuration fields in this register are relevant. 1 The internal timer is disabled and the externally provided timer value is used. All other fields, except CAPTURE, in this register have no effect. CAPTURE can still be used to capture the current timer value.
12 Reserved	This field is reserved. Always write 0 to this field.
11 CAPTURE	Capture Timer Value When this field is set, all other fields are ignored during a write. This field automatically clears to 0 after the command completes. **NOTE:** To ensure that the correct time value is read from the ATVR register, a minimum amount of time must elapse from issuing this command to reading the ATVR register. This minimum time is defined by the greater of either six register clock cycles or six 1588/timestamp clock cycles. 0 No effect. 1 The current time is captured and can be read from the ATVR register.
10 Reserved	This field is reserved. Always write 0 to this field.
9 RESTART	Reset Timer Resets the timer to zero. This has no effect on the counter enable. If the counter is enabled when this field is set, the timer is reset to zero and starts counting from there. When set, all other fields are ignored during a write. This field automatically clears to 0 after the command completes. RESTART should be used when the timer is enabled. **NOTE:** The Reset Timer command requires at least 6 clock cycles of either the register clock or the 1588/timestamp clock, whichever is greater, to complete.
8 Reserved	This field is reserved.
7 PINPER	Enables event signal output assertion on period event. **NOTE:** Not all devices contain the event signal output. See the chip configuration details. 0 Disable. 1 Enable.
6 Reserved	This field is reserved.
5 Reserved	This field is reserved. **NOTE:** This field must be written always with one.
4 PEREN	Enable Periodical Event 0 Disable. 1 A period event interrupt can be generated (EIR[TS_TIMER]) and the event signal output is asserted when the timer wraps around according to the periodic setting ATPER. The timer period value must be set before setting this bit.

Table continues on the next page...

ENET_ATCR field descriptions (continued)

Field	Description
	NOTE: Not all devices contain the event signal output. See the chip configuration details.
3 OFFRST	Reset Timer On Offset Event 0 The timer is not affected and no action occurs, besides clearing OFFEN, when the offset is reached. 1 If OFFEN is set, the timer resets to zero when the offset setting is reached. The offset event does not cause a timer interrupt.
2 OFFEN	Enable One-Shot Offset Event 0 Disable. 1 The timer can be reset to zero when the given offset time is reached (offset event). The field is cleared when the offset event is reached, so no further event occurs until the field is set again. The timer offset value must be set before setting this field.
1 Reserved	This field is reserved.
0 EN	Enable Timer 0 The timer stops at the current value. 1 The timer starts incrementing.

23.5.88 Timer Value Register (ENET_ATVR)

Address: 218_8000h base + 404h offset = 218_8404h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ATIME																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_ATVR field descriptions

Field	Description
ATIME	A write sets the timer. A read returns the last captured value. To read the current value, issue a capture command (i.e., set ATCR[CAPTURE]) prior to reading this register.

23.5.89 Timer Offset Register (ENET_ATOFF)

Address: 218_8000h base + 408h offset = 218_8408h

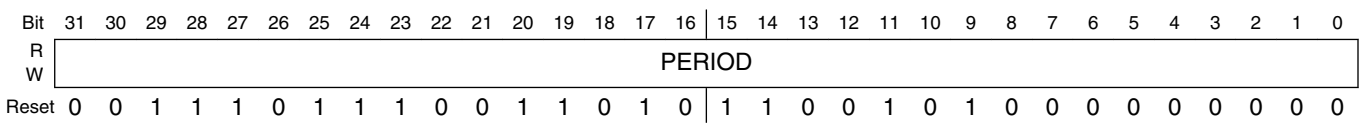
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OFFSET																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_ATOFF field descriptions

Field	Description
OFFSET	Offset value for one-shot event generation. When the timer reaches the value, an event can be generated to reset the counter. If the increment value in ATINC is given in true nanoseconds, this value is also given in true nanoseconds.

23.5.90 Timer Period Register (ENET_ATPER)

Address: 218_8000h base + 40Ch offset = 218_840Ch

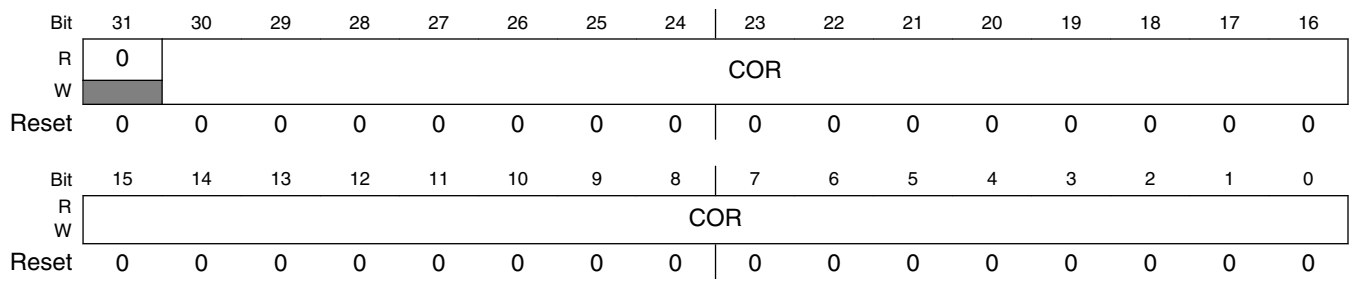


ENET_ATPER field descriptions

Field	Description
PERIOD	Value for generating periodic events. Each instance the timer reaches this value, the period event occurs and the timer restarts. If the increment value in ATINC is given in true nanoseconds, this value is also given in true nanoseconds. The value should be initialized to 1,000,000,000 (1×10^9) to represent a timer wrap around of one second. The increment value set in ATINC should be set to the true nanoseconds of the period of clock ts_clk, hence implementing a true 1 second counter. **NOTE:** The value of PERIOD has the following constraint: 2^{32} – ENET_ATINC[INC_COR] – 3×ENET_ATINC[INC] ≥ PERIOD > 0.

23.5.91 Timer Correction Register (ENET_ATCOR)

Address: 218_8000h base + 410h offset = 218_8410h



ENET_ATCOR field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COR	Correction Counter Wrap-Around Value

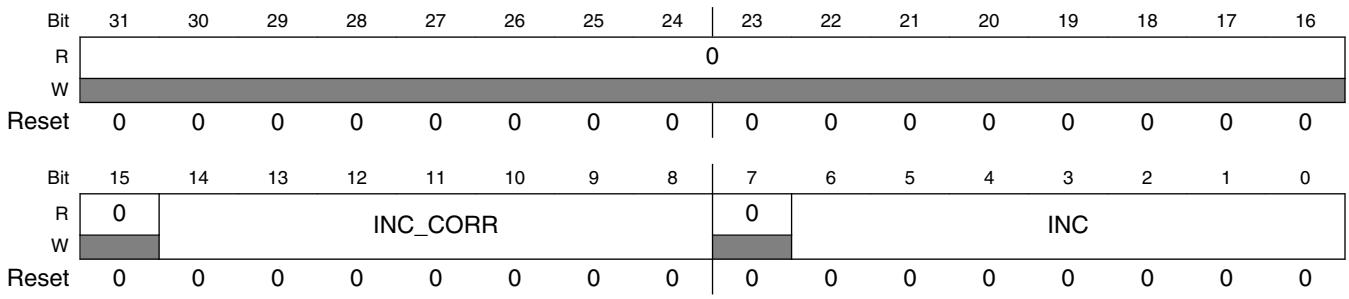
Table continues on the next page...

ENET_ATCOR field descriptions (continued)

Field	Description
	Defines after how many timer clock cycles (ts_clk) the correction counter should be reset and trigger a correction increment on the timer. The amount of correction is defined in ATINC[INC_CORR]. A value of 0 disables the correction counter and no corrections occur. **NOTE:** This value is given in clock cycles, not in nanoseconds as all other values.

23.5.92 Time-Stamping Clock Period Register (ENET_ATINC)

Address: 218_8000h base + 414h offset = 218_8414h

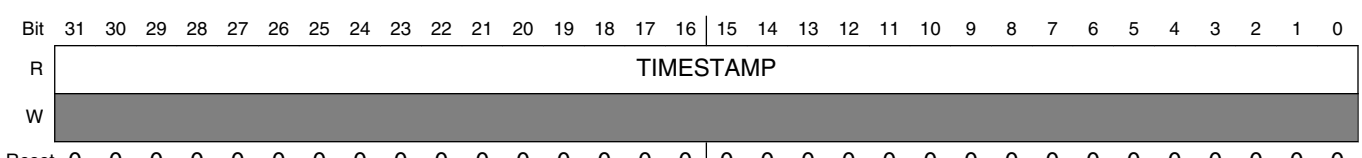


ENET_ATINC field descriptions

Field	Description
31–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 INC_CORR	Correction Increment Value This value is added every time the correction timer expires (every clock cycle given in ATCOR). A value less than INC slows down the timer. A value greater than INC speeds up the timer.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
INC	Clock Period Of The Timestamping Clock (ts_clk) In Nanoseconds The timer increments by this amount each clock cycle. For example, set to 10 for 100 MHz, 8 for 125 MHz, 5 for 200 MHz. **NOTE:** For highest precision, use a value that is an integer fraction of the period set in ATPER.

23.5.93 Timestamp of Last Transmitted Frame (ENET_ATSTMP)

Address: 218_8000h base + 418h offset = 218_8418h



ENET_ATSTMP field descriptions

Field	Description
TIMESTAMP	Timestamp of the last frame transmitted by the core that had TxBD[TS] set . This register is only valid when EIR[TS_AVAIL] is set.

23.5.94 Timer Global Status Register (ENET_TGSR)

Address: 218_8000h base + 604h offset = 218_8604h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

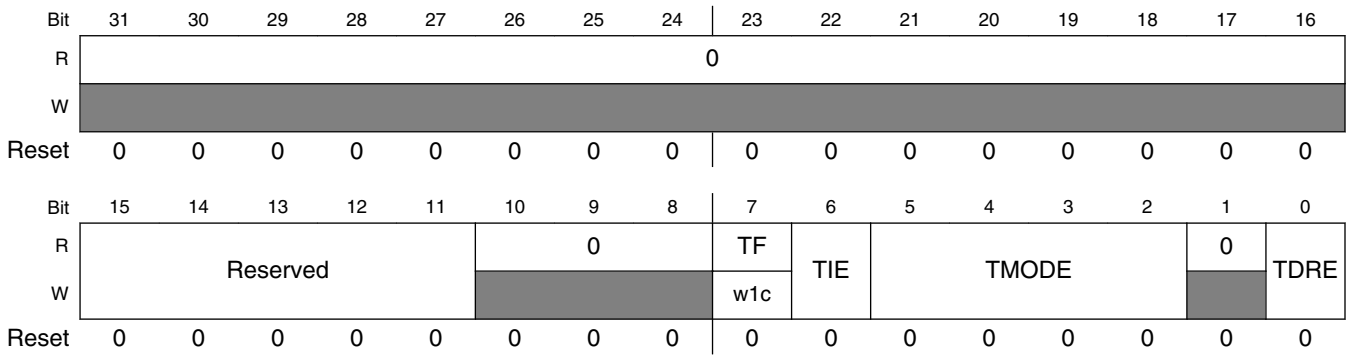
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												TF3	TF2	TF1	TF0
W													w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_TGSR field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 TF3	Copy Of Timer Flag For Channel 3 0 Timer Flag for Channel 3 is clear 1 Timer Flag for Channel 3 is set
2 TF2	Copy Of Timer Flag For Channel 2 0 Timer Flag for Channel 2 is clear 1 Timer Flag for Channel 2 is set
1 TF1	Copy Of Timer Flag For Channel 1 0 Timer Flag for Channel 1 is clear 1 Timer Flag for Channel 1 is set
0 TF0	Copy Of Timer Flag For Channel 0 0 Timer Flag for Channel 0 is clear 1 Timer Flag for Channel 0 is set

23.5.95 Timer Control Status Register (ENET_TCSR*n*)

Address: 218_8000h base + 608h offset + (8d × i), where i=0d to 3d



ENET_TCSR*n* field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–11 Reserved	This field is reserved. This field must be written with 0.
10–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 TF	Timer Flag Sets when input capture or output compare occurs. This flag is double buffered between the module clock and 1588 clock domains. When this field is 1, it can be cleared to 0 by writing 1 to it. 0 Input Capture or Output Compare has not occurred. 1 Input Capture or Output Compare has occurred.
6 TIE	Timer Interrupt Enable 0 Interrupt is disabled 1 Interrupt is enabled
5–2 TMODE	Timer Mode Updating the Timer Mode field takes a few cycles to register because it is synchronized to the 1588 clock. The version of Timer Mode returned on a read is from the 1588 clock domain. When changing Timer Mode, always disable the channel and read this register to verify the channel is disabled first. 0000 Timer Channel is disabled. 0001 Timer Channel is configured for Input Capture on rising edge. 0010 Timer Channel is configured for Input Capture on falling edge. 0011 Timer Channel is configured for Input Capture on both edges. 0100 Timer Channel is configured for Output Compare - software only. 0101 Timer Channel is configured for Output Compare - toggle output on compare. 0110 Timer Channel is configured for Output Compare - clear output on compare. 0111 Timer Channel is configured for Output Compare - set output on compare. 1000 Reserved

Table continues on the next page...

ENET_TCSR*n* field descriptions (continued)

Field	Description
	1010 Timer Channel is configured for Output Compare - clear output on compare, set output on overflow. 10X1 Timer Channel is configured for Output Compare - set output on compare, clear output on overflow. 110X Reserved 1110 Timer Channel is configured for Output Compare - pulse output low on compare for one 1588-clock cycle. 1111 Timer Channel is configured for Output Compare - pulse output high on compare for one 1588-clock cycle.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 TDRE	Timer DMA Request Enable 0 DMA request is disabled 1 DMA request is enabled

23.5.96 Timer Compare Capture Register (ENET_TCCR*n*)

Address: 218_8000h base + 60Ch offset + (8d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TCC																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_TCCR*n* field descriptions

Field	Description
TCC	Timer Capture Compare This register is double buffered between the module clock and 1588 clock domains. When configured for compare, the 1588 clock domain updates with the value in the module clock domain whenever the Timer Channel is first enabled and on each subsequent compare. Write to this register with the first compare value before enabling the Timer Channel. When the Timer Channel is enabled, write the second compare value either immediately, or at least before the first compare occurs. After each compare, write the next compare value before the previous compare occurs and before clearing the Timer Flag. The compare occurs one 1588 clock cycle after the IEEE 1588 Counter increments past the compare value in the 1588 clock domain. If the compare value is less than the value of the 1588 Counter when the Timer Channel is first enabled, then the compare does not occur until following the next overflow of the 1588 Counter. If the compare value is greater than the IEEE 1588 Counter when the 1588 Counter overflows, or the compare value is less than the value of the IEEE 1588 Counter after the overflow, then the compare occurs one 1588 clock cycle following the overflow. When configured for capture, the value of the IEEE 1588 Counter is captured into the 1588 clock domain and then updated into the module clock domain, provided the Timer Flag is clear. Always read the capture value before clearing the Timer Flag.

23.6 Functional description

This section provides a complete functional description of the MAC-NET core.

23.6.1 Ethernet MAC frame formats

The IEEE 802.3 standard defines the Ethernet frame format as follows:

- Minimum length of 64 bytes
- Maximum length of 1518 bytes excluding the preamble and the start frame delimiter (SFD) bytes

An Ethernet frame consists of the following fields:

- Seven bytes preamble
- Start frame delimiter (SFD)
- Two address fields
- Length or type field
- Data field
- Frame check sequence (CRC value)
- Extension field is defined only for Gigabit Ethernet half-duplex implementations and is not supported by the MAC core

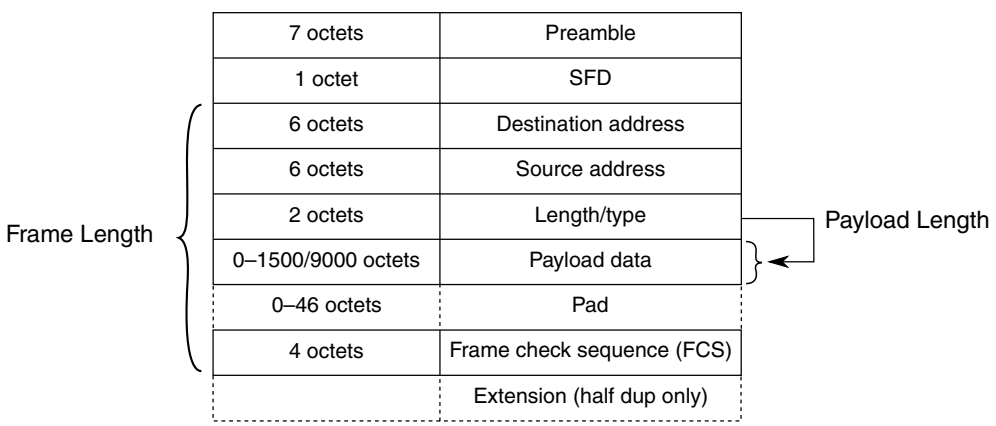


Figure 23-2. MAC frame format overview

Optionally, MAC frames can be VLAN-tagged with an additional four-byte field inserted between the MAC source address and the type/length field. VLAN tagging is defined by the IEEE P802.1q specification. VLAN-tagged frames have a maximum length of 1522 bytes, excluding the preamble and the SFD bytes.

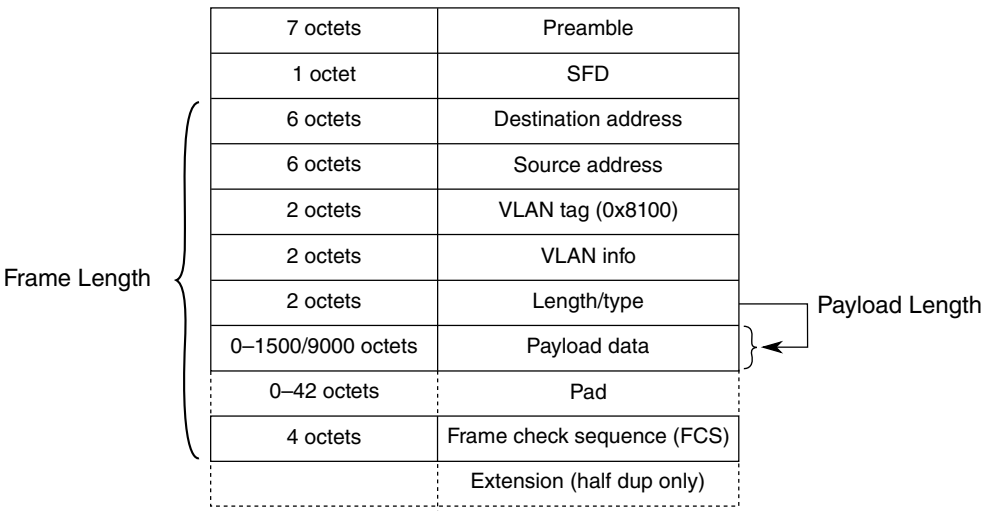


Figure 23-3. VLAN-tagged MAC frame format overview

Table 23-5. MAC frame definition

Term	Description
Frame length	Defines the length, in octets, of the complete frame without preamble and SFD. A frame has a valid length if it contains at least 64 octets and does not exceed the programmed maximum length.
Payload length	The length/type field indicates the length of the frame's payload section. The most significant byte is sent/received first. • If the length/type field is set to a value less than 46, the payload is padded so that the minimum frame length requirement (64 bytes) is met. For VLAN-tagged frames, a value less than 42 indicates a padded frame. • If the length/type field is set to a value larger than the programmed frame maximum length (e.g. 1518) it is interpreted as a type field.
Destination and source address	48-bit MAC addresses. The least significant byte is sent/received first and the first two least significant bits of the MAC address distinguish MAC frames, as detailed in MAC address check.

Note

Although the IEEE specification defines a maximum frame length, the MAC core provides the flexibility to program any value for the frame maximum length.

23.6.1.1 Pause Frames

The receiving device generates a pause frame to indicate a congestion to the emitting device, which should stop sending data.

Pause frames are indicated by the length/type set to 0x8808. The two first bytes of a pause frame following the type, defines a 16-bit opcode field set to 0x0001 always. A 16-bit pause quanta is defined in the frame payload bytes 2 (P1) and 3 (P2) as defined in the following table. The P1 pause quanta byte is the most significant.

Table 23-6. Pause Frame Format (Values in Hex)

1	2	3	4	5	6	7	8	9	10	11	12	13	14
55	55	55	55	55	55	55	D5	01	80	C2	00	00	01
Preamble							SFD	Multicast Destination Address					
15	16	17	18	19	20	21	22	23	24	25	26	27 –68	
00	00	00	00	00	00	88	08	00	01	hi	lo	00	
Source Address						Type		Opcode		P1	P2	pad (42)	
69	70	71	72										
26	6B	AE	0A										
CRC-32													

There is no payload length field found within a pause frame and a pause frame is always padded with 42 bytes (0x00).

If a pause frame with a pause value greater than zero (XOFF condition) is received, the MAC stops transmitting data as soon the current frame transfer is completed. The MAC stops transmitting data for the value defined in pause quanta. One pause quanta fraction refers to 512 bit times.

If a pause frame with a pause value of zero (XON condition) is received, the transmitter is allowed to send data immediately (see Full-duplex flow control operation for details).

23.6.1.2 Magic packets

A magic packet is a unicast, multicast, or broadcast packet, which carries a defined sequence in the payload section.

Magic packets are received and inspected only under specific conditions as described in Magic packet detection.

The defined sequence to decode a magic packet is formed with a synchronization stream which consists of six consecutive 0xFF bytes, and is followed by sequence of sixteen consecutive unicast MAC addresses of the node to be awakened.

This sequence can be located anywhere in the magic packet payload. The magic packet is formed with a standard Ethernet header, optional padding, and CRC.

23.6.2 IP and higher layers frame format

The following sections use the term datagram to describe the protocol specific data unit that is found within the payload section of its container entity.

For example, an IP datagram specifies the payload section of an Ethernet frame. A TCP datagram specifies the payload section within an IP datagram.

23.6.2.1 Ethernet types

IP datagrams are carried in the payload section of an Ethernet frame. The Ethernet frame type/length field discriminates several datagram types.

The following table lists the types of interest:

Table 23-7. Ethernet type value examples

Type	Description
0x8100	VLAN-tagged frame. The actual type is found 4 octets later in the frame.
0x0800	IPv4
0x0806	ARP
0x86DD	IPv6

23.6.2.2 IPv4 datagram format

The following figure shows the IP Version 4 (IPv4) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words. The first byte sent/received is the leftmost byte of the first word (in other words, version/IHL field).

The IP header can contain further options, which are always padded if necessary to guarantee the payload following the header is aligned to a 32-bit boundary.

The IP header is immediately followed by the payload, which can contain further protocol headers (for example, TCP or UDP, as indicated by the protocol field value). The complete IP datagram is transported in the payload section of an Ethernet frame.

Table 23-8. IPv4 header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Version	IHL	TOS		Length			
Fragment ID				Flags	Fragment offset		
TTL		Protocol		Header checksum			
Source address							
Destination address							
Options							

Table 23-9. IPv4 header fields

Field name	Description
Version	4-bit IP version information. 0x4 for IPv4 frames.
IHL	4-bit Internet header length information. Determines number of 32-bit words found within the IP header. If no options are present, the default value is 0x5.
TOS	Type of service/DiffServ field.
Length	Total length of the datagram in bytes, including all octets of header and payload.
Fragment ID, flags, fragment offset	Fields used for IP fragmentation.
TTL	Time-to-live. In effect, is decremented at each router arrival. If zero, datagram must be discarded.
Protocol	Identifier of protocol that follows in the datagram.
Header checksum	Checksum of IP header. For computational purposes, this field's value is zero.
Source address	Source IP address.
Destination address	Destination IP address.

23.6.2.3 IPv6 datagram format

The following figure shows the IP version 6 (IPv6) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words and has a fixed length of ten words (40 bytes). The next header field identifies the type of the header that follows the IPv6 header. It is defined similar to the protocol identifier within IPv4, with new definitions for identifying extension headers. These headers can be inserted between the IPv6 header and the protocol header, which will shift the protocol header accordingly.The accelerator currently only supports IPv6 without extension headers (in other words, the next header specifies TCP, UDP, or IMCP).

The first byte sent/received is the leftmost byte of the first word (in other words, version/ traffic class fields).

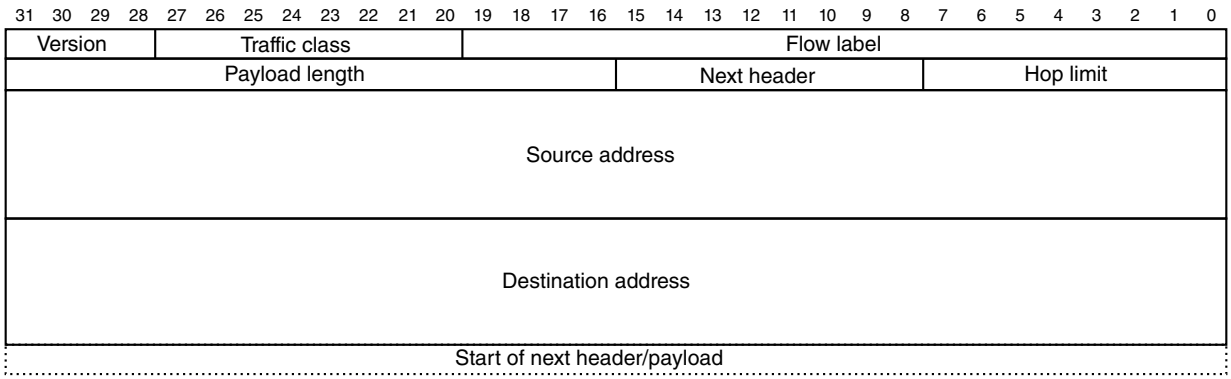


Figure 23-4. IPv6 header format

Table 23-10. IPv6 header fields

Field name	Description
Version	4-bit IP version information. 0x6 for all IPv6 frames.
Traffic class	8-bit field defining the traffic class.
Flow label	20-bit flow label identifying frames of the same flow.
Payload length	16-bit length of the datagram payload in bytes. It includes all octets following the IPv6 header.
Next header	Identifies the header that follows the IPv6 header. This can be the protocol header or any IPv6 defined extension header.
Hop limit	Hop counter, decremented by one by each station that forwards the frame. If hop limit is 0 the frame must be discarded.
Source address	128-bit IPv6 source address.
Destination address	128-bit IPv6 destination address.

23.6.2.4 Internet Control Message Protocol (ICMP) datagram format

An internet control message protocol (ICMP) is found following the IP header, if the protocol identifier is 1. The ICMP datagram has a four-octet header followed by additional message data.

Table 23-11. ICMP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Type		Code		Checksum			
ICMP message data							

Table 23-12. IP header fields

Field name	Description
Type	8-bit type information
Code	8-bit code that is related to the message type

Table continues on the next page...

Table 23-12. IP header fields (continued)

Field name	Description
Checksum	16-bit one's complement checksum over the complete ICMP datagram

23.6.2.5 User Datagram Protocol (UDP) datagram format

A user datagram protocol header is found after the IP header, when the protocol identifier is 17.

The payload of the datagram is after the UDP header. The header byte order follows the conventions given for the IP header above.

Table 23-13. UDP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source port				Destination port			
Length				Checksum			

Table 23-14. UDP header fields

Field name	Description
Source port	Source application port
Destination port	Destination application port
Length	Length of user data which immediately follows the header, including the UDP header (that is, minimum value is 8)
Checksum	Checksum over the complete datagram and some IP header information

23.6.2.6 TCP datagram format

A TCP header is found following the IP header, when the protocol identifier has a value of 6.

The TCP payload immediately follows the TCP header.

Table 23-15. TCP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source port				Destination port			
Sequence number							
Acknowledgement number							

Table continues on the next page...

Table 23-15. TCP header format (continued)

Offset	Reserved	Flags	Window	
Checksum			Urgent pointer	
Options				

Table 23-16. TCP header fields

Field name	Description
Source port	Source application port
Destination port	Destination application port
Sequence number	Transmit sequence number
Ack. number	Receive sequence number
Offset	Data offset, which is number of 32-bit words within TCP header — if no options selected, defaults to value of 5
Flags	URG, ACK, PSH, RST, SYN, FIN flags
Window	TCP receive window size information
Checksum	Checksum over the complete datagram (TCP header and data) and IP header information
Options	Additional 32-bit words for protocol options

23.6.3 IEEE 1588 message formats

The following sections describe the IEEE 1588 message formats.

23.6.3.1 Transport encapsulation

The precision time protocol (PTP) datagrams are encapsulated in Ethernet frames using the UDP/IP transport mechanism, or optionally, with the newer 1588v2 directly in Ethernet frames (layer 2).

Typically, multicast addresses are used to allow efficient distribution of the synchronization messages.

23.6.3.1.1 UDP/IP

The 1588 messages (v1 and v2) can be transported using UDP/IP multicast messages.

Table 23-17 shows IP multicast groups defined for PTP. The table also shows their respective MAC layer multicast address mapping according to RFC 1112 (last three octets of IP follow the fixed value of 01-00-5E).

Table 23-17. UDP/IP multicast domains

Name	IP Address	MAC Address mapping
DefaultPTPdomain	224.0.1.129	01-00-5E-00-01-81
AlternatePTPdomain1	224.0.1.130	01-00-5E-00-01-82
AlternatePTPdomain2	224.0.1.131	01-00-5E-00-01-83
AlternatePTPdomain3	224.0.1.132	01-00-5E-00-01-84

Table 23-18. UDP port numbers

Message type	UDP port	Note
Event	319	Used for SYNC and DELAY_REQUEST messages
General	320	All other messages (for example, follow-up, delay-response)

23.6.3.1.2 Native Ethernet (PTPv2)

In addition to using UDP/IP frames, IEEE 1588v2 defines a native Ethernet frame format that uses ethertype = 0x88F7. The payload of the Ethernet frame immediately contains the PTP datagram, starting with the PTPv2 header.

Besides others, version 2 adds a peer delay mechanism to allow delay measurements between individual point-to-point links along a path over multiple nodes. The following multicast domains are also defined in PTPv2.

Table 23-19. PTPv2 multicast domains

Name	MAC address
Normal messages	01-1B-19-00-00-00
Peer delay messages	01-80-C2-00-00-0E

23.6.3.2 PTP header

All PTP frames contain a common header that determines the protocol version and the type of message, which defines the remaining content of the message.

All multi-octet fields are transmitted in big-endian order (the most significant byte is transmitted/received first).

The last four bits of versionPTP are at the same position (second byte) for PTPv1 and PTPv2 headers. This allows accurate identification by inspecting the first two bytes of the message.

23.6.3.2.1 PTPv1 header

Table 23-20. Common PTPv1 message header

Offset	Octets	Bits							
		7	6	5	4	3	2	1	0
0	2	versionPTP = 0x0001							
2	2	versionNetwork							
4	16	subdomain							
20	1	messageType							
21	1	sourceCommunicationTechnology							
22	6	sourceUuid							
28	2	sourcePortId							
30	2	sequenceId							
32	1	control							
33	1	0x00							
34	2	flags							
36	4	reserved							

The type of message is encoded in the messageType and control fields as shown in Table 23-21 :

Table 23-21. PTPv1 message type identification

messageType	control	Message Name	Message
0x01	0x0	SYNC	Event message
0x01	0x1	DELAY_REQ	Event message
0x02	0x2	FOLLOW_UP	General message
0x02	0x3	DELAY_RESP	General message
0x02	0x4	MANAGEMENT	General message
other	other	—	Reserved

The field sequenceId is used to non-ambiguously identify a message.

23.6.3.2.2 PTPv2 header

Table 23-22. Common PTPv2 message header

Offset	Octets	Bits							
		7	6	5	4	3	2	1	0
0	1	transportSpecific				messageId			
1	1	reserved				versionPTP = 0x2			
2	2	messageLength							
4	1	domainNumber							
5	1	reserved							
6	2	flags							
8	8	correctionField							
16	4	reserved							
20	10	sourcePortIdentity							
30	2	sequenceId							
32	1	control							
33	1	logMeanMessageInterval							

The type of message is encoded in the field messageId as follows:

Table 23-23. PTPv2 message type identification

messageId	Message name	Message
0x0	SYNC	Event message
0x1	DELAY_REQ	Event message
0x2	PATH_DELAY_REQ	Event message
0x3	PATH_DELAY_RESP	Event message
0x4–0x7	—	Reserved
0x8	FOLLOW_UP	General message
0x9	DELAY_RESP	General message
0xa	PATH_DELAY_FOLLOW_UP	General message
0xb	ANNOUNCE	General message
0xc	SIGNALING	General message
0xd	MANAGEMENT	General message

The PTPv2 flags field contains further details on the type of message, especially if one-step or two-step implementations are used. The one- or two-step implementation is controlled by the TWO_STEP bit in the first octet of the flags field as shown below. Reserved bits are cleared.

Table 23-24. PTPv2 message flags field definitions

Bit	Name	Description
0	ALTERNATE_MASTER	See IEEE 1588 Clause 17.4
1	TWO_STEP	1 Two-step clock 0 One-step clock
2	UNICAST	1 Transport layer address uses a unicast destination address 0 Multicast is used
3	—	Reserved
4	—	Reserved
5	Profile specific	
6	Profile specific	
7	—	Reserved

23.6.4 MAC receive

The MAC receive engine performs the following tasks:

- Check frame framing
- Remove frame preamble and frame SFD field
- Discard frame based on frame destination address field
- Terminate pause frames
- Check frame length
- Remove payload padding if it exists
- Calculate and verify CRC-32
- Write received frames in the core receive FIFO

If the MAC is programmed to operate in half-duplex mode, it will also check if the frame is received with a collision.

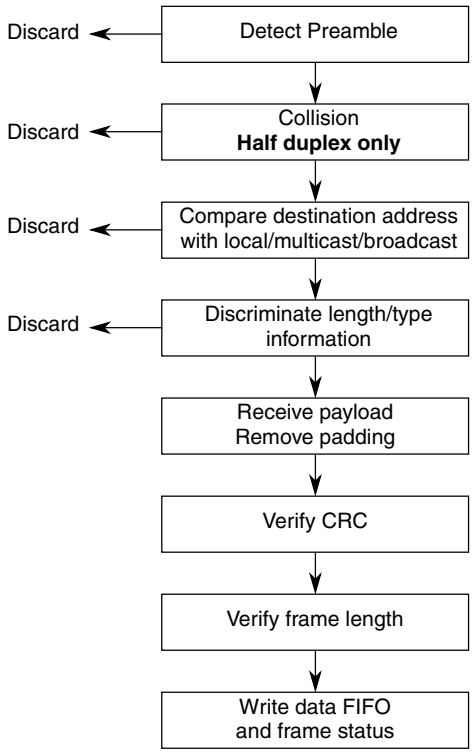


Figure 23-5. MAC receive flow

23.6.4.1 Collision detection in half-duplex mode

If the packet is received with a collision detected during reception of the first 64 bytes, the packet is discarded (if frame size was less than ~14 octets) or transmitted to the user application with an error and RxBD[CE] set.

23.6.4.2 Preamble processing

The IEEE 802.3 standard allows a maximum size of 56 bits (seven bytes) for the preamble, while the MAC core allows any preamble length, including zero length preamble.

The MAC core checks for the start frame delimiter (SFD) byte. If the next byte of the preamble, which is different from 0x55, is not 0xD5, the frame is discarded.

Although the IEEE specification dictates that the inner-packet gap should be at least 96 bits, the MAC core is designed to accept frames separated by only 64 10/100-Mbit/s operation (MII) bits.

The MAC core removes the preamble and SFD bytes.

23.6.4.3 MAC address check

The destination address bit 0 differentiates between multicast and unicast addresses.

- If bit 0 is 0, the MAC address is an individual (unicast) address.
- If bit 0 is 1, the MAC address defines a group (multicast) address.
- If all 48 bits of the MAC address are set, it indicates a broadcast address.

23.6.4.3.1 Unicast address check

If a unicast address is received, the destination MAC address is compared to the node MAC address programmed by the host in the PADDR1/2 registers.

If the destination address matches any of the programmed MAC addresses, the frame is accepted.

If Promiscuous mode is enabled (RCR[PROM] = 1) no address checking is performed and all unicast frames are accepted.

23.6.4.3.2 Multicast and unicast address resolution

The hash table algorithm used in the group and individual hash filtering operates as follows.

- The 48-bit destination address is mapped into one of 64 bits, represented by 64 bits in ENET*n*_GAUR/GALR (group address hash match) or ENET*n*_IAUR/IALR (individual address hash match).
- This mapping is performed by passing the 48-bit address through the on-chip 32-bit CRC generator and selecting the six most significant bits of the CRC-encoded result to generate a number between 0 and 63.
- The msb of the CRC result selects ENET*n*_GAUR (msb = 1) or ENET*n*_GALR (msb = 0).
- The five lsbs of the hash result select the bit within the selected register.
- If the CRC generator selects a bit set in the hash table, the frame is accepted; else, it is rejected.

For example, if eight group addresses are stored in the hash table and random group addresses are received, the hash table prevents roughly 56/64 (or 87.5%) of the group address frames from reaching memory. Those that do reach memory must be further filtered by the processor to determine if they truly contain one of the eight desired addresses.

The effectiveness of the hash table declines as the number of addresses increases.

The user must initialize the hash table registers. Use this CRC32 polynomial to compute the hash:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$

If Promiscuous mode is enabled (ENET*n*_RCR[PROM] = 1) all unicast and multicast frames are accepted regardless of ENET*n*_GAUR/GALR and ENET*n*_IAUR/IALR settings.

23.6.4.3.3 Broadcast address reject

All broadcast frames are accepted if BC_REJ is cleared or ENET*n*_RCR[PROM] is set. If PROM is cleared when ENET*n*_RCR[BC_REJ] is set, all broadcast frames are rejected.

Table 23-25. Broadcast address reject programming

PROM	BC_REJ	Broadcast frames
0	0	Accepted
0	1	Rejected
1	0	Accepted
1	1	Accepted

23.6.4.3.4 Miss-bit implementation

For higher layer filtering purposes, RxBD[M] indicates an address miss when the MAC operates in promiscuous mode and accepts a frame that would otherwise be rejected.

If a group/individual hash or exact match does not occur and Promiscuous mode is enabled (RCR[PROM] = 1), the frame is accepted and the M bit is set in the buffer descriptor; otherwise, the frame is rejected.

This means the status bit is set in any of the following conditions during Promiscuous mode:

- A broadcast frame is received when BC_REJ is set
- A unicast is received that does not match either:

- Node address (PALR[PADDR1] and PAUR[PADDR2])
- Hash table for unicast (IAUR[IADDR1] and IALR[IADDR2])

- A multicast is received that does not match the GAUR[GADDR1] and GALR[GADDR2] hash table entries

23.6.4.4 Frame length/type verification: payload length check

If the length/type is less than 0x600 and NLC is set, the MAC checks the payload length and reports any error in the frame status word and interrupt bit PLR.

If the length/type is greater than or equal to 0x600, the MAC interprets the field as a type and no payload length check is performed.

The length check is performed on VLAN and stacked VLAN frames. If a padded frame is received, no length check can be performed due to the extended frame payload because padded frames can never have a payload length error.

23.6.4.5 Frame length/type verification: frame length check

When the receive frame length exceeds MAX_FL bytes, the BABR interrupt is generated and the RxBD[LG] bit is set.

The frame is not truncated unless the frame length exceeds the value programmed in ENET*n*_FTRL[TRUNC_FL]. If the frame is truncated, RxBD[TR] is set. In addition, a truncated frame always has the CRC error indication set (RxBD[CR]).

23.6.4.6 VLAN frames processing

VLAN frames have a length/type field set to 0x8100 immediately followed by a 16-Bit VLAN control information field.

VLAN-tagged frames are received as normal frames because the VLAN tag is not interpreted by the MAC function, and are pushed complete with the VLAN tag to the user application. If the length/type field of the VLAN-tagged frame, which is found four octets later in the frame, is less than 42, the padding is removed. In addition, the frame status word (RxBD[NO]) indicates that the current frame is VLAN tagged.

23.6.4.7 Pause frame termination

The receive engine terminates pause frames and does not transfer them to the receive FIFO. The quanta is extracted and sent to the MAC transmit path via a small internal clock rate decoupling asynchronous FIFO.

The quanta is written only if a correct CRC and frame length are detected by the control state machine. If not, the quanta is discarded and the MAC transmit path is not paused.

Good pause frames are ignored if ENET*n*_RCR[FCE] is cleared and are forwarded to the client interface when ENET*n*_RCR[PAUFWD] is set.

23.6.4.8 CRC check

The CRC-32 field is checked and forwarded to the core FIFO interface if ENET*n*_RCR[CRCFWD] is cleared and ENET*n*_RCR[PADEN] is cleared. When CRCFWD is set (regardless of PADEN), the CRC-32 field is checked and terminated (not transmitted to the FIFO).

The CRC polynomial, as specified in the 802.3 standard, is:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

The 32 bits of the CRC value are placed in the frame check sequence (FCS) field with the x^{31} term as right-most bit of the first octet. The CRC bits are thus received in the following order: x^{31}, x^{30},..., x^{1}, x^{0}.

If a CRC error is detected, the frame is marked invalid and RxBD[CR] is set.

23.6.4.9 Frame padding removal

When a frame is received with a payload length field set to less than 46 (42 for VLAN-tagged frames and 38 for frames with stacked VLANs), the zero padding can be removed before the frame is written into the data FIFO depending on the setting of ENET*n*_RCR[PADEN].

Note

If a frame is received with excess padding (in other words, the length field is set as mentioned above, but the frame has more than 64 octets) and padding removal is enabled, then the

padding is removed as normal and no error is reported if the frame is otherwise correct (for example: good CRC, less than maximum length, and no other error).

23.6.5 MAC transmit

Frame transmission starts when the transmit FIFO holds enough data.

After a transfer starts, the MAC transmit function performs the following tasks:

- Generates preamble and SFD field before frame transmission
- Generates XOFF pause frames if the receive FIFO reports a congestion or if ENET*n*_TCR[TFC_PAUSE] is set with ENET*n*_OPD[PAUSE_DUR] set to a non-zero value
- Generates XON pause frames if the receive FIFO congestion condition is cleared or if TFC_PAUSE is set with PAUSE_DUR cleared
- Suspends Ethernet frame transfer (XOFF) if a non-zero pause quanta is received from the MAC receive path
- Adds padding to the frame if required
- Calculates and appends CRC-32 to the transmitted frame
- Sends the frame with correct inter-packet gap (IPG) (deferring)

When the MAC is configured to operate in half-duplex mode, the following additional tasks are performed:

- Collision detection
- Frame retransmit after back-off timer expires

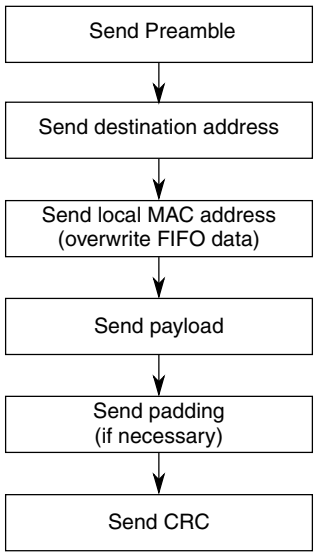


Figure 23-6. Frame transmit overview

23.6.5.1 Frame payload padding

The IEEE specification defines a minimum frame length of 64 bytes.

If the frame sent to the MAC from the user application has a size smaller than 60 bytes, the MAC automatically adds padding bytes (0x00) to comply with the Ethernet minimum frame length specification. Transmit padding is always performed and cannot be disabled.

If the MAC is not allowed to append a CRC (TxBD[TC] = 1), the user application is responsible for providing frames with a minimum length of 64 octets.

23.6.5.2 MAC address insertion

On each frame received from the core transmit FIFO interface, the source MAC address is either:

- Replaced by the address programmed in the PADDR1/2 fields (ENET*n*_TCR[ADDINS] = 1)
- Transparently forwarded to the Ethernet line (ENET*n*_TCR[ADDINS] = 0)

23.6.5.3 CRC-32 generation

The CRC-32 field is optionally generated and appended at the end of a frame.

The CRC polynomial, as specified in the 802.3 standard, is:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

The 32 bits of the CRC value are placed in the FCS field so that the x^{31} term is the right-most bit of the first octet. The CRC bits are thus transmitted in the following order: x^{31}, x^{30},..., x^{1}, x^{0}.

23.6.5.4 Inter-packet gap (IPG)

In full-duplex mode, after frame transmission and before transmission of a new frame, an IPG (programmed in ENET*n*_TIPG) is maintained. The minimum IPG can be programmed between 8 and 26 byte-times (64 and 208 bit-times).

In half-duplex mode, the core constantly monitors the line. Actual transmission of the data onto the network occurs only if it has been idle for a 96-bit time period, and any back-off time requirements have been satisfied. In accordance with the standard, the core begins to measure the IPG from CRS/GMII_CRS de-assertion.

23.6.5.5 Collision detection and handling — half-duplex operation only

A collision occurs on a half-duplex network when concurrent transmissions from two or more nodes take place. During transmission, the core monitors the line condition and detects a collision when the PHY device asserts COL.

When the core detects a collision while transmitting, it stops transmission of the data and transmits a 32-bit jam pattern. If the collision is detected during the preamble or the SFD transmission, the jam pattern is transmitted after completing the SFD, which results in a minimum 96-bit fragment. The jam pattern is a fixed pattern that is not compared to the actual frame CRC, and has a very low probability (0.532) of having a jam pattern identical to the CRC.

If a collision occurs before transmission of 64 bytes (including preamble and SFD), the MAC core waits for the backoff period and retransmits the packet data (stored in a 64-byte re-transmit buffer) that has already been sent on the line. The backoff period is generated from a pseudo-random process (truncated binary exponential backoff).

If a collision occurs after transmission of 64 bytes (including preamble and SFD), the MAC discards the remainder of the frame, optionally sets the LC interrupt bit, and sets TxBD[LCE].

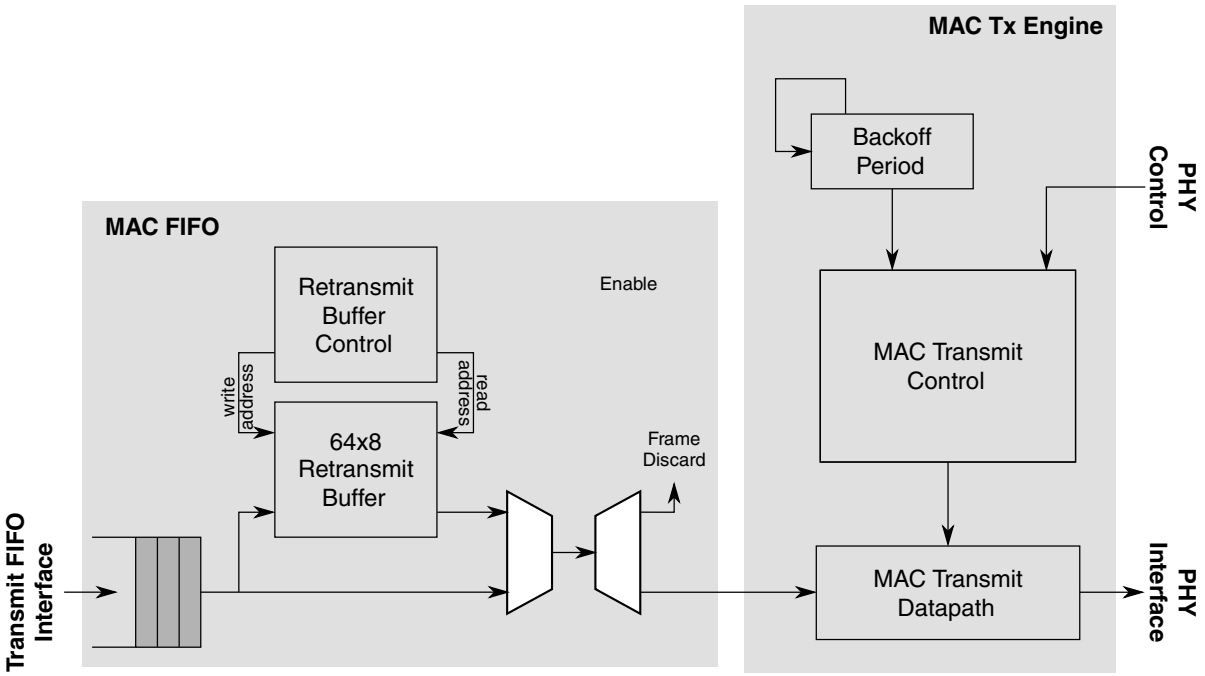


Figure 23-7. Packet re-transmit overview

The backoff time is represented by an integer multiple of slot times. One slot is equal to a 512-bit time period. The number of the delay slot times, before the n^{th} re-transmission attempt, is chosen as a uniformly-distributed random integer in the range:

- $0 < r < 2^k$
- $k = \min(n, N)$; where n is the number of retransmissions and $N = 10$

For example, after the first collision, the backoff period is 0 or 1 slot time. If a collision occurs on the first retransmission, the backoff period is 0, 1, 2, or 3, and so on.

The maximum backoff time (in 512-bit time slots) is limited by N = 10 as specified in the IEEE 802.3 standard.

If a collision occurs after 16 consecutive retransmissions, the core reports an excessive collision condition (ENET*n*_EIR[RL] interrupt field and TxBD[EE]) and discards the current packet from the FIFO.

In networks violating the standard requirements, a collision may occur after transmission of the first 64 bytes. In this case, the core stops the current packet transmission and discards the rest of the packet from the transmit FIFO. The core resumes transmission with the next packet available in the core transmit FIFO.

warning

Ethernet PHYs that support the SQE Test, or "heartbeat,"
feature must disable this feature. When this feature is enabled,

the PHY asserts the collision signal after a frame is transmitted to indicate to the ENET that the PHY's collision logic is working. This may cause data corruption in the next frame from the ENET. This corrupted frame contains up to 21 zero bytes which start somewhere within the MAC destination address field. The ENET, however, will still generate a good FCS (CRC-32) but with corrupted data.

23.6.6 Full-duplex flow control operation

Three conditions are handled by the core's flow control engine:

- Remote device congestion — The remote device connected to the same Ethernet segment as the core reports congestion and requests that the core stop sending data.
- Core FIFO congestion — When the core's receive FIFO reaches a user-programmable threshold (RX section empty), the core sends a pause frame back to the remote device requesting the data transfer to stop.
- Local device congestion — Any device connected to the core can request (typically, via the host processor) the remote device to stop transmitting data.

23.6.6.1 Remote device congestion

When the MAC transmit control gets a valid pause quanta from the receive path and if ENET*n*_RCR[FCE] is set, the MAC transmit logic:

- Completes the transfer of the current frame.
- Stops sending data for the amount of time specified by the pause quanta in 512 bit time increments.
- Sets ENET*n*_TCR[RFC_PAUSE].

Frame transfer resumes when the time specified by the quanta expires and if no new quanta value is received, or if a new pause frame with a quanta value set to 0x0000 is received. The MAC also resets RFC_PAUSE to zero.

If ENET*n*_RCR[FCE] cleared, the MAC ignores received pause frames.

Optionally and independent of ENET*n*_RCR[FCE], pause frames are forwarded to the client interface if PAUFWD is set.

23.6.6.2 Local device/FIFO congestion

The MAC transmit engine generates pause frames when the local receive FIFO is not able to receive more than a pre-defined number of words (FIFO programmable threshold) or when pause frame generation is requested by the local host processor.

- To generate a pause frame, the host processor sets ENET*n*_TCR[TFC_PAUSE]. A single pause frame is generated when the current frame transfer is completed and TFC_PAUSE is automatically cleared. Optionally, an interrupt is generated.
- An XOFF pause frame is generated when the receive FIFO asserts its section empty flag (internal). An XOFF pause frame is generated automatically, when the current frame transfer completes.
- An XON pause frame is generated when the receive FIFO deasserts its section empty flag (internal). An XON pause frame is generated automatically, when the current frame transfer completes.

When an XOFF pause frame is generated, the pause quanta (payload byte P1 and P2) is filled with the value programmed in ENET*n*_OPD[PAUSE_DUR].

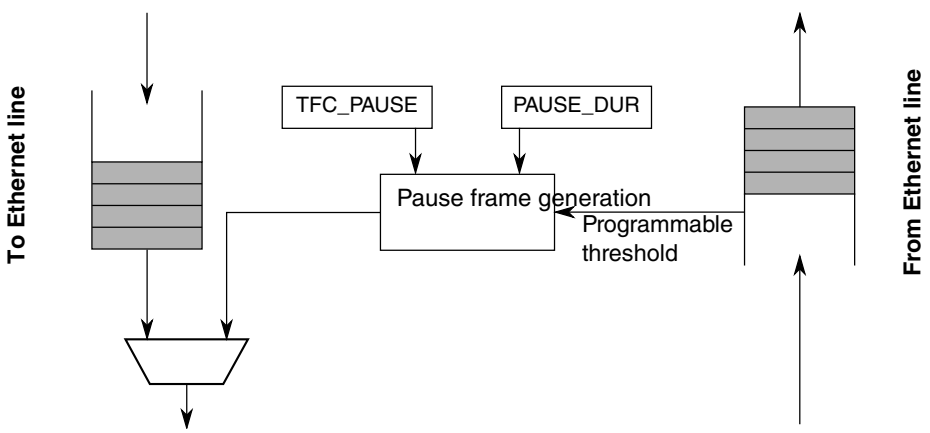


Figure 23-8. Pause frame generation overview

Note

Although the flow control mechanism should prevent any FIFO overflow on the MAC core receive path, the core receive FIFO is protected. When an overflow is detected on the receive FIFO, the current frame is truncated with an error indication set in the frame status word. The frame should subsequently be discarded by the user application.

23.6.7 Magic packet detection

Magic packet detection wakes a node that is put in power-down mode by the node management agent. Magic packet detection is supported only if the MAC is configured in sleep mode.

23.6.7.1 Sleep mode

To put the MAC in Sleep mode, set ENET*n*_ECR[SLEEP]. At the same time ENET*n*_ECR[MAGICEN] should also be set to enable magic packet detection.

In addition, if ENET is enabled, write 1 to ENET*n*_ECR[SLEEP] before entering into low power mode.

When the MAC is in Sleep mode:

- The transmit logic is disabled.
- The FIFO receive/transmit functions are disabled.
- The receive logic is kept in Normal mode, but it ignores all traffic from the line except magic packets. They are detected so that a remote agent can wake the node.

23.6.7.2 Magic packet detection

The core is designed to detect magic packets (see Magic packets) with the destination address set to:

- Any multicast address
- The broadcast address
- The unicast address programmed in PADDR1/2

When a magic packet is detected, EIR[WAKEUP] is set and none of the statistic registers are incremented.

23.6.7.3 Wakeup

When a magic packet is detected, indicated by ENET*n*_EIR[WAKEUP], ENET*n*_ECR[SLEEP] should be cleared to resume normal operation of the MAC. Clearing the SLEEP bit automatically masks ENET*n*_ECR[MAGICEN], disabling magic packet detection.

23.6.8 IP accelerator functions

The following sections describe the IP accelerator functions.

23.6.8.1 Checksum calculation

The IP and ICMP, TCP, UDP checksums are calculated with one's complement arithmetic summing up 16-bit values.

- For ICMP, the checksum is calculated over the complete ICMP datagram, in other words without IP header.
- For TCP and UDP, the checksums contain the header and data sections and values from the IP header, which can be seen as a pseudo-header that is not actually present in the data stream.

Table 23-26. IPv4 pseudo-header for checksum calculation

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source address							
Destination address							
Zero		Protocol		TCP/UDP length			

Table 23-27. IPv6 pseudo-header for checksum calculation

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source address							
Destination address							
TCP/UDP length							
Zero						Next header	

The TCP/UDP length value is the length of the TCP or UDP datagram, which is equal to the payload of an IP datagram. It is derived by subtracting the IP header length from the complete IP datagram length that is given in the IP header (IPv4), or directly taken from the IP header (IPv6). The protocol field is the corresponding value from the IP header. The Zero fields are all zeroes.

For IPv6, the complete 128-bit addresses are considered. The next header value identifies the upper layer protocol as either TCP or UDP. It may differ from the next header value of the IPv6 header if extension headers are inserted before the protocol header.

The checksum calculation uses 16-bit words in network byte order: The first byte sent/received is the MSB, and the second byte sent/received is the LSB of the 16-bit value to add to the checksum. If the frame ends on an odd number of bytes, a zero byte is appended for checksum calculation only, and is not actually transmitted.

23.6.8.2 Additional padding processing

According to IEEE 802.3, any Ethernet frame must have a minimum length of 64 octets.

The MAC usually removes padding on receive when a frame with length information is received. Because IP frames have a type value instead of length, the MAC does not remove padding for short IP frames, as it is not aware of the frame contents.

The IP accelerator function can be configured to remove the Ethernet padding bytes that might follow the IP datagram.

On transmit, the MAC automatically adds padding as necessary to fill any frame to a 64-byte length.

23.6.8.3 32-bit Ethernet payload alignment

The data FIFOs allow inserting two additional arbitrary bytes in front of a frame. This extends the 14-byte Ethernet header to a 16-byte header, which leads to alignment of the Ethernet payload, following the Ethernet header, on a 32-bit boundary.

This function can be enabled for transmit and receive independently with the corresponding SHIFT16 bits in the ENET*n*_TACC and ENET*n*_RACC registers.

When enabled, the valid frame data is arranged as shown in Table 23-28.

Table 23-28. 64-bit interface data structure with SHIFT16 enabled

63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0	

Table continues on the next page...

Table 23-28. 64-bit interface data structure with SHIFT16 enabled (continued)

Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0	Any value	Any value
Byte 13	Byte 12	Byte 11	Byte 10	Byte 9	Byte 8	Byte 7	Byte 6
...							

23.6.8.3.1 Receive processing

When ENET*n*_RACC[SHIFT16] is set, each frame is received with two additional bytes in front of the frame.

The user application must ignore these first two bytes and find the first byte of the frame in bits 23–16 of the first word from the RX FIFO.

Note

SHIFT16 must be set during initialization and kept set during the complete operation, because it influences the FIFO write behavior.

23.6.8.3.2 Transmit processing

When ENET*n*_TACC[SHIFT16] is set, the first two bytes of the first word written (bits 15–0) are discarded immediately by the FIFO write logic.

The SHIFT16 bit can be enabled/disabled for each frame individually if required, but can be changed only between frames.

23.6.8.4 Received frame discard

Because the receive FIFO must be operated in store and forward mode (ENET*n*_RSFL cleared), received frames can be discarded based on the following errors:

- The MAC function receives the frame with an error:
  - The frame has an invalid payload length
  - Frame length is greater than MAX_FL
  - Frame received with a CRC-32 error
  - Frame truncated due to receive FIFO overflow
  - Frame is corrupted as PHY signaled an error (RX_ERR asserted during reception)

- An IP frame is detected and the IP header checksum is wrong
- An IP frame with a valid IP header and a valid IP header checksum is detected, the protocol is known but the protocol-specific checksum is wrong

If one of the errors occurs and the IP accelerator function is configured to discard frames (ENET*n*_RACC), the frame is automatically discarded. Statistics are maintained normally and are not affected by this discard function.

23.6.8.5 IPv4 fragments

When an IPv4 IP fragment frame is received, only the IP header is inspected and its checksum verified. 32-bit alignment operates the same way on fragments as it does on normal IP frames, as specified above.

The IP fragment frame payload is not inspected for any protocol headers. As such, a protocol header would only exist in the very first fragment. To assist in protocol-specific checksum verification, the one's-complement sum is calculated on the IP payload (all bytes following the IP header) and provided with the frame status word.

The frame fragment status field (RxBD[FRAG]) is set to indicate a fragment reception, and the one's-complement sum of the IP payload is available in RxBD[Payload checksum].

Note

After all fragments have been received and reassembled, the application software can take advantage of the payload checksum delivered with the frame's status word to calculate the protocol-specific checksum of the datagram.

For example, if a TCP payload is delivered by multiple IP fragments, the application software can calculate the pseudo-header checksum value from the first fragment, and add the payload checksums delivered with the status for all fragments to verify the TCP datagram checksum.

23.6.8.6 IPv6 support

The following sections describe the IPv6 support.

23.6.8.6.1 Receive processing

An Ethernet frame of type 0x86DD identifies an IP Version 6 frame (IPv6) frame. If an IPv6 frame is received, the first IP header is inspected (first ten words), which is available in every IPv6 frame.

If the receive SHIFT16 function is enabled, the IP header is aligned on a 32-bit boundary allowing more efficient processing (see 32-bit Ethernet payload alignment).

For TCP and UDP datagrams, the pseudo-header checksum calculation is performed and verified.

To assist in protocol-specific checksum verification, the one's-complement sum is always calculated on the IP payload (all bytes following the IP header) and provided with the frame status word. For example, if extension headers were present, their sums can be subtracted in software from the checksum to isolate the TCP/UDP datagram checksum, if required.

23.6.8.6.2 Transmit processing

For IPv6 transmission, the SHIFT16 function is supported to process 32-bit aligned datagrams.

IPv6 has no IP header checksum; therefore, the IP checksum insertion configuration is ignored.

The protocol checksum is inserted only if the next header of the IP header is a known protocol (TCP, UDP, or ICMP). If a known protocol is detected, the checksum over all bytes following the IP header is calculated and inserted in the correct position.

The pseudo-header checksum calculation is performed for TCP and UDP datagrams accordingly.

23.6.9 Resets and stop controls

The following sections describe the resets and stop controls.

23.6.9.1 Hardware reset

To reset the Ethernet module, set ENET*n*_ECR[RESET].

23.6.9.2 Soft reset

When ENET*n*_ECR[ETHER_EN] is cleared during operation, the following occurs:

- uDMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers.
- A currently ongoing transmit is terminated by asserting GMII/TXER to the PHY.
- A currently ongoing transmit FIFO write from the application is terminated by stopping the write to the FIFO, and all further data from the application is ignored. All subsequent writes are ignored until re-enabled.
- A currently ongoing receive FIFO read is terminated. The RxBD has arbitrary values in this case.

23.6.9.3 Hardware freeze

When the processor enters debug mode and ECR[DBGEN] is set, the MAC enters a freeze state where it stops all transmit and receive activities gracefully.

The following happens when the MAC enters hardware freeze:

- A currently ongoing receive transaction on the receive application interface is completed as normal. No further frames are read from the FIFO.
- A currently ongoing transmit transaction on the transmit application interface is completed as normal (in other words, until writing end-of-packet (EOP)).
- A currently ongoing frame receive is completed normally, after which no further frames are accepted from the MII/GMII.
- A currently ongoing frame transmit is completed normally, after which no further frames are transmitted.

23.6.9.4 Graceful stop

During a graceful stop, any currently ongoing transactions are completed normally and no further frames are accepted. The MAC can resume from a graceful stop without the need for a reset (for example, clearing ETHER_EN is not required).

The following conditions lead to a graceful stop of the MAC transmit or receive datapaths.

23.6.9.4.1 Graceful transmit stop (GTS)

When gracefully stopped, the MAC is no longer reading frame data from the transmit FIFO and has completed any ongoing transmission.

In any of the following conditions, the transmit datapath stops after an ongoing frame transmission has been completed normally.

- ENET*n*_TCR[GTS] is set by software.
- ENET*n*_TCR[TFC_PAUSE] is set by software requesting a pause frame transmission. The status (and register bit) is cleared after the pause frame has been sent.
- A pause frame was received stopping the transmitter. The stopped situation is terminated when the pause timer expires or a pause frame with zero quanta is received.
- MAC is placed in Sleep mode by software or the processor entering Stop mode (see Sleep mode).
- The MAC is in Hardware Freeze mode.

When the transmitter has reached its stopped state, the following events occur:

- The GRA interrupt is asserted, when transitioned into stopped.
- In Hardware Freeze mode, the GRA interrupt does not wait for the application write completion and asserts when the transmit state machine (in other words, line side of TX FIFO) reaches its stopped state.

23.6.9.4.2 Graceful receive stop (GRS)

When gracefully stopped, the MAC is no longer writing frames into the receive FIFO.

The receive datapath stops after any ongoing frame reception has been completed normally, if any of the following conditions occur:

- MAC is placed in Sleep mode either by the software or the processor is in Stop mode). The MAC continues to receive frames and search for magic packets if enabled (see Magic packet detection). However, no frames are written into the receive FIFO, and therefore are not forwarded to the application.
- The MAC is in Hardware Freeze mode. The MAC does not accept any frames from the MII/GMII.

When the receive datapath is stopped, the following events occur:

- If the RX is in the stopped state, RCR[GRS] is set
- The GRA interrupt is asserted when the transmitter and receiver are stopped
- Any ongoing receive transaction to the application (RX FIFO read) continues normally until the frame end of package (EOP) is reached. After this, the following occurs:
  - When Sleep mode is active, all further frames are discarded, flushing the RX FIFO
  - In Hardware Freeze mode, no further frames are delivered to the application and they stay in the receive FIFO.

Note

The assertion of GRS does not wait for an ongoing FIFO read transaction on the application side of the FIFO (FIFO read).

23.6.9.4.3 Graceful stop interrupt (GRA)

The graceful stopped interrupt (GRA) is asserted for the following conditions:

- In Sleep mode, the interrupt asserts only after both TX and RX datapaths are stopped.
- In Hardware Freeze mode, the interrupt asserts only after both TX and RX datapaths are stopped.
- The MAC transmit datapath is stopped for any other condition (GTS, TFC_PAUSE, pause received).

The GRA interrupt is triggered only once when the stopped state is entered. If the interrupt is cleared while the stop condition persists, no further interrupt is triggered.

23.6.10 IEEE 1588 functions

To allow for IEEE 1588 or similar time synchronization protocol implementations, the MAC is combined with a time-stamping module to support precise time-stamping of incoming and outgoing frames. Set ENET*n*_ECR[EN1588] to enable 1588 support.

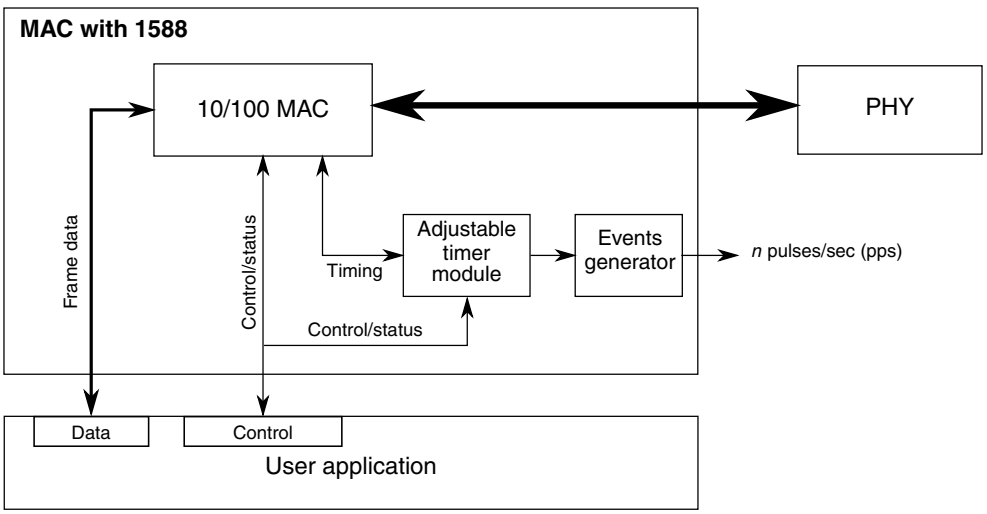


Figure 23-9. IEEE 1588 functions overview

23.6.10.1 Adjustable timer module

The adjustable timer module (TSM) implements the free-running counter (FRC), which generates the timestamps. The FRC operates with the time-stamping clock, which can be set to any value depending on your system requirements.

Through dedicated correction logic, the timer can be adjusted to allow synchronization to a remote master and provide a synchronized timing reference to the local system. The timer can be configured to cause an interrupt after a fixed time period, to allow synchronization of software timers or perform other synchronized system functions.

The timer is typically used to implement a period of one second; hence, its value ranges from 0 to (1×10^9)-1. The period event can trigger an interrupt, and software can maintain the seconds and hours time values as necessary.

23.6.10.1.1 Adjustable timer implementation

The adjustable timer consists of a programmable counter/accumulator and a correction counter. The periods of both counters and their increment rates are freely configurable, allowing very fine tuning of the timer.

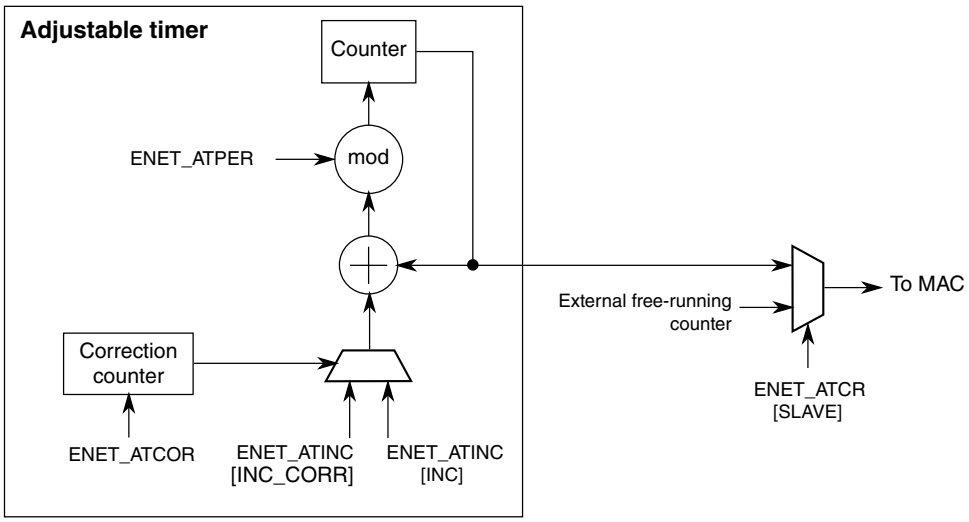


Figure 23-10. Adjustable timer implementation detail

The counter produces the current time. During each time-stamping clock cycle, a constant value is added to the current time as programmed in ENET*n*_ATINC. The value depends on the chosen time-stamping clock frequency. For example, if it operates at 125 MHz, setting the increment to eight represents 8 ns.

The period, configured in ENET*n*_ATPER, defines the modulo when the counter wraps. In a typical implementation, the period is set to 1×10^9 so that the counter wraps every second, and hence all timestamps represent the absolute nanoseconds within the one second period. When the period is reached, the counter wraps to start again respecting the period modulo. This means it does not necessarily start from zero, but instead the counter is loaded with the value (Current + Inc $-(1 \times 10^9)$), assuming the period is set to 1×10^9.

The correction counter operates fully independently, and increments by one with each time-stamping clock cycle. When it reaches the value configured in ENET*n*_ATCOR, it restarts and instructs the timer once to increment by the correction value, instead of the normal value.

The normal and correction increments are configured in ENET*n*_ATINC. To speed up the timer, set the correction increment more than the normal increment value. To slow down the timer, set the correction increment less than the normal increment value.

The correction counter only defines the distance of the corrective actions, not the amount. This allows very fine corrections and low jitter (in the range of 1 ns) independent of the chosen clock frequency.

By enabling slave mode (ENET*n*_ATCR[SLAVE] = 1), the timer is ignored and the current time is externally provided from one of the external modules. See the Chip Configuration details for which clock source is used. This is useful if multiple modules

within the system must operate from a single timer. When slave mode is enabled, you still must set ENET*n*_ATINC[INC] to the value of the master, since it is used for internal comparisons.

23.6.10.2 Transmit timestamping

Only 1588 event frames need to be time-stamped on transmit. The client application (for example, the MAC driver) should detect 1588 event frames and set TxBD[TS] together with the frame.

If TxBD[TS] is set, the MAC records the timestamp for the frame in ENET*n*_ATSTMP. ENET*n*_EIR[TS_AVAIL] is set to indicate that a new timestamp is available.

Software implements a handshaking procedure by setting TxBD[TS] when it transmits the frame for which a timestamp is needed, and then waits for ENET*n*_EIR[TS_AVAIL] to determine when the timestamp is available. The timestamp is then read from ENET*n*_ATSTMP. This is done for all event frames. Other frames do not use TxBD[TS] and, therefore, do not interfere with the timestamp capture.

23.6.10.3 Receive timestamping

When a frame is received, the MAC latches the value of the timer when the frame's start of frame delimiter (SFD) field is detected, and provides the captured timestamp on RxBD[1588 timestamp]. This is done for all received frames.

23.6.10.4 Time synchronization

The adjustable timer module is available to synchronize the local clock of a node to a remote master. It implements a free running 32-bit counter, and also contains an additional correction counter.

The correction counter increases or decreases the rate of the free running counter, enabling very fine granular changes of the timer for synchronization, yet adding only very low jitter when performing corrections.

The application software implements, in a slave scenario, the required control algorithm, setting the correction to compensate for local oscillator drifts and locking the timer to the remote master clock on the network.

The timer and all timestamp-related information should be configured to show the true nanoseconds value of a second (in other words, the timer is configured to have a period of one second). Hence, the values range from 0 to (1×10^9)–1. In this application, the seconds counter is implemented in software using an interrupt function that is executed when the nanoseconds counter wraps at 1×10^9.

23.6.10.5 Input Capture and Output Compare

The Input Capture Output Compare block can be used to provide precise hardware timing for input and output events.

23.6.10.5.1 Input capture

The TCCR*n* capture registers latch the time value when the corresponding external event occurs. An event can be a rising-, falling-, or either-edge of one of the 1588_TMR*n* signals. An event will cause the corresponding TCSR*n*[TF] timer flag to be set, indicating that an input capture has occurred. If the corresponding interrupt is enabled with the TCSR*n*[TIE] field, an interrupt can be generated.

23.6.10.5.2 Output compare

The TCCR*n* compare registers are loaded with the time at which the corresponding event should occur. When the ENET free-running counter value matches the output compare reference value in the TCCR*n* register, the corresponding flag, TCSR*n*[TF], is set, indicating that an output compare has occurred. The corresponding interrupt, if enabled by TCSR*n*[TIE], will be generated.The corresponding 1588_TMR*n* output signal will be asserted according to TCSR*n*[TMODE].

NOTE

If TSCR*n*[TMODE] is set to 10X1b or 1010b then the timer output pin toggle-on-overflow will occur only when PINPER, PEREN, and EN bits of the ATCR register are one.

23.6.10.5.3 DMA requests

A DMA request can be enabled by setting TCSR*n*[TDRE]. The corresponding DMA request is generated when the TCSR*n*[TF] timer flag is set. When the DMA has completed, the corresponding TCSRn[TF] flag is cleared.

23.6.11 FIFO thresholds

The core FIFO thresholds are fully programmable to dynamically change the FIFO operation.

For example, store and forward transfer can be enabled by a simple change in the FIFO threshold registers.

The thresholds are defined in 64-bit words.

The receive and transmit FIFOs both have a depth of 512 words.

23.6.11.1 Receive FIFO

Four programmable thresholds are available, which can be set to any value to control the core operation as follows.

Table 23-29. Receive FIFO thresholds definition

Register	Description
ENET*n*_RSFL [RX_SECTION_FULL]	When the FIFO level reaches the ENET*n*_RSFL value, the MAC status signal is asserted to indicate that data is available in the receive FIFO (cut-through operation). Once asserted, if the FIFO empties below the threshold set with ENET*n*_RAEM and if the end-of-frame is not yet stored in the FIFO, the status signal is deasserted again. If a frame has a size smaller than the threshold (in other words, an end-of-frame is available for the frame), the status is also asserted. To enable store and forward on the receive path, clear ENET*n*_RSFL. The MAC status signal is asserted only when a complete frame is stored in the receive FIFO. When programming a non-zero value to ENET*n*_RSFL (cut-through operation) it should be greater than ENET*n*_RAEM.
ENET*n*_RAEM [RX_ALMOST_EMPTY]	When the FIFO level reaches the ENET*n*_RAEM value, and the end-of-frame has not been received, the core receive read control stops the FIFO read (and subsequently stops transferring data to the MAC client application). It continues to deliver the frame, if again more data than the threshold or the end-of-frame is available in the FIFO. Set ENET*n*_RAEM to a minimum of six.
ENET*n*_RAFL [RX_ALMOST_FULL]	When the FIFO level approaches the maximum and there is no more space remaining for at least ENETn_RAFL number of words, the MAC control logic stops writing data in the FIFO and truncates the receive frame to avoid FIFO overflow. The corresponding error status is set when the frame is delivered to the application. Set ENET*n*_RAFL to a minimum of 4.
ENET*n*_RSEM [RX_SECTION_EMPTY]	When the FIFO level reaches the ENET*n*_RSEM value, an indication is sent to the MAC transmit logic, which generates an XOFF pause frame. This indicates FIFO congestion to the remote Ethernet client. When the FIFO level goes below the value programmed in ENET*n*_RSEM, an indication is sent to the MAC transmit logic, which generates an XON pause frame. This indicates the FIFO congestion is cleared to the remote Ethernet client. Clearing ENET*n*_RSEM disables any pause frame generation.

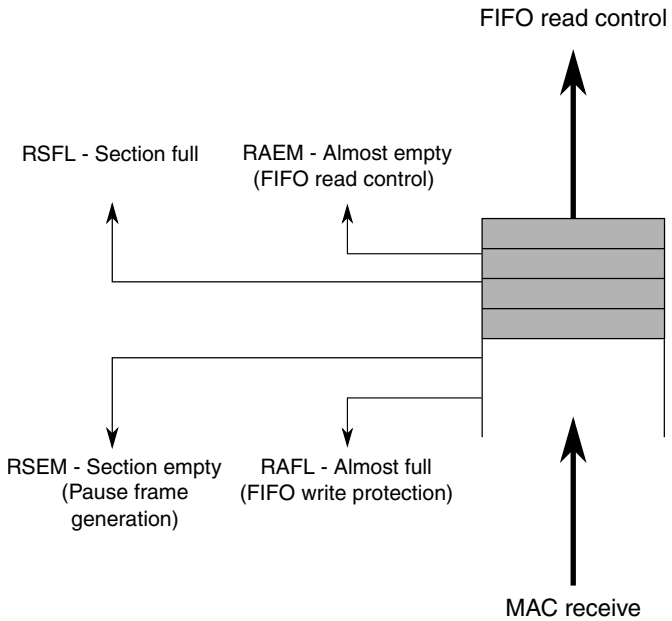


Figure 23-11. Receive FIFO overview

23.6.11.2 Transmit FIFO

Four programmable thresholds are available which control the core operation as described below.

Table 23-30. Transmit FIFO thresholds definition

Register	Description
ENET*n*_TAEM [TX_ALMOST _EMPTY]	When the FIFO level reaches the ENET*n*_TAEM value and no end-of-frame is available for the frame, the MAC transmit logic avoids a FIFO underflow by stopping FIFO reads and transmitting the Ethernet frame with an MII error indication. Set ENET*n*_TAEM to a minimum of 4.
ENET*n*_TAFL [TX_ALMOST _FULL]	When the FIFO level approaches the maximum, so that there is no more space for at least ENET*n*_TAFL number of words, the MAC deasserts its control signal to the application. If the application does not react on this signal, the FIFO write control logic avoids FIFO overflow by truncating the current frame and setting the error status. As a result, the frame is transmitted with an GMII/MII error indication. Set ENET*n*_TAFL to a minimum of 4. Larger values allow more latency for the application to react on the MAC control signal deassertion, before the frame is truncated. A typical setting is 8, which offers 3–4 clock cycles of latency to the application to react on the MAC control signal deassertion.
ENET*n*_TSEM [TX_SECTION _EMPTY]	When the FIFO level reaches the ENET*n*_TSEM value, a MAC status signal is deasserted to indicate that the transmit FIFO is getting full. This gives the ENET module an indication to slow or stop its write transaction to avoid a buffer overflow. This is a pure indication function to the application. It has no effect within the MAC. When ENET*n*_TSEM is 0, the signal is never deasserted.
ENET*n*_TFWR	When the FIFO level reaches the ENET*n*_TFWR value and when STRFWD is cleared, the MAC transmit control logic starts frame transmission before the end-of-frame is available in the FIFO (cut-through operation).

Table 23-30. Transmit FIFO thresholds definition

Register	Description
	If a complete frame has a size smaller than the ENET*n*_TFWR threshold, the MAC also transmits the frame to the line. To enable store and forward on the transmit path, set STRFWD. In this case, the MAC starts to transmit data only when a complete frame is stored in the transmit FIFO.

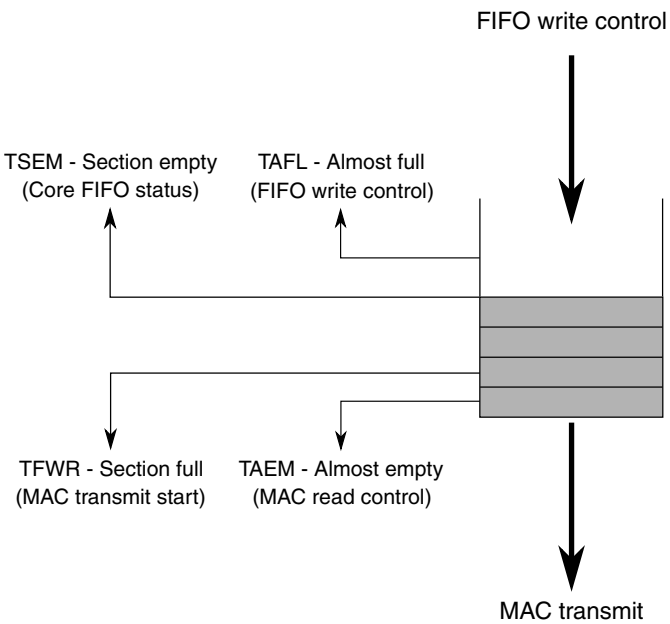


Figure 23-12. Transmit FIFO overview

23.6.12 Loopback options

The core implements external and internal loopback options, which are controlled by the ENET*n*_RCR register fields found here.

The core implements external and internal loopback options, which are controlled by the following ENET*n*_RCR register fields:

Table 23-31. Loopback options

Register field	Description
LOOP	Internal MII loopback. The MAC transmit is returned to the MAC receive. No data is transmitted to the external interfaces. In MII internal loopback, MII_TXCLK and MII_RXCLK must be provided with a clock signal (2.5 MHz for 10 Mbit/s, and 25 MHz for 100 Mbit/s))

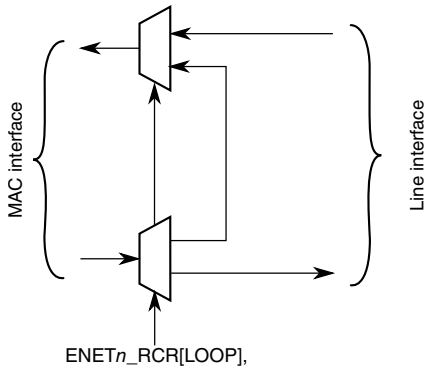


Figure 23-13. Loopback options

23.6.13 Legacy buffer descriptors

To support the Ethernet controller on previous chips, legacy FEC buffer descriptors are available. To enable legacy support, write 0 to ENET*n*_ECR[1588EN].

NOTE

- The legacy buffer descriptor tables show the byte order for little-endian chips. DBSWP must be set to 1 after reset to enable little-endian mode.

23.6.13.1 Legacy receive buffer descriptor

The following table shows the legacy FEC receive buffer descriptor. Table 23-35 contains the descriptions for each field.

Table 23-32. Legacy FEC receive buffer descriptor (RxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data length															
Offset + 2	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 4	Rx data buffer pointer — low halfword															
Offset + 6	Rx data buffer pointer — high halfword															

23.6.13.2 Legacy transmit buffer descriptor

The following table shows the legacy FEC transmit buffer descriptor. Table 23-37 contains the descriptions for each field.

Table 23-33. Legacy FEC transmit buffer descriptor (TxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data Length															
Offset + 2	R	TO1	W	TO2	L	TC	ABC[1]	—	—	—	—	—	—	—	—	—
Offset + 4	Tx Data Buffer Pointer — low halfword															
Offset + 6	Tx Data Buffer Pointer — high halfword															

1. This field is not supported by the uDMA.

23.6.14 Enhanced buffer descriptors

This section provides a description of the enhanced operation of the driver/uDMA via the buffer descriptors.

It is followed by a detailed description of the receive and transmit descriptor fields. To enable the enhanced features, set ENET*n*_ECR[1588EN].

NOTE

The enhanced buffer descriptor tables show the byte order for little-endian chips. DBSWP must be set to 1 after reset to enable little-endian mode.

23.6.14.1 Enhanced receive buffer descriptor

The following table shows the enhanced uDMA receive buffer descriptor. Table 23-35 contains the descriptions for each field.

Table 23-34. Enhanced uDMA receive buffer descriptor (RxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data length															
Offset + 2	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 4	Rx data buffer pointer – low halfword															
Offset + 6	Rx data buffer pointer – high halfword															

Table continues on the next page...

Table 23-34. Enhanced uDMA receive buffer descriptor (RxBD) (continued)

Offset + 8	—	—	—	—	—	—	—	—	—	—	ICE	PCR	—	VLAN	IPV6	FRAG
Offset + A	ME	—	—	—	—	PE	CE	UC	INT	—	—	—	—	—	—	—
Offset + C	Payload checksum															
Offset + E	Header length					—	—	—	Protocol type							
Offset + 10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp – low halfword															
Offset + 16	1588 timestamp – high halfword															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 23-35. Receive buffer descriptor field definitions

Word	Field	Description
Offset + 0	15–0 Data Length	Data length. Written by the MAC. Data length is the number of octets written by the MAC into this BD's data buffer if L is cleared (the value is equal to EMRBR), or the length of the frame including CRC if L is set. It is written by the MAC once as the BD is closed.
Offset + 2	15 E	Empty. Written by the MAC (= 0) and user (= 1). 0 The data buffer associated with this BD is filled with received data, or data reception has aborted due to an error condition. The status and length fields have been updated as required. 1 The data buffer associated with this BD is empty, or reception is currently in progress.
Offset + 2	14 RO1	Receive software ownership. This field is reserved for use by software. This read/write field is not modified by hardware, nor does its value affect hardware.
Offset + 2	13 W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location. 1 The next buffer descriptor is found at the location defined in ENET*n*_RDSR.
Offset + 2	12 RO2	Receive software ownership. This field is reserved for use by software. This read/write field is not modified by hardware, nor does its value affect hardware.
Offset + 2	11 L	Last in frame. Written by the uDMA. 0 The buffer is not the last in a frame. 1 The buffer is the last in a frame.
Offset + 2	10–9	Reserved, must be cleared.
Offset + 2	8 M	Miss. Written by the MAC. This field is set by the MAC for frames accepted in promiscuous mode, but flagged as a miss by the internal address recognition. Therefore, while in promiscuous mode, you can use the this field to quickly determine whether the frame was destined to this station. This field is valid only if the L and PROM bits are set. 0 The frame was received because of an address recognition hit. 1 The frame was received because of promiscuous mode.

Table continues on the next page...

Table 23-35. Receive buffer descriptor field definitions (continued)

Word	Field	Description
		The information needed for this field comes from the promiscuous_miss(ff_rx_err_stat[26]) sideband signal.
Offset + 2	7 BC	Set if the DA is broadcast (FFFF_FFFF_FFFF).
Offset + 2	6 MC	Set if the DA is multicast and not BC.
Offset + 2	5 LG	Receive frame length violation. Written by the MAC. A frame length greater than RCR[MAX_FL] was recognized. This field is valid only if the L field is set. The receive data is not altered in any way unless the length exceeds TRUNC_FL bytes.
Offset + 2	4 NO	Receive non-octet aligned frame. Written by the MAC. A frame that contained a number of bits not divisible by 8 was received, and the CRC check that occurred at the preceding byte boundary generated an error or a PHY error occurred. This field is valid only if the L field is set. If this field is set, the CR field is not set.
Offset + 2	3	Reserved, must be cleared.
Offset + 2	2 CR	Receive CRC or frame error. Written by the MAC. This frame contains a PHY or CRC error and is an integral number of octets in length. This field is valid only if the L field is set.
Offset + 2	1 OV	Overrun. Written by the MAC. A receive FIFO overrun occurred during frame reception. If this field is set, the other status fields, M, LG, NO, and CR, lose their normal meaning and are zero. This field is valid only if the L field is set.
Offset + 2	0 TR	Set if the receive frame is truncated (frame length >TRUNC_FL). If the TR field is set, the frame must be discarded and the other error fields must be ignored because they may be incorrect.
Offset + 4	15–0 Data buffer pointer low	Receive data buffer pointer, low halfword
0ffset + 6	15–0 Data buffer pointer high	Receive data buffer pointer, high halfword The receive buffer pointer, containing the address of the associated data buffer, must always be evenly divisible by 16. The buffer must reside in memory external to the MAC. The Ethernet controller never modifies this value.
Offset + 8	15–6	Reserved, must be cleared.
Offset + 8	5 ICE	IP header checksum error. This is an accelerator option. This field is written by the uDMA. Set when either a non-IP frame is received or the IP header checksum was invalid. An IP frame with less than 3 bytes of payload is considered to be an invalid IP frame. This field is only valid if the L field is set.
Offset + 8	4 PCR	Protocol checksum error. This is an accelerator option. This field is written by the uDMA. Set when the checksum of the protocol is invalid or an unknown protocol is found and checksumming could not be performed. This field is only valid if the L field is set.
Offset + 8	3	Reserved, must be cleared.
Offset + 8	2 VLAN	VLAN. This is an accelerator option. This field is written by the uDMA. It means that the frame has a VLAN tag. This field is valid only if the L field is set.
Offset + 8	1 IPV6	IPV6 Frame. This field is written by the uDMA. This field indicates that the frame has an IPv6 frame type. If this field is not set it means that an IPv4 or other protocol frame was received. This field is valid only if the L field is set.
Offset + 8	0 FRAG	IPv4 Fragment.This is an accelerator option.This field is written by the uDMA. It indicates that the frame is an IPv4 fragment frame. This field is only valid when the L field is set.

Table continues on the next page...

Table 23-35. Receive buffer descriptor field definitions (continued)

Word	Field	Description
Offset + A	15 ME	MAC error. This field is written by the uDMA. This field means that the frame stored in the system memory was received with an error (typically, a receive FIFO overflow). This field is only valid when the L field is set.
Offset + A	14–11	Reserved, must be cleared.
Offset + A	10 PE	PHY Error. This field is written by the uDMA. Set to "1"when the frame was received with an Error character on the PHY interface. The frame is invalid. This field is valid only when the L field is set.
Offset + A	9 CE	Collision. This field is written by the uDMA. Set when the frame was received with a collision detected during reception. The frame is invalid and sent to the user application. This field is valid only when the L field is set.
Offset + A	8 UC	Unicast. This field is written by the uDMA, and means that the frame is unicast. This field is valid regardless of whether the L field is set.
Offset + A	7 INT	Generate RXB/RXF interrupt. This field is set by the user to indicate that the uDMA is to generate an interrupt on the *dma_int_rxb* / *dma_int_rxfevent*.
Offset + A	6–0	Reserved, must be cleared.
Offset + C	15–0 Payload checksum	Internet payload checksum. This is an accelerator option. It is the one's complement sum of the payload section of the IP frame. The sum is calculated over all data following the IP header until the end of the IP payload. This field is valid only when the L field is set.
Offset + E	15–11 Header length	Header length. This is an accelerator option. This field is written by the uDMA. This field is the sum of 32-bit words found within the IP and its following protocol headers. If an IP datagram with an unknown protocol is found, then the value is the length of the IP header. If no IP frame or an erroneous IP header is found, the value is 0. The following values are minimum values if no header options exist in the respective headers: • ICMP/IP: 6 (5 IP header, 1 ICMP header) • UDP/IP: 7 (5 IP header, 2 UDP header) • TCP/IP: 10 (5 IP header, 5 TCP header) This field is only valid if the L field is set.
Offset + E	10–8	Reserved, must be cleared.
Offset + E	7–0 Protocol type	Protocol type. This is an accelerator option. The 8-bit protocol field found within the IP header of the frame. It is valid only when ICE is cleared. This field is valid only when the L field is set.
Offset + 10	15–0	Reserved, must be cleared.
Offset + 12	15 BDU	Last buffer descriptor update done. Indicates that the last BD data has been updated by uDMA. This field is written by the user (=0) and uDMA (=1).
Offset + 12	14–0	Reserved, must be cleared.
Offset + 14	15–0 1588 timestamp	This value is written by the uDMA. It is only valid if the L field is set.
Offset + 16		
Offset + 18 – Offset + 1E	15–0	Reserved, must be cleared.

23.6.14.2 Enhanced transmit buffer descriptor

Table 23-36. Enhanced transmit buffer descriptor (TxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data length															
Offset + 2	R	TO1	W	TO2	L	TC	—	—	—	—	—	—	—	—	—	—
Offset + 4	Tx Data Buffer Pointer – low halfword															
Offset + 6	Tx Data Buffer Pointer – high halfword															
Offset + 8	TXE	—	UE	EE	FE	LCE	OE	TSE	—	—	—	—	—	—	—	—
Offset + A	—	INT	TS	PINS	IINS	—	—	—	—	—	—	—	—	—	—	—
Offset + C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp – low halfword															
Offset + 16	1588 timestamp – high halfword															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 23-37. Enhanced transmit buffer descriptor field definitions

Word	Field	Description
Offset + 0	15–0 Data Length	Data length, written by user. Data length is the number of octets the MAC should transmit from this BD's data buffer. It is never modified by the MAC.
Offset + 2	15 R	Ready. Written by the MAC and you. 0 The data buffer associated with this BD is not ready for transmission. You are free to manipulate this BD or its associated data buffer. The MAC clears this field after the buffer has been transmitted or after an error condition is encountered. 1 The data buffer, prepared for transmission by you, has not been transmitted or currently transmits. You may write no fields of this BD after this field is set.
Offset + 2	14 TO1	Transmit software ownership. This field is reserved for software use. This read/write field is not modified by hardware and its value does not affect hardware.
Offset + 2	13 W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location 1 The next buffer descriptor is found at the location defined in ETDSR.

Table continues on the next page...

Table 23-37. Enhanced transmit buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 2	12 TO2	Transmit software ownership. This field is reserved for use by software. This read/write field is not modified by hardware and its value does not affect hardware.
Offset + 2	11 L	Last in frame. Written by user. 0 The buffer is not the last in the transmit frame 1 The buffer is the last in the transmit frame
Offset + 2	10 TC	Transmit CRC. Written by user, and valid only when L is set. 0 End transmission immediately after the last data byte 1 Transmit the CRC sequence after the last data byte This field is valid only when the L field is set.
Offset + 2	9 ABC	Append bad CRC. **Note:** This field is not supported by the uDMA and is ignored.
Offset + 2	8–0	Reserved, must be cleared.
Offset + 4	15–0 Data buffer pointer low	Tx data buffer pointer, low halfword
Offset + 6	15–0 Data buffer pointer high	Tx data buffer pointer, high halfword. The buffer must reside in memory external to the MAC. This value is never modified by the Ethernet controller.
Offset + 8	15 TXE	Transmit error occurred. This field is written by the uDMA. This field indicates that there was a transmit error of some sort reported with the frame. Effectively this field is an OR of the other error fields including UE, EE, FE, LCE, OE, and TSE. This field is valid only when the L field is set.
Offset + 8	14	Reserved, must be cleared.
Offset + 8	13 UE	Underflow error. This field is written by the uDMA. This field indicates that the MAC reported an underflow error on transmit. This field is valid only when the L field is set.
Offset + 8	12 EE	Excess Collision error. This field is written by the uDMA. This field indicates that the MAC reported an excess collision error on transmit. This field is valid only when the L field is set.
Offset + 8	11 FE	Frame with error. This field is written by the uDMA. This field indicates that the MAC reported that the uDMA reported an error when providing the packet. This field is valid only when the L field is set.
Offset + 8	10 LCE	Late collision error. This field is written by the uDMA. This field indicates that the MAC reported that there was a Late Collision on transmit. This field is valid only when the L field is set.
Offset + 8	9 OE	Overflow error. This field is written by the uDMA. This field indicates that the MAC reported that there was a FIFO overflow condition on transmit. This field is only valid when the L field is set.
Offset + 8	8 TSE	Timestamp error. This field is written by the uDMA. This field indicates that the MAC reported a different frame type then a timestamp frame. This field is valid only when the L field is set.
Offset + 8	7–0	Reserved, must be cleared.
Offset + A	15	Reserved, must be cleared.

Table continues on the next page...

Table 23-37. Enhanced transmit buffer descriptor field definitions (continued)

Word	Field	Description
Offset + A	14 INT	Generate interrupt flags. This field is written by the user. This field is valid regardless of the L field and must be the same for all EBD for a given frame. The uDMA does not update this value.
Offset + A	13 TS	Timestamp. This field is written by the user. This indicates that the uDMA is to generate a timestamp frame to the MAC. This field is valid regardless of the L field and must be the same for all EBD for the given frame. The uDMA does not update this value.
Offset + A	12 PINS	Insert protocol specific checksum. This field is written by the user. If set, the MAC's IP accelerator calculates the protocol checksum and overwrites the corresponding checksum field with the calculated value. The checksum field must be cleared by the application generating the frame. The uDMA does not update this value. This field is valid regardless of the L field and must be the same for all EBD for a given frame.
Offset + A	11 IINS	Insert IP header checksum. This field is written by the user. If set, the MAC's IP accelerator calculates the IP header checksum and overwrites the corresponding header field with the calculated value. The checksum field must be cleared by the application generating the frame. The uDMA does not update this value. This field is valid regardless of the L field and must be the same for all EBD for a given frame.
Offset + A	10–0	Reserved, must be cleared.
Offset + C	15–0	Reserved, must be cleared.
Offset + E	15–0	Reserved, must be cleared.
Offset + 10	15–0	Reserved, must be cleared.
Offset + 12	15 BDU	Last buffer descriptor update done. Indicates that the last BD data has been updated by uDMA. This field is written by the user (=0) and uDMA (=1).
Offset + 12	14–0	Reserved, must be cleared.
Offset + 14	15–0 1588 timestamp	This value is written by the uDMA . It is valid only when the L field is set.
Offset + 16		
Offset + 18–Offset + 1E	15–0	Reserved, must be cleared.

23.6.15 Client FIFO application interface

The FIFO interface is completely asynchronous from the Ethernet line, and the transmit and receive interface can operate at a different clock rate.

All transfers to/from the user application are handled independently of the core operation, and the core provides a simple interface to user applications based on a two-signal handshake.

23.6.15.1 Data structure description

The data structure defined in the following tables for the FIFO interface must be respected to ensure proper data transmission on the Ethernet line. Byte 0 is sent to and received from the line first.

Table 23-38. FIFO interface data structure

	63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
Word 0	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Word 1	Byte 15	Byte 14	Byte 13	Byte 12	Byte 11	Byte 10	Byte 9	Byte 8
...	...							

The size of a frame on the FIFO interface may not be a modulo of 64-bit.

The user application may not care about the Ethernet frame formats in full detail. It needs to provide and receive an Ethernet frame with the following structure:

- Ethernet MAC destination address
- Ethernet MAC source address
- Optional 802.1q VLAN tag (VLAN type and info field)
- Ethernet length/type field
- Payload

Frames on the FIFO interface do not contain preamble and SFD fields, which are inserted and discarded by the MAC on transmit and receive, respectively.

- On receive, CRC and frame padding can be stripped or passed through transparently.
- On transmit, padding and CRC can be provided by the user application, or appended automatically by the MAC independently for each frame. No size restrictions apply.

Note

On transmit, if ENET*n*_TCR[ADDINS] is set, bytes 6–11 of each frame can be set to any value, since the MAC overwrites the bytes with the MAC address programmed in the ENET*n*_PAUR and ENET*n*_PALR registers.

Table 23-39. FIFO interface frame format

Byte number	Field
0–5	Destination MAC address

Table continues on the next page...

Table 23-39. FIFO interface frame format (continued)

Byte number	Field
6–11	Source MAC address
12–13	Length/type field
14–N	Payload data

VLAN-tagged frames are supported on both transmit and receive, and implement additional information (VLAN type and info).

Table 23-40. FIFO interface VLAN frame format

Byte number	Field
0–5	Destination MAC address
6–11	Source MAC address
12–15	VLAN tag and info
16–17	Length/type field
18–N	Payload data

Note

The standard defines that the LSB of the MAC address is sent/ received first, while for all the other header fields — in other words, length/type, VLAN tag, VLAN info, and pause quanta — the MSB is sent/received first.

23.6.15.2 Data structure examples

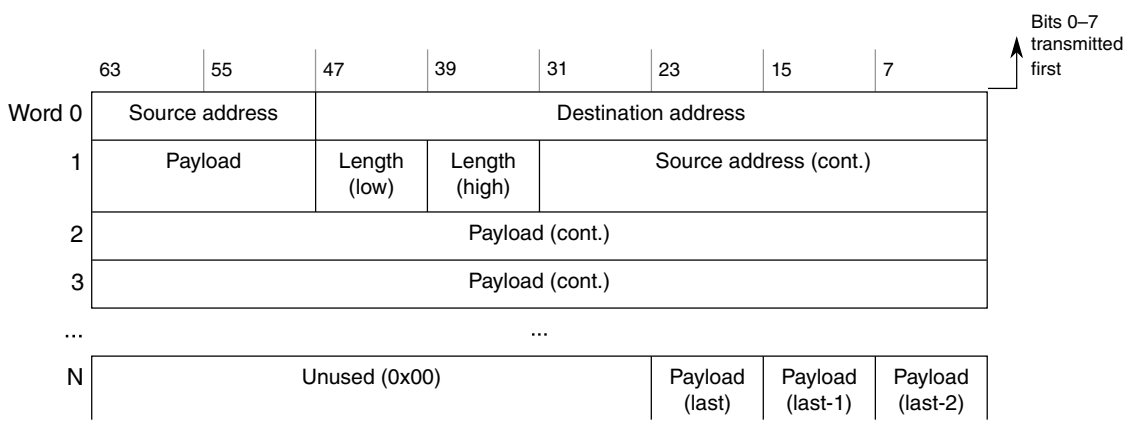


Figure 23-14. Normal Ethernet frame 64-bit mapping example

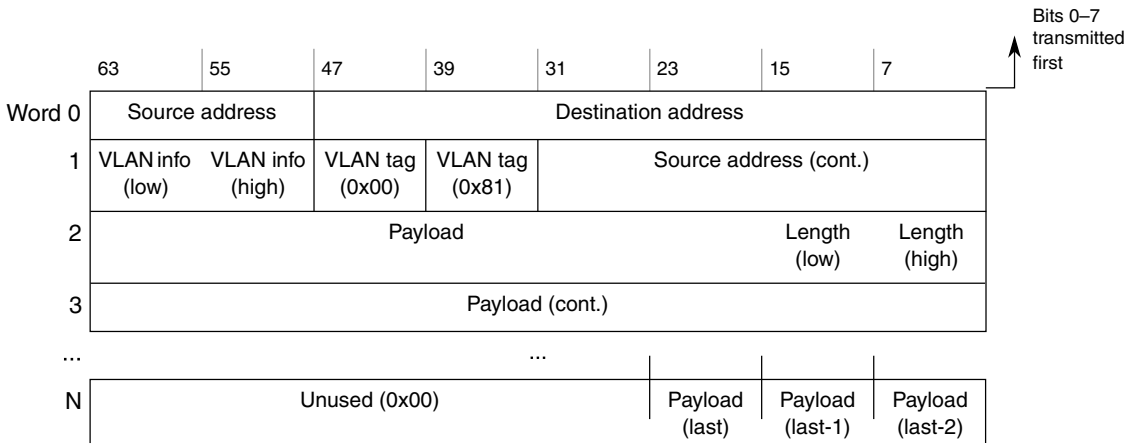


Figure 23-15. VLAN-tagged frame 64-bit mapping example

If CRC forwarding is enabled (CRCFWD = 0), the last four valid octets of the frame contain the FCS field. The non-significant bytes of the last word can have any value.

23.6.15.3 Frame status

A MAC layer status word and an accelerator status word is available in the receive buffer descriptor.

See Enhanced buffer descriptors for details.

The status is available with each frame with the last data of the frame.

If the frame status contains a MAC layer error (for example, CRC or length error), RxBD[ME] is also set with the last data of the frame.

23.6.16 FIFO protection

The following sections describe the FIFO protection mechanisms.

23.6.16.1 Transmit FIFO underflow

During a frame transfer, when the transmit FIFO reaches the almost empty threshold with no end-of-frame indication stored in the FIFO, the MAC logic:

- Stops reading data from the FIFO

- Asserts the MII error signal (MII_TXER) (1 in Figure 23-16) to indicate that the fragment already transferred is not valid
- Deasserts the MII transmit enable signal (MII_TXEN) to terminate the frame transfer (2)

After an underflow, when the application completes the frame transfer (3), the MAC transmit logic discards any new data available in the FIFO until the end of packet is reached (4) and sets the enhanced TxBD[UE] field.

The MAC starts to transfer data on the MII interface when the application sends a new frame with a start of frame indication (5).

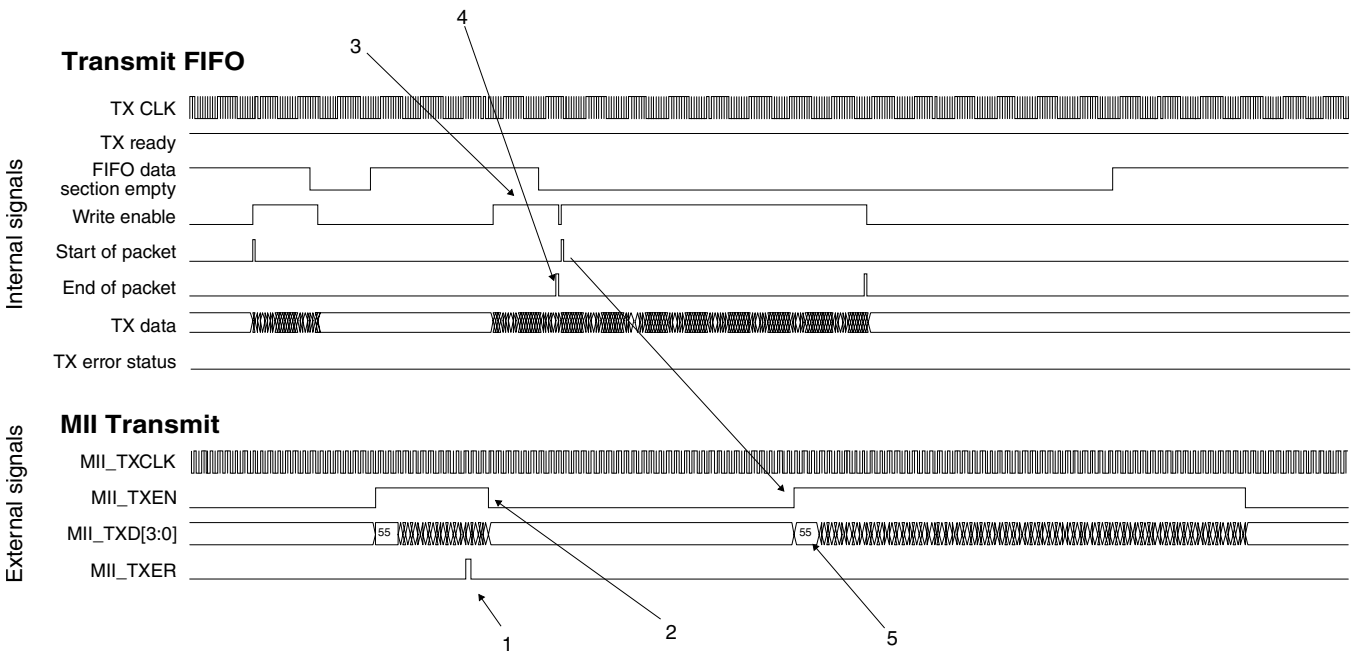


Figure 23-16. Transmit FIFO underflow protection

23.6.16.2 Transmit FIFO overflow

On the transmit path, when the FIFO reaches the programmable almost full threshold, the internal MAC ready signal is deasserted. The application should stop sending new data.

However, if the application keeps sending data, the transmit FIFO overflows, corrupting contents that were previously stored. The core logic sets the enhanced TxBD[OE] field for the next frame transmitted to indicate this overflow occurence.

Note

Overflow is a fatal error and must be addressed by resetting the core or clearing ENET*n*_ECR[ETHER_EN], to clear the FIFOs and prepare for normal operation again.

23.6.16.3 Receive FIFO overflow

During a frame reception, if the client application is not able to receive data (1), the MAC receive control truncates the incoming frame when the FIFO reaches the programmable almost-full threshold to avoid an overflow.

The frame is subsequently received on the FIFO interface with an error indication (enhanced RxBD[ME] field set together with receive end-of-packet) (2) with the truncation error status field set (3).

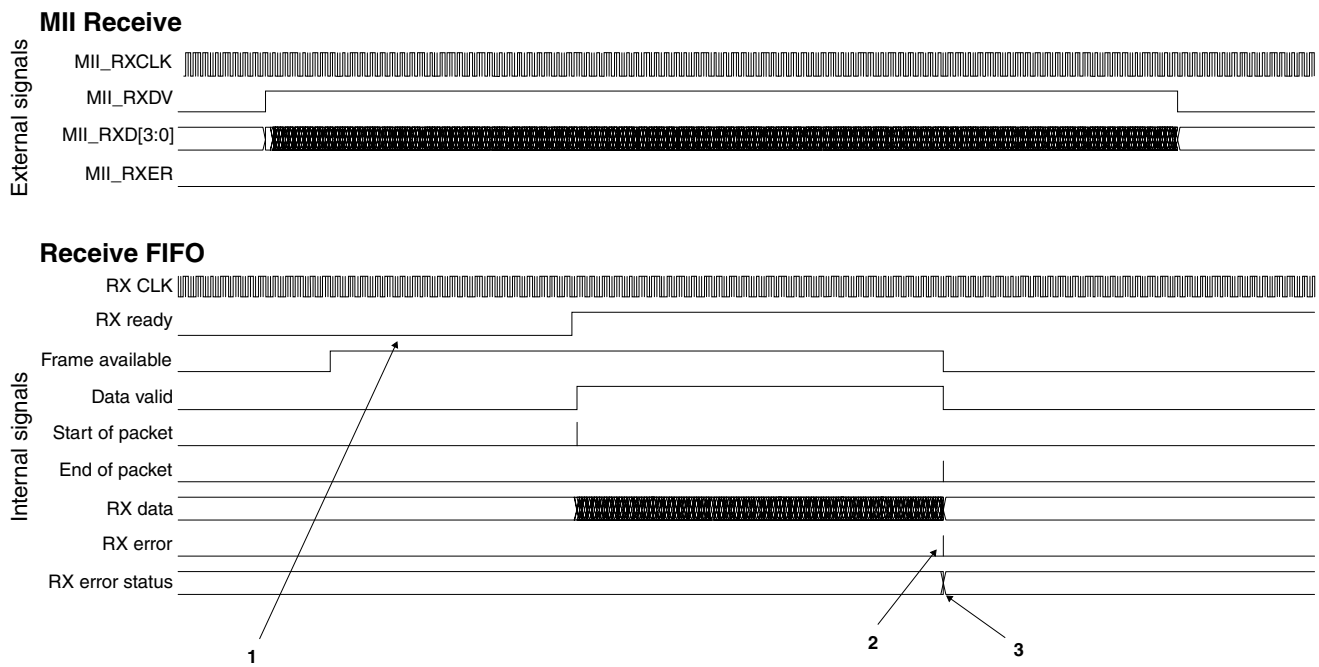


Figure 23-17. Receive FIFO overflow protection

23.6.17 PHY management interface

The MDIO interface is a two-wire management interface. The MDIO management interface implements a standardized method to access the PHY device management registers.

The core implements a master MDIO interface, which can be connected to up to 32 PHY devices.

23.6.17.1 MDIO clause 22 frame format

The core MDIO master controller communicates with the slave (PHY device) using frames that are defined in the following table.

A complete frame has a length of 64 bits made up of an optional 32-bit preamble, 14-bit command, 2-bit bus direction change, and 16-bit data. Each bit is transferred on the rising edge of the MDIO clock (MDC signal). The MDIO data signal is tri-stated between frames.

The core PHY management interface supports the standard MDIO specification (IEEE 802.3 Clause 22).

Table 23-41. MDIO clause 22 frame structure

ST	OP	PHYADR	REGADR	TA	DATA

Table 23-42. MDIO frame field descriptions

Field	Description
ST (2 bits)	Start indication field, programmed with ENET*n*_MMFR[ST] and equal to 01 for Standard MDIO (Clause 22).
OP (2 bits)	Opcode defines type of operation. Programmed with ENET*n*_MMFR[OP]. 01 Write operation 10 Read operation
PHYADR (5 bits)	Five-bit PHY device address, programmed with ENET*n*_MMFR[PA]. Up to 32 devices can be addressed.
REGADR (5 bits)	Five-bit register address, programmed with ENET*n*_MMFR[RA]. Each PHY can implement up to 32 registers.
TA (2 bits)	Turnaround time, programmed with ENET*n*_MMFR[TA]. Two bit-times are reserved for read operations to switch the data bus from write to read. The PHY device presents its register contents in the data phase and drives the bus from the second bit of the turnaround phase.
Data (16 bits)	Data, set by ENET*n*_MMFR[DATA]. Written to or read from the PHY

23.6.17.2 MDIO clause 45 frame format

The extended MDIO frame structure defined in IEEE 802.3 Clause 45 introduces indirect addressing. First, a write transaction to an address register is done, followed by a write or read transaction which will put the 16-bit data in the register or retrieve the register contents respectively. A preamble of 32 bits of logical ones is sent prior to every transaction. The MDIO data signal is tri-stated between frames.

The extended MDIO defines four transactions, which are determined by the two-bit opcode field.

Table 23-43. MDIO clause 45 frame structure

ST	OP	PRTAD	DEVAD	TA	ADDR/DATA

All bits are transmitted from left to right (Preamble bits first) and all fields have their Most-Significant bit sent first (leftmost in above table). The complete frame has a length of 64 bits (32-bit preamble, 14-bit command, 2-bit bus direction change, 16-bit data). Each bit is transferred with the rising edge of the MDIO clock (MDC).

The fields and transactions are summarized in the following tables.

Table 23-44. MDIO clause 45 frame field descriptions

Field	Description
ST	Start indication. Indicates the end of the preamble and start of the frame. This value is 00 for extended MDIO (Clause 45) frames.
OP	Opcode defines if a read or write operation is performed and is programmed with ENET*n*_MMFR[OP]. See Table 23-45 for more information. 00 Address write 01 Write operation 10 Read inc. operation 11 Read operation
PRTAD	The port address specifies a MDIO port. Each Port can have up to 32 devices which each can have a separate set of registers.
DEVAD	Device address. Up to 32 devices can be addressed (within a port).
TA	Turnaround time, programmed with ENET*n*_MMFR[TA]. Two bit-times are reserved for read operations to switch the data bus from write to read. The PHY device presents its register contents in the data phase and drives the bus from the second bit of the turnaround phase.
ADDR/DATA	16-bit address (for address write) or data, set by ENET*n*_MMFR[DATA], written to or read from the PHY.

Table 23-45. MDIO Clause 45 Transactions

Transaction Type	Description
Address	A write transaction to the internal address register of the device/port. The data section of the frame contains the value to be stored in the device's internal address "pointer" register for further transactions.
Write	Data write to a register. The 16 bit data will be written to the register identified by the device-internal address.
Read	Data is read from the register identified by the device-internal address.
Read inc.	Read with address postincrement. The register identified by the device-internal address is read. After this, the device-internal address is incremented. If the address register is all '1' (0xFFFF) no increment is done (i.e. increment does not wrap around).

23.6.17.3 MDIO clock generation

The MDC clock is generated from the internal bus clock (i.e., IPS bus clock) divided by the value programmed in ENET*n*_MSCR[MII_SPEED].

23.6.17.4 MDIO operation

To perform an MDIO access, set the MDIO command register (ENET*n*_MMFR) according to the description provided in MII Management Frame Register (ENETn_MMFR).

To check when the programmed access completes, read the ENET*n*_EIR[MII] field.

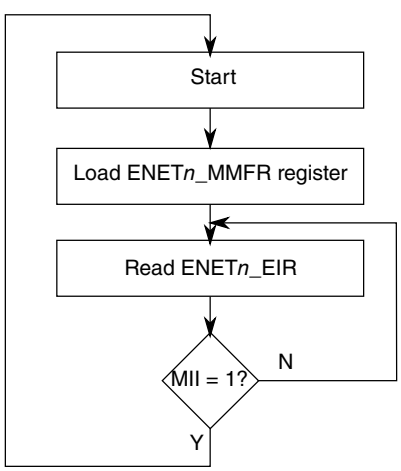


Figure 23-18. MDIO access overview

23.6.18 Ethernet interfaces

The following Ethernet interfaces are implemented:

- Fast Ethernet MII (Media Independent Interface)
- RMII 10/100 using interface converters/gaskets
- RGMII 10/100/1000 by way of interface converters/gaskets

The following table shows how to configure ENET registers to select each interface.

Mode	ECR[SPEED]	RCR[RMII_10T]	RCR[RMII_MODE]	RCR[RGMII_EN]
MII - 10 Mbit/s	0	—	0	0
MII - 100 Mbit/s	0	—	0	0
RMII - 10 Mbit/s	0	1	1	0
RMII - 100 Mbit/s	0	0	1	0
RGMII - 10 Mbit/s	0	1	0	1
RGMII - 100 Mbit/s	0	0	0	1
RGMII - 1000 Mbit/s[1]	1	—	0	1

1. Actual throughput is greater than 100 Mbit/s but significantly less than 1000 Mbit/s.

23.6.18.1 RMII interface

In RMII receive mode, for normal reception following assertion of CRS_DV, RXD[1:0] is 00 until the receiver determines that the receive event has a proper start-of-stream delimiter (SSD).

The preamble appears (RXD[1:0]=01) and the MACs begin capturing data following detection of SFD.

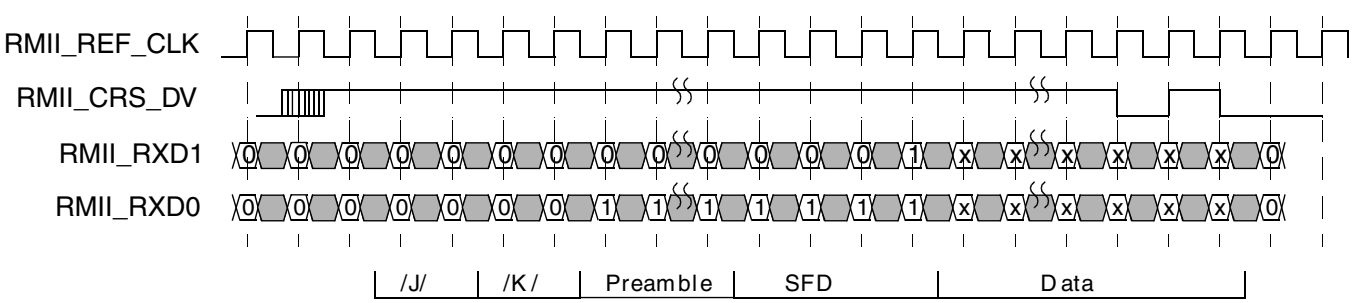


Figure 23-19. RMII receive operation

If a false carrier is detected (bad SSD), then RXD[1:0] is 10 until the end of the receive event. This is a unique pattern since a false carrier can only occur at the beginning of a packet where the preamble is decoded (RXD[1:0] = 01).

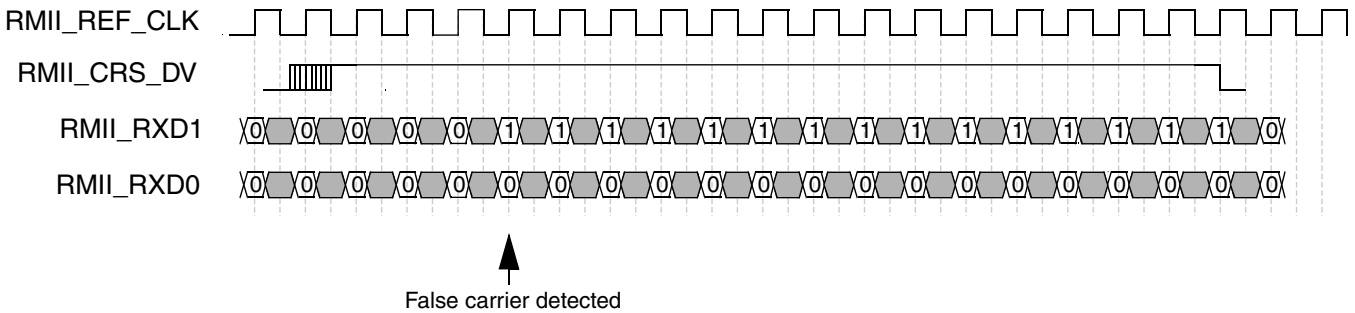


Figure 23-20. RMII receive operation with false carrier

In RMII transmit mode, TXD[1:0] provides valid data for each REF_CLK period while TXEN is asserted.

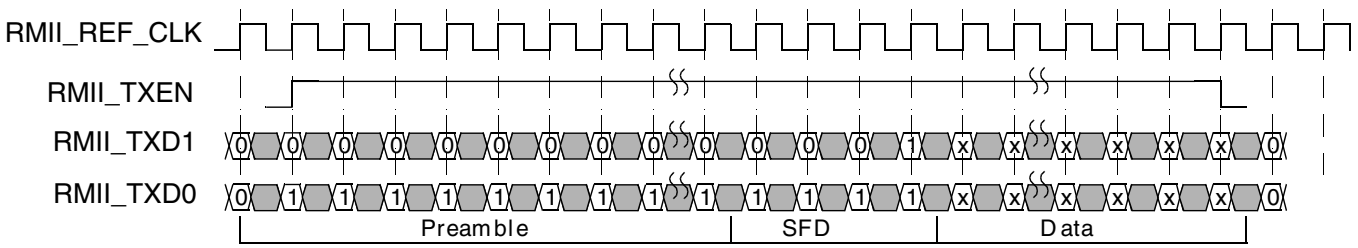


Figure 23-21. RMII transmit operation

23.6.18.2 RGMII interface

In RGMII modes, the data and control information is multiplexed by taking advantage of both edges of the reference clocks.

The data signals contain the lower four data bits on the rising edge and the upper four bits on the falling edge. The control signals are multiplexed into a single clock cycle using the same technique.

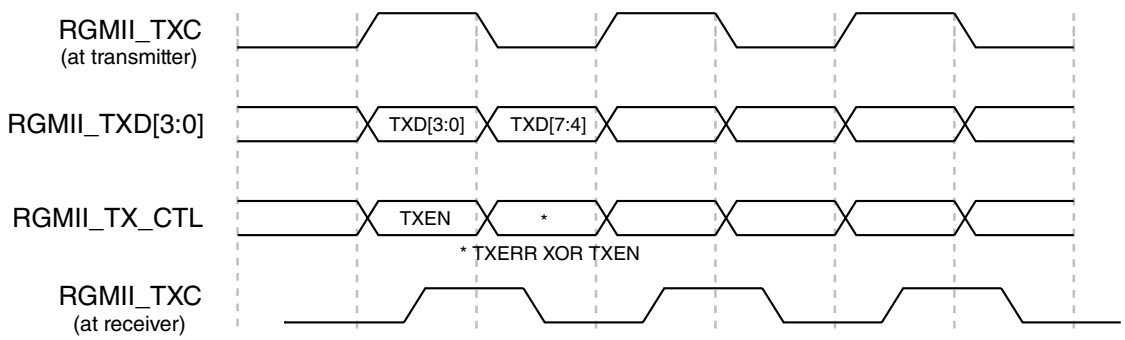


Figure 23-22. RGMII transmit operation

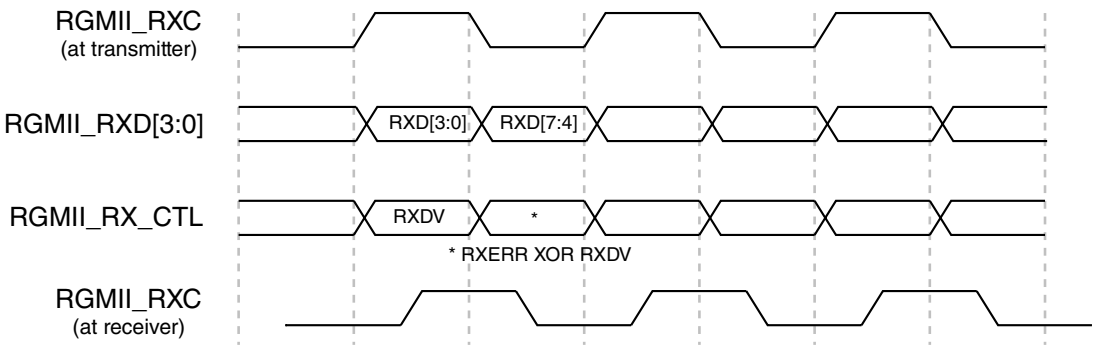


Figure 23-23. RGMII receive operation

23.6.18.3 MII Interface — transmit

On transmit, all data transfers are synchronous to MII_TXCLK rising edge. The MII data enable signal MII_TXEN is asserted to indicate the start of a new frame, and remains asserted until the last byte of the frame is present on the MII_TXD[3:0] bus.

Between frames, MII_TXEN remains deasserted.

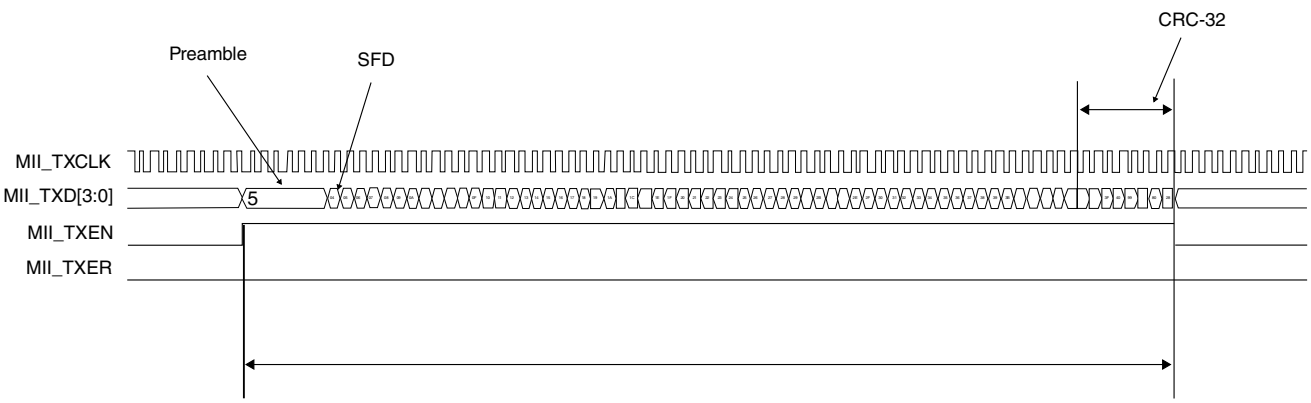


Figure 23-24. MII transmit operation

If a frame is received on the FIFO interface with an error (for example, RxBD[ME] set) the frame is subsequently transmitted with the MII_TXER error signal for one clock cycle at any time during the packet transfer.

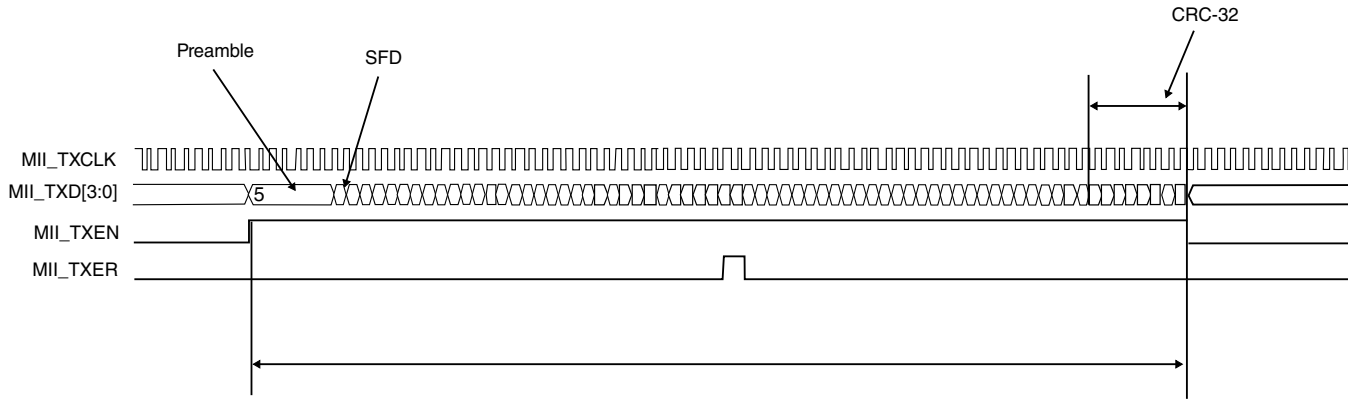


Figure 23-25. MII transmit operation — errored frame

23.6.18.3.1 Transmit with collision — half-duplex

When a collision is detected during a frame transmission (MII_COL asserted), the MAC stops the current transmission, sends a 32-bit jam pattern, and re-transmits the current frame.

(See Collision detection in half-duplex mode for details)

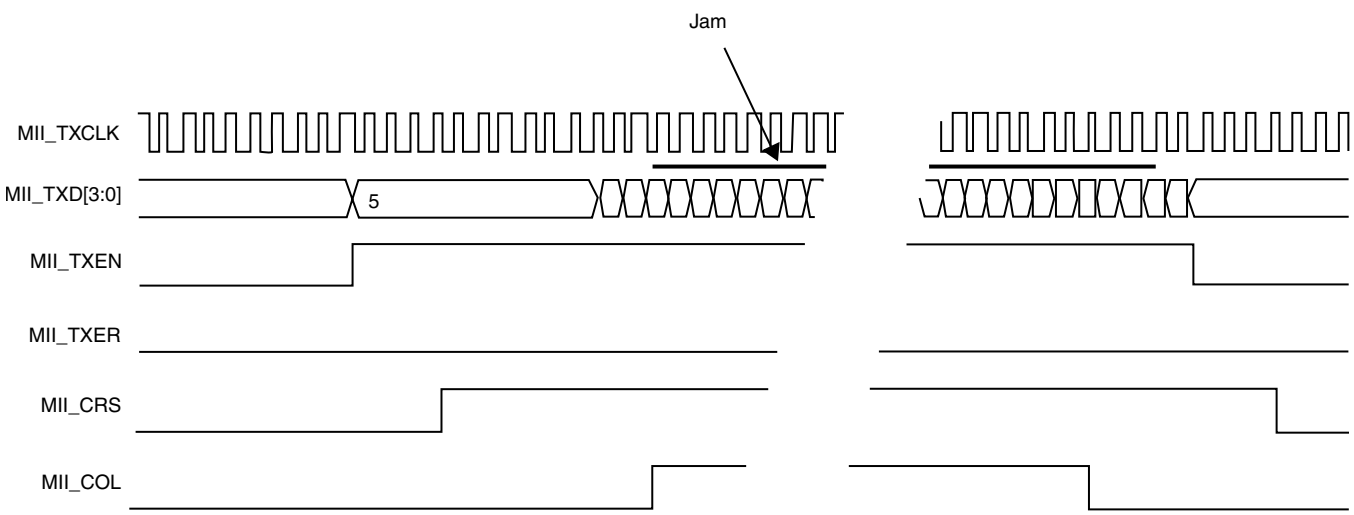


Figure 23-26. MII transmit operation — transmission with collision

23.6.18.4 MII interface — receive

On receive, all signals are sampled on the MII_RXCLK rising edge. The MII data enable signal, MII_RXDV, is asserted by the PHY to indicate the start of a new frame and remains asserted until the last byte of the frame is present on MII_RXD[3:0] bus.

Between frames, MII_RXDV remains deasserted.

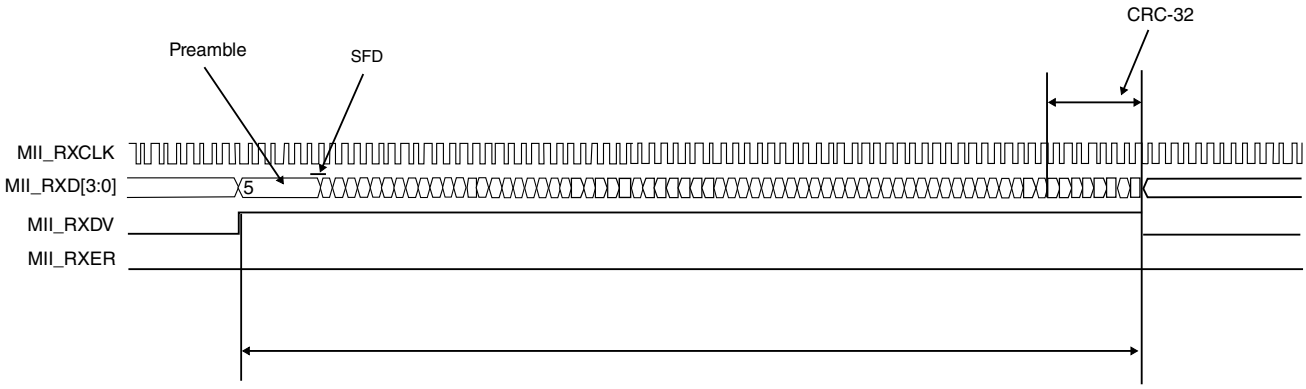


Figure 23-27. MII receive operation

If the PHY detects an error on the frame received from the line, the PHY asserts the MII error signal, MII_RXER, for at least one clock cycle at any time during the packet transfer.

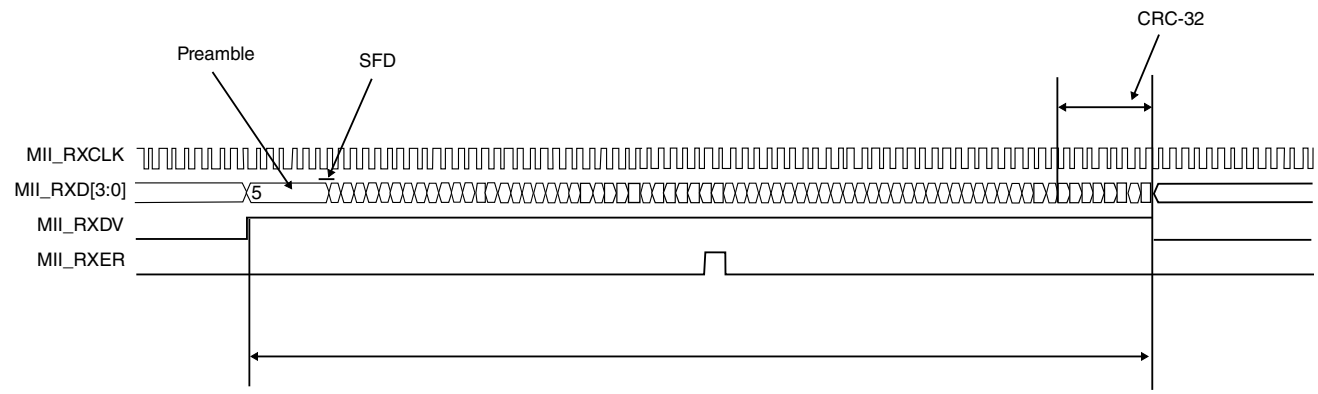


Figure 23-28. MII receive operation — errored frame

A frame received on the MII interface with a PHY error indication is subsequently transferred on the FIFO interface with RxBD[ME] set.

Chapter 24
Electrophoretic Display Controller (EPDC)

24.1 Overview

This chapter describes the architecture of the EPDC. It provides a detailed description of the block for digital design and software driver development.

The EPDC is a feature-rich, low power and high performance direct drive active matrix EPD controller. It is specifically designed to drive E•INK^{TM} EPD panels supporting a wide variety of TFT backplanes. The goal of the EPDC is to provide an efficient SoC integration of this functionality for e-paper applications, allowing a significant BOM cost saving over an external solution, while reaching much higher levels of performance at lower power. The EPDC module is defined in the context of an optimized HW/SW partitioning and works in conjunction with the PXP IP module to form a complete display processing solution.

The key features of the EPDC are as follows:

- TFT resolutions up to 4096 x 4096 pixels with 20 Hz refresh (programmable up to 8191 x 8191)
- TFT resolutions up to 1650 x 2332 pixels at 106 Hz refresh
- Industry standard bus interfaces (AMBA AXI and APB)
- Up to 5-bit pixel representation for up to 32 greyscale levels
- Up to 64 concurrent updates with partial update support, except for 32(5-bit) grey level panel for which only 16 concurrent updates can be used
- Automatic collision handling when used in conjunction with the i.MX device driver
- Dual-scan TFT drive mode to support ultra high resolution/refresh rate displays
- Flexible direct drive TFT interface supporting next generation source driver, gate driver and panel architectures, including LVDS, DDR and multi-level source drivers
- Unified generic configurable timing mode (Pigeon Mode) available on most panel timing control signals
- High performance pixel pipeline architecture to guarantee refresh performance at high pixel rates without the need for high internal clocking

- Ability to process multiple updates asynchronously to refresh/update operations with ability to intercept each frame scan will multiple update requests
- Performance tuning capabilities which can interface with SoC level memory arbitration mechanisms further guaranteeing frame refresh operation
- Decoupled clocking architecture allowing for independent and asynchronous clock sources for memory bus, peripheral bus and pixel clock domains
- Full and partial update mode support
- Support for up to 256 waveform modes, also support optimal waveform auto-selection based on grey level of the pixels being updated
- Low power mode operation via architectural clock gating
- Update buffer analysis functions to get information like collision rectangle, grey level

24.1.1 EPDC Block Diagram

The top-level view of the EPDC is shown in the following figure.

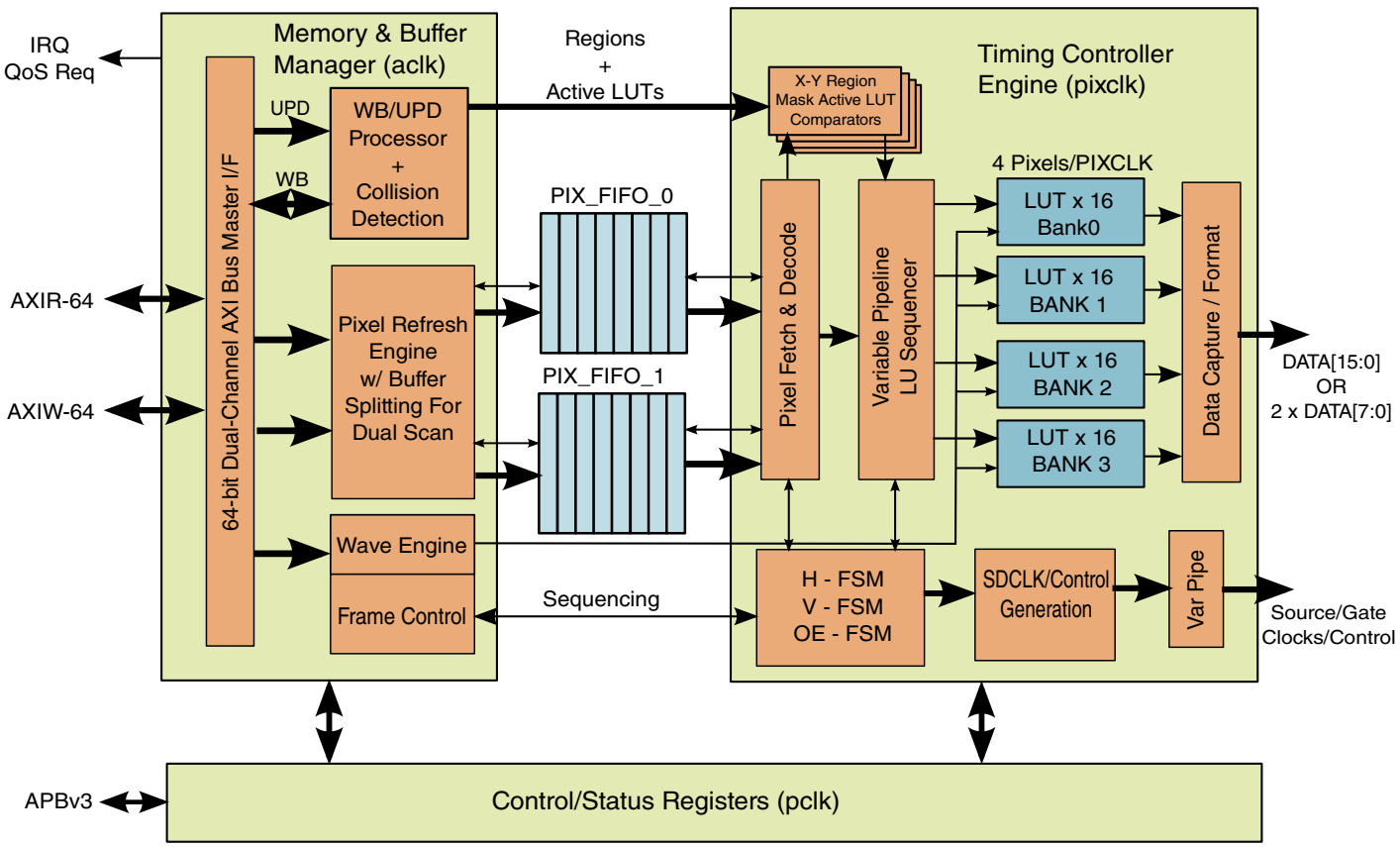


Figure 24-1. EPDC Top-Level Block Diagram

The EPD function can be separated into two major asynchronous processes. The first is the front end portion which is responsible for processing display updates. The second is the function of driving waveforms to the TFT panel in order to update the screen contents (the refresh operation). The EPDC is comprised of two major submodules (MBM and TCE) which mirror the architectural split in the EPD algorithm. These submodules communicate through a number of control/sequencing signals and data (through the pixel FIFOs and LUT memories):

- Memory and Buffer Manager (MBM): This submodule is responsible for all memory-related operations and LUT life cycle control, which involves the frame count management for all updates. This module is clocked by both *aclk* and *pixclk*. It also encapsulates the APB register interface which is clocked by *aclk* too.
- Timing Control Engine (TCE): This submodule is responsible for performing TFT scan frame refreshes. It is clocked by *pixclk*. It also houses the LUT memory system.

24.2 External Signals

The following table describes the external signals of EPDC:

Table 24-1. EPDC External Signals

Signal	Description	Pad	Mode	Direction
EPDC_BDR0	Panel-Border Control (SW controlled)	EIM_DA2	ALT8	O
EPDC_BDR1	Panel-Border Control (SW controlled)	EIM_DA3	ALT8	O
EPDC_DATA00	Source Driver-Shift signal	EIM_A16	ALT8	O
EPDC_DATA01	Source Driver-Shift signal	EIM_DA10	ALT8	O
EPDC_DATA02	Source Driver-Shift signal	EIM_DA12	ALT8	O
EPDC_DATA03	Source Driver-Shift signal	EIM_DA11	ALT8	O
EPDC_DATA04	Source Driver-Shift signal	EIM_LBA	ALT8	O
EPDC_DATA05	Source Driver-Shift signal	EIM_EB2	ALT8	O
EPDC_DATA06	Source Driver-Shift signal	EIM_CS0	ALT8	O
EPDC_DATA07	Source Driver-Shift signal	EIM_RW	ALT8	O
EPDC_DATA08	Source Driver-Shift signal	EIM_CS1	ALT8	O
EPDC_DATA09	Source Driver-Shift signal	EIM_DA15	ALT8	O
EPDC_DATA10	Source Driver-Shift signal	EIM_D16	ALT8	O
EPDC_DATA11	Source Driver-Shift signal	EIM_D23	ALT8	O
EPDC_DATA12	Source Driver-Shift signal	EIM_D19	ALT8	O
EPDC_DATA13	Source Driver-Shift signal	EIM_DA13	ALT8	O
EPDC_DATA14	Source Driver-Shift signal	EIM_DA14	ALT8	O
EPDC_DATA15	Source Driver-Shift signal	EIM_A25	ALT8	O
EPDC_GDCLK	Gate Driver-Clock	EIM_A21	ALT8	O
EPDC_GDOE	Gate Driver-Output Enable	EIM_A23	ALT8	O
EPDC_GDRL	Gate Driver-Shift direction	EIM_A24	ALT8	O
EPDC_GDSP	Gate Driver-Start Pulse	EIM_A22	ALT8	O
EPDC_PWR_COM	Panel-Power control (SW controlled)	EIM_EB0	ALT8	O
EPDC_PWR_CTRL0	Panel-Power control (SW controlled)	EIM_A18	ALT8	O
EPDC_PWR_CTRL1	Panel-Power control (SW controlled)	EIM_A19	ALT8	O
EPDC_PWR_CTRL2	Panel-Power control (SW controlled)	EIM_A20	ALT8	O
EPDC_PWR_CTRL3	Panel-Power control (SW controlled)	EIM_D28	ALT8	O
EPDC_PWR_IRQ	Panel-Power irq	EIM_OE	ALT8	I
EPDC_PWR_STAT	Panel-Power status good	EIM_A17	ALT8	I
EPDC_PWR_WAKE	Panel-Power control wake signal (SW controlled)	EIM_D29	ALT8	O
EPDC_SDCE0	Source Driver-Chip-enable/Start-Pulse	EIM_DA4	ALT8	O
		EIM_EB3	ALT8	
EPDC_SDCE1	Source Driver-Chip-enable/Start-Pulse	EIM_DA5	ALT8	O

Table continues on the next page...

Table 24-1. EPDC External Signals (continued)

Signal	Description	Pad	Mode	Direction
EPDC_SDCE2	Source Driver-Chip-enable/Start-Pulse	EIM_DA6	ALT8	O
EPDC_SDCE3	Source Driver-Chip-enable/Start-Pulse	EIM_DA7	ALT8	O
EPDC_SDCE4	Source Driver-Chip-enable/Start-Pulse	EIM_DA8	ALT8	O
EPDC_SDCE5	Source Driver-Chip-enable/Start-Pulse	EIM_DA9	ALT8	O
EPDC_SDCE6	Source Driver-Chip-enable/Start-Pulse	EIM_D22	ALT8	O
EPDC_SDCE7	Source Driver-Chip-enable/Start-Pulse	EIM_D24	ALT8	O
EPDC_SDCE8	Source Driver-Chip-enable/Start-Pulse	EIM_D25	ALT8	O
EPDC_SDCE9	Source Driver-Chip-enable/Start-Pulse	EIM_BCLK	ALT8	O
EPDC_SDCLK_N	Negative Source Driver-Shift Clock	EIM_DA0	ALT8	O
EPDC_SDCLK_P	Positive Source Driver-Shift Clock	EIM_D31	ALT8	O
EPDC_SDLE	Source Driver-Latch Enable	EIM_DA1	ALT8	O
EPDC_SDOE	Source Driver-Output Enable	EIM_D27	ALT8	O
EPDC_SDOED	Source Driver-Output Enable (to VDD)	EIM_D26	ALT8	O
EPDC_SDOEZ	Source Driver-Output Enable (to Zero)	EIM_D30	ALT8	O
EPDC_SDSHR	Source Driver-Shift dir	EIM_EB1	ALT8	O
EPDC_VCOM0	Panel-VCOM	EIM_D17	ALT8	O
EPDC_VCOM1	Panel-VCOM	EIM_D18	ALT8	O

24.3 Programming Model

From an application perspective, the EPDC HW/SW solution aims to abstract much of the details of driving an EPD display from the user space.

This is achieved through a blend of HW features and the kernel-level SW driver implementation of the EPD frame buffer. The scope of this chapter mostly pertains to EPDC HW but in order to correctly define the context of the programming model, certain assumptions about the driver architecture (including use of the ePXP IP) are described below.

24.3.1 Assumptions

The scope of the EPDC does not include certain functions performed by the driver and application layer.

These features are as follows:

- Maintenance of update and collision lists. This includes, but is not limited to, responsibility for managing update order. Updates are processed in the order they are received by the EPDC. Actual start times of multiple updates can occur on the same frame scan. All pixels contained within an update will always begin on the same frame scan.
- Border control. This function is assumed to be performed by a GPIO functionality.
- EPD Panel Power Management. It is assumed that the driver and application layers will control the panel PMIC functions including relevant interfaces such as I^2C and 4-wire power sequencing signals.
- Reading and decompression of the waveform file. This is assumed to be unpacked, reformatted and written to system memory into a form that is easily accessed by the EPDC.
- Waveform mode selection. The EPDC provides a mechanism through its histogram analysis feature (which can be run as part of any frame/region processing operations) to allow the SW driver to characterize the region/buffer in terms of gray-map content. For example, EPDC can determine if the update region (considering only those pixels undergoing a transition) contains only 2, 4, 8 or 16 gray levels, HW can automatically select waveform mode according to a programmed LUT holding mapping between grey level and waveform mode. SW driver can also manually choose the appropriate waveform update mode (e.g., when doing an update to repair ghosting artifacts).

24.3.2 Register Space (Write/Set/Clear/Toggle)

The EPDC registers utilize four word address locations per 32-bit register.

For control registers (especially interrupts), this provides for a unique address for the following operations:

- Write (0x0). The base address of the register allows full writes to the 32-bit register.
- Set (0x4). This address allows a set operation on the register or its fields; writing a 1 to a register bit/field with the set address will set that bit. Writing a 0 to the set address has no effect.
- Clear (0x8). This address allows a clear operation on the register or its fields; writing a 1 to a register bit/field with the clear address will clear that bit or field.

NOTE

Interrupt status bits (such as those contained in EPDC_IRQ) must be cleared with the clear address only. Writing to the EPDC_IRQ register with a 0 will not clear the interrupt status/ source. Writing a 0 to the clear address has no effect.

- Toggle (0xc). This address allows a toggle operation on the register or its fields; writing a 1 to a register bit/field with the toggle address will invert the state of that bit/field. This address is especially useful because it provides the ability to perform a read-modify-write operation with a single write operation. Writing a 0 to the toggle address has no effect.

24.3.3 Interrupts

24.3.3.1 Interrupt Sources

The EPDC contains 72 (64 of LUT completes IRQs and 8 of common IRQs) unique interrupt sources. Each interrupt source has an interrupt status bit captured in EPDC_IRQ /EPDC_IRQ1/EPDC_IRQ2 .

When an interrupt event occurs, the appropriate bit within the EPDC_IRQ* register is set by the EPDC. Each of the bits in EPDC_IRQ is first AND'ed with its enable bit (in EPDC_IRQ_MASK*), and then the masked bits are logically OR'ed together to generate the final output.

24.3.3.2 Enabling/Masking Interrupts

Each of the EPDC interrupt sources can be individually enabled through the bits in the EPDC_IRQ_MASK* register.

Even if an interrupt source is masked out (IRQ_EN bit set to 0), its status will still be available in the EPDC_IRQ* register. The difference is that it does not result in the interrupt line being asserted.

24.3.3.3 Handling/Clearing Interrupts

When the software driver is entered as a result of an EPDC interrupt it should read the EPDC_IRQ* interrupt status register.

Because multiple interrupts may have fired, in the case of a collision for example, both WB_CMPLT_IRQ and LUT_COL IRQ fields of EPDC_IRQ will be set. Once the interrupt has been handled, the driver should clear the interrupt source by writing to the CLEAR address (see Register Space (Write/Set/Clear/Toggle) for details).

When collision has occurred and the LUT_COL_IRQ bit is set, SW will handle the collision and clear LUT_COL_IRQ. Please note HW will also clear the entire EPDC_STATUS_COL* register at the sametime on LUT_COL_IRQ clearing by SW.

24.3.4 Controller Setup

Before screen update operations can be performed, the EPDC should be configured appropriately.

These rudimentary setup operations include:

- Waveform data must be present in memory (in a format that is consistent with the method in which the EPDC accesses it). Because the format of the waveform data is supplier proprietary information, it is not discussed in this chapter.
- Configuring the panel architecture parameters (source and gate-driver configuration)
- Configuring line and frame timing parameters
- Initializing the working buffer and panel

24.3.4.1 Memory Requirements

EPDC requires continuous access to the unpacked waveform data, the update region buffer, and its working buffer.

These are all expected to reside in system memory (typically external DRAM). As such, each has particular requirements:

- The Working Buffer (WB), which is pointed to by EPDC_WB_ADDR, must have EPDC_RES[HORIZONTAL] x EPDC_RES[VERTICAL] x 2 bytes allocated. This buffer is used by the EPDC to process updates and perform TFT refresh operations.
  - EPDC_RES[HORIZONTAL] mod 4 must equal to 0 for memory allocation calculations because the EPDC constructs each line of the working buffer at a 64-bit boundary.
- The Update Buffer, which is pointed to by EPDC_UPD_ADDR (and can be redefined with a different address for each update), must have the following allocation in memory:
  a. when stride feature not enabled

    - EPDC_UPD_SIZE[HEIGHT] x EPDC_UPD_SIZE[WIDTH] bytes
    - Note that EPDC_UPD_SIZE[WIDTH] mod 8 must equal to 0 for memory allocation purposes. This means that for the address size allocation, the WIDTH field must be rounded up to the nearest number so that it can be divided by 8. This is because the EPDC reads each line of the update buffer at a 64-bit boundary (it should be noted that this is consistent with the ePXP IP output). The actual value programmed for the update can be at the pixel granularity.
- The waveform data set size is a function of mode count, number of temperature tables and number of frames required for each temperature-compensated mode.

In summary, the memory requirements are as follows (in bytes):

- WB-roundup4(EPDC_RES[HORIZONTAL]) x EPDC_RES[VERTICAL] x 2

24.3.4.2 Panel Architecture Configuration

The EPDC is designed to directly drive EPD panels which typically expose the source and gate driver IC interfaces (these are often abstracted in traditional displays such as TFT-LCD).

The source and gate driver ICs are responsible for driving the TFT matrix. There are variations in both the panel architectures and the underlying source and gate driver IC functionality.

All the key configuration parameters are defined in the EDPC_RES, EPDC_FORMAT, EPDC_TCE_CTRL, EPDC_TCE_SDCFG and EPDC_TCE_GDCFG registers as follows.

- EPDC_RES:
    - HORIZONTAL: The panel horizontal resolution (in pixels). The horizontal resolution must be an integer multiple of the source driver PIXELS_PER_SDCLK value. It also must be an integer multiple of the PIXELS_PER_CE value.
    - VERTICAL: The panel vertical resolution (in pixels). This can be any integer value.
- EPDC_FORMAT:
    - DEFAULT_TFT_PIXEL: This field should almost always be left at 0x00. This register field is used as the value for the TFT pixel during a frame scan when a pixel is not being updated. This condition is commonplace especially when using partial updates or updates which do not encompass the entire screen resolution. This value must always correspond to the state the source driver should see when

no voltage change is needed. Incorrectly setting this value will damage the EPD material.

- TFT_PIXEL_FORMAT: This enumerated field defines the width of the TFT pixel. The TFT pixel is defined as the per-pixel voltage control value. The most common pixel format is 2B (2 bits per pixel). Panels which support source driver voltage modulation (multi-level voltage control) should use the 4B format. For source drivers that support 7 levels, the 4B format can be used and the unused DATA output pins should be left floating at the board level connection (specifically DATA3, 7, 11, 15). The 2BV and 4BV are variations which require the waveform data to supply a 2-bit VCOM value for each pixel.

- EPDC_TCE_CTRL:
  - SDDO_WIDTH: This selects 8- or 16-bit DATA operation using enumerations 8-bit and 16-bit. This field is used to describe useful data for panels that support LVDS. DATA[15:8] would be used to drive differential data. In these modes, SDDO_WIDTH would be set to 8-bit.
  - VCOM_MODE, VCOM_VAL: This VCOM_MODE bit allows the EPDC to either use the VCOM value supplied in the waveform or a software programmable value defined in the VCOM_VAL field, for panels which support VCOM modulation.
  - DDR_MODE: This mode-bit should be set for panels which expect source-driver data to be available on both edges of the SDCLK. This is common place for panels that support differential signaling, but the feature is not limited to LVDS panels. For example, the EPDC supports DDR modes which make full use of DATA[15:0]. In these modes, DDR_MODE = 1 and LVDS_MODE = 0.
  - LVDS_MODE: This mode always requires DDR_MODE to be set. When this mode bit is set, differential signaling is enabled such that DATA[15:8] always drives the inverse of the pixel data which is presented on DATA[7:0]. Because LVDS source-drivers use 2 pins per data-bit, they are typically configured to work in DDR mode. Differential signaling is also supported on the SDCE pins (as an option selected by LVDS_MODE_CE). In this mode SDCE[9:5] are used to drive the inverse differential signals for SDCE[4:0].
  - LVDS_MODE_CE: When LVDS_MODE is set, this bit can also be set to allow SDCE[9:5] to act as differential pairs for each SDCE[4:0].
  - DUAL_SCAN: Setting this field enables the dual scan function. When this field is set, all other parameters (except resolution) should be programmed in reference to each half of the panel. Because DUAL_SCAN modes require 2x the pixel-generation capability, there are limitations to the available TFT modes supported in this configuration (see Table 24-2 for details).
  - SCAN_DIR0, SCAN_DIR1: These fields determine the vertical scan direction of the panel.

    - SCAN_DIR0: Determines the vertical scan direction of the panel when DUAL_SCAN = 0. When DUAL_SCAN = 1, it determines the vertical scan direction of the top-half of the panel (0 means scan down and 1 means scan up).
    - SCAN_DIR1 only has meaning when DUAL_SCAN = 1. It defines the vertical scan direction of the bottom half of the panel.
  - PIXELS_PER_SDCLK: This is a key configuration and timing parameter. Each source driver latches a number of TFT pixels per shift clock period (SDCLK). This register defines how many pixels are driven per SDCLK period. When DDR_MODE = 1, pixels are driven on both edges of the clock, so this value must be adjusted accordingly. See Table 24-2 for the required values per mode.
- EPDC_TCE_SDCFG:
  - SDCLK_HOLD: Holds the SDCLK low during the LINE_BEGIN time. The purpose of this field is to allow the user to set a LINE_BEGIN time which is not an integer multiple of SDCLKs (for fine tuning the line-time).
  - SDSHR: Defines the value of the SDSHR output signal which determines the source driver output order mapping.
  - NUM_CE: Defines the total number of source driver chip-enables (must be 1-10). This number doesn't always correspond to the number of source drivers in the panel. Many panels require only a single chip-enable signal which is active during the entire line data time. This signal is often called SPH (Horizontal Start Pulse) in such cases.
  - PIXELS_PER_CE: This register defines the span of each chip-enable expressed in pixels. For panels that utilize a single global CE, this value should be set to the horizontal resolution of the panel.
  - SDDO_REFORMAT: This enumerated field allows for TFT-pixel level reordering within DATA. For most panels it is recommended to set SDDO_REFORMAT to the REVERSE_PIXELS enumeration.
  - SDDO_INVERT: When this field is set, the values on DATA are inverted relative to the values extracted from the waveform LUT operations.
- EPDC_TCE_GDCFG:
  - GDRL: Defines the value of the GDRL output signal which determines the gate driver pulse shift direction.
  - GDOE_MODE: This field selects between two possible methods for driving the GDOE signal. When GDOE_MODE = 0, the GDOE output is always driven during the frame scan time, except during FRAME_SYNC when it is driven low. When GDOE_MODE = 1, the GDOE signal mimics the GDCLK signal but is only active during the frame-data time (FD). For most panels the recommended setting is 0.
  - GDSP_MODE: This field selects between two possible methods for driving the GDSP signal. When GDSP_MODE = 0, the GDSP signal is driven on the first

line clock time of the FRAME_BEGIN time. The signal is driven for one GDCLK period (same as line time), and can be shifted using the GDSP_OFFSET control. When GDSP_MODE = 1, GDSP is active during the entire FRAME_SYNC time, and GDSP_OFFSET has not effect in this mode.

24.3.4.3 TFT Panel Timing Configuration

The EPDC provides a simple primary set of registers to define TFT line and frame (refresh) timing.

In addition to these, there are also low-level timing and polarity registers to provide additional flexibility in dealing with future TFT architectures.

The following figure shows how the horizontal and vertical timing is defined in terms of the various timing parameters. When configuring the TFT timing, the following goals should be met:

- Arrive at a refresh rate that matches the waveform requirement (for example, 50 Hz for 50 Hz waveform). The refresh rate is defined as the time between each frame sync event and can be expressed as a multiple of the line timing and the total number of vertical lines.
- Meet the various timing requirements of the gate and source driver clock and control signals (the blanking period provide an infrastructure for controlling these).
- Stay below the maximum source-driver clock (usually referred to as CL) frequency.
- Arrive at a source driver clock frequency which in turn defines the exact pixclk frequency. In most cases, E-INK™ provides the expected SDCLK frequency.
- Meet the line-timing (LT) requirements of the E-INK panels and their associated waveforms.

For proper waveform performance, it is critical to match the refresh frequency to that required by the waveform and associated panel. Deviation from this frequency results in short term inaccuracy of color and in the long term, complete loss of coherency between the EPDC's internal representation of color and the physical representation on the screen.

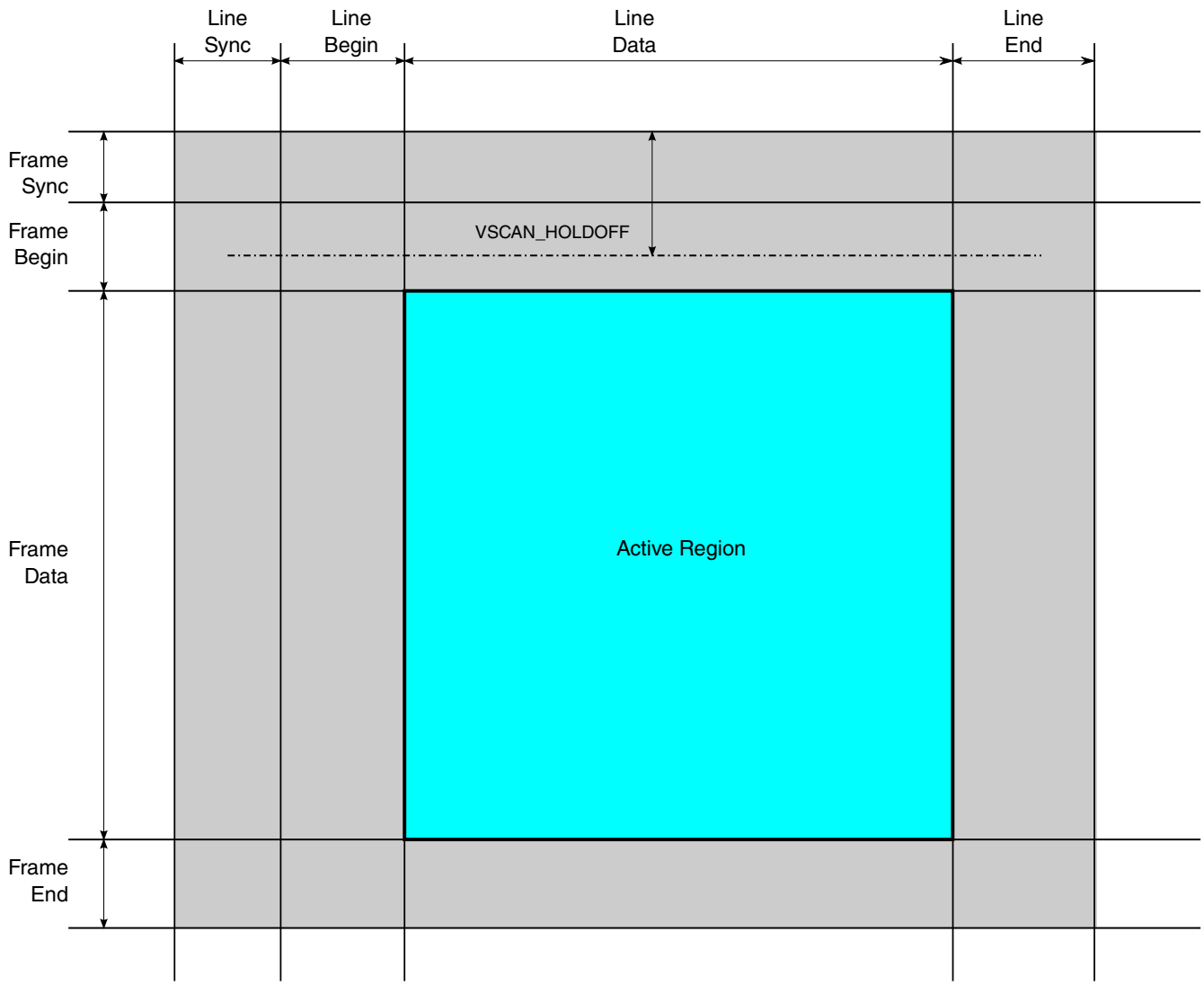


Figure 24-2. Frame Timing

The following equations define the timing of the panel scan frame (where LT = line-timing and FT = frame-timing):

$$LT = TPIXCLK\ (LS + LB + (HRES \times \left(\frac{Ratio}{PIXELSPERSDCLK}\right) + LE)$$

$$FT = LT \times (FS + FB + VRES + LE)$$

Ratio is defined between the internal PIXCLK and the external source driver clock SDCLK. The ratio is automatically determined by the EPDC according to Table 24-2 . A simple method to determine the ratio is as follows:

- When DDR_MODE = 1, RATIO = 4
- When DDR_MODE = 0
  - If PIXELS_PER_SDCLK = 8, RATIO = 4
  - Else, RATIO = 2

Typically it is recommended to seed these equations with an estimated (or provided) SDCLK frequency and adjust the various timings until the desired refresh rate is attained.

The following figure shows line timing for cases, where DDR_MODE = 0 (those cases where data is only sampled by the source driver on the rising edge of SDCLK). Depending on the mode (see Table 24-2), the ratio is a function of TFT pixel width and PIXELS_PER_SDCLK, so the timings are shown generically. For the purposes of this diagram, RATIO can either be 2 or 4.

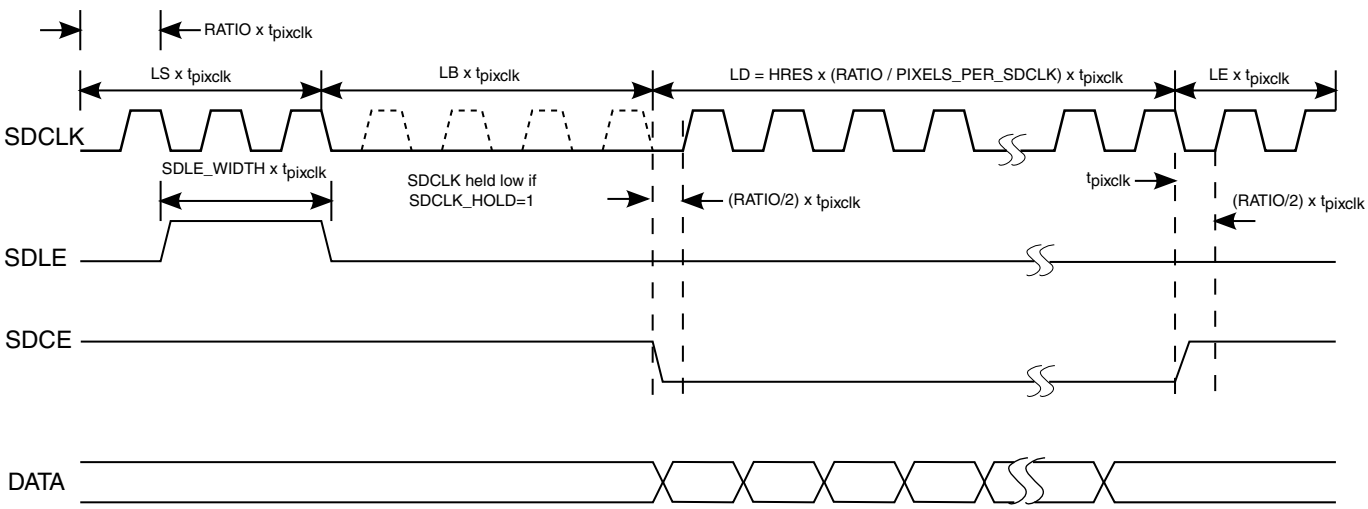


Figure 24-3. Line Timing (DDR_MODE = 0)

As per Table 24-2 , when DDR_MODE=1 (for example, for source drivers that require data to be driven on each edge of SDCLK), the RATIO is always set to 4. The timing for such cases is shown in the figure below.

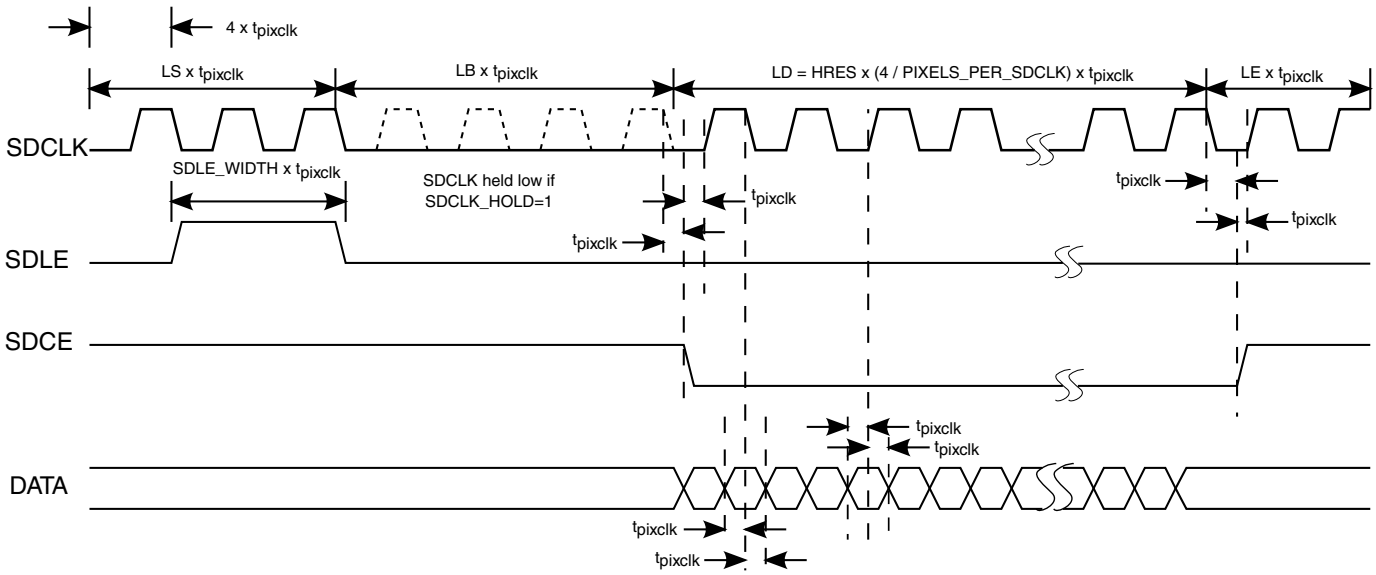


Figure 24-4. Line Timing (DDR_MODE = 1) - RATIO = 4

Vertical timing or frame timing (FT) is always expressed in terms of line timing (LT). This is shown in the following figure, followed by a detailed view of the GDCLK timing (which has programmable duty cycle to allow programming of gate-on/gate-off time) in Figure 24-6. The following important points should be noted in regard to gate driver timing:

- All gate driver timing signals are independently referenced to the LINE_SYNC leading edge (this includes GDCLK_OFFSET, GDSP_OFFSET, and GDOE_OFFSET).
- GDSP_OFFSET is only supported for GDSP_MODE = 1
- For GDOE_MODE = 1, if GDOE is expected to be delayed relative to GDCLK, the GDOE_OFFSET value should be set at a value appropriately greater than GDCLK_OFFSET.
- All offset settings are programmed in terms of PIXCLK cycles.
- By default, if all offset values are set to 0, the gate driver signals are always one PIXCLK cycle delayed from the LINE_SYNC leading edge.
- The LINE_SYNC leading edge doesn't depend on the LINE_SYNC_WIDTH value. In cases where LINE_SYNC_WIDTH is set to some value that is less than LINE_SYNC, a virtual leading edge which determines the line start time still exists.

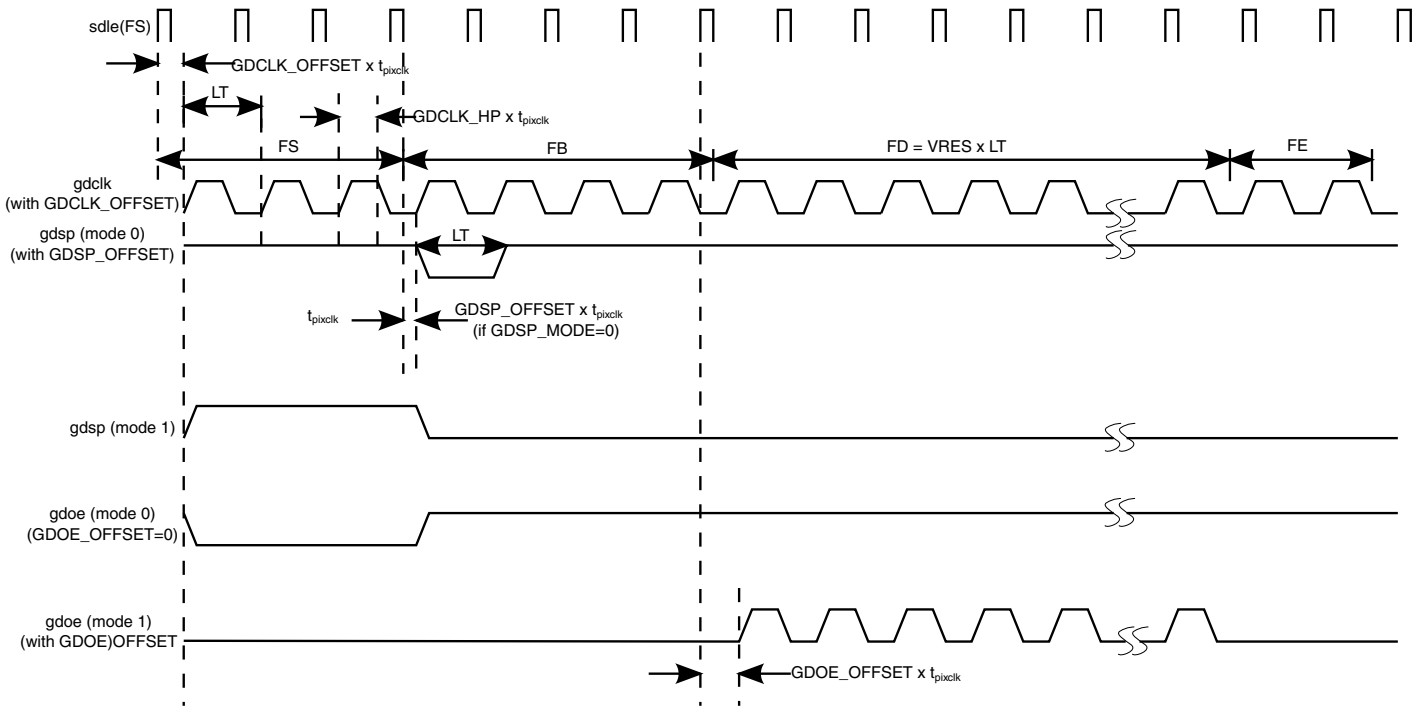


Figure 24-5. Frame Timing (Vertical)

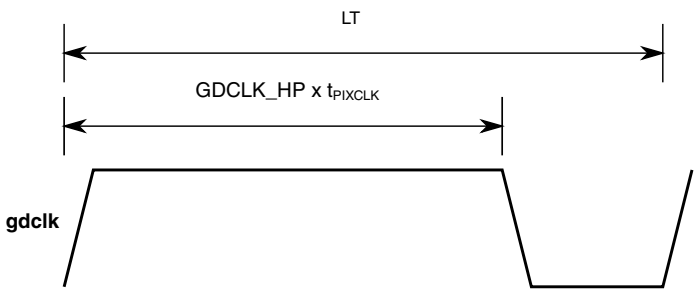


Figure 24-6. GDCLK Duty Cycle

An example calculation based on the 800 x 600 E-INK 6" panel with 50 Hz waveforms is shown as follows:

- PIXELS_PER_SDCLK = 4
- Ratio = 2 (2 bpp, 8-bit single-ended, SDR per Table 24-2)
- tPIXCLK = 17.64 MHz (assuming available chip clock frequency)
- LS = 20 x tPIXCLK
- LB = 8 x tPIXCLK
- LD = 800 x (2/4) x tPIXCLK = 400 x tPIXCLK
- LE = 142 x tPIXCLK
- LT = (20 + 8 + 400 + 142) x tPIXCLK = 32.31293 uS

LINE_SYNC_WIDTH does not affect line timing. It should be set to the same value as LINE_SYNC, unless LB and LE are short and it is used to shorten the width of SDLE whilst still maintaining the same line-timing.

It is important to meet the target line time of the panel which was 32.312 uS. Then the frame-rate can be calculated and given the vertical timings:

- FS = 4 x LT
- FB = 4 x LT
- FD = 600 x LT
- FE = 10 x LT
- FT = (4 + 4 + 600 + 10) x LT = 50.07665 Hz

In addition to this, E-INK panels require a particular duty cycle for the GDCLK signal which is specified as gate-on-time and gate-off-time. The sum of gate-off and gate-on should equal LT and the value of gate-on can be programmed through HW_EPDC_TCE_TIMING2[GDCLK_HP], or GDCLK High-Pulse time.

24.3.4.4 Source Driver and Pixel Clock Configuration

The EPDC supports a number of source driver architectures. Each architecture forces a particular shift clock pixel rate.

In addition to this, the selection of the architecture requires a particular clock ratio between the internal pixel clock (pixclk) and external shift clock EPDC_SDCLK. The table below outlines the various configurations and clock ratios. It also lists the relevant register settings for this configuration.

Table 24-2. Interface Modes

High-Level Mode		HW_EPDC_CTRL [DUAL_SCAN]	EPDC_DATA pins used	HW_EPDC_FORMAT [TFT_PIXEL_FORMAT]	HW_EPDC_TCE_CTRL [PIXELS_PER_SDCLK]	HW_EPDC_TCE_CTRL [LVDS_MODE]	HW_EPDC_TCE_CTRL [DDR_MODE]	HW_EPDC_TCE_CTRL [SDDO_WIDTH]	Required PIXCLK (RATIO)
Single Scan	2bpp, 8-bit single-ended, SDR	0	[7:0]	2B[V]	FOUR	0	0	8BIT	SDCLK x 2
	2bpp, 16-bit single-ended, SDR	0	[15:0]	2B[V]	EIGHT	0	0	16BIT	SDCLK x 4
	2bpp, 8-bit, DDR (LVDS option)	0	[7:0],[15:8]	2B[V]	EIGHT	1|0	1	8BIT	SDCLK x 4

Table continues on the next page...

Table 24-2. Interface Modes (continued)

High-Level Mode		HW_EPDC_CTRL [DUAL_SCAN]	EPDC_DATA pins used	HW_EPDC_FORMAT [TFT_PIXEL_FORMAT]	HW_EPDC_TCE_CTRL [PIXELS_PER_SDCLK]	HW_EPDC_TCE_CTRL [LVDS_MODE]	HW_EPDC_TCE_CTRL [DDR_MODE]	HW_EPDC_TCE_CTRL [SDDO_WIDTH]	Required PIXCLK (RATIO)
	4bpp, 8-bit single-ended, SDR	0	[7:0]	4B[V]	TWO	0	0	8BIT	SDCLK x 2
	4bpp, 16-bit, single-ended SDR	0	[15:0]	4B[V]	FOUR	0	0	16BIT	SDCLK x 2
	4bpp, 8-bit, DDR (LVDS option)	0	[7:0],[15:8]	4B[V]	FOUR	1|0	1	8BIT	SDCLK x 4
	4bpp, 16-bit, single-ended, DDR	0	[15:0]	4B[V]	EIGHT	0	1	16BIT	SDCLK x 4
Dual Scan	2bpp, 8-bit single-ended, SDR	1	[7:0],[15:8]	2B[V]	FOUR	0	0	8BIT	SDCLK x 2
	2bpp, 8-bit, DDR	1	[7:0],[15:8]	2B[V]	EIGHT	0	1	8BIT	SDCLK x 4
	4bpp, 8-bit single-ended, SDR	1	[7:0],[15:8]	4B[V]	TWO	0	0	8BIT	SDCLK x 2
	4bpp, 8-bit, DDR	1	[7:0],[15:8]	4B[V]	FOUR	0	1	8BIT	SDCLK x 4

24.3.4.5 Initializing the Display

Because of the nature of the EPD technology, a typical scenario would involve the EPDC's working buffer (WB) being maintained in system memory. In such power states (where memory is maintained), it is not required to initialize the display (even if the EPD panel had been powered off for example).

In low-power modes where system memory is not maintained, there are two possible methods to initialize the display.

The first method includes initialization from an unknown state and the second mode simply requires software to load the last state of the WB into memory and configure the EPDC to point to it.

24.3.4.5.1 Reset/Clocks and Buffer Preparation

In cases in which the user wants to initialize the screen to a new known state, the following sequence should be followed. Assume that all relevant display power supplies are active.

First, the user must complete a soft reset sequence which includes enabling the main EPDC block-level clock gate. This sequence is described in the example code below. Setting SFTRST enables the CLKGATE. In order to correctly cycle reset values through pipeline registers, the CLKGATE bit stays asserted for some finite amount of time before it sets. After this time, it is safe to clear both the SFTRST and CLKGATE. The example below shows a typical use of these registers.

```
EPDC_CTRL_SET(BM_EPDC_CTRL_SFTRST);
while (!EPDC_CTRL.B.CLKGATE);
EPDC_CTRL_CLR(BM_EPDC_CTRL_SFTRST | BM_EPDC_CTRL_CLKGATE);
while (EPDC_CTRL_RD() & (BM_EPDC_CTRL_SFTRST | BM_EPDC_CTRL_CLKGATE));
```

Before any display update is performed, the waveform address and working buffer (WB) pointers must be set (both must be aligned to a 64-bit double long word address):

- EPDC_WVADDR: Must point to the base address of the waveform data (managed by the i.MX driver)
- EPDC_WB_ADDR: An area of memory must be assigned for the EPDC's working buffer (WB). Once assigned, this memory space should be reserved purely for the use of the EPDC. The requirements for the memory allocation are described in Memory Requirements.

After this, the various panel configuration parameters must be set including resolutions, source and gate driver formats, TFT and buffer pixel formats.

24.3.4.5.2 Performing an Initialization Display Update

Once all the panel timing and format parameters are defined, a command sequence must be sent to the EPDC with the properties found here.

- Update size must be full screen resolution
- HW_EPDC_UPD_CTRL[UPDATE_MODE] must be set to FULL
- HW_EPDC_UPD_CTRL[WAVEFORM_MODE] must be set to the INIT waveform (typically 0x00)

- HW_EPDC_UPD_CTRL[USE_FIXED] must be set
- HW_EPDC_UPD_FIXED[FIXNP_EN], FIXCP_EN must be set to 1, and FIXNP and FIXCP must be set to 0xFF

The use of HW_EPDC_UPD_FIXED allows the EPDC working buffer to be primed to a state that matches the result of the initialization waveform.

If the application loads the previous WB of the EPDC, these steps are not necessary.

24.3.5 Dual-Scan Configuration

The EPDC supports ultra-high resolution/refresh rate EPD panels.

At certain resolutions/refresh rates, there exist electrical limitations of the TFT devices and TFT driver ICs such as source drivers. In order to support driving such displays, the EPDC supports the dual-scan driver method.

The specific method involves driving a TFT configuration where there are two sets of source drivers. All source driver control signals are expected to be shared across all source drivers (except the SDCE signals unless a unified CE is presented by the panel).

In this mode, EPDC_DATA[7:0] is reserved to drive the upper half of the display and EPDC_DATA[15:8] is reserved to drive the lower half of the panel. The update and working buffers do not have to be aware of the dual-scan operation because the MBM knows how to access the WB for refresh operations based on the settings of the EPDC_TCE_CTRL[SCAN_DIR_0,1]. The EPDC supports four scan configurations for dual-scan which are shown in the following figures. Each figure shows an example configuration (that is to say, the EPDC is not limited to the support of only 2 source drivers per side).

The dual-scan drive method allows the SDCLK frequency to be halved relative to a single-scan method for the same resolution and refresh rate. In addition to this, each TFT is only charged in half frequency (for any given SDCLK cycle, twice as many pixels are being driven per frame scan).

Because all 16 DATA signals must be utilized for dual-scan, there is no support for the LVDS_MODE. The supported source-driver modes for dual scan are shown in Table 24-2. For the 2bpp DDR mode, the EPDC is capable of generating a total of 16 pixels per SDCLK cycle.

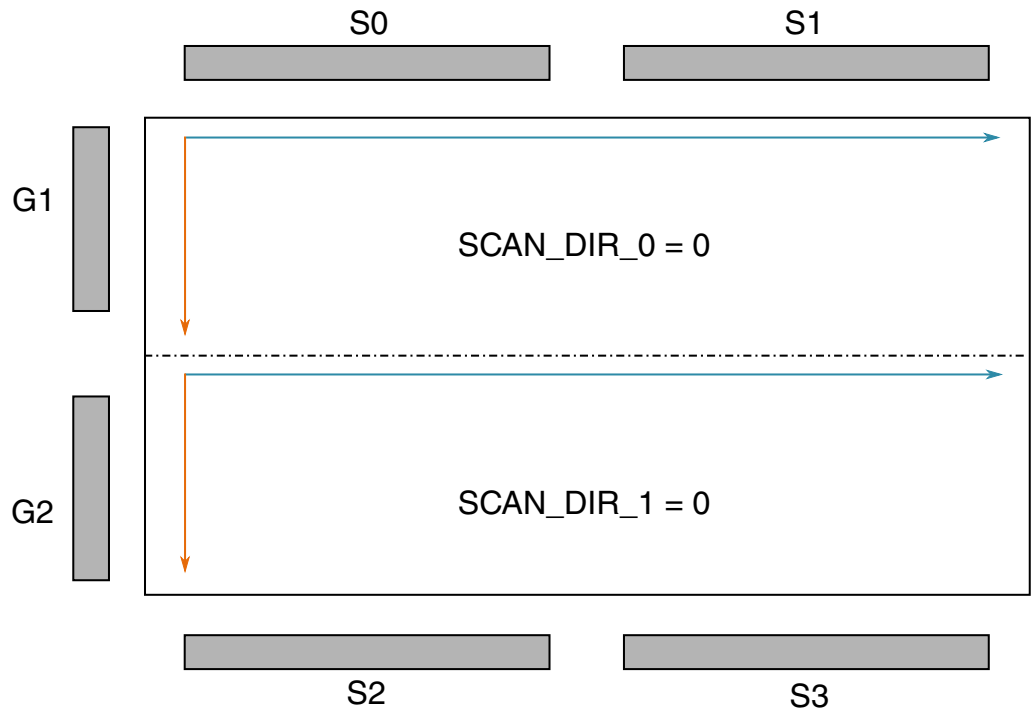


Figure 24-7. Dual-Scan Method (Scan = 00)

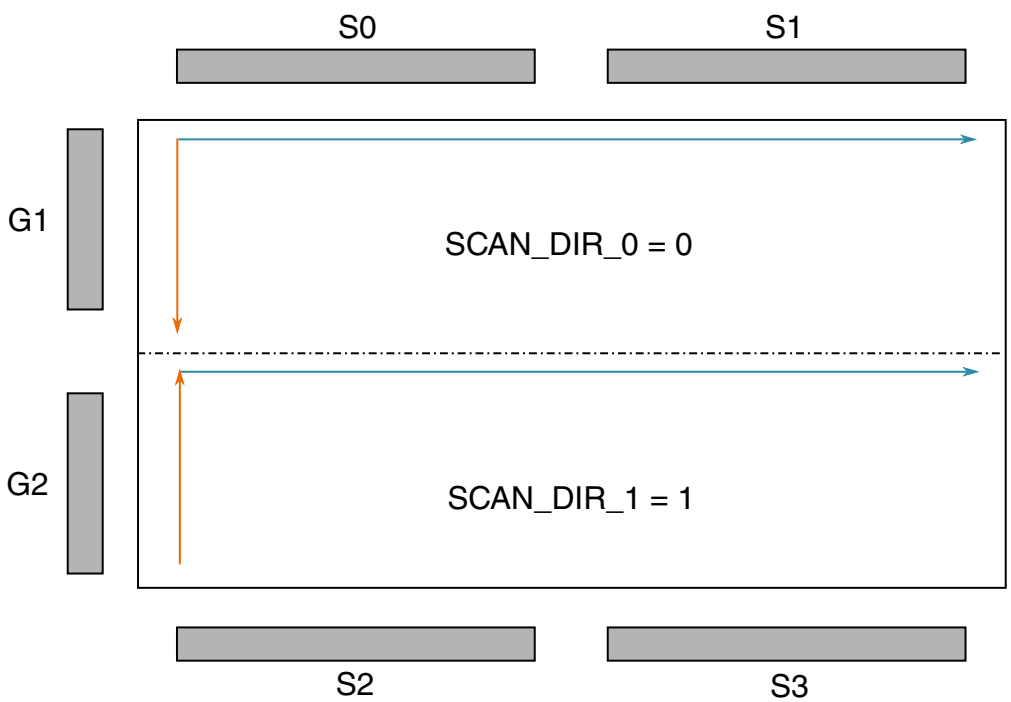


Figure 24-8. Dual-Scan Method (Scan = 01)

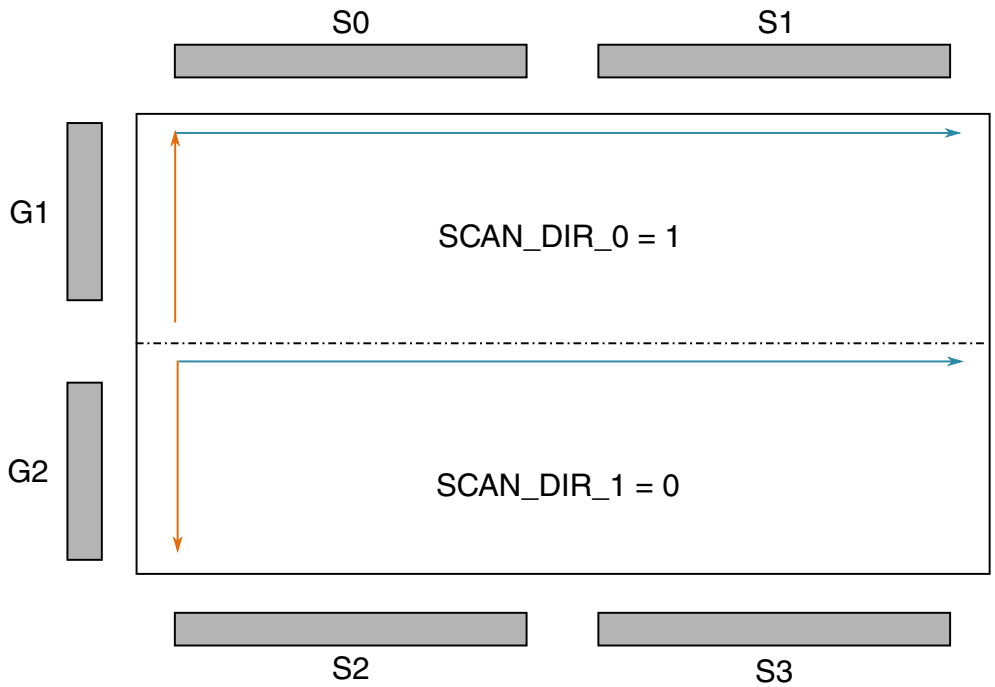


Figure 24-9. Dual-Scan Method (Scan = 10)

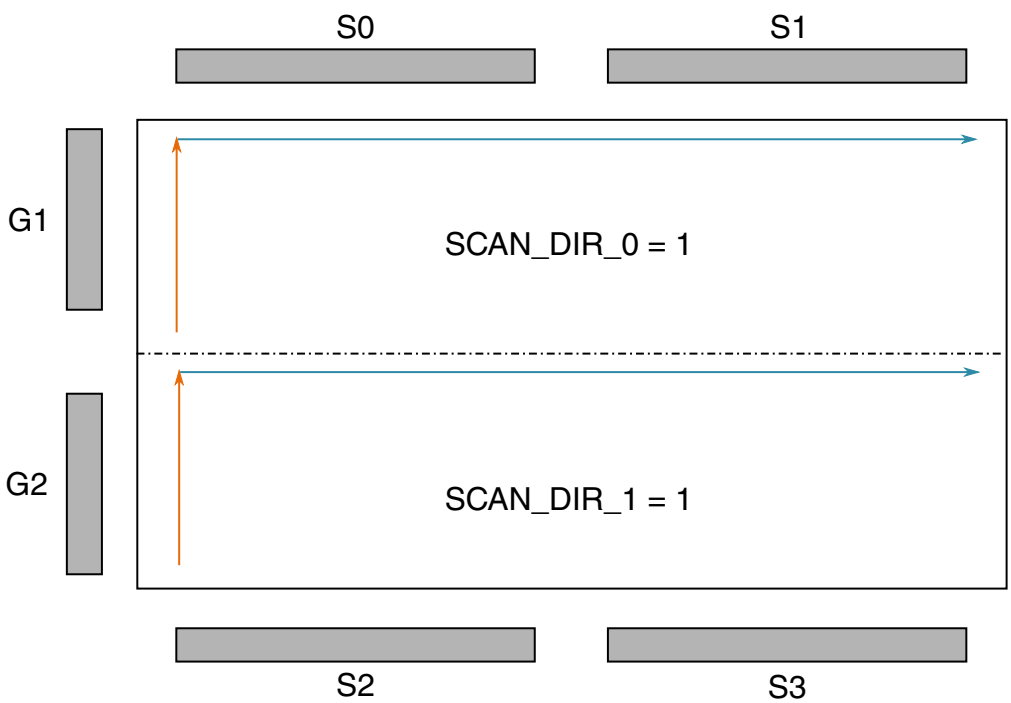


Figure 24-10. Dual-Scan Method (Scan = 11)

24.3.6 Update Buffer Analysis Functions

EPDC can perform several analyses on the update buffer.

The information collected during processing will be reported to the register after update buffer processing is complete.

- **Collision Rectangle Detection**

  For a collided update buffer, this function will report a minimal rectangle that can cover all collided pixels.

  When a collision occurs, it's possible that not all, but only part of that update buffer generate a collision, so only the pixels inside a minimal collision rectangle need to be re-submited.

  Details of minimal collision rectangle reported in UPD_COL_CORD/ UPD_COL_SIZE.

- **Grey Level Detection (histogram)**

  This function reports the minimal grey level that covers all pixels and which must be updated. This histogram differs from that of PXP; here we ignore those pixels which need no update, while the PXP histogram is calculated on the whole update buffer. The histogram feature is capable of reporting a histogram for 1, 2, 4, 8, or 16 grey levels. The valid grey levels value should be programmed into the HIST(1/2/4/8/16)_PARAM registers before using this functionality.

- **Dry-Run Mode**

  Dry-Run mode allows the user to run an update for the purpose of obtaining information without performing any real action. This mode is enabled by setting UPD_CTRL[DRY_RUN] = 1 when writing UPD_CTRL register.

  In this mode, Update Buffer will be read and checked against the current Working Buffer. As with a normal update, an interrupt will be generated when the Working Buffer processing completes.

  Several pieces of information about the update can then be acquired from the EPDC registers: whether a collision occurred, the minimal collision area, and histogram data. However, the result (Next Pixel, LUT info) will not be written back into the Working Buffer for further panel scan operation, nor will the LUT waveform be loaded.

The LUT resource specified by UPD_CTRL[LUT_SEL] is not used here; the user can start this dry-run update when no LUTs are available.

24.3.7 Waveform Mode Selection (AUTOWV)

This function is provided to support waveform selection based on the grey level (histogram) information collected during the update buffer processing stage.

Then AUTOWV LUT is provided to hold a mapping between the grey level of NP(pixel value from update buffer) and optimal waveform mode. This LUT mapping should be programmed before using this function by writing into the AUTOWV_LUT register with AUTOWV_LUT[ADDR]="grey level", AUTOWV_LUT[DATA]="waveform mode" for each possible grey level reported by the EPDC histogram.

After AUTOWV_LUT is programmed, AUTOWV mode can be enabled by setting UPD_CTRL[AUTOWV]=1. When enabling AUTOWV, the waveform mode written into UPD_CTRL[WAVEFORM_MODE] will be ignored and the loading waveform will initially be put on hold.

After the Update Buffer processing completes, EPDC reports the minimal grey level, retrieves the mapped waveform mode from AUTOWV LUT, and writes it back into UPD_CTRL[WAVEFORM_MODE].

If the user also sets UPD_CTRL[DRY_RUN], EPDC will be finished at this point.

Otherwise, EPDC continues with the following, depending upon the UPD_CTRL[AUTOWV_PAUSE] setting:

1. UPD_CTRL[AUTOWV_PAUSE] = 0, AUTO MODE)

   EPDC will begin waveform loading immediately without any software interaction

2. UPD_CTRL[AUTOWV_PAUSE] = 1, MANUAL MODE)

   EPDC waits for software to decide whether the selected waveform is ok, then if necessary software can use some other criteria to select a waveform mode. After software selects the waveform mode, it writes again into UPD_CTRL with final waveform mode, and waveform loading then panel scan will start.

24.3.8 Panel Interface Generator (Pigeon Mode)

There are seventeen panel interface signal outputs (DATA/SDCLK not included), each of them with dedicated timing purpose as default. This is called "Legacy Mode".

Pigeon Mode is a timing mode which can be independently enabled on any of the seventeen timing signals, with a unified flexible configuration. Signals within pigeon mode can be programmed into any supported signals, and are interchangeable.

The following are the legecy timing signals which support pigeon mode:

- PIGEON[00] - SDCE0
- PIGEON[01] - SDCE1
- PIGEON[02] - SDCE2
- PIGEON[03] - SDCE3
- PIGEON[04] - SDCE4
- PIGEON[05] - SDCE5
- PIGEON[06] - SDCE6
- PIGEON[07] - SDCE7
- PIGEON[08] - SDCE8
- PIGEON[09] - SDCE9
- PIGEON[10] - SDOE
- PIGEON[11] - SDL3
- PIGEON[12] - SDOEZ
- PIGEON[13] - SDOED
- PIGEON[14] - GDSP
- PIGEON[15] - GDOE
- PIGEON[16] - GDCLK

Working Theory

Each pigeon signal has one local counter with a configurable start point and incremental condition. It will be compared to configuration register value for signal assertion/de-assertion control, plus delta offset for data alignment and other options like polarity/logic operation. A detailed running scenario is as follows:

1. Start local counter on the MASK rising edge (reference point/start point)
2. Increment on event selected through INC_SEL
3. Count and match SET_CNT: assert signal, reset counter

   (SET_CNT==0 means assert immediately on MASK's rising edge)

4. Count and match CLR_CNT: de-assert signal, stop counter

   (CLR_CNT==0 means de-assert on MASK's falling edge)

NOTE

When local counter is running, further changes to MASK are not cared unless CLR_CNT==0

MASK - start point for local counter

Created using any combination of below options (ANDed)

1. STATE_MASK = (FS|FB|FD|FE) AND (LS|LB|LD|LE)

   8 bits to select in which vertical/horizontal state your counter starts ticking. This is the most common use-case because timing signals generally relate to scan states.

   For example, for a line timing signal start on Line Begin phase during Frame Begin/ Frame Data lines, use the configuration below:

Table 24-3. STATE_MASK combinations

STATE_MASK	LS	LB =1	LD	LE
FS				
FB =1		X		
FD =1		X		
FE				

2. MASK_CNT/MASK_CNT_SEL(global counter)

   Sometimes a more accurate reference point is required, such as" line 20 in a frame", or "line 12 in Frame Begin state" or "pixel 23 in LD phase". For such use-cases, several Global Counters shared by all pigeons are provided. Global counter type (line counter/frame counter/state counter, etc.) is selected using MASK_CNT_SEL, when global counter matches the MASK_CNT value. The pigeon local counter will start ticking.
3. Use another pigeon signal as mask (SIG_LOGIC=MASK)

   For some tightly coupled signals it is possible to use one as a reference to generate another.

INC_SEL- select local counter tick event

1. pclk - pixel clk
2. line - line start pulse
3. frame - frame start pulse
4. another - another pigeon signal

OFFSET- offset on pclk basis

Some signals need to come out slightly earlier or later than programmed. For example, the CE type signal usually aligns with the Line Data phase, but some panels need it as one cycle pulse before Line Data. The user can set OFFSET to a negative value to achieve this.

Global Counters (selectable through MASK_CNT_SEL)

- HCNT / VCNT

  normal pclk counter / line counter
- HSTATE_*CNT* / VSTATE_*CNT*

  similar to above, but reset when state changes
- HSTATE_*CYCLE* / VSTATE_*CYCLE*

  (see figure below for definition of CYCLE/PERIOD/CNT)

  Some panels have multiple Gate Drivers/Source Drivers, so Frame Data / Line Data state may be further split to match each driver, and signals such as CE[n] are only valid during part n of Line Data. For such signals, use HW_EPDC_PIGEON_CTRL.*_PERIOD to specify PERIOD where CNT is reset and CYCLE is included. Then the user can select CYCLE as MASK_CNT to generate mask for CE[n].
- FRAME_*CNT* / FRAME_*CYCLE* (only for frame-crossing signals)

  frame cycle counter doesn't have a reset condition; use HW_EPDC_PIGEON_CTRL1.FRAME_CNT_CYCLES to reset it.

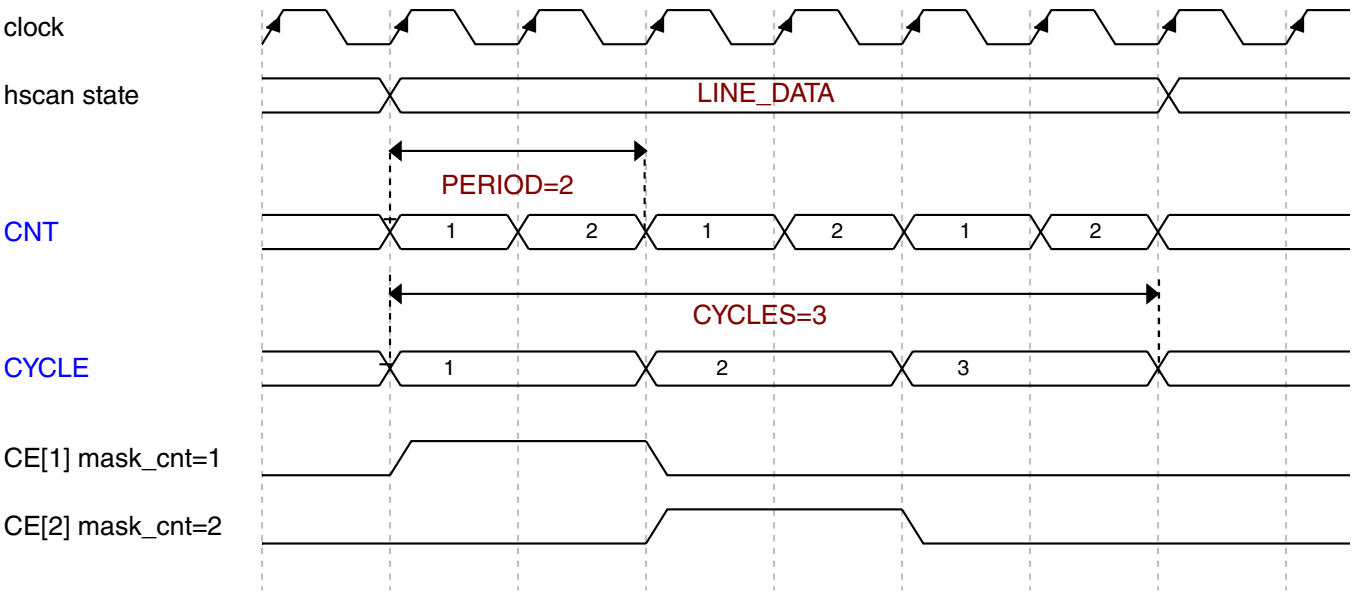


Figure 24-11. Definition of CNT, CYCLE, PERIOD

The following are register settings for the figure above:

- HW_EPDC_PIGEON_CTRL0.LD_PERIOD=2
- MASK_CNT_SEL = 1 // HSTATE_CYCLE
- MASK_CNT = 1 // CE[1]
- MASK_CNT = 2 // CE[2]

24.3.9 Display Update Programming

The EPDC is designed to communicate with the kernel driver through the interrupts and status registers.

The driver entry is always assumed to occur as a result of an interrupt.

24.3.9.1 Initiating a Display Update

The typical flow for performing a display update involves the following:

- EPDC_STATUS[WB_BUSY] must be 0. The EPDC cannot process new updates if it is currently processing an update.
- EPDC_STATUS[LUTS_BUSY] must be 0. The EPDC cannot accept new updates if all LUT resources are currently active.

Assuming both conditions above are met, an update request is programmed in the following manner (it is important to note that the last step must be writing to EPDC_UPD_CTRL which initiates the display update).

The following steps can be performed before WB_BUSY and LUTS_BUSY are clear:

- At this time it is assumed that the correct temperature is selected and written to EPDC_TEMP.
- EPDC_UPD_ADDR must be set to the 8-bit buffer for the update. This buffer must be aligned to a 64-bit word address. In addition, each line address must begin at a 64-bit word address. Both criteria can be met by utilizing the PXP pixel processing engine which always aligns lines at a 64-bit raster (see Memory Requirements for details).
- EPDC_UPD_CORD defines the insertion co-ordinate into the panel resolution.
- EPDC_UPD_SIZE defines the dimension of the update (pixels).
- EPDC_UPD_FIXED is used for panel initialization or rectangular fill operations.

Writing the EPDC_UPD_CTRL register initiates a display update:

- USE_FIXED-This field is set when EPDC_UPD_FIXED is used.
- LUT_SEL-The driver should read EPDC_STATUS_NEXLUT[NEXT_LUT] (qualified by NEXT_LUT_VALID) to select an available LUT to assign this update.
- WAVEFORM_MODE selects the E-INK waveform mode (for example, INIT, DU, GC16 and so on). The number corresponds to the waveform number in the waveform specification document.
- UPDATE_MODE selects one of two enumerated updated modes.
  - FULL-In this mode, all pixels defined in the update rectangle have the waveform applied.
  - PARTIAL-In this mode, the EPDC shall only apply the waveform to pixels which are changing, otherwise the EPDC_FORMAT[DEFAULT_TFT_PIXEL] is applied to the pixels which are not changing.

24.3.9.2 Update Processing and Collisions

After the update is initiated (by writing EPDC_UPD_CTRL), the EPDC performs update-buffer processing. This involves processing this update region into its working buffer (WB).

During this time, the EPDC also performs collision detection. At the end of the WB processing, the EPDC asserts the WB_CMPLT_IRQ interrupt. If a collision is detected, the LUT_COL_IRQ interrupt status bit will also be asserted.

The EPDC performs image updates to the EPD by applying waveforms to individual pixels. Waveforms are applied to groups of pixels (ranging from a single pixel to the entire display). Each individual update request is bound to a rectangle. Each pixel within this rectangle has a waveform applied to it using the waveform and update mode specified in EPDC_UPD_CTRL. After this process is initiated, a waveform takes an amount of time to complete which is usually determined by the specifications of the waveform mode. For proper display operation and optimum performance, a waveform must not be interrupted once it has begun.

A collision is defined simply as the interruption of the application of waveform to a pixel or group of pixels. As such, it must be avoided. Because the waveforms take a finite amount of time to complete, the EPDC provides a mechanism to automatically manage collisions, which allows the application to perform multiple updates resulting in a richer user experience without having to be cognizant of the panel and waveform characteristics.

Depending on the update mode (FULL or PARTIAL) a collision is defined as follows:

- For a full-update mode, a collision is defined as the temporal and spatial union of rectangles pertaining to active unique update requests. An active update is one which is still in the process of being updated (that is, a waveform is being applied).
- For a partial-update mode, a collision is defined as the temporal, spatial and differential union of pixels pertaining to unique update requests which are active. The difference between a full-updated and a partial-updated collision is that in the latter case, the collision state occurs only if the next-pixel states differ between the updates. Thus, two partial-update rectangles may intersect, but if all of the pixels in the intersecting update regions are being updated to the same gray-state, the original update and associated LUT/waveform are not disturbed and the intersection does not constitute a collision.
- It should be noted that collisions may be a one-to-many effect. In the case of partial updates, there may be a spatial and temporal union of rectangles with no differential pixel-level union. A subsequent update however might form such a pixel-level differential union to all active updates. Since each update pertains to an active LUT, these many collided updates are refered to as victim LUTs. The update/LUT that caused the collision is referred to as the aggressor LUT.

An example of a partial-update mode collision is shown in the following figure. Each update encompasses the outer rectangle which meets the criteria of the spatial union. The red areas show the pixels which are currently transitioning to gray (from update 1) but need to transition to white (in update 2). These pixels meet the criteria of the differential union and lastly in the temporal domain, the period of time where LUT0 and LUT1 overlap is the temporal union. The pixels that did not change (grey and green areas in the LUT assignment box) are not part of the collision area. In the case of the gray

overlapping pixels, LUT0 can continue to update. In the case of the white pixels (which were previously inactive) a new LUT (if available can be assigned) and forms the green region. Note that a subsequent update could potentially collide with the green pixels in update 2.

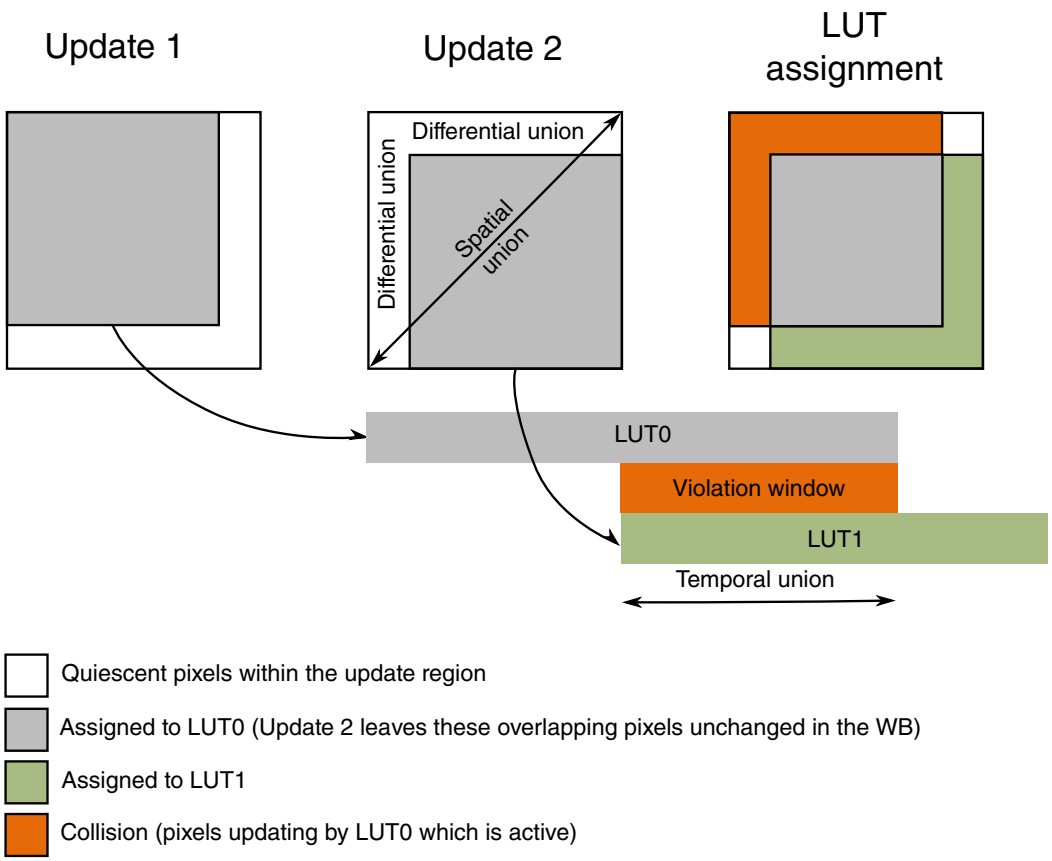


Figure 24-12. Partial Update Collision Handling

NOTE

For FULL update modes, a collision occurs at any intersection of active rectangles/LUTs because in FULL update mode, all pixels within the update rectangle have the waveform applied (even if pixel values are not changing).

It can be seen that when collision is detected (during the WB update process), the EPDC will "block" the collided pixels being updated, and any pixels which are not collided will be assigned to the requested LUT process. When a collision occurs, the EPDC will raise the LUT_COL_IRQ interrupt status bit (this will always happen in conjunction with WB_CMPLT_IRQ).

Because the driver is always entered via an interrupt, when the EPDC interrupt fires, driver should not only check for WB_CMPLT_IRQ but also for the LUT_COL_IRQ status bit within the EPDC_IRQ register. If LUT_COL_IRQ is set, this means that a

collision has occurred. If this is the case the driver should check the value of the EPDC_STATUS_COL register. This register contains one bit for each LUT that has been collided with, that is, it holds the vector of the victim LUTs as a result of the last update LUT. Since the driver knows the value of the LUT of the last update, it can store this update LUT value along with this associated victim LUT vector in a collision list.

Because the EPDC can drive up to 16 concurrent updates (which can be overlapping when using PARTIAL mode updates), a unique interrupt status bit is provided per LUT so that when an update completes (physically completes the waveform), the SW driver can know which LUT completed. In addition to the LUT completion interrupts (which are mapped into the EPDC_IRQ[LUTn_CMPLT_IRQ] interrupt status bits), the current state of each LUT can also be read from the EPDC_STATUS_LUTS status register.

The driver can use the LUT status interrupts in conjunction with EPDC_STATUS_LUTS to perform regular checks on the completion of victim LUTs against the vectors stored for each colliding LUT in the collision list. The driver may also choose to serialize updates such that it might wait until all victim LUTs have completed before re-issuing the colliding update.

The figure below shows a more extended sequence of a collision. In this case there are a series of updates coming from the application layer. They are A, B, C and D. Updates A, B, and C are set without any collision. When update D is requested a collision is detected. Using the interrupt status collision registers, the SW-driver stores D in a collision-list. For that list entry it also store B which is the victim LUT. It uses this information to check when a LUT completes to see if the victim LUTs for any update in the collision list have completed. If they have completed (through a LUT completion interrupt), the SW-driver then sends this update (D in this case) again to the EPDC. It should be noted that from the application perspective it appears that all updates A, B, C and D have been accepted, that is, the collision handling is completely hidden from user-space. Using the collision detection and LUT status interrupts, the application (in conjunction with the driver) could also choose to completely serialize the requests (such that between the first and second D update, no other update is accepted).

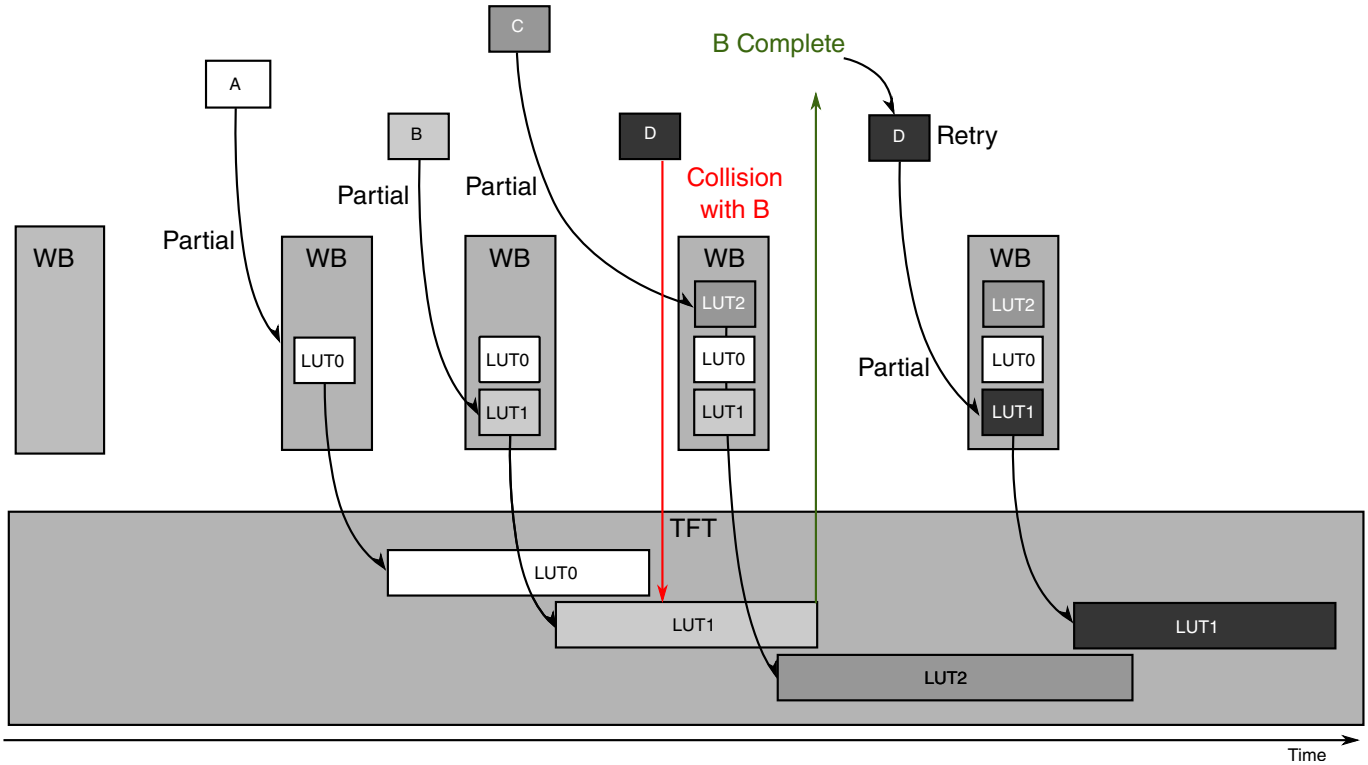


Figure 24-13. Collision Sequence

24.3.9.3 Multiple Update Flow

In most applications, such as e-book readers, besides full-screen page turns, user interfaces require multiple updates to be performed.

Examples of this include cursors, pen input, drop-down menus (with highlighting) and other movable graphical objects. Each update takes such a limited amount of time to complete (up to 1 second depending on the waveform mode) that it would be too prohibitive to only process updates serially.

As described in Initiating a Display Update, there are two conditions that must be met before the SW driver can issue a new update request:

- The Working buffer (WB) update process must be complete.
- There must be an available LUT.

Assuming a free LUT is available, the EPDC is designed to have the capability to start new updates at each TFT frame scan (even being able to commence multiple updates within a frame scan blanking period). The general flow for sequential updates (concurrent or serialized) is shown in the following figure.

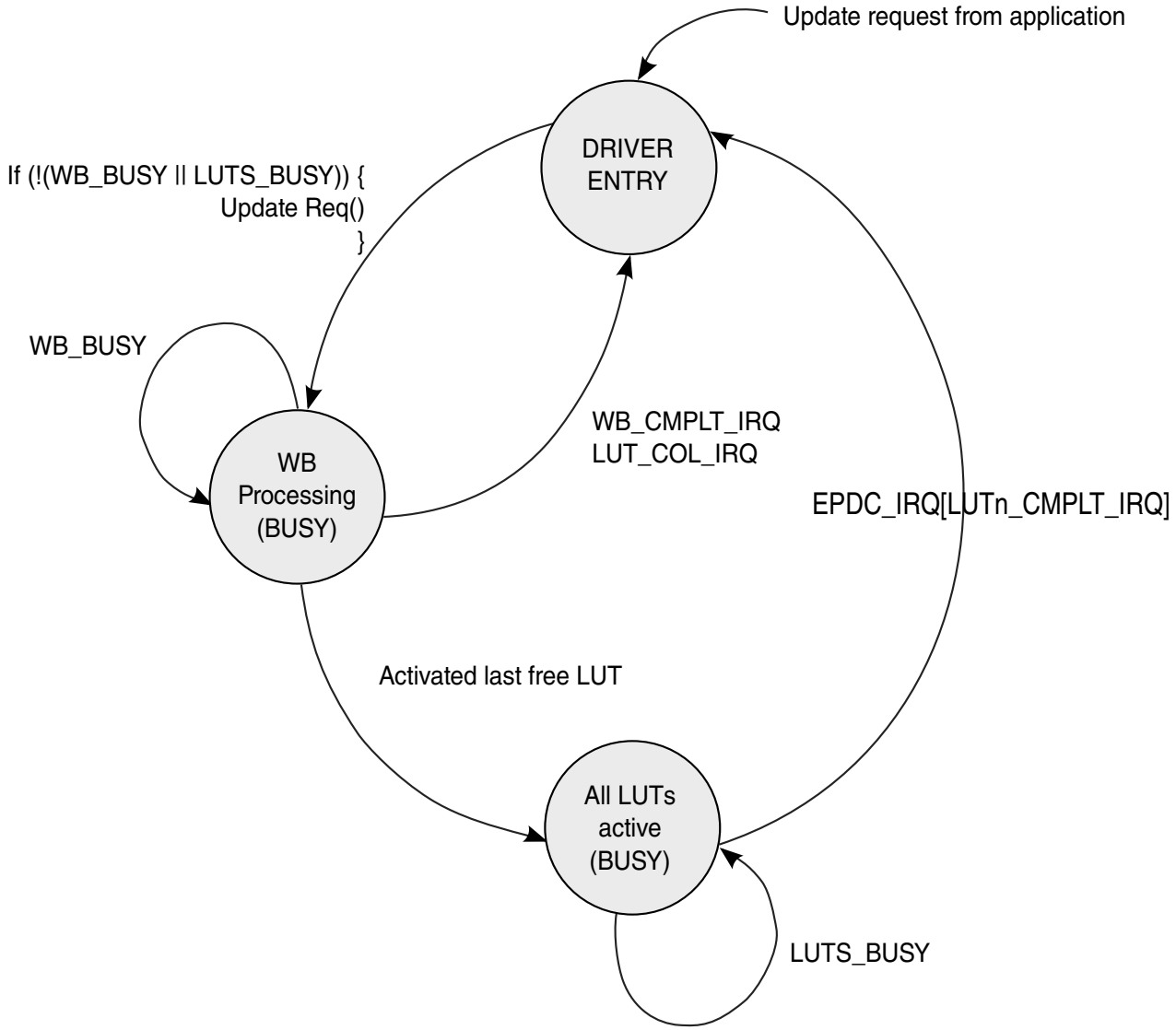


Figure 24-14. Driver Update Flow

This flow assumes that the driver is entered via interrupts. The two busy states are shown to illustrate the conditions under which the driver will not issue new updates. In general, the driver should be throttled by the WB_CMPLT_IRQ interrupt. Because the EPDC can support up to 64 concurrent updates, it is possible to re-enter the update-request (show as Update_Req() in the figure above) any-time that the WB is not busy (even though LUTs are active and the display is being updated). It is possible that when there are 63 LUTs currently active and a new update is sent, there will be a WB_CMPLT_IRQ interrupt issued, but the driver at that time would be in a busy state (because LUTS_BUSY is signaled). The driver will be re-entered on the next LUTn_CMPLT_IRQ interrupt during which time it can issue a new update (since WB_BUSY and LUTS_BUSY status bits are low).

The update-processing process (which is controlled by the MBM module) is asynchronous to the actual TFT refresh operations (which carry out the physical screen updates), so there is no particular relationship between the WB_BUSY status and the state of the screen update. In contrast, the LUTn_CMPLT_IRQ interrupt status and EPDC_STATUS_LUTS provide the status of the physical waveform update. This means that when LUTn_CMPLT_IRQ interrupt fires, it means that the last physical TFT frame scan just completed for that update/LUT.

The EPDC implements proprietary methods to efficiently process the WB such that the WB processing time (the time from the SW driver writing EPDC_UPD_CTRL to EPDC_IRQ[WB_CMPLT_IRQ] fires), is dependent on the size of the update rectangle. This allows smaller updates to be processed faster. In most EPD applications, this is ideal because it's typically smaller sprites that require the fastest performance. In addition, the EPDC implements methods to allow store pending updates and activate them on the next available frame-scan. For example, if the TFT scanning is currently in the active region and during this time a series of small update requests are sent, the EPDC can begin all of those updates on the next available blanking period.

The figure below shows an example of two larger update requests, one to LUT4 and one to LUT5. These update requests are shown in the context of active LUTs and thus active frame scan times. The time when both WB_BUSY and LUT_BUSY is high are defined as "blocking" because no new updates can be accepted. It should also be noted that new updates are physically commenced by the EPDC on blanking periods. From a SW programming model perspective, the EPDC_STATUS_LUTS[LUTn_STATUS] is activated immediately upon the update request being written to the EPDC. The physical frame-scanning for this particular update does not being commenced until the next available frame scan.

It should be noted that updates and WB processing can actually occur during the blanking period (or can begin in the active scan time and continue into the blanking period). For this reason, the EPDC provides some tuning controls via EPDC_TCE_CTRL[VSCAN_HOLDOFF] such that more time can be given to finish processing update request and less time "locking down" the set of LUTs that will be activated for the upcoming active scan. The EPDC will start to pre-fill its refresh pixel FIFOs after the VSCAN_HOLDOFF period. Its possible to tune this value to allow the minimum time for the pixel FIFO pre-fill operation (making sure that refresh under-runs do not occur). If a pixel-FIFO under-run occurs, EPDC_IRQ[TCE_UNDERRUN_IRQ] will fire.

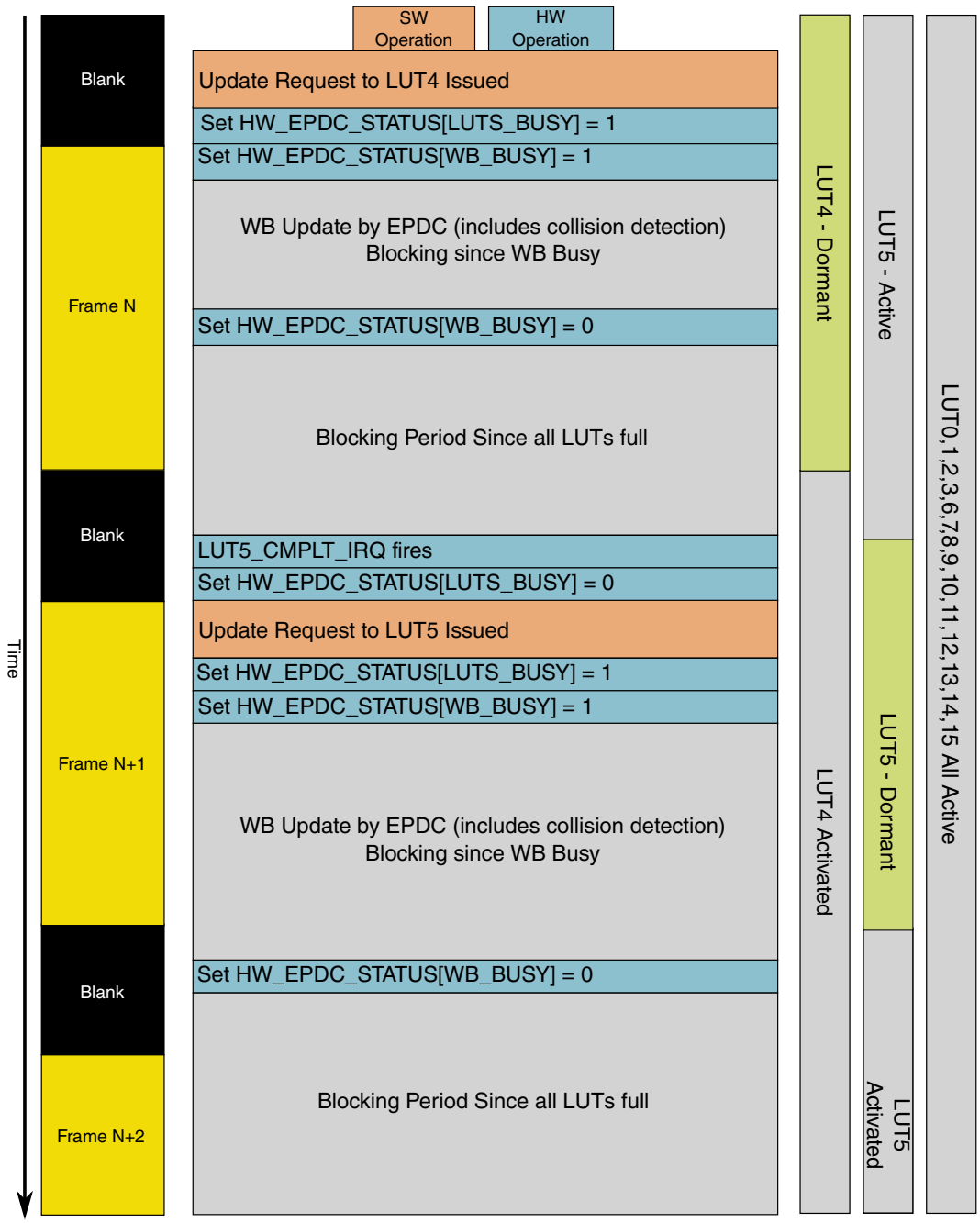


Figure 24-15. Temporal Flow of Update Processing

24.3.10 Architectural Clock Gating (Low Power Mode)

As with most modern IP, the EPDC contains low-level hardware controlled automatic clock gating throughout the design.

These clock gating elements typically gate clocks locally for individual or groups of sequential elements (flip flops) when these registers are not currently active.

This tier of clock gating is useful for reducing power during active operation of the EPDC.

Most e-paper applications utilizing EPDs involve the processor and display controller to be a low-power inactive state in most of the time. This is because EPDs only require refresh operations when the display content is being updated. As such, in between update operations, the EPDC provides a module level clock-gating feature that can be controlled by the SW driver as part of a higher-level power management solution.

The EPDC module can have all its clocks completely gated-off by the setting of EPDC_CTRL[CLKGATE]. There are a couple of considerations that the SW driver must adhere to before placing the EPDC into this low power state:

- All active LUTs (updates) must complete
- Any TFT operations must be complete (including any final blanking operations)

In order to maximize display update frequency (the ability accept new updates during each frame scan), the LUT completion interrupts fire on the end of the last frame's active scan period for that update. This means that the TCE still must compete the final FRAME_END blanking time to guarantee coherency of its state machines with the panel state (receiving the final LUT completion IRQ is not enough to determine that the EPDC can have all of its clocks shut down). Based on this, the EPDC provides the necessary interrupt and status bits to allow the driver to correctly perform the clock-gating operation by using the following method:

- Once the final EPDC_IRQ[LUTn_CMPLT_IRQ] is reached, the driver should enable the TCE_IDLE interrupt mask bit via EPDC_IRQ_MASK[TCE_IDLE_IRQ_EN], unmask other interrupts and return.
- Once the final blanking time of the last LUT (FRAME_END time) is completed and the TCE reaches its idle state, the EPDC will assert the EPDC_IRQ[TCE_IDLE_IRQ] interrupt which will cause the driver to re-enter. At this point, the SW knows that the EPDC is in a completely idle state and can perform a set on EPDC_CTRL[CLKGATE].

Alternative implementations of the driver could simply involve polling for the EPDC_IRQ[TCE_IDLE_IRQ] interrupt status bit (which operates regardless of the interrupt being masked or not) before setting the clock gate (this prevents two iterations of interrupt based driver-entry, but requires a wait loop in the interrupt handler which might be undesirable).

It should be noted that the time interval between the last LUTs LUTn_CMPLT_IRQ time and TCE_IDLE_IRQ interrupt is equal to the FRAME_END time.

When the application is ready to send more updates, the driver can immediately re-enable the EPDC by un-gating the module-level clock by clearing the EPDC_CTRL[CLKGATE] field. This un-gating of the clock is instant and the driver is free to immediately send update requests.

24.3.11 Performance Tuning and Considerations

The EPDC is designed to meet the performance requirements of the most demanding e-paper applications. These performance targets must be met in the context of highly integrated SoCs.

In such a context, system memory is typically shared across multiple masters including the Arm platform. Because of this, there is no guarantee of latency from system memory. EPDC provides a number of mechanisms to deal with this as well as some ability to tune various parameters allowing the user to trade-off between update processing performance and refresh functionality.

24.3.11.1 Memory and Bus Bandwidth Requirements

The EPDC MBM module contains up to 6 internal AXI requesting functions (5 read and 1 write). These are arbitrated internally before being sent to the SoC memory system.

The EPDC interfaces to the SoC bus system via a dual-channel 64-bit AXI bus master port. The following table shows the various internal requesters and their properties. Note that there are two round-robin (RR) loops. Because there is only one write source, and the EPDC supports dual-channel AXI, there is no arbitration on the WB write operations.

Table 24-4. EPDC Bus Requestor Profiles

Operation	Priority	Burst Length (x8 Bytes)	Maximum Outstanding Transactions	FIFO size
Waveform LUT read	0	8	2	16x68
WB pixel read FIFO0 (refresh)	RR 1	8 or 16	8	256x64
WB pixel read FIFO1 (refresh)	RR 1	8 or 16	8	256x64
WB read for update processing	RR 2	8	2	16x64
WB write for update processing	N/A	8	2	16x64
UPD read for update processing	RR 2	8	2	16x64

As described in Memory Requirements, it can be seen that all lines in a buffer are padded to the nearest 8 byte boundary. This is important to be noted for the bandwidth calculations. The maximum bandwidth scenario involves processing a full-screen update

while performing refresh operations, but refresh operations always have the highest priority. Based on this, the memory bandwidth required by the EPDC is a function of the refresh pixel rate which in turn is a function of resolution, frame-rate and blanking/active time.

The refresh bandwidth can be calculated as follows:

Active Time = Active time of the frame-scan time as a fraction (for example, 0.8)

Frame Rate = TFT frame refresh rate (Hz) (for example, 106 Hz)

Bandwidth (MB/s) = roundup4(EPDC_RES[HORIZONTAL]) x EPDC_RES[VERTICAL] x Frame Rate x (1/Active Time) x 2 x$(1/1024^2)$

For example, a panel with resolution 2048 x 1536 with 106 Hz refresh and estimated 80% active time, 20% blanking time, the refresh bandwidth is:

BW (MB/S) = 2048 x1536 x 106 x (1/0.8) x 2 x $(1/1024^2)$ = 795 MB/s

(Note that 2048 mod 4 = 0.)

The WB requires 2 bytes per pixel.

Remaining bandwidth is used for other operations. Because update processing operations are not real-time (they do not necessarily have to occur at the refresh frame-rate), the time required to perform the update can be calculated as a function of the available bandwidth (actual bandwidth minus refresh bandwidth) and the update size.

24.3.11.2 Pixel Latency FIFO

The EPDC contains one working buffer latency FIFO which is used to load working buffer pixel data which is used by the TCE to perform the panel refresh operations.

The FIFO is sized at 1024 pixels each in order to provide significant system memory latency tolerance. The pixel FIFO is dedicated for the main screen and the second is in dual-scan cases for the second half of the screen.

Under no circumstance should the EPDC pixel FIFO reach an under-run condition. The EPDC provides an interrupt status bit to flag such a condition (TCE_UNDERRUN_IRQ). During development, this interrupt must be enabled. The pixel values stored in the FIFO are using by the TCE to perform look-up operations (from the LUTs) to generate TFT voltage control pixels. If an under-run occurs, the result will be unknown data being used as the source for the look-ups, which can damage the panel.

24.3.11.3 Basic Watermarking Control

The EPDC provides a basic pixel pre-fetching control mechanism that is applied to the beginning of a new update (from an idle state). This control is provided by EPDC_FIFOCTRL[FIFO_INIT_LEVEL].

The control allows the value to be set between 0-255, with a reset value of 128. Because the internal width of the FIFO holds 4-pixels per entry, the actual number of pixels is (FIFO_INIT_LEVEL+1) x 4.

The EPDC uses this value to pre-fill the pixel latency FIFO(s) to this level before the TCE commences the refresh operations which drive the update to the panel. This control has no effect on the FIFO(s) when the EPDC is performing sequential frame-scans.

It must be noted that this value must be set less than (EPDC_RES[VERTICAL] x EPDC_RE[HORIZONTAL]) /4.

24.3.11.4 Update/Refresh Tuning (VSCAN_HOLDOFF)

During active frame-scan time, the EPDC provides the ability to process and display new update requests, because all pixels in the WB pertaining to a particular update are bound to a single LUT.

Until all pixels pertaining to that update have been processed in the WB, its LUT cannot be activated. As shown in Figure 24-15, the blanking period provides time to pre-load the pixel FIFOs and for completing WB processing updates and loading the associated frame of waveform data into LUT memories.

The EPDC_TCE_CTRL[VSCAN_HOLDOFF] control field allows control over the vertical blanking period allotment for working buffer processing and pre-loading of the pixel FIFO. The time window defined by VSCAN_HOLDOFF is allotted for working-buffer processing and LUT loading (of newly processed updates). The remainder of the time is allotted for pre-filling the pixel FIFO.

Note that aggressive use of VSCAN_HOLDOFF can result in pixel FIFO under-run as the FIFO pre-fill time is compromised.

24.3.11.5 System-Level Arbitration Control

As previously mentioned, the TCE refresh operation is time critical.

Because the EPDC is designed to be integrated into a multimaster SoC, system memory is shared between multiple requesters in the system.

The EPDC provides a dynamic Quality of Service (QoS) priority elevation mechanism that can be leveraged at the SoC level.

The EPDC_FIFOCTRL registers can be used to drive the epdc_panic output signal during times when the EPDC requests require priority elevation. As long as epdc_panic is asserted, transaction requests from the EPDC will have their priority elevated at the SoC-level bus arbitration control points.

The registers and associated functionality are as follows:

- EPDC_FIFOCTRL[ENABLE_PRIORITY] must be set to enable epdc_panic functionality.
- EPDC_FIFOCTRL[FIFO_L_LEVEL] determines FIFO threshold which causes epdc_panic to assert when crossed.
- EPDC_FIFOCTRL[FIFO_H_LEVEL] once epdc_panic has been asserted due to the pixel FIFO(s) crossing FIFO_L_LEVEL, this value, which must be set higher than FIFO_L_LEVEL, determines the point at which epdc_panic will de-assert as the FIFO(s) fills.

Note that the delta between FIFO_H_LEVEL and FIFO_L_LEVEL is intended as a de-bounce mechanism. The purpose is to avoid a constant panic situation, which would cause thrashing of the epdc_panic signal.

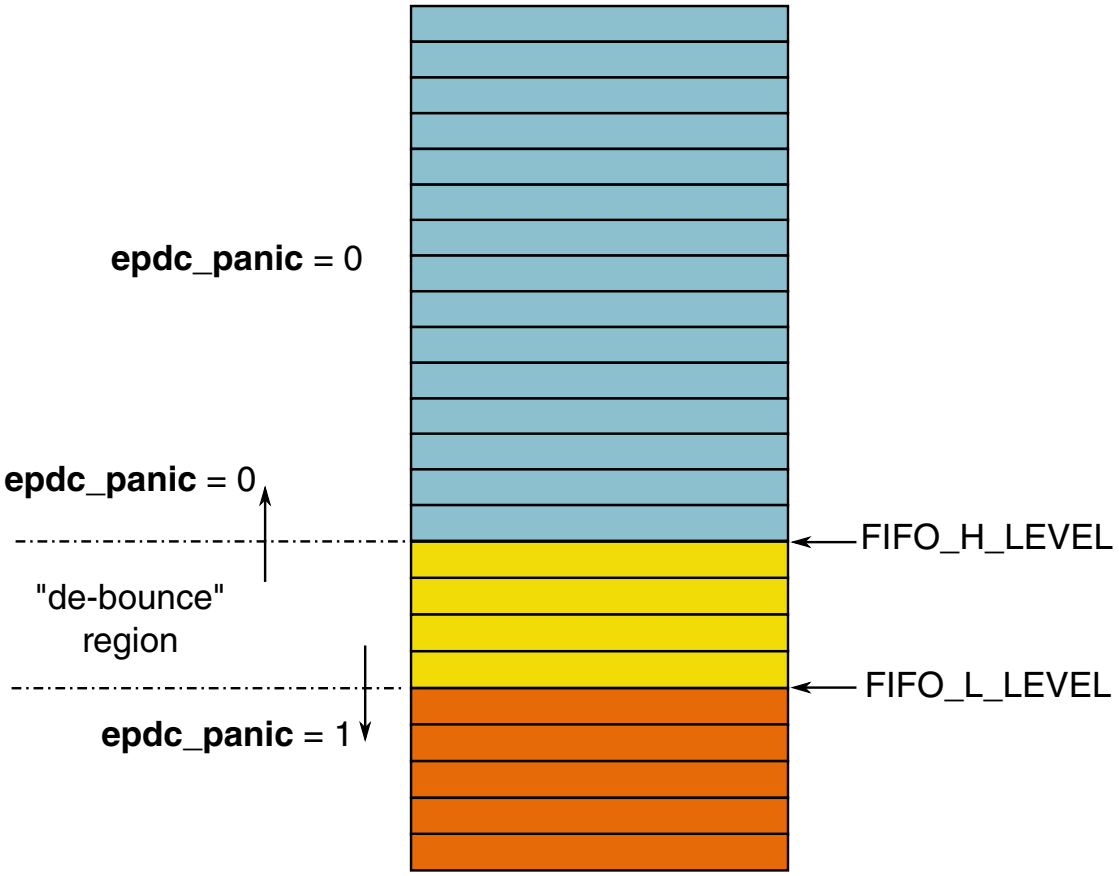


Figure 24-16. FIFO Priority Elevation

24.4 EPDC Memory Map/Register Definition

EPDC Register Format Summary

EPDC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20F_4000	EPDC Control Register (EPDC_CTRL)	32	R/W	C000_0000h	24.4.1/1249
20F_4004	EPDC Control Register (EPDC_CTRL_SET)	32	R/W	C000_0000h	24.4.1/1249
20F_4008	EPDC Control Register (EPDC_CTRL_CLR)	32	R/W	C000_0000h	24.4.1/1249
20F_400C	EPDC Control Register (EPDC_CTRL_TOG)	32	R/W	C000_0000h	24.4.1/1249
20F_4020	EPDC Waveform Address Pointer (EPDC_WVADDR)	32	R/W	0000_0000h	24.4.2/1250
20F_4030	EPDC Working Buffer Address (EPDC_WB_ADDR)	32	R/W	0000_0000h	24.4.3/1251
20F_4040	EPDC Screen Resolution (EPDC_RES)	32	R/W	0000_0000h	24.4.4/1252

Table continues on the next page...

EPDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20F_4050	EPDC Format Control Register (EPDC_FORMAT)	32	R/W	0000_0000h	24.4.5/1252
20F_4054	EPDC Format Control Register (EPDC_FORMAT_SET)	32	R/W	0000_0000h	24.4.5/1252
20F_4058	EPDC Format Control Register (EPDC_FORMAT_CLR)	32	R/W	0000_0000h	24.4.5/1252
20F_405C	EPDC Format Control Register (EPDC_FORMAT_TOG)	32	R/W	0000_0000h	24.4.5/1252
20F_40A0	EPDC FIFO control register (EPDC_FIFOCTRL)	32	R/W	0080_0000h	24.4.6/1254
20F_40A4	EPDC FIFO control register (EPDC_FIFOCTRL_SET)	32	R/W	0080_0000h	24.4.6/1254
20F_40A8	EPDC FIFO control register (EPDC_FIFOCTRL_CLR)	32	R/W	0080_0000h	24.4.6/1254
20F_40AC	EPDC FIFO control register (EPDC_FIFOCTRL_TOG)	32	R/W	0080_0000h	24.4.6/1254
20F_4100	EPDC Update Region Address (EPDC_UPD_ADDR)	32	R/W	0000_0000h	24.4.7/1255
20F_4110	EPDC Update Region Stride (EPDC_UPD_STRIDE)	32	R/W	0000_0000h	24.4.8/1256
20F_4120	EPDC Update Command Co-ordinate (EPDC_UPD_CORD)	32	R/W	0000_0000h	24.4.9/1257
20F_4140	EPDC Update Command Size (EPDC_UPD_SIZE)	32	R/W	0000_0000h	24.4.10/ 1257
20F_4160	EPDC Update Command Control (EPDC_UPD_CTRL)	32	R/W	0000_0000h	24.4.11/ 1258
20F_4164	EPDC Update Command Control (EPDC_UPD_CTRL_SET)	32	R/W	0000_0000h	24.4.11/ 1258
20F_4168	EPDC Update Command Control (EPDC_UPD_CTRL_CLR)	32	R/W	0000_0000h	24.4.11/ 1258
20F_416C	EPDC Update Command Control (EPDC_UPD_CTRL_TOG)	32	R/W	0000_0000h	24.4.11/ 1258
20F_4180	EPDC Update Fixed Pixel Control (EPDC_UPD_FIXED)	32	R/W	0000_0000h	24.4.12/ 1259
20F_4184	EPDC Update Fixed Pixel Control (EPDC_UPD_FIXED_SET)	32	R/W	0000_0000h	24.4.12/ 1259
20F_4188	EPDC Update Fixed Pixel Control (EPDC_UPD_FIXED_CLR)	32	R/W	0000_0000h	24.4.12/ 1259
20F_418C	EPDC Update Fixed Pixel Control (EPDC_UPD_FIXED_TOG)	32	R/W	0000_0000h	24.4.12/ 1259
20F_41A0	EPDC Temperature Register (EPDC_TEMP)	32	R/W	0000_0000h	24.4.13/ 1260
20F_41C0	Waveform Mode Lookup Table Control Register. (EPDC_AUTOWV_LUT)	32	R/W	0000_0000h	24.4.14/ 1261
20F_4200	EPDC Timing Control Engine Control Register (EPDC_TCE_CTRL)	32	R/W	0000_0010h	24.4.15/ 1261
20F_4204	EPDC Timing Control Engine Control Register (EPDC_TCE_CTRL_SET)	32	R/W	0000_0010h	24.4.15/ 1261
20F_4208	EPDC Timing Control Engine Control Register (EPDC_TCE_CTRL_CLR)	32	R/W	0000_0010h	24.4.15/ 1261
20F_420C	EPDC Timing Control Engine Control Register (EPDC_TCE_CTRL_TOG)	32	R/W	0000_0010h	24.4.15/ 1261
20F_4220	EPDC Timing Control Engine Source-Driver Config Register (EPDC_TCE_SDCFG)	32	R/W	0000_0000h	24.4.16/ 1263

Table continues on the next page...

EPDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20F_4224	EPDC Timing Control Engine Source-Driver Config Register (EPDC_TCE_SDCFG_SET)	32	R/W	0000_0000h	24.4.16/ 1263
20F_4228	EPDC Timing Control Engine Source-Driver Config Register (EPDC_TCE_SDCFG_CLR)	32	R/W	0000_0000h	24.4.16/ 1263
20F_422C	EPDC Timing Control Engine Source-Driver Config Register (EPDC_TCE_SDCFG_TOG)	32	R/W	0000_0000h	24.4.16/ 1263
20F_4240	EPDC Timing Control Engine Gate-Driver Config Register (EPDC_TCE_GDCFG)	32	R/W	0000_0000h	24.4.17/ 1265
20F_4244	EPDC Timing Control Engine Gate-Driver Config Register (EPDC_TCE_GDCFG_SET)	32	R/W	0000_0000h	24.4.17/ 1265
20F_4248	EPDC Timing Control Engine Gate-Driver Config Register (EPDC_TCE_GDCFG_CLR)	32	R/W	0000_0000h	24.4.17/ 1265
20F_424C	EPDC Timing Control Engine Gate-Driver Config Register (EPDC_TCE_GDCFG_TOG)	32	R/W	0000_0000h	24.4.17/ 1265
20F_4260	EPDC Timing Control Engine Horizontal Timing Register 1 (EPDC_TCE_HSCAN1)	32	R/W	0000_0000h	24.4.18/ 1266
20F_4264	EPDC Timing Control Engine Horizontal Timing Register 1 (EPDC_TCE_HSCAN1_SET)	32	R/W	0000_0000h	24.4.18/ 1266
20F_4268	EPDC Timing Control Engine Horizontal Timing Register 1 (EPDC_TCE_HSCAN1_CLR)	32	R/W	0000_0000h	24.4.18/ 1266
20F_426C	EPDC Timing Control Engine Horizontal Timing Register 1 (EPDC_TCE_HSCAN1_TOG)	32	R/W	0000_0000h	24.4.18/ 1266
20F_4280	EPDC Timing Control Engine Horizontal Timing Register 2 (EPDC_TCE_HSCAN2)	32	R/W	0000_0000h	24.4.19/ 1266
20F_4284	EPDC Timing Control Engine Horizontal Timing Register 2 (EPDC_TCE_HSCAN2_SET)	32	R/W	0000_0000h	24.4.19/ 1266
20F_4288	EPDC Timing Control Engine Horizontal Timing Register 2 (EPDC_TCE_HSCAN2_CLR)	32	R/W	0000_0000h	24.4.19/ 1266
20F_428C	EPDC Timing Control Engine Horizontal Timing Register 2 (EPDC_TCE_HSCAN2_TOG)	32	R/W	0000_0000h	24.4.19/ 1266
20F_42A0	EPDC Timing Control Engine Vertical Timing Register (EPDC_TCE_VSCAN)	32	R/W	0000_0000h	24.4.20/ 1267
20F_42A4	EPDC Timing Control Engine Vertical Timing Register (EPDC_TCE_VSCAN_SET)	32	R/W	0000_0000h	24.4.20/ 1267
20F_42A8	EPDC Timing Control Engine Vertical Timing Register (EPDC_TCE_VSCAN_CLR)	32	R/W	0000_0000h	24.4.20/ 1267
20F_42AC	EPDC Timing Control Engine Vertical Timing Register (EPDC_TCE_VSCAN_TOG)	32	R/W	0000_0000h	24.4.20/ 1267
20F_42C0	EPDC Timing Control Engine OE timing control Register (EPDC_TCE_OE)	32	R/W	0000_0000h	24.4.21/ 1268
20F_42C4	EPDC Timing Control Engine OE timing control Register (EPDC_TCE_OE_SET)	32	R/W	0000_0000h	24.4.21/ 1268
20F_42C8	EPDC Timing Control Engine OE timing control Register (EPDC_TCE_OE_CLR)	32	R/W	0000_0000h	24.4.21/ 1268

Table continues on the next page...

EPDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20F_42CC	EPDC Timing Control Engine OE timing control Register (EPDC_TCE_OE_TOG)	32	R/W	0000_0000h	24.4.21/ 1268
20F_42E0	EPDC Timing Control Engine Driver Polarity Register (EPDC_TCE_POLARITY)	32	R/W	0000_001Eh	24.4.22/ 1268
20F_4300	EPDC Timing Control Engine Timing Register 1 (EPDC_TCE_TIMING1)	32	R/W	0000_0000h	24.4.23/ 1269
20F_4304	EPDC Timing Control Engine Timing Register 1 (EPDC_TCE_TIMING1_SET)	32	R/W	0000_0000h	24.4.23/ 1269
20F_4308	EPDC Timing Control Engine Timing Register 1 (EPDC_TCE_TIMING1_CLR)	32	R/W	0000_0000h	24.4.23/ 1269
20F_430C	EPDC Timing Control Engine Timing Register 1 (EPDC_TCE_TIMING1_TOG)	32	R/W	0000_0000h	24.4.23/ 1269
20F_4310	EPDC Timing Control Engine Timing Register 2 (EPDC_TCE_TIMING2)	32	R/W	0000_0001h	24.4.24/ 1271
20F_4314	EPDC Timing Control Engine Timing Register 2 (EPDC_TCE_TIMING2_SET)	32	R/W	0000_0001h	24.4.24/ 1271
20F_4318	EPDC Timing Control Engine Timing Register 2 (EPDC_TCE_TIMING2_CLR)	32	R/W	0000_0001h	24.4.24/ 1271
20F_431C	EPDC Timing Control Engine Timing Register 2 (EPDC_TCE_TIMING2_TOG)	32	R/W	0000_0001h	24.4.24/ 1271
20F_4320	EPDC Timing Control Engine Timing Register 3 (EPDC_TCE_TIMING3)	32	R/W	0000_0001h	24.4.25/ 1271
20F_4324	EPDC Timing Control Engine Timing Register 3 (EPDC_TCE_TIMING3_SET)	32	R/W	0000_0001h	24.4.25/ 1271
20F_4328	EPDC Timing Control Engine Timing Register 3 (EPDC_TCE_TIMING3_CLR)	32	R/W	0000_0001h	24.4.25/ 1271
20F_432C	EPDC Timing Control Engine Timing Register 3 (EPDC_TCE_TIMING3_TOG)	32	R/W	0000_0001h	24.4.25/ 1271
20F_4380	EPDC Pigeon Mode Control Register 0 (EPDC_PIGEON_CTRL0)	32	R/W	0000_0000h	24.4.26/ 1272
20F_4384	EPDC Pigeon Mode Control Register 0 (EPDC_PIGEON_CTRL0_SET)	32	R/W	0000_0000h	24.4.26/ 1272
20F_4388	EPDC Pigeon Mode Control Register 0 (EPDC_PIGEON_CTRL0_CLR)	32	R/W	0000_0000h	24.4.26/ 1272
20F_438C	EPDC Pigeon Mode Control Register 0 (EPDC_PIGEON_CTRL0_TOG)	32	R/W	0000_0000h	24.4.26/ 1272
20F_4390	EPDC Pigeon Mode Control Register 1 (EPDC_PIGEON_CTRL1)	32	R/W	0000_0000h	24.4.27/ 1272
20F_4394	EPDC Pigeon Mode Control Register 1 (EPDC_PIGEON_CTRL1_SET)	32	R/W	0000_0000h	24.4.27/ 1272
20F_4398	EPDC Pigeon Mode Control Register 1 (EPDC_PIGEON_CTRL1_CLR)	32	R/W	0000_0000h	24.4.27/ 1272
20F_439C	EPDC Pigeon Mode Control Register 1 (EPDC_PIGEON_CTRL1_TOG)	32	R/W	0000_0000h	24.4.27/ 1272

Table continues on the next page...

EPDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20F_43C0	EPDC IRQ Mask Register for LUT 0~31 (EPDC_IRQ_MASK1)	32	R/W	0000_0000h	24.4.28/ 1273
20F_43C4	EPDC IRQ Mask Register for LUT 0~31 (EPDC_IRQ_MASK1_SET)	32	R/W	0000_0000h	24.4.28/ 1273
20F_43C8	EPDC IRQ Mask Register for LUT 0~31 (EPDC_IRQ_MASK1_CLR)	32	R/W	0000_0000h	24.4.28/ 1273
20F_43CC	EPDC IRQ Mask Register for LUT 0~31 (EPDC_IRQ_MASK1_TOG)	32	R/W	0000_0000h	24.4.28/ 1273
20F_43D0	EPDC IRQ Mask Register for LUT 32~63 (EPDC_IRQ_MASK2)	32	R/W	0000_0000h	24.4.29/ 1273
20F_43D4	EPDC IRQ Mask Register for LUT 32~63 (EPDC_IRQ_MASK2_SET)	32	R/W	0000_0000h	24.4.29/ 1273
20F_43D8	EPDC IRQ Mask Register for LUT 32~63 (EPDC_IRQ_MASK2_CLR)	32	R/W	0000_0000h	24.4.29/ 1273
20F_43DC	EPDC IRQ Mask Register for LUT 32~63 (EPDC_IRQ_MASK2_TOG)	32	R/W	0000_0000h	24.4.29/ 1273
20F_43E0	EPDC Interrupt Register for LUT 0~31 (EPDC_IRQ1)	32	R/W	0000_0000h	24.4.30/ 1274
20F_43E4	EPDC Interrupt Register for LUT 0~31 (EPDC_IRQ1_SET)	32	R/W	0000_0000h	24.4.30/ 1274
20F_43E8	EPDC Interrupt Register for LUT 0~31 (EPDC_IRQ1_CLR)	32	R/W	0000_0000h	24.4.30/ 1274
20F_43EC	EPDC Interrupt Register for LUT 0~31 (EPDC_IRQ1_TOG)	32	R/W	0000_0000h	24.4.30/ 1274
20F_43F0	EPDC Interrupt Registerr for LUT 32~63 (EPDC_IRQ2)	32	R/W	0000_0000h	24.4.31/ 1274
20F_43F4	EPDC Interrupt Registerr for LUT 32~63 (EPDC_IRQ2_SET)	32	R/W	0000_0000h	24.4.31/ 1274
20F_43F8	EPDC Interrupt Registerr for LUT 32~63 (EPDC_IRQ2_CLR)	32	R/W	0000_0000h	24.4.31/ 1274
20F_43FC	EPDC Interrupt Registerr for LUT 32~63 (EPDC_IRQ2_TOG)	32	R/W	0000_0000h	24.4.31/ 1274
20F_4400	EPDC IRQ Mask Register (EPDC_IRQ_MASK)	32	R/W	0000_0000h	24.4.32/ 1275
20F_4404	EPDC IRQ Mask Register (EPDC_IRQ_MASK_SET)	32	R/W	0000_0000h	24.4.32/ 1275
20F_4408	EPDC IRQ Mask Register (EPDC_IRQ_MASK_CLR)	32	R/W	0000_0000h	24.4.32/ 1275
20F_440C	EPDC IRQ Mask Register (EPDC_IRQ_MASK_TOG)	32	R/W	0000_0000h	24.4.32/ 1275
20F_4420	EPDC Interrupt Register (EPDC_IRQ)	32	R/W	0000_0000h	24.4.33/ 1276
20F_4424	EPDC Interrupt Register (EPDC_IRQ_SET)	32	R/W	0000_0000h	24.4.33/ 1276

Table continues on the next page...

EPDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20F_4428	EPDC Interrupt Register (EPDC_IRQ_CLR)	32	R/W	0000_0000h	24.4.33/ 1276
20F_442C	EPDC Interrupt Register (EPDC_IRQ_TOG)	32	R/W	0000_0000h	24.4.33/ 1276
20F_4440	EPDC Status Register - LUTs (EPDC_STATUS_LUTS1)	32	R/W	0000_0000h	24.4.34/ 1278
20F_4444	EPDC Status Register - LUTs (EPDC_STATUS_LUTS1_SET)	32	R/W	0000_0000h	24.4.34/ 1278
20F_4448	EPDC Status Register - LUTs (EPDC_STATUS_LUTS1_CLR)	32	R/W	0000_0000h	24.4.34/ 1278
20F_444C	EPDC Status Register - LUTs (EPDC_STATUS_LUTS1_TOG)	32	R/W	0000_0000h	24.4.34/ 1278
20F_4450	EPDC Status Register - LUTs (EPDC_STATUS_LUTS2)	32	R/W	0000_0000h	24.4.35/ 1278
20F_4454	EPDC Status Register - LUTs (EPDC_STATUS_LUTS2_SET)	32	R/W	0000_0000h	24.4.35/ 1278
20F_4458	EPDC Status Register - LUTs (EPDC_STATUS_LUTS2_CLR)	32	R/W	0000_0000h	24.4.35/ 1278
20F_445C	EPDC Status Register - LUTs (EPDC_STATUS_LUTS2_TOG)	32	R/W	0000_0000h	24.4.35/ 1278
20F_4460	EPDC Status Register - Next Available LUT (EPDC_STATUS_NEXTLUT)	32	R/W	0000_013Fh	24.4.36/ 1278
20F_4480	EPDC LUT Collision Status (EPDC_STATUS_COL1)	32	R/W	0000_0000h	24.4.37/ 1280
20F_4484	EPDC LUT Collision Status (EPDC_STATUS_COL1_SET)	32	R/W	0000_0000h	24.4.37/ 1280
20F_4488	EPDC LUT Collision Status (EPDC_STATUS_COL1_CLR)	32	R/W	0000_0000h	24.4.37/ 1280
20F_448C	EPDC LUT Collision Status (EPDC_STATUS_COL1_TOG)	32	R/W	0000_0000h	24.4.37/ 1280
20F_4490	EPDC LUT Collision Status (EPDC_STATUS_COL2)	32	R/W	0000_0000h	24.4.38/ 1280
20F_4494	EPDC LUT Collision Status (EPDC_STATUS_COL2_SET)	32	R/W	0000_0000h	24.4.38/ 1280
20F_4498	EPDC LUT Collision Status (EPDC_STATUS_COL2_CLR)	32	R/W	0000_0000h	24.4.38/ 1280
20F_449C	EPDC LUT Collision Status (EPDC_STATUS_COL2_TOG)	32	R/W	0000_0000h	24.4.38/ 1280
20F_44A0	EPDC General Status Register (EPDC_STATUS)	32	R/W	0000_0000h	24.4.39/ 1281
20F_44A4	EPDC General Status Register (EPDC_STATUS_SET)	32	R/W	0000_0000h	24.4.39/ 1281
20F_44A8	EPDC General Status Register (EPDC_STATUS_CLR)	32	R/W	0000_0000h	24.4.39/ 1281

Table continues on the next page...

EPDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20F_44AC	EPDC General Status Register (EPDC_STATUS_TOG)	32	R/W	0000_0000h	24.4.39/ 1281
20F_44C0	EPDC Collision Region Co-ordinate (EPDC_UPD_COL_CORD)	32	R/W	0000_0000h	24.4.40/ 1283
20F_44E0	EPDC Collision Region Size (EPDC_UPD_COL_SIZE)	32	R/W	0000_0000h	24.4.41/ 1283
20F_4600	1-level Histogram Parameter Register. (EPDC_HIST1_PARAM)	32	R/W	0000_0000h	24.4.42/ 1284
20F_4610	2-level Histogram Parameter Register. (EPDC_HIST2_PARAM)	32	R/W	0000_0F00h	24.4.43/ 1285
20F_4620	4-level Histogram Parameter Register. (EPDC_HIST4_PARAM)	32	R/W	0F0A_0500h	24.4.44/ 1285
20F_4630	8-level Histogram Parameter 0 Register. (EPDC_HIST8_PARAM0)	32	R/W	0604_4000h	24.4.45/ 1286
20F_4640	8-level Histogram Parameter 1 Register. (EPDC_HIST8_PARAM1)	32	R/W	0F0D_0B09h	24.4.46/ 1287
20F_4650	16-level Histogram Parameter 0 Register. (EPDC_HIST16_PARAM0)	32	R/W	0302_0100h	24.4.47/ 1288
20F_4660	16-level Histogram Parameter Register. (EPDC_HIST16_PARAM1)	32	R/W	0706_0504h	24.4.48/ 1288
20F_4670	16-level Histogram Parameter Register. (EPDC_HIST16_PARAM2)	32	R/W	0B0A_0908h	24.4.49/ 1289
20F_4680	16-level Histogram Parameter Register. (EPDC_HIST16_PARAM3)	32	R/W	0F0E_0D0Ch	24.4.50/ 1290
20F_4700	EPDC General Purpose I/O Debug register (EPDC_GPIO)	32	R/W	0000_0000h	24.4.51/ 1291
20F_4704	EPDC General Purpose I/O Debug register (EPDC_GPIO_SET)	32	R/W	0000_0000h	24.4.51/ 1291
20F_4708	EPDC General Purpose I/O Debug register (EPDC_GPIO_CLR)	32	R/W	0000_0000h	24.4.51/ 1291
20F_470C	EPDC General Purpose I/O Debug register (EPDC_GPIO_TOG)	32	R/W	0000_0000h	24.4.51/ 1291
20F_47F0	EPDC Version Register (EPDC_VERSION)	32	R/W	0200_0000h	24.4.52/ 1292
20F_4800	Panel Interface Signal Generator Register 0_0 (EPDC_PIGEON_0_0)	32	R/W	0000_0F00h	24.4.53/ 1292
20F_4810	Panel Interface Signal Generator Register 0_1 (EPDC_PIGEON_0_1)	32	R/W	0000_0000h	24.4.54/ 1294
20F_4820	Panel Interface Signal Generator Register 0_1 (EPDC_PIGEON_0_2)	32	R/W	0000_0000h	24.4.55/ 1294
20F_4840	Panel Interface Signal Generator Register 1_0 (EPDC_PIGEON_1_0)	32	R/W	0000_0F00h	24.4.56/ 1295
20F_4850	Panel Interface Signal Generator Register 1_1 (EPDC_PIGEON_1_1)	32	R/W	0000_0000h	24.4.57/ 1296

Table continues on the next page...

EPDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20F_4860	Panel Interface Signal Generator Register 1_1 (EPDC_PIGEON_1_2)	32	R/W	0000_0000h	24.4.58/ 1297
20F_4880	Panel Interface Signal Generator Register 2_0 (EPDC_PIGEON_2_0)	32	R/W	0000_0F00h	24.4.59/ 1297
20F_4890	Panel Interface Signal Generator Register 2_1 (EPDC_PIGEON_2_1)	32	R/W	0000_0000h	24.4.60/ 1298
20F_48A0	Panel Interface Signal Generator Register 2_1 (EPDC_PIGEON_2_2)	32	R/W	0000_0000h	24.4.61/ 1299
20F_48C0	Panel Interface Signal Generator Register 3_0 (EPDC_PIGEON_3_0)	32	R/W	0000_0F00h	24.4.62/ 1300
20F_48D0	Panel Interface Signal Generator Register 3_1 (EPDC_PIGEON_3_1)	32	R/W	0000_0000h	24.4.63/ 1301
20F_48E0	Panel Interface Signal Generator Register 3_1 (EPDC_PIGEON_3_2)	32	R/W	0000_0000h	24.4.64/ 1301
20F_4900	Panel Interface Signal Generator Register 4_0 (EPDC_PIGEON_4_0)	32	R/W	0000_0F00h	24.4.65/ 1302
20F_4910	Panel Interface Signal Generator Register 4_1 (EPDC_PIGEON_4_1)	32	R/W	0000_0000h	24.4.66/ 1303
20F_4920	Panel Interface Signal Generator Register 4_1 (EPDC_PIGEON_4_2)	32	R/W	0000_0000h	24.4.67/ 1304
20F_4940	Panel Interface Signal Generator Register 5_0 (EPDC_PIGEON_5_0)	32	R/W	0000_0F00h	24.4.68/ 1304
20F_4950	Panel Interface Signal Generator Register 5_1 (EPDC_PIGEON_5_1)	32	R/W	0000_0000h	24.4.69/ 1306
20F_4960	Panel Interface Signal Generator Register 5_1 (EPDC_PIGEON_5_2)	32	R/W	0000_0000h	24.4.70/ 1306
20F_4980	Panel Interface Signal Generator Register 6_0 (EPDC_PIGEON_6_0)	32	R/W	0000_0F00h	24.4.71/ 1307
20F_4990	Panel Interface Signal Generator Register 6_1 (EPDC_PIGEON_6_1)	32	R/W	0000_0000h	24.4.72/ 1308
20F_49A0	Panel Interface Signal Generator Register 6_1 (EPDC_PIGEON_6_2)	32	R/W	0000_0000h	24.4.73/ 1309
20F_49C0	Panel Interface Signal Generator Register 7_0 (EPDC_PIGEON_7_0)	32	R/W	0000_0F00h	24.4.74/ 1309
20F_49D0	Panel Interface Signal Generator Register 7_1 (EPDC_PIGEON_7_1)	32	R/W	0000_0000h	24.4.75/ 1310
20F_49E0	Panel Interface Signal Generator Register 7_1 (EPDC_PIGEON_7_2)	32	R/W	0000_0000h	24.4.76/ 1311
20F_4A00	Panel Interface Signal Generator Register 8_0 (EPDC_PIGEON_8_0)	32	R/W	0000_0F00h	24.4.77/ 1312
20F_4A10	Panel Interface Signal Generator Register 8_1 (EPDC_PIGEON_8_1)	32	R/W	0000_0000h	24.4.78/ 1313
20F_4A20	Panel Interface Signal Generator Register 8_1 (EPDC_PIGEON_8_2)	32	R/W	0000_0000h	24.4.79/ 1313

Table continues on the next page...

EPDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20F_4A40	Panel Interface Signal Generator Register 9_0 (EPDC_PIGEON_9_0)	32	R/W	0000_0F00h	24.4.80/ 1314
20F_4A50	Panel Interface Signal Generator Register 9_1 (EPDC_PIGEON_9_1)	32	R/W	0000_0000h	24.4.81/ 1315
20F_4A60	Panel Interface Signal Generator Register 9_1 (EPDC_PIGEON_9_2)	32	R/W	0000_0000h	24.4.82/ 1316
20F_4A80	Panel Interface Signal Generator Register 10_0 (EPDC_PIGEON_10_0)	32	R/W	0000_0F00h	24.4.83/ 1316
20F_4A90	Panel Interface Signal Generator Register 10_1 (EPDC_PIGEON_10_1)	32	R/W	0000_0000h	24.4.84/ 1318
20F_4AA0	Panel Interface Signal Generator Register 10_1 (EPDC_PIGEON_10_2)	32	R/W	0000_0000h	24.4.85/ 1318
20F_4AC0	Panel Interface Signal Generator Register 11_0 (EPDC_PIGEON_11_0)	32	R/W	0000_0F00h	24.4.86/ 1319
20F_4AD0	Panel Interface Signal Generator Register 11_1 (EPDC_PIGEON_11_1)	32	R/W	0000_0000h	24.4.87/ 1320
20F_4AE0	Panel Interface Signal Generator Register 11_1 (EPDC_PIGEON_11_2)	32	R/W	0000_0000h	24.4.88/ 1321
20F_4B00	Panel Interface Signal Generator Register 12_0 (EPDC_PIGEON_12_0)	32	R/W	0000_0F00h	24.4.89/ 1321
20F_4B10	Panel Interface Signal Generator Register 12_1 (EPDC_PIGEON_12_1)	32	R/W	0000_0000h	24.4.90/ 1322
20F_4B20	Panel Interface Signal Generator Register 12_1 (EPDC_PIGEON_12_2)	32	R/W	0000_0000h	24.4.91/ 1323
20F_4B40	Panel Interface Signal Generator Register 13_0 (EPDC_PIGEON_13_0)	32	R/W	0000_0F00h	24.4.92/ 1324
20F_4B50	Panel Interface Signal Generator Register 13_1 (EPDC_PIGEON_13_1)	32	R/W	0000_0000h	24.4.93/ 1325
20F_4B60	Panel Interface Signal Generator Register 13_1 (EPDC_PIGEON_13_2)	32	R/W	0000_0000h	24.4.94/ 1325
20F_4B80	Panel Interface Signal Generator Register 14_0 (EPDC_PIGEON_14_0)	32	R/W	0000_0F00h	24.4.95/ 1326
20F_4B90	Panel Interface Signal Generator Register 14_1 (EPDC_PIGEON_14_1)	32	R/W	0000_0000h	24.4.96/ 1327
20F_4BA0	Panel Interface Signal Generator Register 14_1 (EPDC_PIGEON_14_2)	32	R/W	0000_0000h	24.4.97/ 1328
20F_4BC0	Panel Interface Signal Generator Register 15_0 (EPDC_PIGEON_15_0)	32	R/W	0000_0F00h	24.4.98/ 1328
20F_4BD0	Panel Interface Signal Generator Register 15_1 (EPDC_PIGEON_15_1)	32	R/W	0000_0000h	24.4.99/ 1330
20F_4BE0	Panel Interface Signal Generator Register 15_1 (EPDC_PIGEON_15_2)	32	R/W	0000_0000h	24.4.100/ 1330
20F_4C00	Panel Interface Signal Generator Register 16_0 (EPDC_PIGEON_16_0)	32	R/W	0000_0F00h	24.4.101/ 1331

Table continues on the next page...

EPDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20F_4C10	Panel Interface Signal Generator Register 16_1 (EPDC_PIGEON_16_1)	32	R/W	0000_0000h	24.4.102/ 1332
20F_4C20	Panel Interface Signal Generator Register 16_1 (EPDC_PIGEON_16_2)	32	R/W	0000_0000h	24.4.103/ 1333

24.4.1 EPDC Control Register (EPDC_CTRL*n*)

EPDC Main control register

This register controls various high-level functions of the EPDC

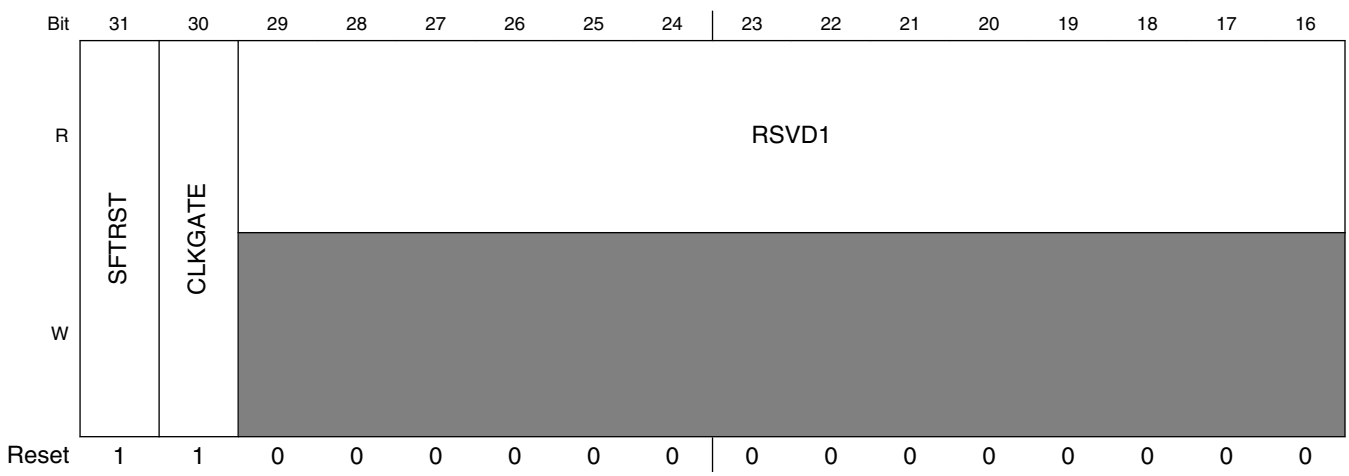
EXAMPLE

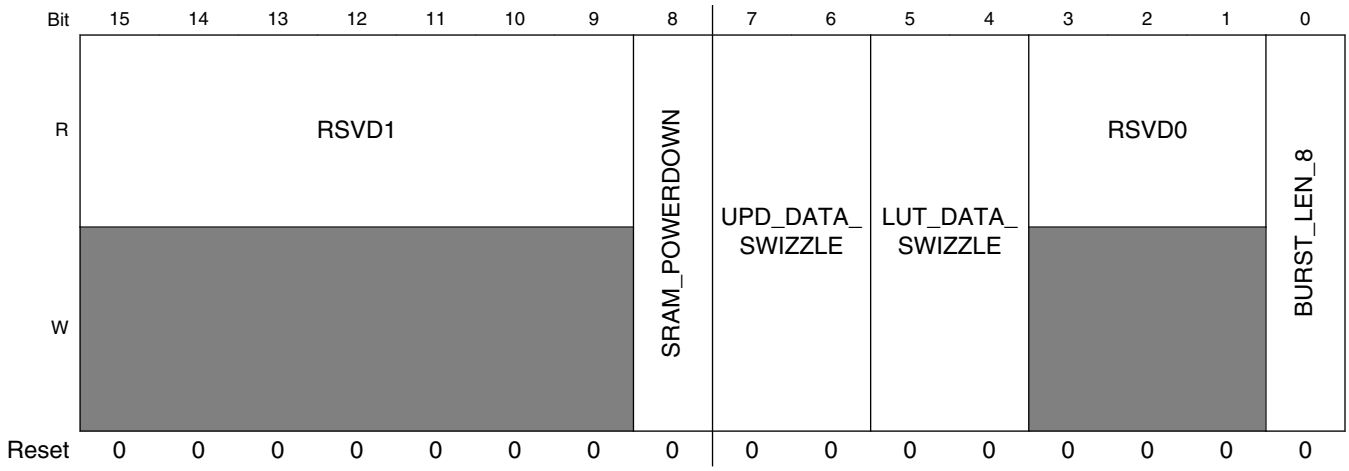
```
/* Reset */
__raw_writel(EPDC_CTRL_SFTRST, EPDC_CTRL_SET);
while (!(__raw_readl(EPDC_CTRL) & EPDC_CTRL_CLKGATE))
    ;
__raw_writel(EPDC_CTRL_SFTRST, EPDC_CTRL_CLEAR);

/* Enable clock gating (clear to enable) */
__raw_writel(EPDC_CTRL_CLKGATE, EPDC_CTRL_CLEAR);
while (__raw_readl(EPDC_CTRL) & (EPDC_CTRL_SFTRST | EPDC_CTRL_CLKGATE))
    ;

/* EPDC_CTRL */
reg_val = __raw_readl(EPDC_CTRL);
reg_val &= ~EPDC_CTRL_UPD_DATA_SWIZZLE_MASK;
reg_val |= EPDC_CTRL_UPD_DATA_SWIZZLE_NO_SWAP;
reg_val &= ~EPDC_CTRL_LUT_DATA_SWIZZLE_MASK;
reg_val |= EPDC_CTRL_LUT_DATA_SWIZZLE_NO_SWAP;
__raw_writel(reg_val, EPDC_CTRL_SET);
```

Address: 20F_4000h base + 0h offset + (4d × i), where i=0d to 3d





EPDC_CTRL*n* field descriptions

Field	Description
31 SFTRST	Set this bit to zero to enable normal EPDC operation. Set this bit to one (default) to disable clocking with the EPDC and hold it in its reset (lowest power) state. This bit can be turned on and then off to reset the EPDC block to its default state.
30 CLKGATE	This bit must be set to zero for normal operation. When set to one it gates off the clocks to the block.
29–9 RSVD1	Reserved.
8 SRAM_ POWERDOWN	Enable Power-down of embedded SRAM memories
7–6 UPD_DATA_ SWIZZLE	Specifies how to swap the bytes for the UPD data before the WB construction. Plesae note this swizzle operate right after data fetch from bus, no matter it's aligned access or not. Supported configurations: 0x0 **NO_SWAP** — No byte swapping.(Little endian) 0x1 **ALL_BYTES_SWAP** — Swizzle all bytes, swap bytes 0,3 and 1,2 (aka Big Endian). 0x2 **HWD_SWAP** — Swap half-words. 0x3 **HWD_BYTE_SWAP** — Swap bytes within each half-word.
5–4 LUT_DATA_ SWIZZLE	Specifies how to swap the bytes for the LUT data before store to LUTRAM. Supported configurations: 0x0 **NO_SWAP** — No byte swapping.(Little endian) 0x1 **ALL_BYTES_SWAP** — Swizzle all bytes, swap bytes 0,3 and 1,2 (aka Big Endian). 0x2 **HWD_SWAP** — Swap half-words. 0x3 **HWD_BYTE_SWAP** — Swap bytes within each half-word.
3–1 RSVD0	Reserved.
0 BURST_LEN_8	0- EPDC display fifo logic will issue AXI bursts of length 16. When set to 1, the block will issue bursts of length 8.

24.4.2 EPDC Waveform Address Pointer (EPDC_WVADDR)

EPDC Waveform Address Pointer

EXAMPLE

```
/* Reset */
__raw_writel(EPDC_CTRL_SFTRST, EPDC_CTRL_SET);
while (!(__raw_readl(EPDC_CTRL) & EPDC_CTRL_CLKGATE))
    ;
__raw_writel(EPDC_CTRL_SFTRST, EPDC_CTRL_CLEAR);

/* Enable clock gating (clear to enable) */
__raw_writel(EPDC_CTRL_CLKGATE, EPDC_CTRL_CLEAR);
while (__raw_readl(EPDC_CTRL) & (EPDC_CTRL_SFTRST | EPDC_CTRL_CLKGATE))
    ;

/* EPDC_CTRL */
reg_val = __raw_readl(EPDC_CTRL);
reg_val &= ~EPDC_CTRL_UPD_DATA_SWIZZLE_MASK;
reg_val |= EPDC_CTRL_UPD_DATA_SWIZZLE_NO_SWAP;
reg_val &= ~EPDC_CTRL_LUT_DATA_SWIZZLE_MASK;
reg_val |= EPDC_CTRL_LUT_DATA_SWIZZLE_NO_SWAP;
__raw_writel(reg_val, EPDC_CTRL_SET);
```

Address: 20F_4000h base + 20h offset = 20F_4020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_WVADDR field descriptions

Field	Description
ADDR	Start address of waveform tables. This address needs to be aligned to a 64-bit word boundary.

24.4.3 EPDC Working Buffer Address (EPDC_WB_ADDR)

EPDC Working Buffer Address

Address: 20F_4000h base + 30h offset = 20F_4030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

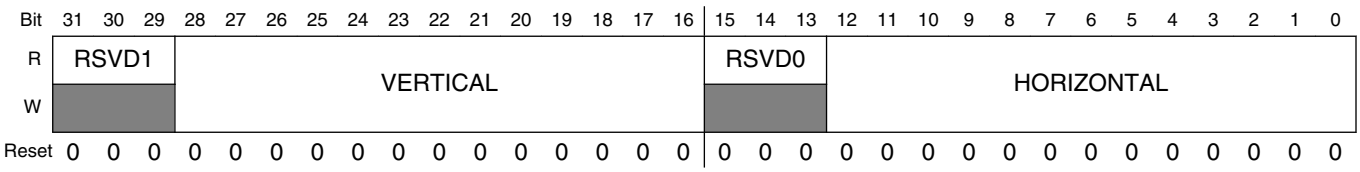
EPDC_WB_ADDR field descriptions

Field	Description
ADDR	Address for EPDC working buffer. This address must be a aligned to a 64-bit double-word boundary.

24.4.4 EPDC Screen Resolution (EPDC_RES)

EPDC Screen Resolution. This register defines the horizontal and vertical resolution of the target display panel

Address: 20F_4000h base + 40h offset = 20F_4040h



EPDC_RES field descriptions

Field	Description
31–29 RSVD1	Reserved.
28–16 VERTICAL	Vertical Resoltion (in pixels)
15–13 RSVD0	Reserved.
HORIZONTAL	Horizontal Resolution (in pixels)

24.4.5 EPDC Format Control Register (EPDC_FORMAT*n*)

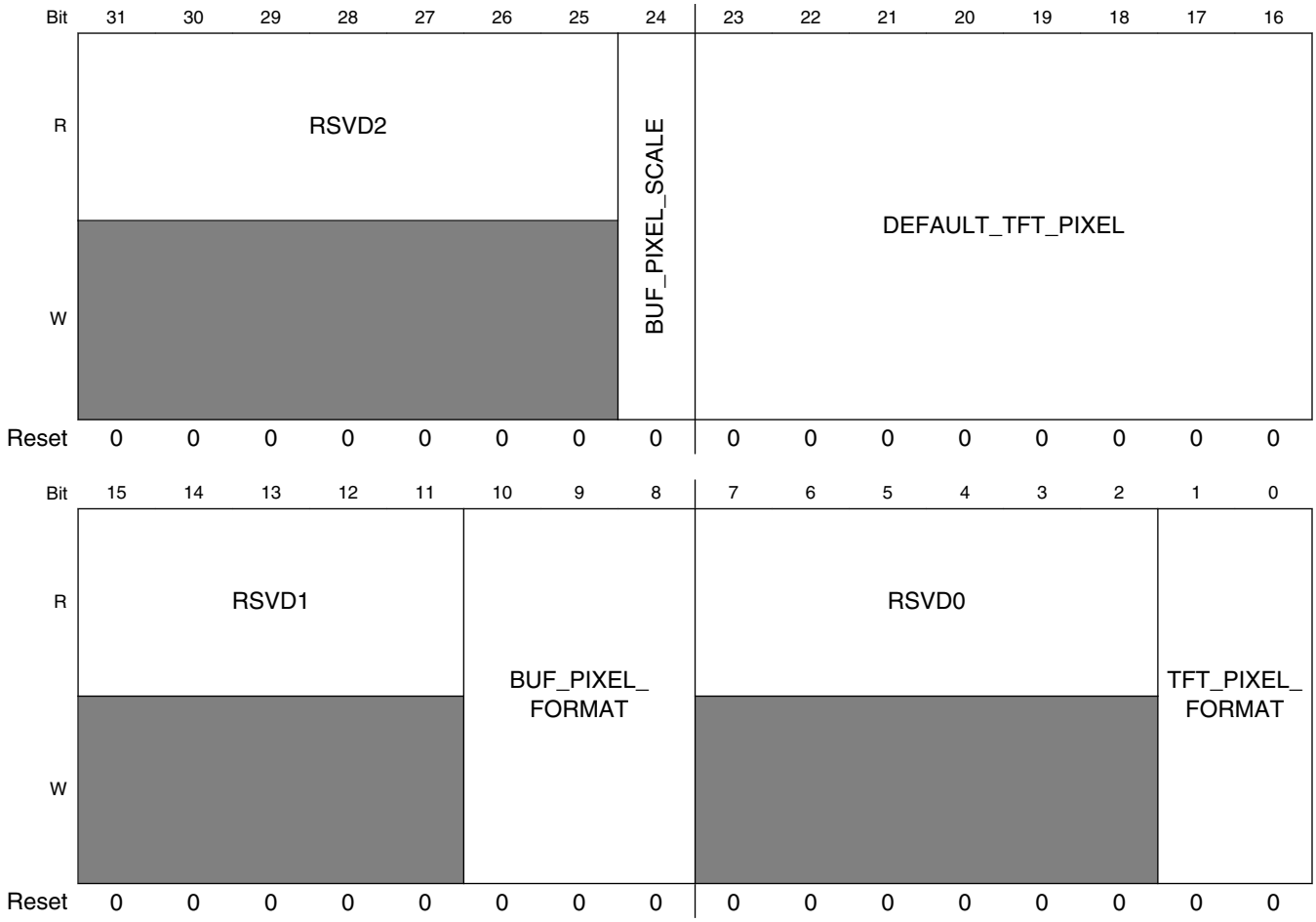
EPDC Pixel format control register. Defines formats for buffer and TFT pixels.

This register controls various functions throughout the digital portion of the chip.

EXAMPLE

```
/* EPDC_FORMAT - 2bit TFT and 4bit Buf pixel format */
reg_val = EPDC_FORMAT_TFT_PIXEL_FORMAT_2BIT
    | EPDC_FORMAT_BUF_PIXEL_FORMAT_P4N
    | ((0x0 << EPDC_FORMAT_DEFAULT_TFT_PIXEL_OFFSET) &
       EPDC_FORMAT_DEFAULT_TFT_PIXEL_MASK);
__raw_writel(reg_val, EPDC_FORMAT);
```

Address: 20F_4000h base + 50h offset + (4d × i), where i=0d to 3d



EPDC_FORMAT*n* field descriptions

Field	Description
31–25 RSVD2	Reserved.
24 BUF_PIXEL_ SCALE	Selects method of conversion from 8-bit input 0x0 **TRUNCATE** — Use Truncate method (LSB) 0x1 **ROUND** — Use rounding method (with saturation)
23–16 DEFAULT_TFT_ PIXEL	Default TFT pixel value. This value is used as the source-driver voltage value (TFT-pixel) for either partial-updates where a pixel has not changed or for any part of the screen which is not being updated during active frame scans.
15–11 RSVD1	Reserved.
10–8 BUF_PIXEL_ FORMAT	EPDC Input Buffer Pixel format. All update buffers are expected to have 8-bit grayscale pixels. This register defines which MSB's of those pixels are used. It must be noted that this format must match the waveform (e.g. P4N is not compatible with 3-bit waveforms) 0x2 **P2N** — 2-bit pixel 0x3 **P3N** — 3-bit pixel 0x4 **P4N** — 4-bit pixel 0x5 **P5N** — 5-bit pixel

Table continues on the next page...

EPDC_FORMAT*n* field descriptions (continued)

Field	Description
7–2 RSVD0	Reserved.
TFT_PIXEL_ FORMAT	EPDC TFT Pixel Format. This defines how many bits of the DATA bus are required per pixel. This field must be consistent with the waveform and panel architecture. 0x0 **2B** — 2-bit 0x1 **2BV** — 2-bit and VCOM 0x2 **4B** — 4-bit 0x3 **4BV** — 4-bit and VCOM

24.4.6 EPDC FIFO control register (EPDC_FIFOCTRL*n*)

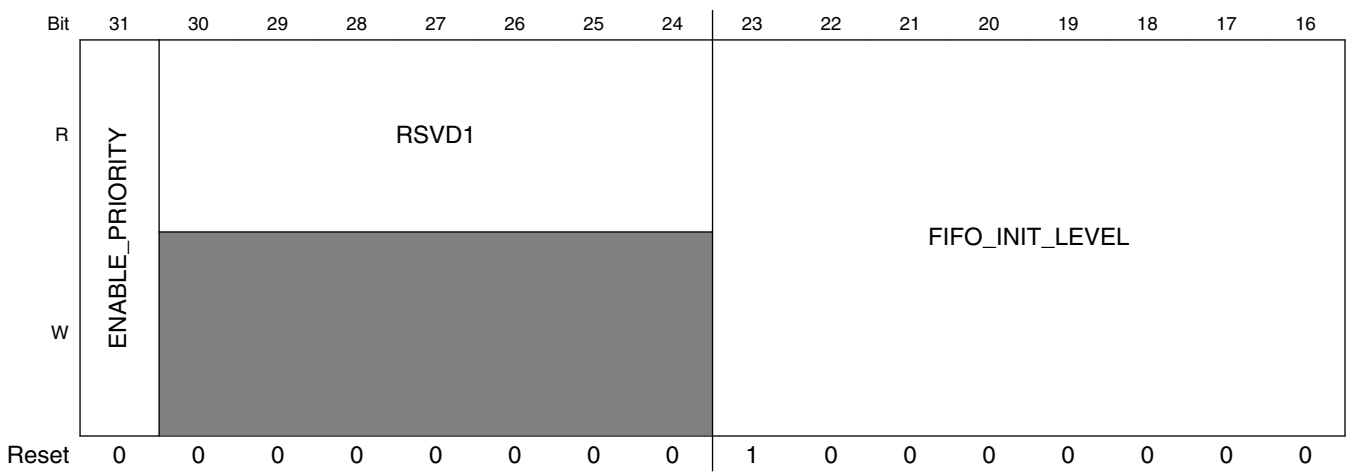
Allows for programmability of pixel FIFO watermarks used in conjunction with system arbitration hardware

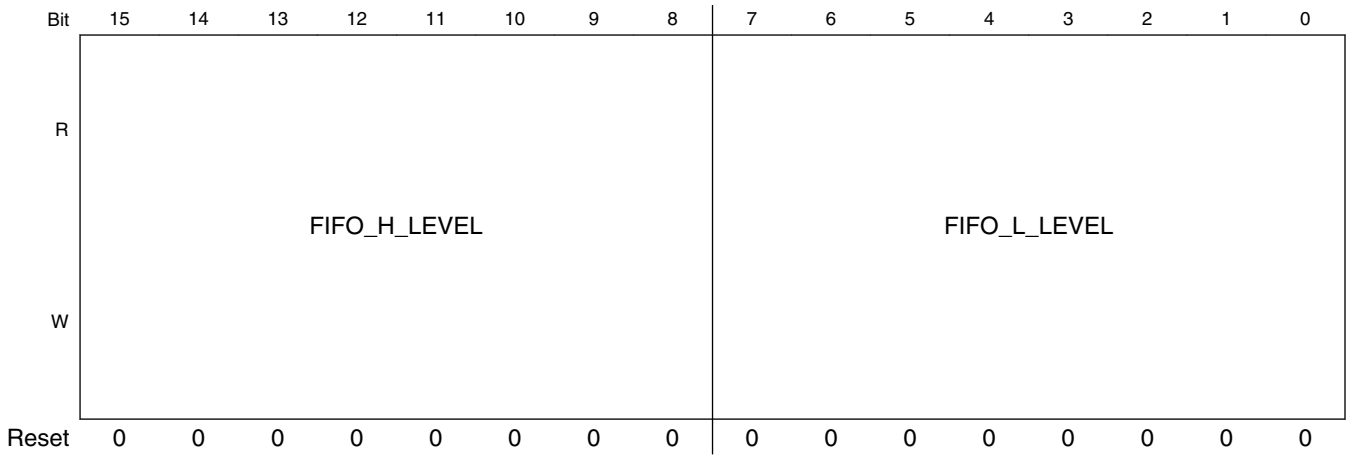
This register houses FIFO control bits

EXAMPLE

```
/* EPDC_FIFOCTRL (disabled) */
reg_val =
    ((100 << EPDC_FIFOCTRL_FIFO_INIT_LEVEL_OFFSET) &
     EPDC_FIFOCTRL_FIFO_INIT_LEVEL_MASK)
    | ((200 << EPDC_FIFOCTRL_FIFO_H_LEVEL_OFFSET) &
       EPDC_FIFOCTRL_FIFO_H_LEVEL_MASK)
    | ((100 << EPDC_FIFOCTRL_FIFO_L_LEVEL_OFFSET) &
       EPDC_FIFOCTRL_FIFO_L_LEVEL_MASK);
__raw_writel(reg_val, EPDC_FIFOCTRL);
```

Address: 20F_4000h base + A0h offset + (4d × i), where i=0d to 3d





EPDC_FIFOCTRL*n* field descriptions

Field	Description
31 ENABLE_ PRIORITY	Enable watermark-based priority elevation mechanism. 1=Enabled, 0=Disabled. (Only applies to FIFO_H_LEVEL and FIFO_L_LEVEL)
30–24 RSVD1	Reserved.
23–16 FIFO_INIT_ LEVEL	This register sets the watermark for the pixel-fifo.
15–8 FIFO_H_LEVEL	Upper level value of FIFO watermark. Must be greater than FIFO_L_LEVEL. When the pixel FIFO reaches this level or above, the priority elevation request is negated
FIFO_L_LEVEL	Lower level value of FIFO watermark. When the pixel FIFO reaches this level or below, the priority elevation request is asserted.

24.4.7 EPDC Update Region Address (EPDC_UPD_ADDR)

EPDC Update Region Address

The address alignment depends on whether the update region stride register (EPDC_UPD_STRIDE) is set or not.

If EPDC_UPD_STRIDE is zero, the update region address must start and end on the 8-byte aligned addresses.

If EPDC_UPD_STRIDE is non-zero, the update region address can start and/or end on any byte addressing. But aligned address still recomended be get best bus performance.

EXAMPLE1

```
/* no update stride case */
u32 stride = 0;
u32 addr = 0x100;    /* 8-byte aligned */
```

```
__raw_writel(stride, EPDC_UPD_STRIDE);
__raw_writel(addr, EPDC_UPD_ADDR);
```

EXAMPLE2

```
/* update stride = 345 */
u32 stride = 345;
u32 addr = 0x123;    /* 8-byte unaligned is OK */
__raw_writel(stride, EPDC_UPD_STRIDE);
__raw_writel(addr, EPDC_UPD_ADDR);
```

Address: 20F_4000h base + 100h offset = 20F_4100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_UPD_ADDR field descriptions

Field	Description
ADDR	Address for incoming region update. This address points to update region which will be processed into the working buffer.

24.4.8 EPDC Update Region Stride (EPDC_UPD_STRIDE)

EPDC Update Region Stride

When UPD_STRIDE==0 (stride feature disabled), UPD buffer line must start from 64-bit boundary and end on 64-bit boundary(padding if not). When UPD_STRIDE!=0 (stride feature enabled), UPD buffer line can start or end on any byte address, UPD_WIDTH should be set to real line bytes count as normal, while UPD_STRIDE set to byte distance between two lines' start.

EXAMPLE

```
see details on stride feature introduction
```

Address: 20F_4000h base + 110h offset = 20F_4110h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STRIDE																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_UPD_STRIDE field descriptions

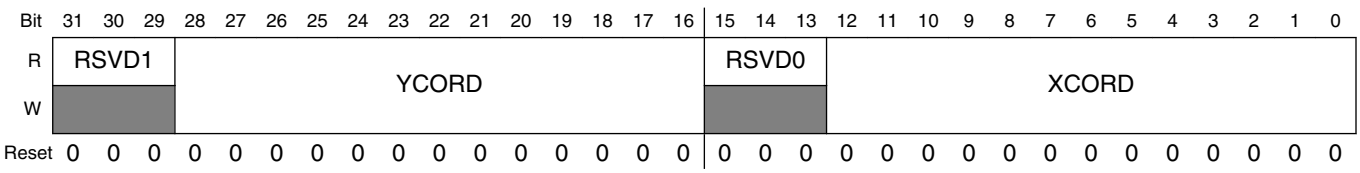
Field	Description
STRIDE	line stride for incoming region update

24.4.9 EPDC Update Command Co-ordinate (EPDC_UPD_CORD)

EPDC Update Command Co-ordinate

This register is used for setting up the coordinate of a new update region.

Address: 20F_4000h base + 120h offset = 20F_4120h



EPDC_UPD_CORD field descriptions

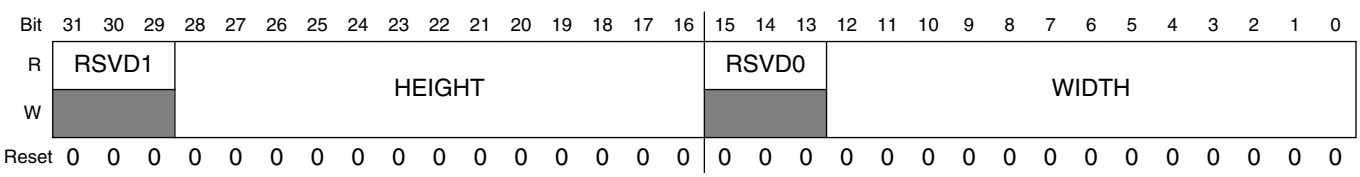
Field	Description
31–29 RSVD1	Reserved.
28–16 YCORD	Y co-ordinate for incoming region update
15–13 RSVD0	Reserved.
XCORD	X co-ordinate for incoming region update

24.4.10 EPDC Update Command Size (EPDC_UPD_SIZE)

EPDC Update Command Size

This register sets up the size of an update region.

Address: 20F_4000h base + 140h offset = 20F_4140h



EPDC_UPD_SIZE field descriptions

Field	Description
31–29 RSVD1	Reserved.
28–16 HEIGHT	Height (in pixels)

Table continues on the next page...

EPDC_UPD_SIZE field descriptions (continued)

Field	Description
15–13 RSVD0	Reserved.
WIDTH	Width (in pixels)

24.4.11 EPDC Update Command Control (EPDC_UPD_CTRL*n*)

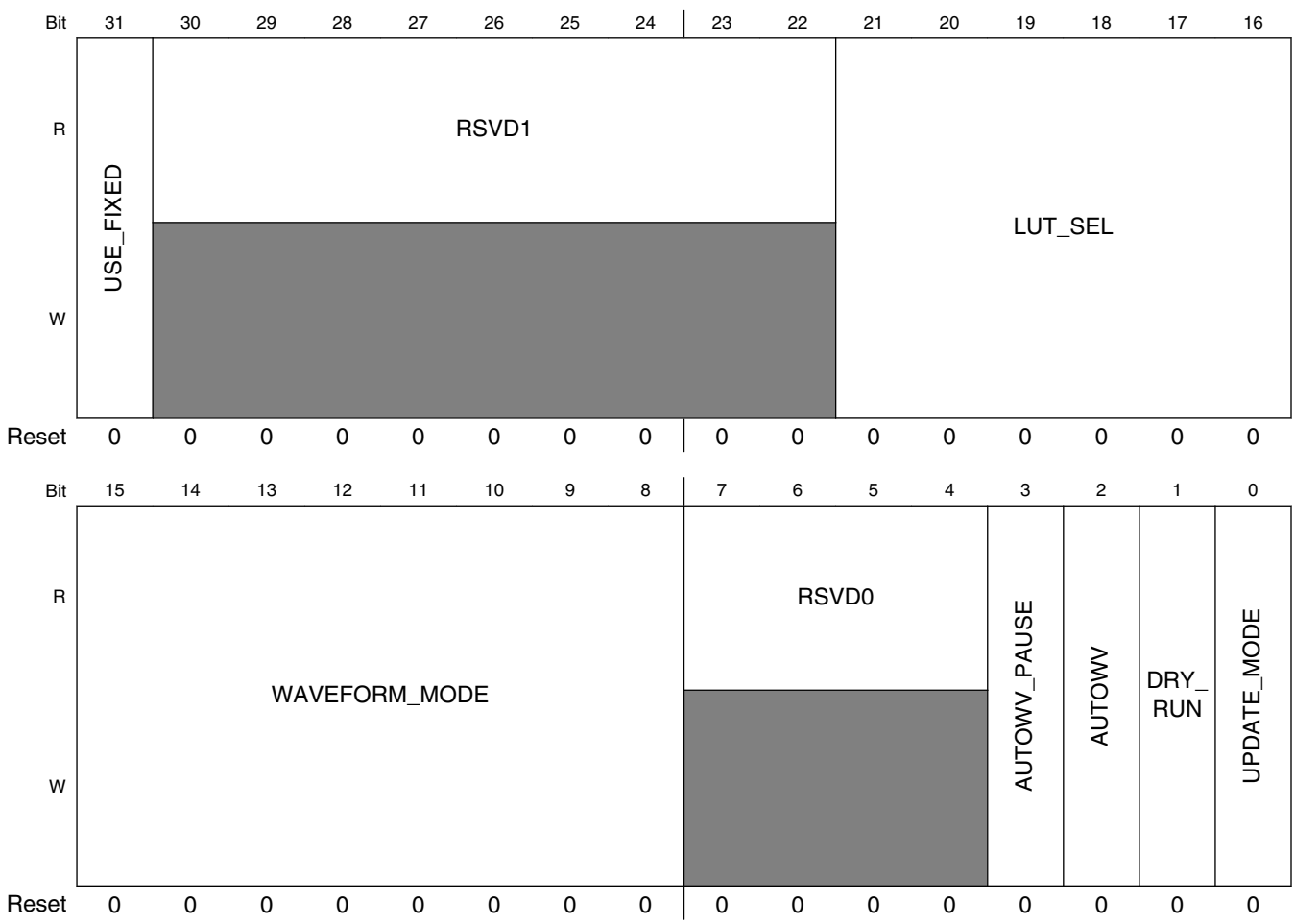
EPDC Update Command Control. Writing to this registers triggers and update request operation.

Update control register

EXAMPLE

```
HW_DIGCTL_CTRL_CLR(BM_DIGCTL_CTRL_USB_CLKGATE);  // enable USB clock
```

Address: 20F_4000h base + 160h offset + (4d × i), where i=0d to 3d



EPDC_UPD_CTRL*n* field descriptions

Field	Description
31 USE_FIXED	Use fixed pixel values (requires programming of HW_EPDC_UPD_FIXED)
30–22 RSVD1	Reserved.
21–16 LUT_SEL	LUT select 0-63
15–8 WAVEFORM_ MODE	Waveform Mode 0-255
7–4 RSVD0	Reserved.
3 AUTOWV_ PAUSE	0x0 **AUTO** — epdc will analyze update buffer, report histogram, then update waveform mode using the programmed mode mapping in AUTOWV_LUT and start LUT loading 0x1 **MANUAL** — epdc will analyze update buffer, report histogram, then pause and waiting software to write again with selected waveform mode to start lut loading
2 AUTOWV	enable automatical waveform mode selection
1 DRY_RUN	Enable Dry Run mode(set to 1). WB won't be updated in this mode, and lut_sel will be ignored, so actually you don't need to wait for LUT available to use this feature
0 UPDATE_MODE	Update Mode 0x0 **PARTIAL** — Partial Update : only process changed pixels in region 0x1 **FULL** — Full Update : process all pixels in region

24.4.12 EPDC Update Fixed Pixel Control (EPDC_UPD_FIXED*n*)

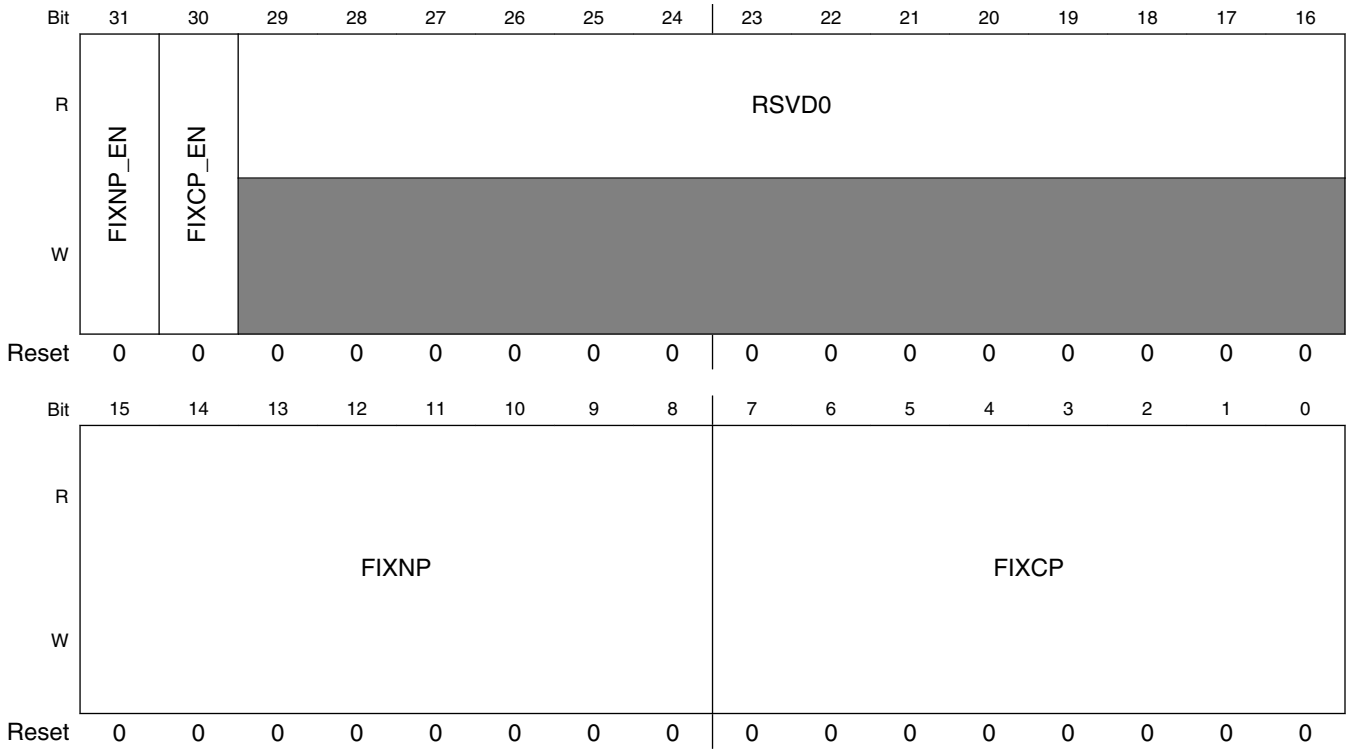
EPDC Update Control register for fixed-pixel updates (enabled via HW_EPDC_UPD_CTRL[USE_FIXED])

Update control register for fixed-pixel update values

EXAMPLE

```
TBD.
```

Address: 20F_4000h base + 180h offset + (4d × i), where i=0d to 3d



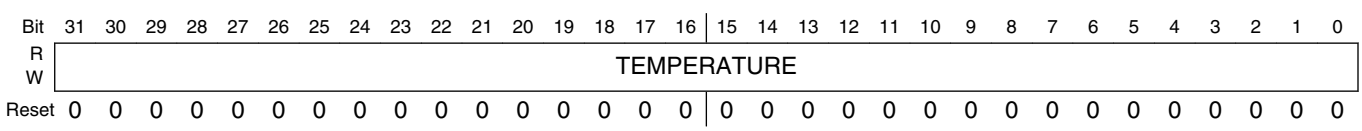
EPDC_UPD_FIXED*n* field descriptions

Field	Description
31 FIXNP_EN	If set to 1, current updated region has the NP value defined by FIXNP
30 FIXCP_EN	If set to 1, current updated region has the CP value defined by FIXCP
29–16 RSVD0	Reserved.
15–8 FIXNP	NP value if fixenp_en is set to 1. Data in Y8 format.
FIXCP	CP value if fixecp_en is set to 1. Data in Y8 format.

24.4.13 EPDC Temperature Register (EPDC_TEMP)

EPDC Temperature Compensation Register

Address: 20F_4000h base + 1A0h offset = 20F_41A0h



EPDC_TEMP field descriptions

Field	Description
TEMPERATURE	Temperature Value. This value is simply an index (not a temperature value). The index is used by the EPDC to access the correct temperature compensated waveform.

24.4.14 Waveform Mode Lookup Table Control Register. (EPDC_AUTOWV_LUT)

This register is used to access the waveform mode lookup table.

DATA -> AUTOWV_LUT[ADDR] : Writing this reg with 'ADDR' and 'DATA' info will get 'DATA' written to AUTOWV_LUT mem indexed with 'ADDR'

Address: 20F_4000h base + 1C0h offset = 20F_41C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD1								DATA								RSVD0								ADDR							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_AUTOWV_LUT field descriptions

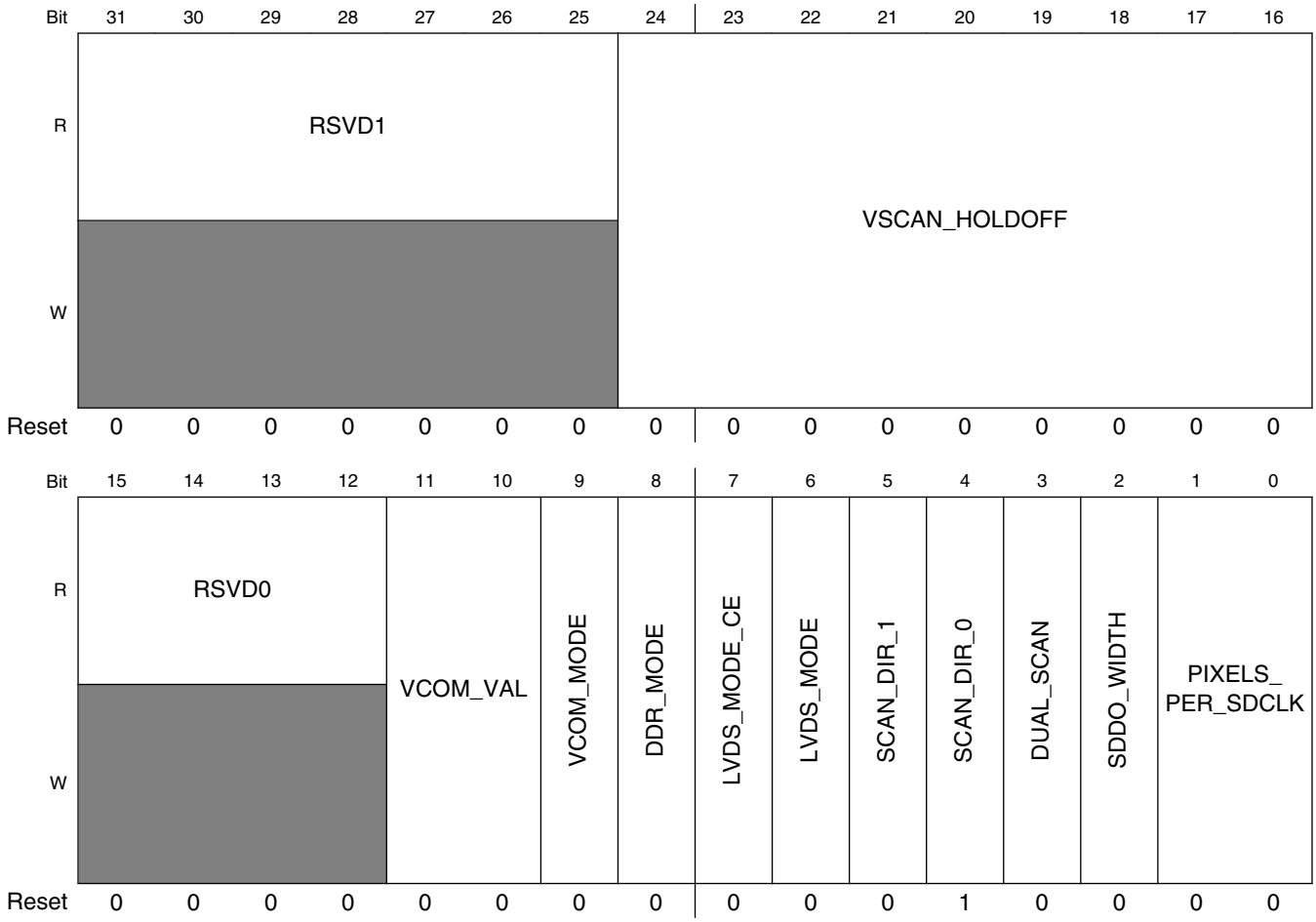
Field	Description
31–24 RSVD1	Reserved, always set to zero.
23–16 DATA	DATA
15–8 RSVD0	Reserved, always set to zero.
ADDR	ADDR

24.4.15 EPDC Timing Control Engine Control Register (EPDC_TCE_CTRL*n*)

TCE general control register

This register houses Horizontal scan timing. Note that line data length is derived from HW_EPDC_RES.

Address: 20F_4000h base + 200h offset + (4d × i), where i=0d to 3d



EPDC_TCE_CTRL*n* field descriptions

Field	Description
31–25 RSVD1	Reserved.
24–16 VSCAN_ HOLDOFF	This period (expressed in vertical lines), sets the portion of the vertical blanking available for new LUTs to be activated. The remainder of the blanking period is reserved for pre-filling the TCE pixel FIFOs. Increasing this value allows for multiple smaller updates to be intercepted by the current frame scan. This number should not exceed FRAME_END+FRAME_SYNC+FRAME_BEGIN. Increasing this value can improve the ability for any given update to intercept the next available frame-scan. Excessive values can result in TCE FIFO under-runs.
15–12 RSVD0	Reserved.
11–10 VCOM_VAL	When VCOM_MODE = MANUAL, this value is used to manually set the VCOM value for the VCOM[1:0] pins
9 VCOM_MODE	This field determines the method used to drive the VCOM signal. 0x0 **MANUAL** — VCOM Value is set manually using VCOM_VAL field 0x1 **AUTO** — VCOM Value is used from waveform
8 DDR_MODE	If set, DATA data is driven on both positive and negative edges of SDCLK. Note that this mode is not supported when SDDO_BUS_FORMAT=16BIT and LVDS is not used. This must always be set when LVDS_MODE is set.

Table continues on the next page...

EPDC_TCE_CTRL*n* field descriptions (continued)

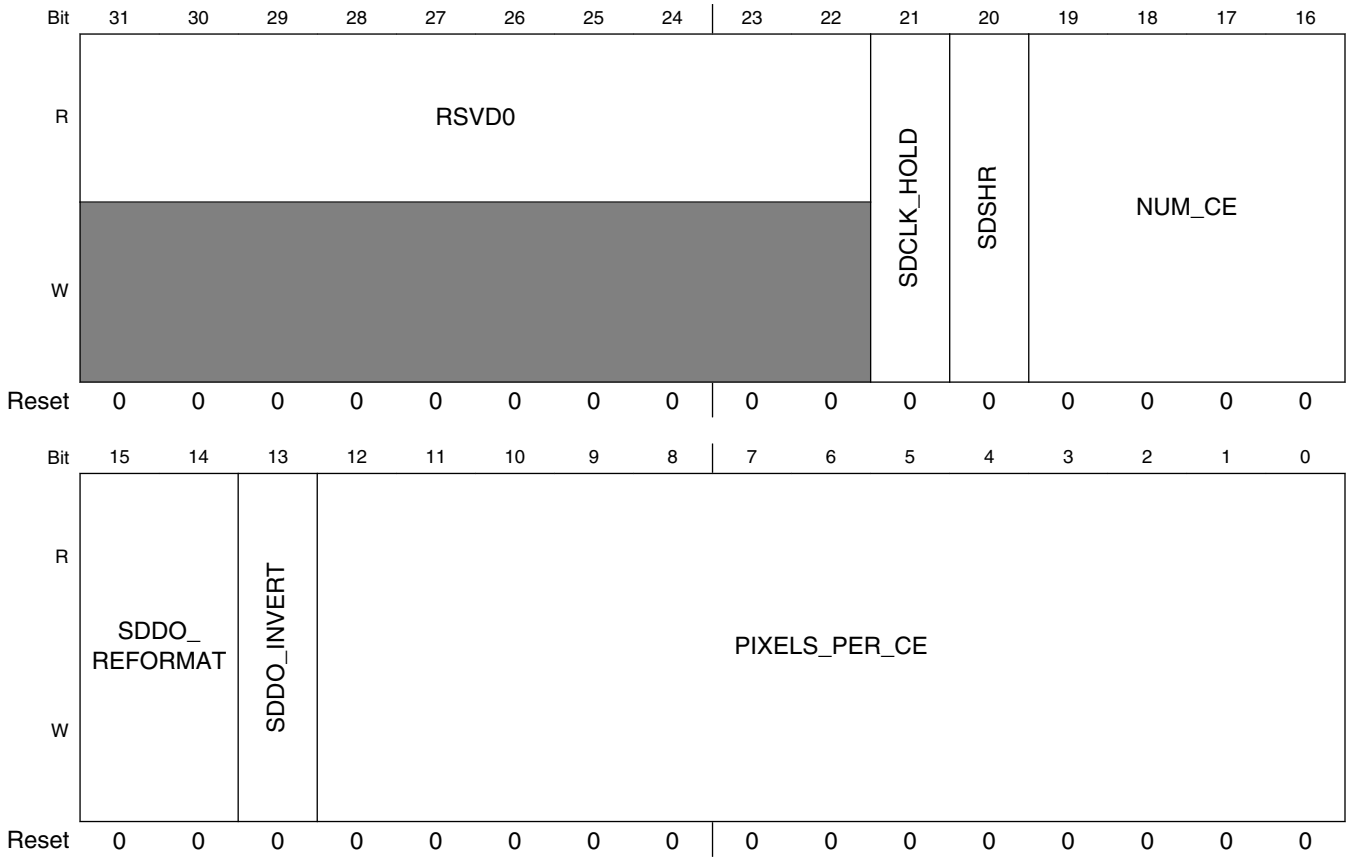
Field	Description
7 LVDS_MODE_CE	If set (together with LVDS_MODE=1), SDCE[9:5] shall be driven as the differential inverse of SDCE[4:0]. In this mode the EPDC only supports 5 CE lines.
6 LVDS_MODE	If set, the upper 8-bit of the DATA bus are used for LVDS differential signalling. Note that this can only be used when SDDO_BUS_FORMAT is set to 16BIT, i.e. LVDS signaling is not supported with an 8-bit DATA interface. Note that for LVDS_MODE, DDR_MODE must also be set.
5 SCAN_DIR_1	Determines scan direction for each half of the TFT panel 0x1 **UP** — Scan this region from bottom to top 0x0 **DOWN** — Scan this region from top to bottom
4 SCAN_DIR_0	Determines scan direction for each half of the TFT panel 0x1 **UP** — Scan this region from bottom to top 0x0 **DOWN** — Scan this region from top to bottom
3 DUAL_SCAN	Enables dual scan-mode. Note in this mode, SDDO_BUS_FORMAT=16BIT must be selected. and PIXELS_PER_CLK applies to each 8-bit segment of the DATA bus.
2 SDDO_WIDTH	Selects either 8 or 16 bit DATA bus format 0x0 **8BIT** — Connect to 8-bit source driver 0x1 **16BIT** — Connct to 16-bit source driver
PIXELS_PER_SDCLK	Number of TFT pixels per SDCLK period. Note that this value forms the division of the PIXLK to generate the SDCLK such that SDCLK = PIXCLK/PIXELS_PER_SDCLK. For dual-scan mode (DUAL_SCAN=1), this applies to 8-bit half of the 16-bit DATA. It should be noted that when DDR_MODE is enabled, both edges of the clock have to be accounted for in this value, so for example with an 8-bit DATA, 2-bit TFT pixel, this field should be set to EIGHT (four pixels on the pos-edge and 4-pixels on the neg-edge) 0x0 **RESERVED** — Reserved 0x1 **TWO** — Two TFT-pixels per SDCLK 0x2 **FOUR** — Four TFT-pixels per SDCLK 0x3 **EIGHT** — Eight TFT-pixels per SDCLK

24.4.16 EPDC Timing Control Engine Source-Driver Config Register (EPDC_TCE_SDCFG*n*)

Source-driver configuration register

This register houses source-driver configuration.

Address: 20F_4000h base + 220h offset + (4d × i), where i=0d to 3d



EPDC_TCE_SDCFG*n* field descriptions

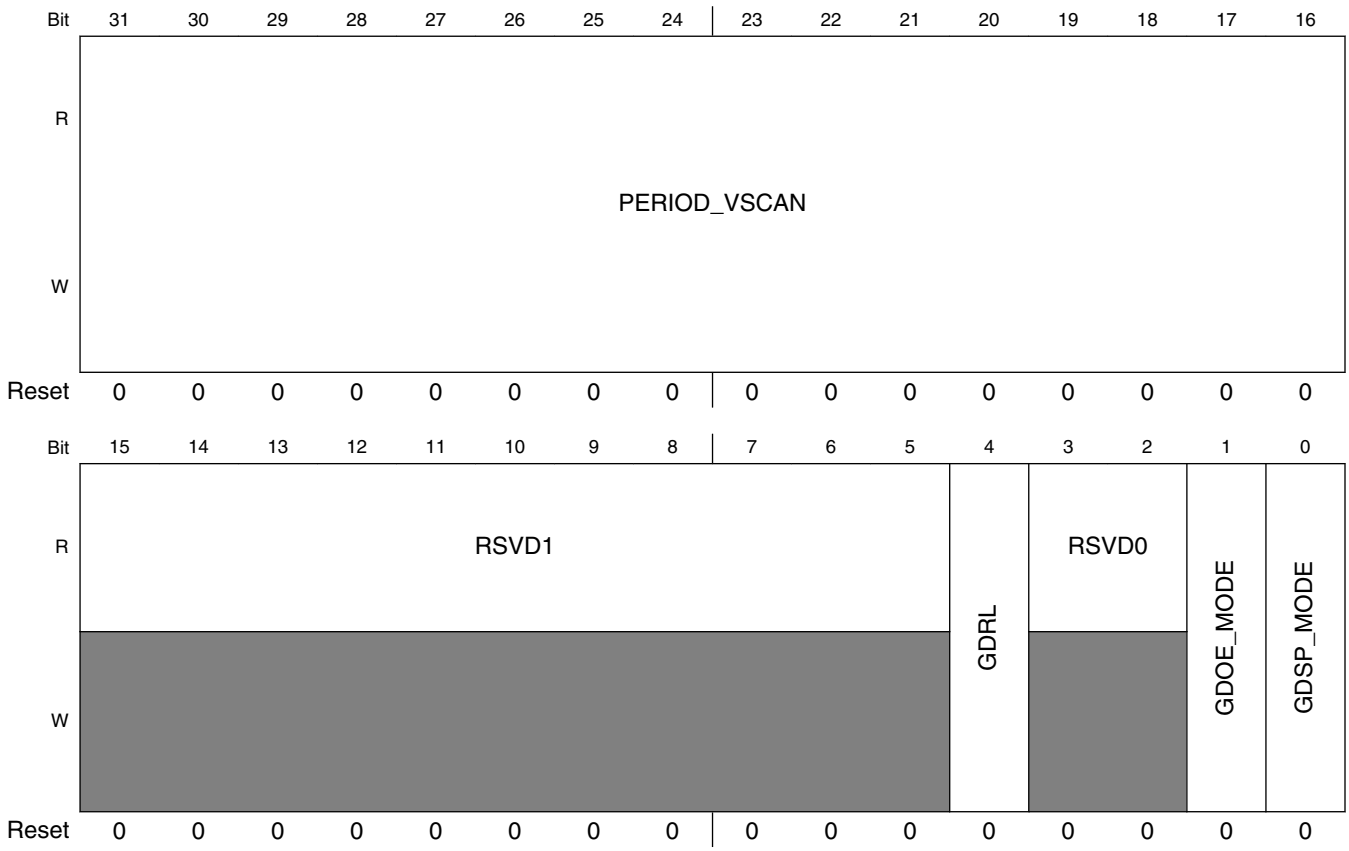
Field	Description
31–22 RSVD0	Reserved.
21 SDCLK_HOLD	Setting this bit to 1 holds the SDCLK low during LINE_BEGIN
20 SDSHR	Value for source-driver shift direction output port
19–16 NUM_CE	Number of source driver IC chip-enables. Must be 1-10
15–14 SDDO_ REFORMAT	This register defines the various re-formatting options to enable more flexibility in the source-driver interface: 0x0 **STANDARD** — No change. 0x1 **FLIP_PIXELS** — Reverses the order of the pixels on DATA. This register setting is sensitive to the TFT pixel width (TFT_PIXEL_FORMAT), e.g. for TFT_PIXEL_FORMAT=2B on an 8-bit bus P3,P2,P1,P0 becomes P0,P1,P2,P3, whereas with TFT_PIXEL_FORMAT=4B, on an 8-bit bus, P1,P0 becomes P0,P1
13 SDDO_INVERT	Setting this bit to 1 reverses the polarity of each DATA bit so 0xAAAA in 16-bit mode for example becomes 0x5555. This setting can be made in addition to the SDDO_REFORMAT register setting.
PIXELS_PER_ CE	Number of pixels (outputs) per source-driver IC. Please note that HW_EPDC_RES[HORIZONTAL] must be an integer multiple of PINS_PER_CE.

24.4.17 EPDC Timing Control Engine Gate-Driver Config Register (EPDC_TCE_GDCFG*n*)

Source-driver configuration register

This register houses gate-driver configuration.

Address: 20F_4000h base + 240h offset + (4d × i), where i=0d to 3d



EPDC_TCE_GDCFG*n* field descriptions

Field	Description
31–16 PERIOD_VSCAN	when vscan state is splited, this reg defines the counter period
15–5 RSVD1	Reserved.
4 GDRL	Value for gate-driver right/left shift output port
3–2 RSVD0	Reserved.
1 GDOE_MODE	Selects method for driving GDOE signal. When set to 0, GDOE is driven at all times during the frame-scan except FRAME_SYNC. When set to 1, GDOE is driven as a delayed version of GDCLK delayed by HW_EPDC_TCE_TIMING3[GDOE_OFFSET].

Table continues on the next page...

EPDC_TCE_GDCFG*n* field descriptions (continued)

Field	Description
0 GDSP_MODE	Selects method for driving GDSP pulse. When set to 0, GDSP is is always fixed to have a pulse width of one line-time. When set to 1, GDSP has a pulse-width determined by the FRAME_SYNC setting. Note that GDSP_MODE=1 is not compatible with the GDSP_OFFSET function

24.4.18 EPDC Timing Control Engine Horizontal Timing Register 1 (EPDC_TCE_HSCAN1*n*)

Horizontal scan timing registers. Note that all timing values are expressed in terms of the EPDC's internal PIXCLK, which depending on the PIXELS_PER_SDCLK register setting is either 2:1 or 4:1

This register houses Horizontal scan timing. Note that line data length is derived from HW_EPDC_RES.

Address: 20F_4000h base + 260h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD1				LINE_SYNC_WIDTH												RSVD0				LINE_SYNC											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_TCE_HSCAN1*n* field descriptions

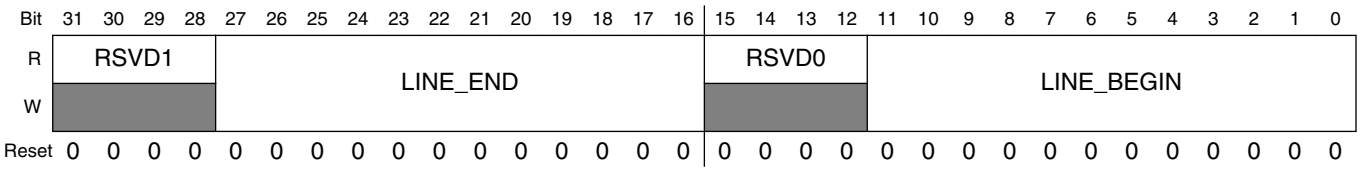
Field	Description
31–28 RSVD1	Reserved.
27–16 LINE_SYNC_ WIDTH	Number of PIXCLK cycles for the SDLE active time. Note that this value cannot be larger than LINE_SYNC and must be greater than 0. Typically it is recommended to set this value to be the same as LINE_SYNC
15–12 RSVD0	Reserved.
LINE_SYNC	Number of PIXCLK cycles for line sync duration. Note that this value encompasses the LINE_SYNC_WIDTH duration. This value must be programmed to a multiple of SDCLK cycles

24.4.19 EPDC Timing Control Engine Horizontal Timing Register 2 (EPDC_TCE_HSCAN2*n*)

Horizontal scan timing registers. Note that all timing values are expressed in terms of the EPDC's internal PIXCLK, which depending on the PIXELS_PER_SDCLK register setting is either 2:1 or 4:1

This register houses Horizontal scan timing. Note that line data length is derived from HW_EPDC_RES.

Address: 20F_4000h base + 280h offset + (4d × i), where i=0d to 3d



EPDC_TCE_HSCAN2*n* field descriptions

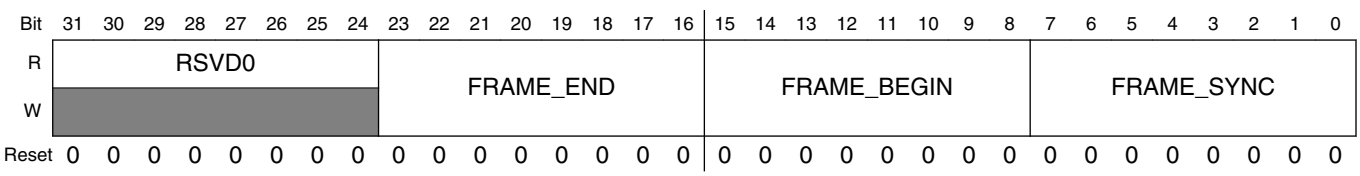
Field	Description
31–28 RSVD1	Reserved.
27–16 LINE_END	Number of PIXCLK cycles for line end duration. This defines the duration from the de-assertion of SDCE and assertion of the next SDLE.
15–12 RSVD0	Reserved.
LINE_BEGIN	Number of PIXCLK cycles for line begin duration. This defines the interval between de-assertion of SDLE and assertion of the SDCE signals. This value must be programmed to a multiple of SDCLK cycles

24.4.20 EPDC Timing Control Engine Vertical Timing Register (EPDC_TCE_VSCAN*n*)

Vertical scan timing registers

This register houses vertical scan timing. Note that frame data length is derived from HW_EPDC_RES.

Address: 20F_4000h base + 2A0h offset + (4d × i), where i=0d to 3d



EPDC_TCE_VSCAN*n* field descriptions

Field	Description
31–24 RSVD0	Reserved.
23–16 FRAME_END	Number of lines for frame end duration.
15–8 FRAME_BEGIN	Number of lines for frame begin duration.
FRAME_SYNC	Number of lines for frame sync duration.

24.4.21 EPDC Timing Control Engine OE timing control Register (EPDC_TCE_OE*n*)

This register contain delay programming values for the SDOEZ and SDOED source driver control signals

This register contain delay programming values for the SDOZ and SDOE source driver control signals

Address: 20F_4000h base + 2C0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SDOED_WIDTH								SDOED_DLY								SDOEZ_WIDTH								SDOEZ_DLY							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_TCE_OE*n* field descriptions

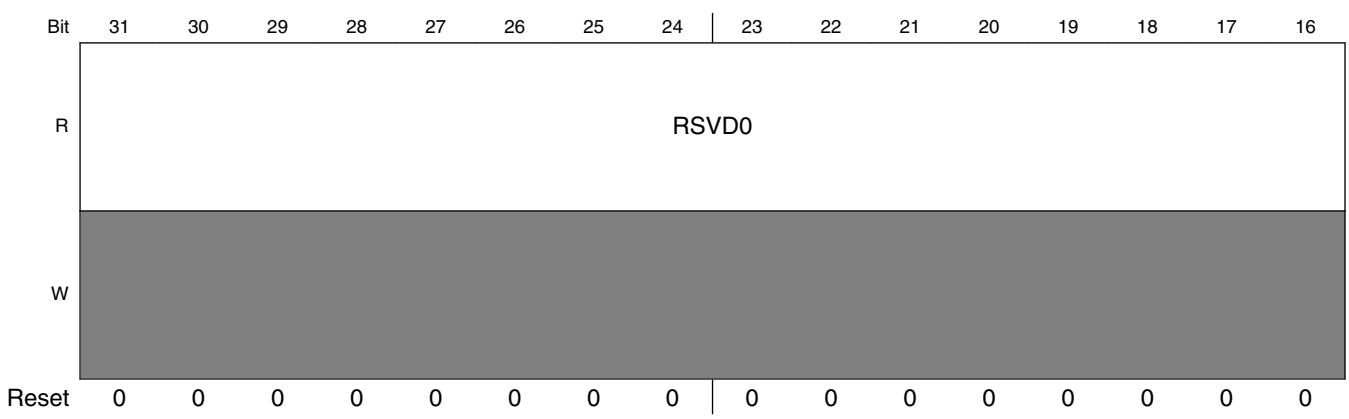
Field	Description
31–24 SDOED_WIDTH	Number of PIXCLK cycles from SDOED high to SDOED falling (Must be greater than 0)
23–16 SDOED_DLY	Number of PIXCLK cycles from SDOEZ low to SDOED rising (Must be greater than 0)
15–8 SDOEZ_WIDTH	Number of PIXCLK cycles from SDOEZ high to SDOEZ falling (Must be greater than 0)
SDOEZ_DLY	Number of PIXCLK cycles from SDLE falling edge to SDOEZ rising (Must be greater than 0)

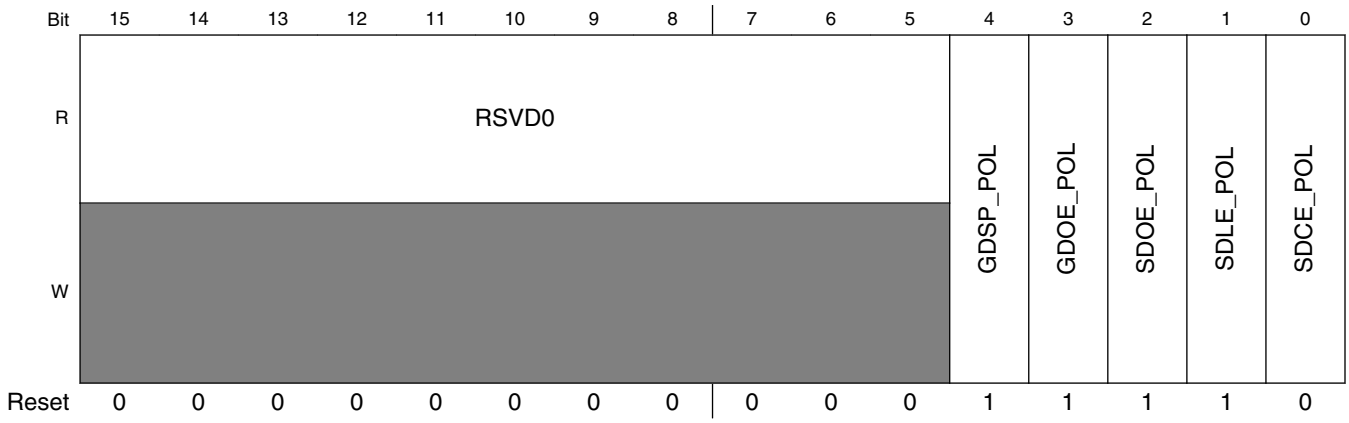
24.4.22 EPDC Timing Control Engine Driver Polarity Register (EPDC_TCE_POLARITY)

This registers allows for programming the polarity of source/gate driver control signals

This register houses FIFO control bits

Address: 20F_4000h base + 2E0h offset = 20F_42E0h





EPDC_TCE_POLARITY field descriptions

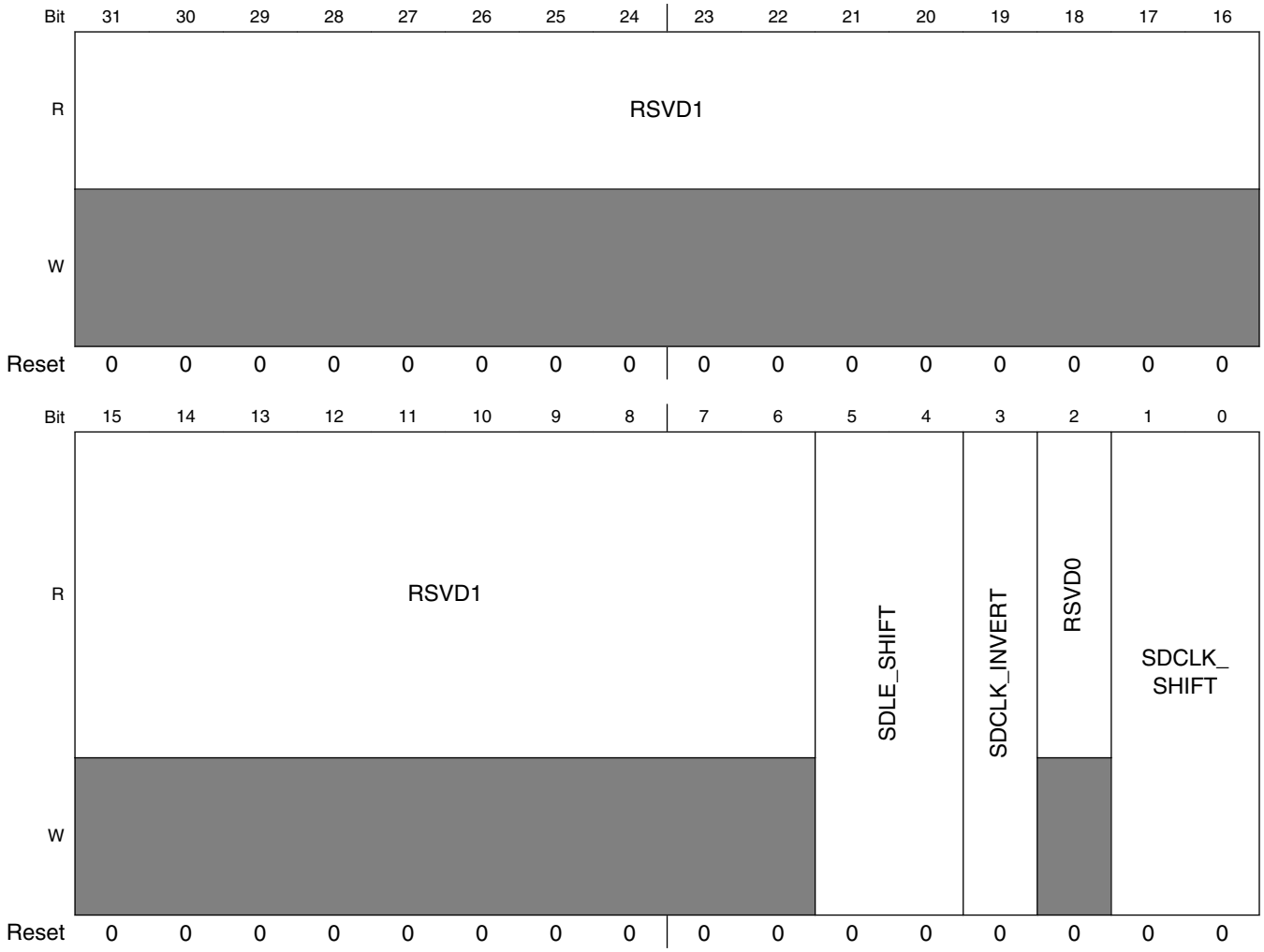
Field	Description
31–5 RSVD0	Reserved.
4 GDSP_POL	0 = Active Low, 1 = Active High. Applies to the GDSP output
3 GDOE_POL	0 = Active Low, 1 = Active High. Applies to the GDOE output
2 SDOE_POL	0 = Active Low, 1 = Active High. Applies to the SDOE. Does not apply to SDOEZ and SDOED outputs
1 SDLE_POL	0 = Active Low, 1 = Active High. Applies to the SDLE output
0 SDCE_POL	0 = Active Low, 1 = Active High. Applies to all 10 SDCE outputs

24.4.23 EPDC Timing Control Engine Timing Register 1 (EPDC_TCE_TIMING1*n*)

This register contains various timing adjustment controls

This register houses general purpose timing adjustment registers

Address: 20F_4000h base + 300h offset + (4d × i), where i=0d to 3d



EPDC_TCE_TIMING1*n* field descriptions

Field	Description
31–6 RSVD1	Reserved.
5–4 SDLE_SHIFT	This register can be used to implement additional timing setup/hold adjustment of source driver signals by adjusting the SDCLK up to 3 PIXCLK cycles 0x0 **NONE** — No shift of SDLE 0x1 **ONE** — Shift SDLE 1 pixclk cycle 0x2 **TWO** — Shift SDLE 2 pixclk cycles 0x3 **THREE** — Shift SDLE 3 pixclk cycles
3 SDCLK_INVERT	Invert phase of SDCLK
2 RSVD0	Reserved.
SDCLK_SHIFT	This register can be used to implement additional timing setup/hold adjustment of source driver signals by adjusting the SDCLK up to 4 cycles

Table continues on the next page...

EPDC_TCE_TIMING1*n* field descriptions (continued)

Field	Description
	0x0 **NONE** — No shift of SDCLK 0x1 **ONE** — Shift SDCLK 1 pixclk cycle 0x2 **TWO** — Shift SDCLK 2 pixclk cycles 0x3 **THREE** — Shift SDCLK 3 pixclk cycles

24.4.24 EPDC Timing Control Engine Timing Register 2 (EPDC_TCE_TIMING2*n*)

This register contains various timing adjustment controls

This register houses general purpose timing adjustment registers

Address: 20F_4000h base + 310h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GDCLK_HP																GDSP_OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

EPDC_TCE_TIMING2*n* field descriptions

Field	Description
31–16 GDCLK_HP	This register controls the GDCLK high-pulse width. It is expressed by N PIXCLKs where N=1 to 65535. Note that GDCLK will always have a period equal to the line-clock timing. A value of 0 is not supported. It is recommended that this value be set to at least a half line-clock time. For panels which use GDCLK to drive GDOE, this high-pulse width should be set to cover tha majority of the line timing
GDSP_OFFSET	This register allows the user to shift the GDSP pulse by N PIXCLKs where N=1 to 65535. Note that GDSP will always have a pulse width equivalent to the line-clock timing. A value of 0 is not supported.

24.4.25 EPDC Timing Control Engine Timing Register 3 (EPDC_TCE_TIMING3*n*)

This register contains various timing adjustment controls

This register houses general purpose timing adjustment registers

Address: 20F_4000h base + 320h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GDOE_OFFSET																GDCLK_OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

EPDC_TCE_TIMING3*n* field descriptions

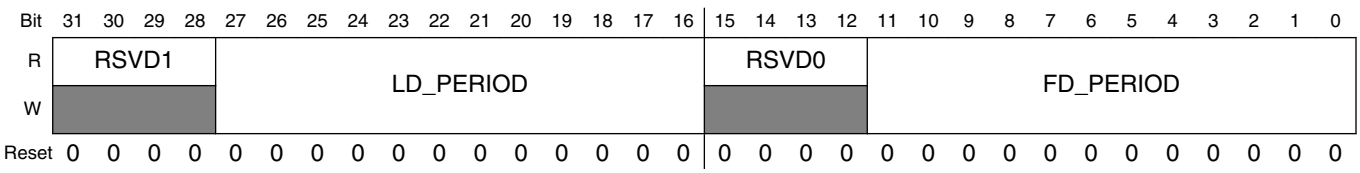
Field	Description
31–16 GDOE_OFFSET	When using GDOE_MODE=1, this register sets the delay from GDCLK to the GDOE in terms of N PIXCLK cycles
GDCLK_ OFFSET	This register allows the user to shift the GDCLK from the line time by N PIXCLK cycles.

24.4.26 EPDC Pigeon Mode Control Register 0 (EPDC_PIGEON_CTRL0*n*)

This register contains global counter settings for Pigeon Mode

This register houses general purpose timing adjustment registers

Address: 20F_4000h base + 380h offset + (4d × i), where i=0d to 3d



EPDC_PIGEON_CTRL0*n* field descriptions

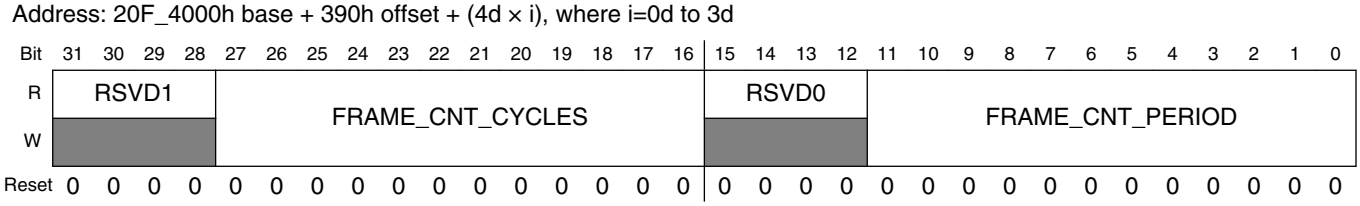
Field	Description
31–28 RSVD1	Reserved.
27–16 LD_PERIOD	period of pclk counter during LD phase
15–12 RSVD0	Reserved.
FD_PERIOD	period of line counter during FD phase

24.4.27 EPDC Pigeon Mode Control Register 1 (EPDC_PIGEON_CTRL1*n*)

This register contains global counter setting for pigeon mode

This register houses general purpose timing adjustment registers

Address: 20F_4000h base + 390h offset + (4d × i), where i=0d to 3d



EPDC_PIGEON_CTRL1*n* field descriptions

Field	Description
31–28 RSVD1	Reserved.
27–16 FRAME_CNT_ CYCLES	max cycles of frame counter
15–12 RSVD0	Reserved.
FRAME_CNT_ PERIOD	period of frame counter

24.4.28 EPDC IRQ Mask Register for LUT 0~31 (EPDC_IRQ_MASK1*n*)

Controls masking EPDC LUT complete interrupts

This register controls LUT0~31 IRQ masks for EPDC interrupts

Address: 20F_4000h base + 3C0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUTN_CMPLT_IRQ_EN																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_IRQ_MASK1*n* field descriptions

Field	Description
LUTN_CMPLT_ IRQ_EN	LUT0~31 Complete Interrupt Enable

24.4.29 EPDC IRQ Mask Register for LUT 32~63 (EPDC_IRQ_MASK2*n*)

Controls masking EPDC LUT complete interrupts

This register controls LUT0~31 IRQ masks for EPDC interrupts

Address: 20F_4000h base + 3D0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUTN_CMPLT_IRQ_EN																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_IRQ_MASK2*n* field descriptions

Field	Description
LUTN_CMPLT_IRQ_EN	LUT32~64 Complete Interrupt Enable

24.4.30 EPDC Interrupt Register for LUT 0~31 (EPDC_IRQ1*n*)

EPDC LUT Completion IRQs. The IRQ for a specific LUT is triggered when it's corrrsponding physical update is competed on the screen. Each interrupt has a corresponding mask register in HW_EPDC_IRQ_MASK

This register houses the interrupt bits for the LUT Completions

Address: 20F_4000h base + 3E0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUTN_CMPLT_IRQ																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_IRQ1*n* field descriptions

Field	Description
LUTN_CMPLT_IRQ	LUT 0~31 Complete Interrupt

24.4.31 EPDC Interrupt Registerr for LUT 32~63 (EPDC_IRQ2*n*)

EPDC LUT Completion IRQs. The IRQ for a specific LUT is triggered when it's corrrsponding physical update is competed on the screen. Each interrupt has a corresponding mask register in HW_EPDC_IRQ_MASK

This register houses the interrupt bits for the LUT Completions

Address: 20F_4000h base + 3F0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUTN_CMPLT_IRQ																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_IRQ2*n* field descriptions

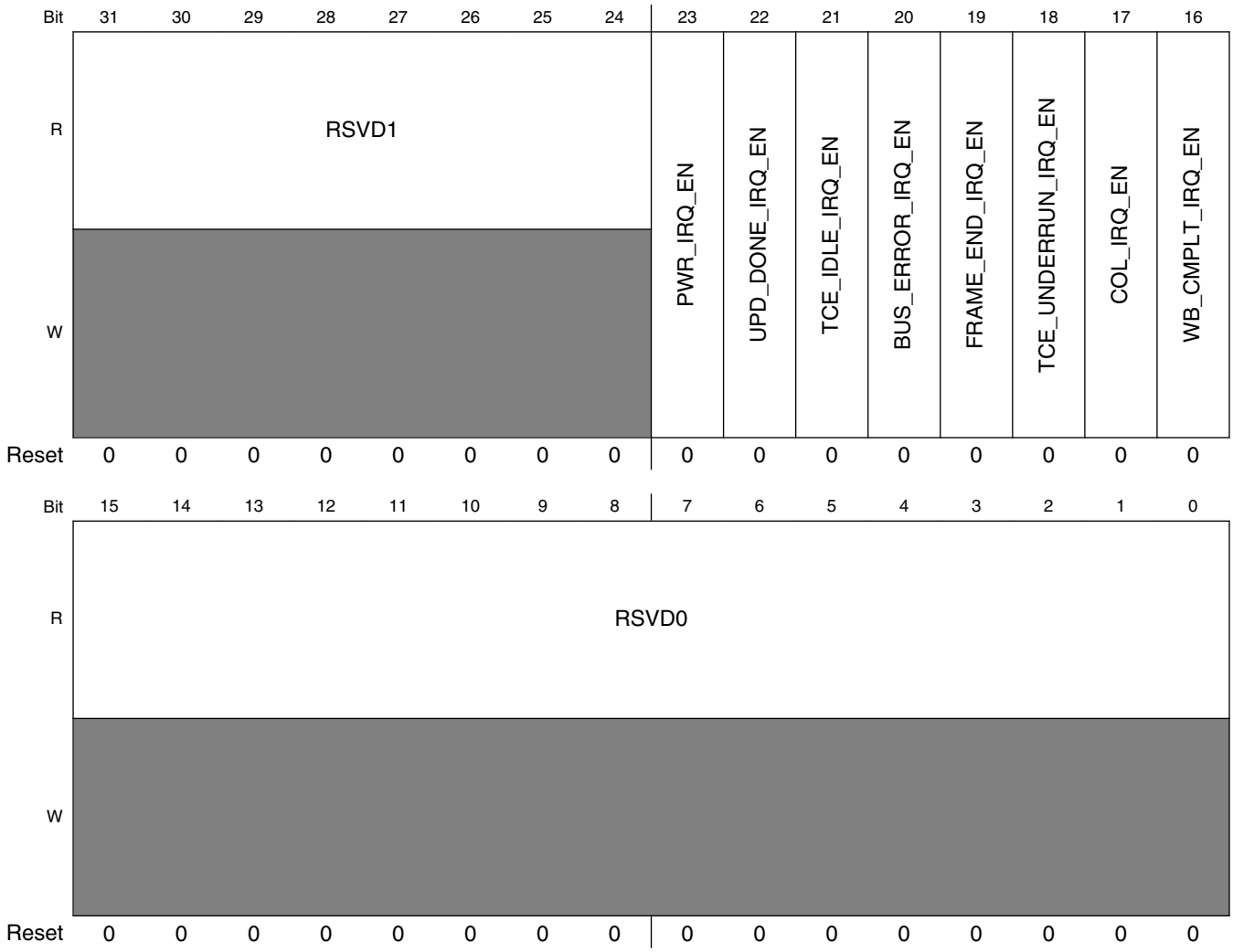
Field	Description
LUTN_CMPLT_IRQ	LUT 32~64 Complete Interrupt

24.4.32 EPDC IRQ Mask Register (EPDC_IRQ_MASK*n*)

Controls masking for all EPDC interrupts

This register controls IRQ masks for all EPDC interrupts

Address: 20F_4000h base + 400h offset + (4d × i), where i=0d to 3d



EPDC_IRQ_MASK*n* field descriptions

Field	Description
31–24 RSVD1	Reserved.
23 PWR_IRQ_EN	Enable power interrupt
22 UPD_DONE_ IRQ_EN	Enable UPD complete interrupt

Table continues on the next page...

EPDC_IRQ_MASK*n* field descriptions (continued)

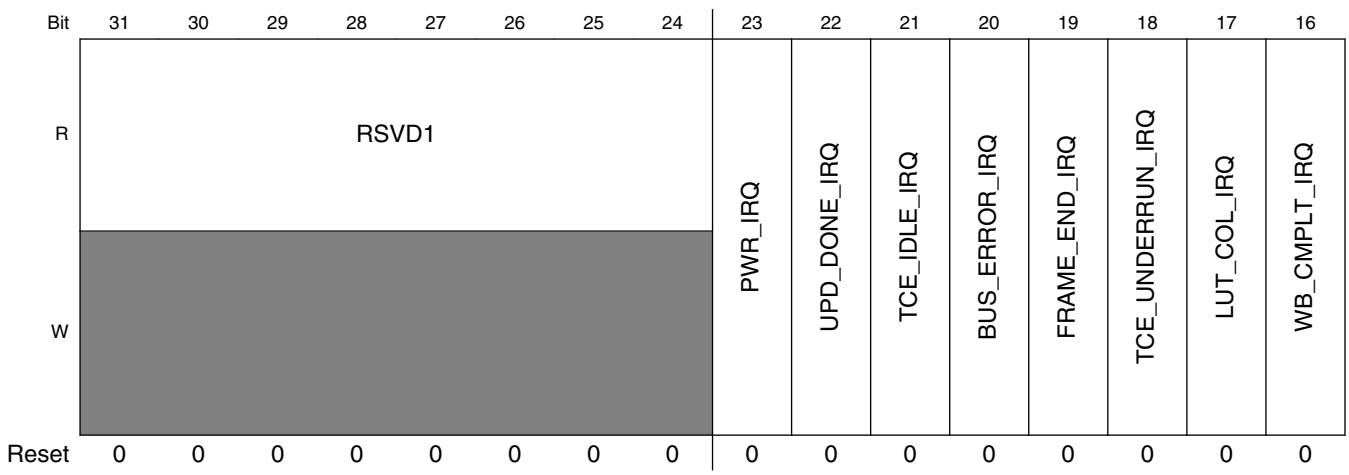
Field	Description
21 TCE_IDLE_IRQ_EN	Enable TCE Idle interrupt detection.
20 BUS_ERROR_IRQ_EN	Enable AXI BUS ERROR interrupt detection.
19 FRAME_END_IRQ_EN	If this bit is set, EPDC will assert the current frame end interrupt. This irq is only available during updating period.
18 TCE_UNDERRUN_IRQ_EN	Enable pixel FIFO under-run condition detection.
17 COL_IRQ_EN	Enable collision detection interrupts for all LUTs
16 WB_CMPLT_IRQ_EN	Enable WB complete interrupt
RSVD0	Reserved.

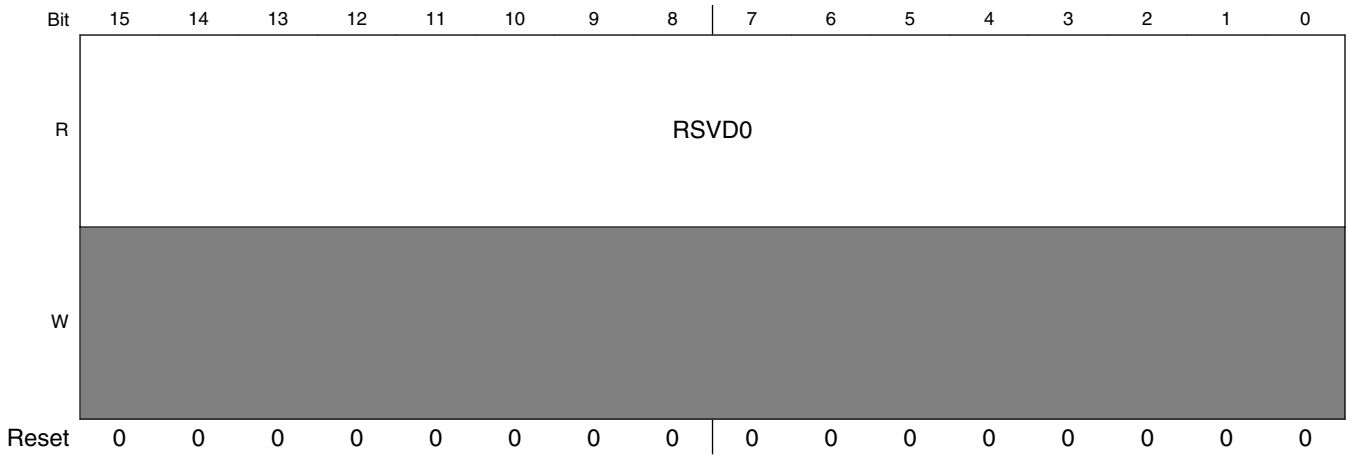
24.4.33 EPDC Interrupt Register (EPDC_IRQ*n*)

EPDC LUT Completion IRQs. The IRQ for a specific LUT is triggered when it's corrrsponding physical update is competed on the screen. Each interrupt has a corresponding mask register in HW_EPDC_IRQ_MASK

This register houses the interrupt bits for the LUT Completions

Address: 20F_4000h base + 420h offset + (4d × i), where i=0d to 3d





EPDC_IRQ*n* field descriptions

Field	Description
31–24 RSVD1	Reserved.
23 PWR_IRQ	Power Interrupt
22 UPD_DONE_IRQ	Working buffer process complete Interrupt
21 TCE_IDLE_IRQ	Interrupt to indicate that the TCE has completed TFT frame scans and is in an idle state.
20 BUS_ERROR_IRQ	Interrupt to indicate AXI BUS error occurs.
19 FRAME_END_IRQ	Interrupt to indicate EPDC has completed the current frame and is in the vertical blanking period.
18 TCE_UNDERRUN_IRQ	Interrupt to indicate that a pixel FIFO under-run has occured.
17 LUT_COL_IRQ	Collision detection interrupt. Check HW_EPDC_STATUS_COL.
16 WB_CMPLT_IRQ	Working buffer process complete Interrupt
RSVD0	Reserved.

24.4.34 EPDC Status Register - LUTs (EPDC_STATUS_LUTS1*n*)

EPDC Status Register - LUTS 0~31

Address: 20F_4000h base + 440h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LUTN_STS																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_STATUS_LUTS1*n* field descriptions

Field	Description
LUTN_STS	LUT 0~31 Status : 1=ACTIVE, 0=IDLE

24.4.35 EPDC Status Register - LUTs (EPDC_STATUS_LUTS2*n*)

EPDC Status Register - LUTS 0~31

Address: 20F_4000h base + 450h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LUTN_STS																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_STATUS_LUTS2*n* field descriptions

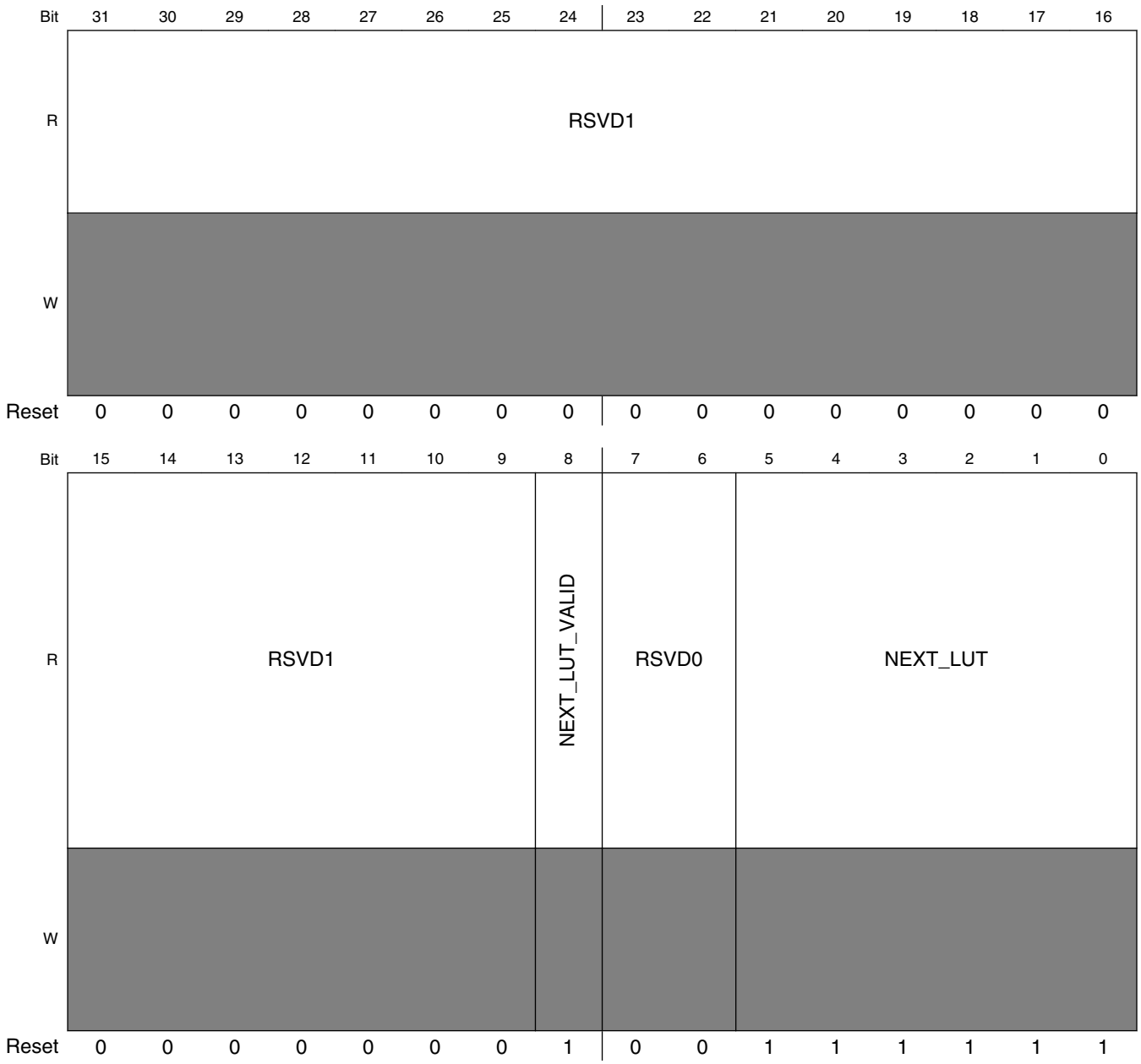
Field	Description
LUTN_STS	LUT 32~63 Status : 1=ACTIVE, 0=IDLE

24.4.36 EPDC Status Register - Next Available LUT (EPDC_STATUS_NEXTLUT)

Holds value of next available LUT. Can be used for fast LUT assignment. This value can be read and then used in an update command as part of the HW_EPDC_UPD_CTRL register write

The DIGCTL Status Register provides a read-only view to various input conditions and internal states.

Address: 20F_4000h base + 460h offset = 20F_4460h



EPDC_STATUS_NEXTLUT field descriptions

Field	Description
31–9 RSVD1	Reserved.
8 NEXT_LUT_ VALID	This bitfield can be used to check against a LUTs full condition

Table continues on the next page...

EPDC_STATUS_NEXTLUT field descriptions (continued)

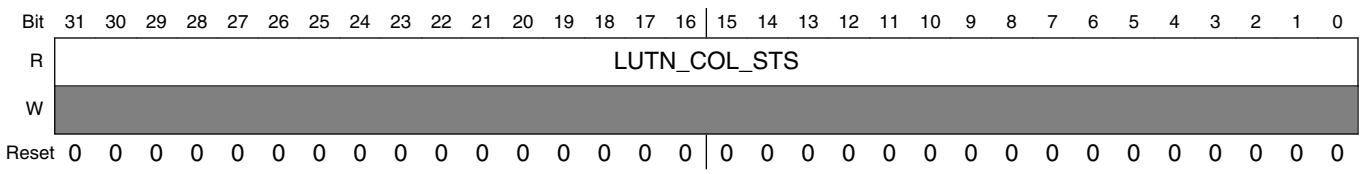
Field	Description
7–6 RSVD0	Reserved.
NEXT_LUT	Next available LUT value

24.4.37 EPDC LUT Collision Status (EPDC_STATUS_COL1*n*)

EPDC LUT Collision Status Register and works in conjuction with HW_EPDC_IRQ[LUT_COL_IRQ]. When a collision occurs the interrupt is set and all status bits are set for LUTs which were touched by the collision. It does not set the bit for the LUT which caused the collision. There is a single interrupt mask which is used to control all the IRQ bits in this register (in HW_EPDC_IRQ_MASK). Note that a collision caused by a LUT which was set-up for no collision detection will not trigger any collision LUT IRQ or status update. Note that clearing the interrupt bit HW_EPDC_IRQ[LUT_COL_IRQ] clears this register

This register houses the status bits for the LUT Collisions

Address: 20F_4000h base + 480h offset + (4d × i), where i=0d to 3d



EPDC_STATUS_COL1*n* field descriptions

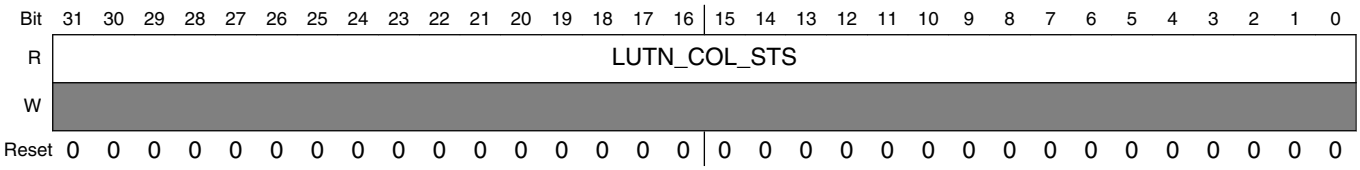
Field	Description
LUTN_COL_STS	LUTn Collision Status

24.4.38 EPDC LUT Collision Status (EPDC_STATUS_COL2*n*)

EPDC LUT Collision Status Register and works in conjuction with HW_EPDC_IRQ[LUT_COL_IRQ]. When a collision occurs the interrupt is set and all status bits are set for LUTs which were touched by the collision. It does not set the bit for the LUT which caused the collision. There is a single interrupt mask which is used to control all the IRQ bits in this register (in HW_EPDC_IRQ_MASK). Note that a collision caused by a LUT which was set-up for no collision detection will not trigger any collision LUT IRQ or status update. Note that clearing the interrupt bit HW_EPDC_IRQ[LUT_COL_IRQ] clears this register

This register houses the status bits for the LUT Collisions

Address: 20F_4000h base + 490h offset + (4d × i), where i=0d to 3d



EPDC_STATUS_COL2*n* field descriptions

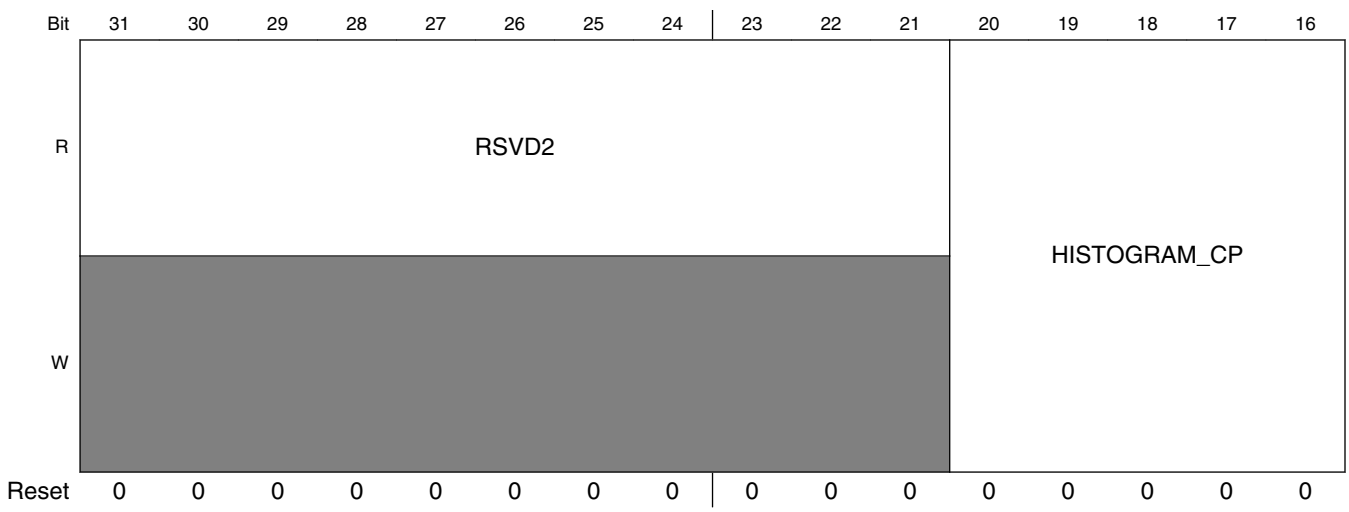
Field	Description
LUTN_COL_STS	LUTn Collision Status

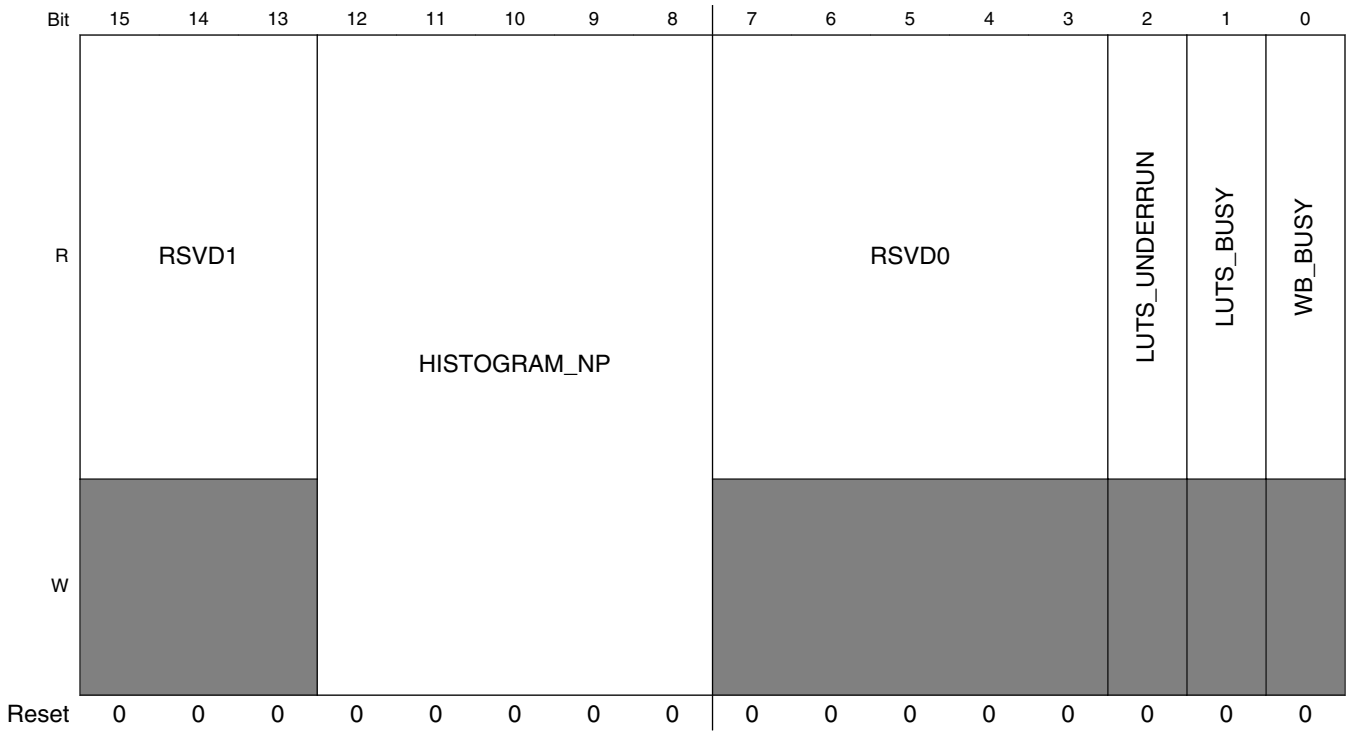
24.4.39 EPDC General Status Register (EPDC_STATUS*n*)

Register to house non LUT specific status bits

This register houses general status bits

Address: 20F_4000h base + 4A0h offset + (4d × i), where i=0d to 3d





EPDC_STATUS*n* field descriptions

Field	Description
31–21 RSVD2	Reserved.
20–16 HISTOGRAM_CP	Indicates which histogram matched the existing bitmap(CP). Bit[0] indicates that the bitmap pixels were fully contained within the HIST1 (single color) histogram. Bit[1] indicates that the bitmap pixels were fully contained within the HIST2 (black / white) histogram. Bit[2] indicates that the bitmap pixels were fully contained within the HIST4 (2-bit grayscale) histogram. Bit[3] indicates that the bitmap pixels were fully contained within the HIST8 (3-bit grayscale) histogram. Bit[4] indicates that the bitmap pixels were fully contained within the HIST16 (4-bit grayscale) histogram.
15–13 RSVD1	Reserved.
12–8 HISTOGRAM_NP	Indicates which histogram matched the processed bitmap(NP). Bit[0] indicates that the bitmap pixels were fully contained within the HIST1 (single color) histogram. Bit[1] indicates that the bitmap pixels were fully contained within the HIST2 (black / white) histogram. Bit[2] indicates that the bitmap pixels were fully contained within the HIST4 (2-bit grayscale) histogram. Bit[3] indicates that the bitmap pixels were fully contained within the HIST8 (3-bit grayscale) histogram. Bit[4] indicates that the bitmap pixels were fully contained within the HIST16 (4-bit grayscale) histogram.
7–3 RSVD0	Reserved.
2 LUTS_UNDERRUN	Provides a summary status of LUT fill. 1= not enough time for active luts read during blanking period before vscan_holdoff. 0=complete all active luts fill during blanking period before VSCAN_HOLDOFF.

Table continues on the next page...

EPDC_STATUS*n* field descriptions (continued)

Field	Description
1 LUTS_BUSY	Provides a summary status of LUTs. 1= All LUTs are busy, 0= LUTs are available
0 WB_BUSY	Working buffer process is busy cannot accept new update requests. When WB_BUSY is 1, software should wait for the WB_CMPLT_IRQ interrupt. When this interrupt occurs WB_BUSY is cleared immediately. This is a real-time status of the process.

24.4.40 EPDC Collision Region Co-ordinate (EPDC_UPD_COL_CORD)

EPDC Collision Region Co-ordinate, cleared when new update issued

This register only valid after WB completion and collision happens.

Address: 20F_4000h base + 4C0h offset = 20F_44C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD1			YCORD													RSVD0			XCORD												
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_UPD_COL_CORD field descriptions

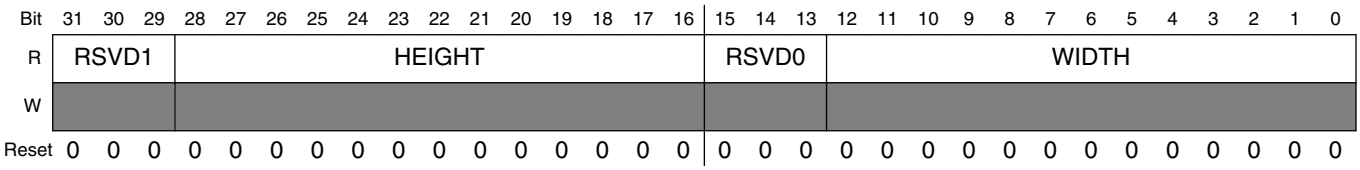
Field	Description
31–29 RSVD1	Reserved.
28–16 YCORD	Y co-ordinate for collision region of the latest completed update
15–13 RSVD0	Reserved.
XCORD	X co-ordinate for collision region of the latest completed update

24.4.41 EPDC Collision Region Size (EPDC_UPD_COL_SIZE)

EPDC Collision Region Size of the latest completed update cleared when new update issued

This register only valid after WB completion and collision happens.

Address: 20F_4000h base + 4E0h offset = 20F_44E0h



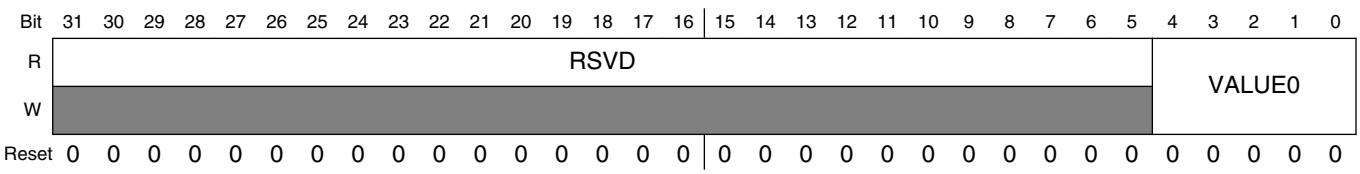
EPDC_UPD_COL_SIZE field descriptions

Field	Description
31–29 RSVD1	Reserved.
28–16 HEIGHT	Height (in pixels)
15–13 RSVD0	Reserved.
WIDTH	Width (in pixels)

24.4.42 1-level Histogram Parameter Register. (EPDC_HIST1_PARAM)

This register specifies the valid values for a 1-level(single color) histogram. If all pixels in a bitmap is only one color, STATUS[0] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the TFT_PIXEL_FORMAT control field.

Address: 20F_4000h base + 600h offset = 20F_4600h



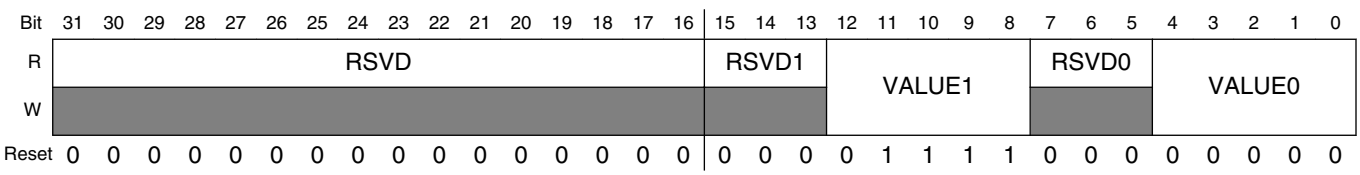
EPDC_HIST1_PARAM field descriptions

Field	Description
31–5 RSVD	Reserved, always set to zero.
VALUE0	value for 1-level histogram

24.4.43 2-level Histogram Parameter Register. (EPDC_HIST2_PARAM)

This register specifies the valid values for a 2-level histogram. If all pixels in a bitmap match these two values, STATUS[0] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the TFT_PIXEL_FORMAT control field.

Address: 20F_4000h base + 610h offset = 20F_4610h



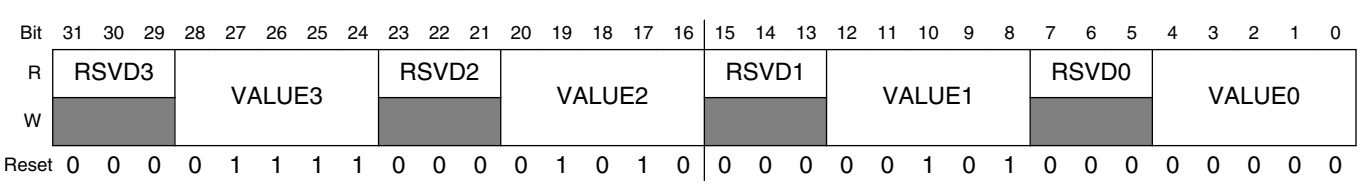
EPDC_HIST2_PARAM field descriptions

Field	Description
31–16 RSVD	Reserved, always set to zero.
15–13 RSVD1	Reserved, always set to zero.
12–8 VALUE1	White value for 2-level histogram
7–5 RSVD0	Reserved, always set to zero.
VALUE0	Black value for 2-level histogram

24.4.44 4-level Histogram Parameter Register. (EPDC_HIST4_PARAM)

This register specifies the valid values for a 4-level histogram. If all pixels in a bitmap match these two values, STATUS[1] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the TFT_PIXEL_FORMAT control field.

Address: 20F_4000h base + 620h offset = 20F_4620h



EPDC_HIST4_PARAM field descriptions

Field	Description
31–29 RSVD3	Reserved, always set to zero.
28–24 VALUE3	GRAY3 (White) value for 4-level histogram
23–21 RSVD2	Reserved, always set to zero.
20–16 VALUE2	GRAY2 value for 4-level histogram
15–13 RSVD1	Reserved, always set to zero.
12–8 VALUE1	GRAY1 value for 4-level histogram
7–5 RSVD0	Reserved, always set to zero.
VALUE0	GRAY0 (Black) value for 4-level histogram

24.4.45 8-level Histogram Parameter 0 Register. (EPDC_HIST8_PARAM0)

This register specifies four of the valid values for an 8-level histogram. If all pixels in a bitmap match these two values, STATUS[2] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the TFT_PIXEL_FORMAT control field.

Address: 20F_4000h base + 630h offset = 20F_4630h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD3			VALUE3					RSVD2			VALUE2					RSVD1			VALUE1					RSVD0			VALUE0				
W																																
Reset	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_HIST8_PARAM0 field descriptions

Field	Description
31–29 RSVD3	Reserved, always set to zero.
28–24 VALUE3	GRAY3 value for 8-level histogram
23–21 RSVD2	Reserved, always set to zero.
20–16 VALUE2	GRAY2 value for 8-level histogram

Table continues on the next page...

EPDC_HIST8_PARAM0 field descriptions (continued)

Field	Description
15–13 RSVD1	Reserved, always set to zero.
12–8 VALUE1	GRAY1 value for 8-level histogram
7–5 RSVD0	Reserved, always set to zero.
VALUE0	GRAY0 (Black) value for 8-level histogram

24.4.46 8-level Histogram Parameter 1 Register. (EPDC_HIST8_PARAM1)

This register specifies four of the valid values for an 8-level histogram. If all pixels in a bitmap match these two values, STATUS[2] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the TFT_PIXEL_FORMAT control field.

Address: 20F_4000h base + 640h offset = 20F_4640h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD7			VALUE7					RSVD6			VALUE6					RSVD5			VALUE5					RSVD4			VALUE4				
W																																
Reset	0	0	0	0	1	1	1	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0	1	1	0	0	0	0	1	0	0	1

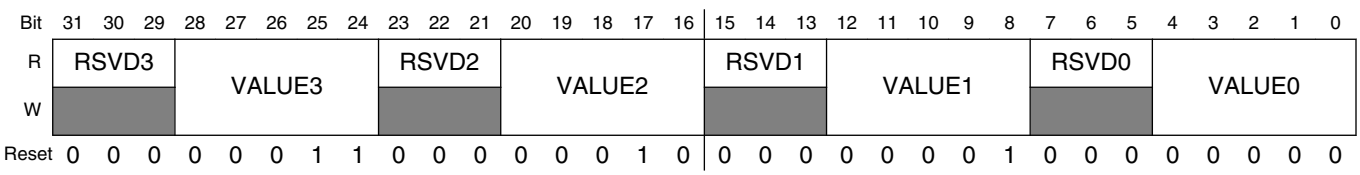
EPDC_HIST8_PARAM1 field descriptions

Field	Description
31–29 RSVD7	Reserved, always set to zero.
28–24 VALUE7	GRAY7 (White) value for 8-level histogram
23–21 RSVD6	Reserved, always set to zero.
20–16 VALUE6	GRAY6 value for 8-level histogram
15–13 RSVD5	Reserved, always set to zero.
12–8 VALUE5	GRAY5 value for 8-level histogram
7–5 RSVD4	Reserved, always set to zero.
VALUE4	GRAY4 value for 8-level histogram

24.4.47 16-level Histogram Parameter 0 Register. (EPDC_HIST16_PARAM0)

This register specifies four of the valid values for a 16-level histogram. If all pixels in a bitmap match these two values, STATUS[3] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the TFT_PIXEL_FORMAT control field.

Address: 20F_4000h base + 650h offset = 20F_4650h



EPDC_HIST16_PARAM0 field descriptions

Field	Description
31–29 RSVD3	Reserved, always set to zero.
28–24 VALUE3	GRAY3 value for 16-level histogram
23–21 RSVD2	Reserved, always set to zero.
20–16 VALUE2	GRAY2 value for 16-level histogram
15–13 RSVD1	Reserved, always set to zero.
12–8 VALUE1	GRAY1 value for 16-level histogram
7–5 RSVD0	Reserved, always set to zero.
VALUE0	GRAY0 (Black) value for 16-level histogram

24.4.48 16-level Histogram Parameter Register. (EPDC_HIST16_PARAM1)

This register specifies four of the valid values for a 16-level histogram. If all pixels in a bitmap match these two values, STATUS[3] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the TFT_PIXEL_FORMAT control field.

Address: 20F_4000h base + 660h offset = 20F_4660h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD7			VALUE7					RSVD6			VALUE6					RSVD5			VALUE5					RSVD4			VALUE4				
W																																
Reset	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0

EPDC_HIST16_PARAM1 field descriptions

Field	Description
31–29 RSVD7	Reserved, always set to zero.
28–24 VALUE7	GRAY7 value for 16-level histogram
23–21 RSVD6	Reserved, always set to zero.
20–16 VALUE6	GRAY6 value for 16-level histogram
15–13 RSVD5	Reserved, always set to zero.
12–8 VALUE5	GRAY5 value for 16-level histogram
7–5 RSVD4	Reserved, always set to zero.
VALUE4	GRAY4 value for 16-level histogram

24.4.49 16-level Histogram Parameter Register. (EPDC_HIST16_PARAM2)

This register specifies four of the valid values for a 16-level histogram. If all pixels in a bitmap match these two values, STATUS[3] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the TFT_PIXEL_FORMAT control field.

Address: 20F_4000h base + 670h offset = 20F_4670h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD11			VALUE11					RSVD10			VALUE10					RSVD9			VALUE9					RSVD8			VALUE8				
W																																
Reset	0	0	0	0	1	0	1	1	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0

EPDC_HIST16_PARAM2 field descriptions

Field	Description
31–29 RSVD11	Reserved, always set to zero.

Table continues on the next page...

EPDC_HIST16_PARAM2 field descriptions (continued)

Field	Description
28–24 VALUE11	GRAY11 value for 16-level histogram
23–21 RSVD10	Reserved, always set to zero.
20–16 VALUE10	GRAY10 value for 16-level histogram
15–13 RSVD9	Reserved, always set to zero.
12–8 VALUE9	GRAY9 value for 16-level histogram
7–5 RSVD8	Reserved, always set to zero.
VALUE8	GRAY8 value for 16-level histogram

24.4.50 16-level Histogram Parameter Register. (EPDC_HIST16_PARAM3)

This register specifies four of the valid values for a 16-level histogram. If all pixels in a bitmap match these two values, STATUS[3] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the TFT_PIXEL_FORMAT control field.

Address: 20F_4000h base + 680h offset = 20F_4680h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD15			VALUE15					RSVD14			VALUE14					RSVD13			VALUE13					RSVD12			VALUE12				
W																																
Reset	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0

EPDC_HIST16_PARAM3 field descriptions

Field	Description
31–29 RSVD15	Reserved, always set to zero.
28–24 VALUE15	GRAY15 (White) value for 16-level histogram
23–21 RSVD14	Reserved, always set to zero.
20–16 VALUE14	GRAY14 value for 16-level histogram
15–13 RSVD13	Reserved, always set to zero.

Table continues on the next page...

EPDC_HIST16_PARAM3 field descriptions (continued)

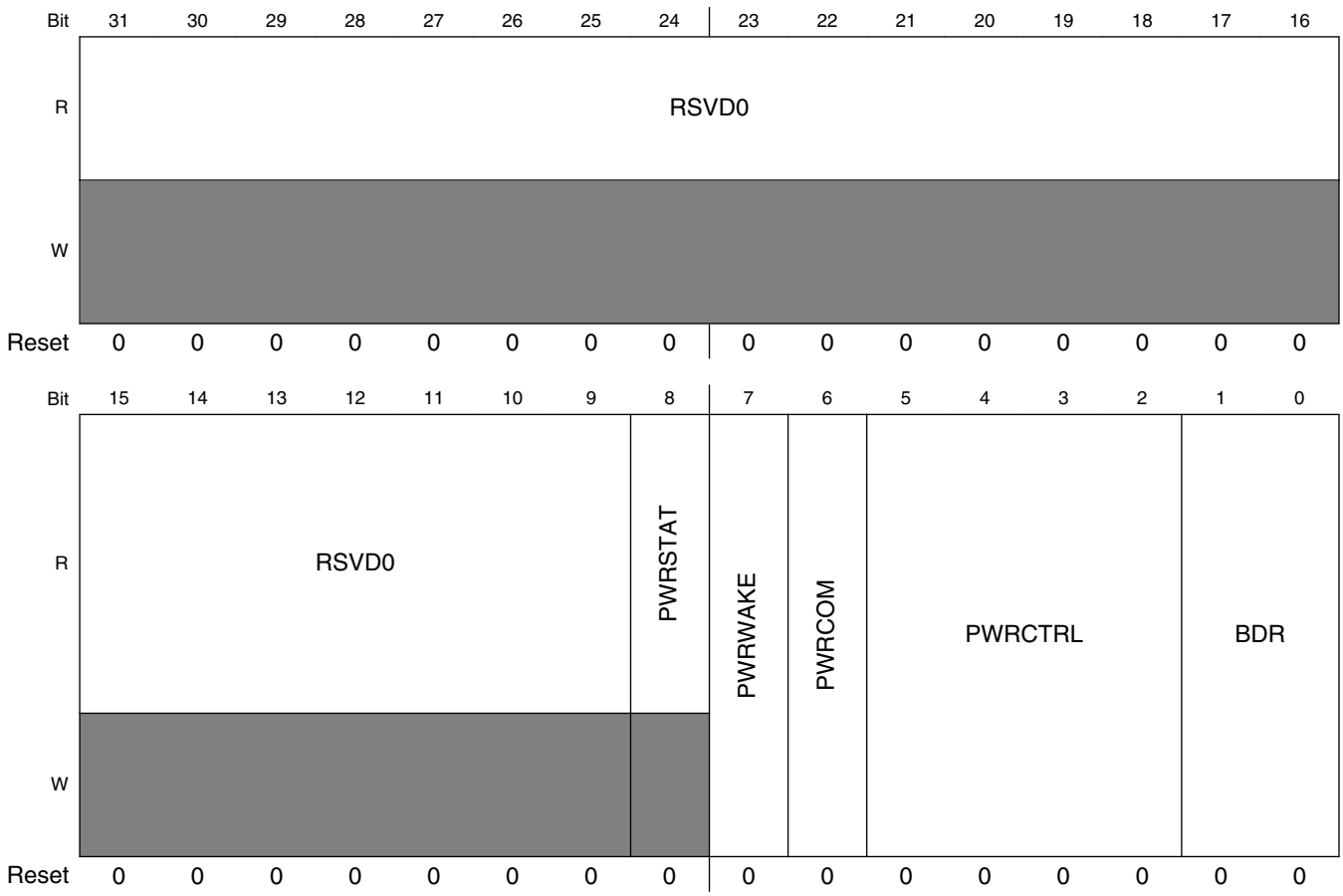
Field	Description
12–8 VALUE13	GRAY13 value for 16-level histogram
7–5 RSVD12	Reserved, always set to zero.
VALUE12	GRAY12 value for 16-level histogram

24.4.51 EPDC General Purpose I/O Debug register (EPDC_GPIO*n*)

GPIO register to control BDR[1:0], EPDC_PWR_CTRL[3:0] and EPDC_PWR_COM output signals

Houses software control signal reisters

Address: 20F_4000h base + 700h offset + (4d × i), where i=0d to 3d



EPDC_GPIO*n* field descriptions

Field	Description
31–9 RSVD0	Reserved.
8 PWRSTAT	reflect ipp_epdc_pwrstat input
7 PWRWAKE	Controls ipp_epdc_pwrwake output
6 PWRCOM	Controls ipp_epdc_pwrcom output
5–2 PWRCTRL	Controls ipp_epdc_pwrctrl[3:0] output
BDR	Controls ipp_epdc_bdr[1:0] output

24.4.52 EPDC Version Register (EPDC_VERSION)

This register reflects the version number for the EPDC.

EXAMPLE

```
No Example.
```

Address: 20F_4000h base + 7F0h offset = 20F_47F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR								MINOR								STEP															
W																																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_VERSION field descriptions

Field	Description
31–24 MAJOR	Fixed read-only value reflecting the MAJOR field of the RTL version.
23–16 MINOR	Fixed read-only value reflecting the MINOR field of the RTL version.
STEP	Fixed read-only value reflecting the stepping of the RTL version.

24.4.53 Panel Interface Signal Generator Register 0_0 (EPDC_PIGEON_0_0)

parameters for timing signal generation

Address: 20F_4000h base + 800h offset = 20F_4800h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_0_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN 0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state 0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse 0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.54 Panel Interface Signal Generator Register 0_1 (EPDC_PIGEON_0_1)

parameters for timing signal generation

Address: 20F_4000h base + 810h offset = 20F_4810h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_0_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value 0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.55 Panel Interface Signal Generator Register 0_1 (EPDC_PIGEON_0_2)

parameters for timing signal generation

Address: 20F_4000h base + 820h offset = 20F_4820h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																							SIG_ANOTHER					SIG_LOGIC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_0_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation

Table continues on the next page...

EPDC_PIGEON_0_2 field descriptions (continued)

Field	Description
	other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.56 Panel Interface Signal Generator Register 1_0 (EPDC_PIGEON_1_0)

parameters for timing signal generation

Address: 20F_4000h base + 840h offset = 20F_4840h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_1_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN 0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state

Table continues on the next page...

EPDC_PIGEON_1_0 field descriptions (continued)

Field	Description
	0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse 0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.57 Panel Interface Signal Generator Register 1_1 (EPDC_PIGEON_1_1)

parameters for timing signal generation

Address: 20F_4000h base + 850h offset = 20F_4850h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_1_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value 0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.58 Panel Interface Signal Generator Register 1_1 (EPDC_PIGEON_1_2)

parameters for timing signal generation

Address: 20F_4000h base + 860h offset = 20F_4860h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																								SIG_ANOTHER				SIG_LOGIC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_1_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.59 Panel Interface Signal Generator Register 2_0 (EPDC_PIGEON_2_0)

parameters for timing signal generation

Address: 20F_4000h base + 880h offset = 20F_4880h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_2_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN 0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state 0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse 0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.60 Panel Interface Signal Generator Register 2_1 (EPDC_PIGEON_2_1)

parameters for timing signal generation

Address: 20F_4000h base + 890h offset = 20F_4890h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_2_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value 0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.61 Panel Interface Signal Generator Register 2_1 (EPDC_PIGEON_2_2)

parameters for timing signal generation

Address: 20F_4000h base + 8A0h offset = 20F_48A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																							SIG_ANOTHER					SIG_LOGIC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_2_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.62 Panel Interface Signal Generator Register 3_0 (EPDC_PIGEON_3_0)

parameters for timing signal generation

Address: 20F_4000h base + 8C0h offset = 20F_48C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_3_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN 0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state 0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse

Table continues on the next page...

EPDC_PIGEON_3_0 field descriptions (continued)

Field	Description
	0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.63 Panel Interface Signal Generator Register 3_1 (EPDC_PIGEON_3_1)

parameters for timing signal generation

Address: 20F_4000h base + 8D0h offset = 20F_48D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_3_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value 0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.64 Panel Interface Signal Generator Register 3_1 (EPDC_PIGEON_3_2)

parameters for timing signal generation

Address: 20F_4000h base + 8E0h offset = 20F_48E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																							SIG_ANOTHER					SIG_LOGIC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_3_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.65 Panel Interface Signal Generator Register 4_0 (EPDC_PIGEON_4_0)

parameters for timing signal generation

Address: 20F_4000h base + 900h offset = 20F_4900h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_4_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN

Table continues on the next page...

EPDC_PIGEON_4_0 field descriptions (continued)

Field	Description
	0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state 0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse 0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.66 Panel Interface Signal Generator Register 4_1 (EPDC_PIGEON_4_1)

parameters for timing signal generation

Address: 20F_4000h base + 910h offset = 20F_4910h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_4_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value

Table continues on the next page...

EPDC_PIGEON_4_1 field descriptions (continued)

Field	Description
	0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.67 Panel Interface Signal Generator Register 4_1 (EPDC_PIGEON_4_2)

parameters for timing signal generation

Address: 20F_4000h base + 920h offset = 20F_4920h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																							SIG_ANOTHER					SIG_LOGIC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_4_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.68 Panel Interface Signal Generator Register 5_0 (EPDC_PIGEON_5_0)

parameters for timing signal generation

Address: 20F_4000h base + 940h offset = 20F_4940h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_5_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN 0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state 0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse 0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.69 Panel Interface Signal Generator Register 5_1 (EPDC_PIGEON_5_1)

parameters for timing signal generation

Address: 20F_4000h base + 950h offset = 20F_4950h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_5_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value 0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.70 Panel Interface Signal Generator Register 5_1 (EPDC_PIGEON_5_2)

parameters for timing signal generation

Address: 20F_4000h base + 960h offset = 20F_4960h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																							SIG_ANOTHER					SIG_LOGIC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_5_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation

Table continues on the next page...

EPDC_PIGEON_5_2 field descriptions (continued)

Field	Description
	other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.71 Panel Interface Signal Generator Register 6_0 (EPDC_PIGEON_6_0)

parameters for timing signal generation

Address: 20F_4000h base + 980h offset = 20F_4980h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_6_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN 0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state

Table continues on the next page...

EPDC_PIGEON_6_0 field descriptions (continued)

Field	Description
	0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse 0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.72 Panel Interface Signal Generator Register 6_1 (EPDC_PIGEON_6_1)

parameters for timing signal generation

Address: 20F_4000h base + 990h offset = 20F_4990h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_6_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value 0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.73 Panel Interface Signal Generator Register 6_1 (EPDC_PIGEON_6_2)

parameters for timing signal generation

Address: 20F_4000h base + 9A0h offset = 20F_49A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																							SIG_ANOTHER					SIG_LOGIC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_6_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.74 Panel Interface Signal Generator Register 7_0 (EPDC_PIGEON_7_0)

parameters for timing signal generation

Address: 20F_4000h base + 9C0h offset = 20F_49C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_7_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN 0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state 0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse 0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.75 Panel Interface Signal Generator Register 7_1 (EPDC_PIGEON_7_1)

parameters for timing signal generation

Address: 20F_4000h base + 9D0h offset = 20F_49D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_7_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value 0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.76 Panel Interface Signal Generator Register 7_1 (EPDC_PIGEON_7_2)

parameters for timing signal generation

Address: 20F_4000h base + 9E0h offset = 20F_49E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																							SIG_ANOTHER					SIG_LOGIC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_7_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.77 Panel Interface Signal Generator Register 8_0 (EPDC_PIGEON_8_0)

parameters for timing signal generation

Address: 20F_4000h base + A00h offset = 20F_4A00h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_8_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN 0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state 0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse

Table continues on the next page...

EPDC_PIGEON_8_0 field descriptions (continued)

Field	Description
	0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.78 Panel Interface Signal Generator Register 8_1 (EPDC_PIGEON_8_1)

parameters for timing signal generation

Address: 20F_4000h base + A10h offset = 20F_4A10h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_8_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value 0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.79 Panel Interface Signal Generator Register 8_1 (EPDC_PIGEON_8_2)

parameters for timing signal generation

Address: 20F_4000h base + A20h offset = 20F_4A20h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																							SIG_ANOTHER					SIG_LOGIC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_8_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.80 Panel Interface Signal Generator Register 9_0 (EPDC_PIGEON_9_0)

parameters for timing signal generation

Address: 20F_4000h base + A40h offset = 20F_4A40h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_9_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN

Table continues on the next page...

EPDC_PIGEON_9_0 field descriptions (continued)

Field	Description
	0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state 0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse 0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.81 Panel Interface Signal Generator Register 9_1 (EPDC_PIGEON_9_1)

parameters for timing signal generation

Address: 20F_4000h base + A50h offset = 20F_4A50h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_9_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value

Table continues on the next page...

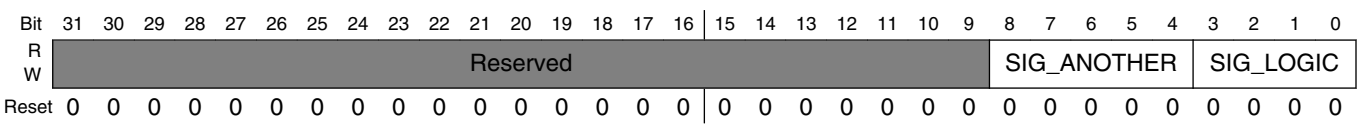
EPDC_PIGEON_9_1 field descriptions (continued)

Field	Description
	0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.82 Panel Interface Signal Generator Register 9_1 (EPDC_PIGEON_9_2)

parameters for timing signal generation

Address: 20F_4000h base + A60h offset = 20F_4A60h



EPDC_PIGEON_9_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.83 Panel Interface Signal Generator Register 10_0 (EPDC_PIGEON_10_0)

parameters for timing signal generation

Address: 20F_4000h base + A80h offset = 20F_4A80h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_10_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN 0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state 0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse 0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.84 Panel Interface Signal Generator Register 10_1 (EPDC_PIGEON_10_1)

parameters for timing signal generation

Address: 20F_4000h base + A90h offset = 20F_4A90h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_10_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value 0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.85 Panel Interface Signal Generator Register 10_1 (EPDC_PIGEON_10_2)

parameters for timing signal generation

Address: 20F_4000h base + AA0h offset = 20F_4AA0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																							SIG_ANOTHER					SIG_LOGIC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_10_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation

Table continues on the next page...

EPDC_PIGEON_10_2 field descriptions (continued)

Field	Description
	other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.86 Panel Interface Signal Generator Register 11_0 (EPDC_PIGEON_11_0)

parameters for timing signal generation

Address: 20F_4000h base + AC0h offset = 20F_4AC0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_11_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN 0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state

Table continues on the next page...

EPDC_PIGEON_11_0 field descriptions (continued)

Field	Description
	0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse 0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.87 Panel Interface Signal Generator Register 11_1 (EPDC_PIGEON_11_1)

parameters for timing signal generation

Address: 20F_4000h base + AD0h offset = 20F_4AD0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_11_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value 0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.88 Panel Interface Signal Generator Register 11_1 (EPDC_PIGEON_11_2)

parameters for timing signal generation

Address: 20F_4000h base + AE0h offset = 20F_4AE0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																								SIG_ANOTHER						SIG_LOGIC	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_11_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.89 Panel Interface Signal Generator Register 12_0 (EPDC_PIGEON_12_0)

parameters for timing signal generation

Address: 20F_4000h base + B00h offset = 20F_4B00h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_12_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN 0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state 0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse 0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.90 Panel Interface Signal Generator Register 12_1 (EPDC_PIGEON_12_1)

parameters for timing signal generation

Address: 20F_4000h base + B10h offset = 20F_4B10h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_12_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value 0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.91 Panel Interface Signal Generator Register 12_1 (EPDC_PIGEON_12_2)

parameters for timing signal generation

Address: 20F_4000h base + B20h offset = 20F_4B20h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																							SIG_ANOTHER					SIG_LOGIC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_12_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.92 Panel Interface Signal Generator Register 13_0 (EPDC_PIGEON_13_0)

parameters for timing signal generation

Address: 20F_4000h base + B40h offset = 20F_4B40h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_13_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN 0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state 0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse

Table continues on the next page...

EPDC_PIGEON_13_0 field descriptions (continued)

Field	Description
	0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.93 Panel Interface Signal Generator Register 13_1 (EPDC_PIGEON_13_1)

parameters for timing signal generation

Address: 20F_4000h base + B50h offset = 20F_4B50h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_13_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value 0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.94 Panel Interface Signal Generator Register 13_1 (EPDC_PIGEON_13_2)

parameters for timing signal generation

Address: 20F_4000h base + B60h offset = 20F_4B60h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																							SIG_ANOTHER					SIG_LOGIC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_13_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.95 Panel Interface Signal Generator Register 14_0 (EPDC_PIGEON_14_0)

parameters for timing signal generation

Address: 20F_4000h base + B80h offset = 20F_4B80h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_14_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN

Table continues on the next page...

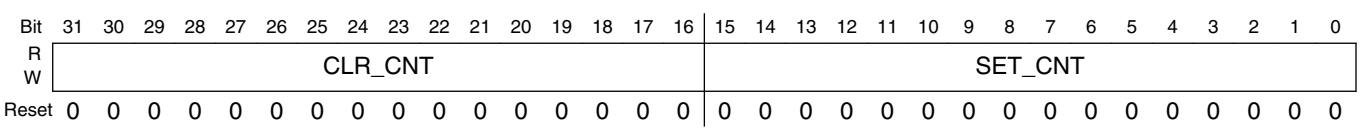
EPDC_PIGEON_14_0 field descriptions (continued)

Field	Description
	0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state 0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse 0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.96 Panel Interface Signal Generator Register 14_1 (EPDC_PIGEON_14_1)

parameters for timing signal generation

Address: 20F_4000h base + B90h offset = 20F_4B90h



EPDC_PIGEON_14_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value

Table continues on the next page...

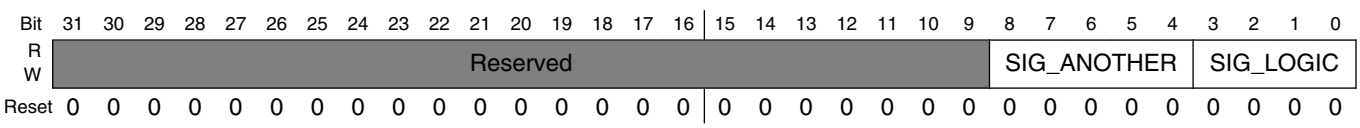
EPDC_PIGEON_14_1 field descriptions (continued)

Field	Description
	0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.97 Panel Interface Signal Generator Register 14_1 (EPDC_PIGEON_14_2)

parameters for timing signal generation

Address: 20F_4000h base + BA0h offset = 20F_4BA0h



EPDC_PIGEON_14_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.98 Panel Interface Signal Generator Register 15_0 (EPDC_PIGEON_15_0)

parameters for timing signal generation

Address: 20F_4000h base + BC0h offset = 20F_4BC0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_15_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN 0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state 0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse 0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.99 Panel Interface Signal Generator Register 15_1 (EPDC_PIGEON_15_1)

parameters for timing signal generation

Address: 20F_4000h base + BD0h offset = 20F_4BD0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_15_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value 0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.100 Panel Interface Signal Generator Register 15_1 (EPDC_PIGEON_15_2)

parameters for timing signal generation

Address: 20F_4000h base + BE0h offset = 20F_4BE0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																							SIG_ANOTHER					SIG_LOGIC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_15_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation

Table continues on the next page...

EPDC_PIGEON_15_2 field descriptions (continued)

Field	Description
	other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

24.4.101 Panel Interface Signal Generator Register 16_0 (EPDC_PIGEON_16_0)

parameters for timing signal generation

Address: 20F_4000h base + C00h offset = 20F_4C00h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

EPDC_PIGEON_16_0 field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x80 **FS** — FRAME SYNC 0x40 **FB** — FRAME BEGIN 0x20 **FD** — FRAME DATA 0x10 **FE** — FRAME END 0x8 **LS** — LINE SYNC 0x4 **LB** — LINE BEGIN 0x2 **LD** — LINE DATA 0x1 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state

Table continues on the next page...

EPDC_PIGEON_16_0 field descriptions (continued)

Field	Description
	0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle 0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=aligne with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — line start pulse 0x2 **FRAME** — frame start pulse 0x3 **SIG_ANOTHER** — use another signal as tick event
1 POL	polarity of signal output 0x0 **ACTIVE_HIGH** — normal signal (active high) 0x1 **ACTIVE_LOW** — inverted signal (active low)
0 EN	enable pigeon mode on this signal

24.4.102 Panel Interface Signal Generator Register 16_1 (EPDC_PIGEON_16_1)

parameters for timing signal generation

Address: 20F_4000h base + C10h offset = 20F_4C10h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_16_1 field descriptions

Field	Description
31–16 CLR_CNT	deassert signal output when counter match this value 0x0 **CLEAR_USING_MASK** — keep active until mask off
SET_CNT	assert signal output when counter match this value 0x0 **START_ACTIVE** — start as active

24.4.103 Panel Interface Signal Generator Register 16_1 (EPDC_PIGEON_16_2)

parameters for timing signal generation

Address: 20F_4000h base + C20h offset = 20F_4C20h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																							SIG_ANOTHER					SIG_LOGIC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPDC_PIGEON_16_2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8–4 SIG_ANOTHER	select another signal for logic operation or as mask or counter tick event
SIG_LOGIC	logic operation with another signal sigout : final output signal of this generator mask : final mask of this generator this_sig : intermediate signal of this generator before logic operation other_masks : intermediate mask result of this generator before logic operation sig_another : signal selected other generators 0x0 **DIS** — no logic operation 0x1 **AND** — sigout = sig_another AND this_sig 0x2 **OR** — sigout = sig_another OR this_sig 0x3 **MASK** — mask = sig_another AND other_masks

Chapter 25
Enhanced Periodic Interrupt Timer (EPIT)

25.1 Overview

EPIT is a 32-bit set-and-forget timer that is capable of providing precise interrupts at regular intervals with minimal processor intervention. EPIT begins counting after it is enabled by software.

The following figure shows the EPIT block diagram.

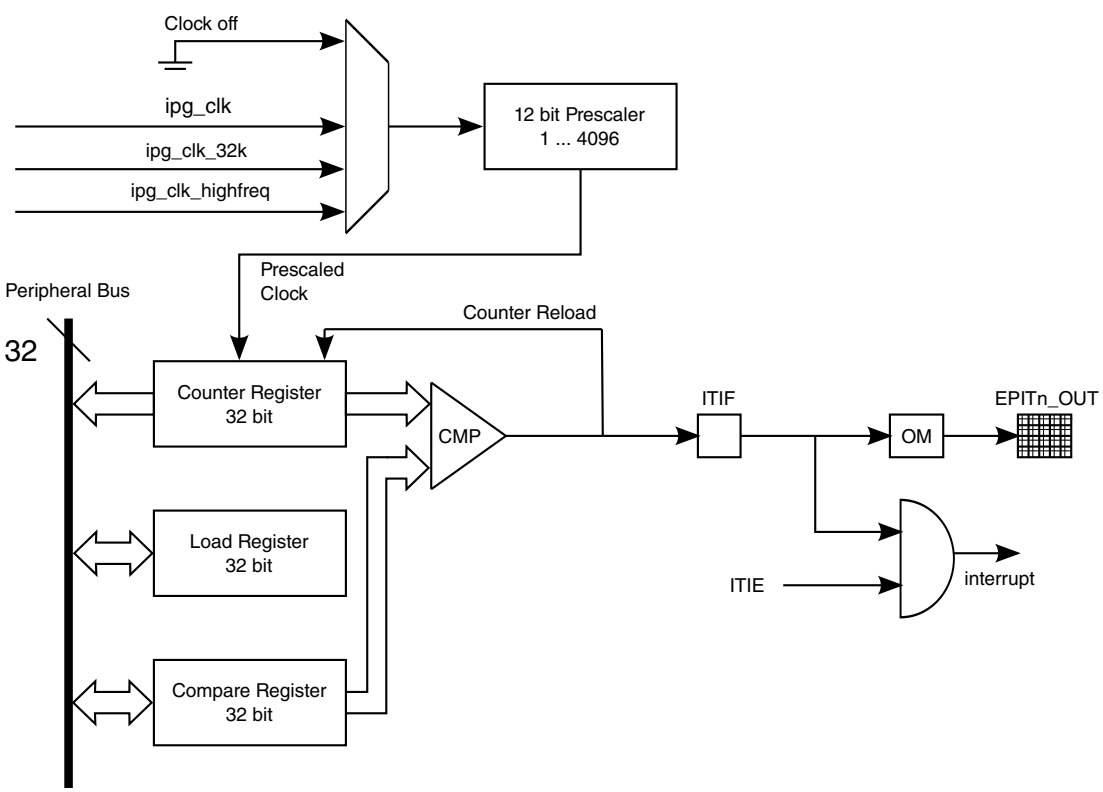


Figure 25-1. EPIT block diagram

25.1.1 EPIT features

EPIT has the following key features:

- 32-bit down counter with clock source selection
- 12-bit prescaler for division of input clock frequency
- Counter value that can be programmed on the fly
- Can be programmed to be active during low-power and debug modes
- Interrupt generation when counter reaches the compare value

25.1.2 EPIT modes and operations

EPIT supports the following modes: set-and-forget and free running. See the following sections for more information.

- Operating in set-and-forget mode
- Operating in free-running mode

See Operations for a description of the operations that EPIT supports.

25.2 External signals

The following table describes EPIT's I/O signals.

Table 25-1. EPIT External Signals

Signal	Description	Pad	Mode	Direction
EPIT1_OUT	Output 1 pin at chip boundary for indicating the occurrence of an output compare event through a specified transition.	EIM_D19	ALT6	O
		GPIO_0	ALF4	
		GPIO_7	ALT2	
EPIT2_OUT	Output 2 pin at chip boundary for indicating the occurrence of an output compare event through a specified transition.	EIM_D20	ALT6	O
		GPIO_8	ALT2	

25.3 Clocks

The table found here describes the clock sources for EPIT.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 25-2. EPIT Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_32k	ckil_sync_clk_root	Low-frequency reference clock (32 kHz)
ipg_clk_highfreq	perclk_clk_root	High-frequency reference clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

The clock that feeds the prescaler can be selected from among the following sources:

- **High-frequency reference clock (ipg_clk_highfreq)**

  This clock is provided by the Clock Control Module (CCM). This clock remains on during low-power mode when the peripheral clock is turned off, allowing EPIT to use this clock in low-power mode. In normal mode, the CCM synchronizes this clock to ahb_clk; in low-power mode, CCM switches to an unsynchronized version.

- **Low-frequency reference clock (ipg_clk_32k)**

  This 32 kHz reference clock is provided by the CCM. This clock remains on in low-power mode when the peripheral clock is turned off, so EPIT can use this clock during low-power mode. In normal mode, the CCM synchronizes this clock to ahb_clk; in low-power mode, CCM switches to an unsynchronized version. This clock is derived from the external 32 kHz crystal.

- **Peripheral clock (ipg_clk)**

  This is the peripheral clock (PER Clock) which is provided (and optionally gated) by the CCM. This clock is typically used in normal operations. In low-power modes, if the EPIT is programmed to be disabled (via STOPEN or WAITEN), then the peripheral clock can be switched off.

The clock input source is determined by the CLKSRC field in the control register. The clock input to the prescaler can also be disabled by setting CLKSRC to 0b00. **This field value should only be changed after first disabling the EPIT by clearing the EN bit in the EPIT_EPITCR**. For other programming requirements that apply while changing clock source, refer section Change of Clock Source.

The PRESCALER field in the control register is used to select the divide ratio of the input clock that drives the main counter. The prescaler can divide the input clock by a value between 1 and 4096. A change in the value of the PRESCALER field is immediately reflected on its output clock frequency. The following figure shows the timing for a change in the prescaler value.

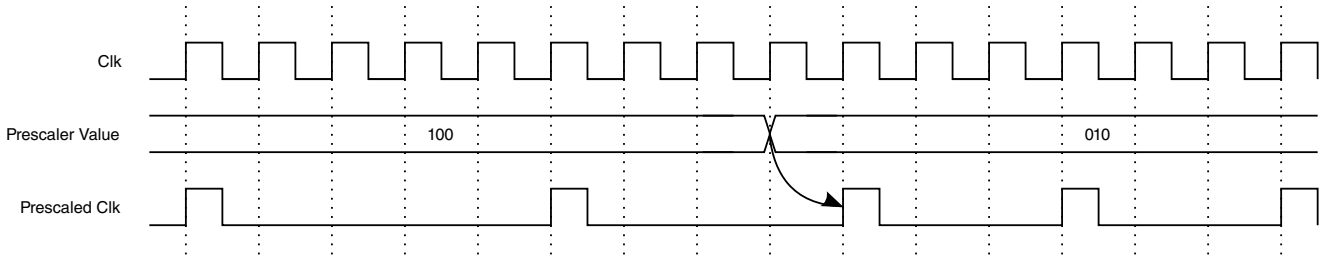


Figure 25-2. Prescaler Value Change Diagram

25.4 Functional Description

This section provides a complete functional description of the block.

25.4.1 Operating modes

EPIT can operate in either set-and-forget or free-running mode. Use EPIT_CR[RLD] to select the desired mode.

25.4.1.1 Operating in set-and-forget mode

To select this mode of operation, set the RLD bit in the control register (EPIT_CR).

In this mode, the counter obtains its data from the load register (EPIT_LR); it cannot be written to directly from the block data bus. Whenever the counter reaches zero, the value in EPIT_LR is loaded into the counter. This value is then decremented to zero.

To directly initialize the counter instead of waiting for the count to reach zero, set the EPIT counter overwrite enable bit (EPIT_CR[IOVW]) and write to EPIT_LR with the required initialization value.

25.4.1.2 Operating in free-running mode

To select this mode of operation, clear the RLD bit.

In this mode, the counter rolls over from 0000 0000h to FFFF FFFFh without reloading from the modulus register. After rolling over, the counter continues counting down.

To directly initalize the counter, set the EPIT counter overwrite enable bit (EPIT_EPITCL[IOVW]) and write to EPIT_EPITLR with the required initialization value.

25.4.2 Operations

EPIT has a single 32-bit down counter, which starts counting when the block is enabled by software.

The start value of the counter is loaded from the EPIT load register, which can be written to at any time by the processor. The value in the compare register determines the time that the interrupt occurs.

When EPIT is disabled (EN = 0), both the main counter and the prescaler counter freeze their count at their current count values. When EPIT is re-enabled (EN = 1), the ENMOD bit, which is a RW bit, decides the counter value:

- If ENMOD is set, the main counter is loaded with the load value (If RLD = 1)/ FFFF FFFFh (If RLD = 0) and the prescaler counter is reset (000h).
- If ENMOD is cleared, both main counter and prescaler counter restart counting from their frozen values.

If EPIT is programmed to be disabled in a low-power mode (STOP/WAIT), both the main counter and the prescaler counter freeze at their current count values when EPIT enters low-power mode. When EPIT exits the low-power mode, both the main counter and the prescaler counter start counting from their frozen values regardless of the ENMOD bit.

A hardware reset resets all EPIT registers to their respective reset values. There is a software reset which has the same effect on all registers except for the EN, ENMOD, STOPEN and WAITEN bits in the control register. The state of these bits are not affected by software reset. A software reset can be asserted even when the EPIT is disabled.

25.4.3 Compare Event

When the programmed value of EPIT_EPITCMPR matches the value in EPIT_EPITCNR a compare status flag is set, and an interrupt is generated if the OCIEN bit is set in the control register.

The compare output pin is set, cleared, toggled, or not affected at all depending on the setting of the output mode (OM) bits in the control register. If an interrupt is required at rollover (when the counter value reaches 0x0000_0000 and the new value is loaded) then the compare register value should be set equal to the load register value in set-and-forget mode, or equal to 0xFFFF_FFFF in free-running mode.

The following figure shows the timing for a compare event and interrupt.

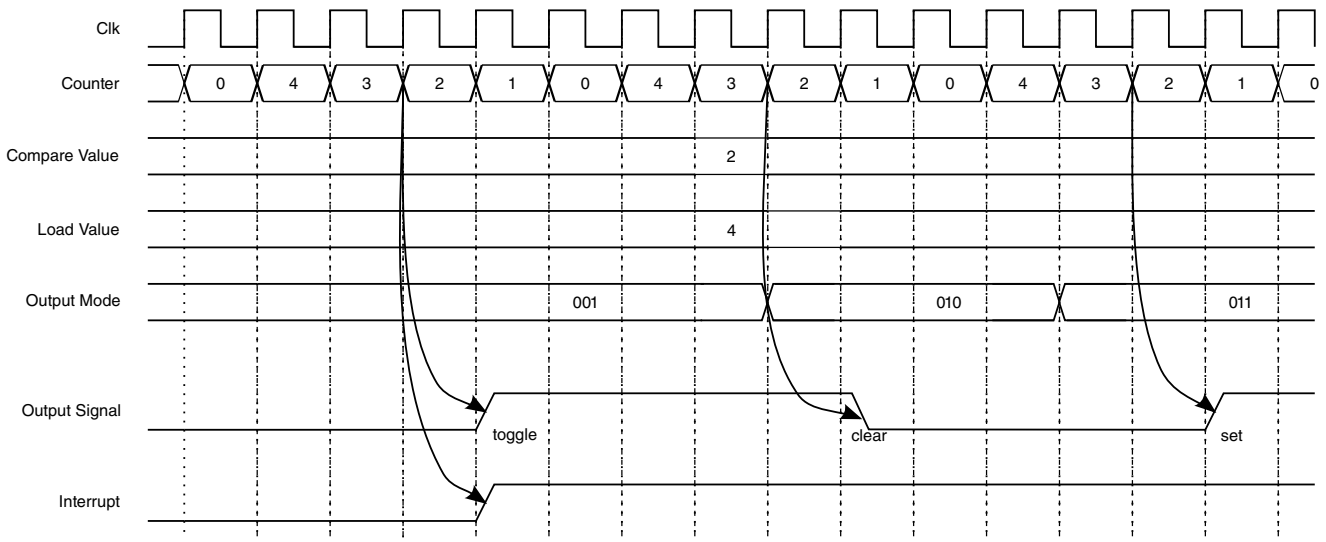


Figure 25-3. Compare Event and Interrupt Timing Diagram

EPIT will generate a compare event in the next count if the EPITx_CNR from the previous count equals the new EPITx_CMPR configured before re-enabling the EPIT in the next count. Even in case a new start counter value was updated in EPITx_LR before re-enabling the EPIT for the next round. To avoid this, configure the EPITx_CMPR to previous EPITx_CNR+1. Or, in set and forget mode, configure EPITx_LR with IOVW=1, before disabling EPIT. Also can do an extra disable/enable iteration to clear OCIF and update EPITx_CNR.

25.4.3.1 Counter Value Overwrite

The EPIT counter value can be overwritten to acquire a desired value at any point of time. The procedure for this is to set the IOVW bit in the control register and then write the desired value into the load register.

This results in the load register acquiring that value and also the counter being overwritten with it. If the EPIT is running the counter resumes counting from the overwritten value.

25.4.3.2 Low-Power Mode Behavior

The EPIT timer's behavior in low-power modes depends on which clock source is being used.

If the selected clock source is available and the corresponding low-power enable bit is set, then the EPIT continues to function in the low-power mode. If the EPIT is programmed to be disabled in a low-power mode (STOP/WAIT), then main counter and the prescaler counter freeze at the current count values when the EPIT enters low-power mode. When the EPIT exits the low-power mode, both main counter and prescaler counter start counting from their frozen values irrespective of the ENMOD bit.

25.4.3.3 Debug Mode Behavior

In debug mode, the user has the option to run or halt the EPIT timers. If the DBGEN bit is reset in the EPIT Control Register, the timer is halted.

When debug mode is exited, the timer operation reverts to what it was prior to entering debug mode.

25.5 Initialization/ Application Information

25.5.1 Change of Clock Source

The CLKSRC field in EPIT_EPITCR determines the clock source. This field value should be changed only after disabling the EPIT (EN = 0).

Below is the software sequence which must be followed while changing clock source.

1. Disable the EPIT - set EN=0 in EPIT_EPITCR.
2. Disable EPIT ouput - program OM=00 in the EPIT_EPITCR.
3. Disable EPIT interrupts.
4. Program CLKSRC to desired clock source in EPIT_EPITCR.
5. Clear the EPIT status register (EPIT_EPITSR), that is, write "1" to clear (w1c).
6. Set ENMOD= 1 in the EPIT_EPITCR, to bring the EPIT Counter to defined state (EPIT_EPITLR value or 0xFFFF_FFFF).
7. Enable EPIT - set (EN=1) in the EPIT_EPITCR
8. Enable the EPIT interrupts.

25.6 EPIT Memory Map/Register Definition

The EPIT includes five user-accessible 32-bit registers. The following table summarizes these registers and their addresses.

Peripheral bus write access to the EPIT control register (EPITCR) and the EPIT load register (EPITLR) results in one cycle of wait state, while other valid peripheral bus accesses are with 0 wait state.

EPIT memory map

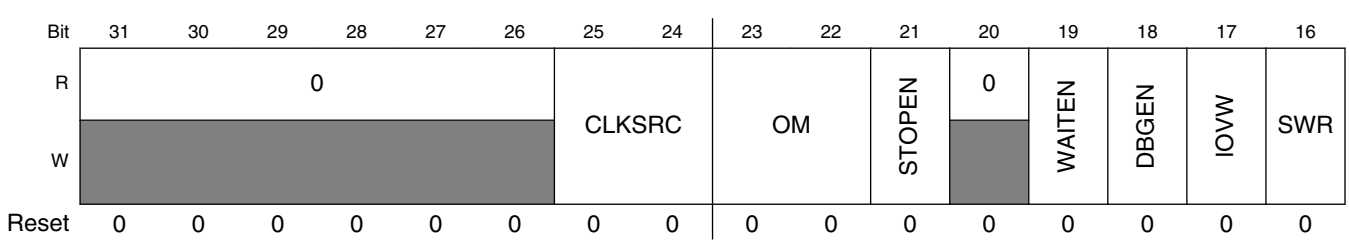
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20D_0000	Control register (EPIT1_CR)	32	R/W	0000_0000h	25.6.1/1342
20D_0004	Status register (EPIT1_SR)	32	R/W	0000_0000h	25.6.2/1345
20D_0008	Load register (EPIT1_LR)	32	R/W	FFFF_FFFFh	25.6.3/1345
20D_000C	Compare register (EPIT1_CMPR)	32	R/W	0000_0000h	25.6.4/1346
20D_0010	Counter register (EPIT1_CNR)	32	R	FFFF_FFFFh	25.6.5/1346
20D_4000	Control register (EPIT2_CR)	32	R/W	0000_0000h	25.6.1/1342
20D_4004	Status register (EPIT2_SR)	32	R/W	0000_0000h	25.6.2/1345
20D_4008	Load register (EPIT2_LR)	32	R/W	FFFF_FFFFh	25.6.3/1345
20D_400C	Compare register (EPIT2_CMPR)	32	R/W	0000_0000h	25.6.4/1346
20D_4010	Counter register (EPIT2_CNR)	32	R	FFFF_FFFFh	25.6.5/1346

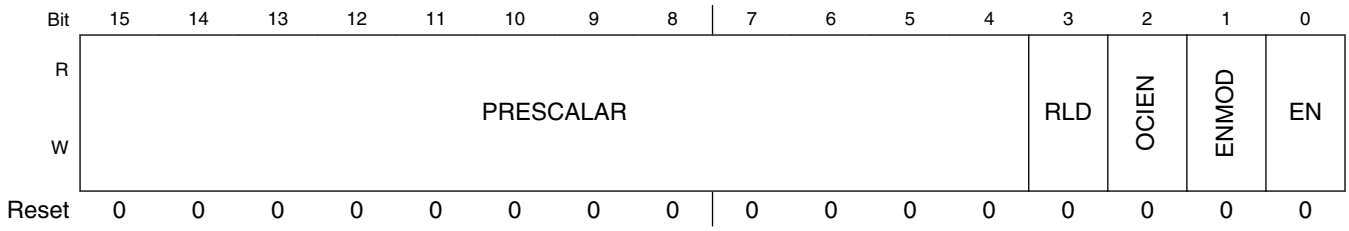
25.6.1 Control register (EPITx_CR)

The EPIT control register (EPIT_CR) is used to configure the operating settings of the EPIT. It contains the clock division prescaler value and also the interrupt enable bit. Additionally, it contains other control bits which are described below.

Peripheral Bus Write access to EPIT Control Register (EPIT_CR) results in one cycle of the wait state, while other valid peripheral bus accesses are with 0 wait state.

Address: Base address + 0h offset





EPITx_CR field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.
25–24 CLKSRC	Select clock source These bits determine which clock input is to be selected for running the counter. This field value should only be changed when the EPIT is disabled by clearing the EN bit in this register. For other programming requirements while changing clock source, refer to Change of Clock Source. 00 Clock is off 01 Peripheral clock 10 High-frequency reference clock 11 Low-frequency reference clock
23–22 OM	EPIT output mode.This bit field determines the mode of EPIT output on the output pin. 00 EPIT output is disconnected from pad 01 Toggle output pin 10 Clear output pin 11 Set output pin
21 STOPEN	EPIT stop mode enable. This read/write control bit enables the operation of the EPIT during stop mode. This bit is reset by a hardware reset and unaffected by software reset. 0 EPIT is disabled in stop mode 1 EPIT is enabled in stop mode
20 Reserved	This read-only field is reserved and always has the value 0.
19 WAITEN	This read/write control bit enables the operation of the EPIT during wait mode. This bit is reset by a hardware reset. A software reset does not affect this bit. 0 EPIT is disabled in wait mode 1 EPIT is enabled in wait mode
18 DBGEN	This bit is used to keep the EPIT functional in debug mode. When this bit is cleared, the input clock is gated off in debug mode.This bit is reset by hardware reset. A software reset does not affect this bit. 0 Inactive in debug mode 1 Active in debug mode
17 IOVW	EPIT counter overwrite enable. This bit controls the counter data when the modulus register is written. When this bit is set, all writes to the load register overwrites the counter contents and the counter starts subsequently counting down from the programmed value. 0 Write to load register does not result in counter value being overwritten. 1 Write to load register results in immediate overwriting of counter value.

Table continues on the next page...

EPITx_CR field descriptions (continued)

Field	Description
16 SWR	Software reset. The EPIT is reset when this bit is set to 1. It is a self clearing bit. This bit is set when the block is in reset state and is cleared when the reset procedure is over. Setting this bit resets all the registers to their reset values, except for the EN, ENMOD, STOPEN, WAITEN and DBGEN bits in this control register 0 EPIT is out of reset 1 EPIT is undergoing reset
15–4 PRESCALAR	Counter clock prescaler value. This bit field determines the prescaler value by which the clock is divided before it goes to the counter 0x000 Divide by 1 0x001 Divide by 2... 0xFFF Divide by 4096
3 RLD	Counter reload control. This bit is cleared by hardware reset. It decides the counter functionality, whether to run in free-running mode or set-and-forget mode. 0 When the counter reaches zero it rolls over to 0xFFFF_FFFF (free-running mode) 1 When the counter reaches zero it reloads from the modulus register (set-and-forget mode)
2 OCIEN	Output compare interrupt enable. This bit enables the generation of interrupt on occurrence of compare event. 0 Compare interrupt disabled 1 Compare interrupt enabled
1 ENMOD	EPIT enable mode. When EPIT is disabled (EN=0), both main counter and prescaler counter freeze their count at current count values. ENMOD bit is a r/w bit that determines the counter value when the EPIT is enabled again by setting EN bit. If ENMOD bit is set, then main counter is loaded with the load value (If RLD=1)/ 0xFFFF_FFFF (If RLD=0) and prescaler counter is reset, when EPIT is enabled (EN=1). If ENMOD is programmed to 0 then both main counter and prescaler counter restart counting from their frozen values when EPIT is enabled (EN=1). If EPIT is programmed to be disabled in a low-power mode (STOP/WAIT/ DEBUG), then both the main counter and the prescaler counter freeze at their current count values when EPIT enters low-power mode. When EPIT exits the low-power mode, both main counter and prescaler counter start counting from their frozen values irrespective of the ENMOD bit. This bit is reset by a hardware reset. A software reset does not affect this bit. 0 Counter starts counting from the value it had when it was disabled. 1 Counter starts count from load value (RLD=1) or 0xFFFF_FFFF (If RLD=0)
0 EN	This bit enables the EPIT. EPIT counter and prescaler value when EPIT is enabled (EN = 1), is dependent upon ENMOD and RLD bit as described for ENMOD bit. It is recommended that all registers be properly programmed before setting this bit. This bit is reset by a hardware reset. A software reset does not affect this bit. 0 EPIT is disabled 1 EPIT is enabled

25.6.2 Status register (EPITx_SR)

The EPIT status register (EPIT_SR) has a single status bit for the output compare event. The bit is a write 1 to clear bit.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															OCIF
W																w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPITx_SR field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 OCIF	Output compare interrupt flag. This bit is the interrupt flag that is set when the content of counter equals the content of the compare register (EPIT_CMPR). The bit is a write 1 to clear bit. 0 Compare event has not occurred 1 Compare event occurred

25.6.3 Load register (EPITx_LR)

The EPIT load register (EPIT_LR) contains the value that is to be loaded into the counter when EPIT counter reaches zero if the RLD bit in EPIT_CR is set. If the IOVW bit in the EPIT_CR is set then a write to this register overwrites the value of the EPIT counter register in addition to updating this registers value. This overwrite feature is active even if the RLD bit is not set.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LOAD																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

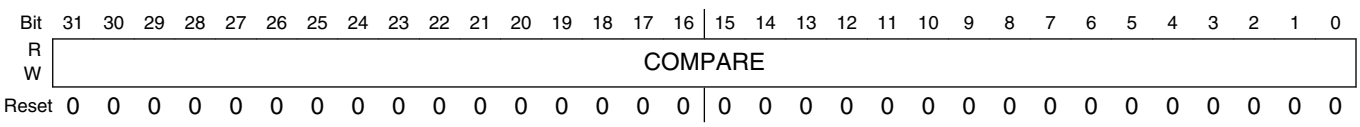
EPITx_LR field descriptions

Field	Description
LOAD	Load value. Value that is loaded into the counter at the start of each count cycle.

25.6.4 Compare register (EPITx_CMPR)

The EPIT compare register (EPIT_CMPR) holds the value that determines when a compare event is generated.

Address: Base address + Ch offset



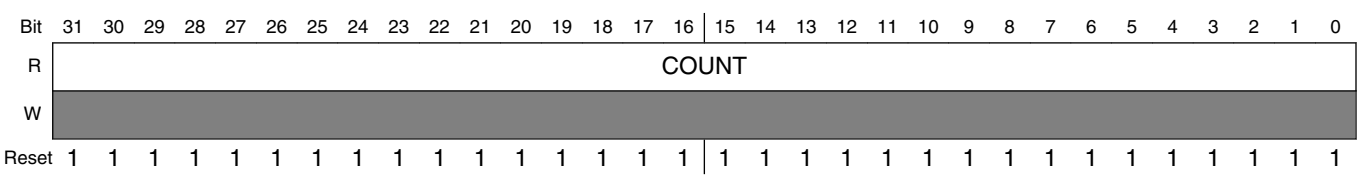
EPITx_CMPR field descriptions

Field	Description
COMPARE	Compare Value. When the counter value equals this bit field value a compare event is generated.

25.6.5 Counter register (EPITx_CNR)

The EPIT counter register (EPIT_CNR) contains the current count value and can be read at any time without disturbing the counter. This is a read-only register and any attempt to write into it generates a transfer error. But if the IOVW bit in EPIT_CR is set, the value of this register can be overwritten with a write to EPIT_LR. This change is reflected when this register is subsequently read.

Address: Base address + 10h offset



EPITx_CNR field descriptions

Field	Description
COUNT	Counter value. This contains the current value of the counter.

Chapter 26
Enhanced Serial Audio Interface (ESAI)

26.1 Overview

The Enhanced Serial Audio Interface (ESAI) provides a full-duplex serial port for serial communication with a variety of serial devices, including industry-standard codecs, Sony/Phillips Digital Interface (SPDIF) transceivers, and other DSPs.

The ESAI consists of independent transmitter and receiver sections, each section with its own clock generator. It is a superset of the 56300 Family ESSI peripheral and of the 56000 Family SAI peripheral.

All serial transfers in the module are synchronized to a clock. Additional synchronization signals are used to delineate the word frames. The normal mode of operation is used to transfer data at a periodic rate, one word per period. The network mode is similar in that it is also intended for periodic transfers; however, it supports up to 32 words (time slots) per period. This mode can be used to build time division multiplexed (TDM) networks. In contrast, the on-demand mode is intended for non-periodic transfers of data and to transfer data serially at high speed when the data becomes available.

The following figure shows the ESAI block diagram.

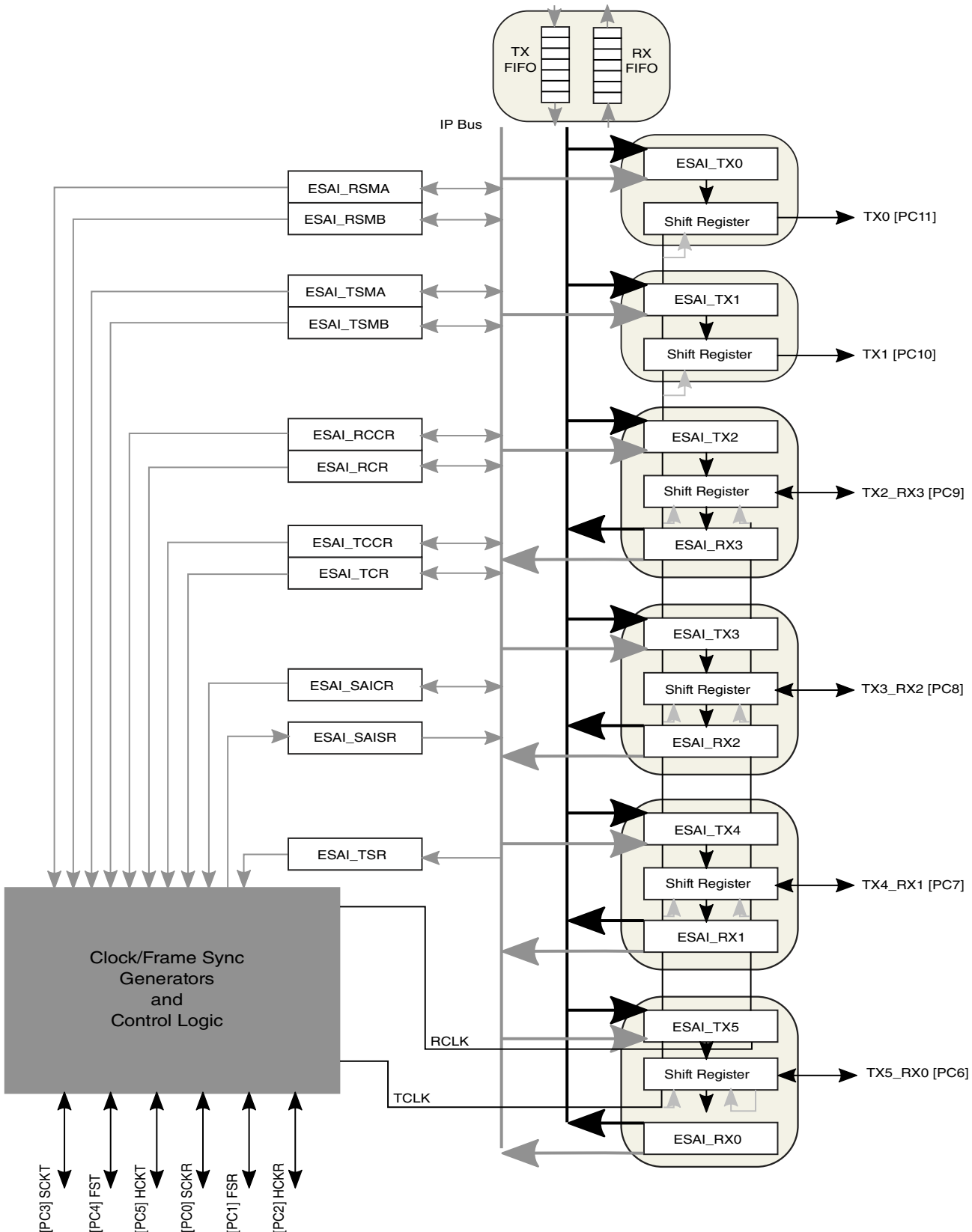


Figure 26-1. ESAI Block Diagram

26.1.1 Features

- Independent (asynchronous mode) or shared (synchronous mode) transmit and receive sections with separate or shared internal/external clocks and frame syncs, operating in Master or Slave mode.
- Up to six transmitters and four receivers with TX2_RX3, TX3_RX2, TX4_RX1, and TX5_RX0 pins shared by transmitters 2 to 5 and receivers 0 to 3. TX0 AND TX1 pins are used by transmitters 0 and 1 only.
- Programmable data interface modes such as I2S, LSB aligned, MSB aligned
- Programmable word length (8, 12, 16, 20 or 24bits)
- Flexible selection between system clock or external oscillator as input clock source, programmable internal clock divider and frame sync generation
- AC97 support
- Time Slot Mask Registers for reduced ARM platform overhead (for both Transmit and Receive)
- 128-word Transmit FIFO shared by six transmitters
- 128-word Receive FIFO shared by four receivers

26.1.2 Modes of Operation

ESAI has three basic operating modes and many data/operation formats.

ESAI operating mode are selected by the ESAI control registers (ESAI_TCCR, ESAI_TCR, ESAI_RCCR, ESAI_RCR, and ESAI_SAICR). The main operating modes are described in the following section.

26.1.2.1 Normal/Network/On-Demand Mode Selection

Selecting between the normal mode and network mode is accomplished by clearing or setting the TMOD0-TMOD1 bits in the ESAI_TCR register for the transmitter section, as well as in the RMOD0-RMOD1 bits in the ESAI_RCR register for the receiver section.

For normal mode, the ESAI functions with one data word of I/O per frame (per enabled transmitter or receiver). The normal mode is typically used to transfer data to or from a single device.

For the network mode, 2 to 32 time slots per frame may be selected. During each frame, 0 to 32 data words of I/O may be received or transmitted. In either case, the transfers are periodic. The frame sync signal indicates the first time slot in the frame. Network mode is typically used in time division multiplexed (TDM) networks of codecs, DSPs with multiple words per frame, or multi-channel devices.

Selecting the network mode and setting the frame rate divider to zero (DC=00000) selects the on-demand mode. This special case does not generate a periodic frame sync. A frame sync pulse is generated only when data is available to transmit. The on-demand mode requires that the transmit frame sync be internal (output) and the receive frame sync be external (input). Therefore, for simplex operation, the synchronous mode could be used; however, for full-duplex operation, the asynchronous mode must be used. Data transmission that is data driven is enabled by writing data into each TX. Although the ESAI is double buffered, only one word can be written to each TX, even if the transmit shift register is empty. The receive and transmit interrupts function as usual using TDE and RDF; however, transmit underruns are impossible for on-demand transmission and are disabled.

26.1.2.2 Synchronous/Asynchronous Operating Modes

The transmit and receive sections of the ESAI may be synchronous or asynchronous, that is, the transmitter and receiver sections may use common clock and synchronization signals (synchronous operating mode), or they may have their own separate clock and sync signals (asynchronous operating mode).

The SYN bit in the ESAI_SAICR register selects synchronous or asynchronous operation. Because the ESAI is designed to operate either synchronously or asynchronously, separate receive and transmit interrupts are provided.

When SYN is cleared, the ESAI transmitter and receiver clocks and frame sync sources are independent. If SYN is set, the ESAI transmitter and receiver clocks and frame sync come from the transmitter section (either external or internal sources).

Data clock and frame sync signals can be generated internally by the ARM Core or may be obtained from external sources. If internally generated, the ESAI clock generator is used to derive high frequency clock, bit clock and frame sync signals from the ARM Core internal system clock.

26.1.2.3 Frame Sync Selection

The frame sync can be either a bit-long or word-long signal.

The transmitter frame format is defined by the TFSL bit in the ESAI_TCR register. The receiver frame format is defined by the RFSL bit in the ESAI_RCR register.

1. In the word-long frame sync format, the frame sync signal is asserted during the entire word data transfer period. This frame sync length is compatible with codecs, SPI serial peripherals, serial A/D and D/A converters, shift registers and telecommunication PCM serial I/O.
2. In the bit-long frame sync format, the frame sync signal is asserted for one bit clock immediately before the data transfer period. This frame sync length is compatible with Intel and National components, codecs and telecommunication PCM serial I/O.

The relative timing of the word length frame sync as referred to the data word is specified by the TFSR bit in the ESAI_TCR register for the transmitter section and by the RFSR bit in the ESAI_RCR register for the receive section. The word length frame sync may be generated (or expected) with the first bit of the data word, or with the last bit of the previous word. TFSR and RFSR are ignored when a bit length frame sync is selected.

Polarity of the frame sync signal may be defined as positive (asserted high) or negative (asserted low). The TFSP bit in the ESAI_TCCR register specifies the polarity of the frame sync for the transmitter section. The RFSP bit in the ESAI_RCCR register specifies the polarity of the frame sync for the receiver section.

The ESAI receiver looks for a receive frame sync leading edge (trailing edge if RFSP is set) only when the previous frame is completed. If the frame sync goes high before the frame is completed (or before the last bit of the frame is received in the case of a bit frame sync or a word length frame sync with RFSR set), the current frame sync is not recognized, and the receiver is internally disabled until the next frame sync. Frames do not have to be adjacent, that is, a new frame sync does not have to immediately follow the previous frame. Gaps of arbitrary periods can occur between frames. Enabled transmitters are tri-stated during these gaps.

When operating in the synchronous mode (SYN=1), all clocks including the frame sync are generated by the transmitter section.

26.1.2.4 Shift Direction Selection

Some data formats, such as those used by codecs, specify MSB first while other data formats, such as the AES-EBU digital audio interface, specify LSB first.

The MSB/LSB first selection is made by programming RSHFD bit in the ESAI_RCR register for the receiver section and by programming the TSHFD bit in the ESAI_TCR register for the transmitter section.

26.2 External Signals

Three to twelve pins are required for operation, depending on the operating mode selected and the number of transmitters and receivers enabled.

The TX0 and TX1 pins are used by transmitters 0 and 1 only. The TX2_RX3, TX3_RX2, TX4_RX1, and TX5_RX0 pins are shared by transmitters 2 to 5 with receivers 0 to 3. The actual mode of operation is selected under software control. All transmitters operate fully synchronized under control of the same transmitter clock signals. All receivers operate fully synchronized under control of the same receiver clock signals.

The following table describes the external signals of ESAI:

Table 26-1. ESAI External Signals

Signal	Description	Pad	Mode	Direction
ESAI_RX_CLK	RX serial bit clock for the ESAI interface. The direction can be programmed.	ENET_MDIO	ALT2	IO
		GPIO_1	ALT0	
ESAI_RX_FS	RX frame sync signal for the ESAI interface.	ENET_REF_CLK	ALT2	IO
		GPIO_9	ALT0	
ESAI_RX_HF_CLK	RX high frequency clock for the ESAI interface.	ENET_RX_ER	ALT2	IO
		GPIO_3	ALT0	
ESAI_TX0	Used for transmitting data from the ESAI_TX0 serial transmit shift register.	GPIO_17	ALT0	IO
		NANDF_CS2	ALT2	
ESAI_TX1	Used for transmitting data from the ESAI_TX1 serial transmit shift register.	GPIO_18	ALT0	IO
		NANDF_CS3	ALT2	
ESAI_TX2_RX3	Used as TX2 for transmitting data from the ESAI_TX2 serial transmit shift register when programmed as a transmitter pin Used as the RX3 signal for receiving serial data to the ESAI_RX3 serial receive shift register when programmed as a receiver pin	ENET_TXD1	ALT2	IO
		GPIO_5	ALT0	
ESAI_TX3_RX2	Used as TX3 for transmitting data from the ESAI_TX3 serial transmit shift register when programmed as a transmitter pin Used as the RX2 signal for receiving serial data to the ESAI_RX2 serial receive shift register when programmed as a receiver pin	ENET_TX_EN	ALT2	IO
		GPIO_16	ALT0	

Table continues on the next page...

Table 26-1. ESAI External Signals (continued)

Signal	Description	Pad	Mode	Direction
ESAI_TX4_RX1	Used as TX4 for transmitting data from the ESAI_TX4 serial transmit shift register when programmed as a transmitter pin Used as the RX1 signal for receiving serial data to the ESAI_RX1 serial receive shift register when programmed as a receiver pin	ENET_TXD0	ALT2	IO
		GPIO_7	ALT0	
ESAI_TX5_RX0	Used as TX5 for transmitting data from the ESAI_TX5 serial transmit shift register when programmed as a transmitter pin Used as the RX0 signal for receiving serial data to the ESAI_RX0 serial receive shift register when programmed as a receiver pin	ENET_MDC	ALT2	IO
		GPIO_8	ALT0	
ESAI_TX_CLK	TX serial bit clock for the ESAI interface. The direction can be programmed.	ENET_CRS_DV	ALT2	IO
		GPIO_6	ALT0	
ESAI_TX_FS	Frame sync for both the transmitters and receivers in the synchronous mode (SYN=1) and for the transmitters only in asynchronous mode	ENET_RXD1	ALT2	IO
		GPIO_2	ALT0	
ESAI_TX_HF_CLK	TX high frequency clock for the ESAI interface.	ENET_RXD0	ALT2	IO
		GPIO_4	ALT0	

26.2.1 Serial Transmit 0 Data Pin

TX0 is used for transmitting data from the ESAI_TX0 serial transmit shift register.

TX0 is an output when data is being transmitted from the ESAI_TX0 shift register. In the on-demand mode with an internally generated bit clock, the TX0 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

TX0 may be programmed as a disconnected pin (PC11) when the ESAI TX0 function is not being used. See Table 26-14.

26.2.2 Serial Transmit 1 Data Pin

TX1 is used for transmitting data from the ESAI_TX1 serial transmit shift register.

TX1 is an output when data is being transmitted from the ESAI_TX1 shift register. In the on-demand mode with an internally generated bit clock, the TX1 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

TX1 may be programmed as a disconnected pin (PC10) when the ESAI TX1 function is not being used. See Table 26-14.

26.2.3 Serial Transmit 2/Receive 3 Data Pin

TX2_RX3 is used as the TX2 for transmitting data from the ESAI_TX2 serial transmit shift register when programmed as a transmitter pin, or as the RX3 signal for receiving serial data to the ESAI_RX3 serial receive shift register when programmed as a receiver pin.

TX2_RX3 is an input when data is being received by the ESAI_RX3 shift register. TX2_RX3 is an output when data is being transmitted from the ESAI_TX2 shift register. In the on-demand mode with an internally generated bit clock, the TX2_RX3 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

TX2_RX3 may be programmed as a disconnected pin (PC9) when the ESAI TX2 and RX3 functions are not being used. See Table 26-14.

26.2.4 Serial Transmit 3/Receive 2 Data Pin

TX3_RX2 is used as the TX3 signal for transmitting data from the ESAI_TX3 serial transmit shift register when programmed as a transmitter pin, or as the RX2 signal for receiving serial data to the ESAI_RX2 serial receive shift register when programmed as a receiver pin.

TX3_RX2 is an input when data is being received by the ESAI_RX2 shift register. TX3_RX2 is an output when data is being transmitted from the ESAI_TX3 shift register. In the on-demand mode with an internally generated bit clock, the TX3_RX2 pin

becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

TX3_RX2 may be programmed as a disconnected pin (PC8) when the ESAI TX3 and RX2 functions are not being used. See Table 26-14.

26.2.5 Serial Transmit 4/Receive 1 Data Pin

TX4_RX1 is used as the TX4 signal for transmitting data from the ESAI_TX4 serial transmit shift register when programmed as transmitter pin, or as the RX1 signal for receiving serial data to the RX1 serial receive shift register when programmed as a receiver pin.

TX4_RX1 is an input when data is being received by the ESAI_RX1 shift register. TX4_RX1 is an output when data is being transmitted from the ESAI_TX4 shift register. In the on-demand mode with an internally generated bit clock, the TX4_RX1 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

TX4_RX1 may be programmed as a disconnected pin (PC7) when the ESAI TX4 and RX1 functions are not being used. See Table 26-14.

26.2.6 Serial Transmit 5/Receive 0 Data Pin

TX5_RX0 is used as the TX5 signal for transmitting data from the ESAI_TX5 serial transmit shift register when programmed as transmitter pin, or as the RX0 signal for receiving serial data to the ESAI_RX0 serial shift register when programmed as a receiver pin.

TX5_RX0 is an input when data is being received by the ESAI_RX0 shift register. TX5_RX0 is an output when data is being transmitted from the ESAI_TX5 shift register. In the on-demand mode with an internally generated bit clock, the TX5_RX0 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

TX5_RX0 may be programmed as a disconnected pin (PC6) when the ESAI TX5 and RX0 functions are not being used. See Table 26-14.

26.2.7 Receiver Serial Clock

SCKR is a bidirectional pin providing the receivers serial bit clock for the ESAI interface.

The direction of this pin is determined by the RCKD bit in the ESAI_RCCR register. The SCKR operates as a clock input or output used by all the enabled receivers in the asynchronous mode (SYN = 0), or as serial flag 0 pin in the synchronous mode (SYN = 1).

When this pin is configured as serial flag pin, its direction is determined by the RCKD bit in the ESAI_RCCR register. When configured as the output flag OF0, this pin reflects the value of the OF0 bit in the ESAI_SAICR register, and the data in the OF0 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When this pin is configured as the input flag IF0, the data value at the pin is stored in the IF0 bit in the ESAI_SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

SCKR may be programmed as a disconnected pin (PC0) when the ESAI SCKR function is not being used. See Table 26-14.

NOTE

Although the external ESAI serial clocks can be independent of and asynchronous to the internal ESAI system clock, the external ESAI serial clock cannot exceed the internal ESAI system clock divided by 6.

For SCKR pin mode definitions, see Table 26-11.

The table below provides a list of asynchronous-mode receiver clock sources. For more information about EXTAL/ESAI clocking control bits (ERI, ERO), refer to ESAI Control Register (ESAI_ECR).

Table 26-2. Receiver Clock Sources (Asynchronous Mode Only)

RHCKD	RFSD	RCKD	ERI	ERO	Receiver Bit Clock Source	OUTPUTS		
0	0	0	N/A	N/A	SCKR	-	-	-
0	0	1	N/A	N/A	HCKR	-	-	SCKR
0	1	0	N/A	N/A	SCKR	-	FSR	-
0	1	1	N/A	N/A	HCKR	-	FSR	SCKR
1	0	0	0	0	SCKR	HCKR	-	-
1	0	0	0	1	SCKR	HCKR	-	-

Table continues on the next page...

Table 26-2. Receiver Clock Sources (Asynchronous Mode Only) (continued)

RHCKD	RFSD	RCKD	ERI	ERO	Receiver Bit Clock Source	OUTPUTS		
1	0	0	1	0	SCKR	HCKR	-	-
1	0	0	1	1	SCKR	HCKR	-	-
1	0	1	0	0	Fsys[1]	HCKR	-	SCKR
1	0	1	0	1	Fsys	HCKR	-	SCKR
1	0	1	1	0	EXTAL[2]	HCKR	-	SCKR
1	0	1	1	1	EXTAL	HCKR	-	SCKR
1	1	0	0	0	SCKR	HCKR	FSR	-
1	1	0	0	1	SCKR	HCKR	FSR	-
1	1	0	1	0	SCKR	HCKR	FSR	-
1	1	0	1	1	SCKR	HCKR	FSR	-
1	1	1	0	0	Fsys	HCKR	FSR	SCKR
1	1	1	0	1	Fsys	HCKR	FSR	SCKR
1	1	1	1	0	EXTAL	HCKR	FSR	SCKR
1	1	1	1	1	EXTAL	HCKR	FSR	SCKR
Fsys = ipg_clk_esai EXTAL is the on-chip clock source other than ipg_clk_esai ESAI system clock, and it is from esai_clk_root in CCM.								

26.2.8 Transmitter Serial Clock

SCKT is a bidirectional pin providing the transmitters serial bit clock for the ESAI interface.

The direction of this pin is determined by the TCKD bit in the ESAI_TCCR register. The SCKT is a clock input or output used by all the enabled transmitters in the asynchronous mode (SYN = 0) or by all the enabled transmitters and receivers in the synchronous mode (SYN = 1).

The following table provides a list of asynchronous-mode transmitter clock sources.

Table 26-3. Transmitter Clock Sources (Asynchronous Mode Only)

THCKD	TFSD	TCKD	ETI	ETO	Transmitter Bit Clock Source	OUTPUTS		
0	0	0	N/A	N/A	SCKT	-	-	-
0	0	1	N/A	N/A	HCKT	-	-	SCKT
0	1	0	N/A	N/A	SCKT	-	FST	-

Table continues on the next page...

Table 26-3. Transmitter Clock Sources (Asynchronous Mode Only) (continued)

THCKD	TFSD	TCKD	ETI	ETO	Transmitter Bit Clock Source	OUTPUTS		
0	1	1	N/A	N/A	HCKT	-	FST	SCKT
1	0	0	0	0	SCKT	HCKT	-	-
1	0	0	0	1	SCKT	HCKT	-	-
1	0	0	1	0	SCKT	HCKT	-	-
1	0	0	1	1	SCKT	HCKT	-	-
1	0	1	0	0	Fsys[1]	HCKT	-	SCKT
1	0	1	0	1	Fsys	HCKT	-	SCKT
1	0	1	1	0	EXTAL[2]	HCKT	-	SCKT
1	0	1	1	1	EXTAL	HCKT	-	SCKT
1	1	0	0	0	SCKR	HCKT	FST	-
1	1	0	0	1	SCKR	HCKT	FST	-
1	1	0	1	0	SCKR	HCKT	FST	-
1	1	0	1	1	SCKR	HCKT	FST	-
1	1	1	0	0	Fsys	HCKT	FST	SCKT
1	1	1	0	1	Fsys	HCKT	FST	SCKT
1	1	1	1	0	EXTAL	HCKT	FST	SCKT
1	1	1	1	1	EXTAL	HCKT	FST	SCKT

Fsys = ipg_clk_esai

EXTAL is the on-chip clock sources other than ipg_clk_easi ESAI system clock, and it is from esai_clk_root in CCM

SCKT may be programmed as a disconnected pin (PC3) when the ESAI SCKT function is not being used. See Table 26-14.

For more information about EXTAL/ESAI clocking control bits (ETI, ETO), see ESAI Control Register (ESAI_ECR).

NOTE

Although the external ESAI serial clocks can be independent of and asynchronous to the internal ESAI system clock, the external ESAI serial clock cannot exceed the internal ESAI system clock divided by 6.

26.2.9 Frame Sync for Receiver

FSR is a bidirectional pin providing the receivers frame sync signal for the ESAI interface. The direction of this pin is determined by the RFSD bit in ESAI_RCR register.

In the asynchronous mode (SYN=0), the FSR pin operates as the frame sync input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as either the serial flag 1 pin (TEBE=0), or as the transmitter external buffer enable control (TEBE=1, RFSD=1). For FSR pin mode definitions, see Table 26-12; for receiver clock signals, see Table 26-2.

When this pin is configured as serial flag pin, its direction is determined by the RFSD bit in the ESAI_RCCR register. When configured as the output flag OF1, this pin reflects the value of the OF1 bit in the ESAI_SAICR register, and the data in the OF1 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When configured as the input flag IF1, the data value at the pin is stored in the IF1 bit in the ESAI_SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

FSR may be programmed as a disconnected pin (PC1) when the ESAI FSR function is not being used. See Table 26-14.

26.2.10 Frame Sync for Transmitter

FST is a bidirectional pin providing the frame sync for both the transmitters and receivers in the synchronous mode (SYN=1) and for the transmitters only in asynchronous mode (SYN=0).

See Table 26-3. The direction of this pin is determined by the TFSD bit in the ESAI_TCCR register. When configured as an output, this pin is the internally generated frame sync signal. When configured as an input, this pin receives an external frame sync signal for the transmitters (and the receivers in synchronous mode).

FST may be programmed as a disconnected pin (PC4) when the ESAI FST function is not being used. See Table 26-14.

26.2.11 High Frequency Clock for Transmitter

HCKT is a bidirectional pin providing the transmitters high frequency clock for the ESAI interface.

The direction of this pin is determined by the THCKD bit in the ESAI_TCCR register. In the asynchronous mode (SYN=0), the HCKT pin operates as the high frequency clock input or output used by all enabled transmitters. In the synchronous mode (SYN=1), it operates as the high frequency clock input or output used by all enabled transmitters and receivers. When programmed as input this pin is used as an alternative high frequency

clock source to the ESAI transmitter rather than the ARM Core main clock. When programmed as output it can serve as a high frequency sample clock (to external DACs for example) or as an additional system clock (see Table 26-3).

HCKT may be programmed as a disconnected pin (PC5) when the ESAI HCKT function is not being used. See Table 26-14.

26.2.12 High Frequency Clock for Receiver

HCKR is a bidirectional pin providing the receivers high frequency clock for the ESAI interface.

The direction of this pin is determined by the RHCKD bit in the ESAI_RCCR register. In the asynchronous mode (SYN=0), the HCKR pin operates as the high frequency clock input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as the serial flag 2 pin. For HCKR pin mode definitions, see Table 26-13; for receiver clock signals, see Table 26-2.

When this pin is configured as serial flag pin, its direction is determined by the RHCKD bit in the ESAI_RCCR register. When configured as the output flag OF2, this pin reflects the value of the OF2 bit in the ESAI_SAICR register, and the data in the OF2 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When configured as the input flag IF2, the data value at the pin is stored in the IF2 bit in the ESAI_SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

HCKR may be programmed as a disconnected pin (PC2) when the ESAI HCKR function is not being used. See Table 26-14.

26.2.13 Serial I/O Flags

Three ESAI pins (FSR, SCKR and HCKR) are available as serial I/O flags when the ESAI is operating in the synchronous mode (SYN=1).

Their operation is controlled by RCKD, RFSD, TEBE bits in the ESAI_RCR, ESAI_RCCR and ESAI_SAICR registers.The output data bits (OF2, OF1 and OF0) and the input data bits (IF2, IF1 and IF0) are double buffered to/from the HCKR, FSR and SCKR pins. Double buffering the flags keeps them in sync with the TX and RX data lines.

Each flag can be separately programmed. Flag 0 (SCKR pin) direction is selected by RCKD, RCKD=1 for output and RCKD=0 for input. Flag 1 (FSR pin) is enabled when the pin is not configured as external transmitter buffer enable (TEBE=0) and its direction is selected by RFSD, RFSD=1 for output and RFSD=0 for input. Flag 2 (HCKR pin) direction is selected by RHCKD, RHCKD=1 for output and RHCKD=0 for input.

When programmed as input flags, the SCKR, FSR and HCKR logic values, respectively, are latched at the same time as the first bit of the receive data word is sampled. Because the input was latched, the signal on the input flag pin (SCKR, FSR or HCKR) can change without affecting the input flag until the first bit of the next receive data word. When the received data words are transferred to the receive data registers, the input flag latched values are then transferred to the IF0, IF1 and IF2 bits in the SAISR register, where they may be read by software.

When programmed as output flags, the SCKR, FSR and HCKR logic values are driven by the contents of the OF0, OF1 and OF2 bits in the ESAI_SAICR register respectively, and they are driven when the transmit data registers are transferred to the transmit shift registers. The value on SCKR, FSR and HCKR is stable from the time the first bit of the transmit data word is transmitted until the first bit of the next transmit data word is transmitted. Software may change the OF0-OF2 values thus controlling the SCKR, FSR and HCKR pin values for each transmitted word. The normal sequence for setting output flags when transmitting data is as follows: wait for TDE (transmitter empty) to be set; first write the flags, and then write the transmit data to the transmit registers. OF0, OF1, and OF2 are double buffered so that the flag states appear on the pins when the transmit data is transferred to the transmit shift register, that is, the flags are synchronous with the data.

26.3 Clocks

The table found here describes the clock sources for ESAI.

Please see clock control block for clock setting, configuration and gating information.

Table 26-4. ESAI Clocks

Clock name	Clock Root	Description
extal_clk	esai_clk_root	ESAI system clock
ipg_clk_esai	ahb_clk_root	Bus clock
ipg_clk_s	ipg_clk_root	Peripheral access clock
mem_clk	ahb_clk_root	Mem clock

26.4 Functional Description

This section provides a complete functional description of the block.

26.4.1 ESAI After Reset

Hardware or software reset clears the port control register bits and the port direction control register bits, which configure all ESAI I/O pins as disconnected and both ESAI FIFOs are also in reset state.

The ESAI is in personal reset state while all ESAI pins are programmed as disconnected, and it is active only if at least one of the ESAI I/O pins is programmed as an ESAI pin.

26.4.2 ESAI Interrupt Requests

The ESAI can generate eight different interrupt requests.

Ordered from the highest to the lowest priority):

1. **ESAI Receive Data with Exception Status**

   Occurs when the receive exception interrupt is enabled (REIE=1 in the RCR register), at least one of the enabled receive data registers is full (RDF=1) and a receiver overrun error has occurred (ROE=1 in the SAISR register). ROE is cleared by first reading the SAISR and then reading all the enabled receive data registers.

2. **ESAI Receive Even Data**

   Occurs when the receive even slot data interrupt is enabled (REDIE=1), at least one of the enabled receive data registers is full (RDF=1), the data is from an even slot (REDF=1) and no exception has occurred (ROE=0 or REIE=0).

   Reading all enabled receiver data registers clears RDF and REDF.

3. **ESAI Receive Data**

   Occurs when the receive interrupt is enabled (RIE=1), at least one of the enabled receive data registers is full (RDF=1), no exception has occurred (ROE=0 or REIE=0) and no even slot interrupt has occurred (REDF=0 or REDIE=0). Reading all enabled receiver data registers clears RDF.

4. **ESAI Receive Last Slot Interrupt**

Occurs, if enabled (RLIE=1), after the last slot of the frame ended (in network mode only) regardless of the receive mask register setting. The receive last slot interrupt may be used for resetting the receive mask slot register, reconfiguring the DMA channels and reassigning data memory pointers. Using the receive last slot interrupt guarantees that the previous frame was serviced with the previous setting and the new frame is serviced with the new setting without synchronization problems. Note that the maximum receive last slot interrupt service time should not exceed N-1 ESAI bits service time (where N is the number of bits in a slot).

5. **ESAI Transmit Data with Exception Status**

   Occurs when the transmit exception interrupt is enabled (TEIE=1), at least one transmit data register of the enabled transmitters is empty (TDE=1) and a transmitter underrun error has occurred (TUE=1). TUE is cleared by first reading the SAISR and then writing to all the enabled transmit data registers, or to the TSR register.

6. **ESAI Transmit Last Slot Interrupt**

   Occurs, if enabled (TLIE=1), at the start of the last slot of the frame in network mode regardless of the transmit mask register setting. The transmit last slot interrupt may be used for resetting the transmit mask slot register, reconfiguring the DMA channels and reassigning data memory pointers. Using the transmit last slot interrupt guarantees that the previous frame was serviced with the previous setting and the new frame is serviced with the new setting without synchronization problems. Note that the maximum transmit last slot interrupt service time should not exceed N-1 ESAI bits service time (where N is the number of bits in a slot).

7. **ESAI Transmit Even Data**

   Occurs when the transmit even slot data interrupt is enabled (TEDIE=1), at least one of the enabled transmit data registers is empty (TDE=1), the slot is an even slot (TEDE=1) and no exception has occurred (TUE=0 or TEIE=0). Writing to all the TX registers of the enabled transmitters or to TSR clears this interrupt request.

8. **ESAI Transmit Data**

   Occurs when the transmit interrupt is enabled (TIE=1), at least one of the enabled transmit data registers is empty (TDE=1), no exception has occurred (TUE=0 or TEIE=0) and no even slot interrupt has occurred (TEDE=0 or TEDIE=0). Writing to all the TX registers of the enabled transmitters, or to the TSR clears this interrupt request.

26.4.3 ESAI DMA Requests from the FIFOs

The ESAI can generate two different DMA requests:

1. ESAI Transmit FIFO Empty - Asserts when the number of empty slots in the ESAI transmit FIFO exceeds the threshold programmed in the ESAI Transmit FIFO Configuration Register (TFCR). Automatically negates when the number of empty slots is less than the threshold programmed in the ESAI Transmit FIFO Configuration Register.
2. ESAI Receive FIFO Full - Asserts when the number of data words in the ESAI receive FIFO exceeds the threshold programmed in the ESAI Receive FIFO Configuration Register (RFCR). Automatically negates when the number of words is less than the threshold programmed in the ESAI Receive FIFO Configuration Register.

26.4.4 ESAI Transmit and Receive Shift Registers

26.4.4.1 ESAI Transmit Shift Registers

The transmit shift registers contain the data being transmitted.

See Figure 26-2 and Figure 26-3.

Data is shifted out to the serial transmit data pins by the selected (internal/external) bit clock when the associated frame sync I/O is asserted.

The number of bits shifted out before the shift registers are considered empty and may be written to again can be 8, 12, 16, 20, 24 or 32 bits (determined by the slot length control bits in the TCR register). Data is shifted out of these registers MSB first if TSHFD=0 and LSB first if TSHFD=1.

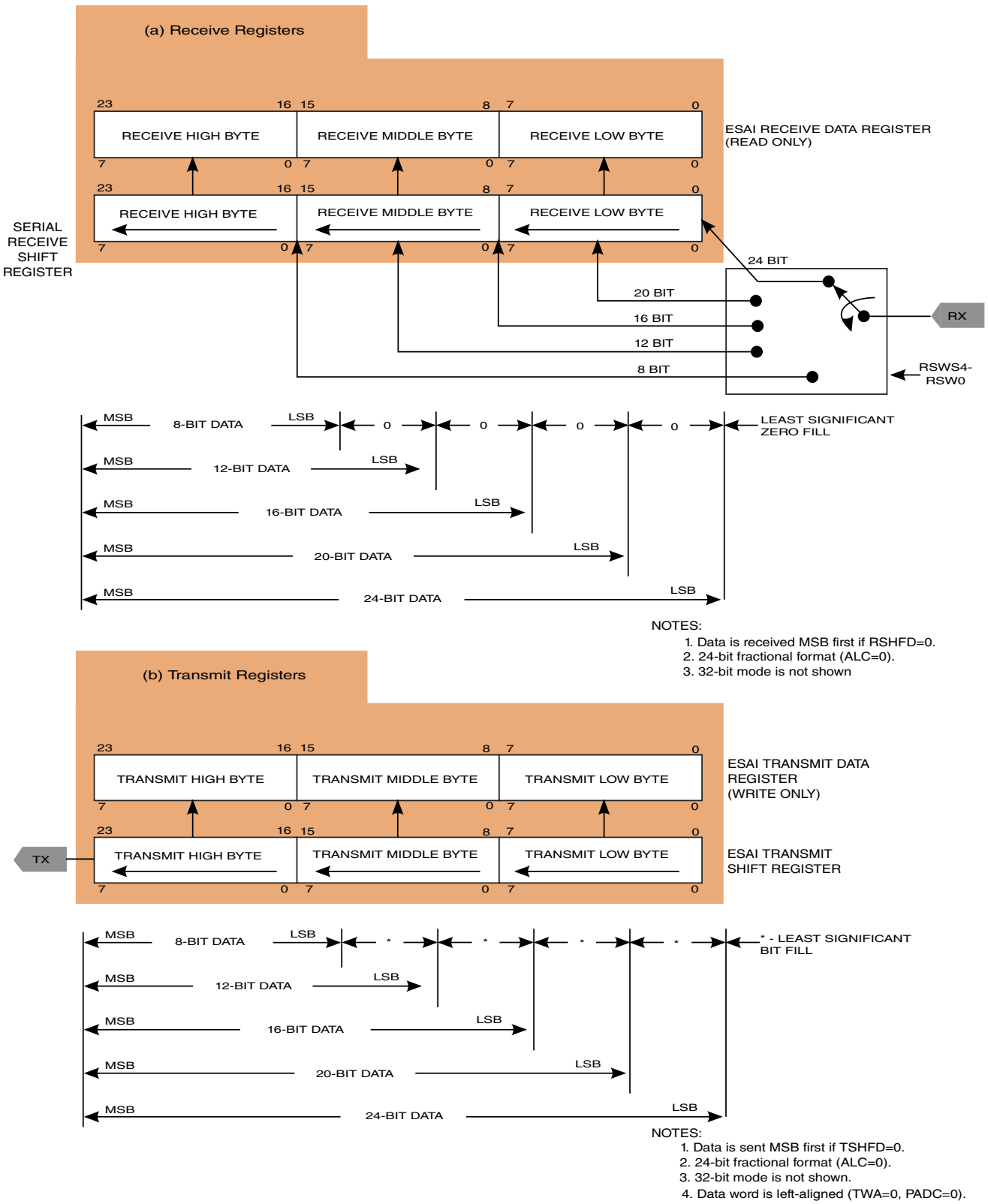


Figure 26-2. ESAI Data Path Programming Model ([R/T]SHFD=0)

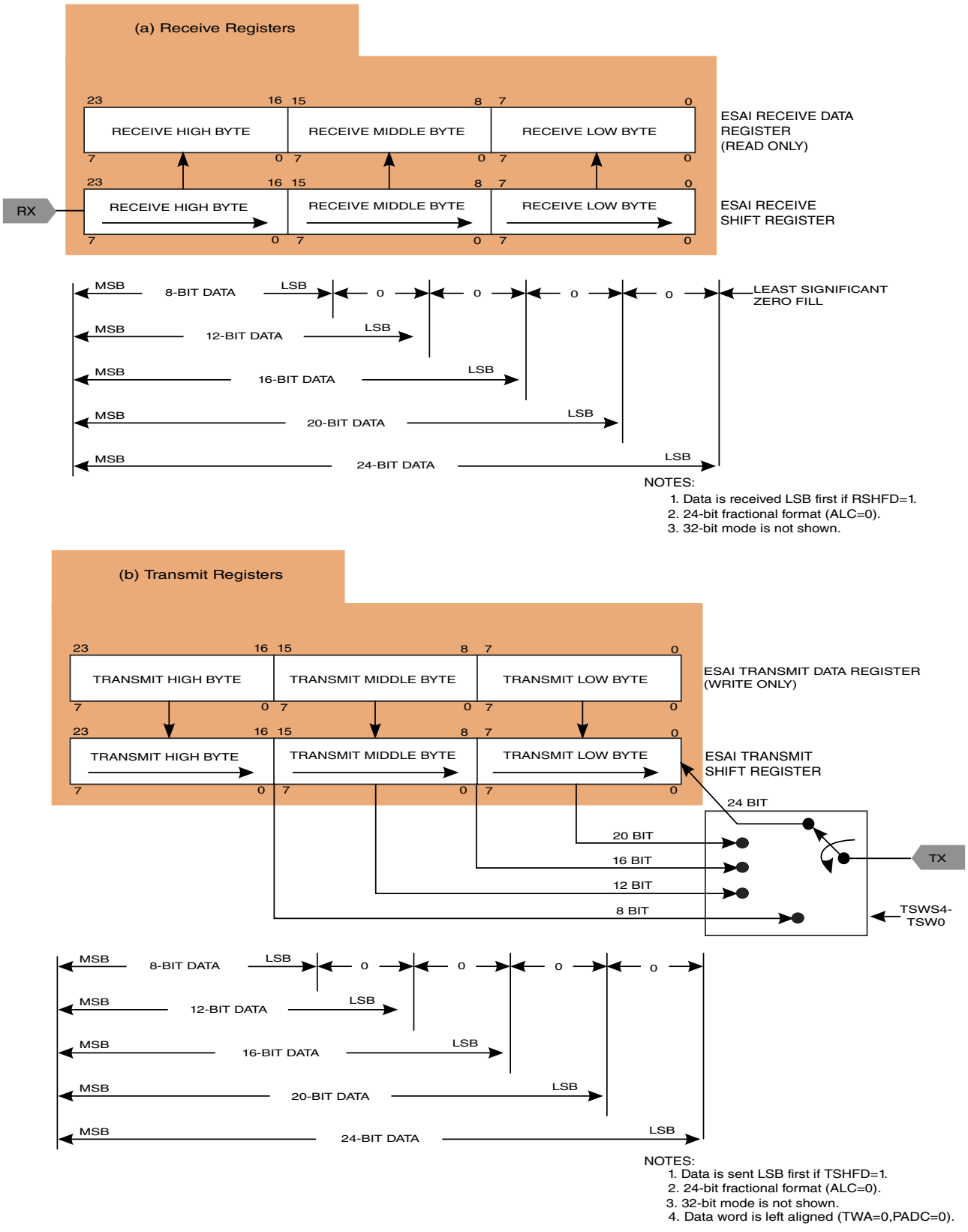


Figure 26-3. ESAI Data Path Programming Model ([R/T]SHFD=1)

26.4.4.2 ESAI Receive Shift Registers

The receive shift registers (Figure 26-2 and Figure 26-3) receive the incoming data from the serial receive data pins. Data is shifted in by the selected (internal/external) bit clock when the associated frame sync I/O is asserted. Data is assumed to be received MSB first if RSHFD=0 and LSB first if RSHFD=1. Data is transferred to the ESAI receive data registers after 8, 12, 16, 20, 24, or 32 serial clock cycles were counted, depending on the slot length control bits in the ESAI_RCR register.

26.5 Initialization Information

26.5.1 ESAI Initialization

The correct way to initialize the ESAI is as follows:

1. Enable the ESAI logic clock by asserting bit 0 of ESAI Control Register (ESAI_ECR[0]).
2. Hardware, software, ESAI individual reset. Note that asserting bit 1 of ESAI Control Register only reset the ESAI core logic, including configuration registers, but not the ESAI FIFOs.
3. Reset ESAI FIFOs by asserting bit 1 of ESAI_TFCR and ESAI_RFCR.
4. Clear the ESAI_TSMA/ESAI_TSMB.
5. Program ESAI control registers. (The transmit/receive enable bits of TCR/RCR should not be set.)
6. Program ESAI FIFOs via TFCR and RFCR. (Enable Transmit/Receive FIFO, enable transmitters/receivers, transmit initialization and set Transmit FIFO/Receive FIFO watermark.)
7. Write initial words to ESAI Transmit Data Register (ESAI_ETDR), at least one word per enabled transmitter slot but as many as desired. For example 4 channels with 2 slot-per-channel are enabled, then 8 words need to be written into ESAI_ETDR
8. Remove ESAI personal reset by configuring ESAI_PCRC and ESAI_PRRC.
9. Enabled Transmitters/Receivers in ESAI_TCR/ESAI_RCR.
10. Configure time slot registers, first set ESAI_TSMB, then ESAI_TSMA.

During program execution, all ESAI pins may be defined disconnected, causing the ESAI to stop serial activity and enter the individual reset state.

All status bits of the interface are set to their reset state however, the control bits are not affected. This procedure allows the programmer to reset the ESAI separately from the other internal peripherals. During individual reset, internal DMA accesses to the data registers of the ESAI are not valid and data read is undefined.

The programmer must use an individual ESAI reset when changing the ESAI control registers (except for TEIE, REIE, TLIE, RLIE, TIE, RIE, TE0-TE5, RE0-RE3) to ensure proper operation of the interface.

NOTE

If the ESAI receiver section is already operating with some of the receivers and enabling additional receivers on the fly, that is, without first putting the ESAI receiver in the personal reset state by setting their REx control bits, it will result in erroneous data being received as the first data word for the newly enabled receivers.

26.5.2 ESAI Initialization Examples

26.5.2.1 Initializing the ESAI using Personal Reset

1. Enable the ESAI logic clock by setting bit 0 of ESAI Control Register(ESAI_ECR[0]).
2. The ESAI should be in its personal reset state (ESAI_PCRC = 0x000 and ESAI_PRRC = 0x000). In the personal reset state, both the transmitter and receiver sections of the ESAI are simultaneously reset. The TPR bit in the ESAI_TCR register may be used to reset just the transmitter section. The RPR bit in the ESAI_RCR register may be used to reset just the receiver section.
3. Clear the ESAI_TSMA/ESAI_TSMB and ESAI_RSMA/ESAI_RSMB.
4. Configure the control registers (ESAI_TCCR, ESAI_TCR, ESAI_RCCR, ESAI_RCR) and ESAI FIFOs configuration Registers (ESAI_TFCR, ESAI_RFCR) according to the operating mode, but do not enable transmitters (TE5-TE0 = 0x0) or receivers (RE3-RE0 = 0x0). It is possible to set the interrupt enable bits which are in use during the operation (no interrupt occurs).
5. Enable the ESAI by setting the ESAI_PCRC and ESAI_PRRC register bits according to pins which are in use during operation.
6. Write initial words to ESAI Transmit Data Register (ESAI_ETDR), at least one word per enabled transmitter slot but as many as desired. For example 4 channels with 2 slot-per-channel are enabled, then 8 words need to be written into ESAI_ETDR. This step is needed even if DMA is used to service the transmitters.
7. Enable the transmitters and receivers.

8. Configure time slot registers, first set ESAI_TSMB, then ESAI_TSMA, then ESAI_RSMB, then ESAI_RSMA.
9. From now on ESAI can be serviced either by polling, interrupts, or DMA.

Operation proceeds as follows:

- For internally generated clock and frame sync, these signals are active immediately after ESAI is enabled (step 4 above).
- Data is received only when one of the receive enable (REx) bits is set and after the occurrence of frame sync signal (either internally or externally generated).
- Data is transmitted only when the transmitter enable (TEx) bit is set and after the occurrence of frame sync signal (either internally or externally generated). The transmitter outputs remain tri-stated after TEx bit is set until the frame sync occurs.

26.5.2.2 Initializing the ESAI Transmitter Section

1. It is assumed that the ESAI is operational; that is, at least one pin is defined as an ESAI pin.
2. Enable the ESAI logic clock by setting bit 0 of ESAI Control Register(ESAI_ECR[0])
3. The transmitter section should be in its individual reset state (TPR = 1) and also reset the ESAI Transmit FIFO (ESAI_TFCR[1] = 1).
4. Clear the ESAI_TSMA/ESAI_TSMB.
5. Configure the control registers ESAI_TCCR and ESAI_TCR according to the operating mode, configure the Transmit FIFO Configuration Register (bring transmit FIFO out of reset, enable Transmit FIFO, enable transmitters, transmit initialization and set watermark). Make sure to clear the transmitter enable bits (TE0-TE5). TPR must remain set.
6. Take the transmitter section out of the individual reset state by clearing TPR.
7. Write initial words to ESAI Transmit Data Register (ESAI_ETDR), at least one word per enabled transmitter slot but as many as desired. For example 4 channels with 2 slot-per-channel are enabled, then 8 words need to be written into ESAI_ETDR
8. Enable the transmitters by setting their TE bits.
9. Configure time slot registers, first set ESAI_TSMB, then ESAI_TSMA.
10. Data is transmitted only when the transmitter enable (TEx) bit is set and after the occurrence of frame sync signal (either internally or externally generated). The transmitter outputs remain tri-stated after TEx bit is set until the frame sync occurs.
11. From now on the transmitters are operating and can be serviced either by polling, interrupts, or DMA.

26.5.2.3 Initializing the ESAI Receiver Section

1. It is assumed that the ESAI is operational; that is, at least one pin is defined as an ESAI pin.
2. Enable the ESAI logic clock by setting bit 0 of ESAI Control Register (ESAI_ECR[0])
3. The receiver section should be in its individual reset state (RPR = 1) and also reset the ESAI Receive FIFO (ESAI_RFCR[1] = 1).
4. Clear the ESAI_RSMA/ESAI_RSMB.
5. Configure the control registers ESAI_RCCR and ESAI_RCR according to the operating mode, configure the Receive FIFO Configuration Register (bring receive FIFO out of reset, enable Receive FIFO, receivers, and set watermark). Making sure to clear the receiver enable bits (RE0-RE3). RPR must remain set.
6. Take the receiver section out of the individual reset state by clearing RPR.
7. Enable the receivers by setting their RE bits.
8. Configure time slot registers, first set ESAI_RSMB, then set ESAI_RSMA.
9. From now on the receivers are operating and can be serviced either by polling, interrupts, or DMA.

26.6 ESAI Memory Map/Register Definition

ESAI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
202_4000	ESAI Transmit Data Register (ESAI_ETDR)	32	W (always reads 0)	0000_0000h	26.6.1/1372
202_4004	ESAI Receive Data Register (ESAI_ERDR)	32	R	0000_0000h	26.6.2/1372
202_4008	ESAI Control Register (ESAI_ECR)	32	R/W	0000_0000h	26.6.3/1373
202_400C	ESAI Status Register (ESAI_ESR)	32	R	0000_0000h	26.6.4/1374
202_4010	Transmit FIFO Configuration Register (ESAI_TFCR)	32	R/W	0000_0000h	26.6.5/1376
202_4014	Transmit FIFO Status Register (ESAI_TFSR)	32	R	0000_0000h	26.6.6/1378
202_4018	Receive FIFO Configuration Register (ESAI_RFCR)	32	R/W	0000_0000h	26.6.7/1379
202_401C	Receive FIFO Status Register (ESAI_RFSR)	32	R	0000_0000h	26.6.8/1381
202_4080	Transmit Data Register n (ESAI_TX0)	32	W (always reads 0)	0000_0000h	26.6.9/1382
202_4084	Transmit Data Register n (ESAI_TX1)	32	W (always reads 0)	0000_0000h	26.6.9/1382

Table continues on the next page...

ESAI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
202_4088	Transmit Data Register n (ESAI_TX2)	32	W (always reads 0)	0000_0000h	26.6.9/1382
202_408C	Transmit Data Register n (ESAI_TX3)	32	W (always reads 0)	0000_0000h	26.6.9/1382
202_4090	Transmit Data Register n (ESAI_TX4)	32	W (always reads 0)	0000_0000h	26.6.9/1382
202_4094	Transmit Data Register n (ESAI_TX5)	32	W (always reads 0)	0000_0000h	26.6.9/1382
202_4098	ESAI Transmit Slot Register (ESAI_TSR)	32	W (always reads 0)	0000_0000h	26.6.10/ 1382
202_40A0	Receive Data Register n (ESAI_RX0)	32	R	0000_0000h	26.6.11/ 1383
202_40A4	Receive Data Register n (ESAI_RX1)	32	R	0000_0000h	26.6.11/ 1383
202_40A8	Receive Data Register n (ESAI_RX2)	32	R	0000_0000h	26.6.11/ 1383
202_40AC	Receive Data Register n (ESAI_RX3)	32	R	0000_0000h	26.6.11/ 1383
202_40CC	Serial Audio Interface Status Register (ESAI_SAISR)	32	R	0000_0000h	26.6.12/ 1384
202_40D0	Serial Audio Interface Control Register (ESAI_SAICR)	32	R/W	0000_0000h	26.6.13/ 1386
202_40D4	Transmit Control Register (ESAI_TCR)	32	R/W	0000_0000h	26.6.14/ 1389
202_40D8	Transmit Clock Control Register (ESAI_TCCR)	32	R/W	0000_0000h	26.6.15/ 1397
202_40DC	Receive Control Register (ESAI_RCR)	32	R/W	0000_0000h	26.6.16/ 1401
202_40E0	Receive Clock Control Register (ESAI_RCCR)	32	R/W	0000_0000h	26.6.17/ 1405
202_40E4	Transmit Slot Mask Register A (ESAI_TSMA)	32	R/W	0000_FFFFh	26.6.18/ 1408
202_40E8	Transmit Slot Mask Register B (ESAI_TSMB)	32	R/W	0000_FFFFh	26.6.19/ 1409
202_40EC	Receive Slot Mask Register A (ESAI_RSMA)	32	R/W	0000_FFFFh	26.6.20/ 1410
202_40F0	Receive Slot Mask Register B (ESAI_RSMB)	32	R/W	0000_FFFFh	26.6.21/ 1411
202_40F8	Port C Direction Register (ESAI_PRRC)	32	R/W	0000_0000h	26.6.22/ 1412

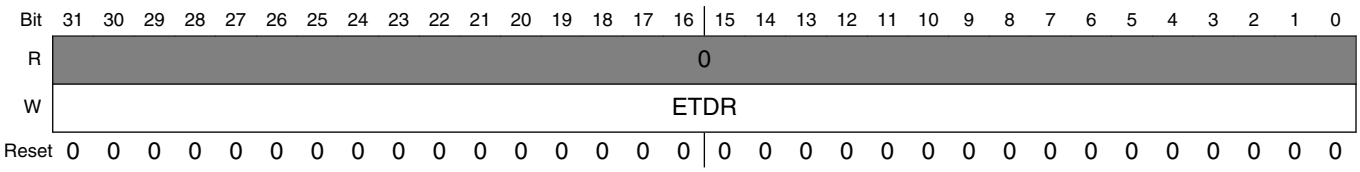
Table continues on the next page...

ESAI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
202_40FC	Port C Control Register (ESAI_PCRC)	32	R/W	0000_0000h	26.6.23/ 1412

26.6.1 ESAI Transmit Data Register (ESAI_ETDR)

Address: 202_4000h base + 0h offset = 202_4000h

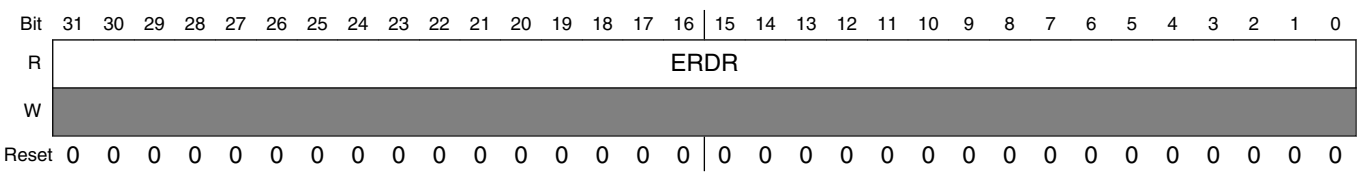


ESAI_ETDR field descriptions

Field	Description
ETDR	ESAI Transmit Data Register. Writing to this register stores the data written into the ESAI Transmit FIFO. Writing to this register when the Transmit FIFO is full causes the data written to be lost (the existing data within the FIFO is not overwritten). When multiple ESAI transmitters are enabled, the data for each transmitter must be interleaved from lowest transmitter to highest transmitter (for example, if transmitters 0, 2 and 3 are enabled then data must be written as follows: transmitter #0, transmitter #2, transmitter #3, transmitter #0, transmitter #2, transmitter #3, transmitter #0, etc). Data within the ESAI Transmit FIFO is passed to the ESAI transmit shifter registers as defined by the Transmit Word Alignment configuration bits.

26.6.2 ESAI Receive Data Register (ESAI_ERDR)

Address: 202_4000h base + 4h offset = 202_4004h

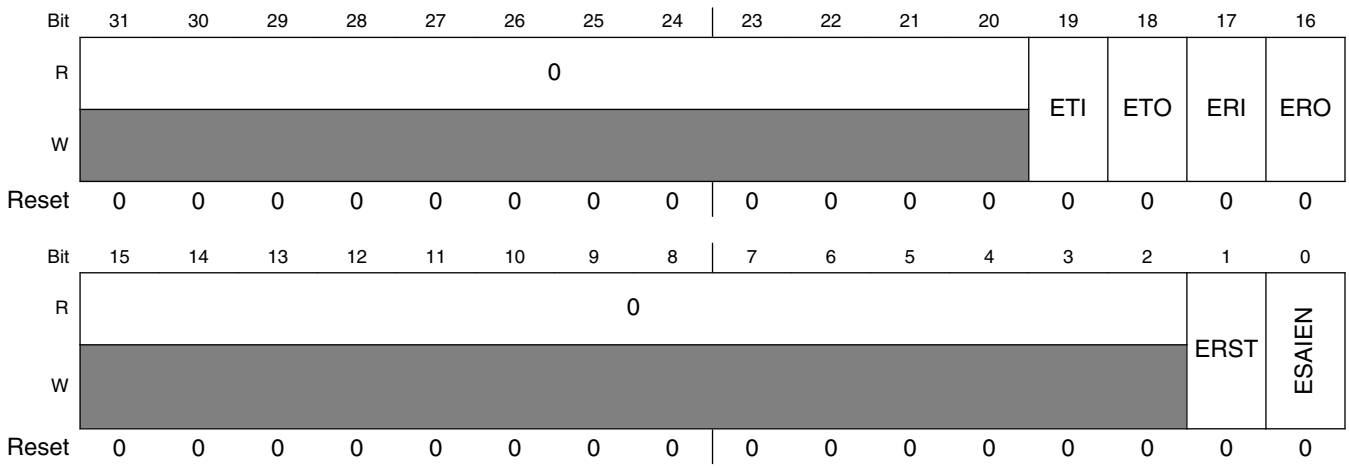


ESAI_ERDR field descriptions

Field	Description
ERDR	ESAI Receive Data Register. Reading this register returns the data within the ESAI Receive FIFO. Reading this register when the Receive FIFO is empty returns the last valid data word. When multiple ESAI receivers are enabled, the data for each receiver is interleaved from lowest receiver to highest receiver (for example, if receivers 0, 2 and 3 are enabled then data is returned as follows: receiver #0, receiver #2, receiver #3, receiver #0, receiver #2, receiver #3, receiver #0, etc). Data is passed from the ESAI receive shift registers to the ESAI Receive FIFO as defined by the Receiver Word Alignment configuration bits either zero or sign-extended based on the Receive Extension control bit.

26.6.3 ESAI Control Register (ESAI_ECR)

Address: 202_4000h base + 8h offset = 202_4008h



ESAI_ECR field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 ETI	EXTAL Transmitter In. Mux EXTAL in place of the High Frequency Transmitter Clock input pin. HCKT can still be used to drive a divided down EXTAL or as GPIO. 0 HCKT pin has normal function. 1 EXTAL muxed into HCKT input.
18 ETO	EXTAL Transmitter Out. Drive the EXTAL input on the High Frequency Transmitter Clock pin. 0 HCKT pin has normal function. 1 EXTAL driven onto HCKT pin.
17 ERI	EXTAL Receiver In. Mux EXTAL in place of the High Frequency Receiver Clock input pin. HCKR can still be used to drive a divided down EXTAL or as GPIO. 0 HCKR pin has normal function. 1 EXTAL muxed into HCKR input.
16 ERO	EXTAL Receiver Out. Drive the EXTAL input on the High Frequency Receiver Clock pin. 0 HCKR pin has normal function. 1 EXTAL driven onto HCKR pin.
15–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 ERST	ESAI Reset. Reset the ESAI core logic (including configuration registers) but not the ESAI FIFOs. 0 ESAI not reset. 1 ESAI reset.
0 ESAIEN	ESAI Enable. Enables/disables the ESAI logic clock. Enable the ESAI before reading or writing other ESAI registers.

Table continues on the next page...

ESAI_ECR field descriptions (continued)

Field	Description
	0 ESAI disabled. 1 ESAI enabled.

26.6.4 ESAI Status Register (ESAI_ESR)

Address: 202_4000h base + Ch offset = 202_400Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					TINIT	RFF	TFE	TLS	TDE	TED	TD	RLS	RDE	RED	RD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_ESR field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 TINIT	Transmit Initialization. Indicates that the Transmit FIFO is writing the first word for each enabled transmitter into the Transmit Data Registers. This bit sets when the Transmit FIFO is enabled (provided Transmit Initialization is enabled) and clears after the Transmit Data Registers have been initialized. The Transmit Enable bits in the Transmit Control Register should not be set until this flag has cleared. 0 Transmitter has finished initializing the Transmit Data Registers (or Transmit FIFO is not enabled or Transmit Initialization is not enabled). 1 Transmitter has not finished initializing the Transmit Data Registers.
9 RFF	Receive FIFO Full. Indicates that the number of data words in the Receive FIFO has equaled or exceeded the Receive FIFO Watermark. This flag also drives the ESAI Receiver DMA request line. ESAI FIFO DMA requests see ESAI DMA Requests from the FIFOs. 0 Number of words in Receive FIFO less than Receive FIFO watermark. 1 Number of words in Receive FIFO is equal to or greater than Receive FIFO watermark.
8 TFE	Transmit FIFO Empty. Indicates that the number of empty slots in the Transmit FIFO has met or exceeded the Transmit FIFO Watermark. This flag also drives the ESAI Transmitter DMA request line. ESAI FIFO DMA request see ESAI DMA Requests from the FIFOs. 0 Number of empty slots in Transmit FIFO less than Transmit FIFO watermark. 1 Number of empty slots in Transmit FIFO is equal to or greater than Transmit FIFO watermark.
7 TLS	Transmit Last Slot. Reading this register when TLS is set will negate the Transmit Last Slot interrupt. 0 TLS is not the highest priority active interrupt. 1 TLS is the highest priority active interrupt.

Table continues on the next page...

ESAI_ESR field descriptions (continued)

Field	Description
6 TDE	Transmit Data Exception. 0 TDE is not the highest priority active interrupt. 1 TDE is the highest priority active interrupt.
5 TED	Transmit Even Data. 0 TED is not the highest priority active interrupt. 1 TED is the highest priority active interrupt.
4 TD	Transmit Data. 0 TD is not the highest priority active interrupt. 1 TD is the highest priority active interrupt.
3 RLS	Receive Last Slot. Reading this register when RLS is set will negate the Receive Last Slot interrupt. 0 RLS is not the highest priority active interrupt. 1 RLS is the highest priority active interrupt.
2 RDE	Receive Data Exception. 0 RDE is not the highest priority active interrupt. 1 RDE is the highest priority active interrupt.
1 RED	Receive Even Data. 0 RED is not the highest priority active interrupt. 1 RED is the highest priority active interrupt.
0 RD	Receive Data. 0 RD is not the highest priority active interrupt. 1 RD is the highest priority active interrupt.

26.6.5 Transmit FIFO Configuration Register (ESAI_TFCR)

Address: 202_4000h base + 10h offset = 202_4010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0										TFIN	Reserved	TIEN	TWA		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TFWM								TE5	TE4	TE3	TE2	TE1	TE0	TFR	TFE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_TFCR field descriptions

Field	Description
31–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 TFIN	Tx FIFO Interrupt Enable Generate an interrupt if the "Tx FIFO Empty" flag is set. Indicate the number of empty slots in the Tx FIFO has met or exceed watermark.
20 Reserved	This field is reserved. Reserved
19 TIEN	Transmitter Initialization Enable. Enables the initialization of the Transmit Data Registers when the Transmitter FIFO is enabled. TIEN=1 is recommended. 0 Transmit Data Registers are not initialized from the FIFO once the Transmit FIFO is enabled. Software must manually initialize the Transmit Data Registers separately. 1 Transmit Data Registers are initialized from the FIFO once the Transmit FIFO is enabled.
18–16 TWA	Transmit Word Alignment. Configures the alignment of the data written into the ESAI Transmit Data Register and then passed to the relevant 24 bit Transmit shift register. **NOTE:** The settings shown below assume TCR[TWA]=0, TCR[PADC]=1 and TCR[TSHFD]=0. 000 MSB of data is bit 31. Data bits 7-0 are ignored when passed to transmit shift register. 001 MSB of data is bit 27. Data bits 3-0 are ignored when passed to transmit shift register.

Table continues on the next page...

ESAI_TFCR field descriptions (continued)

Field	Description
	010 MSB of data is bit 23. 011 MSB of data is bit 19. Bottom 4 bits of transmit shift register are zeroed. 100 MSB of data is bit 15. Bottom 8 bits of transmit shift register are zeroed. 101 MSB of data is bit 11. Bottom 12 bits of transmit shift register are zeroed. 110 MSB of data is bit 7. Bottom 16 bits of transmit shift register are zeroed. 111 MSB of data is bit 3. Bottom 20 bits of transmit shift register are zeroed.
15–8 TFWM	Transmit FIFO Watermark. These bits configure the threshold at which the Transmit FIFO Empty flag will set. The TFE is set when the number of empty slots in the Transmit FIFO equal or exceed the selected threshold.
7 TE5	Transmitter #5 FIFO Enable. This bit enables transmitter #5 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #5 is not using the Transmit FIFO. 1 Transmitter #5 is using the Transmit FIFO.
6 TE4	Transmitter #4 FIFO Enable. This bit enables transmitter #4 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #4 is not using the Transmit FIFO. 1 Transmitter #4 is using the Transmit FIFO.
5 TE3	Transmitter #3 FIFO Enable. This bit enables transmitter #3 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #3 is not using the Transmit FIFO. 1 Transmitter #3 is using the Transmit FIFO.
4 TE2	Transmitter #2 FIFO Enable. This bit enables transmitter #2 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #2 is not using the Transmit FIFO. 1 Transmitter #2 is using the Transmit FIFO.
3 TE1	Transmitter #1 FIFO Enable. This bit enables transmitter #1 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #1 is not using the Transmit FIFO. 1 Transmitter #1 is using the Transmit FIFO.
2 TE0	Transmitter #0 FIFO Enable. This bit enables transmitter #0 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #0 is not using the Transmit FIFO. 1 Transmitter #0 is using the Transmit FIFO.
1 TFR	Transmit FIFO Reset. This bit resets the Transmit FIFO pointers. 0 Transmit FIFO not reset. 1 Transmit FIFO reset.
0 TFE	Transmit FIFO Enable. This bit enables the use of the Transmit FIFO. 0 Transmit FIFO disabled. 1 Transmit FIFO enabled.

26.6.6 Transmit FIFO Status Register (ESAI_TFSR)

Address: 202_4000h base + 14h offset = 202_4014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

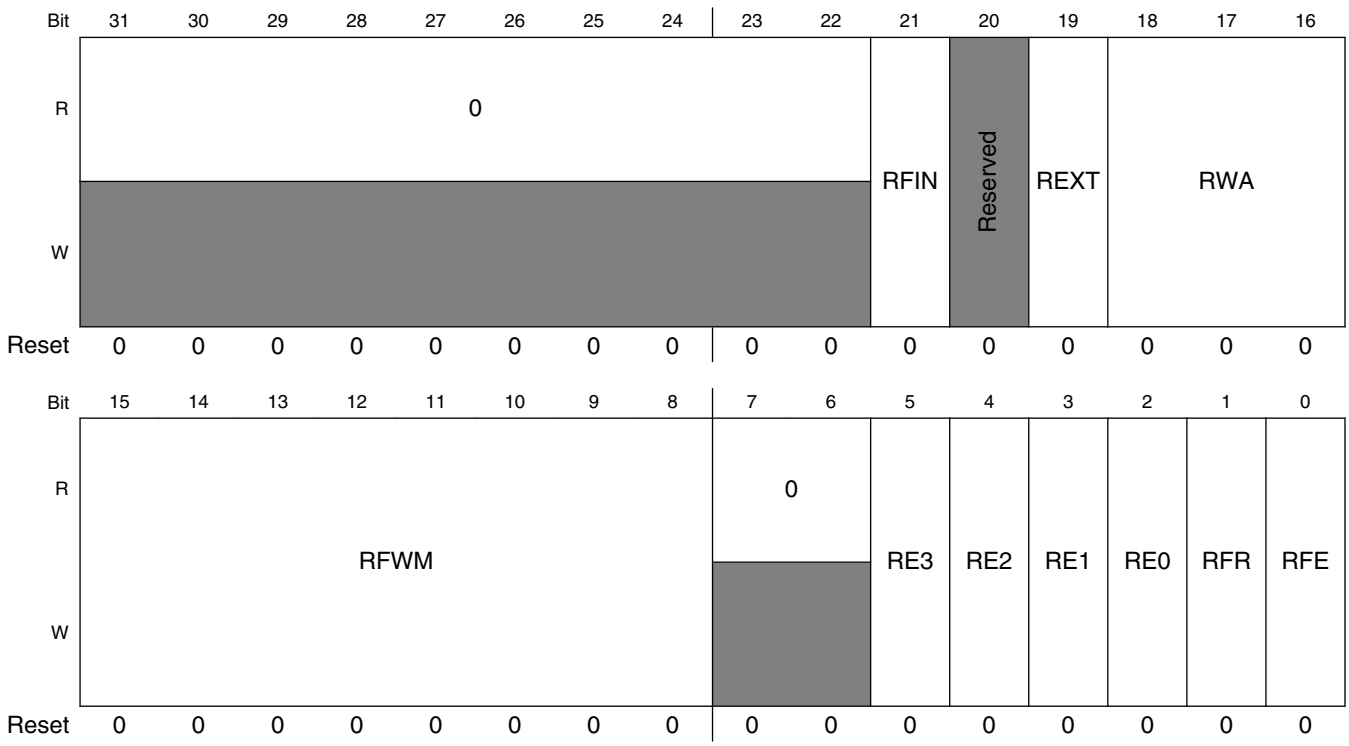
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	NTFO			0	NTFI			TFCNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_TFSR field descriptions

Field	Description
31–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–12 NTFO	Next Transmitter FIFO Out. Indicates which Transmit Data Register receives the top word of the Transmit FIFO. This will usually equal the lowest enabled transmitter, unless the transmit FIFO is empty. 000 Transmitter #0 receives next word from the Transmit FIFO. 001 Transmitter #1 receives next word from the Transmit FIFO. 010 Transmitter #2 receives next word from the Transmit FIFO. 011 Transmitter #3 receives next word from the Transmit FIFO. 100 Transmitter #4 receives next word from the Transmit FIFO. 101 Transmitter #5 receives next word from the Transmit FIFO. 110 Reserved. 111 Reserved.
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 NTFI	Next Transmitter FIFO In. Indicates which transmitter receives the next word written to the FIFO. 000 Transmitter #0 receives next word written to the Transmit FIFO. 001 Transmitter #1 receives next word written to the Transmit FIFO. 010 Transmitter #2 receives next word written to the Transmit FIFO. 011 Transmitter #3 receives next word written to the Transmit FIFO. 100 Transmitter #4 receives next word written to the Transmit FIFO. 101 Transmitter #5 receives next word written to the Transmit FIFO. 110 Reserved. 111 Reserved.
TFCNT	Transmit FIFO Counter. These bits indicate the number of data words stored in the Transmit FIFO.

26.6.7 Receive FIFO Configuration Register (ESAI_RFCR)

Address: 202_4000h base + 18h offset = 202_4018h



ESAI_RFCR field descriptions

Field	Description
31–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 RFIN	Rx FIFO Interrupt Enable Generate an interrupt if the "Rx FIFO Empty" flag is set. Indicate the number of data in the Rx FIFO has met or exceed water mark.
20 Reserved	This field is reserved. Reserved
19 REXT	Receive Extension. Enables the receive data to be returned sign extended when the Receive Word Alignment is configured to return data where the MSB is not aligned with bit 31. 0 Receive data is zero extended. 1 Receive data is sign extended.
18–16 RWA	Receive Word Alignment. Configures the alignment of the data passed from the relevant 24 bit Receive shift register and read out the ESAI Receive Data Register. 000 MSB of data is at bit 31. Data bits 7-0 are zeroed. 001 MSB of data is at bit 27. Data bits 3-0 are zeroed. 010 MSB of data is at bit 23. 011 MSB of data is at bit 19. Data bits 3-0 from receive shift register are ignored.

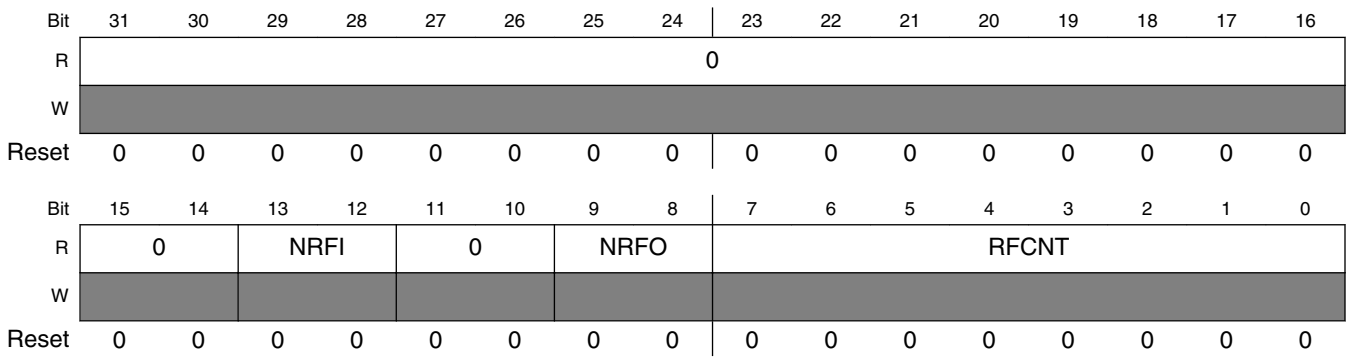
Table continues on the next page...

ESAI_RFCR field descriptions (continued)

Field	Description
	100 MSB of data is at bit 15. Data bits 7-0 from receive shift register are ignored. 101 MSB of data is at bit 11. Data bits 11-0 from receive shift register are ignored. 110 MSB of data is at bit 7. Data bits 15-0 from receive shift register are ignored. 111 MSB of data is at bit 3. Data bits 19-0 from receive shift register are ignored.
15–8 RFWM	Receive FIFO Watermark. These bits configure the threshold at which the Receive FIFO Full flag will set. The RFF is set when the number of words in the Receive FIFO equal or exceed the selected threshold. It can be set to a non-zero value.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 RE3	Receiver #3 FIFO Enable. This bit enables receiver #3 to use the Receive FIFO. Do not change this bit when the Receiver FIFO is enabled. 0 Receiver #3 is not using the Receive FIFO. 1 Receiver #3 is using the Receive FIFO.
4 RE2	Receiver #2 FIFO Enable. This bit enables receiver #2 to use the Receive FIFO. Do not change this bit when the Receiver FIFO is enabled. 0 Receiver #2 is not using the Receive FIFO. 1 Receiver #2 is using the Receive FIFO.
3 RE1	Receiver #1 FIFO Enable. This bit enables receiver #1 to use the Receive FIFO. Do not change this bit when the Receiver FIFO is enabled. 0 Receiver #1 is not using the Receive FIFO. 1 Receiver #1 is using the Receive FIFO.
2 RE0	Receiver #0 FIFO Enable. This bit enables receiver #0 to use the Receive FIFO. Do not change this bit when the Receiver FIFO is enabled. 0 Receiver #0 is not using the Receive FIFO. 1 Receiver #0 is using the Receive FIFO.
1 RFR	Receive FIFO Reset. This bit resets the Receive FIFO pointers. 0 Receive FIFO not reset. 1 Receive FIFO reset.
0 RFE	Receive FIFO Enable. This bit enables the use of the Receive FIFO. 0 Receive FIFO disabled. 1 Receive FIFO enabled.

26.6.8 Receive FIFO Status Register (ESAI_RFSR)

Address: 202_4000h base + 1Ch offset = 202_401Ch



ESAI_RFSR field descriptions

Field	Description
31–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–12 NRFI	Next Receiver FIFO In. Indicates which Receiver Data Register the Receive FIFO will load next. This will usually equal the lowest enabled receiver, unless the receive FIFO is full. 00 Receiver #0 returns next word to the Receive FIFO. 01 Receiver #1 returns next word to the Receive FIFO. 10 Receiver #2 returns next word to the Receive FIFO. 11 Receiver #3 returns next word to the Receive FIFO.
11–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 NRFO	Next Receiver FIFO Out. Indicates which receiver returns the top word of the Receive FIFO. 00 Receiver #0 returns next word from the Receive FIFO. 01 Receiver #1 returns next word from the Receive FIFO. 10 Receiver #2 returns next word from the Receive FIFO. 11 Receiver #3 returns next word from the Receive FIFO.
RFCNT	Receive FIFO Counter. These bits indicate the number of data words stored in the Receive FIFO.

26.6.9 Transmit Data Register n (ESAI_TX*n*)

ESAI_TX5, ESAI_TX4, ESAI_TX3, ESAI_TX2, ESAI_TX1 and ESAI_TX0 are 32-bit write-only registers. Data to be transmitted is written into these registers and is automatically transferred to the transmit shift registers (Figure 26-2 and Figure 26-3). The data written (8, 12, 16, 20, or 24 bits) should occupy the most significant portion of the TXn according to the ALC control bit setting. The unused bits (least significant portion and the 8 most significant bits when ALC=1) of the TXn are don't care bits. The Core is interrupted whenever the TXn becomes empty if the transmit data register empty interrupt has been enabled.

Address: 202_4000h base + 80h offset + (4d × i), where i=0d to 5d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0																							
W									TXn																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_TX*n* field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXn	Stores the data to be transmitted and is automatically transferred to the transmit shift registers. See ESAI Transmit Shift Registers.

26.6.10 ESAI Transmit Slot Register (ESAI_TSR)

Address: 202_4000h base + 98h offset = 202_4098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0																							
W									TSR																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_TSR field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ESAI_TSR field descriptions (continued)

Field	Description
TSR	The write-only Transmit Slot Register (ESAI_TSR) is effectively a null data register that is used when the data is not to be transmitted in the available transmit time slot. The transmit data pins of all the enabled transmitters are in the high-impedance state for the respective time slot where TSR has been written. The Transmitter External Buffer Enable pin (FSR pin when SYN=1, TEBE=1, RFSD=1) disables the external buffers during the slot when the ESAI_TSR register has been written.

26.6.11 Receive Data Register n (ESAI_RX*n*)

ESAI_RX3, ESAI_RX2, ESAI_RX1, and ESAI_RX0 are 32-bit read-only registers that accept data from the receive shift registers when they become full (Figure 26-2 and Figure 26-3). The data occupies the most significant portion of the receive data registers, according to the ALC control bit setting. The unused bits (least significant portion and 8 most significant bits when ALC=1) read as zeros. The Core is interrupted whenever RXn becomes full if the associated interrupt is enabled.

Address: 202_4000h base + A0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RXn																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

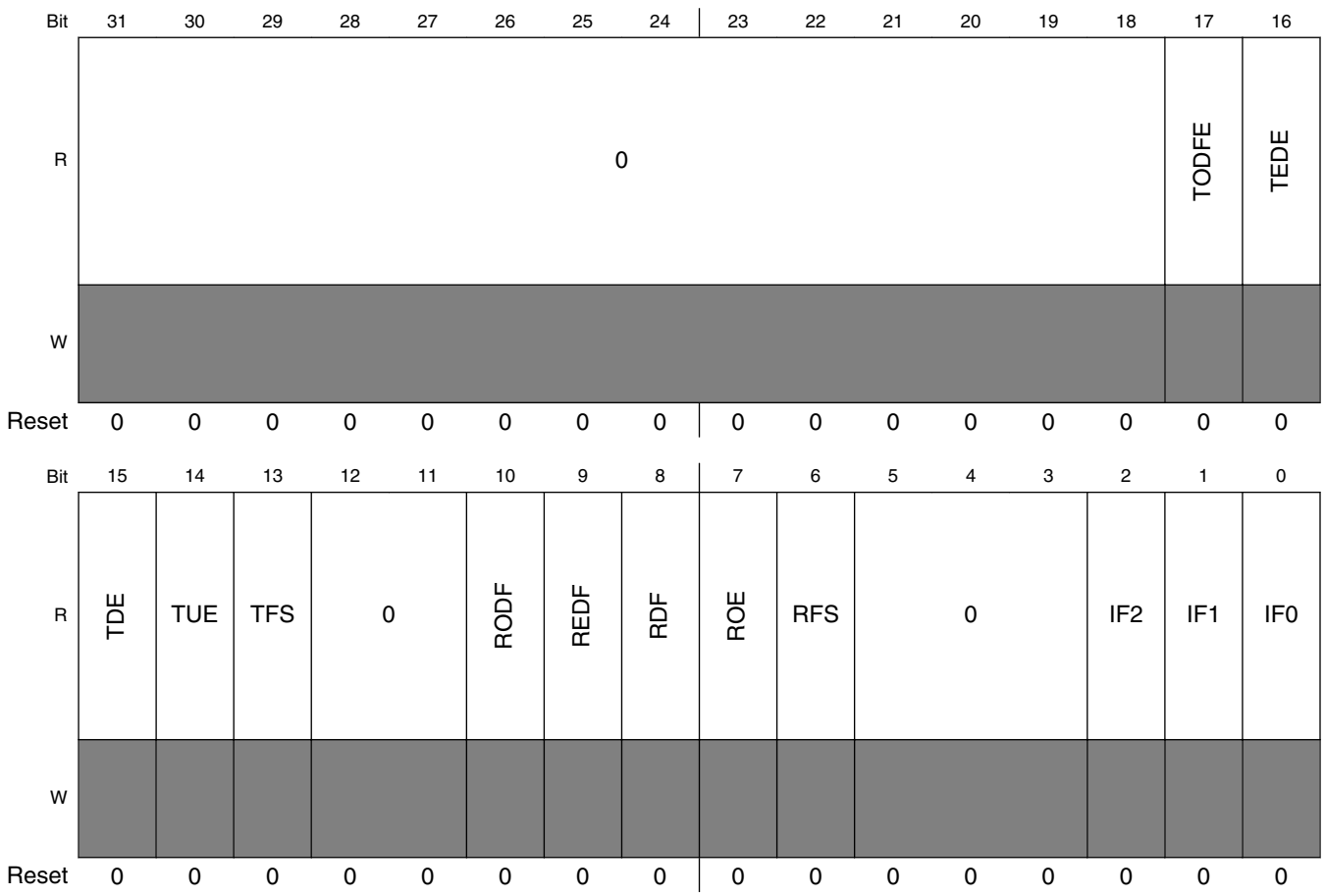
ESAI_RX*n* field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RXn	Accept data from the receive shift registers when they become full See ESAI Receive Shift Registers

26.6.12 Serial Audio Interface Status Register (ESAI_SAISR)

The Status Register (ESAI_SAISR) is a read-only status register used by the ARM Core to read the status and serial input flags of the ESAI.

Address: 202_4000h base + CCh offset = 202_40CCh



ESAI_SAISR field descriptions

Field	Description
31–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 TODFE	ESAI_SAISR Transmit Odd-Data Register Empty. When set, TODFE indicates that the enabled transmitter data registers became empty at the beginning of an odd time slot. Odd time slots are all odd-numbered slots (1, 3, 5, and so on). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. This flag is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TODFE indicates that data should be written to all the TX registers of the enabled transmitters or to the transmit slot register (ESAI_TSR). TODE is cleared when the Core writes to all the transmit data registers of the

Table continues on the next page...

ESAI_SAISR field descriptions (continued)

Field	Description
	enabled transmitters, or when the Core writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TODFE is set. Hardware, software, ESAI individual reset clear TODFE.
16 TEDE	ESAI_SAISR Transmit Even-DataRegister Empty. When set, TEDE indicates that the enabled transmitter data registers became empty at the beginning of an even time slot. Even time slots are all even-numbered slots (0, 2, 4, 6, etc.). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. The zero time slot is considered even. This flag is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TEDE indicates that data should be written to all the TX registers of the enabled transmitters or to the transmit slot register (ESAI_TSR). TEDE is cleared when the Core writes to all the transmit data registers of the enabled transmitters, or when the Core writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TEDE is set. Hardware, software, ESAI individual reset clear TEDE.
15 TDE	ESAI_SAISR Transmit Data Register Empty. TDE is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TDE indicates that data should be written to all the TX registers of the enabled transmitters or to the transmit slot register (ESAI_TSR). TDE is cleared when the Core writes to all the transmit data registers of the enabled transmitters, or when the Core writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TDE is set. Hardware, software, ESAI individual reset clear TDE.
14 TUE	ESAI_SAISR Transmit Underrun Error Flag. TUE is set when at least one of the enabled serial transmit shift registers is empty (no new data to be transmitted) and a transmit time slot occurs. When a transmit underrun error occurs, the previous data (which is still present in the TX registers that were not written) is retransmitted. If TEIE is set, an ESAI transmit data with exception (underrun error) interrupt request is issued when TUE is set. Hardware, software, ESAI individual reset clear TUE. TUE is also cleared by reading the ESAI_SAISR with TUE set, followed by writing to all the enabled transmit data registers or to ESAI_TSR.
13 TFS	ESAI_SAISR Transmit Frame Sync Flag. When set, TFS indicates that a transmit frame sync occurred in the current time slot. TFS is set at the start of the first time slot in the frame and cleared during all other time slots. Data written to a transmit data register during the time slot when TFS is set is transmitted (in network mode), if the transmitter is enabled, during the second time slot in the frame. TFS is useful in network mode to identify the start of a frame. TFS is cleared by hardware, software, ESAI individual reset. TFS is valid only if at least one transmitter is enabled, that is, one or more of TE0, TE1, TE2, TE3, TE4 and TE5 are set. (In normal mode, TFS always reads as a one when transmitting data because there is only one time slot per frame - the "frame sync" time slot)
12–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 RODF	ESAI_SAISR Receive Odd-Data Register Full. When set, RODF indicates that the received data in the receive data registers of the enabled receivers have arrived during an odd time slot when operating in the network mode. Odd time slots are all odd-numbered slots (1, 3, 5, and so on). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. RODF is set when the contents of the receive shift registers are transferred to the receive data registers. RODF is cleared when the Core reads all the enabled receive data registers or cleared by hardware, software, ESAI individual resets.
9 REDF	ESAI_SAISR Receive Even-Data Register Full. When set, REDF indicates that the received data in the receive data registers of the enabled receivers have arrived during an even time slot when operating in the network mode. Even time slots are all even-numbered slots (0, 2, 4, 6, and so on). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. The zero time slot is considered even. REDF is set when the contents of the receive shift registers are transferred to the receive data registers. REDF is cleared when the Core reads all the enabled receive data registers or

Table continues on the next page...

ESAI_SAISR field descriptions (continued)

Field	Description
	cleared by hardware, software, ESAI individual resets. If REDIE is set, an ESAI receive even slot data interrupt request is issued when REDF is set.
8 RDF	ESAI_SAISR Receive Data Register Full. RDF is set when the contents of the receive shift register of an enabled receiver is transferred to the respective receive data register. RDF is cleared when the Core reads the receive data register of all enabled receivers or cleared by hardware, software, ESAI individual reset. If RIE is set, an ESAI receive data interrupt request is issued when RDF is set.
7 ROE	ESAI_SAISR Receive Overrun Error Flag. The ROE flag is set when the serial receive shift register of an enabled receiver is full and ready to transfer to its receiver data register (RXn) and the register is already full (RDF=1). If REIE is set, an ESAI receive data with exception (overrun error) interrupt request is issued when ROE is set. Hardware, software, ESAI individual reset clear ROE. ROE is also cleared by reading the SAISR with ROE set, followed by reading all the enabled receive data registers.
6 RFS	ESAI_SAISR Receive Frame Sync Flag. When set, RFS indicates that a receive frame sync occurred during reception of the words in the receiver data registers. This indicates that the data words are from the first slot in the frame. When RFS is clear and a word is received, it indicates (only in the network mode) that the frame sync did not occur during reception of that word. RFS is cleared by hardware, software, ESAI individual reset. RFS is valid only if at least one of the receivers is enabled (REx=1). (In normal mode, RFS always reads as a one when reading data because there is only one time slot per frame - the "frame sync" time slot)
5–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 IF2	ESAI_SAISR Serial Input Flag 2. The IF2 bit is enabled only when the HCKR pin is defined as ESAI in the Port Control Register, SYN=1 and RHCKD=0, indicating that HCKR is an input flag and the synchronous mode is selected. Data present on the HCKR pin is latched during reception of the first received data bit after frame sync is detected. The IF2 bit is updated with this data when the receive shift registers are transferred into the receiver data registers. IF2 reads as a zero when it is not enabled. Hardware, software, ESAI individual reset clear IF2.
1 IF1	ESAI_SAISR Serial Inout Flag 1. The IF1 bit is enabled only when the FSR pin is defined as ESAI in the Port Control Register, SYN =1, RFSD=0 and TEBE=0, indicating that FSR is an input flag and the synchronous mode is selected. Data present on the FSR pin is latched during reception of the first received data bit after frame sync is detected. The IF1 bit is updated with this data when the receiver shift registers are transferred into the receiver data registers. IF1 reads as a zero when it is not enabled. Hardware, software, ESAI individual reset clear IF1.
0 IF0	ESAI_SAISR Serial Input Flag 0. The IF0 bit is enabled only when the SCKR pin is defined as ESAI in the Port Control Register, SYN=1 and RCKD=0, indicating that SCKR is an input flag and the synchronous mode is selected. Data present on the SCKR pin is latched during reception of the first received data bit after frame sync is detected. The IF0 bit is updated with this data when the receiver shift registers are transferred into the receiver data registers. IF0 reads as a zero when it is not enabled. Hardware, software, ESAI individual reset clear IF0.

26.6.13 Serial Audio Interface Control Register (ESAI_SAICR)

The read/write Common Control Register (ESAI_SAICR) contains control bits for functions that affect both the receive and transmit sections of the ESAI.

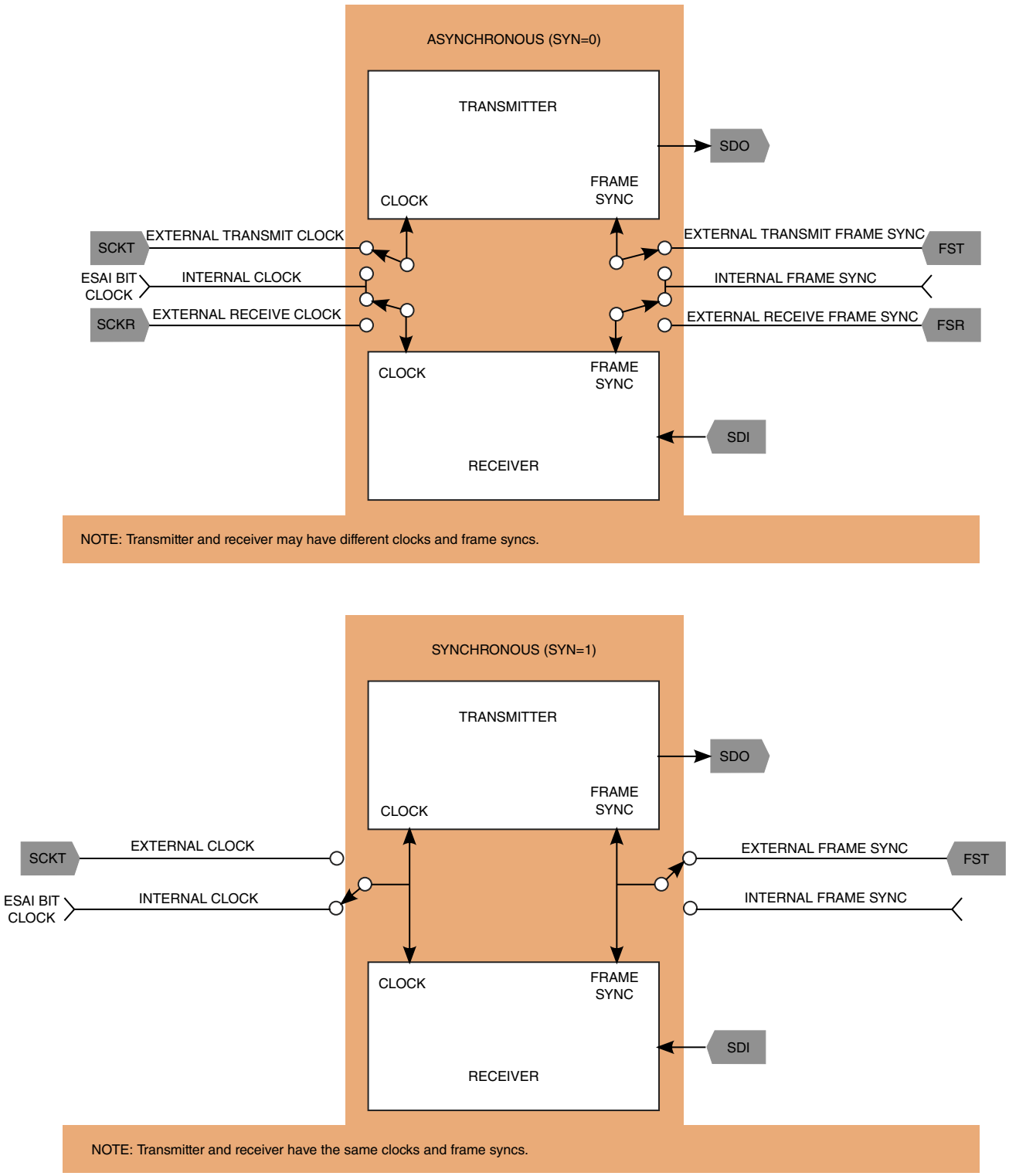


Figure 26-4. SAICR SYN Bit Operation

Address: 202_4000h base + D0h offset = 202_40D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							ALC	TEBE	SYN	0			OF2	OF1	OF0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_SAICR field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 ALC	ESAI_SAICR Alignment Control. The ESAI is designed for 24-bit fractional data, thus shorter data words are left aligned to the MSB (bit 23). Some applications use 16-bit fractional data. In those cases, shorter data words may be left aligned to bit 15. The Alignment Control (ALC) bit supports these applications. If ALC is set, transmitted and received words are left aligned to bit 15 in the transmit and receive shift registers. If ALC is cleared, transmitted and received word are left aligned to bit 23 in the transmit and receive shift registers. While ALC is set, 20-bit and 24-bit words may not be used, and word length control should specify 8-, 12-, or 16-bit words; otherwise, results are unpredictable.
7 TEBE	ESAI_SAICR Transmit External Buffer Enable. The Transmitter External Buffer Enable (TEBE) bit controls the function of the FSR pin when in the synchronous mode. If the ESAI is configured for operation in the synchronous mode (SYN=1), and TEBE is set while FSR pin is configured as an output (RFSD=1), the FSR pin functions as the transmitter external buffer enable control to enable the use of an external buffers on the transmitter outputs. If TEBE is cleared, the FSR pin functions as the serial I/O flag 1. See Port C Control Register for a summary of the effects of TEBE on the FSR pin.
6 SYN	ESAI_SAICR Synchronous Mode Selection. The Synchronous Mode Selection (SYN) bit controls whether the receiver and transmitter sections of the ESAI operate synchronously or asynchronously with respect to each other (see Port C Control Register). When SYN is cleared, the asynchronous mode is chosen and independent clock and frame sync signals are used for the transmit and receive sections. When SYN is set, the synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals. When in the synchronous mode (SYN=1), the transmit and receive sections use the transmitter section clock generator as the source of the clock and frame sync for both sections. Also, the receiver clock pins SCKR, FSR and HCKR now operate as I/O flags. Refer to Table 26-11, Table 26-12, and Table 26-13 for the effects of SYN on the receiver clock pins.
5–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 OF2	ESAI_SAICR Serial Output Flag 2. The Serial Output Flag 2 (OF2) is a data bit used to hold data to be send to the OF2 pin. When the ESAI is in the synchronous clock mode (SYN=1), the HCKR pin is configured as the ESAI flag 2. If the receiver high frequency clock direction bit (RHCKD) is set, the HCKR pin is the output flag OF2, and data present in the OF2 bit is written to the OF2 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.
1 OF1	ESAI_SAICR Serial Output Flag 1. The Serial Output Flag 1 (OF1) is a data bit used to hold data to be send to the OF1 pin. When the ESAI is in the synchronous clock mode (SYN=1), the FSR pin is configured as the ESAI flag 1. If the receiver frame sync direction bit (RFSD) is set and the TEBE bit is cleared, the FSR pin is the output flag OF1, and data present in the OF1 bit is written to the OF1 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.

Table continues on the next page...

ESAI_SAICR field descriptions (continued)

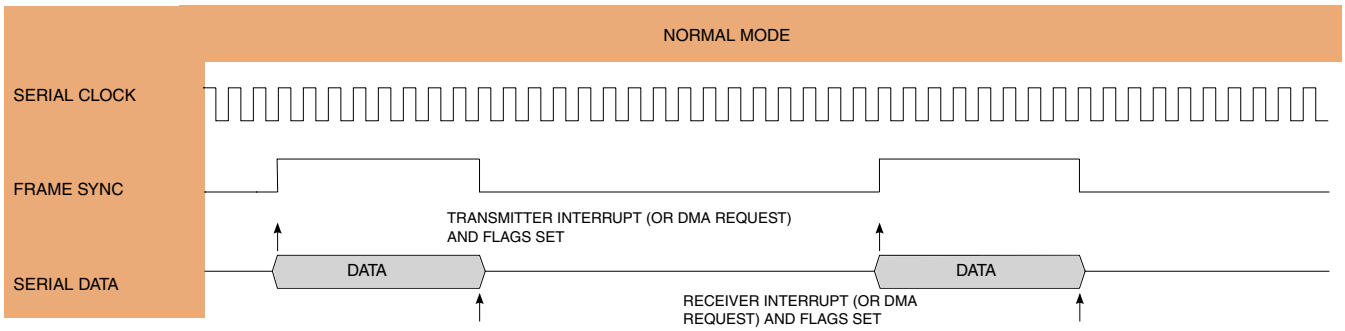
Field	Description
0 OF0	ESAI_SAICR Serial Output Flag 0. The Serial Output Flag 0 (OF0) is a data bit used to hold data to be send to the OF0 pin. When the ESAI is in the synchronous clock mode (SYN=1), the SCKR pin is configured as the ESAI flag 0. If the receiver serial clock direction bit (RCKD) is set, the SCKR pin is the output flag OF0, and data present in the OF0 bit is written to the OF0 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.

26.6.14 Transmit Control Register (ESAI_TCR)

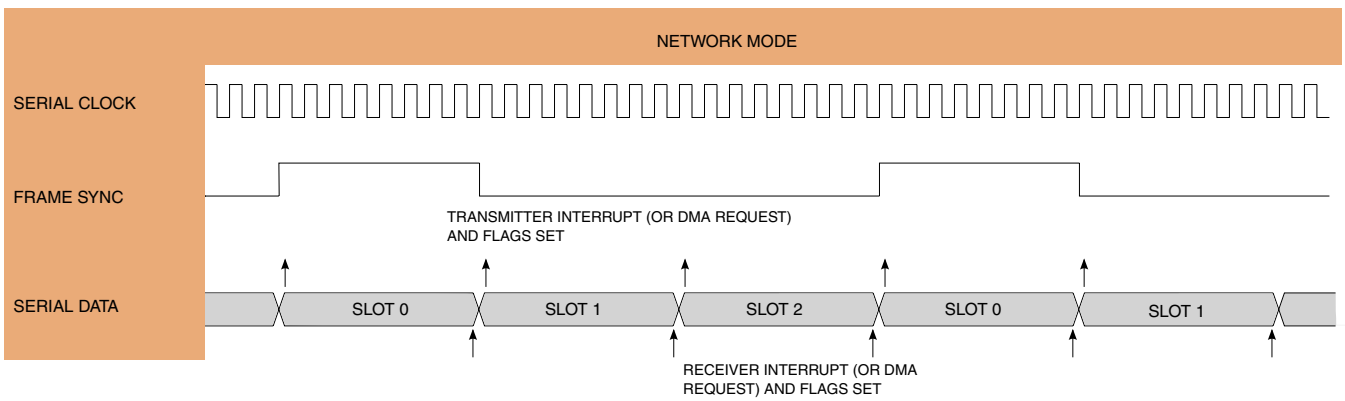
The read/write Transmit Control Register (ESAI_TCR) controls the ESAI transmitter section. Interrupt enable bits for the transmitter section are provided in this control register. Operating modes are also selected in this register.

Table 26-5. Transmit Network Mode Selection

TMOD1	TMOD0	TDC4-TDC0	Transmitter Network Mode
0	0	0x0-0x1F	Normal Mode
0	1	0x0	On-Demand Mode
0	1	0x1-0x1F	Network Mode
1	0	X	Reserved
1	1	0x0C	AC97



NOTE: Interrupts occur and data is transferred once per frame sync.



NOTE: Interrupts occur and a word may be transferred at every time slot.

Figure 26-5. Normal and Network Operation

Table 26-6. ESAI Transmit Slot and Word Length Selection

TSWS4	TSWS3	TSWS2	TSWS1	TSWS0	SLOT LENGTH	WORD LENGTH
0	0	0	0	0	8	8
0	0	1	0	0	12	8
0	0	0	0	1		12
0	1	0	0	0	16	8
0	0	1	0	1		12
0	0	0	1	0		16
0	1	1	0	0	20	8
0	1	0	0	1		12
0	0	1	1	0		16
0	0	0	1	1		20
1	0	0	0	0	24	8
0	1	1	0	1		12
0	1	0	1	0		16
0	0	1	1	1		20

Table continues on the next page...

Table 26-6. ESAI Transmit Slot and Word Length Selection (continued)

TSWS4	TSWS3	TSWS2	TSWS1	TSWS0	SLOT LENGTH	WORD LENGTH
1	1	1	1	0		24
1	1	0	0	0	32	8
1	0	1	0	1		12
1	0	0	1	0		16
0	1	1	1	1		20
1	1	1	1	1		24
0	1	0	1	1	Reserved	
0	1	1	1	0		
1	0	0	0	1		
1	0	0	1	1		
1	0	1	0	0		
1	0	1	1	0		
1	0	1	1	1		
1	1	0	0	1		
1	1	0	1	0		
1	1	0	1	1		
1	1	1	0	0		
1	1	1	0	1		

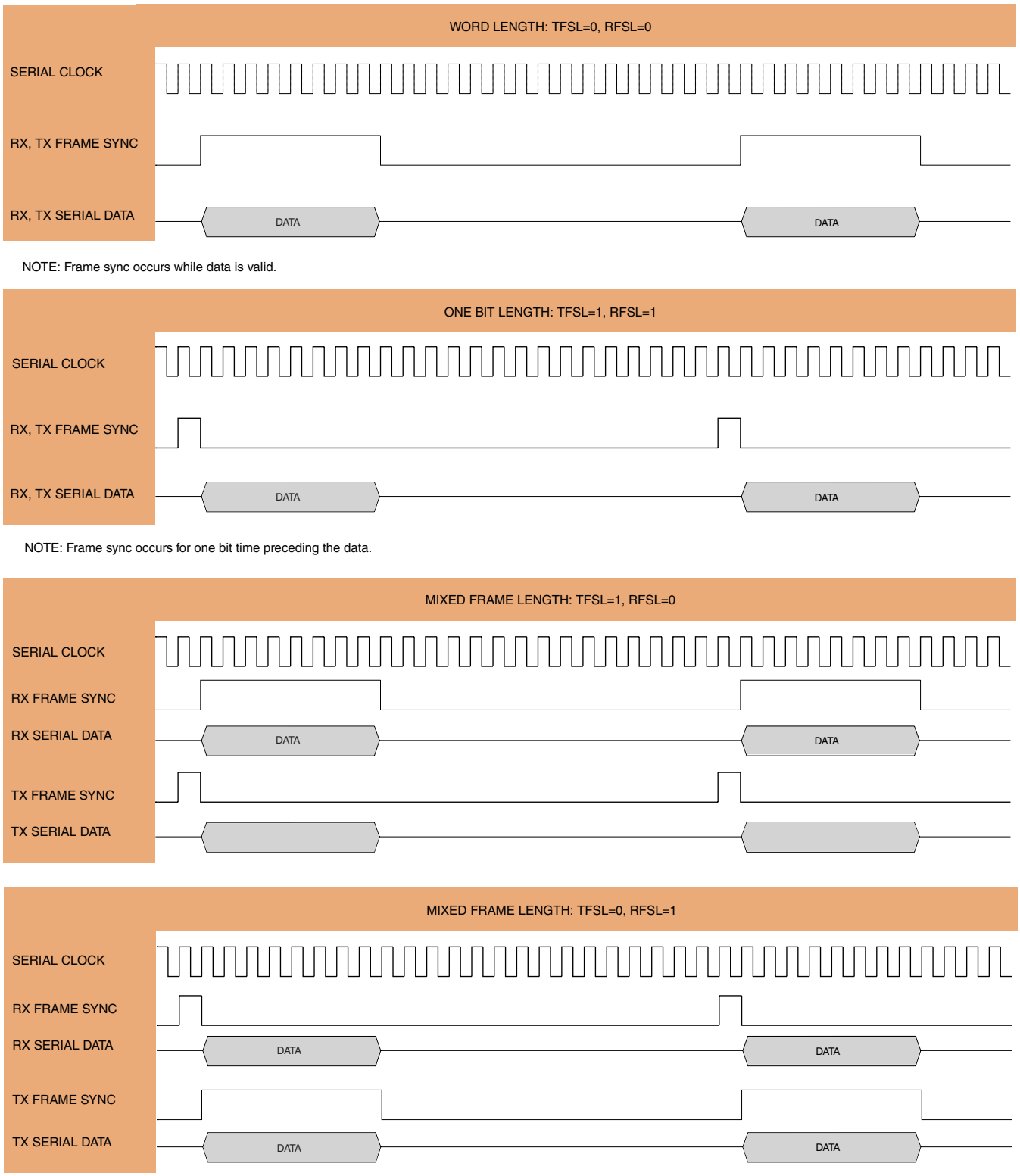
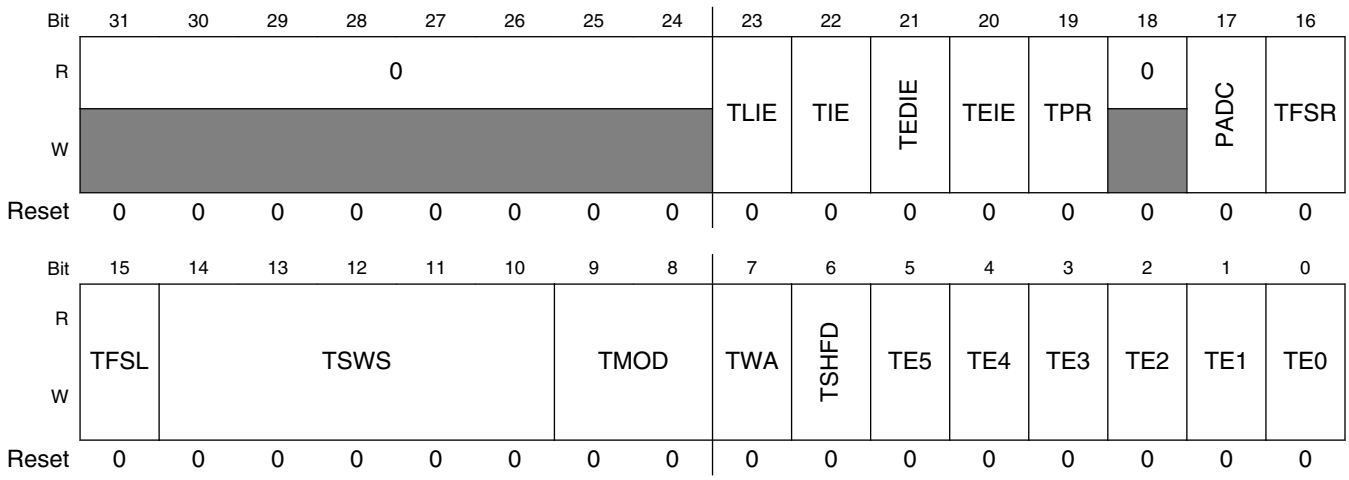


Figure 26-6. Frame Length Selection

Address: 202_4000h base + D4h offset = 202_40D4h



ESAI_TCR field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 TLIE	ESAI_TCR Transmit Last Slot Interrupt Enable. TLIE enables an interrupt at the beginning of last slot of a frame in network mode. When TLIE is set the Core is interrupted at the start of the last slot in a frame in network mode regardless of the transmit mask register setting. When TLIE is cleared the transmit last slot interrupt is disabled. TLIE is disabled when TDC=0x00000 (on-demand mode). The use of the transmit last slot interrupt is described in ESAI Interrupt Requests.
22 TIE	ESAI_TCR Transmit Interrupt Enable. The Core is interrupted when TIE and the TDE flag in the ESAI_SAISR status register are set. When TIE is cleared, this interrupt is disabled. Writing data to all the data registers of the enabled transmitters or to ESAI_TSR clears TDE, thus clearing the interrupt. Transmit interrupts with exception have higher priority than normal transmit data interrupts, therefore if exception occurs (TUE is set) and TEIE is set, the ESAI requests an ESAI transmit data with exception interrupt from the interrupt controller.
21 TEDIE	ESAI_TCR Transmit Even Slot Data Interrupt Enable. The TEDIE control bit is used to enable the transmit even slot data interrupts. If TEDIE is set, the transmit even slot data interrupts are enabled. If TEDIE is cleared, the transmit even slot data interrupts are disabled. A transmit even slot data interrupt request is generated if TEDIE is set and the TEDE status flag in the ESAI_SAISR status register is set. Even time slots are all even-numbered time slots (0, 2, 4, etc.) when operating in network mode. The zero time slot in the frame is marked by the frame sync signal and is considered to be even. Writing data to all the data registers of the enabled transmitters or to ESAI_TSR clears the TEDE flag, thus servicing the interrupt. Transmit interrupts with exception have higher priority than transmit even slot data interrupts, therefore if exception occurs (TUE is set) and TEIE is set, the ESAI requests an ESAI transmit data with exception interrupt from the interrupt controller.
20 TEIE	ESAI_TCR Transmit Exception Interrupt Enable. When TEIE is set, the Core is interrupted when both TDE and TUE in the ESAI_SAISR status register are set. When TEIE is cleared, this interrupt is disabled. Reading the ESAI_SAISR status register followed by writing to all the data registers of the enabled transmitters clears TUE, thus clearing the pending interrupt.
19 TPR	ESAI_TCR Transmit Section Personal Reset. The TPR control bit is used to put the transmitter section of the ESAI in the personal reset state. The receiver section is not affected. When TPR is cleared, the transmitter section may operate normally. When TPR is set, the transmitter section enters the personal reset state immediately. When in the personal reset state, the status bits are reset to the same state as after hardware reset. The control bits are not affected by the personal reset state. The transmitter data pins are tri-stated while in the personal reset state; if a stable logic level is desired, the transmitter data pins should be defined as GPIO outputs, or external pull-up or pull-down resistors should be used. The

Table continues on the next page...

ESAI_TCR field descriptions (continued)

Field	Description
	transmitter clock outputs drive zeroes while in the personal reset state. Note that to leave the personal reset state by clearing TPR, the procedure described in ESAI Initialization Examples should be followed.
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 PADC	ESAI_TCR Transmit Zero Padding Control. When PADC is cleared, zero padding is disabled. When PADC is set, zero padding is enabled. PADC, in conjunction with the TCR[TWA] control bit, determines the way that padding is done for operating modes where the word length is less than the slot length. See the TCR[TWA] bit description in bit 7 for more details. Because the data word is shorter than the slot length, the data word is extended until achieving the slot length, according to the following rule: If the data word is left-aligned (TCR[TWA]=0), and zero padding is disabled (PADC=0), the last data bit is repeated after the data word has been transmitted. If zero padding is enabled (PADC=1), zeroes are transmitted after the data word has been transmitted. If the data word is right-aligned (TCR[TWA]=1), and zero padding is disabled (PADC=0), the first data bit is repeated before the transmission of the data word. If zero padding is enabled (PADC=1), zeroes are transmitted before the transmission of the data word.
16 TFSR	ESAI_TCR Transmit Frame Sync Relative Timing. TFSR determines the relative timing of the transmit frame sync signal as referred to the serial data lines, for a word length frame sync only (TFSL=0). When TFSR is cleared the word length frame sync occurs together with the first bit of the data word of the first slot. When TFSR is set the word length frame sync starts one serial clock cycle earlier, that is, together with the last bit of the previous data word.
15 TFSL	ESAI_TCR Transmit Frame Sync Length. The TFSL bit selects the length of frame sync to be generated or recognized. If TFSL is cleared, a word-length frame sync is selected. If TFSL is set, a 1-bit clock period frame sync is selected. See Figure 1-21 for examples of frame length selection.
14–10 TSWS	ESAI_TCR Tx Slot and Word Length Select (TSWS4-TSWS0). The TSWS4-TSWS0 bits are used to select the length of the slot and the length of the data words being transferred through the ESAI. The word length must be equal to or shorter than the slot length. The possible combinations are shown in Table 26-6. See also the ESAI data path programming model in Figure 26-2 and Figure 26-3.
9–8 TMOD	ESAI_TCR Transmit Network Mode Control (TMOD1-TMOD0). The TMOD1 and TMOD0 bits are used to define the network mode of ESAI transmitters, as shown in Table 26-6. In the normal mode, the frame rate divider determines the word transfer rate - one word is transferred per frame sync during the frame sync time slot, as shown in Figure 26-5. In network mode, it is possible to transfer a word for every time slot, as shown in Figure 26-5. For further details, refer to Modes of Operation In order to comply with AC-97 specifications, TSWS4-TSWS0 should be set to 00011 (20-bit slot, 20-bit word length), TFSL and TFSR should be cleared, and TDC4-TDC0 should be set to 0x0C (13 words in frame). If TMOD=0b11 and the above recommendations are followed, the first slot and word will be 16 bits long, and the next 12 slots and words will be 20 bits long, as required by the AC97 protocol.
7 TWA	ESAI_TCR Transmit Word Alignment Control. The Transmitter Word Alignment Control (TCR[TWA]) bit defines the alignment of the data word in relation to the slot. This is relevant for the cases where the word length is shorter than the slot length. If TCR[TWA] is cleared, the data word is left-aligned in the slot frame during transmission. If TCR[TWA] is set, the data word is right-aligned in the slot frame during transmission. Because the data word is shorter than the slot length, the data word is extended until achieving the slot length, according to the following rule: If the data word is left-aligned (TCR[TWA]=0), and zero padding is disabled (PADC=0), the last data bit is repeated after the data word has been transmitted. If zero padding is enabled (PADC=1), zeroes are transmitted after the data word has been transmitted.

Table continues on the next page...

ESAI_TCR field descriptions (continued)

Field	Description
	If the data word is right-aligned (TCR[TWA]=1), and zero padding is disabled (PADC=0), the first data bit is repeated before the transmission of the data word. If zero padding is enabled (PADC=1), zeroes are transmitted before the transmission of the data word.
6 TSHFD	ESAI_TCR Transmit Shift Direction. The TSHFD bit causes the transmit shift registers to shift data out MSB first when TSHFD equals zero or LSB first when TSHFD equals one (see Figure 26-2 and Figure 26-3 .
5 TE5	ESAI_TCR ESAI Transmit 5 Enable. TE5 enables the transfer of data from ESAI_TX5 to the transmit shift register #5. When TE5 is set and a frame sync is detected, the transmit #5 portion of the ESAI is enabled for that frame. When TE5 is cleared, the transmitter #5 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to ESAI_TX5 when TE5 is cleared but the data is not transferred to the transmit shift register #5. The SDO5/SDI0 pin is the data input pin for ESAI_RX0 if TE5 is cleared and RE0 in the ESAI_RCR register is set. If both RE0 and TE5 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE0 and TE5 should not be set at the same time. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE5 and setting it again disables the transmitter #5 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO5/SDI0 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE5 can be left enabled.
4 TE4	ESAI_TCR ESAI Transmit 4 Enable. TE4 enables the transfer of data from ESAI_TX4 to the transmit shift register #4. When TE4 is set and a frame sync is detected, the transmit #4 portion of the ESAI is enabled for that frame. When TE4 is cleared, the transmitter #4 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to ESAI_TX4 when TE4 is cleared but the data is not transferred to the transmit shift register #4. The SDO4/SDI1 pin is the data input pin for ESAI_RX1 if TE4 is cleared and RE1 in the RCR register is set. If both RE1 and TE4 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE1 and TE4 should not be set at the same time. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE4 and setting it again disables the transmitter #4 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO4/SDI1 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE4 can be left enabled.
3 TE3	ESAI_TCR ESAI Transmit 3 Enable. TE3 enables the transfer of data from ESAI_TX3 to the transmit shift register #3. When TE3 is set and a frame sync is detected, the transmit #3 portion of the ESAI is enabled for that frame. When TE3 is cleared, the transmitter #3 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to ESAI_TX3 when TE3 is cleared but the data is not transferred to the transmit shift register #3. The SDO3/SDI2 pin is the data input pin for ESAI_RX2 if TE3 is cleared and RE2 in the ESAI_RCR register is set. If both RE2 and TE3 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE2 and TE3 should not be set at the same time. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE3 and setting it again disables the transmitter #3 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO3/SDI2 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE3 can be left enabled.

Table continues on the next page...

ESAI_TCR field descriptions (continued)

Field	Description
2 TE2	ESAI_TCR ESAI Transmit 2 Enable. TE2 enables the transfer of data from ESAI_TX2 to the transmit shift register #2. When TE2 is set and a frame sync is detected, the transmit #2 portion of the ESAI is enabled for that frame. When TE2 is cleared, the transmitter #2 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to ESAI_TX2 when TE2 is cleared but the data is not transferred to the transmit shift register #2. The SDO2/SDI3 pin is the data input pin for ESAI_RX3 if TE2 is cleared and RE3 in the ESAI_RCR register is set. If both RE3 and TE2 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE3 and TE2 should not be set at the same time. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE2 and setting it again disables the transmitter #2 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO2/SDI3 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE2 can be left enabled.
1 TE1	ESAI_TCR ESAI Transmit 1 Enable. TE1 enables the transfer of data from TX1 to the transmit shift register #1. When TE1 is set and a frame sync is detected, the transmit #1 portion of the ESAI is enabled for that frame. When TE1 is cleared, the transmitter #1 is disabled after completing transmission of data currently in the ESAI transmit shift register. The SDO1 output is tri-stated, and any data present in TX1 is not transmitted, that is, data can be written to TX1 with TE1 cleared, but data is not transferred to the transmit shift register #1. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE1 and setting it again disables the transmitter #1 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO1 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE1 can be left enabled.
0 TE0	ESAI_TCR ESAI Transmit 0 Enable. TE0 enables the transfer of data from ESAI_TX0 to the transmit shift register #0. When TE0 is set and a frame sync is detected, the transmit #0 portion of the ESAI is enabled for that frame. When TE0 is cleared, the transmitter #0 is disabled after completing transmission of data currently in the ESAI transmit shift register. The SDO0 output is tri-stated, and any data present in ESAI_TX0 is not transmitted, that is, data can be written to ESAI_TX0 with TE0 cleared, but data is not transferred to the transmit shift register #0. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE0 and setting it again disables the transmitter #0 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO0 pin remains in the high-impedance state.The on-demand mode transmit enable sequence can be the same as the normal mode, or TE0 can be left enabled.

26.6.15 Transmit Clock Control Register (ESAI_TCCR)

The read/write Transmitter Clock Control Register (ESAI_TCCR) controls the ESAI transmitter clock generator bit and frame sync rates, the bit clock and high frequency clock sources and the directions of the HCKT, FST and SCKT signals. In the synchronous mode (SYN=1), the bit clock defined for the transmitter determines the receiver bit clock as well. ESAI_TCCR also controls the number of words per frame for the serial data. Hardware and software reset clear all the bits of the ESAI_TCCR register.

Care should be taken in asynchronous mode whenever the frame sync clock (FSR, FST) is not sourced directly from its associated bit clock (SCKR, SCKT). Proper phase relationships must be maintained between these clocks in order to guarantee proper operation of the ESAI.

NOTE

ARM Core clock is ipg_clk_esai in block ESAI which is from CCM's ahb_clk_root.

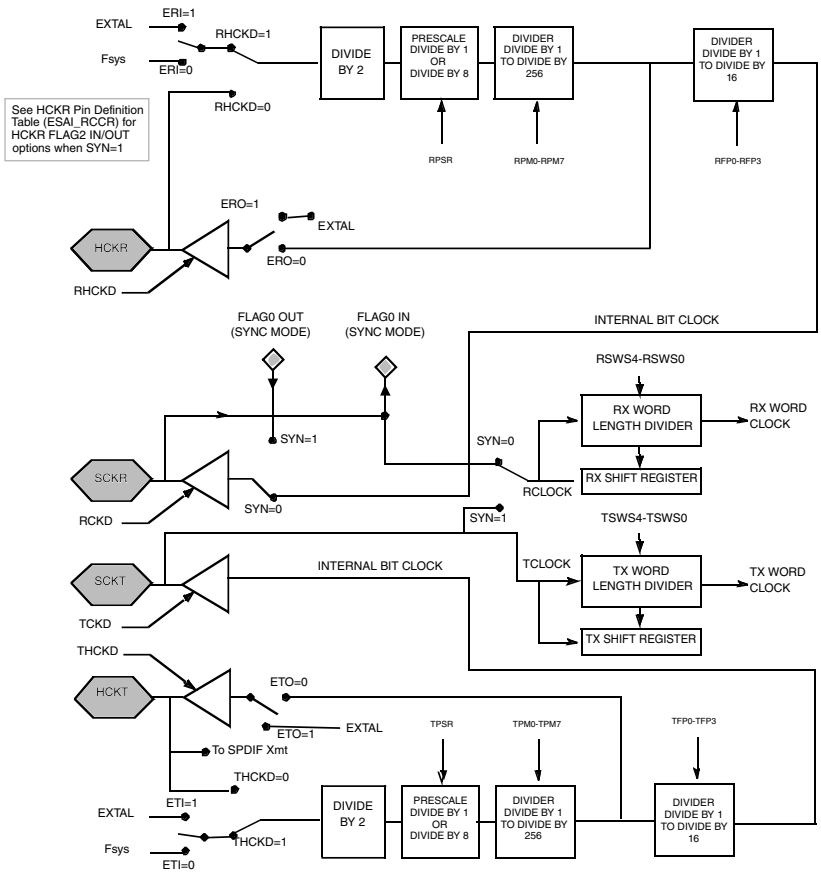


Figure 26-7. ESAI Clock Generator Functional Block Diagram

NOTE

1. ETI, ETO, ERI and ERO bit descriptions are covered in ESAI Control Register (ESAI_ECR). 2. Fsys is the ESAI system 133 MHz clock. 3. EXTAL is the on-chip clock sources other than ESAI system 133MHz clock.

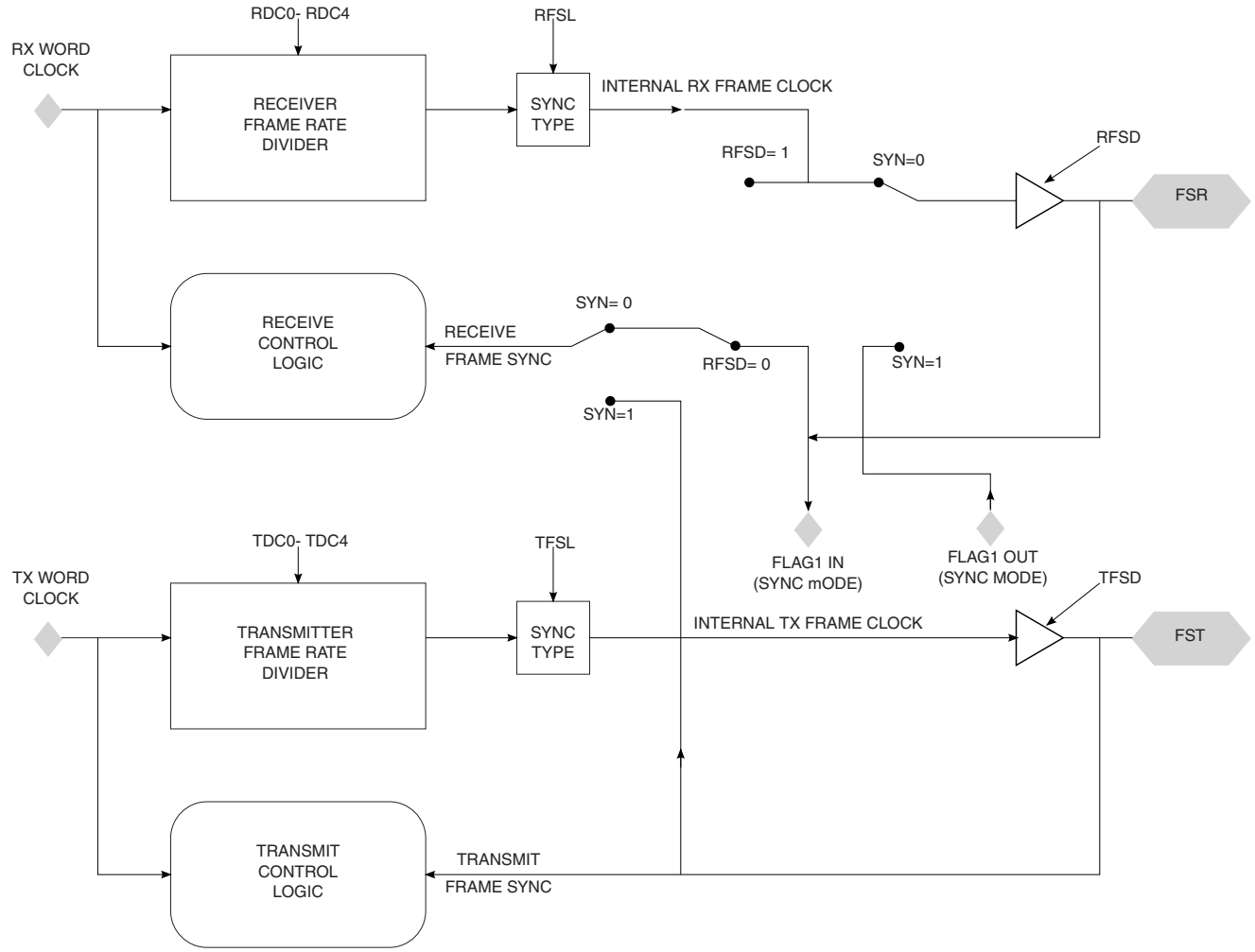


Figure 26-8. ESAI Frame Sync Generator Functional Block Diagram

Table 26-7. Transmitter High Frequency Clock Divider

TFP3-TFP0	Divide Ratio
0x0	1
0x1	2
0x2	3
0x3	4
...	...
0xF	16

Address: 202_4000h base + D8h offset = 202_40D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								THCKD	TFSD	TCKD	THCKP	TFSP	TCKP	TFP	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TFP		TDC					TPSR	TPM							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_TCCR field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 THCKD	ESAI_TCCR Transmit High Frequency Clock Direction. THCKD controls the direction of the HCKT pin. When THCKD is cleared, HCKT is an input; when THCKD is set, HCKT is an output (see Table 26-3).
22 TFSD	ESAI_TCCR Transmit Frame Sync Signal Direction. TFSD controls the direction of the FST pin. When TFSD is cleared, FST is an input; when TFSD is set, FST is an output (see Table 26-3).
21 TCKD	ESAI_TCCR Transmit Clock Source Direction. The Transmitter Clock Source Direction (TCKD) bit selects the source of the clock signal used to clock the transmit shift registers in the asynchronous mode (SYN=0) and the transmit shift registers and the receive shift registers in the synchronous mode (SYN=1). When TCKD is set, the internal clock source becomes the bit clock for the transmit shift registers and word length divider and is the output on the SCKT pin. When TCKD is cleared, the clock source is external; the internal clock generator is disconnected from the SCKT pin, and an external clock source may drive this pin (see Table 26-3).
20 THCKP	ESAI_TCCR Transmit High Frequency Clock Polarity The Transmitter High Frequency Clock Polarity (THCKP) bit controls the polarity of the HCKT. 0 - Normal polarity 1 - Inverted polarity
19 TFSP	ESAI_TCCR Transmit Frame Sync Polarity. The Transmitter Frame Sync Polarity (TFSP) bit determines the polarity of the transmit frame sync signal. When TFSP is cleared, the frame sync signal polarity is positive, that is, the frame start is indicated by a high level on the frame sync pin. When TFSP is set, the frame sync signal polarity is negative, that is, the frame start is indicated by a low level on the frame sync pin.
18 TCKP	ESAI_TCCR Transmit Clock Polarity. The Transmitter Clock Polarity (TCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If TCKP is cleared the data and the frame sync are clocked out on the rising edge of the transmit bit clock and latched in on the falling edge of the transmit bit clock. If TCKP is set the falling edge of the transmit clock is used to clock the data out and frame sync and the rising edge of the transmit clock is used to latch the data and frame sync in.
17–14 TFP	ESAI_TCCR Tx High Frequency Clock Divider. The TFP3-TFP0 bits control the divide ratio of the transmitter high frequency clock to the transmitter serial bit clock when the source of the high frequency clock and the bit clock is the internal ARM Core clock. When the HCKT input is being driven from an external high frequency clock, the TFP3-TFP0 bits specify an additional division ratio in the clock divider chain. Table 26-7 shows the specification for the divide ratio. Figure 26-7 shows the ESAI high frequency clock generator functional diagram.

Table continues on the next page...

ESAI_TCCR field descriptions (continued)

Field	Description
13–9 TDC	ESAI_TCCR Tx Frame Rate Divider Control. The TDC4-TDC0 bits control the divide ratio for the programmable frame rate dividers used to generate the transmitter frame clocks. In network mode, this ratio may be interpreted as the number of words per frame minus one. The divide ratio may range from 2 to 32 (TDC=0x00001 to 0x11111) for network mode. A divide ratio of one (TDC=0x00000) in network mode is a special case (on-demand mode). In normal mode, this ratio determines the word transfer rate. The divide ratio may range from 1 to 32 (TDC=0x00000 to 0x11111) for normal mode. In normal mode, a divide ratio of 1 (TDC=0x00000) provides continuous periodic data word transfers. A bit-length frame sync (TFSL=1) must be used in this case. The ESAI frame sync generator functional diagram is shown in Figure 26-8
8 TPSR	ESAI_TCCR Transmit Prescaler Range. The TPSR bit controls a fixed divide-by-eight prescaler in series with the variable prescaler. This bit is used to extend the range of the prescaler for those cases where a slower bit clock is desired. When TPSR is set, the fixed prescaler is bypassed. When TPSR is cleared, the fixed divide-by-eight prescaler is operational (see Figure 26-7). The maximum internally generated bit clock frequency is Fsys/6; the minimum internally generated bit clock frequency is Fsys/(2 x 8 x 256 x 16)=Fsys/65536. **NOTE:** Do not use the combination TPSR = 1 and TPM7-RPM0 = 0x00, which causes synchronization problems when using the internal core clock as source (THCKD = 1 or TCKD = 1).
TPM	ESAI_TCCR Transmit Prescale Modulus Select. The TPM7-TPM0 bits specify the divide ratio of the prescale divider in the ESAI transmitter clock generator. A divide ratio from 1 to 256 (TPM=0x00 to 0xFF) may be selected. The bit clock output is available at the transmit serial bit clock (SCKT) pin. The bit clock output is also available internally for use as the bit clock to shift the transmit and receive shift registers. The ESAI transmit clock generator functional diagram is shown in Figure 26-7.

26.6.16 Receive Control Register (ESAI_RCR)

The read/write Receive Control Register (ESAI_RCR) controls the ESAI receiver section. Interrupt enable bits for the receivers are provided in this control register. The receivers are enabled in this register (0,1,2 or 3 receivers can be enabled) if the input data pin is not used by a transmitter. Operating modes are also selected in this register.

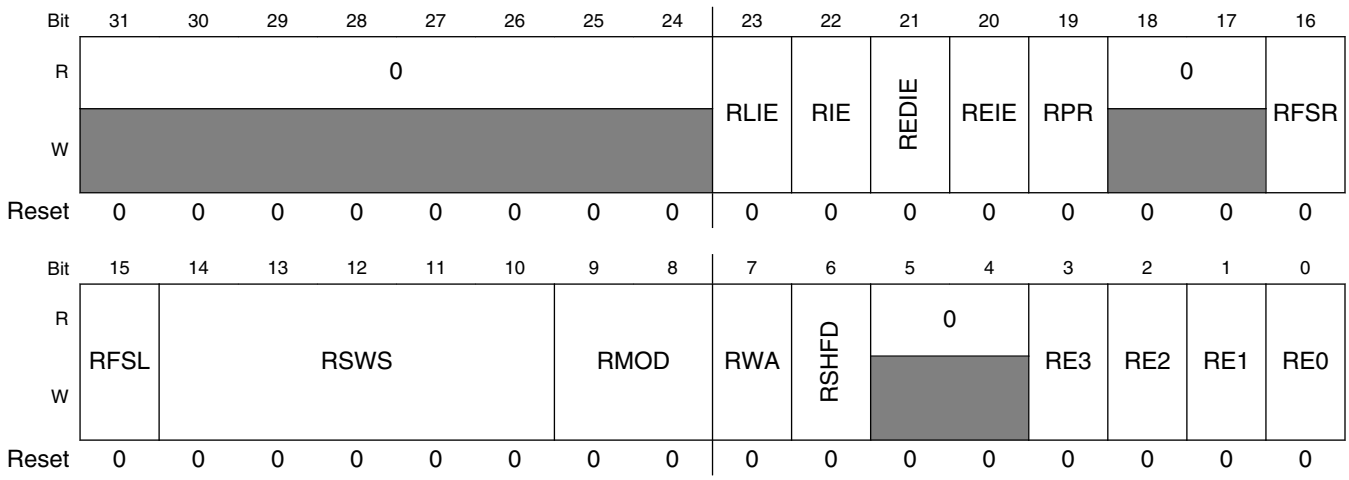
Table 26-8. ESAI Receive Network Mode Selection

RMOD1	RMOD0	RDC4-RDC0	Receiver Network Mode
0	0	0x0-0x1F	Normal Mode
0	1	0x0	On-Demand Mode
0	1	0x1-0x1F	Network Mode
1	0	X	Reserved
1	1	0x0C	AC97

Table 26-9. ESAI Receive Slot and Word Length Selection

RSWS4	RSWS3	RSWS2	RSWS1	RSWS0	SLOT LENGTH	WORD LENGTH
0	0	0	0	0	8	8
0	0	1	0	0	12	8
0	0	0	0	1		12
0	1	0	0	0	16	8
0	0	1	0	1		12
0	0	0	1	0		16
0	1	1	0	0	20	8
0	1	0	0	1		12
0	0	1	1	0		16
0	0	0	1	1		20
1	0	0	0	0	24	8
0	1	1	0	1		12
0	1	0	1	0		16
0	0	1	1	1		20
1	1	1	1	0		24
1	1	0	0	0	32	8
1	0	1	0	1		12
1	0	0	1	0		16
0	1	1	1	1		20
1	1	1	1	1		24
0	1	0	1	1	Reserved	
0	1	1	1	0		
1	0	0	0	1		
1	0	0	1	1		
1	0	1	0	0		
1	0	1	1	0		
1	0	1	1	1		
1	1	0	0	1		
1	1	0	1	0		
1	1	0	1	1		
1	1	1	0	0		
1	1	1	0	1		

Address: 202_4000h base + DCh offset = 202_40DCh



ESAI_RCR field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 RLIE	ESAI_RCR Receive Last Slot Interrupt Enable. RLIE enables an interrupt after the last slot of a frame ended in network mode only. When RLIE is set the Core is interrupted after the last slot in a frame ended regardless of the receive mask register setting. When RLIE is cleared the receive last slot interrupt is disabled. Hardware and software reset clear RLIE. RLIE is disabled when RDC=00000 (on-demand mode). The use of the receive last slot interrupt is described in ESAI Interrupt Requests.
22 RIE	ESAI_RCR Receive Interrupt Enable. The Core is interrupted when RIE and the RDF flag in the ESAI_SAISR status register are set. When RIE is cleared, this interrupt is disabled. Reading the receive data registers of the enabled receivers clears RDF, thus clearing the interrupt. Receive interrupts with exception have higher priority than normal receive data interrupts, therefore if exception occurs (ROE is set) and REIE is set, the ESAI requests an ESAI receive data with exception interrupt from the interrupt controller.
21 REDIE	ESAI_RCR Receive Even Slot Data Interrupt Enable. The REDIE control bit is used to enable the receive even slot data interrupts. If REDIE is set, the receive even slot data interrupts are enabled. If REDIE is cleared, the receive even slot data interrupts are disabled. A receive even slot data interrupt request is generated if REDIE is set and the REDF status flag in the ESAI_SAISR status register is set. Even time slots are all even-numbered time slots (0, 2, 4, etc.) when operating in network mode. The zero time slot is marked by the frame sync signal and is considered to be even. Reading all the data registers of the enabled receivers clears the REDF flag, thus servicing the interrupt. Receive interrupts with exception have higher priority than receive even slot data interrupts, therefore if exception occurs (ROE is set) and REIE is set, the ESAI requests an ESAI receive data with exception interrupt from the interrupt controller.
20 REIE	ESAI_RCR Receive Exception Interrupt Enable. When REIE is set, the Core is interrupted when both RDF and ROE in the ESAI_SAISR status register are set. When REIE is cleared, this interrupt is disabled. Reading the ESAI_SAISR status register followed by reading the enabled receivers data registers clears ROE, thus clearing the pending interrupt.
19 RPR	ESAI_RCR Receiver Section Personal Reset. The RPR control bit is used to put the receiver section of the ESAI in the personal reset state. The transmitter section is not affected. When RPR is cleared, the receiver section may operate normally. When RPR is set, the receiver section enters the personal reset state immediately. When in the personal reset state, the status bits are reset to the same state as after hardware reset.The control bits are not affected by the personal reset state.The receiver data pins are disconnected while in the personal reset state.

Table continues on the next page...

ESAI_RCR field descriptions (continued)

Field	Description
	NOTE: To leave the personal reset state by clearing RPR, the procedure described in ESAI Initialization Examples should be followed.
18–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 RFSR	ESAI_RCR Receiver Frame Sync Relative Timing. RFSR determines the relative timing of the receive frame sync signal as referred to the serial data lines, for a word length frame sync only. When RFSR is cleared the word length frame sync occurs together with the first bit of the data word of the first slot. When RFSR is set the word length frame sync starts one serial clock cycle earlier, that is, together with the last bit of the previous data word.
15 RFSL	ESAI_RCR Receiver Frame Sync Length. The RFSL bit selects the length of the receive frame sync to be generated or recognized. If RFSL is cleared, a word-length frame sync is selected. If RFSL is set, a 1-bit clock period frame sync is selected. Refer to Figure 26-6 for examples of frame length selection.
14–10 RSWS	ESAI_RCR Receiver Slot and Word Select. The RSWS4-RSWS0 bits are used to select the length of the slot and the length of the data words being received through the ESAI. The word length must be equal to or shorter than the slot length. The possible combinations are shown in Table 26-9. See also the ESAI data path programming model in Figure 26-2 and Figure 26-3.
9–8 RMOD	ESAI_RCR Receiver Network Mode Control. The RMOD1 and RMOD0 bits are used to define the network mode of the ESAI receivers, as shown in Table 26-8. In the normal mode, the frame rate divider determines the word transfer rate - one word is transferred per frame sync during the frame sync time slot, as shown in Figure 26-5. In network mode, it is possible to transfer a word for every time slot, as shown in Figure 26-5. For more details, see Modes of Operation. In order to comply with AC-97 specifications, RSWS4-RSWS0 should be set to 0x00011 (20-bit slot, 20-bit word); RFSL and RFSR should be cleared, and RDC4-RDC0 should be set to 0x0C (13 words in frame).
7 RWA	ESAI_RCR Receiver Word Alignment Control. The Receiver Word Alignment Control (RWA) bit defines the alignment of the data word in relation to the slot. This is relevant for the cases where the word length is shorter than the slot length. If RWA is cleared, the data word is assumed to be left-aligned in the slot frame. If RWA is set, the data word is assumed to be right-aligned in the slot frame. If the data word is shorter than the slot length, the data bits which are not in the data word field are ignored. For data word lengths of less than 24 bits, the data word is right-extended with zeroes before being stored in the receive data registers.
6 RSHFD	ESAI_RCR Receiver Shift Direction. The RSHFD bit causes the receiver shift registers to shift data in MSB first when RSHFD is cleared or LSB first when RSHFD is set (see Figure 26-2 and Figure 26-3).
5–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 RE3	ESAI_RCR ESAI Receiver 3 Enable. When RE3 is set and TE2 is cleared, the ESAI receiver 3 is enabled and samples data at the SDO2/SDI3 pin. ESAI_TX2 and ESAI_RX3 should not be enabled at the same time (RE3=1 and TE2=1). When RE3 is cleared, receiver 3 is disabled by inhibiting data transfer into ESAI_RX3. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the ESAI_RX3 data register. If RE3 is set while some of the other receivers are already in operation, the first data word received in ESAI_RX3 will be invalid and must be discarded.
2 RE2	ESAI_RCR ESAI Receiver 2 Enable. When RE2 is set and TE3 is cleared, the ESAI receiver 2 is enabled and samples data at the SDO3/SDI2 pin. ESAI_TX3 and ESAI_RX2 should not be enabled at the same time (RE2=1 and TE3=1). When RE2 is cleared, receiver 2 is disabled by inhibiting data transfer into ESAI_RX2. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the ESAI_RX2 data register. If RE2 is set while some of the other receivers are already in operation, the first data word received in ESAI_RX2 will be invalid and must be discarded.

Table continues on the next page...

ESAI_RCR field descriptions (continued)

Field	Description
1 RE1	ESAI_RCR ESAI Receiver 1 Enable. When RE1 is set and TE4 is cleared, the ESAI receiver 1 is enabled and samples data at the SDO4/SDI1 pin. ESAI_TX4 and ESAI_RX1 should not be enabled at the same time (RE1=1 and TE4=1). When RE1 is cleared, receiver 1 is disabled by inhibiting data transfer into ESAI_RX1. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the ESAI_RX1 data register. If RE1 is set while some of the other receivers are already in operation, the first data word received in ESAI_RX1 will be invalid and must be discarded.
0 RE0	ESAI_RCR ESAI Receiver 0 Enable. When RE0 is set and TE5 is cleared, the ESAI receiver 0 is enabled and samples data at the SDO5/SDI0 pin. ESAI_TX5 and ESAI_RX0 should not be enabled at the same time (RE0=1 and TE5=1). When RE0 is cleared, receiver 0 is disabled by inhibiting data transfer into ESAI_RX0. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the ESAI_RX0 data register. If RE0 is set while some of the other receivers are already in operation, the first data word received in ESAI_RX0 will be invalid and must be discarded.

26.6.17 Receive Clock Control Register (ESAI_RCCR)

The read/write Receiver Clock Control Register (ESAI_RCCR) controls the ESAI receiver clock generator bit and frame sync rates, word length, and number of words per frame for the serial data. The ESAI_RCCR control bits are described in the following paragraphs.

NOTE

ARM Core clock is ipg_clk_esai in block ESAI which is from CCM's ahb_clk_root.

Table 26-10. Receiver High Frequency Clock Divider

RFP3-RFP0	Divide Ratio
0x0	1
0x1	2
0x2	3
0x3	4
...	...
0xF	16

Table 26-11. SCKR Pin Definition Table

Control Bits		SCKR PIN
SYN	RCKD	
0	0	SCKR input

Table continues on the next page...

Table 26-11. SCKR Pin Definition Table (continued)

Control Bits		SCKR PIN
SYN	RCKD	
0	1	SCKR output
1	0	IF0
1	1	OF0

Table 26-12. FSR Pin Definition Table

Control Bits			FSR Pin
SYN	TEBE	RFSD	
0	X	0	FSR input
0	X	1	FSR output
1	0	0	IF1
1	0	1	OF1
1	1	0	reserved
1	1	1	Transmitter Buffer Enable

Table 26-13. HCKR Pin Definition Table

Control Bits		HCKR PIN
SYN	RHCKD	
0	0	HCKR input
0	1	HCKR output
1	0	IF2
1	1	OF2

Address: 202_4000h base + E0h offset = 202_40E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								RHCKD	RFSD	RCKD	RHCKP	RFSP	RCKP	RFP	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RFP		RDC					RPSR	RPM							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_RCCR field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 RHCKD	ESAI_RCCR Receiver High Frequency Clock Direction. The Receiver High Frequency Clock Direction (RHCKD) bit selects the source of the receiver high frequency clock when in the asynchronous mode (SYN=0) and the IF2/OF2 flag direction in the synchronous mode (SYN=1). In the asynchronous mode, when RHCKD is set, the internal clock generator becomes the source of the receiver high frequency clock and is the output on the HCKR pin. In the asynchronous mode, when RHCKD is cleared, the receiver high frequency clock source is external; the internal clock generator is disconnected from the HCKR pin, and an external clock source may drive this pin. When RHCKD is cleared, HCKR is an input; when RHCKD is set, HCKR is an output. In the synchronous mode when RHCKD is set, the HCKR pin becomes the OF2 output flag. If RHCKD is cleared, the HCKR pin becomes the IF2 input flag. Refer to Table 26-2 and Table 26-13.
22 RFSD	ESAI_RCCR Receiver Frame Sync Signal Direction. The Receiver Frame Sync Signal Direction (RFSD) bit selects the source of the receiver frame sync signal when in the asynchronous mode (SYN=0) and the IF1/OF1/Transmitter Buffer Enable flag direction in the synchronous mode (SYN=1). In the asynchronous mode, when RFSD is set, the internal clock generator becomes the source of the receiver frame sync and is the output on the FSR pin. In the asynchronous mode, when RFSD is cleared, the receiver frame sync source is external; the internal clock generator is disconnected from the FSR pin, and an external clock source may drive this pin. In the synchronous mode when RFSD is set, the FSR pin becomes the OF1 output flag or the Transmitter Buffer Enable, according to the TEBE control bit. If RFSD is cleared, the FSR pin becomes the IF1 input flag. Refer to Table 26-2 and Table 26-12 .
21 RCKD	ESAI_RCCR Receiver Clock Source Direction. The Receiver Clock Source Direction (RCKD) bit selects the source of the clock signal used to clock the receive shift register in the asynchronous mode (SYN=0) and the IF0/OF0 flag direction in the synchronous mode (SYN=1). In the asynchronous mode, when RCKD is set, the internal clock source becomes the bit clock for the receive shift registers and word length divider and is the output on the SCKR pin. In the asynchronous mode when RCKD is cleared, the clock source is external; the internal clock generator is disconnected from the SCKR pin, and an external clock source may drive this pin. In the synchronous mode when RCKD is set, the SCKR pin becomes the OF0 output flag. If RCKD is cleared, the SCKR pin becomes the IF0 input flag. Refer to Table 26-2 and Table 26-11.
20 RHCKP	ESAI_RCCR Receiver High Frequency Clock Polarity. The Receiver High Frequency Clock Polarity (RHCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If RHCKP is cleared the data and the frame sync are clocked out on the rising edge of the receive high frequency bit clock and the frame sync is latched in on the falling edge of the receive bit clock. If RHCKP is set the falling edge of the receive clock is used to clock the data and frame sync out and the rising edge of the receive clock is used to latch the frame sync in.
19 RFSP	ESAI_RCCR Receiver Frame Sync Polarity. The Receiver Frame Sync Polarity (RFSP) determines the polarity of the receive frame sync signal. When RFSP is cleared the frame sync signal polarity is positive, that is, the frame start is indicated by a high level on the frame sync pin. When RFSP is set the frame sync signal polarity is negative, that is, the frame start is indicated by a low level on the frame sync pin.
18 RCKP	The Receiver Clock Polarity (RCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If RCKP is cleared the data and the frame sync are clocked out on the rising edge of the receive bit clock and the frame sync is latched in on the falling edge of the receive bit clock. If RCKP is set the falling edge of the receive clock is used to clock the data and frame sync out and the rising edge of the receive clock is used to latch the frame sync in.
17–14 RFP	ESAI_RCCR Rx High Frequency Clock Divider. The RFP3-RFP0 bits control the divide ratio of the receiver high frequency clock to the receiver serial bit clock when the source of the receiver high frequency clock and the bit clock is the internal Arm Core clock. When the HCKR input is being driven

Table continues on the next page...

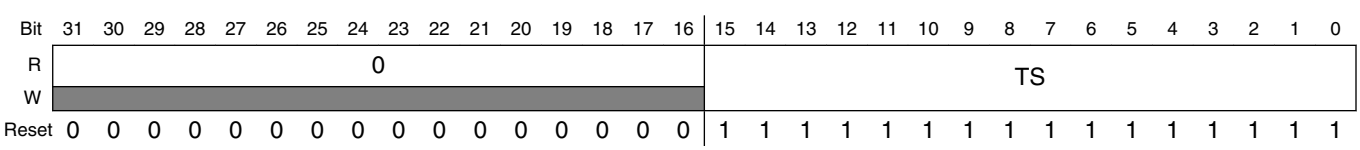
ESAI_RCCR field descriptions (continued)

Field	Description
	from an external high frequency clock, the RFP3-RFP0 bits specify an additional division ration in the clock divider chain. Table 26-10 provides the specification of the divide ratio. Figure 26-7 shows the ESAI high frequency generator functional diagram.
13–9 RDC	ESAI_RCCR Rx Frame Rate Divider Control. The RDC4-RDC0 bits control the divide ratio for the programmable frame rate dividers used to generate the receiver frame clocks. In network mode, this ratio may be interpreted as the number of words per frame minus one. The divide ratio may range from 2 to 32 (RDC=0x00001 to 0x11111) for network mode. A divide ratio of one (RDC=0x00000) in network mode is a special case (on-demand mode). In normal mode, this ratio determines the word transfer rate. The divide ratio may range from 1 to 32 (RDC=0x00000 to 0x11111) for normal mode. In normal mode, a divide ratio of one (RDC=0x00000) provides continuous periodic data word transfers. A bit-length frame sync (RFSL=1) must be used in this case. The ESAI frame sync generator functional diagram is shown in Figure 26-8 .
8 RPSR	ESAI_RCCR Receiver Prescaler Range. The RPSR controls a fixed divide-by-eight prescaler in series with the variable prescaler. This bit is used to extend the range of the prescaler for those cases where a slower bit clock is desired. When RPSR is set, the fixed prescaler is bypassed. When RPSR is cleared, the fixed divide-by-eight prescaler is operational (see Figure 26-7). The maximum internally generated bit clock frequency is Fsys/6, the minimum internally generated bit clock frequency is Fsys/(2 x 8 x 256 x 16)=Fsys/65536. (Do not use the combination RPSR=1 and RPM7-RPM0 =0x00, which causes synchronization problems when using the internal Core clock as source (RHCKD=1 or RCKD=1))
RPM	ESAI_RCCR Receiver Prescale Modulus Select. The RPM7-RPM0 bits specify the divide ratio of the prescale divider in the ESAI receiver clock generator. A divide ratio from 1 to 256 (RPM=0x00 to 0xFF) may be selected. The bit clock output is available at the receiver serial bit clock (SCKR) pin. The bit clock output is also available internally for use as the bit clock to shift the receive shift registers. The ESAI receive clock generator functional diagram is shown in Figure 26-7.

26.6.18 Transmit Slot Mask Register A (ESAI_TSMA)

The Transmit Slot Mask Register A together with Transmit Slot Mask Register B (ESAI_TSMA and ESAI_TSMB) are two read/write registers used by the transmitters in network mode to determine for each slot whether to transmit a data word and generate a transmitter empty condition (TDE=1), or to tri-state the transmitter data pins. Fields ESAI_TSMA[TS] and ESAI_TSMB[TS] are concatenated to form the 32-bit field TS[31:0]. Bit number n in TS is the enable/disable control bit for transmission in slot number n.

Address: 202_4000h base + E4h offset = 202_40E4h



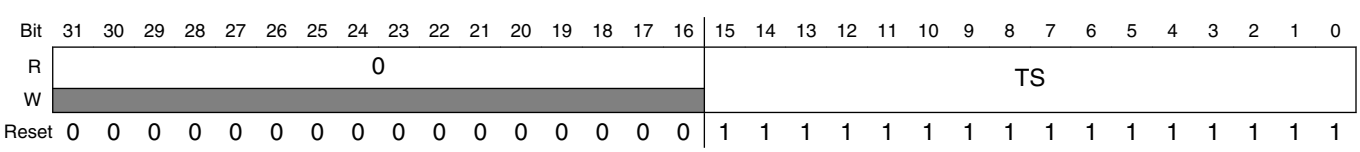
ESAI_TSMA field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TS	Lower 16 bits of TS When bit number N in ESAI_TSMA is cleared, all the transmit data pins of the enabled transmitters are tri-stated during transmit time slot number N. The data is still transferred from the transmit data registers to the transmit shift registers but neither the TDE nor the TUE flags are set. This means that during a disabled slot, no transmitter empty interrupt is generated. The Core is interrupted only for enabled slots. Data that is written to the transmit data registers when servicing this request is transmitted in the next enabled transmit time slot. When bit number N in ESAI_TSMA register is set, the transmit sequence is as usual: data is transferred from the TX registers to the shift registers and transmitted during slot number N, and the TDE flag is set. Using the slot mask in ESAI_TSMA does not conflict with using TSR. Even if a slot is enabled in ESAI_TSMA, the user may choose to write to TSR instead of writing to the transmit data registers TXn. This causes all the transmit data pins of the enabled transmitters to be tri-stated during the next slot. Data written to the ESAI_TSMA affects the next frame transmission. The frame being transmitted is not affected by this data and would comply to the last ESAI_TSMA setting. Data read from ESAI_TSMA returns the last written data. After hardware or software reset, the ESAI_TSMA register is preset to 0x0000FFFF, which means that all 16 possible slots are enabled for data transmission. When operating in normal mode, bit 0 of the ESAI_TSMA register must be set, otherwise no output is generated.

26.6.19 Transmit Slot Mask Register B (ESAI_TSMB)

The Transmit Slot Mask Register B together with Transmit Slot Mask Register A (ESAI_TSMA and ESAI_TSMB) are two read/write registers used by the transmitters in network mode to determine for each slot whether to transmit a data word and generate a transmitter empty condition (TDE=1), or to tri-state the transmitter data pins. Fields ESAI_TSMA[TS] and ESAI_TSMB[TS] are concatenated to form the 32-bit field TS[31:0]. Bit number n in TS is the enable/disable control bit for transmission in slot number n.

Address: 202_4000h base + E8h offset = 202_40E8h



ESAI_TSMB field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TS	When bit number N in ESAI_TSMB is cleared, all the transmit data pins of the enabled transmitters are tri-stated during transmit time slot number N. The data is still transferred from the transmit data registers to the transmit shift registers but neither the TDE nor the TUE flags are set. This means that during a disabled slot, no transmitter empty interrupt is generated. The Core is interrupted only for enabled slots. Data that is written to the transmit data registers when servicing this request is transmitted in the next enabled transmit time slot. When bit number N in ESAI_TSMB register is set, the transmit sequence is as usual: data is transferred from the TX registers to the shift registers and transmitted during slot number N, and the TDE flag is set. Using the slot mask in ESAI_TSMB does not conflict with using TSR. Even if a slot is enabled in TSMB, the user may chose to write to TSR instead of writing to the transmit data registers TXn. This causes all the transmit data pins of the enabled transmitters to be tri-stated during the next slot. Data written to the ESAI_TSMB affects the next frame transmission. The frame being transmitted is not affected by this data and would comply to the last ESAI_TSMB setting. Data read from ESAI_TSMB returns the last written data. After hardware or software reset, the ESAI_TSMB register is preset to 0x0000FFFF, which means that all 16 possible slots are enabled for data transmission.

26.6.20 Receive Slot Mask Register A (ESAI_RSMA)

The Receive Slot Mask Register A together with Receive Slot Mask Register B (ESAI_RSMA and ESAI_RSMB) are two read/write registers used by the receiver in network mode to determine for each slot whether to receive a data word and generate a receiver full condition (RDF=1), or to ignore the received data. Fields ESAI_RSMA[RS] and ESAI_RSMB[RS] are concatenated to form the 32-bit field RS[31:0]. Bit number n in RS is an enable/disable control bit for receiving data in slot number n.

Address: 202_4000h base + ECh offset = 202_40ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																RS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ESAI_RSMA field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RS	When bit number N in the ESAI_RSMA register is cleared, the data from the enabled receivers input pins are shifted into their receive shift registers during slot number N. The data is not transferred from the receive shift registers to the receive data registers, and neither the RDF nor the ROE flag is set. This

Table continues on the next page...

ESAI_RSMA field descriptions (continued)

Field	Description
	means that during a disabled slot, no receiver full interrupt is generated. The Core is interrupted only for enabled slots. When bit number N in the ESAI_RSMA is set, the receive sequence is as usual: data which is shifted into the enabled receivers shift registers is transferred to the receive data registers and the RDF flag is set. Data written to the ESAI_RSMA affects the next received frame. The frame being received is not affected by this data and would comply to the last ESAI_RSMA setting. Data read from ESAI_RSMA returns the last written data. After hardware or software reset, the ESAI_RSMA register is preset to 0x0000FFFF, which means that all 16 possible slots are enabled for data reception. When operating in normal mode, bit 0 of the ESAI_RSMA register must be set to one, otherwise no input is received.

26.6.21 Receive Slot Mask Register B (ESAI_RSMB)

The Receive Slot Mask Register B together with Receive Slot Mask Register A (ESAI_RSMA and ESAI_RSMB) are two read/write registers used by the receiver in network mode to determine for each slot whether to receive a data word and generate a receiver full condition (RDF=1), or to ignore the received data. Fields ESAI_RSMA[RS] and ESAI_RSMB[RS] are concatenated to form the 32-bit field RS[31:0]. Bit number n in RS is an enable/disable control bit for receiving data in slot number n.

Address: 202_4000h base + F0h offset = 202_40F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																RS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ESAI_RSMB field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RS	When bit number N in the ESAI_RSMB register is cleared, the data from the enabled receivers input pins are shifted into their receive shift registers during slot number N. The data is not transferred from the receive shift registers to the receive data registers, and neither the RDF nor the ROE flag is set. This means that during a disabled slot, no receiver full interrupt is generated. The Core is interrupted only for enabled slots. When bit number N in the ESAI_RSMB is set, the receive sequence is as usual: data which is shifted into the enabled receivers shift registers is transferred to the receive data registers and the RDF flag is set. Data written to the ESAI_RSMB affects the next received frame. The frame being received is not affected by this data and would comply to the last ESAI_RSMB setting. Data read from ESAI_RSMB returns the last written data.

Table continues on the next page...

ESAI_RSMB field descriptions (continued)

Field	Description
	After hardware or software reset, the ESAI_RSMB register is preset to 0x0000FFFF, which means that all 16 possible slots are enabled for data reception.

26.6.22 Port C Direction Register (ESAI_PRRC)

There are two registers to control the ESAI personal reset status: Port C Direction Register (ESAI_PRRC) and Port C Control Register (ESAI_PCRC).

The read/write 32-bit Port C Direction Register (ESAI_PRRC) in conjunction with the Port C Control Register (ESAI_PCRC) controls the functionality of the ESAI personal reset state. Table 26-14 provides the port pin configurations. Hardware and software reset clear all ESAI_PRRC bits.

Address: 202_4000h base + F8h offset = 202_40F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																PDC															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_PRRC field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PDC	See Table 26-14.

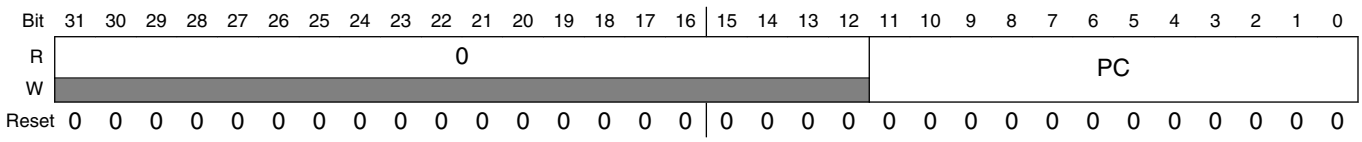
26.6.23 Port C Control Register (ESAI_PCRC)

The read/write 32-bit Port C Control Register (ESAI_PCRC) in conjunction with the Port C Direction Register (ESAI_PRRC) controls the functionality of the ESAI personal reset state. Each of the PC(11:0) bits controls the functionality of the corresponding port pin. Table 26-14 provides the port pin configurations. Hardware and software reset clear all ESAI_PCRC bits.

Table 26-14. PCRC and PRRC Bits Functionality

PDC[i]	PC[i]	Port Pin[i] Function
0	0	Disconnected
1	1	ESAI

Address: 202_4000h base + FCh offset = 202_40FCh



ESAI_PCRC field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PC	See Table 26-14.

Chapter 27
Flexible Controller Area Network (FLEXCAN)

27.1 Overview

The Flexible Controller Area Network (FLEXCAN) module is a communication controller implementing the CAN protocol according to the CAN 2.0B protocol specification.

The CAN protocol was primarily designed to be used as a vehicle serial data bus meeting the specific requirements of this field: real-time processing, reliable operation in the EMI environment of a vehicle, cost-effectiveness and required bandwidth. The FLEXCAN module is a full implementation of the CAN protocol specification, which supports both standard and extended message frames. 64 Message Buffers are supported by the Flexcan module.

27.1.1 Block Diagram

A general block diagram is shown in the figure below,which describes the main sub-blocks implemented in the FLEXCAN module, including the associated memory for storing Mailboxes, Rx Global Mask Registers, Rx Individual Mask Registers, Rx FIFO and Rx FIFO ID Filters.

Support for 64 Mailboxes and 6-deep Rx FIFO is provided. The functions of the sub-modules are described in subsequent sections.

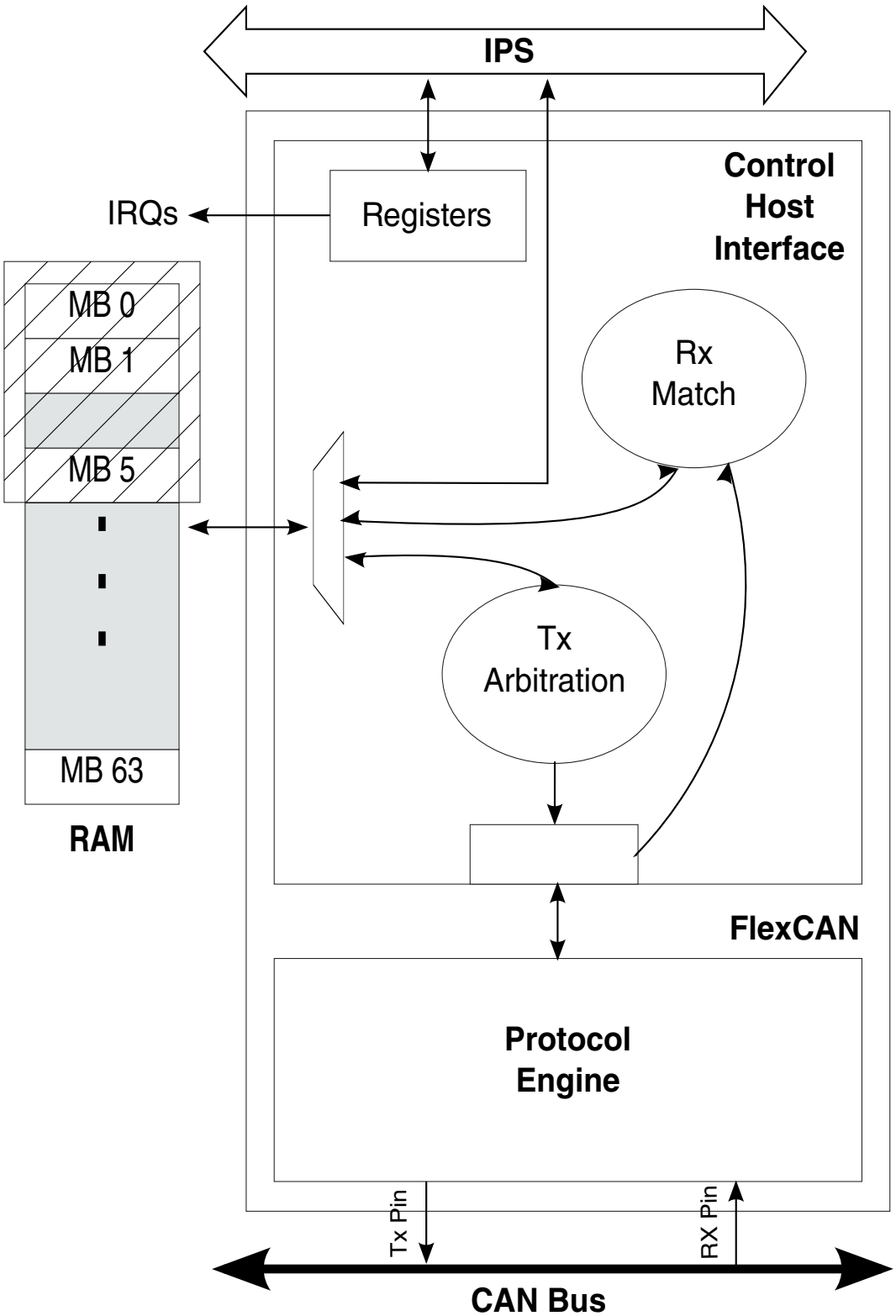


Figure 27-1. FLEXCAN Block Diagram

27.1.2 FLEXCAN Module Features

The FLEXCAN module includes these distinctive legacy features:

- Version 2.0B
  - Standard data and remote frames
  - Extended data and remote frames
  - Zero to eight bytes data length
  - Programmable bit rate up to 1 Mb/sec
  - Content-related addressing
- Flexible Mailboxes of eight bytes data length
- Each Mailbox is configurable as Rx or Tx, all supporting standard and extended messages
- Individual Rx Mask Registers per Mailbox
- Full featured Rx FIFO with storage capacity for 6 frames and internal pointer handling
- Transmission abort capability
- Powerful Rx FIFO ID filtering, capable of matching incoming IDs against either 128 extended, 256 standard or 512 partial (8 bits) IDs, with up to 32 individual masking capability
- 100% backwards compatibility with previous FLEXCAN version
- Unused structures space can be used as general purpose RAM space
- Listen only mode capability
- Programmable loop-back mode supporting self-test operation
- Programmable transmission priority scheme: lowest ID, lowest buffer number or highest priority
- Time Stamp based on 16-bit free-running timer
- Global network time, synchronized by a specific message
- Maskable interrupts independent of the transmission medium (an external transceiver is assumed)
- Short latency time due to an arbitration scheme for high-priority messages
- Low power modes, with programmable wake up on bus activity
- Configurable Glitch filter width to filter the noise on CAN bus when waking up
- Remote request frames may be handled automatically or by software.
- ID filter configuration in Normal Mode
- CAN bit time settings and configuration bits can only be written in Freeze Mode
- Tx mailbox status (Lowest priority buffer or empty buffer)
- SYNC bit status to inform that the module is synchronous with CAN bus
- CRC status for transmitted message
- Selectable priority between Mailboxes and Rx FIFO during matching process

27.1.3 Modes of Operation

The FLEXCAN module has four functional modes: Normal Mode (User and Supervisor), Freeze Mode, Listen-Only Mode and Loop-Back Mode. There are also two low power modes: Disable Mode and Stop Mode.

- Normal Mode (User or Supervisor):

In Normal Mode, the module operates receiving and/or transmitting message frames, errors are handled normally and all the CAN Protocol functions are enabled. User and Supervisor Modes differ in the access to some restricted control registers.

- Freeze Mode:

It is enabled when the FRZ bit in the MCR Register is asserted. If enabled, Freeze Mode is entered when the HALT bit in MCR is set or when Debug Mode is requested at MCU level and the FRZ_ACK bit in the MCR Register is asserted by the FlexCAN. In this mode, no transmission or reception of frames is done and synchronicity to the CAN bus is lost. See Freeze Mode for more information.

- Listen-Only Mode:

The module enters this mode when the LOM bit in the Control Register is asserted. In this mode, transmission is disabled, all error counters are frozen and the module operates in a CAN Error Passive mode. Only messages acknowledged by another CAN station will be received. If FLEXCAN detects a message that has not been acknowledged, it will flag a BIT0 error (without changing the REC), as if it was trying to acknowledge the message.

- Loop-Back Mode:

The module enters this mode when the LPB bit in the Control Register is asserted. In this mode, FLEXCAN performs an internal loop back that can be used for self test operation. The bit stream output of the transmitter is internally fed back to the receiver input. The FLEXCAN_RX input pin is ignored and the FLEXCAN_TX output goes to the recessive state (logic '1'). FLEXCAN behaves as it normally does when transmitting and treats its own transmitted message as a message received from a remote node. In this mode, FLEXCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field to ensure proper reception of its own message. Both transmit and receive interrupts are generated.

- Module Disable Mode:

This low power mode is entered when the MDIS bit in the MCR Register is asserted and the LPM_ACK is asserted by the FlexCAN. When disabled, the module requests to disable the clocks to the CAN Protocol Engine and Controller Host Interface sub-modules. Exit from this mode is done by negating the MDIS bit in the MCR Register. See Module Disable Mode for more information.

- Stop Mode:

This low power mode is entered when Stop Mode is requested at Arm level and the LPM_ACK bit in the MCR Register is asserted by the FlexCAN. When in Stop Mode, the module puts itself in an inactive state and then informs the Arm that the clocks can be shut down globally. Exit from this mode happens when the Stop Mode request is removed or when activity is detected on the CAN bus and the Self Wake Up mechanism is enabled. See Stop Mode for more information.

27.2 External Signals

The FLEXCAN module has two I/O signals.

Table 27-1. FLEXCAN External Signals

Signal	Description	Pad	Mode	Direction
FLEXCAN1_RX	FLEXCAN receive pin. This pin is the receive pin from the CAN bus transceiver. Dominant state is represented by logic level '0'. Recessive state is represented by logic level '1'.	GPIO_8	ALT3	I
		KEY_ROW2	ALT2	
		SD3_CLK	ALT2	
FLEXCAN1_TX	FLEXCAN transmit pin. This pin is the transmit pin to the CAN bus transceiver. Dominant state is represented by logic level '0'. Recessive state is represented by logic level '1'.	GPIO_7	ALT3	O
		KEY_COL2	ALT2	
		SD3_CMD	ALT2	
FLEXCAN2_RX	FLEXCAN receive pin. This pin is the receive pin from the CAN bus transceiver. Dominant state is represented by logic level '0'. Recessive state is represented by logic level '1'.	KEY_ROW4	ALT0	I
		SD3_DAT1	ALT2	
FLEXCAN2_TX	FLEXCAN transmit pin. This pin is the transmit pin to the CAN bus transceiver. Dominant state is represented by logic level '0'. Recessive state is represented by logic level '1'.	KEY_COL4	ALT0	O
		SD3_DAT0	ALT2	

27.3 Clocks

The table found here describes the clock sources for FLEXCAN.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 27-2. FLEXCAN Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_chi	ipg_clk_root	CHI clock
ipg_clk_pe	can_clk_root	Protocol Engine clock
ipg_clk_pe_nogate	can_clk_root	Protocol Engine clock (no gating)
ipg_clk_s	ipg_clk_root	Peripheral access clock
mem_ram_CLK	ipg_clk_root	RAM clock

27.4 Message Buffer Structure

Message Buffer Address: Base + 0x0080-0x047C.

The Message Buffer structure used by the FLEXCAN module is represented in the following table.

Both Extended and Standard Frames (29-bit Identifier and 11-bit Identifier, respectively) used in the CAN specification are represented.

Each individual Message buffer is formed by 16 bytes.

Table 27-3. Message Buffer Structure

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0					CODE					SRR	IDE	RTR	DLC				TIME STAMP															
0x4	PRIO				ID Standard												ID Extended															
0x8	DATA BYTE 0								DATA BYTE 1								DATA BYTE 2								DATA BYTE 3							
0xC	DATA BYTE 4								DATA BYTE 5								DATA BYTE 6								DATA BYTE 7							

CODE - Message Buffer Code

This 4-bit field can be accessed (read or write) by the CPU and by the FLEXCAN module itself, as part of the message buffer matching and arbitration process. The encoding is shown in the following tables. See Functional Description for additional information.

Table 27-4. Message Buffer Code for Rx buffers

CODE Description	Rx Code BEFORE receive New Frame	SRV[1]	Rx Code AFTER successful reception[2]	RRS[3]	Comment
0b0000: INACTIVE- MB is not active.	INACTIVE	-	-	-	MB does not participate in the matching process.
0b0100: EMPTY - MB is active and empty.	EMPTY	-	FULL	-	When a frame is received successfully (after move-in process. Refer to Move-in for details), the CODE field is automatically updated to FULL.
0b0010: FULL - MB is full.	FULL	Yes	FULL	-	The act of reading the C/S word followed by unlocking the MB (SRV) does not make the code return to EMPTY. It remains FULL. If a new frame is moved to the MB after the MB was serviced, the code still remains FULL. Refer to Matching Process for matching details related to FULL code.
		No	OVERRUN	-	If the MB is FULL and a new frame is moved to this MB before the CPU services it, the CODE field is automatically updated to OVERRUN. Refer to Matching Process for details about overrun behavior.
0b0110: OVERRUN - MB is being overwritten into a full buffer.	OVERRUN	Yes	FULL	-	If the CODE field indicates OVERRUN and CPU has serviced the MB, when a new frame is moved to the MB, the code returns to FULL.
		No	OVERRUN	-	If the CODE field already indicates OVERRUN, and another new frame must be moved, the MB will be overwritten again, and the code will remain OVERRUN. Refer to Matching Process for details about overrun behavior.
0b1010: RANSWER[4] - A frame was configured to recognize a Remote Request Frame and transmit a Response Frame in return.	RANSWER	-	TANSWER(0b1110)	0	A Remote Answer was configured to recognize a remote request frame received, after that a MB is set to transmit a response frame. The code is automatically changed to TANSWER (0b1110). Refer to Matching Process for details. If CTRL2[RRS] is negated, transmit a response frame whenever a remote request frame with the same ID is received.
			-	1	This code is ignored during matching and arbitration process. Refer to Matching Process for details.

Table continues on the next page...

Table 27-4. Message Buffer Code for Rx buffers (continued)

CODE Description	Rx Code BEFORE receive New Frame	SRV[1]	Rx Code AFTER successful reception[2]	RRS[3]	Comment
CODE[0]=1b1: BUSY - FlexCAN is updating the contents of the MB. The CPU must not access the MB.	BUSY[5]	-	FULL	-	Indicates that the MB is being updated, it will be negated automatically and does not interfere on the next CODE.
			OVERRUN	-	

1. SRV: Serviced MB. MB was read and unlocked by reading TIMER or other MB.
2. A frame is considered successful reception after the frame to be moved to MB (move-in process). Refer to Move-in for details)
3. Remote Request Stored bit from CTRL2 register. Refer to CTRL2 for details.
4. Code 4'b1010 is not considered as a Tx and a MB with this code should not to be aborted.
5. Note that for Tx MBs, the BUSY bit should be ignored upon read, except when AEN bit is set in the MCR register. If this bit is asserted, the corresponding MB does not participate in the matching process.

Table 27-5. Message Buffer Code for Tx buffers

CODE Description	Tx Code BEFORE tx frame	MB RTR	Tx Code AFTER successful transmission	Comment
0b1000: INACTIVE - MB is not active	INACTIVE	-	-	MB does not participate in the arbitration process.
0b1001: ABORT - MB is aborted	ABORT	-	-	MB does not participate in the arbitration process.
0b1100: DATA - MB is a Tx Data Frame (MB RTR must be 0)	DATA	0	INACTIVE	Transmit data frame unconditionally once. After transmission, the MB automatically returns to the INACTIVE state.
0b1100: REMOTE - MB is a Tx Remote Request Frame (MB RTR must be 1)	REMOTE	1	EMPTY	Transmit remote request frame unconditionally once. After transmission, the MB automatically becomes an Rx Empty MB with the same ID.
0b1110: TANSWER - MB is a Tx Response Frame from an incoming Remote Request Frame	TANSWER	-	RANSWER	This is an intermediate code that is automatically written to the MB by the CHI as a result of match to a remote request frame. The remote response frame will be transmitted unconditionally once and then the code will automatically return to RANSWER (0b1010). The CPU can also write this code with the same effect. The remote response frame can be either a data frame or another remote request frame depending on the RTR bit value. Refer to Matching Process and Arbitration process for details.

SRR - Substitute Remote Request

Fixed recessive bit, used only in extended format. It must be set to '1' by the user for transmission (Tx Buffers) and will be stored with the value received on the CAN bus for Rx receiving buffers. It can be received as either recessive or dominant. If FLEXCAN receives this bit as dominant, then it is interpreted as arbitration loss.

1= Recessive value is compulsory for transmission in Extended Format frames

0= Dominant is not a valid value for transmission in Extended Format frames

IDE - ID Extended Bit

This bit identifies whether the frame format is standard or extended. It is also used as part of the reception filter.

1= Frame format is extended

0= Frame format is standard

RTR - Remote Transmission Request

This bit affects the behavior of Remote Frames and is part of the reception filter. Refer to the tables above and RRS bit in Control 2 Register (FLEXCAN_CTRL2) for additional details.

If FLEXCAN transmits this bit as '1' (recessive) and receives it as '0' (dominant), it is interpreted as arbitration loss. If this bit is transmitted as '0' (dominant), then if it is received as '1' (recessive), the FLEXCAN module treats it as bit error. If the value received matches the value transmitted, it is considered as a successful bit transmission.

1= Indicates the current MB has a Remote Frame to be transmitted if MB is Tx. If the MB is Rx then incoming Remote Request Frames may be stored.

0= Indicates the current MB has a Data Frame to be transmitted. In Rx MB it may be considered in matching processes.

DLC - Length of Data in Bytes

This 4-bit field is the length (in bytes) of the Rx or Tx data, which is located in offset 0x08 through 0x0F of the MB space (see the first table above). In reception, this field is written by the FLEXCAN module, copied from the DLC (Data Length Code) field of the received frame. In transmission, this field is written by the Arm and corresponds to the DLC field value of the frame to be transmitted. When RTR=1, the Frame to be transmitted is a Remote Frame and does not include the data field, regardless of the Length field. The DLC field indicates which DATA BYTEs are valid as shown in the table below.

TIME STAMP - Free-Running Counter Time Stamp

This 16-bit field is a copy of the Free-Running Timer, captured for Tx and Rx frames at the time when the beginning of the Identifier field appears on the CAN bus.

PRIO - Local priority

This 3-bit field is only used when MCR[LPRIO_EN] bit is asserted and it only makes sense for Tx mailboxes. These bits are not transmitted. They are appended to the regular ID to define the transmission priority. See Arbitration process.

ID - Frame Identifier

In Standard Frame format, only the 11 most significant bits (28 to 18) are used for frame identification in both receive and transmit cases. The 18 least significant bits are ignored. In Extended Frame format, all bits are used for frame identification in both receive and transmit cases.

DATA BYTE 0-7 - Data Field

Up to eight bytes can be used for a data frame.

For Rx frames, the data is stored as it is received from the CAN bus. DATA BYTE (n) is valid only if *n* is less than DLC as shown in the table below.

For Tx frames, the CPU prepares the data field to be transmitted within the frame.

Table 27-6. DATA BYTEs validity

DLC	Valid DATA BYTEs
0	none
1	DATA BYTE 0
2	DATA BYTE 0-1
3	DATA BYTE 0-2
4	DATA BYTE 0-3
5	DATA BYTE 0-4
6	DATA BYTE 0-5
7	DATA BYTE 0-6
8	DATA BYTE 0-7

27.5 Rx FIFO Structure

When the MCR[RFEN] bit is set, the memory area from 0x80 to 0xDC (which is normally occupied by MBs 0 to 5) is used by the reception FIFO engine.

The region 0x80-0x8C contains the output of the FIFO which must be read by the CPU as a Message Buffer. This output contains the oldest message received and not read yet. The region 0x90-0xDC is reserved for internal use of the FIFO engine.

An additional memory area, that starts at 0xE0 and may extend up to 0x2DC (normally occupied by MBs 6 up to 37) depending on the CTRL2[RFFN] field setting, contains the ID Filter Table (configurable from 8 to 128 memory positions) that specifies filtering criteria for accepting frames into the FIFO.Table 27-7 shows the Rx FIFO data structure.

Each ID Filter Table Element occupies an entire 32-bit word and can be compounded by one, two or four Identifier Acceptance Filters (IDAF) depending on the MCR[IDAM] field setting. Table 27-8, Table 27-9 and Table 27-10 show the IDAF indexation. Table 27-11 show the three different formats that the IDAF can assume, depending on the MCR[IDAM] field setting. Note that all elements of the table must have the same format. See Rx FIFO for more information.

Out of reset, the ID Filter Table flexible memory area defaults to 0xE0 and only extends to 0xFC, which corresponds to MBs 6 to 7 for RFFN=0.

Table 27-7. Rx FIFO Structure

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x80										SRR	IDE	RTR	DLC				TIME STAMP															
0x84										ID Standard															ID Extended							
0x88	Data Byte 0												Data Byte 1												Data Byte 2							
0x8C	Data Byte 4												Data Byte 5												Data Byte 6							
0x90 to 0xDC	Reserved																															
0xE0	ID Filter Table Element 0																															
0xE4	ID Filter Table Element 1																															
0xE8 to 0x2D4	ID Filter Table Elements 2 through 125																															
0x2D8	ID Filter Table Element 126																															
0x2DC	ID Filter Table Element 127																															

Table 27-8. Position of ID Filter Table Elements in format A

Element	31 … 0
0	IDAF0
1	IDAF1
2 through 125	Identifier Acceptance Filter 2 through 125
126	IDAF126
127	IDAF127

Table 27-9. Position of ID Filter Table Elements in format B

Element	31 … 16	15 … 0
0	IDAF0	IDAF1
1	IDAF2	IDAF3
2 through 125	Identifier Acceptance Filter 4 through 251	
126	IDAF252	IDAF253
127	IDAF254	IDAF255

Table 27-10. Position of ID Filter Table Elements in format C

Element	31 … 24	23 … 16	15 … 8	7 … 0
0	IDAF0	IDAF1	IDAF2	IDAF3
1	IDAF4	IDAF5	IDAF6	IDAF7
2 through 125	Identifier Acceptance Filter 8 through 503			
126	IDAF504	IDAF505	IDAF506	IDAF507
127	IDAF508	IDAF509	IDAF510	IDAF511

Table 27-11. Identifier Acceptance Filter Format A,B and C

Format	31	30	29 … 19	18 … 16	15	14	13 … 8	7 … 0
A	RTR	IDE	RXIDA (Standard = 29-19, Extended = 29-1)					
B	RTR	IDE	RXIDB_0 (Standard = 29-19, Extended = 29-16)		RTR	IDE	RXIDB_1 (Standard = 13-3, Extended = 13-0)	
C	RXIDC_0		RXIDC_1		RXIDC_2			RXIDC_3

Table 27-11. Identifier Acceptance Filter Format A,B and C

	(Std/Ext = 31-24)	(Std/Ext = 23-16)	(Std/Ext = 15-8)	(Std/Ext = 7-0)

RTR - Remote Frame

This bit specifies whether Remote Request Frames are accepted into the FIFO if they match the target ID in Formats A and B. If Format C is chosen the acceptance does not depend on whether the frame is a Remote Request Frame or not.

1= Remote Frames can be accepted and data frames are rejected

0= Remote Frames are rejected and data frames can be accepted

IDE - Extended Frame

Specifies if either Extended or Standard Format frames are accepted into the FIFO if they match the target ID in Formats A and B. If Format C is chosen the acceptance does not depend on whether the frame is of the Extended or Standard Format.

1= Extended frames can be accepted and standard frames are rejected

0= Extended frames are rejected and standard frames can be accepted

RXIDA - Rx Frame Identifier (Format A)

Specifies an ID to be used as acceptance criteria for the FIFO. In the Standard Format (IDAF's or incoming frame's IDE bit is negated), only the 11 most significant bits (29 to 19) are used for frame identification. In the Extended Format (both IDAF's and incoming frame's IDE are asserted), all bits are used.

RXIDB_0, RXIDB_1 - Rx Frame Identifier (Format B)

Specifies an ID to be used as acceptance criteria for the FIFO. In the Standard Format (IDAF's or incoming frame's IDE bit is negated), the 11 most significant bits (29 to 19 and 13 to 3) are used for frame identification. In the Extended Format (both IDAF's and incoming frame's IDE are asserted), all 14 bits of the field are compared with the 14 most significant bits of the Identifier of the incoming frame. The 15 least significant bits of the Identifier of an incoming Extended Format frame do not affect the acceptance.

RXIDC_0, RXIDC_1, RXIDC_2, RXIDC_3 - Rx Frame Identifier (Format C)

Specifies an ID to be used as acceptance criteria for the FIFO. In both Standard Format and Extended Format, all 8 bits of the field are compared to the 8 most significant bits of the Identifier of the incoming frame. The 3 least significant bits of the Identifier of an incoming Standard Format frame and the 21 least significant bits of the Identifier of an incoming Extended Format frame do not affect the acceptance.

27.6 Functional Description

This section provides a complete functional description of the block.

27.6.1 Functional Overview

The FLEXCAN module is a CAN protocol engine with a very flexible mailbox system for transmitting and receiving CAN frames. The mailbox system consists of a set of 64 Message Buffers (MB) that store configuration and control data, time stamp, message ID and data (see Message Buffer Structure). The memory corresponding to the first 38 MBs can be configured to support a FIFO reception scheme with a powerful ID filtering mechanism, capable of checking incoming frames against a table of IDs (up to 128 extended IDs or 256 standard IDs or 512 8-bit ID slices), with individual mask register for up to 32 ID Filter Table elements. Simultaneous reception through FIFO and mailbox is supported. For mailbox reception, a *matching* algorithm makes it possible to store received frames only into MBs that have the same ID. A masking scheme makes it possible to match the ID programmed on the MB with a range of Identifiers on received CAN frames. For transmission, an *arbitration* algorithm decides the prioritization of MBs to be transmitted based on the message ID (optionally augmented by 3 local priority bits) or the MB ordering.

Before proceeding with the functional description, an important concept must be explained. A Message Buffer is said to be "active" at a given time if it can participate in both the Matching and Arbitration processes. An Rx MB with a 0b0000 code is inactive (refer to Table 27-4). Similarly, a Tx MB with a 0b1000 or 0b1001 code is also inactive (refer to Table 27-5).

27.6.2 Transmit Process

In order to transmit a CAN frame, the CPU must prepare a Message Buffer for transmission by executing the procedure found here.

1. Check if the respective interruption bit is set and clear it.
2. If the MB is active (transmission pending), write the ABORT code (0b1001) to the CODE field of the Control and Status word to request an abortion of the transmission. Wait for the corresponding IFLAG to be asserted by polling the IFLAG register or by the interrupt request if enabled by the respective IMASK. Then read back the CODE field to check if the transmission was aborted or transmitted (see Transmission Abort Mechanism). If backwards compatibility is desired (MCR[AEN]

bit negated), just write the INACTIVE code (0b1000) to the CODE field to inactivate the MB but then the pending frame may be transmitted without notification (see Message Buffer Inactivation).
3. Write the ID word.
4. Write the data bytes.
5. Write the DLC, Control and Code fields of the Control and Status word to activate the MB.

Once the MB is activated, it will participate into the arbitration process and eventually be transmitted according to its priority.

At the end of the successful transmission, the value of the Free Running Timer at the time of the second bit of frame's Identifier field is written into the MB's Time Stamp field, the CODE field in the Control and Status word is updated, the CRC Register is updated, a status flag is set in the Interrupt Flag Register and an interrupt is generated if allowed by the corresponding Interrupt Mask Register bit. The new CODE field after transmission depends on the code that was used to activate the MB in step four (see Table 27-4 and Table 27-5 in Message Buffer Structure).

When the Abort feature is enabled (MCR[AEN] bit is asserted), after the Interrupt Flag is asserted for a Mailbox configured as transmit buffer, the Mailbox is blocked, therefore the CPU is not able to update it until it negates the Interrupt Flag. It means that the CPU must clear the corresponding IFLAG before starting to prepare this MB for a new transmission or reception.

27.6.3 Arbitration process

The arbitration process scans the Mailboxes searching the Tx one that holds the message to be sent in the next opportunity. This Mailbox is called the *arbitration winner*.

The scan starts from the lowest Mailbox number and runs toward the higher ones.

The arbitration process is triggered in the following events:

- From the CRC field of the CAN frame. The start point depends on the CTRL2[TASD] field value. See Control 2 Register (FLEXCAN_CTRL2) for details.
- During the error delimiter field of the CAN frame
- During the Overload Delimiter field of a CAN frame.
- When the winner is inactivated and the CAN bus has still not reached the first bit of the Intermission field.
- When Arm write to the C/S word of a winner MB and the CAN bus has still not reached the first bit of the Intermission field.
- When CHI is in Idle state and Arm writes to the C/S word of any MB.

- When FlexCAN exits Bus Off state
- Upon leaving Freeze Mode or Low Power Mode

If the arbitration process does not manage to evaluate all Mailboxes before the CAN bus has reached the first bit of the Intermission field the temporary arbitration winner is invalidated and the FlexCAN will not compete for the CAN bus in the next opportunity.

The arbitration process selects the winner among the active Tx Mailboxes at the end of the scan according to both CTRL1[LBUF] and MCR[LPRIO_EN] bits settings.

27.6.3.1 Lowest Mailbox number first

If CTRL1[LBUF] bit is asserted the first (lowest number) active Tx Mailbox found is the arbitration winner. MCR[LPRIO_EN] bit has no effect when CTRL1[LBUF] is asserted.

27.6.3.2 Highest Mailbox priority first

If CTRL1[LBUF] bit is negated then the arbitration process searches the active Tx Mailbox with the highest priority, and this Mailbox would have a higher probability to win the arbitration on CAN bus.

The sequence of bits considered for this arbitration is called the *arbitration value* of the Mailbox. The highest priority Tx Mailbox is the one that has the least arbitration value among all Tx Mailboxes.

If two or more Mailboxes have equivalent arbitration values the lowest Mailbox number is the arbitration winner.

The composition of the arbitration value depends on MCR[LPRIO_EN] bit setting.

27.6.3.2.1 Local Priority disabled

If MCR[LPRIO_EN] bit is negated the arbitration value is built in the exact sequence of bits as they would be transmitted in a CAN frame (see Table 27-12) in such a way that the Local Priority is disabled.

Table 27-12. Composition of the arbitration value when Local Priority is disabled

Format	Mailbox Arbitration Value (32 bits)				
Standard (IDE = 0)	Standard ID (11 bits)	RTR (1 bit)	IDE (1 bit)	- (18 bits)	- (1 bit)
Extended (IDE = 1)	Extended ID[28:18] (11 bits)	SRR (1 bit)	IDE (1 bit)	Extended ID[17:0] (18 bits)	RTR (1 bit)

27.6.3.2.2 Local Priority enabled

If Local Priority is desired MCR[LPRIO_EN] must be asserted.

In this case the Mailbox PRIO field is included at the very left of the arbitration value (see the table below).

Table 27-13. Composition of the arbitration value when Local Priority is enabled

Format	Mailbox Arbitration Value (35 bits)					
Standard (IDE = 0)	PRIO (3 bits)	Standard ID (11 bits)	RTR (1 bit)	IDE (1 bit)	- (18 bits)	- (1 bit)
Extended (IDE = 1)	PRIO (3 bits)	Extended ID[28:18](11 bits)	SRR (1 bit)	IDE (1 bit)	Extended ID[17:0] (18 bits)	RTR (1 bit)

As the PRIO field is the most significant part of the arbitration value Mailboxes with low PRIO values have higher priority than Mailboxes with high PRIO values regardless the rest of their arbitration values.

Note that the PRIO field is not part of the frame on the CAN bus. Its purpose is only to affect the internal arbitration process.

Once the arbitration winner is found, its content is copied to a hidden auxiliary MB called Tx Serial Message Buffer (Tx SMB), which has the same structure as a normal MB but is not user accessible. This operation is called "move-out" and after it is done, write access to the corresponding MB is blocked (if the AEN bit in MCR is asserted). The write access is released in the following events:

- After the MB is transmitted
- FlexCAN enters in Freeze Mode or Bus Off
- FlexCAN loses the bus arbitration or there is an error during the transmission

At the first opportunity window on the CAN bus, the message on the Tx SMB is transmitted according to the CAN protocol rules. FlexCAN transmits up to eight data bytes, even if the DLC (Data Length Code) field value is greater than that.

Arbitration process can be triggered in the following situations:

- During Rx and Tx frames from CAN CRC field to end of frame. Arbitration start point depends on instantiation parameters NUMBER_OF_MB and TASD. Additionally, TASD value may be changed (see Control 2 Register (FLEXCAN_CTRL2)) to optimize the arbitration start point.
- During CAN Bus Off state from TX_ERR_CNT=124 to 128. Arbitration start point depends on instantiation parameters NUMBER_OF_MB and TASD. Additionally,

TASD value may be changed(see Control 2 Register (FLEXCAN_CTRL2)) to optimize the arbitration start point.

- During C/S write by CPU in Bus Idle. First C/S write starts arbitration process and a second C/S write during this same arbitration restarts the process. If other C/S writes are performed, Tx arbitration process is pending. If there is no arbitration winner after arbitration process has finished, then TX arbitration machine begins a new arbitration process.
- Arbitration winner deactivation during a valid arbitration window.
- Upon Leave Freeze Mode. If there is a re-synchronization during WaitForBusIdle arbitration process is restarted.

Arbitration process stops in the following situation:

- All Mailboxes were scanned.
- A Tx active Mailbox is found in case of Lowest Buffer feature enabled.
- Arbitration winner inactivation or abort during any arbitration process.
- There was not enough time to finish Tx arbitration process. For instance, a deactivation was performed near the end of frame). In this case arbitration process is pending.
- Error or Overload flag in the bus.
- Low Power or Freeze Mode request in Idle state

Arbitration is considered pending as described below:

- It was not possible to finish arbitration process in time.
- C/S write during arbitration if write is performed in a MB which number is lower than the Tx arbitration pointer.
- Any C/S write if there is no Tx Arbitration process in progress.
- Rx Match has just updated a Rx Code to Tx Code.
- Entering Bus off state.

C/S write during arbitration has the following effect:

- If C/S write is performed in the arbitration winner, a new process is restarted immediately.
- C/S write during arbitration if write is performed in a MB which number is higher than the Tx arbitration pointer.

27.6.4 Receive Process

To be able to receive CAN frames into a Mailbox, the CPU must prepare it for reception by executing the steps listed here.

1. If the Mailbox is active (either Tx or Rx) inactivate the Mailbox (see Message Buffer Inactivation), preferably with a *safe inactivation* (see Transmission Abort Mechanism);
2. Write the ID word;
3. Write the EMPTY code (0b0100) to the CODE field of the Control and Status word to activate the Mailbox.

Once the Mailbox is activated in the third step, it will be able to receive frames that match the programmed filter. At the end of a successful reception, the Mailbox is updated by the *move-in process* (see Move-in) as follows:

1. The received Data field (8 bytes at most) is stored;
2. The received Identifier field is stored;
3. The value of the Free Running Timer at the time of the second bit of frame's Identifier field is written into the Mailbox Time Stamp field;
4. The received SRR, IDE, RTR and DLC fields are stored;
5. The CODE field in the Control and Status word is updated. (see Table 27-4 and Table 27-5 in Section Message Buffer Structure)
6. A status flag is set in the Interrupt Flag Register and an interrupt is generated if allowed by the corresponding Interrupt Mask Register bit.

The recommended way for the CPU servicing (read) the frame received in a Mailbox is using the following procedure:

1. Read the Control and Status word of that Mailbox;
2. Check if the BUSY bit is deasserted, indicating that the Mailbox is locked. Repeat step 1) while it is asserted. See Message Buffer Lock Mechanism;
3. Read the contents of the Mailbox. Once Mailbox is locked now, its contents won't be modified by FlexCAN Move-in processes. See Move-in;
4. Acknowledge the proper flag at IFLAG registers;
5. Read the Free Running Timer. It is optional but recommended to unlock Mailbox as soon as possible and make it available for reception.

The CPU should synchronize to frame reception by the status flag bit for the specific Mailbox in one of the IFLAG Registers and not by the CODE field of that Mailbox. Polling the CODE field does not work because once a frame was received and the CPU services the Mailbox (by reading the C/S word followed by unlocking the Mailbox), the CODE field will not return to EMPTY. It will remain FULL. If the CPU tries to workaround this behavior by writing to the C/S word to force an EMPTY code after reading the Mailbox without a prior *safe inactivation*, a newly received message matching the filter of that Mailbox may be lost.

In summary: never do polling by reading directly the C/S word of the Mailboxes. Instead, read the IFLAG registers.

Note that the received frame's Identifier field is always stored in the matching Mailbox, thus the contents of the ID field in a Mailbox may change if the match was due to masking. Note also that FlexCAN does receive frames transmitted by itself if there exists a matching Rx Mailbox, provided the MCR[SRX_DIS] bit is not asserted. If MCR[SRX_DIS] bit is asserted, FlexCAN will not store messages transmitted by itself in any MB, even if it contains a matching MB, and no interrupt flag or interrupt signal will be generated due to the frame reception.

To be able to receive CAN messages through the Rx FIFO, the CPU must enable and configure the Rx FIFO during Freeze Mode(see Rx FIFO). Upon receiving the Frames Available in Rx FIFO interrupt(see Interrupt Masks 1 Register (FLEXCAN_IMASK1), bit IFLAG[BUF5I] - Frames available in Rx FIFO), the CPU should service the received frame using the following procedure:

1. Read the Control and Status word (optional - needed only if a mask was used for IDE and RTR bits);
2. Read the ID field (optional - needed only if a mask was used);
3. Read the Data field;
4. Read the RXFIR register (optional);
5. Clear the Frames Available in Rx FIFO interrupt by writing 1 to IFLAG[BUF5I] bit (mandatory - releases the MB and allows the CPU to read the next Rx FIFO entry)

27.6.5 Matching Process

The matching process scans the MB memory looking for Rx MBs programmed with the same ID as the one received from the CAN bus. If the FIFO is enabled, the priority of scanning can be selected between Mailboxes and FIFO filters. In any case, the matching starts from the lowest number Message Buffer toward the higher ones. If no match is found within the first structure then the other is scanned subsequently. In the event that the FIFO is full, the matching algorithm will always look for a matching MB outside the FIFO region.

As the frame is being received, it is stored in a hidden auxiliary MB called Rx Serial Message Buffer (Rx SMB).

The matching process start point depends on the following conditions:

- if the received frame is a remote frame, the start point is the CRC field of the frame;
- if the received frame is a data frame with DLC field equal to zero, the start point is the CRC field of the frame;
- if the received frame is a data frame with DLC field different than zero, the start point is the DATA field of the frame;

If a matching ID is found in the FIFO table or in one of the Mailboxes, the contents of the SMB will be transferred to the FIFO or to the matched Mailbox by the move-in process. If any CAN protocol error is detected then no match results will be transferred to the FIFO or to the matched Mailbox at the end of reception.

The matching process scans all matching elements of both Rx FIFO (if enabled) and active Rx Mailboxes (CODE is EMPTY, FULL, OVERRUN or RANSWER) in search of a successful comparison with the matching elements of the Rx SMB that is receiving the frame on the CAN bus. The SMB has the same structure of a Mailbox. The reception structures (Rx FIFO or Mailboxes) associated with the matching elements that had a successful comparison are the *matched structures*. The *matching winner* is selected at the end of the scan among those matched structures and depends on conditions described ahead. Refer to the following table for details.

Table 27-14. Matching architecture

Structure	SMB[RTR]	CTRL2[RRS]	CTRL2[EACEN]	MB[IDE]	MB[RTR]	MB[ID][1]	MB[CODE]
Mailbox	0	-	0	cmp[2]	no_cmp[3]	cmp_msk[4]	EMPTY or FULL or OVERRUN
Mailbox	0	-	1	cmp_msk	cmp_msk	cmp_msk	EMPTY or FULL or OVERRUN
Mailbox	1	0	-	cmp	no_cmp	cmp	RANSWER
Mailbox	1	1	0	cmp	no_cmp	cmp_msk	EMPTY or FULL or OVERRUN
Mailbox	1	1	1	cmp_msk	cmp_msk	cmp_msk	EMPTY or FULL or OVERRUN
FIFO[5]	-	-	-	cmp_msk	cmp_msk	cmp_msk	-

1. For Mailbox structure, If SMB[IDE] is asserted, the ID is 29 bits (ID Standard + ID Extended). In case of SMB[IDE] to be negated, the ID is only 11 bits (ID Standard). Please, refer to Message Buffer Structure for ID details. For FIFO structure, the ID depends on IDAM. Please, refer to Rx FIFO Structure for IDAM details.
2. cmp: Compares the SMB contents with the MB contents regardless the masks.
3. no_cmp: The SMB contents are not compared with the MB contents.
4. cmp_msk: Compares the SMB contents with MB contents taking into account the masks.
5. SMB[IDE] and SMB[RTR] are not taken into account when IDAM is type C.

A reception structure is *free-to-receive* when any of the following conditions is satisfied:

- the CODE field of the Mailbox is EMPTY;
- the CODE field of the Mailbox is either FULL or OVERRUN and it has already been serviced (the C/S word was read by the Arm and unlocked as described in Message Buffer Lock Mechanism);

- the CODE field of the Mailbox is either FULL or OVERRUN and an inactivation is performed. (see Message Buffer Inactivation)
- the Rx FIFO is not full.

The scan order for Mailboxes and Rx FIFO is from the matching element with lowest number to the higher ones.

The matching winner search for Mailboxes is affected by the MCR[IRMQ] bit. If it is negated the matching winner is the first matched Mailbox regardless if it is free-to-receive or not . If it is asserted, the matching winner is selected according to the priority below:

1. the first free-to-receive matched Mailbox;
2. the last non free-to-receive matched Mailbox.

It is possible to select the priority of scan between Mailboxes and Rx FIFO by the CTRL2[MRP] bit.

If the selected priority is Rx FIFO first:

- if the Rx FIFO is a matched structure and is free-to-receive then the Rx FIFO is the matching winner regardless of the scan for Mailboxes;
- otherwise (the Rx FIFO is not a matched structure or is not free-to-receive), then the matching winner is searched among Mailboxes as described above.

If the selected priority is Mailboxes first:

- if a free-to-receive matched Mailbox is found, it is the matching winner regardless the scan for Rx FIFO;
- if no matched Mailbox is found, then the matching winner is searched in the scan for the Rx FIFO;
- if both conditions above are not satisfied and a non free-to-receive matched Mailbox is found then the matching winner determination is conditioned by the MCR[IRMQ] bit:
  - if MCR[IRMQ] bit is negated the matching winner is the first matched Mailbox;
  - if MCR[IRMQ] bit is asserted the matching winner is the Rx FIFO if it is a free-to-receive matched structure, otherwise the matching winner is the last non free-to-receive matched Mailbox.

Please, refer to the table below for a summary of matching possibilities.

If a non-safe Mailbox inactivation (see Message Buffer Inactivation) occurs during matching process and the Mailbox inactivated is the temporary matching winner then the temporary matching winner is invalidated. The matching elements scan is not stopped nor

restarted, it continues normally. The consequence is that the current matching process works as if the matching elements compared before the inactivation did not exist, therefore a message may be lost.

Suppose, for example, that the FIFO is disabled, IRMQ is enabled and there are two MBs with the same ID, and FlexCAN starts receiving messages with that ID. Let us say that these MBs are the second and the fifth in the array. When the first message arrives, the matching algorithm will find the first match in MB number 2. The code of this MB is EMPTY, so the message is stored there. When the second message arrives, the matching algorithm will find MB number 2 again, but it is not "free-to-receive", so it will keep looking and find MB number 5 and store the message there. If yet another message with the same ID arrives, the matching algorithm finds out that there are no matching MBs that are "free-to-receive", so it decides to overwrite the last matched MB, which is number 5. In doing so, it sets the CODE field of the MB to indicate OVERRUN.

Table 27-15. Matching Possibilities and Resulting Reception Structures

RFEN	IRMQ	MRP	Matched in MB	Matched in FIFO	Reception Structure	Description
No FIFO, only MB, match is always MB first						
0	0	X[1]	None[2]	-[3]	None	Frame lost by no match
0	0	X	Free[4]	-	FirstMB	
0	1	X	None	-	None	Frame lost by no match
0	1	X	Free	-	FirstMB	
0	1	X	NotFree	-	LastMB	Overrun
FIFO enabled, no match in FIFO is as if FIFO does not exist						
1	0	X	None	None[5]	None	Frame lost by no match
1	0	X	Free	None	FirstMB	
1	1	X	None	None	None	Frame lost by no match
1	1	X	Free	None	FirstMB	
1	1	X	NotFree	None	LastMB	Overrun
FIFO enabled, Queue disabled						
1	0	0	X	NotFull[6]	FIFO	
1	0	0	None	Full[7]	None	Frame lost by FIFO full (FIFO Overflow)
1	0	0	Free	Full	FirstMB	
1	0	0	NotFree	Full	FirstMB	
1	0	1	None	NotFull	FIFO	
1	0	1	None	Full	None	Frame lost by FIFO full (FIFO Overflow)
1	0	1	Free	X	FirstMB	
1	0	1	NotFree	X	FirtsMB	Overrun
FIFO enabled, Queue enabled						
1	1	0	X	NotFull	FIFO	
1	1	0	None	Full	None	Frame lost by FIFO full (FIFO Overflow)

Table continues on the next page...

Table 27-15. Matching Possibilities and Resulting Reception Structures (continued)

RFEN	IRMQ	MRP	Matched in MB	Matched in FIFO	Reception Structure	Description
1	1	0	Free	Full	FirstMB	
1	1	0	NotFree	Full	LastMB	Overrun
1	1	1	None	NotFull	FIFO	
1	1	1	Free	X	FirstMB	
1	1	1	NotFree	NotFull	FIFO	
1	1	1	NotFree	Full	LastMB	Overrun

1. It is a don't care condition.
2. Matched in MB "None" means that the frame has not matched any MB (free-to-receive or non-free-to-receive).
3. It is a forbidden condition.
4. Matched in MB "Free" means that the frame matched at least one MB free-to-receive regardless it has matched MBs non-free-to-receive.
5. Matched in FIFO "None" means that the frame has not matched any filter in FIFO. It is as the FIFO didn't exist (CTRL2[RFEN]=0).
6. Matched in FIFO "NotFull" means that the frame has matched a FIFO filter and has empty slots to receive it.
7. Matched in FIFO "Full" means that the frame has matched a FIFO filter but couldn't store it because it has no empty slots to receive it.

The ability to match the same ID in more than one MB can be exploited to implement a reception queue (in addition to the full featured FIFO) to allow more time for Arm to service the MBs. By programming more than one MB with the same ID, received messages will be queued into the MBs. Arm can examine the Time Stamp field of the MBs to determine the order in which the messages arrived.

Matching to a range of IDs is possible by using ID Acceptance Masks. FlexCAN supports individual masking per MB. Please refer to Rx Mailboxes Global Mask Register (FLEXCAN_RXMGMASK). During the matching algorithm, if a mask bit is asserted, then the corresponding ID bit is compared. If the mask bit is negated, the corresponding ID bit is "don't care". Please note that the Individual Mask Registers are implemented in RAM, so they are not initialized out of reset. Also, they can only be programmed while the module is in Freeze Mode,, otherwise they are blocked by hardware.

FlexCAN also supports an alternate masking scheme with only four mask registers (RGXMASK, RX14MASK, RX15MASK and RXFGMASK) for backward compatibility. This alternate masking scheme is enabled when the IRMQ bit in the MCR Register is negated.

27.6.6 Move Process

There are two types of move process, namely move-in and move-out.

27.6.6.1 Move-in

The move-in process is the copy of a message received by an Rx SMB to a Rx Mailbox or FIFO that has matched it. If the move destination is the Rx FIFO, attributes of the message are also copied to the RXFIR FIFO. Each Rx SMB has its own move-in process, but only one is performed at a given time as described ahead. The move-in starts only when the message held by the Rx SMB has a corresponding matching winner (see Matching Process) and all of the following conditions are true:

- the CAN bus has reached or let past either:
  - the second bit of Intermission field next to the frame that carried the message that is in the Rx SMB;
  - the first bit of an overload frame next to the frame that carried the message that is in the Rx SMB;
- there is no ongoing matching process;
- the destination Mailbox is not locked by Arm;
- there is no ongoing move-in process from another Rx SMB. If more than one move-in processes are to be started at the same time both are performed and the newest substitutes the oldest.

The term *pending move-in* is used throughout the document and stands for a move-to-be that still does not satisfy all of the aforementioned conditions.

The move-in is cancelled and the Rx SMB is able to receive another message if any of the following conditions is satisfied:

- the destination Mailbox is inactivated after the CAN bus has reached the first bit of Intermission field next to the frame that carried the message and its matching process has finished;
- there is a previous pending move-in to the same destination Mailbox.
- the Rx SMB is receiving a frame transmitted by the FlexCAN itself and the self-reception is disabled;
- any CAN protocol error is detected.

Note that the pending move-in is not cancelled if the module enters in Freeze or Low Power Mode. It only stays on hold waiting for exiting Low Power Mode and to be unlocked. If an MB is unlocked during Freeze Mode, the move-in happens immediately.

The move-in process consists of the following steps:

1. if the message is destined to the Rx FIFO, push IDHIT into the RXFIR FIFO;
2. reads the words DATA0-3 and DATA4-7 from the Rx SMB;
3. writes it in the words DATA0-3 and DATA4-7 of the Rx Mailbox;

4. reads the words Control/Status and ID from the Rx SMB;
5. writes it in the words Control/Status and ID of the Rx Mailbox, updating the CODE field according to Table 27-4.

The move-in process is not atomic, in such a way that it is immediately cancelled by the inactivation of the destination Mailbox (see Message Buffer Inactivation) and in this case the Mailbox may be left partially updated, thus incoherent. The exception is if the move-in destination is an Rx FIFO Message Buffer, then the process cannot be cancelled.

The BUSY Bit (least significant bit of the CODE field) of the destination Message Buffer is asserted while the move-in is being performed in such a way that Arm beware that the Message Buffer content is temporarily incoherent.

27.6.6.2 Move-out

The move-out process is the copy of the content from a Tx Mailbox to the Tx SMB when a message for transmission is available (see Arbitration process). The move-out occurs in the following conditions:

- the first bit of Intermission field;
- during Bus off field when TX Error Counter is in the 124 to 128 range;
- during BusIdle field;
- during Wait For Bus Idle field.

The move-out process is not atomic. Only Arm has priority to access the memory concurrently out of BusIdle state. In BusIdle, the move-out has the lowest priority to the concurrent memory accesses.

27.6.7 Data Coherence

In order to maintain data coherency and FlexCAN proper operation, the Arm must obey the rules described in the Transmit Process and Receive Process.

Any form of Arm accessing an MB structure within FlexCAN other than those specified may cause FlexCAN to behave in an unpredictable way.

27.6.7.1 Transmission Abort Mechanism

The abort mechanism provides a safe way to request the abortion of a pending transmission. A feedback mechanism is provided to inform Arm if the transmission was aborted or if the frame could not be aborted and was transmitted instead.

In order to abort a transmission, Arm must write a specific abort code (0b1001) to the CODE field of the Control and Status word. The active MBs configured as transmission must be aborted first and then they may be updated. If the abort code is written to a Mailbox that is currently being transmitted, or to a Mailbox that was already loaded into the SMB for transmission, the write operation is blocked and the MB is kept active, but the abort request is captured and kept pending until one of the following conditions are satisfied:

- The module loses the bus arbitration
- There is an error during the transmission
- The module is put into Freeze Mode
- The module enters in BusOff state
- There is an overload frame

If none of conditions above are reached, the MB is transmitted correctly, the interrupt flag is set in the IFLAG register and an interrupt to the Arm is generated (if enabled). The abort request is automatically cleared when the interrupt flag is set. In the other hand, if one of the above conditions is reached, the frame is not transmitted, therefore the abort code is written into the CODE field, the interrupt flag is set in the IFLAG and an interrupt is (optionally) generated to Arm.

If Arm writes the ABORT code before the transmission begins internally, then the write operation is not blocked, therefore the MB is updated and the interrupt flag is set. In this way Arm just needs to read the abort code to make sure the active MB was *safely inactivated*. Although the AEN bit is asserted and Arm wrote the abort code, in this case the MB is inactivated and not aborted, because the transmission did not start yet. One Mailbox is only aborted when the abort request is captured and kept pending until one of the previous conditions are satisfied.

The abort procedure can be summarized as follows:

1. Arm checks the corresponding IFLAG and clears it, if asserted.
2. Arm writes 0b1001 into the CODE field of the C/S word.
3. Arm waits for the corresponding IFLAG indicating that the frame was either transmitted or aborted.
4. Arm reads the CODE field to check if the frame was either transmitted (CODE=0b1000) or aborted (CODE=0b1001).
5. It is necessary to clear the corresponding IFLAG in order to allow the MB to be reconfigured.

27.6.7.2 Message Buffer Inactivation

Inactivation is a mechanism provided to protect the Mailbox against updates by the FlexCAN internal processes, thus allowing Arm to rely on Mailbox data coherence after having updated it, even in Normal Mode.

If a Mailbox is inactivated it does not participate neither in the arbitration nor in the matching process until it is reactivated. See Transmit Process and Receive Process for more detailed instruction on how to inactivate and reactivate a Mailbox.

In order to inactivate a Mailbox Arm must update its CODE field to INACTIVE (either 0b0000 or 0b1000).

As the user is not able to synchronize the CODE field update with the FlexCAN internal processes an inactivation can lead to undesirable results:

- a frame in the bus that matches the filtering of the inactivated Rx Mailbox may be lost without notice, even if there are other Mailboxes with the same filter;
- a frame containing the message within the inactivated Tx Mailbox may be transmitted without notice.

In order to eliminate such risk and perform a *safe inactivation* Arm must use the following mechanism along with the inactivation itself:

- for Tx Mailboxes, the Transmission Abort (see Transmission Abort Mechanism);

The inactivation automatically unlocks the Mailbox (see Message Buffer Lock Mechanism).

Message Buffers that are part of the Rx FIFO cannot be inactivated. There is no write protection on FIFO region by FlexCAN. Arm must keep the data coherence into FIFO region when RFEN is asserted.

27.6.7.3 Message Buffer Lock Mechanism

Besides MB inactivation, FlexCAN has another data coherence mechanism for the receive process. When Arm reads the Control and Status word of an Rx MB with codes FULL or OVERRUN, FlexCAN assumes that Arm wants to read the whole MB in an atomic operation, and thus it sets an internal lock flag for that MB. The lock is released when Arm reads the Free Running Timer (global unlock operation), or when it reads the Control and Status word of another MB regardless of its code or when Arm writes into C/S word from locked MB. The MB locking is done to prevent a new frame to be written into the MB while Arm is reading it.

The locking mechanism only applies to Rx MBs that are not part of FIFO and have a code different than INACTIVE (0b0000) or EMPTY[1] (0b0100). Also, Tx MBs can not be locked.

Suppose, for example, that the FIFO is disabled and the second and the fifth MBs of the array are programmed with the same ID, and FlexCAN has already received and stored messages into these two MBs. Suppose now that the Arm decides to read MB number 5 and at the same time another message with the same ID is arriving. When Arm reads the Control and Status word of MB number 5, this MB is locked. The new message arrives and the matching algorithm finds out that there are no "free-to-receive" MBs, so it decides to override MB number 5. However, this MB is locked, so the new message can not be written there. It will remain in the SMB waiting for the MB to be unlocked, and only then will be written to the MB. If the MB is not unlocked in time and yet another new message with the same ID arrives, then the new message overwrites the one on the SMB and there will be no indication of lost messages either in the CODE field of the MB or in the Error and Status Register.

While the message is being moved-in from the SMB to the MB, the BUSY bit on the CODE field is asserted. If Arm reads the Control and Status word and finds out that the BUSY bit is set, it should defer accessing the MB until the BUSY bit is negated.

If the BUSY bit is asserted or if the MB is empty, then reading the Control and Status word does not lock the MB.

Inactivation takes precedence over locking. If Arm inactivates a locked Rx Mailbox, then its lock status is negated and the Mailbox is marked as invalid for the current matching round. Any pending message on the SMB will not be transferred anymore to the Mailbox.An MB is unlocked when Arm reads the Free Running Timer Register (see Free Running Timer Register (FLEXCAN_TIMER)), or the C/S word of another MB.

Lock and unlock mechanisms have the same functionality in both Normal and Freeze modes.

An unlock during Normal or Freeze mode results in the move-in of the pending message. However, the move-in is postponed if an unlock occurs during any of the low power modes (see in Modes of Operation specific information on Module Disable or Stop modes) and it will take place only when the module resumes to Normal or Freeze modes.

27.6.8 Rx FIFO

The receive-only FIFO is enabled by asserting the RFEN bit in the MCR.

1. In previous FlexCAN versions, reading the C/S word locks the MB even if it is EMPTY. This behavior is maintained when the IRMQ bit is negated.

The reset value of this bit is zero to maintain software backward compatibility with previous versions of the module that did not have the FIFO feature. The FIFO is 6-message deep, therefore when the FIFO is enabled, the memory region occupied by the first 6 Message Buffers is reserved for use of the FIFO engine (see Rx FIFO Structure). Arm can read the received messages sequentially, in the order they were received, by repeatedly reading a Message Buffer structure at the output of the FIFO.

The IFLAG[BUF5I] (Frames available in Rx FIFO) is asserted when there is at least one frame available to be read from the FIFO. An interrupt is generated if it is enabled by the corresponding mask bit. Upon receiving the interrupt, Arm can read the message (accessing the output of the FIFO as a Message Buffer) and the RXFIR register and then clear the interrupt. If there are more messages in the FIFO the act of clearing the interrupt updates the output of the FIFO with the next message and update the RXFIR with the attributes of that message, reissuing the interrupt to Arm. Otherwise, the flag remains negated. The output of the FIFO is only valid when the IFLAG[BUF5I] is asserted.

The IFLAG[BUF6I] (Rx FIFO Warning) is asserted when the number of unread messages within the Rx FIFO is increased to 5 from 4 due to the reception of a new one, meaning that the Rx FIFO is almost full. The flag remains asserted until Arm clears it.

The IFLAG[BUF7I] (Rx FIFO Overflow) is asserted when an incoming message was lost because the Rx FIFO is full. Note that the flag will not be asserted when the Rx FIFO is full and the message was captured by a Mailbox. The flag remains asserted until the Arm clears it.

Clearing one of those three flags does not affect the state of the other two.

An interrupt is generated if an IFLAG bit is asserted and the corresponding mask bit is asserted too.

A powerful filtering scheme is provided to accept only frames intended for the target application, thus reducing the interrupt servicing work load. The filtering criteria is specified by programming a table of up to 128 32-bit registers, according to CTRL2[RFFN] setting, that can be configured to one of the following formats (see also Rx FIFO Structure):

- Format A: 128 IDAFs (extended or standard IDs including IDE and RTR)
- Format B: 256 IDAFs (standard IDs or extended 14-bit ID slices including IDE and RTR)
- Format C: 512 IDAFs (standard or extended 8-bit ID slices)

Every frame available in the FIFO has a corresponding IDHIT (Identifier Acceptance Filter Hit Indicator) that can be read by accessing the RXFIR register. The RXFIR[IDHIT] field refers to the message at the output of the FIFO and is valid whilst

the IFLAG[BUF5I] flag is asserted. The RXFIR register must be read only before clearing the flag, which guarantees that the information refers to the correct frame within the FIFO.

Up to thirty two elements of the ID Filter Table are individually affected by the Individual Mask Registers (RXIMR0 - RXIMR31), according to CTRL2[RFFN] setting (refer to Control 2 Register (FLEXCAN_CTRL2)), allowing very powerful filtering criteria to be defined. If the MCR[IRMQ] bit is negated (or if the RXIMR are not available for the particular MCU), then the FIFO ID Filter Table is affected by RXFGMASK.

27.6.9 CAN Protocol Related Features

27.6.9.1 Remote Frames

Remote frame is a special kind of frame. The user can program a mailbox to be a Remote Request Frame by writing the mailbox as Transmit with the RTR bit set to '1'. After the remote request frame is transmitted successfully, the mailbox becomes a Receive Message Buffer, with the same ID as before.

When a remote request frame is received by FlexCAN, it can be treated in three ways, depending on Remote Request Storing (CTRL2[RRS]) and Rx FIFO Enable (MCR[RFEN]) bits:

- If RRS is negated the frame's ID is compared to the IDs of the Transmit Message Buffers with the CODE field 0b1010. If there is a matching ID, then this mailbox frame will be transmitted. Note that if the matching mailbox has the RTR bit set, then FlexCAN will transmit a remote frame as a response. The received remote request frame is not stored in a receive buffer. It is only used to trigger a transmission of a frame in response. The mask registers are not used in remote frame matching, and all ID bits (except RTR) of the incoming received frame should match. In the case that a remote request frame was received and matched a mailbox, this message buffer immediately enters the internal arbitration process, but is considered as normal Tx mailbox, with no higher priority. The data length of this frame is independent of the DLC field in the remote frame that initiated its transmission.
- If RRS is asserted the frame's ID is compared to the IDs of the receive mailboxes with the CODE field 0b0100, 0b0010 or 0b0110. If there is a matching ID, then this mailbox will store the remote frame in the same fashion of a data frame. No

automatic remote response frame will be generated. The mask registers are used in the matching process.

- If RFEN is asserted FlexCAN will not generate an automatic response for remote request frames that match the FIFO filtering criteria. If the remote frame matches one of the target IDs, it will be stored in the FIFO and presented to the Arm. Note that for filtering formats A and B, it is possible to select whether remote frames are accepted or not. For format C, remote frames are always accepted (if they match the ID).Remote Request Frames are considered as normal frames, and generate a FIFO overflow when a successful reception occurs and the FIFO is already full.

27.6.9.2 Overload Frames

FLEXCAN does transmit overload frames due to detection of following conditions on CAN bus:

- Detection of a dominant bit in the first/second bit of Intermission
- Detection of a dominant bit at the 7th bit (last) of End of Frame field (Rx frames)
- Detection of a dominant bit at the 8th bit (last) of Error Frame Delimiter or Overload Frame Delimiter

27.6.9.3 Time Stamp

The value of the Free Running Timer is sampled at the beginning of the Identifier field on the CAN bus, and is stored at the end of "move-in" in the TIME STAMP field, providing network behavior with respect to time.

Note that the Free Running Timer can be reset upon a specific frame reception, enabling network time synchronization. Refer to TSYN description in Control 1 Register (FLEXCAN_CTRL1).

27.6.9.4 Protocol Timing

The FLEXCAN module supports a variety of means to setup bit timing parameters that are required by the CAN protocol. The Control Register has various fields used to control bit timing parameters: PRESDIV, PROPSEG, PSEG1, PSEG2 and RJW. See Control 1 Register (FLEXCAN_CTRL1).

The PRESDIV field controls a prescaler that generates the Serial Clock (Sclock), whose period defines the 'time quantum' used to compose the CAN waveform. A time quantum is the atomic unit of time handled by the CAN engine.

$$f_{Tq} = \frac{f_{CANCLK}}{(\text{Prescaler value})}$$

A bit time is subdivided into three segments[2] (reference Table 27-16):

- SYNC_SEG: This segment has a fixed length of one time quantum. Signal edges are expected to happen within this section.
- Time Segment 1: This segment includes the Propagation Segment and the Phase Segment 1 of the CAN standard. It can be programmed by setting the PROPSEG and the PSEG1 fields of the CTRL Register so that their sum (plus 2) is in the range of 2 to 16 time quanta.
- Time Segment 2: This segment represents the Phase Segment 2 of the CAN standard. It can be programmed by setting the PSEG2 field of the CTRL Register (plus 1) to be 2 to 8 time quanta long.

$$\text{Bit Rate} = \frac{f_{Tq}}{(\text{number of Time Quanta})}$$

2. For further explanation of the underlying concepts please refer to ISO/DIS 11519-1, Section 10.3. Reference also the Bosch CAN 2.0A/B protocol specification dated September 1991 for bit timing.

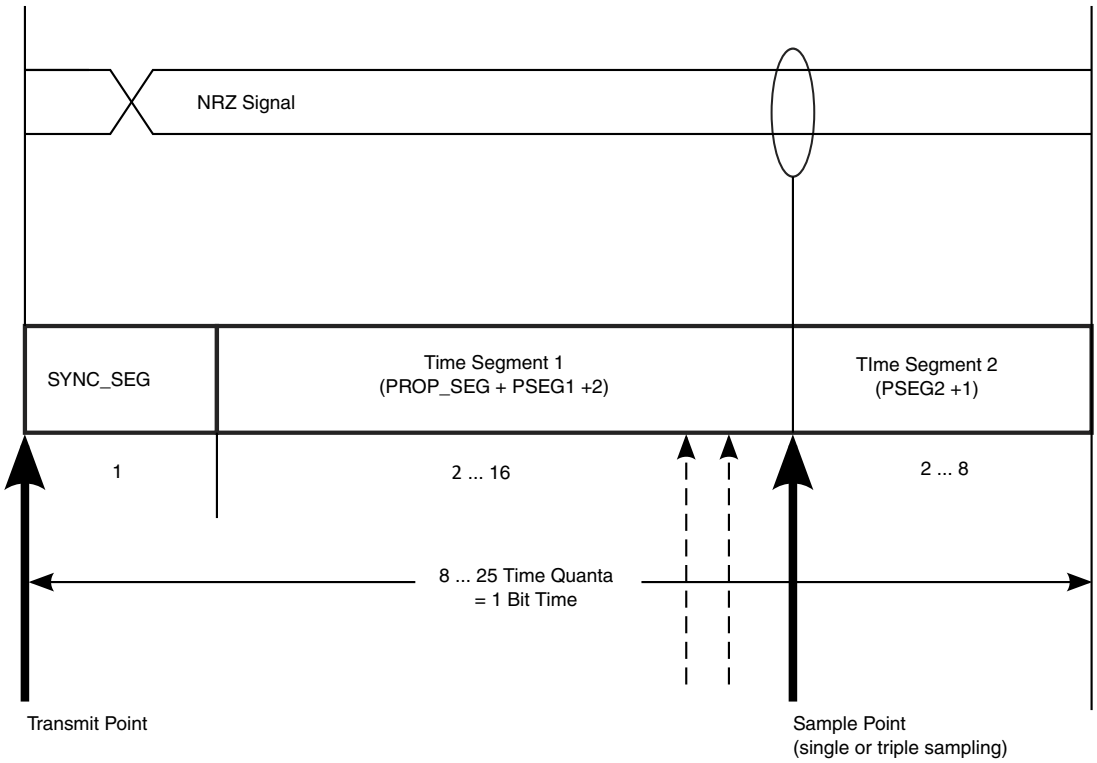


Figure 27-2. Segments within the Bit Time

Whenever CAN bit is used as a measure of duration (e.g. MCR[FRZ_ACK] and MCR[LPM_ACK] in Module Configuration Register (FLEXCAN_MCR)), the number of peripheral clocks in one CAN bit can be calculated as:

$$NCCP = \frac{f_{sys} \times [1 + (PSEG1 + 1) + (PSEG2 + 1) + (PROPSEG + 1)] \times (PRESDIV + 1)}{f_{CANCLK}}$$

where:

NCCP is the number of peripheral clocks in one CAN bit;

f_{CANCLK} is the Protocol Engine (PE) Clock in Hz;

f_{SYS} is the frequency of operation of the system (CHI) clock, in Hz;

PSEG1 is the value in CTRL1[PSEG1] field;

PSEG2 is the value in CTRL1[PSEG2] field;

PROPSEG is the value in CTRL1[PROPSEG] field;

PRESDIV is the value in CTRL1[PRESDIV] field.

For example, 180 CAN bits = 180 x NCCP peripheral clock periods.

Figure 27-2 gives an overview of the CAN compliant segment settings and the related parameter values.

Table 27-16. Time Segment Syntax

Syntax	Description
SYNC_SEG	System expects transitions to occur on the bus during this period.
Transmit Point	A node in transmit mode transfers a new value to the CAN bus at this point.
Sample Point	A node samples the bus at this point. If the three samples per bit option is selected, then this point marks the position of the third sample.

Table 27-17. CAN Standard Compliant Bit Time Segment Settings

Time Segment 1	Time Segment 2	Re-synchronization Jump Width
5 .. 10	2	1 .. 2
4 .. 11	3	1 .. 3
5 .. 12	4	1 .. 4
6 .. 13	5	1 .. 4
7 .. 14	6	1 .. 4
8 .. 15	7	1 .. 4
9 .. 16	8	1 .. 4

NOTE

The user must ensure the bit time settings are in compliance with the CAN Protocol standard (ISO 11898-1).

27.6.9.5 Arbitration and Matching Timing

During normal reception and transmission of frames, the matching, arbitration, move-in and move-out processes are executed during certain time windows inside the CAN frame, as shown in the following figures.

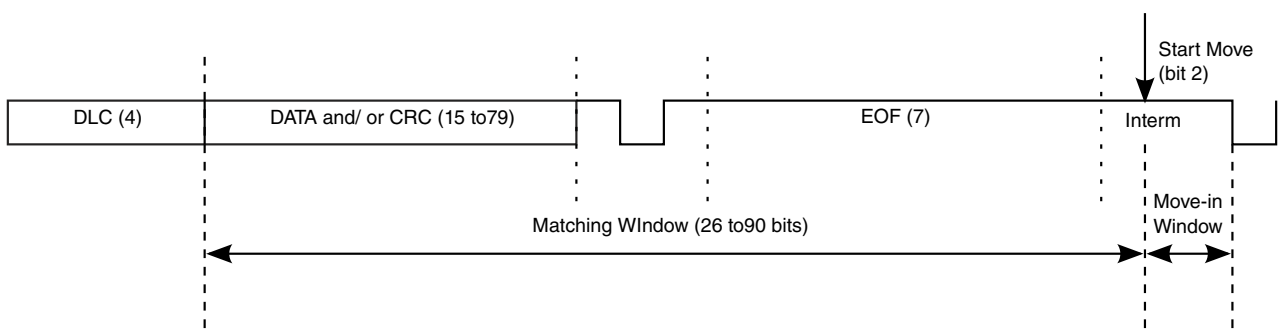


Figure 27-3. Matching and Move-In Time Windows

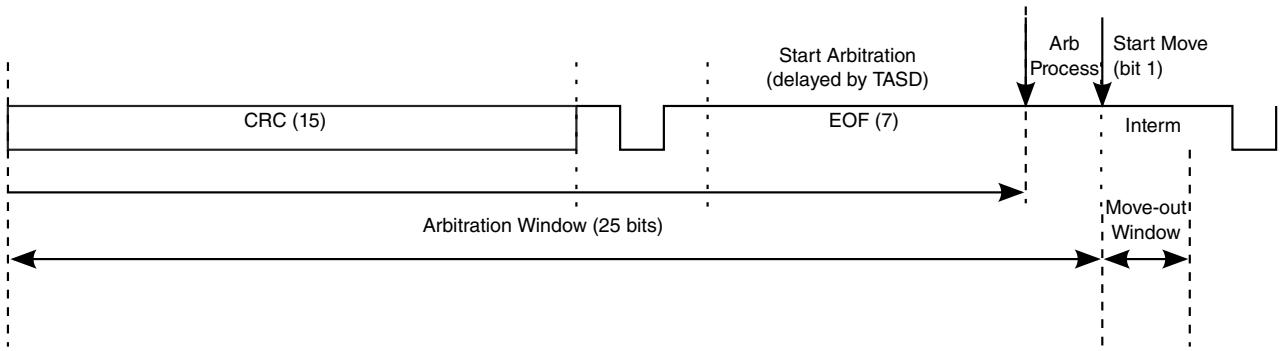


Figure 27-4. Arbitration and Move-Out Time Windows

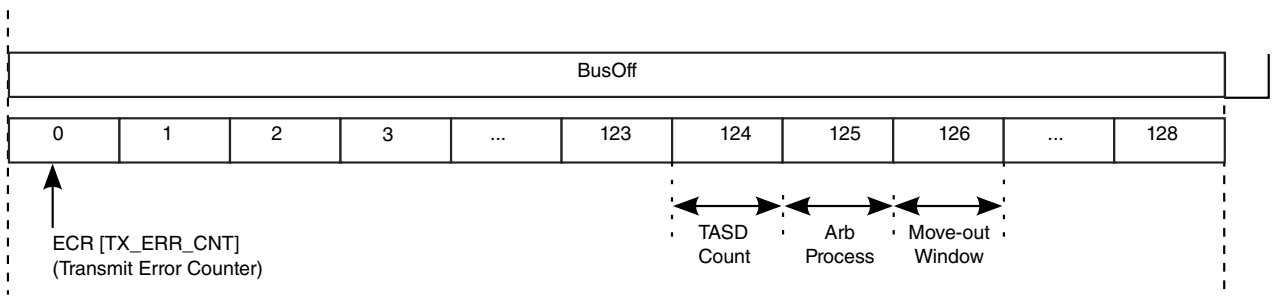


Figure 27-5. Arbitration at the end of Bus Off and Move-Out Time Windows

When doing matching and arbitration, FlexCAN needs to scan the whole Message Buffer memory during the available time window. In order to have sufficient time to do that, the following requirements must be observed:

- A valid CAN bit timing must be programmed, as indicated in Table 27-17
- The peripheral clock frequency can not be smaller than the oscillator clock frequency, i.e. the PLL can not be programmed to divide down the oscillator clock
- There must be a minimum ratio between the peripheral clock frequency and the CAN bit rate, as specified in the following table.

Table 27-18. Minimum Ratio Between Peripheral Clock Frequency and CAN Bit Rate

Number of Message Buffers	RFEN	Minimum Number of Peripheral Clocks per CAN bit
16 and 32	0	16
64	0	25
16	1	16
32	1	17
64	1	30

A direct consequence of the first requirement is that the minimum number of time quanta per CAN bit must be 8, so the oscillator clock frequency should be at least 8 times the CAN bit rate. The minimum frequency ratio specified in Table 27-18 can be achieved by choosing a high enough peripheral clock frequency when compared to the oscillator clock frequency, or by adjusting one or more of the bit timing parameters (PRESDIV, PROPSEG, PSEG1, PSEG2). As an example, taking the case of 64 MBs, if the oscillator and peripheral clock frequencies are equal and the CAN bit timing is programmed to have 8 time quanta per bit, then the prescaler factor (PRESDIV + 1) should be at least 2. For prescaler factor equal to one and CAN bit timing with 8 time quanta per bit, the ratio between peripheral and oscillator clock frequencies should be at least 2.

27.6.10 Modes of Operation Details

The FlexCAN module has four functional modes (Normal Mode, Freeze Mode, Listen-Only Mode and Loop-Back Mode) and two low power modes (Disable Mode and Stop Mode).See in Modes of Operation an introductory description of all these modes of operation. The following sub-sections bring functional details on Freeze mode and the low power modes.

27.6.10.1 Freeze Mode

This mode is requested by Arm through the assertion of the HALT bit in the MCR Register or when the MCU is put into Debug Mode . In both cases it is also necessary that the FRZ bit is asserted in the MCR Register and the module is not in any of the low power modes (Disable, Stop). The acknowledgement is obtained through the assertion by the FlexCAN of FRZ_ACK bit in the same register. The Arm must only consider the FlexCAN in Freeze Mode when both request and acknowledgement conditions are satisfied.

When Freeze Mode is requested during transmission or reception, FlexCAN does the following:

- Waits to be in either Intermission, Passive Error, Bus Off or Idle state
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish. Pending move-in is not taken in account
- Ignores the FLEXCAN_RX input pin and drives the FLEXCAN_TX pin as recessive
- Stops the prescaler, thus halting all CAN protocol activities
- Grants write access to the Error Counters Register, which is read-only in other modes
- Sets the NOT_RDY and FRZ_ACK bits in MCR

After requesting Freeze Mode, the user must wait for the FRZ_ACK bit to be asserted in MCR before executing any other action, otherwise FlexCAN may operate in an unpredictable way. In Freeze mode, all memory mapped registers are accessible, except for CTRL1[CLK_SRC] bit that can be read but cannot be written.

Exiting Freeze Mode is done in one of the following ways:

- Arm negates the FRZ bit in the MCR Register
- The Arm is removed from Debug Mode and the HALT bit is negated

The FRZ_ACK bit is negated after protocol engine recognizes the negation of freeze request. Once out of Freeze Mode, FlexCAN tries to re-synchronize to the CAN bus by waiting for 11 consecutive recessive bits.

27.6.10.2 Module Disable Mode

This low power mode is normally used to temporarily disable a complete FlexCAN block, with no power consumption. It is requested by the Arm through the assertion of the MDIS bit in the MCR Register and the acknowledgement is obtained through the assertion by the FlexCAN of the LPM_ACK bit in the same register. The Arm must only consider the FlexCAN in Disable Mode when both request and acknowledgement conditions are satisfied.

If the module is disabled during Freeze Mode, it requests to disable the clocks to the PE and CHI sub-modules, sets the LPM_ACK bit and negates the FRZ_ACK bit. The ability to shut down the clocks depends on how FlexCAN is integrated into the MCU. If the module is disabled during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and then checks it to be recessive
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish. Pending move-in is not taken in account
- Ignores its FLEXCAN_RX input pin and drives its FLEXCAN_TX pin as recessive
- May shut down the clocks to the PE and CHI sub-modules, depending on how FlexCAN is integrated into the MCU
- Sets the NOT_RDY and LPM_ACK bits in MCR

The Bus Interface Unit continues to operate, enabling the Arm to access memory mapped registers, except the Rx Mailboxes Global Mask Registers, the Rx Buffer 14 Mask Register, the Rx Buffer 15 Mask Register, the Rx FIFO Global Mask Register. The Rx FIFO Information Register, the Message Buffers, the Rx Individual Mask Registers, and the reserved words within RAM may not be accessed when the module is in Disable Mode depending on how FlexCAN RAM is integrated into the Arm. Exiting from this

mode is done by negating the MDIS bit by Arm, which make FlexCAN requests to resume the clocks and negates the LPM_ACK bit after CAN protocol engine recognizes the negation of disable mode requested by Arm.

27.6.10.3 Stop Mode

This is a system low power mode in which system clocks can be stopped for maximum power savings. To enter stop mode, the CPU should manually assert a global Stop Mode request (see the CAN1_STOP_REQ and CAN2_STOP_REQ bit in the register IOMUXC_GPR4) and check the acknowledgement asserted by the FlexCAN (see the CAN1_STOP_ACK and CAN2_STOP_ACK in the register IOMUXC_GPR4). The CPU must only consider the FlexCAN in Stop Mode when both request and acknowledgement conditions are satisfied.

If FlexCAN receives the global Stop Mode request during Freeze Mode, it sets the LPM_ACK bit, negates the FRZ_ACK bit and then sends the Stop Acknowledge signal to the CPU, in order to shut down the clocks globally. If Stop Mode is requested during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and checks it to be recessive
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish. Pending move-in is not taken in account
- Ignores its FLEXCAN_RX input pin and drives its FLEXCAN_TX pin as recessive
- Sets the NOT_RDY and LPM_ACK bits in MCR
- Sends a Stop Acknowledge signal to the CPU, so that it can shut down the clocks globally

Exiting Stop Mode is done in one of the following ways:

- Arm resuming the clocks and removing the Stop Mode request
- Arm resuming the clocks and Stop Mode request as a result of the Self Wake mechanism

In the Self Wake mechanism, if the SLF_WAK bit in MCR Register was set at the time FlexCAN entered Stop Mode, then upon detection of a recessive to dominant transition on the CAN bus, FlexCAN sets the WAK_INT bit in the ESR Register and, if enabled by the WAK_MSK bit in MCR, generates a Wake Up interrupt to the Arm. Upon receiving the interrupt, the Arm should resume the clocks and remove the Stop Mode request manually. FlexCAN will then wait for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, it will not receive the frame that woke it up .

The sensitivity to CAN bus activity can be modified by applying a low-pass filter function to the FLEXCAN_RX input line while in Stop Mode. See the WAK_SRC bit in Module Configuration Register (FLEXCAN_MCR) . This feature can be used to protect FlexCAN from waking up due to short glitches on the CAN bus lines. Such glitches can result from electromagnetic interference within noisy environments, the glitch filter width can be set in Glitch Filter Width Register (FLEXCAN_GFWR).

27.6.11 Interrupts

The module can generate up to 70 interrupt sources (64 interrupts due to message buffers and 6 interrupts due to Ored interrupts from MBs, Bus Off, Error, Tx Warning, Rx Warning and Wake Up)).

The number of actual sources depends on the configured number of message buffers.

Each one of the message buffers can be an interrupt source, if its corresponding IMASK bit is set. There is no distinction between Tx and Rx interrupts for a particular buffer, under the assumption that the buffer is initialized for either transmission or reception. Each of the buffers has assigned a flag bit in the IFLAG Registers. The bit is set when the corresponding buffer completes a successful transmission/reception and is cleared when the Arm writes it to '1' (unless another interrupt is generated at the same time).

If the Rx FIFO is enabled (bit RFEN on MCR set), the interrupts corresponding to MBs 0 to 7 have a different behavior. Bit 7 of the IFLAG1 becomes the "FIFO Overflow" flag; bit 6 becomes the FIFO Warning flag, bit 5 becomes the "Frames Available in FIFO flag" and bits 4-0 are unused. See Interrupt Flags 1 Register (FLEXCAN_IFLAG1) for more information.

A combined interrupt for all MBs is also generated by an Or of all the interrupt sources from MBs. This interrupt gets generated when any of the Mailboxes or FIFO generates an interrupt. The Arm must read the IFLAG Registers to determine which MB or FIFO caused the interrupt.

The other 5 interrupt sources (Bus Off, Error, Tx Warning, Rx Warning and Wake Up generate interrupts like the MB ones, and can be read from both the Error and Status Register 1 and 2. The Bus Off, Error, Tx Warning and Rx Warning interrupt mask bits are located in the Control 1 Register and the Wake-Up interrupt mask bit is located in the MCR.

27.7 Initialization/Application Information

This section provide instructions for initializing the FLEXCAN module.

27.7.1 FLEXCAN Initialization Sequence

The FLEXCAN module may be reset in two ways:

- SOC level hard reset which resets all memory mapped registers asynchronously
- SOFT_RST bit in MCR, which resets some of the memory mapped registers synchronously

Soft reset is synchronous and has to follow an internal request/acknowledge procedure across clock domains. Therefore, it may take some time to fully propagate its effects. The SOFT_RST bit remains asserted while soft reset is pending, so software can poll this bit to know when the reset has completed. Also, soft reset can not be applied while clocks are shut down in any of the low power modes. The low power mode should be exited and the clocks resumed before applying soft reset.

After the module is enabled (MDIS bit negated), FLEXCAN automatically goes to Freeze Mode. In Freeze Mode, FLEXCAN is un-synchronized to the CAN bus, the HALT and FRZ bits in MCR Register are set, the internal state machines are disabled and the FRZ_ACK and NOT_RDY bits in the MCR Register are set. The FLEXCAN_TX pin is in recessive state and FLEXCAN does not initiate any transmission or reception of CAN frames. Note that the Message Buffers and the Rx Individual Mask Registers are not affected by reset, so they are not automatically initialized.

For any configuration change/initialization it is required that FLEXCAN is put into Freeze Mode. The following is a generic initialization sequence applicable to the FLEXCAN module:

- Initialize the Module Configuration Register
  - Enable the individual filtering per MB and reception queue features by setting the IRMQ bit
  - Enable the warning interrupts by setting the WRN_EN bit
  - If required, disable frame self reception by setting the SRX_DIS bit
  - Enable the FIFO by setting the RFEN bit
  - Enable the abort mechanism by setting the AEN bit
  - Enable the local priority feature by setting the LPRIO_EN bit
- Initialize the Control Register
  - Determine the bit timing parameters: PROPSEG, PSEG1, PSEG2, RJW
  - Determine the bit rate by programming the PRESDIV field
  - Determine the internal arbitration mode (LBUF bit)

- Initialize the Message Buffers
  - The Control and Status word of all Message Buffers must be initialized
  - If FIFO was enabled, the 8-entry ID table must be initialized
  - Other entries in each Message Buffer should be initialized as required
- Initialize the Rx Individual Mask Registers
- Set required interrupt mask bits in the IMASK Registers (for all MB interrupts), in CTRL Register (for Bus Off and Error interrupts) and in MCR Register for Wake-Up interrupt
- Negate the HALT bit in MCR

Starting with the last event, FLEXCAN attempts to synchronize to the CAN bus.

27.8 FLEXCAN Memory Map/Register Definition

The complete memory map for a FLEXCAN module with 64 MBs capability is shown in the following table. Each individual register is identified by its complete name and the corresponding mnemonic. The access type can be Supervisor (S) or Unrestricted (U). Most of the registers can be configured to have either Supervisor or Unrestricted access by programming the SUPV bit in the MCR Register. The MCR register allows only Supervisor access regardless the SUPV bit state.

The FLEXCAN module stores CAN messages for transmission and reception using a Mailboxes and Rx FIFO structure.

FLEXCAN memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
209_0000	Module Configuration Register (FLEXCAN1_MCR)	32	R/W	5980_000Fh	27.8.1/1463
209_0004	Control 1 Register (FLEXCAN1_CTRL1)	32	R/W	0000_0000h	27.8.2/1468
209_0008	Free Running Timer Register (FLEXCAN1_TIMER)	32	R/W	0000_0000h	27.8.3/1471
209_0010	Rx Mailboxes Global Mask Register (FLEXCAN1_RXMGMASK)	32	R/W	FFFF_FFFFh	27.8.4/1471
209_0014	Rx Buffer 14 Mask Register (FLEXCAN1_RX14MASK)	32	R/W	FFFF_FFFFh	27.8.5/1472
209_0018	Rx Buffer 15 Mask Register (FLEXCAN1_RX15MASK)	32	R/W	FFFF_FFFFh	27.8.6/1473
209_001C	Error Counter Register (FLEXCAN1_ECR)	32	R/W	0000_0000h	27.8.7/1474
209_0020	Error and Status 1 Register (FLEXCAN1_ESR1)	32	R/W	0000_0000h	27.8.8/1475
209_0024	Interrupt Masks 2 Register (FLEXCAN1_IMASK2)	32	R/W	0000_0000h	27.8.9/1479
209_0028	Interrupt Masks 1 Register (FLEXCAN1_IMASK1)	32	R/W	0000_0000h	27.8.10/ 1479

Table continues on the next page...

FLEXCAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
209_002C	Interrupt Flags 2 Register (FLEXCAN1_IFLAG2)	32	R/W	0000_0000h	27.8.11/ 1480
209_0030	Interrupt Flags 1 Register (FLEXCAN1_IFLAG1)	32	R/W	0000_0000h	27.8.12/ 1480
209_0034	Control 2 Register (FLEXCAN1_CTRL2)	32	R/W	0000_0000h	27.8.13/ 1482
209_0038	Error and Status 2 Register (FLEXCAN1_ESR2)	32	R	0000_0000h	27.8.14/ 1488
209_0044	CRC Register (FLEXCAN1_CRCR)	32	R	0000_0000h	27.8.15/ 1490
209_0048	Rx FIFO Global Mask Register (FLEXCAN1_RXFGMASK)	32	R/W	FFFF_FFFFh	27.8.16/ 1491
209_004C	Rx FIFO Information Register (FLEXCAN1_RXFIR)	32	R	0000_0000h	27.8.17/ 1492
209_0880	Rx Individual Mask Registers (FLEXCAN1_RXIMR0)	32	R/W	0000_0000h	27.8.18/ 1493
209_0884	Rx Individual Mask Registers (FLEXCAN1_RXIMR1)	32	R/W	0000_0000h	27.8.18/ 1493
209_0888	Rx Individual Mask Registers (FLEXCAN1_RXIMR2)	32	R/W	0000_0000h	27.8.18/ 1493
209_088C	Rx Individual Mask Registers (FLEXCAN1_RXIMR3)	32	R/W	0000_0000h	27.8.18/ 1493
209_0890	Rx Individual Mask Registers (FLEXCAN1_RXIMR4)	32	R/W	0000_0000h	27.8.18/ 1493
209_0894	Rx Individual Mask Registers (FLEXCAN1_RXIMR5)	32	R/W	0000_0000h	27.8.18/ 1493
209_0898	Rx Individual Mask Registers (FLEXCAN1_RXIMR6)	32	R/W	0000_0000h	27.8.18/ 1493
209_089C	Rx Individual Mask Registers (FLEXCAN1_RXIMR7)	32	R/W	0000_0000h	27.8.18/ 1493
209_08A0	Rx Individual Mask Registers (FLEXCAN1_RXIMR8)	32	R/W	0000_0000h	27.8.18/ 1493
209_08A4	Rx Individual Mask Registers (FLEXCAN1_RXIMR9)	32	R/W	0000_0000h	27.8.18/ 1493
209_08A8	Rx Individual Mask Registers (FLEXCAN1_RXIMR10)	32	R/W	0000_0000h	27.8.18/ 1493
209_08AC	Rx Individual Mask Registers (FLEXCAN1_RXIMR11)	32	R/W	0000_0000h	27.8.18/ 1493
209_08B0	Rx Individual Mask Registers (FLEXCAN1_RXIMR12)	32	R/W	0000_0000h	27.8.18/ 1493
209_08B4	Rx Individual Mask Registers (FLEXCAN1_RXIMR13)	32	R/W	0000_0000h	27.8.18/ 1493
209_08B8	Rx Individual Mask Registers (FLEXCAN1_RXIMR14)	32	R/W	0000_0000h	27.8.18/ 1493

Table continues on the next page...

FLEXCAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
209_08BC	Rx Individual Mask Registers (FLEXCAN1_RXIMR15)	32	R/W	0000_0000h	27.8.18/ 1493
209_08C0	Rx Individual Mask Registers (FLEXCAN1_RXIMR16)	32	R/W	0000_0000h	27.8.18/ 1493
209_08C4	Rx Individual Mask Registers (FLEXCAN1_RXIMR17)	32	R/W	0000_0000h	27.8.18/ 1493
209_08C8	Rx Individual Mask Registers (FLEXCAN1_RXIMR18)	32	R/W	0000_0000h	27.8.18/ 1493
209_08CC	Rx Individual Mask Registers (FLEXCAN1_RXIMR19)	32	R/W	0000_0000h	27.8.18/ 1493
209_08D0	Rx Individual Mask Registers (FLEXCAN1_RXIMR20)	32	R/W	0000_0000h	27.8.18/ 1493
209_08D4	Rx Individual Mask Registers (FLEXCAN1_RXIMR21)	32	R/W	0000_0000h	27.8.18/ 1493
209_08D8	Rx Individual Mask Registers (FLEXCAN1_RXIMR22)	32	R/W	0000_0000h	27.8.18/ 1493
209_08DC	Rx Individual Mask Registers (FLEXCAN1_RXIMR23)	32	R/W	0000_0000h	27.8.18/ 1493
209_08E0	Rx Individual Mask Registers (FLEXCAN1_RXIMR24)	32	R/W	0000_0000h	27.8.18/ 1493
209_08E4	Rx Individual Mask Registers (FLEXCAN1_RXIMR25)	32	R/W	0000_0000h	27.8.18/ 1493
209_08E8	Rx Individual Mask Registers (FLEXCAN1_RXIMR26)	32	R/W	0000_0000h	27.8.18/ 1493
209_08EC	Rx Individual Mask Registers (FLEXCAN1_RXIMR27)	32	R/W	0000_0000h	27.8.18/ 1493
209_08F0	Rx Individual Mask Registers (FLEXCAN1_RXIMR28)	32	R/W	0000_0000h	27.8.18/ 1493
209_08F4	Rx Individual Mask Registers (FLEXCAN1_RXIMR29)	32	R/W	0000_0000h	27.8.18/ 1493
209_08F8	Rx Individual Mask Registers (FLEXCAN1_RXIMR30)	32	R/W	0000_0000h	27.8.18/ 1493
209_08FC	Rx Individual Mask Registers (FLEXCAN1_RXIMR31)	32	R/W	0000_0000h	27.8.18/ 1493
209_0900	Rx Individual Mask Registers (FLEXCAN1_RXIMR32)	32	R/W	0000_0000h	27.8.18/ 1493
209_0904	Rx Individual Mask Registers (FLEXCAN1_RXIMR33)	32	R/W	0000_0000h	27.8.18/ 1493
209_0908	Rx Individual Mask Registers (FLEXCAN1_RXIMR34)	32	R/W	0000_0000h	27.8.18/ 1493
209_090C	Rx Individual Mask Registers (FLEXCAN1_RXIMR35)	32	R/W	0000_0000h	27.8.18/ 1493
209_0910	Rx Individual Mask Registers (FLEXCAN1_RXIMR36)	32	R/W	0000_0000h	27.8.18/ 1493

Table continues on the next page...

FLEXCAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
209_0914	Rx Individual Mask Registers (FLEXCAN1_RXIMR37)	32	R/W	0000_0000h	27.8.18/ 1493
209_0918	Rx Individual Mask Registers (FLEXCAN1_RXIMR38)	32	R/W	0000_0000h	27.8.18/ 1493
209_091C	Rx Individual Mask Registers (FLEXCAN1_RXIMR39)	32	R/W	0000_0000h	27.8.18/ 1493
209_0920	Rx Individual Mask Registers (FLEXCAN1_RXIMR40)	32	R/W	0000_0000h	27.8.18/ 1493
209_0924	Rx Individual Mask Registers (FLEXCAN1_RXIMR41)	32	R/W	0000_0000h	27.8.18/ 1493
209_0928	Rx Individual Mask Registers (FLEXCAN1_RXIMR42)	32	R/W	0000_0000h	27.8.18/ 1493
209_092C	Rx Individual Mask Registers (FLEXCAN1_RXIMR43)	32	R/W	0000_0000h	27.8.18/ 1493
209_0930	Rx Individual Mask Registers (FLEXCAN1_RXIMR44)	32	R/W	0000_0000h	27.8.18/ 1493
209_0934	Rx Individual Mask Registers (FLEXCAN1_RXIMR45)	32	R/W	0000_0000h	27.8.18/ 1493
209_0938	Rx Individual Mask Registers (FLEXCAN1_RXIMR46)	32	R/W	0000_0000h	27.8.18/ 1493
209_093C	Rx Individual Mask Registers (FLEXCAN1_RXIMR47)	32	R/W	0000_0000h	27.8.18/ 1493
209_0940	Rx Individual Mask Registers (FLEXCAN1_RXIMR48)	32	R/W	0000_0000h	27.8.18/ 1493
209_0944	Rx Individual Mask Registers (FLEXCAN1_RXIMR49)	32	R/W	0000_0000h	27.8.18/ 1493
209_0948	Rx Individual Mask Registers (FLEXCAN1_RXIMR50)	32	R/W	0000_0000h	27.8.18/ 1493
209_094C	Rx Individual Mask Registers (FLEXCAN1_RXIMR51)	32	R/W	0000_0000h	27.8.18/ 1493
209_0950	Rx Individual Mask Registers (FLEXCAN1_RXIMR52)	32	R/W	0000_0000h	27.8.18/ 1493
209_0954	Rx Individual Mask Registers (FLEXCAN1_RXIMR53)	32	R/W	0000_0000h	27.8.18/ 1493
209_0958	Rx Individual Mask Registers (FLEXCAN1_RXIMR54)	32	R/W	0000_0000h	27.8.18/ 1493
209_095C	Rx Individual Mask Registers (FLEXCAN1_RXIMR55)	32	R/W	0000_0000h	27.8.18/ 1493
209_0960	Rx Individual Mask Registers (FLEXCAN1_RXIMR56)	32	R/W	0000_0000h	27.8.18/ 1493
209_0964	Rx Individual Mask Registers (FLEXCAN1_RXIMR57)	32	R/W	0000_0000h	27.8.18/ 1493
209_0968	Rx Individual Mask Registers (FLEXCAN1_RXIMR58)	32	R/W	0000_0000h	27.8.18/ 1493

Table continues on the next page...

FLEXCAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
209_096C	Rx Individual Mask Registers (FLEXCAN1_RXIMR59)	32	R/W	0000_0000h	27.8.18/ 1493
209_0970	Rx Individual Mask Registers (FLEXCAN1_RXIMR60)	32	R/W	0000_0000h	27.8.18/ 1493
209_0974	Rx Individual Mask Registers (FLEXCAN1_RXIMR61)	32	R/W	0000_0000h	27.8.18/ 1493
209_0978	Rx Individual Mask Registers (FLEXCAN1_RXIMR62)	32	R/W	0000_0000h	27.8.18/ 1493
209_097C	Rx Individual Mask Registers (FLEXCAN1_RXIMR63)	32	R/W	0000_0000h	27.8.18/ 1493
209_09E0	Glitch Filter Width Registers (FLEXCAN1_GFWR)	32	R/W	0000_007Fh	27.8.19/ 1493
209_4000	Module Configuration Register (FLEXCAN2_MCR)	32	R/W	5980_000Fh	27.8.1/1463
209_4004	Control 1 Register (FLEXCAN2_CTRL1)	32	R/W	0000_0000h	27.8.2/1468
209_4008	Free Running Timer Register (FLEXCAN2_TIMER)	32	R/W	0000_0000h	27.8.3/1471
209_4010	Rx Mailboxes Global Mask Register (FLEXCAN2_RXMGMASK)	32	R/W	FFFF_FFFFh	27.8.4/1471
209_4014	Rx Buffer 14 Mask Register (FLEXCAN2_RX14MASK)	32	R/W	FFFF_FFFFh	27.8.5/1472
209_4018	Rx Buffer 15 Mask Register (FLEXCAN2_RX15MASK)	32	R/W	FFFF_FFFFh	27.8.6/1473
209_401C	Error Counter Register (FLEXCAN2_ECR)	32	R/W	0000_0000h	27.8.7/1474
209_4020	Error and Status 1 Register (FLEXCAN2_ESR1)	32	R/W	0000_0000h	27.8.8/1475
209_4024	Interrupt Masks 2 Register (FLEXCAN2_IMASK2)	32	R/W	0000_0000h	27.8.9/1479
209_4028	Interrupt Masks 1 Register (FLEXCAN2_IMASK1)	32	R/W	0000_0000h	27.8.10/ 1479
209_402C	Interrupt Flags 2 Register (FLEXCAN2_IFLAG2)	32	R/W	0000_0000h	27.8.11/ 1480
209_4030	Interrupt Flags 1 Register (FLEXCAN2_IFLAG1)	32	R/W	0000_0000h	27.8.12/ 1480
209_4034	Control 2 Register (FLEXCAN2_CTRL2)	32	R/W	0000_0000h	27.8.13/ 1482
209_4038	Error and Status 2 Register (FLEXCAN2_ESR2)	32	R	0000_0000h	27.8.14/ 1488
209_4044	CRC Register (FLEXCAN2_CRCR)	32	R	0000_0000h	27.8.15/ 1490
209_4048	Rx FIFO Global Mask Register (FLEXCAN2_RXFGMASK)	32	R/W	FFFF_FFFFh	27.8.16/ 1491
209_404C	Rx FIFO Information Register (FLEXCAN2_RXFIR)	32	R	0000_0000h	27.8.17/ 1492
209_4880	Rx Individual Mask Registers (FLEXCAN2_RXIMR0)	32	R/W	0000_0000h	27.8.18/ 1493
209_4884	Rx Individual Mask Registers (FLEXCAN2_RXIMR1)	32	R/W	0000_0000h	27.8.18/ 1493

Table continues on the next page...

FLEXCAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
209_4888	Rx Individual Mask Registers (FLEXCAN2_RXIMR2)	32	R/W	0000_0000h	27.8.18/ 1493
209_488C	Rx Individual Mask Registers (FLEXCAN2_RXIMR3)	32	R/W	0000_0000h	27.8.18/ 1493
209_4890	Rx Individual Mask Registers (FLEXCAN2_RXIMR4)	32	R/W	0000_0000h	27.8.18/ 1493
209_4894	Rx Individual Mask Registers (FLEXCAN2_RXIMR5)	32	R/W	0000_0000h	27.8.18/ 1493
209_4898	Rx Individual Mask Registers (FLEXCAN2_RXIMR6)	32	R/W	0000_0000h	27.8.18/ 1493
209_489C	Rx Individual Mask Registers (FLEXCAN2_RXIMR7)	32	R/W	0000_0000h	27.8.18/ 1493
209_48A0	Rx Individual Mask Registers (FLEXCAN2_RXIMR8)	32	R/W	0000_0000h	27.8.18/ 1493
209_48A4	Rx Individual Mask Registers (FLEXCAN2_RXIMR9)	32	R/W	0000_0000h	27.8.18/ 1493
209_48A8	Rx Individual Mask Registers (FLEXCAN2_RXIMR10)	32	R/W	0000_0000h	27.8.18/ 1493
209_48AC	Rx Individual Mask Registers (FLEXCAN2_RXIMR11)	32	R/W	0000_0000h	27.8.18/ 1493
209_48B0	Rx Individual Mask Registers (FLEXCAN2_RXIMR12)	32	R/W	0000_0000h	27.8.18/ 1493
209_48B4	Rx Individual Mask Registers (FLEXCAN2_RXIMR13)	32	R/W	0000_0000h	27.8.18/ 1493
209_48B8	Rx Individual Mask Registers (FLEXCAN2_RXIMR14)	32	R/W	0000_0000h	27.8.18/ 1493
209_48BC	Rx Individual Mask Registers (FLEXCAN2_RXIMR15)	32	R/W	0000_0000h	27.8.18/ 1493
209_48C0	Rx Individual Mask Registers (FLEXCAN2_RXIMR16)	32	R/W	0000_0000h	27.8.18/ 1493
209_48C4	Rx Individual Mask Registers (FLEXCAN2_RXIMR17)	32	R/W	0000_0000h	27.8.18/ 1493
209_48C8	Rx Individual Mask Registers (FLEXCAN2_RXIMR18)	32	R/W	0000_0000h	27.8.18/ 1493
209_48CC	Rx Individual Mask Registers (FLEXCAN2_RXIMR19)	32	R/W	0000_0000h	27.8.18/ 1493
209_48D0	Rx Individual Mask Registers (FLEXCAN2_RXIMR20)	32	R/W	0000_0000h	27.8.18/ 1493
209_48D4	Rx Individual Mask Registers (FLEXCAN2_RXIMR21)	32	R/W	0000_0000h	27.8.18/ 1493
209_48D8	Rx Individual Mask Registers (FLEXCAN2_RXIMR22)	32	R/W	0000_0000h	27.8.18/ 1493
209_48DC	Rx Individual Mask Registers (FLEXCAN2_RXIMR23)	32	R/W	0000_0000h	27.8.18/ 1493

Table continues on the next page...

FLEXCAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
209_48E0	Rx Individual Mask Registers (FLEXCAN2_RXIMR24)	32	R/W	0000_0000h	27.8.18/ 1493
209_48E4	Rx Individual Mask Registers (FLEXCAN2_RXIMR25)	32	R/W	0000_0000h	27.8.18/ 1493
209_48E8	Rx Individual Mask Registers (FLEXCAN2_RXIMR26)	32	R/W	0000_0000h	27.8.18/ 1493
209_48EC	Rx Individual Mask Registers (FLEXCAN2_RXIMR27)	32	R/W	0000_0000h	27.8.18/ 1493
209_48F0	Rx Individual Mask Registers (FLEXCAN2_RXIMR28)	32	R/W	0000_0000h	27.8.18/ 1493
209_48F4	Rx Individual Mask Registers (FLEXCAN2_RXIMR29)	32	R/W	0000_0000h	27.8.18/ 1493
209_48F8	Rx Individual Mask Registers (FLEXCAN2_RXIMR30)	32	R/W	0000_0000h	27.8.18/ 1493
209_48FC	Rx Individual Mask Registers (FLEXCAN2_RXIMR31)	32	R/W	0000_0000h	27.8.18/ 1493
209_4900	Rx Individual Mask Registers (FLEXCAN2_RXIMR32)	32	R/W	0000_0000h	27.8.18/ 1493
209_4904	Rx Individual Mask Registers (FLEXCAN2_RXIMR33)	32	R/W	0000_0000h	27.8.18/ 1493
209_4908	Rx Individual Mask Registers (FLEXCAN2_RXIMR34)	32	R/W	0000_0000h	27.8.18/ 1493
209_490C	Rx Individual Mask Registers (FLEXCAN2_RXIMR35)	32	R/W	0000_0000h	27.8.18/ 1493
209_4910	Rx Individual Mask Registers (FLEXCAN2_RXIMR36)	32	R/W	0000_0000h	27.8.18/ 1493
209_4914	Rx Individual Mask Registers (FLEXCAN2_RXIMR37)	32	R/W	0000_0000h	27.8.18/ 1493
209_4918	Rx Individual Mask Registers (FLEXCAN2_RXIMR38)	32	R/W	0000_0000h	27.8.18/ 1493
209_491C	Rx Individual Mask Registers (FLEXCAN2_RXIMR39)	32	R/W	0000_0000h	27.8.18/ 1493
209_4920	Rx Individual Mask Registers (FLEXCAN2_RXIMR40)	32	R/W	0000_0000h	27.8.18/ 1493
209_4924	Rx Individual Mask Registers (FLEXCAN2_RXIMR41)	32	R/W	0000_0000h	27.8.18/ 1493
209_4928	Rx Individual Mask Registers (FLEXCAN2_RXIMR42)	32	R/W	0000_0000h	27.8.18/ 1493
209_492C	Rx Individual Mask Registers (FLEXCAN2_RXIMR43)	32	R/W	0000_0000h	27.8.18/ 1493
209_4930	Rx Individual Mask Registers (FLEXCAN2_RXIMR44)	32	R/W	0000_0000h	27.8.18/ 1493
209_4934	Rx Individual Mask Registers (FLEXCAN2_RXIMR45)	32	R/W	0000_0000h	27.8.18/ 1493

Table continues on the next page...

FLEXCAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
209_4938	Rx Individual Mask Registers (FLEXCAN2_RXIMR46)	32	R/W	0000_0000h	27.8.18/ 1493
209_493C	Rx Individual Mask Registers (FLEXCAN2_RXIMR47)	32	R/W	0000_0000h	27.8.18/ 1493
209_4940	Rx Individual Mask Registers (FLEXCAN2_RXIMR48)	32	R/W	0000_0000h	27.8.18/ 1493
209_4944	Rx Individual Mask Registers (FLEXCAN2_RXIMR49)	32	R/W	0000_0000h	27.8.18/ 1493
209_4948	Rx Individual Mask Registers (FLEXCAN2_RXIMR50)	32	R/W	0000_0000h	27.8.18/ 1493
209_494C	Rx Individual Mask Registers (FLEXCAN2_RXIMR51)	32	R/W	0000_0000h	27.8.18/ 1493
209_4950	Rx Individual Mask Registers (FLEXCAN2_RXIMR52)	32	R/W	0000_0000h	27.8.18/ 1493
209_4954	Rx Individual Mask Registers (FLEXCAN2_RXIMR53)	32	R/W	0000_0000h	27.8.18/ 1493
209_4958	Rx Individual Mask Registers (FLEXCAN2_RXIMR54)	32	R/W	0000_0000h	27.8.18/ 1493
209_495C	Rx Individual Mask Registers (FLEXCAN2_RXIMR55)	32	R/W	0000_0000h	27.8.18/ 1493
209_4960	Rx Individual Mask Registers (FLEXCAN2_RXIMR56)	32	R/W	0000_0000h	27.8.18/ 1493
209_4964	Rx Individual Mask Registers (FLEXCAN2_RXIMR57)	32	R/W	0000_0000h	27.8.18/ 1493
209_4968	Rx Individual Mask Registers (FLEXCAN2_RXIMR58)	32	R/W	0000_0000h	27.8.18/ 1493
209_496C	Rx Individual Mask Registers (FLEXCAN2_RXIMR59)	32	R/W	0000_0000h	27.8.18/ 1493
209_4970	Rx Individual Mask Registers (FLEXCAN2_RXIMR60)	32	R/W	0000_0000h	27.8.18/ 1493
209_4974	Rx Individual Mask Registers (FLEXCAN2_RXIMR61)	32	R/W	0000_0000h	27.8.18/ 1493
209_4978	Rx Individual Mask Registers (FLEXCAN2_RXIMR62)	32	R/W	0000_0000h	27.8.18/ 1493
209_497C	Rx Individual Mask Registers (FLEXCAN2_RXIMR63)	32	R/W	0000_0000h	27.8.18/ 1493
209_49E0	Glitch Filter Width Registers (FLEXCAN2_GFWR)	32	R/W	0000_007Fh	27.8.19/ 1493

27.8.1 Module Configuration Register (FLEXCANx_MCR)

This register defines global system configurations, such as the module operation mode (e.g., low power) and maximum message buffer configuration.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MDIS	FRZ	RFEN	HALT	NOT_RDY	WAK_MSK	SOFT_RST	FRZ_ACK	SUPV	SLF_WAK	WRN_EN	LPM_ACK	WAK_SRC	Reserved	SRX_DIS	IRMQ
W																
Reset	0	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		LPRIO_EN	AEN	Reserved		IDAM		Reserved	MAXMB						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

FLEXCANx_MCR field descriptions

Field	Description
31 MDIS	This bit controls whether FLEXCAN is enabled or not. When disabled, FLEXCAN shuts down the clocks to the CAN Protocol Interface and Message Buffer Management sub-modules. This is the only bit in MCR not affected by soft reset. See Module Disable Mode for more information. 1 Disable the FLEXCAN module 0 Enable the FLEXCAN module
30 FRZ	The FRZ bit specifies the FLEXCAN behavior when the HALT bit in the MCR Register is set or when Debug Mode is requested at Arm level. When FRZ is asserted, FLEXCAN is enabled to enter Freeze Mode. Negation of this bit field causes FLEXCAN to exit from Freeze Mode. 1 Enabled to enter Freeze Mode 0 Not enabled to enter Freeze Mode
29 RFEN	This bit controls whether the Rx FIFO feature is enabled or not. When RFEN is set, MBs 0 to 5 cannot be used for normal reception and transmission because the corresponding memory region (0x80-0xDC) is used by the FIFO engine as well as additional MBs (up to 32, depending on CTRL2[RFFN] setting) which are used as Rx FIFO ID Filter Table elements.RFEN also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in Table 27-18 (see Arbitration and Matching Timing).This bit can only be written in Freeze mode as it is blocked by hardware in other modes. 1 FIFO enabled 0 FIFO not enabled
28 HALT	Assertion of this bit puts the FLEXCAN module into Freeze Mode. The Arm should clear it after initializing the Message Buffers and Control Register. No reception or transmission is performed by FLEXCAN before this bit is cleared. Freeze Mode can not be entered while FLEXCAN is in any of the low power modes.See Freeze Mode for more information 1 Enters Freeze Mode if the FRZ bit is asserted. 0 No Freeze Mode request.
27 NOT_RDY	This read-only bit indicates that FLEXCAN is either in Disable Mode, Stop Mode or Freeze Mode. It is negated once FLEXCAN has exited these modes. 1 FLEXCAN module is either in Disable Mode, Stop Mode or Freeze Mode 0 FLEXCAN module is either in Normal Mode, Listen-Only Mode or Loop-Back Mode
26 WAK_MSK	This bit enables the Wake Up Interrupt generation. 1 Wake Up Interrupt is enabled 0 Wake Up Interrupt is disabled
25 SOFT_RST	When this bit is asserted, FlexCAN resets its internal state machines and some of the memory mapped registers. The following registers are reset: MCR (except the MDIS bit), TIMER, ECR, ESR1, ESR2, IMASK1, IMASK2, IFLAG1, IFLAG2 and CRCR. Configuration registers that control the interface to the CAN bus are not affected by soft reset. The following registers are unaffected: CTRL1, CTRL2, RXIMR0_RXIMR63, RXGMASK, RX14MASK, RX15MASK, RXFGMASK, RXFIR and all Message Buffers The SOFT_RST bit can be asserted directly by the Arm when it writes to the MCR Register. It may take some time to fully propagate its effect. The SOFT_RST bit remains asserted while reset is pending, and is automatically negated when reset completes. Therefore, software can poll this bit to know when the soft reset has completed. Soft reset cannot be applied while clocks are shut down in any of the low power modes. The module should be first removed from low power mode, and then soft reset can be applied.

Table continues on the next page...

FLEXCANx_MCR field descriptions (continued)

Field	Description
	1 Reset the registers 0 No reset request
24 FRZ_ACK	This read-only bit indicates that FLEXCAN is in Freeze Mode and its prescaler is stopped. The Freeze Mode request cannot be granted until current transmission or reception processes have finished. Therefore the software can poll the FRZ_ACK bit to know when FLEXCAN has actually entered Freeze Mode. If Freeze Mode request is negated, then this bit is negated once the FLEXCAN prescaler is running again. If Freeze Mode is requested while FLEXCAN is in any of the low power modes, then the FRZ_ACK bit will only be set when the low power mode is exited. See Freeze Mode for more information 1 FLEXCAN in Freeze Mode, prescaler stopped 0 FLEXCAN not in Freeze Mode, prescaler running
23 SUPV	This bit configures some of the FLEXCAN registers to be either in Supervisor or User Mode. Reset value of this bit is '1', so the affected registers start with Supervisor access allowance only. This bit can only be written in Freeze mode as it is blocked by hardware in other modes. 1 FlexCAN is in Supervisor Mode. Affected registers allow only Supervisor access. Unrestricted access behaves as though the access was done to an unimplemented register location 0 FlexCAN is in User Mode. Affected registers allow both Supervisor and Unrestricted accesses
22 SLF_WAK	This bit enables the Self Wake Up feature when FLEXCAN is in Stop Mode. If this bit had been asserted by the time FLEXCAN entered Stop Mode, then FLEXCAN will look for a recessive to dominant transition on the bus during these modes. If a transition from recessive to dominant is detected during Stop Mode, then FLEXCAN generates, if enabled to do so, a Wake Up interrupt to the Arm so that it can resume the clocks globally and FlexCAN can request to resume the clocks. This bit can not be written while the module is in Stop Mode. 1 FLEXCAN Self Wake Up feature is enabled 0 FLEXCAN Self Wake Up feature is disabled
21 WRN_EN	When asserted, this bit enables the generation of the TWRN_INT and RWRN_INT flags in the Error and Status Register. If WRN_EN is negated, the TWRN_INT and RWRN_INT flags will always be zero, independent of the values of the error counters, and no warning interrupt will ever be generated.This bit can only be written in Freeze mode as it is blocked by hardware in other modes. 1 TWRN_INT and RWRN_INT bits are set when the respective error counter transition from <96 to ≥ 96. 0 TWRN_INT and RWRN_INT bits are zero, independent of the values in the error counters.
20 LPM_ACK	This read-only bit indicates that FLEXCAN is either in Disable Mode or Stop Mode. Either of these low power modes can not be entered until all current transmission or reception processes have finished, so the Arm can poll the LPM_ACK bit to know when FLEXCAN has actually entered low power mode. See Module Disable Mode, and Stop Mode for more information 1 FLEXCAN is either in Disable Mode, or Stop mode 0 FLEXCAN not in any of the low power modes
19 WAK_SRC	This bit defines whether the integrated low-pass filter is applied to protect the FLEXCAN_RX input from spurious wake up. See Stop Mode for more information. This bit can only be written in Freeze mode as it is blocked by hardware in other modes. 1 FLEXCAN uses the filtered FLEXCAN_RX input to detect recessive to dominant edges on the CAN bus 0 FLEXCAN uses the unfiltered FLEXCAN_RX input to detect recessive to dominant edges on the CAN bus.
18 -	This field is reserved. Reserved

Table continues on the next page...

FLEXCANx_MCR field descriptions (continued)

Field	Description
17 SRX_DIS	This bit defines whether FlexCAN is allowed to receive frames transmitted by itself. If this bit is asserted, frames transmitted by the module will not be stored in any MB, regardless if the MB is programmed with an ID that matches the transmitted frame, and no interrupt flag or interrupt signal will be generated due to the frame reception. This bit can only be written in Freeze mode as it is blocked by hardware in other modes. 1 Self reception disabled 0 Self reception enabled
16 IRMQ	This bit indicates whether Rx matching process will be based either on individual masking and queue or on masking scheme with RXMGMASK, RX14MASK and RX15MASK, RXFGMASK. This bit can only be written in Freeze mode as it is blocked by hardware in other modes. 1 Individual Rx masking and queue feature are enabled. 0 Individual Rx masking and queue feature are disabled.For backward compatibility, the reading of C/S word locks the MB even if it is EMPTY.
15–14 -	This field is reserved. Reserved
13 LPRIO_EN	This bit is provided for backwards compatibility reasons. It controls whether the local priority feature is enabled or not. It is used to extend the ID used during the arbitration process. With this extended ID concept, the arbitration process is done based on the full 32-bit word, but the actual transmitted ID still has 11-bit for standard frames and 29-bit for extended frames.This bit can only be written in Freeze mode as it is blocked by hardware in other modes. 1 Local Priority enabled 0 Local Priority disabled
12 AEN	This bit is supplied for backwards compatibility reasons. When asserted, it enables the Tx abort feature. This feature guarantees a safe procedure for aborting a pending transmission, so that no frame is sent in the CAN bus without notification. This bit can only be written in Freeze mode as it is blocked by hardware in other modes.Write Abort code into Rx Mailboxes can cause unpredictable results when the MCR[AEN] is asserted. 1 Abort enabled 0 Abort disabled
11–10 -	This field is reserved. Reserved
9–8 IDAM	This 2-bit field identifies the format of the elements of the Rx FIFO filter table, as shown below. Note that all elements of the table are configured at the same time by this field (they are all the same format). See Rx FIFO Structure. This bit can only be written in Freeze mode as it is blocked by hardware in other modes. 00 Format A One full ID (standard or extended) per ID filter Table element. 01 Format B Two full standard IDs or two partial 14-bit extended IDs per ID filter Table element. 10 Format C Four partial 8-bit IDs (standard or extended) per ID filter Table element. 11 Format D All frames rejected.
7 -	This field is reserved. Reserved
MAXMB	This 7-bit field defines the number of the last Message Buffers that will take part in the matching and arbitration processes. The reset value (0x0F) is equivalent to 16 MB configuration. This field can only be written in Freeze Mode as it is blocked by hardware in other modes Number of the last MB = MAXMB.

Table continues on the next page...

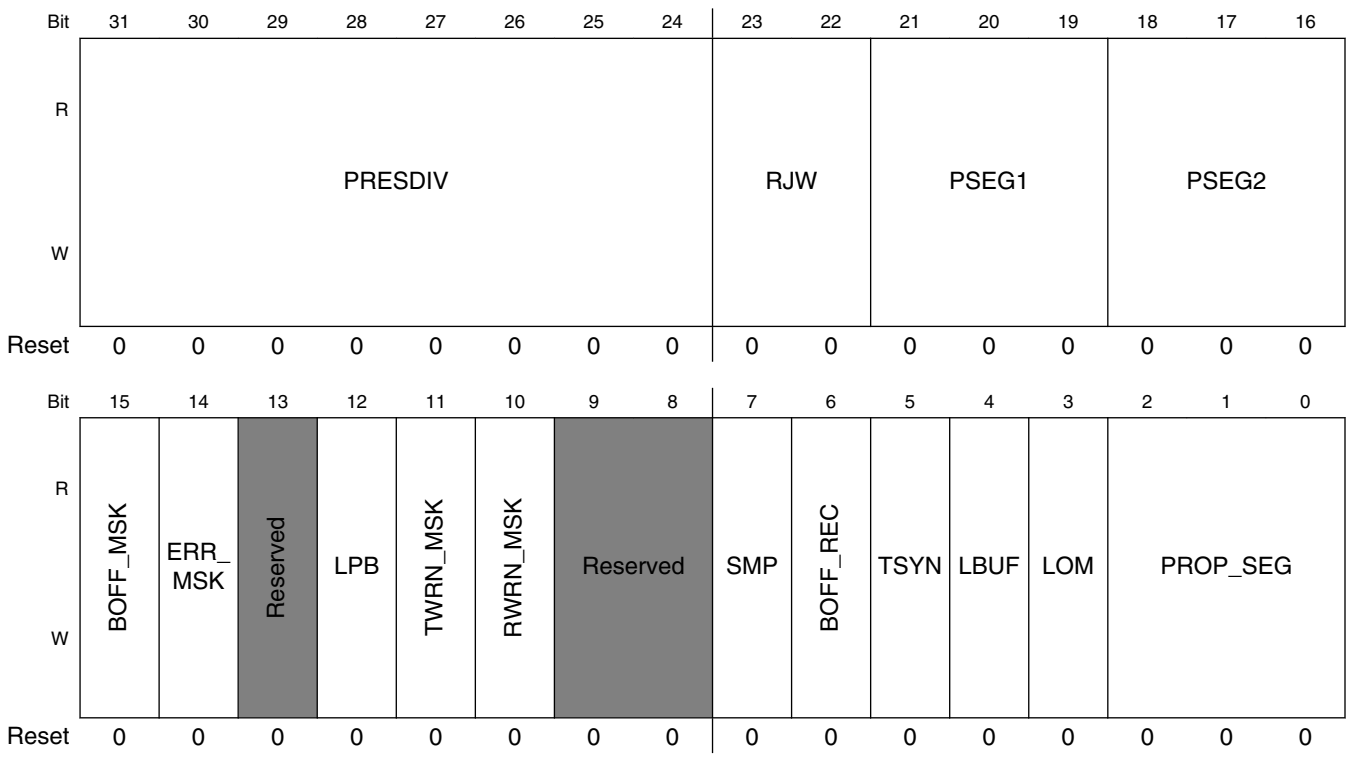
FLEXCANx_MCR field descriptions (continued)

Field	Description
	NOTE: Additionally, the value of MAXMB must encompass the FIFO size defined by CTRL2[RFFN] MAXMB also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in Table 27-18 (see Arbitration and Matching Timing).

27.8.2 Control 1 Register (FLEXCANx_CTRL1)

This register is defined for specific FLEXCAN control features related to the CAN bus, such as bit-rate, programmable sampling point within an Rx bit, Loop Back Mode, Listen Only Mode, Bus Off recovery behavior and interrupt enabling (Bus-Off, Error, Warning). It also determines the Division Factor for the clock prescaler.

Address: Base address + 4h offset



FLEXCANx_CTRL1 field descriptions

Field	Description
31–24 PRESDIV	This 8-bit field defines the ratio between the PE clock frequency and the Serial Clock (Sclock) frequency. The Sclock period defines the time quantum of the CAN protocol. For the reset value, the Sclock frequency is equal to the PE clock frequency. The Maximum value of this register is 0xFF, that gives a minimum Sclock frequency equal to the PE clock frequency divided by 256.For more information refer to Protocol Timing. This field can only be written in Freeze mode as it is blocked by hardware in other modes.

Table continues on the next page...

FLEXCANx_CTRL1 field descriptions (continued)

Field	Description
	Sclock frequency = CPI clock frequency / (PRESDIV+1)
23–22 RJW	This 2-bit field defines the maximum number of time quanta[1] that a bit time can be changed by one re-synchronization. The valid programmable values are 0-3. This field can only be written in Freeze mode as it is blocked by hardware in other modes Resync Jump Width = RJW + 1.
21–19 PSEG1	This 3-bit field defines the length of Phase Buffer Segment 1 in the bit time . The valid programmable values are 0-7. This field can only be written in Freeze mode as it is blocked by hardware in other modes Phase Buffer Segment 1 = (PSEG1 + 1) x Time-Quanta.
18–16 PSEG2	This 3-bit field defines the length of Phase Buffer Segment 2 in the bit time . The valid programmable values are 1-7. This field can only be written in Freeze mode as it is blocked by hardware in other modes Phase Buffer Segment 2 = (PSEG2 + 1) x Time-Quanta.
15 BOFF_MSK	This bit provides a mask for the Bus Off Interrupt. 1 Bus Off interrupt enabled 0 Bus Off interrupt disabled
14 ERR_MSK	This bit provides a mask for the Error Interrupt. 1 Error interrupt enabled 0 Error interrupt disabled
13 -	This field is reserved. Reserved
12 LPB	This bit configures FlexCAN to operate in Loop-Back Mode. In this mode, FlexCAN performs an internal loop back that can be used for self test operation. The bit stream output of the transmitter is fed back internally to the receiver input. The FLEXCAN_RX input pin is ignored and the FLEXCAN_TX output goes to the recessive state (logic '1'). FlexCAN behaves as it normally does when transmitting, and treats its own transmitted message as a message received from a remote node. In this mode, FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field, generating an internal acknowledge bit to ensure proper reception of its own message. Both transmit and receive interrupts are generated. This bit can only be written in Freeze mode as it is blocked by hardware in other modes. 1 Loop Back enabled 0 Loop Back disabled
11 TWRN_MSK	This bit provides a mask for the Tx Warning Interrupt associated with the TWRN_INT flag in the Error and Status Register. This bit is read as zero when MCR[WRN_EN] bit is negated. This bit can only be written if MCR[WRN_EN] bit is asserted. 1 Tx Warning Interrupt enabled 0 Tx Warning Interrupt disabled
10 RWRN_MSK	This bit provides a mask for the Rx Warning Interrupt associated with the RWRN_INT flag in the Error and Status Register. This bit is read as zero when MCR[WRN_EN] bit is negated. This bit can only be written if MCR[WRN_EN] bit is asserted. 1 Rx Warning Interrupt enabled 0 Rx Warning Interrupt disabled
9–8 -	This field is reserved. Reserved
7 SMP	This bit defines the sampling mode of CAN bits at the FLEXCAN_RX. This bit can only be written in Freeze mode as it is blocked by hardware in other modes.

Table continues on the next page...

FLEXCANx_CTRL1 field descriptions (continued)

Field	Description
	1 Three samples are used to determine the value of the received bit: the regular one (sample point) and 2 preceding samples, a majority rule is used 0 Just one sample is used to determine the bit value
6 BOFF_REC	This bit defines how FLEXCAN recovers from Bus Off state. If this bit is negated, automatic recovering from Bus Off state occurs according to the CAN Specification 2.0B. If the bit is asserted, automatic recovering from Bus Off is disabled and the module remains in Bus Off state until the bit is negated by the user. If the negation occurs before 128 sequences of 11 recessive bits are detected on the CAN bus, then Bus Off recovery happens as if the BOFF_REC bit had never been asserted. If the negation occurs after 128 sequences of 11 recessive bits occurred, then FLEXCAN will re-synchronize to the bus by waiting for 11 recessive bits before joining the bus. After negation, the BOFF_REC bit can be re-asserted again during Bus Off, but it will only be effective the next time the module enters Bus Off. If BOFF_REC was negated when the module entered Bus Off, asserting it during Bus Off will not be effective for the current Bus Off recovery. 1 Automatic recovering from Bus Off state disabled 0 Automatic recovering from Bus Off state enabled, according to CAN Spec 2.0 part B
5 TSYN	This bit enables a mechanism that resets the free-running timer each time a message is received in Message Buffer 0. This feature provides means to synchronize multiple FLEXCAN stations with a special "SYNC" message (i.e., global network time). If the RFEN bit in MCR is set (FIFO enabled), the first available Mailbox, according to CTRL2[RFFN] setting, is used for timer synchronization instead of MB0.This bit can only be written in Freeze mode as it is blocked by hardware in other modes. 1 Timer Sync feature enabled 0 Timer Sync feature disabled
4 LBUF	This bit defines the ordering mechanism for Message Buffer transmission. When asserted, the LPRIO_EN bit does not affect the priority arbitration.This bit can only be written in Freeze mode as it is blocked by hardware in other modes. 1 Lowest number buffer is transmitted first 0 Buffer with highest priority is transmitted first
3 LOM	This bit configures FLEXCAN to operate in Listen Only Mode. In this mode, transmission is disabled, all error counters are frozen and the module operates in a CAN Error Passive mode. Only messages acknowledged by another CAN station will be received. If FLEXCAN detects a message that has not been acknowledged, it will flag a BIT0 error (without changing the REC), as if it was trying to acknowledge the message. Listen-Only Mode acknowledgement can be obtained by the state of ESR1[FLT_CONF] field which is Passive Error when Listen-Only Mode is entered. There can be some delay between the Listen-Only Mode request and acknowledge. This bit can only be written in Freeze mode as it is blocked by hardware in other modes. 1 FLEXCAN module operates in Listen Only Mode 0 Listen Only Mode is deactivated
PROP_SEG	This 3-bit field defines the length of the Propagation Segment in the bit time. The valid programmable values are 0-7. This field can only be written in Freeze mode as it is blocked by hardware in other modes Propagation Segment Time = (PROPSEG + 1) * Time-Quanta. Time-Quantum = one Sclock period.

1. One time quantum is equal to the Sclock period.

27.8.3 Free Running Timer Register (FLEXCANx_TIMER)

This register represents a 16-bit free running counter that can be read and written by the Arm. The timer starts from $0000 after Reset, counts linearly to $FFFF, and wraps around.

The timer is clocked by the FLEXCAN bit-clock (which defines the baud rate on the CAN bus). During a message transmission/reception, it increments by one for each bit that is received or transmitted. When there is no message on the bus, it counts using the previously programmed baud rate. During Freeze Mode, disable, and stop mode, the timer is not incremented.

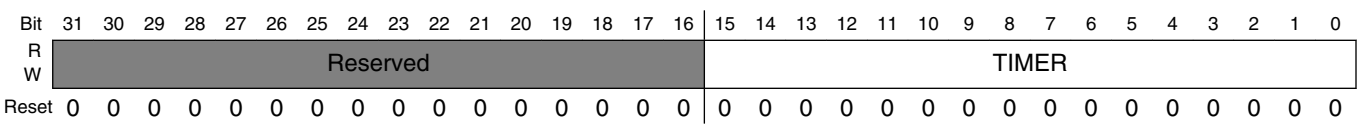
The timer value is captured at the beginning of the identifier field of any frame on the CAN bus. This captured value is written into the Time Stamp entry in a message buffer after a successful reception or transmission of a message.

If bit CTRL1[TSYN] is asserted the Timer is reset whenever a message is received in the first available Mailbox, according to CTRL2[RFFN] setting.

Arm can write to this register anytime. However, if the write occurs at the same time that the Timer is being reset by a reception in the first Mailbox, then the write value is discarded.

Reading this register affects the Mailbox Unlocking procedure. For additional details, refer to Message Buffer Lock Mechanism.

Address: Base address + 8h offset



FLEXCANx_TIMER field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
TIMER	TIMER

27.8.4 Rx Mailboxes Global Mask Register (FLEXCANx_RXMGMASK)

RXMGMASK is provided for legacy support. Asserting the MCR[IRMQ] bit causes the RXMGMASK Register to have no effect on the module operation.

RXMGMASK is used to mask the filter fields of all Rx MBs, excluding MBs 14-15, which have individual mask registers.

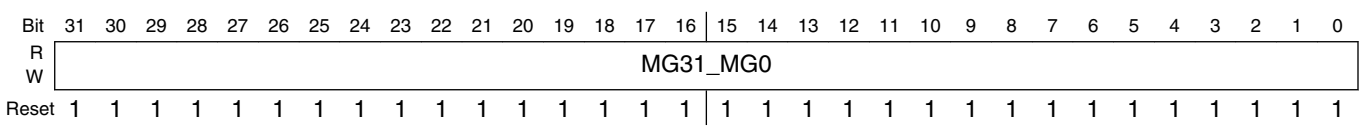
This register can only be written in Freeze mode as it is blocked by hardware in other modes.

Table 27-19. Rx Mailboxes Global Mask usage

SMB[RTR][1]	CTRL2[RRS]	CTRL2[EACEN]	Mailbox filter fields			
			MB[RTR]	MB[IDE]	MB[ID]	reserved
0	-	0	- Note[2]	- Note[3]	MG[28:0]	MG[31:29]
0	-	1	MG[31]	MG[30]	MG[28:0]	MG[29]
1	0	-	-	-	-	MG[31:0]
1	1	0	-	-	MG[28:0]	MG[31:29]
1	1	1	MG[31]	MG[30]	MG[28:0]	MG[29]

1. RTR bit of the Incoming Frame. It is saved into an auxiliary MB called Rx Serial Message Buffer (Rx SMB).
2. If CTRL2[EACEN] bit is negated the RTR bit of Mailbox is never compared with the RTR bit of the Incoming Frame (Rx SMB[RTR]).
3. If CTRL2[EACEN] bit is negated the IDE bit of Mailbox is always compared with the IDE bit of the Incoming Frame (Rx SMB[IDE]).

Address: Base address + 10h offset



FLEXCANx_RXMGMASK field descriptions

Field	Description
MG31_MG0	These bits mask the Mailbox filter bits as shown in the figure above. Note that the alignment with the ID word of the Mailbox is not perfect as the two most significant MG bits affect the fields RTR and IDE which are located in the Control and Status word of the Mailbox. Rx Mailboxes Global Mask Register (FLEXCAN_RXMGMASK) shows in detail which MG bits mask each Mailbox filter field. 1 The corresponding bit in the filter is checked against the one received 0 the corresponding bit in the filter is "don't care"

27.8.5 Rx Buffer 14 Mask Register (FLEXCANx_RX14MASK)

RX14MASK is provided for legacy support, asserting the MCR[IRMQ] bit causes the RX14MASK to have no effect on the module operation.

RX14MASK is used to mask the filter fields of Message Buffer 14.

This register can only be programmed while the module is in Freeze Mode as it is blocked by hardware in other modes.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX14M31_RX14M0																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

FLEXCANx_RX14MASK field descriptions

Field	Description
RX14M31_ RX14M0	These bits mask Mailbox 14 filter bits in the same fashion as RXMGMASK masks other Mailboxes filters (see Rx Mailboxes Global Mask Register (FLEXCAN_RXMGMASK)) 1 The corresponding bit in the filter is checked 0 the corresponding bit in the filter is "don't care"

27.8.6 Rx Buffer 15 Mask Register (FLEXCANx_RX15MASK)

RX15MASK is provided for legacy support, asserting the MCR[IRMQ] bit causes the RX15MASK Register to have no effect on the module operation.

RX15MASK is used to mask the filter fields of Message Buffer 15.

This register can only be programmed while the module is in Freeze Mode as it is blocked by hardware in other modes.

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX15M31_RX15M0																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

FLEXCANx_RX15MASK field descriptions

Field	Description
RX15M31_ RX15M0	These bits mask Mailbox 15 filter bits in the same fashion as RXMGMASK masks other Mailboxes filters (see Rx Mailboxes Global Mask Register (FLEXCAN_RXMGMASK)). 1 The corresponding bit in the filter is checked 0 the corresponding bit in the filter is "don't care"

27.8.7 Error Counter Register (FLEXCANx_ECR)

This register has 2 8-bit fields reflecting the value of two FLEXCAN error counters: Transmit Error Counter (Tx_Err_Counter field) and Receive Error Counter (Rx_Err_Counter field). The rules for increasing and decreasing these counters are described in the CAN protocol and are completely implemented in the FLEXCAN module. Both counters are read only except in Freeze Mode, where they can be written by the Arm.

FLEXCAN responds to any bus state as described in the protocol, e.g. transmit 'Error Active' or 'Error Passive' flag, delay its transmission start time ('Error Passive') and avoid any influence on the bus when in 'Bus Off' state. The following are the basic rules for FLEXCAN bus state transitions.

- If the value of Tx_Err_Counter or Rx_Err_Counter increases to be greater than or equal to 128, the FLT_CONF field in the Error and Status Register is updated to reflect 'Error Passive' state.
- If the FLEXCAN state is 'Error Passive', and either Tx_Err_Counter or Rx_Err_Counter decrements to a value less than or equal to 127 while the other already satisfies this condition, the FLT_CONF field in the Error and Status Register is updated to reflect 'Error Active' state.
- If the value of Tx_Err_Counter increases to be greater than 255, the FLT_CONF field in the Error and Status Register is updated to reflect 'Bus Off' state, and an interrupt may be issued. The value of Tx_Err_Counter is then reset to zero.
- If FLEXCAN is in 'Bus Off' state, then Tx_Err_Counter is cascaded together with another internal counter to count the 128th occurrences of 11 consecutive recessive bits on the bus. Hence, Tx_Err_Counter is reset to zero and counts in a manner where the internal counter counts 11 such bits and then wraps around while incrementing the Tx_Err_Counter. When Tx_Err_Counter reaches the value of 128, the FLT_CONF field in the Error and Status Register is updated to be 'Error Active' and both error counters are reset to zero. At any instance of dominant bit following a stream of less than 11 consecutive recessive bits, the internal counter resets itself to zero without affecting the Tx_Err_Counter value.
- If during system start-up, only one node is operating, then its Tx_Err_Counter increases in each message it is trying to transmit, as a result of acknowledge errors (indicated by the ACK_ERR bit in the Error and Status Register). After the transition to 'Error Passive' state, the Tx_Err_Counter does not increment anymore by acknowledge errors. Therefore the device never goes to the 'Bus Off' state.
- If the Rx_Err_Counter increases to a value greater than 127, it is not incremented further, even if more errors are detected while being a receiver. At the next

successful message reception, the counter is set to a value between 119 and 127 to resume to 'Error Active' state.

Address: Base address + 1Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																RX_ERR_COUNTER								TX_ERR_COUNTER							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXCANx_ECR field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15–8 RX_ERR_ COUNTER	Rx_Err_Counter
TX_ERR_ COUNTER	Tx_Err_Counter

27.8.8 Error and Status 1 Register (FLEXCANx_ESR1)

This register reflects various error conditions, some general status of the device and it is the source of four interrupts to the Arm.

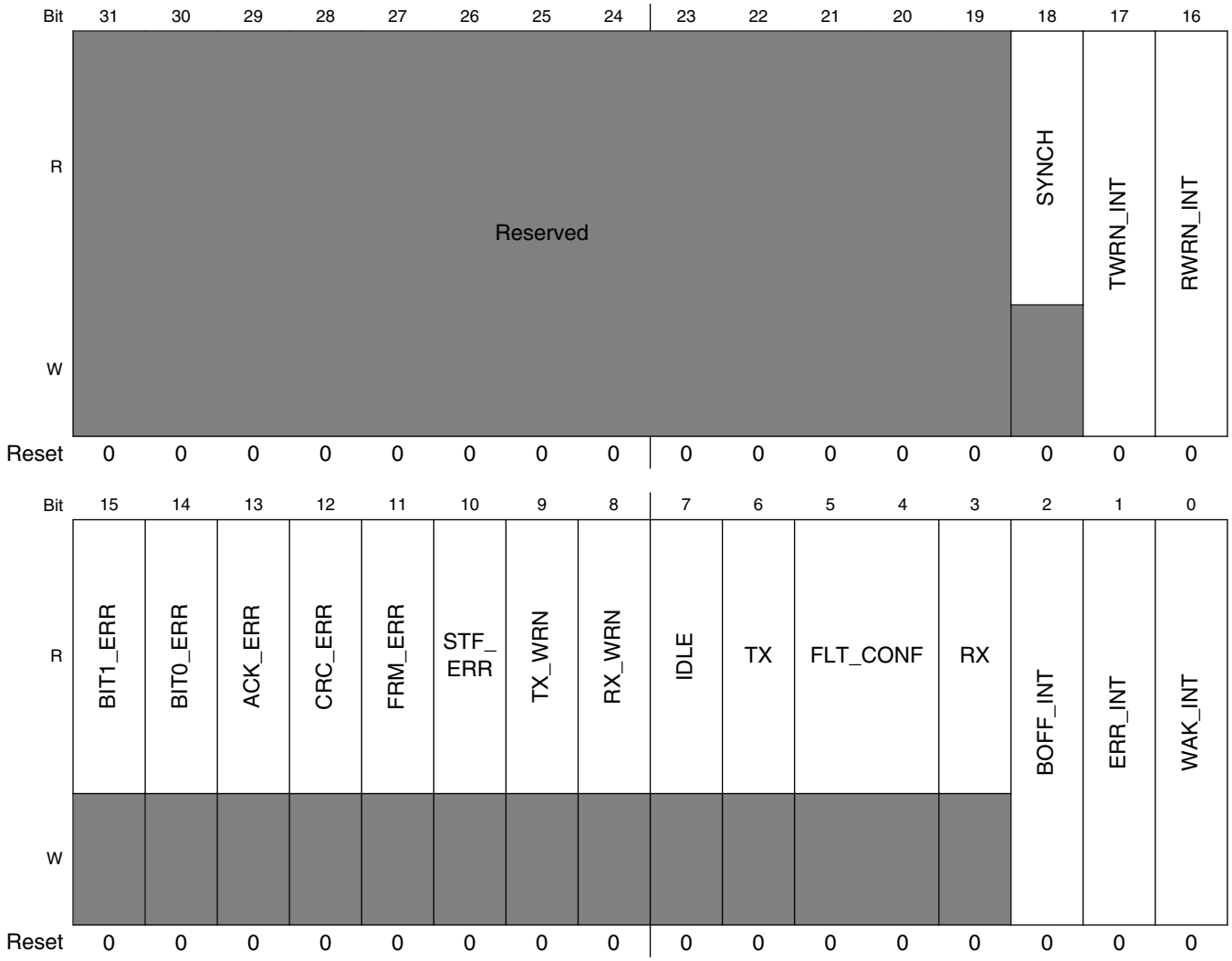
The Arm read action clears bits 15-10, therefore the reported *error conditions*(bits 15-10) are those that occurred since the last time the Arm read this register. Bits 9-3 are status bits .

Some bits in this register are read-only and some are not .

Table 27-20. FlexCAN State

SYNCH	IDLE	TX	RX	FlexCAN state
0	0	0	0	Not synchronized to CAN bus
1	1	x	x	Idle
1	0	1	0	Transmitting
1	0	0	1	Receiving
other combinations				Reserved

Address: Base address + 20h offset



FLEXCANx_ESR1 field descriptions

Field	Description
31–19 -	This field is reserved. Reserved
18 SYNCH	This read-only flag indicates whether the FlexCAN is synchronized to the CAN bus and able to participate in the communication process. It is set and cleared by the FlexCAN. Refer to Table 27-20 1 FlexCAN is synchronized to the CAN bus 0 FlexCAN is not synchronized to the CAN bus
17 TWRN_INT	If the WRN_EN bit in MCR is asserted, the TWRN_INT bit is set when the TX_WRN flag transition from '0' to '1', meaning that the Tx error counter reached 96. If the corresponding mask bit in the Control Register (TWRN_MSK) is set, an interrupt is generated to the Arm. This bit is cleared by writing it to '1'. When WRN_EN is negated, this flag is masked. Arm must clear this flag before disabling the bit. Otherwise it will be set when the WRN_EN is set again. Writing '0' has no effect. This flag is not generated during "Bus Off" state. This bit is not updated during Freeze mode. 1 The Tx error counter transition from < 96 to >= 96 0 No such occurrence

Table continues on the next page...

FLEXCANx_ESR1 field descriptions (continued)

Field	Description
16 RWRN_INT	If the WRN_EN bit in MCR is asserted, the RWRN_INT bit is set when the RX_WRN flag transition from '0' to '1', meaning that the Rx error counters reached 96. If the corresponding mask bit in the Control Register (RWRN_MSK) is set, an interrupt is generated to the Arm. This bit is cleared by writing it to '1'. When WRN_EN is negated, this flag is masked. Arm must clear this flag before disabling the bit. Otherwise it will be set when the WRN_EN is set again. Writing '0' has no effect. This bit is not updated during Freeze mode. 1 The Rx error counter transition from < 96 to >= 96 0 No such occurrence
15 BIT1_ERR	This bit indicates when an inconsistency occurs between the transmitted and the received bit in a message. This bit is not set by a transmitter in case of arbitration field or ACK slot, or in case of a node sending a passive error flag that detects dominant bits. 1 At least one bit sent as recessive is received as dominant 0 No such occurrence
14 BIT0_ERR	This bit indicates when an inconsistency occurs between the transmitted and the received bit in a message. 1 At least one bit sent as dominant is received as recessive 0 No such occurrence
13 ACK_ERR	This bit indicates that an Acknowledge Error has been detected by the transmitter node, i.e., a dominant bit has not been detected during the ACK SLOT. 1 An ACK error occurred since last read of this register 0 No such occurrence
12 CRC_ERR	This bit indicates that a CRC Error has been detected by the receiver node, i.e., the calculated CRC is different from the received. 1 A CRC error occurred since last read of this register. 0 No such occurrence
11 FRM_ERR	This bit indicates that a Form Error has been detected by the receiver node, i.e., a fixed-form bit field contains at least one illegal bit. 1 A Form Error occurred since last read of this register 0 No such occurrence
10 STF_ERR	This bit indicates that a Stuffing Error has been detected. 1 A Stuffing Error occurred since last read of this register. 0 No such occurrence.
9 TX_WRN	This bit indicates when repetitive errors are occurring during message transmission. 1 TX_Err_Counter ≥ 96 0 No such occurrence
8 RX_WRN	This bit indicates when repetitive errors are occurring during message reception. 1 Rx_Err_Counter ≥ 96 0 No such occurrence
7 IDLE	This bit indicates when CAN bus is in IDLE state.Refer to Table 27-20.

Table continues on the next page...

FLEXCANx_ESR1 field descriptions (continued)

Field	Description
	1 CAN bus is now IDLE 0 No such occurrence
6 TX	This bit indicates if FLEXCAN is transmitting a message.Refer to Table 27-20. 1 FLEXCAN is transmitting a message 0 FLEXCAN is receiving a message
5–4 FLT_CONF	If the LOM bit in the Control Register is asserted, after some delay that depends on the CAN bit timing the FLT_CONF field will indicate "Error Passive". The very same delay affects the way how FLT_CONF reflects an update to ECR register by the Arm. It may be necessary up to one CAN bit time to get them coherent again. Since the Control Register is not affected by soft reset, the FLT_CONF field will not be affected by soft reset if the LOM bit is asserted. This 2-bit field indicates the Confinement State of the FLEXCAN module, as shown in below: 00 Error Active 01 Error Passive 1x Bus off
3 RX	This bit indicates if FlexCAN is receiving a message. Refer to Table 27-20. 1 FLEXCAN is transmitting a message 0 FLEXCAN is receiving a message
2 BOFF_INT	This bit is set when FLEXCAN enters 'Bus Off' state. If the corresponding mask bit in the Control Register (BOFF_MSK) is set, an interrupt is generated to the Arm. This bit is cleared by writing it to '1'. Writing '0' has no effect. 1 FLEXCAN module entered 'Bus Off' state 0 No such occurrence
1 ERR_INT	This bit indicates that at least one of the Error Bits (bits 15-10) is set. If the corresponding mask bit in the Control Register (ERR_MSK) is set, an interrupt is generated to the Arm. This bit is cleared by writing it to '1'.Writing '0' has no effect. 1 Indicates setting of any Error Bit in the Error and Status Register 0 No such occurrence
0 WAK_INT	When FLEXCAN is Stop Mode and a recessive to dominant transition is detected on the CAN bus and if the WAK_MSK bit in the MCR Register is set, an interrupt is generated to the Arm. This bit is cleared by writing it to '1'. When SLF_WAK is negated, this flag is masked. Arm must clear this flag before disabling the bit. Otherwise it will be set when the SLF_WAK is set again. Writing '0' has no effect 1 Indicates a recessive to dominant transition received on the CAN bus when the FLEXCAN module is in Stop Mode 0 No such occurrence

27.8.9 Interrupt Masks 2 Register (FLEXCANx_IMASK2)

This register allows any number of a range of 32 Message Buffer Interrupts to be enabled or disabled. It contains one interrupt mask bit per buffer, enabling the Arm to determine which buffer generates an interrupt after a successful transmission or reception (i.e. when the corresponding IFLAG2 bit is set).

Address: Base address + 24h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF63M_BUF32M																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXCANx_IMASK2 field descriptions

Field	Description
BUF63M_ BUF32M	Each bit enables or disables the respective FLEXCAN Message Buffer (MB32 to MB63) Interrupt. Setting or clearing a bit in the IMASK2 Register can assert or negate an interrupt request, if the corresponding IFLAG2 bit is set. 1 The corresponding buffer Interrupt is enabled 0 The corresponding buffer Interrupt is disabled

27.8.10 Interrupt Masks 1 Register (FLEXCANx_IMASK1)

This register allows to enable or disable any number of a range of 32 Message Buffer Interrupts. It contains one interrupt mask bit per buffer, enabling the Arm to determine which buffer generates an interrupt after a successful transmission or reception (i.e., when the corresponding IFLAG1 bit is set).

Address: Base address + 28h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF31M_BUF0M																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXCANx_IMASK1 field descriptions

Field	Description
BUF31M_BUF0M	Each bit enables or disables the respective FLEXCAN Message Buffer (MB0 to MB31) Interrupt. Setting or clearing a bit in the IMASK1 Register can assert or negate an interrupt request, if the corresponding IFLAG1 bit is set

FLEXCANx_IMASK1 field descriptions (continued)

Field	Description
	1 The corresponding buffer Interrupt is enabled 0 The corresponding buffer Interrupt is disabled

27.8.11 Interrupt Flags 2 Register (FLEXCANx_IFLAG2)

This register defines the flags for 32 Message Buffer interrupts. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the corresponding IFLAG2 bit. If the corresponding IMASK2 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing it to '1'. Writing '0' has no effect.Before updating MCR[MAXMB] field, Arm must treat the IFLAG2 bits which MB value is greater than the MCR[MAXMB] to be updated, otherwise they will keep set and be inconsistent with the amount of MBs available.

Address: Base address + 2Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF63I_BUF32I																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXCANx_IFLAG2 field descriptions

Field	Description
BUF63I_BUF32I	Each bit flags the respective FLEXCAN Message Buffer (MB32 to MB63) interrupt. 1 The corresponding buffer has successfully completed transmission or reception 0 No such occurrence

27.8.12 Interrupt Flags 1 Register (FLEXCANx_IFLAG1)

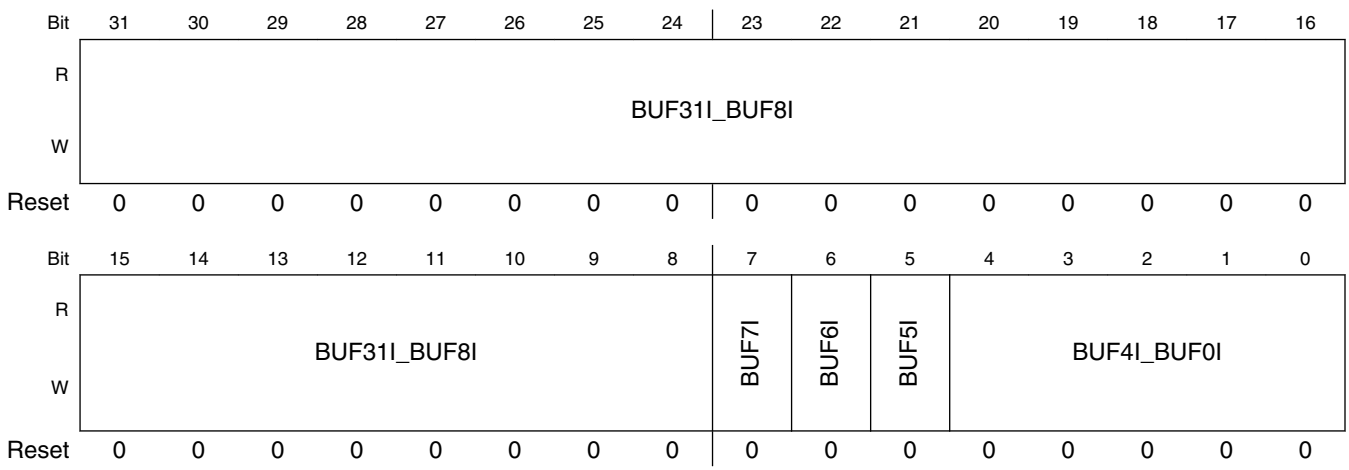
This register defines the flags for 32 Message Buffer interrupts and FIFO interrupts. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the corresponding IFLAG1 bit. If the corresponding IMASK1 bit is set, an interrupt will be generated. The Interrupt flag must be cleared by writing it to '1'. Writing '0' has no effect.

When the RFEN bit in the MCR is set (Rx FIFO enabled), the function of the 8 least significant interrupt flags (BUF7I - BUF0I) is changed to support the FIFO operation. BUF7I, BUF6I and BUF5I indicate operating conditions of the FIFO, while BUF4I to BUF0I are not used. Before enabling the RFEN, Arm must service the IFLAGS asserted

in the Rx FIFO region (see Rx FIFO). Otherwise, these IFLAGS will mistakenly show the related MBs now belonging to FIFO as having contents to be serviced. When the RFEN is negated, the FIFO flags must be cleared. The same care must be taken when a RFFN value is selected extending Rx FIFO filters beyond MB7 (see Control 2 Register (FLEXCAN_CTRL2)). For example, when RFFN is 0x8, the MB0-23 range is occupied by Rx FIFO filters and related IFLAGS must be cleared.

Before updating MCR[MAXMB] field, Arm must service the IFLAG1 which MB value is greater than the MCR[MAXMB] to be updated, otherwise they will keep set and be inconsistent with the amount of MBs available.

Address: Base address + 30h offset



FLEXCANx_IFLAG1 field descriptions

Field	Description
31–8 BUF31I_BUF8I	Each bit flags the respective FLEXCAN Message Buffer (MB8 to MB31) interrupt. 1 The corresponding MB has successfully completed transmission or reception 0 No such occurrence
7 BUF7I	If the Rx FIFO is not enabled, this bit flags the interrupt for MB7. If the MCR[RFEN] bit is asserted, this flag indicates that a message was lost because Rx FIFO is full. Note that the flag will not be asserted when the Rx FIFO is full and the message was captured by a Mailbox. This flag is cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by Arm writes. 1 MB7 completed transmission/reception or FIFO overflow 0 No such occurrence
6 BUF6I	If the Rx FIFO is not enabled, this bit flags the interrupt for MB6. If the MCR[RFEN] bit is asserted, this flag indicates when the number of unread messages within the Rx FIFO is increased to 5 from 4 due to the reception of a new one, meaning that the Rx FIFO is almost full. Note that if the flag is cleared while the number of unread messages is greater than 4 it will not assert again until the number of unread messages within the Rx FIFO is decreased to equal or less than 4. This flag is cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by Arm writes.

Table continues on the next page...

FLEXCANx_IFLAG1 field descriptions (continued)

Field	Description
	1 MB6 completed transmission/reception or FIFO almost full 0 No such occurrence
5 BUF5I	If the Rx FIFO is not enabled, this bit flags the interrupt for MB5. If the Rx FIFO is enabled, this flag indicates that at least one frame is available to be read from the Rx FIFO. This flag is cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by Arm writes. 1 MB5 completed transmission/reception or frames available in the FIFO 0 No such occurrence
BUF4I_BUF0I	If the Rx FIFO is not enabled, these bits flag the interrupts for MB0 to MB4 . If the Rx FIFO is enabled, these flags are not used and must be considered as reserved locations. These flags are cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by Arm writes. 1 Corresponding MB completed transmission/reception 0 No such occurrence

27.8.13 Control 2 Register (FLEXCANx_CTRL2)

This register contains control bits for CAN errors, FIFO features and mode selection.

Table 27-21. Rx FIFO Filters

RFFN[3:0]	Number of Rx FIFO filters	Message Buffers occupied by Rx FIFO and ID Filter Table	Remaining Available Mailboxes[1]	Rx FIFO ID Filter Table Elements Affected by Rx Individual Masks[2]	Rx FIFO ID Filter Table Elements Affected by Rx FIFO Global Mask[2]
0x0	8	MB 0-7	MB 8-63	Elements 0-7	none
0x1	16	MB 0-9	MB 10-63	Elements 0-9	Elements 10-15
0x2	24	MB 0-11	MB 12-63	Elements 0-11	Elements 12-23
0x3	32	MB 0-13	MB 14-63	Elements 0-13	Elements 14-31
0x4	40	MB 0-15	MB 16-63	Elements 0-15	Elements 16-39
0x5	48	MB 0-17	MB 18-63	Elements 0-17	Elements 18-47
0x6	56	MB 0-19	MB 20-63	Elements 0-19	Elements 20-55
0x7	64	MB 0-21	MB 22-63	Elements 0-21	Elements 22-63
0x8	72	MB 0-23	MB 24-63	Elements 0-23	Elements 24-71
0x9	80	MB 0-25	MB 26-63	Elements 0-25	Elements 26-79
0xA	88	MB 0-27	MB 28-63	Elements 0-27	Elements 28-87
0xB	96	MB 0-29	MB 30-63	Elements 0-29	Elements 30-95
0xC	104	MB 0-31	MB 32-63	Elements 0-31	Elements 32-103
0xD	112	MB 0-33	MB 34-63	Elements 0-31	Elements 32-111
0xE	120	MB 0-35	MB 36-63	Elements 0-31	Elements 32-119
0xF	128	MB 0-37	MB 38-63	Elements 0-31	Elements 32-127

1. The number of the last remaining available mailboxes is defined by the MCR[MAXMB] field.
2. If Rx Individual Mask Registers are not enabled then all Rx FIFO filters are affected by the Rx FIFO Global Mask.

Each group of eight filters occupies a memory space equivalent to two Message Buffers which means that the more filters are implemented the less Mailboxes will be available.

Considering that the Rx FIFO occupies the memory space originally reserved for MB0-5, RFFN should be programmed with a value corresponding to a number of filters not greater than the number of available memory words which can be calculated as follows:

$$(SETUP_MB - 6) \times 4$$

where SETUP_MB is MAXMB.

The number of remaining Mailboxes available will be:

$$SETUP_MB - 8 - (RFFN \times 2)$$

If the Number of Rx FIFO Filters programmed through RFFN exceeds the SETUP_MB value, the exceeding ones will not be functional.Unshaded regions in Table 27-22 indicate the valid combinations of MAXMB, RFEN and RFFN, shaded regions are not functional.

Table 27-22. Valid Combinations of MAXMB, RFEN and RFFN

RFFN	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RFEN	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
MAXMB																	
0 - 6																	
7 - 8																	
9 - 10																	
11 - 12																	
13 -14																	
15 - 16																	
17 -18																	
19 - 20																	
21 - 22																	

Table continues on the next page...

Table 27-22. Valid Combinations of MAXMB, RFEN and RFFN (continued)

RFFN	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RFEN	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
MAXMB																	
23 - 24																	
25 - 26																	
27 - 28																	
29 -30																	
31 - 32																	
33 - 34																	
35 - 36																	
37 - 63																	

Address: Base address + 34h offset



FLEXCANx_CTRL2 field descriptions

Field	Description
31 -	must be written as 0
30–29 -	This field is reserved. Reserved

Table continues on the next page...

FLEXCANx_CTRL2 field descriptions (continued)

Field	Description
28 WRMFRZ	Enable unrestricted write access to FlexCAN memory in Freeze mode. This bit can only be written in Freeze mode and has no effect out of Freeze mode. 1 Enable unrestricted write access to FlexCAN memory 0 Keep the write access restricted in some regions of FlexCAN memory
27–24 RFFN	This 4-bit field defines the number of Rx FIFO filters according to Table 27-21 . The maximum selectable number of filters is determined by the Arm. This field can only be written in Freeze mode as it is blocked by hardware in other modes. RFFN defines a number of Message Buffers occupied by Rx FIFO and ID Filter (see Table 27-21) that **may not exceed** the number of available Mailboxes present in module, defined by MCR[MAXMB].Default RFFN value is 0x0, which leads to a total of 8 Rx FIFO filters, occupies the first 8 Message Buffers (MB 0-7) and makes available the next Message Buffers (MB 8-63) for Mailboxes. As a second example, when RFFN is set to 0xD, there will be 112 Rx FIFO filters, located in MB 0-33, and MB 34-63 are available for Mailboxes. Notice that, in this case, individual masks (RXIMR) will just cover Rx FIFO filters in 0-31 range, and filters 32-111 will use RXFGMASK. In case of reducing the number of last Message Buffers, MCR[MAXMB] (see Module Configuration Register (FLEXCAN_MCR)) can be adjusted by the application to minimum of 33, in order to give room to the Rx FIFO and its ID Filter Table defined by RFFN. On the contrary, if the application sets MCR[MAXMB] to 16, for instance, the maximum RFFN is limited to 0x4. RFFN also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in Table 27-18 (see Arbitration and Matching Timing).
23–19 TASD	This 5-bit field indicates how many CAN bits the Tx arbitration process start point can be delayed from the first bit of CRC field on CAN bus. This field can only be written in Freeze mode as it is blocked by hardware in other modes. This field is useful to optimize the transmit performance based on factors such as: peripheral/serial clock ratio, CAN bit timing and number of MBs . The duration of an arbitration process, in terms of CAN bits, is directly proportional to the number of available MBs and CAN baud rate and inversely proportional to the peripheral clock frequency. The optimal arbitration timing is that in which the last MB is scanned right before the first bit of the Intermission field of a CAN frame. Therefore, if there are few MBs and the system/serial clock ratio is high and the CAN baud rate is low then the arbitration can be delayed and vice-versa. If TASD is 0 then the arbitration start is not delayed, thus Arm has less time to configure a Tx MB for the next arbitration, but more time is reserved for arbitration . In the other hand, if TASD is 24 then Arm can configure a Tx MB later and less time is reserved for arbitration. If too little time is reserved for arbitration the FlexCAN may be not able to find winner MBs in time to compete with other nodes for the CAN bus. If the arbitration ends too much time before the first bit of Intermission field then there is a chance that Arm reconfigure some Tx MBs and the winner MB is not the best to be transmitted. The reset value is different on various platforms, according to their peripheral clock frequency, number of MBs and target CAN baud rate. The optimal configuration for TASD can be calculated as:

Table continues on the next page...

FLEXCANx_CTRL2 field descriptions (continued)

Field	Description
	$$TASD = 25 - \left\{ \frac{f_{CANCLK} \times [MAXMB + 3 - (RFEN \times 8) - (RFEN \times RFFN \times 2)] \times 2}{f_{sys} \times [1 + (PSEG1 + 1) + (PSEG2 + 1) + (PROPSEG + 1)] \times (PRESDIV + 1)} \right\}$$ where: f_{CANCLK} is the Protocol Engine (PE) Clock in Hz; PE clock is derrived from CAN_CLK_ROOT in CCM. See the Clock controller module. f_{SYS} is the peripheral clock in Hz; MAXMB is the value in CTRL1[MAXMB] field; RFEN is the value in CTRL1[RFEN] bit; RFFN is the value in CTRL2[RFFN] field; PSEG1 is the value in CTRL1[PSEG1] field; PSEG2 is the value in CTRL1[PSEG2] field; PROPSEG is the value in CTRL1[PROPSEG] field; PRESDIV is the value in CTRL1[PRESDIV] field. Please refer to Arbitration process and Protocol Timing for more details.
18 MRP	If this bit is set the matching process starts from the Mailboxes and if no match occurs the matching continues on the Rx FIFO. This bit can only be written in Freeze mode as it is blocked by hardware in other modes. 1 Matching starts from Mailboxes and continues on Rx FIFO 0 Matching starts from Rx FIFO and continues on Mailboxes
17 RRS	If this bit is asserted Remote Request Frame is submitted to a matching process and stored in the corresponding Message Buffer in the same fashion of a Data Frame. No automatic Remote Response Frame will be generated. If this bit is negated the Remote Request Frame is submitted to a matching process and an automatic Remote Response Frame is generated if a Message Buffer with CODE=0b1010 is found with the same ID. This bit can only be written in Freeze mode as it is blocked by hardware in other modes. 1 Remote Request Frame is stored 0 Remote Response Frame is generated
16 EACEN	This bit controls the comparison of IDE and RTR bits within Rx Mailboxes filters with their corresponding bits in the incoming frame by the matching process. This bit does not affect matching for Rx FIFO. This bit can only be written in Freeze mode as it is blocked by hardware in other modes. 1 Enables the comparison of both Rx Mailbox filter's IDE and RTR bit with their corresponding bits within the incoming frame. Mask bits do apply. 0 Rx Mailbox filter's IDE bit is always compared and RTR is never compared despite mask bits.
-	This field is reserved.

Table continues on the next page...

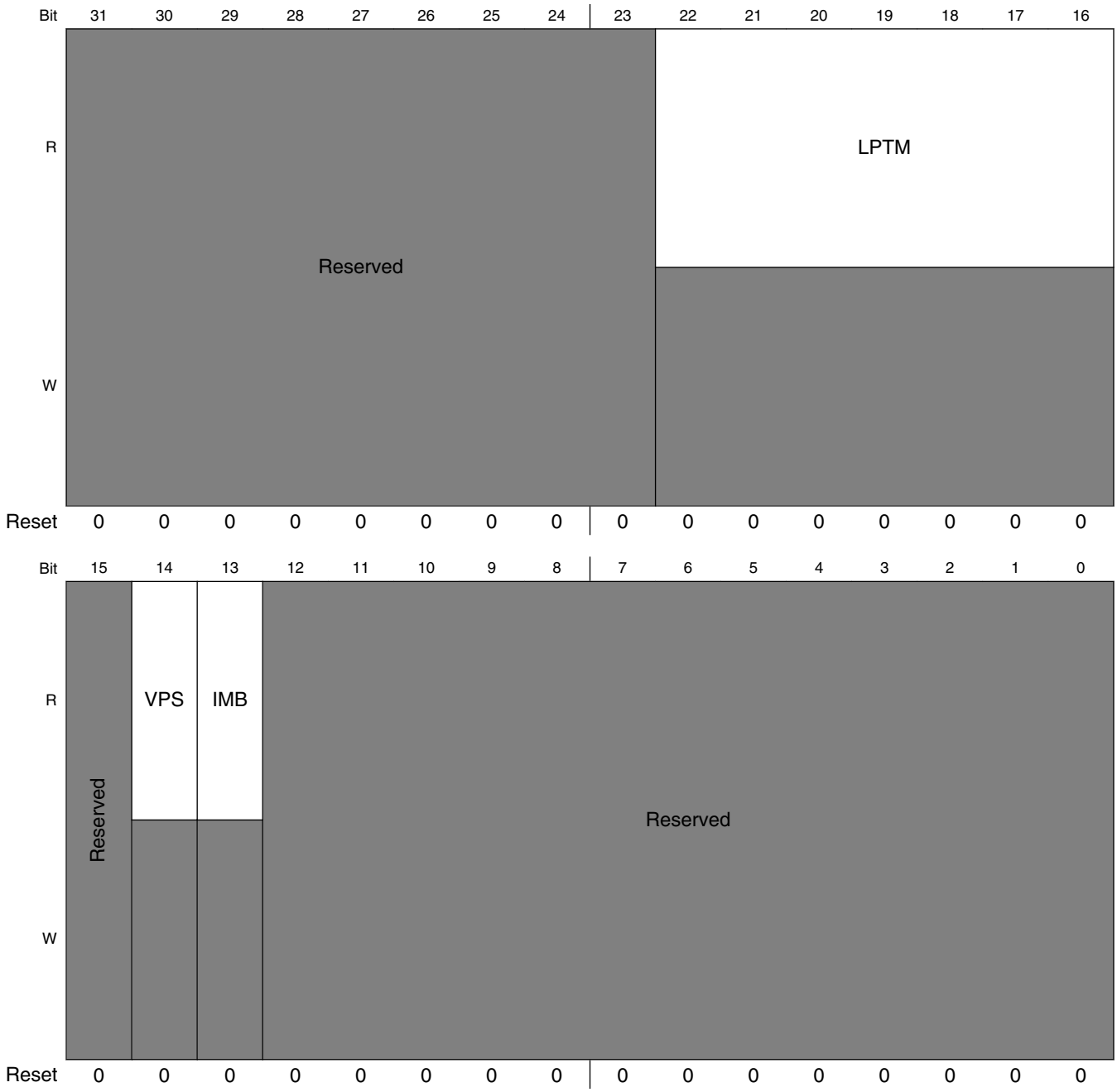
FLEXCANx_CTRL2 field descriptions (continued)

Field	Description
	Reserved

27.8.14 Error and Status 2 Register (FLEXCANx_ESR2)

This register reflects various interrupt flags and some general status.

Address: Base address + 38h offset



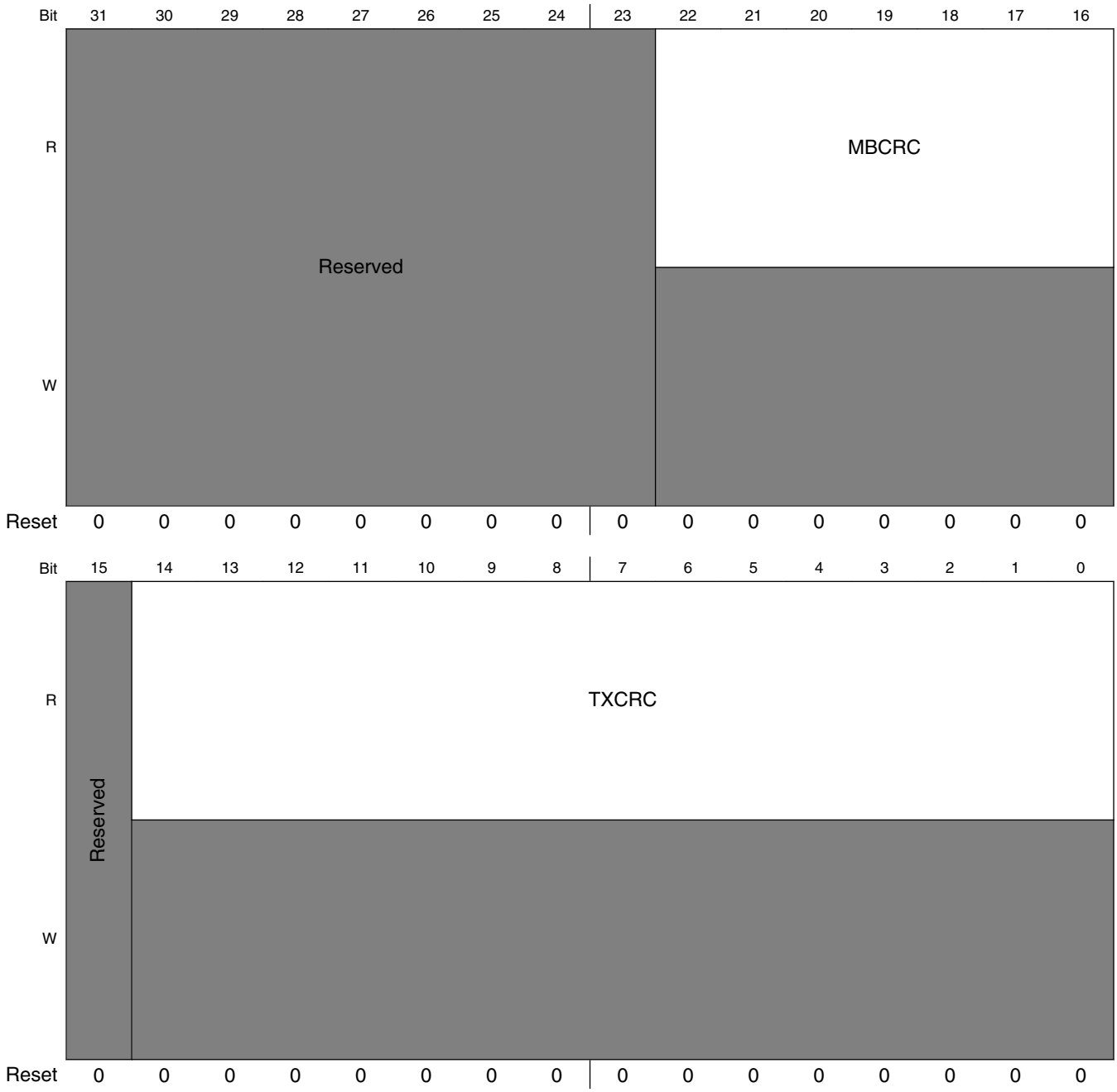
FLEXCANx_ESR2 field descriptions

Field	Description
31–23 -	This field is reserved. Reserved
22–16 LPTM	If ESR2[VPS] is asserted, his 7-bit field indicates the lowest number inactive Mailbox (refer to IMB bit description). If there is no inactive Mailbox then the Mailbox indicated depends on CTRL1[LBUF] bit value . If CTRL1[LBUF] bit is negated then the Mailbox indicated is the one which has the greatest arbitration value (see Highest Mailbox priority first). If CTRL1[LBUF] bit is asserted then the Mailbox indicated is the highest number active Tx Mailbox. If a Tx Mailbox is being transmitted it is not considered in LPTM calculation. If ESR2[IMB] is not asserted and a frame is transmitted successfully, LPTM is updated with its Mailbox number.
15 -	This field is reserved. Reserved
14 VPS	This bit indicates whether IMB and LPTM contents are currently valid or not. VPS is asserted upon every complete Tx arbitration process unless the Arm writes to Control and Status word of a Mailbox that has already been scanned (i.e. it is behind Tx Arbitration Pointer) during the Tx arbitration process . If there is no inactive Mailbox and only one Tx Mailbox which is being transmitted then VPS is not asserted. VPS is negated upon the start of every Tx arbitration process or upon a write to Control and Status word of any Mailbox.ESR2[VPS] is not affected by any Arm write into Control Status (C/S) of a MB which is blocked by abort mechanism. When MCR[AEN] is asserted, the abort code write in C/S of a MB that is been transmitted (pending abort), or any write attempt into a Tx MB with IFLAG set is blocked. 1 Contents of IMB and LPTM are valid 0 Contents of IMB and LPTM are invalid
13 IMB	If ESR2[VPS] is asserted, this bit indicates whether there is any inactive Mailbox (CODE field is either 0b1000 or 0b0000). This bit is asserted in the following cases: (1) During arbitration, if a LPTM is found and it is inactive. (2) If IMB is not asserted and a frame is transmitted successfully. (3) This bit is cleared in all start of arbitration (see Arbitration process). LPTM mechanism have the following behavior: if a MB is successfully transmitted and ESR2[IMB]=0 (no inactive Mailbox), then ESR2[VPS] and ESR2[IMB] are asserted and the index related to the MB just transmitted is loaded into ESR2[LPTM]. 1 If ESR2[VPS] is asserted, there is at least one inactive Mailbox. LPTM content is the number of the first one. 0 If ESR2[VPS] is asserted, the ESR2[LPTM] is not an inactive Mailbox.
-	This field is reserved. Reserved

27.8.15 CRC Register (FLEXCANx_CRCR)

This register provides information about the CRC of transmitted messages

Address: Base address + 44h offset



FLEXCANx_CRCR field descriptions

Field	Description
31–23 -	This field is reserved. Reserved
22–16 MBCRC	This field indicates the number of the Mailbox corresponding to the value in TXCRC field.
15 -	This field is reserved. Reserved
TXCRC	This field indicates the CRC value of the last message transmitted. This field is updated at the same time the Tx Interrupt Flag is asserted.

27.8.16 Rx FIFO Global Mask Register (FLEXCANx_RXFGMASK)

If Rx FIFO is enabled RXFGMASK is used to mask the Rx FIFO ID Filter Table elements that do not have a corresponding RXIMR according to CTRL2[RFFN] field setting.

This register can only be written in Freeze Mode as it is blocked by hardware in other modes.

Table 27-23. Rx FIFO Global Mask usage

Rx FIFO ID Filter Table Elements Format (MCR[IDAM])	Identifier Acceptance Filter fields					
	RTR	IDE	RXIDA	RXIDB	RXIDC	reserved
A	FGM[31]	FGM[30]	FGM[29:1]	-	-	FGM[0]
B	FGM[31] FGM[15]	FGM[30] FGM[14]	-	FGM[29:16] FGM[13:0] 1		-
C	-	-		-	FGM[31:24] FGM[23:16] FGM[15:8] FGM[7:0] 2	

1. If MCR[IDAM] field is equivalent to the format B only the fourteen most significant bits of the Identifier of the incoming frame are compared with the Rx FIFO filter.
2. If MCR[IDAM] field is equivalent to the format C only the eight most significant bits of the Identifier of the incoming frame are compared with the Rx FIFO filter.

Address: Base address + 48h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FGM31_FGM0																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

FLEXCANx_RXFGMASK field descriptions

Field	Description
FGM31_FGM0	These bits mask the ID Filter Table elements bits in a perfect alignment.Rx FIFO Global Mask Register (FLEXCAN_RXFGMASK) shows in detail which FGM bits mask each IDAF field. Clear this register has the effect of disabling the ID Filter. 1 The corresponding bit in the filter is checked 0 The corresponding bit in the filter is "don't care"

27.8.17 Rx FIFO Information Register (FLEXCANx_RXFIR)

RXFIR provides information on Rx FIFO.

This register is the port through which Arm accesses the output of the RXFIR FIFO located in RAM. The RXFIR FIFO is written by the FlexCAN whenever a new message is moved into the Rx FIFO as well as its output is updated whenever the output of the Rx FIFO is updated with the next message. Refer to Rx FIFO to find instructions on reading this register.

Address: Base address + 4Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																							IDHIT								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXCANx_RXFIR field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
IDHIT	This 9-bit field indicates which Identifier Acceptance Filter (see Rx FIFO Structure) was hit by the received message that is in the output of the Rx FIFO . (refer to Rx FIFO for details) If multiple filters match the incoming message ID then the first matching IDAF found (lowest number) by the matching process is indicated. This field is valid only while the IFLAG[BUF5I] is asserted.

27.8.18 Rx Individual Mask Registers (FLEXCANx_RXIMR*n*)

RXIMR are used as acceptance masks for ID filtering in Rx MBs and the Rx FIFO . If the Rx FIFO is not enabled, one mask register is provided for each available Mailbox, providing ID masking capability on a per Mailbox basis.

When the Rx FIFO is enabled (MCR[RFEN] bit is asserted), up to 32 Rx Individual Mask Registers can apply to the Rx FIFO ID Filter Table elements on a one-to-one correspondence depending on CTRL2[RFFN] setting. Refer to Control 2 Register (FLEXCAN_CTRL2) for details .

RXIMR can only be written by the Arm while the module is in Freeze Mode, otherwise they are blocked by hardware .

The Individual Rx Mask Registers are not affected by reset and must be explicitly initialized prior to any reception.

Address: Base address + 880h offset + (4d × i), where i=0d to 63d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MI31_MI0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXCANx_RXIMR*n* field descriptions

Field	Description
MI31_MI0	These bits mask both Mailbox filter and Rx FIFO ID Filter Table element in distinct ways. For Mailbox filter refer to Rx Mailboxes Global Mask Register (FLEXCAN_RXMGMASK). For Rx FIFO ID Filter Table element refer to Rx FIFO Global Mask Register (FLEXCAN_RXFGMASK). 1 The corresponding bit in the filter is checked 0 the corresponding bit in the filter is "don't care"

27.8.19 Glitch Filter Width Registers (FLEXCANx_GFWR)

The Glitch Filter just takes effects when FLEXCAN enters the STOP mode.

Address: Base address + 9E0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								GFWR							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

FLEXCANx_GFWR field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.
GFWR	It determines the Glitch Filter Width. The width will be divided from Oscillator clock by GFWR values. By default, it is 5.33 µs when the oscillator is 24 MHz. Filter Pulse Width = [(GFWR FIELD + 1) x (1 / Osc. Frequency)]

Chapter 28
General Power Controller (GPC)

28.1 Overview

The General Power Control (GPC) block includes the sub-blocks listed here.

- CPU Power Gating Control (PGC)
- GPU/VPU accelerators power gating controller

Each sub-block has its own IP registers.

GPC determines wake-up IRQ for exiting WAIT/STOP mode (with or without CPU power gating).

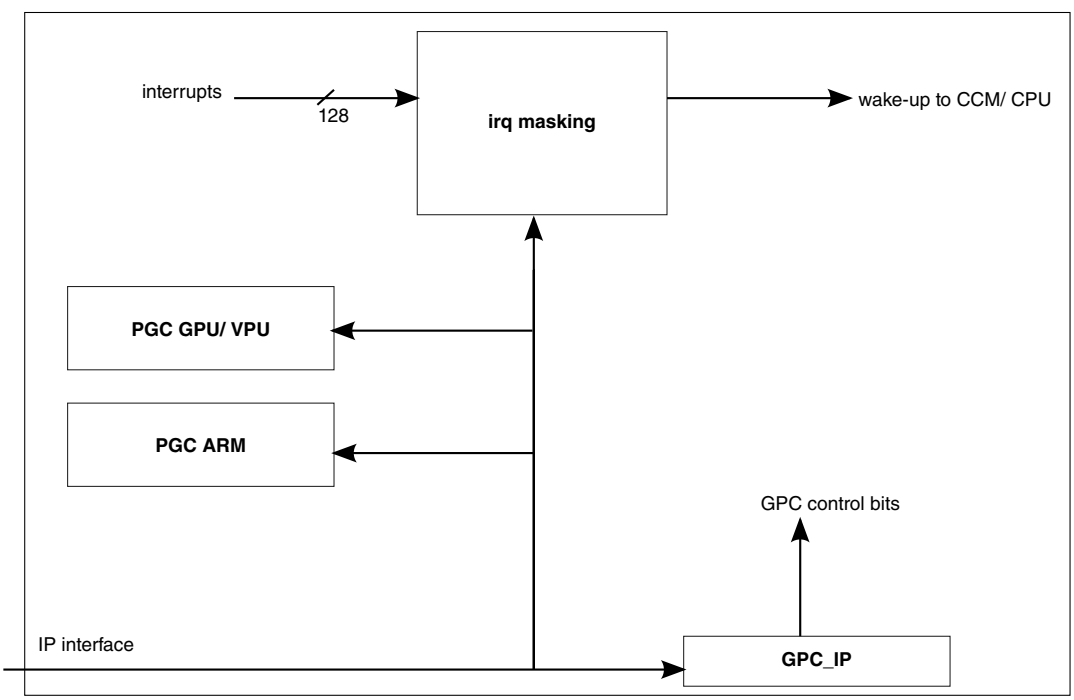


Figure 28-1. GPC Block Diagram

28.2 Clocks

The table found here describes the clock sources for GPC.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 28-1. GPC Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_s	ipg_clk_root	Peripheral access clock
pgc_clk	ipg_clk_root	PGC peripheral clock
sys_clk	ipg_clk_root	Module clock

28.3 Power Gating Control (PGC)

Power Gating (PGC) is applied to the Arm CPU only in STOP low power mode, after all essential CPU registers data are saved by Arm dormant procedure.

If any of the unmasked interrupts appears, CPU is powered up and clock restore request (exit from STOP mode) is sent to CCM.

PGC power down sequence:

- CCM sends power down request when the chip is about to enter stop mode. The user should define which modules will be powered down (CTRL registers of corresponding PGC module, bit 0).

PGC power up sequence:

- One of the power up irq is asserted.
- Power up request is asserted in GPC and in CCM.
- The Power Gated modules are powered up, according to PGC settings of appropriate module.

The Power Gated modules require reset after powering up. The next figure describes GPC-SRC handshake procedure for reset after power gating.

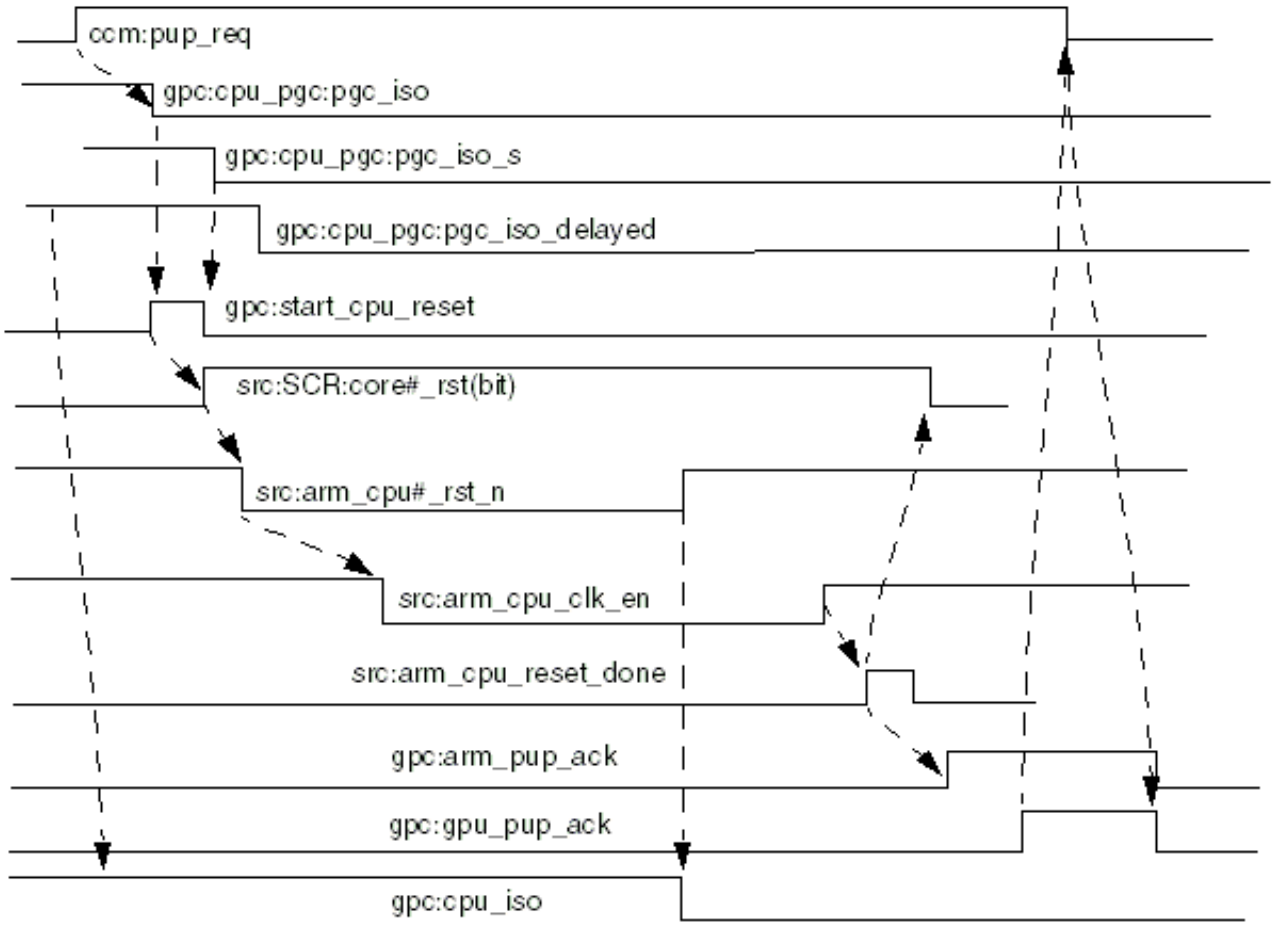


Figure 28-2. GPC-SRC handshake for reset after power gating

28.3.1 Overview

The Power Gating Controller (PGC) is a power management component that controls the power-down and power-up sequencing of individual subsystems.

The sequence timing is programmable using the PGC control registers. Figure 28-3 shows PGC as part of the SoC's overall power management scheme.

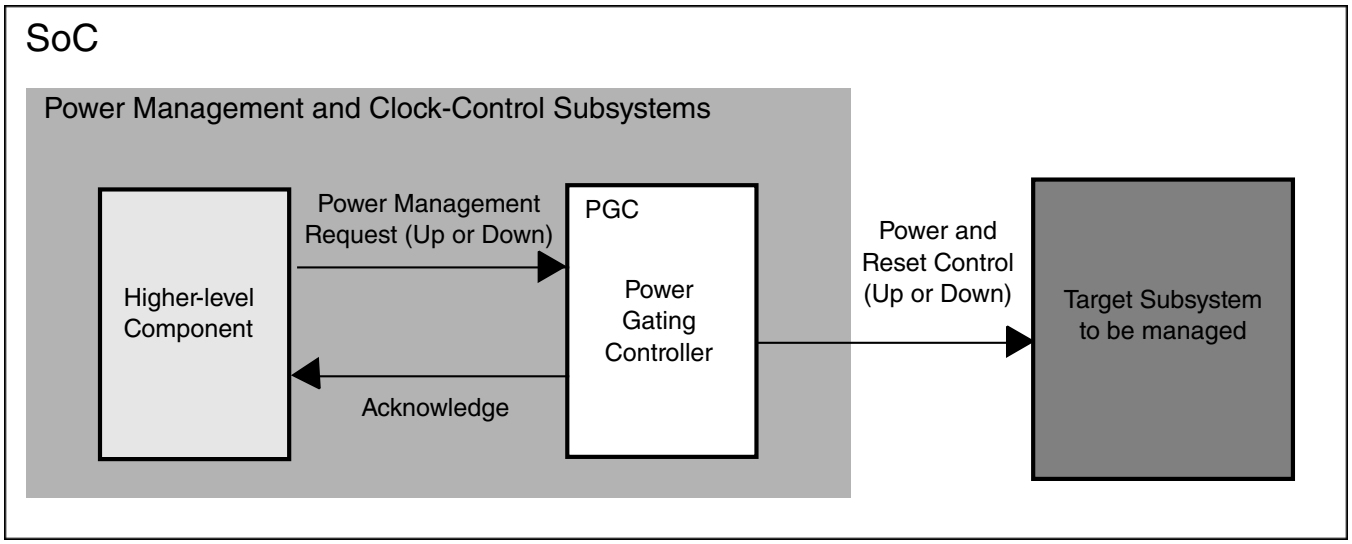


Figure 28-3. PGC Block Diagram

28.3.1.1 Features

Key features of the PGC include:

- Provides the ability to switch off power to a target subsystem.
- Generates power-up and power-down control sequences.
- Provides programmable registers to adjust the timing of the power control signals.

28.4 GPC Interrupt Controller (INTC)

The INTC (Interrupt Controller) detects an interrupt and generates the wakeup signal. It supports up to 128 interrupts.

28.4.1 Interrupt Controller features

The features of the GPC INTC are listed below.

Features:

- Supports up to 128 interrupts
- Provides an option to mask/unmask each interrupt
- Detects interrupts and generates the wake up signal
- 32-bits IP bus interface
- All registers are byte-accessible

28.5 GPC Memory Map/Register Definition

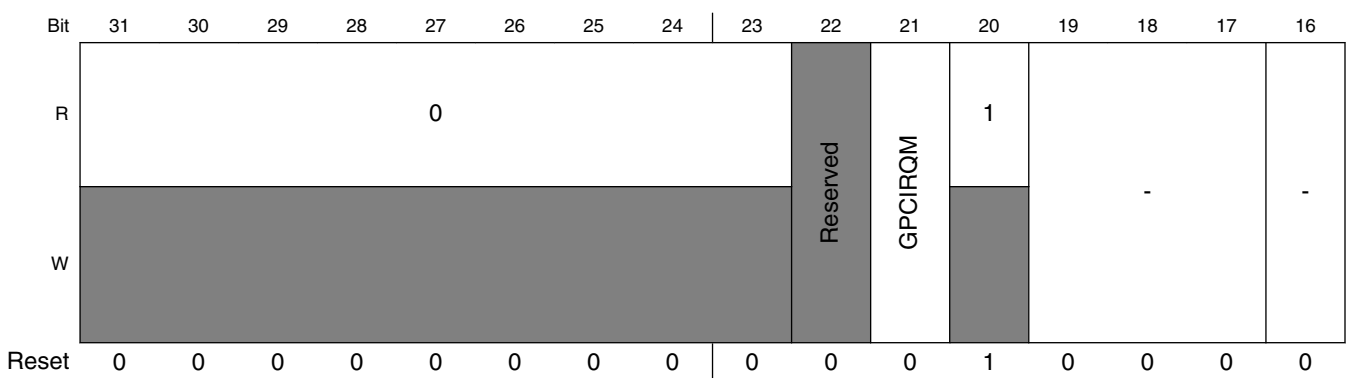
Detailed descriptions of each register can be found below.

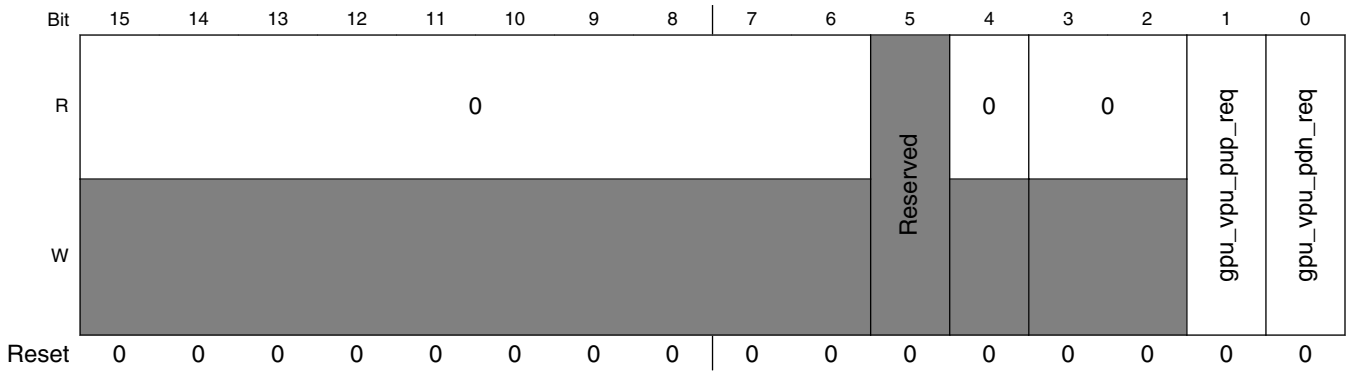
GPC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20D_C000	GPC Interface control register (GPC_CNTR)	32	R/W	0010_0000h	28.5.1/1499
20D_C008	IRQ masking register 1 (GPC_IMR1)	32	R/W	0000_0000h	28.5.2/1501
20D_C00C	IRQ masking register 2 (GPC_IMR2)	32	R/W	0000_0000h	28.5.3/1501
20D_C010	IRQ masking register 3 (GPC_IMR3)	32	R/W	0000_0000h	28.5.4/1501
20D_C014	IRQ masking register 4 (GPC_IMR4)	32	R/W	0000_0000h	28.5.5/1502
20D_C018	IRQ status resister 1 (GPC_ISR1)	32	R	0000_0000h	28.5.6/1502
20D_C01C	IRQ status resister 2 (GPC_ISR2)	32	R	0000_0000h	28.5.7/1503
20D_C020	IRQ status resister 3 (GPC_ISR3)	32	R	0000_0000h	28.5.8/1503
20D_C024	IRQ status resister 4 (GPC_ISR4)	32	R	0000_0000h	28.5.9/1504

28.5.1 GPC Interface control register (GPC_CNTR)

Address: 20D_C000h base + 0h offset = 20D_C000h





GPC_CNTR field descriptions

Field	Description
31–23 Reserved	This read-only field is reserved and always has the value 0.
22 -	This field is reserved. Reserved
21 GPCIRQM	GPC interrupt/event masking 1 interrupt/event is masked 0 not masked
20 Reserved	This read-only field is reserved and always has the value 1.
19–17 -	Reserved This bitfield is readable/writeable, but avoid writing to this reserved bitfield.
16 -	Reserved This bitfield is readable/writeable, but avoid writing to this reserved bitfield.
15–6 Reserved	This read-only field is reserved and always has the value 0.
5 -	This field is reserved. Reserved
4 Reserved	This read-only field is reserved and always has the value 0.
3–2 Reserved	This read-only field is reserved and always has the value 0.
1 gpu_vpu_pup_req	GPU/VPU Power Up request. Self-cleared bit. * Note: Power switch for GPU/VPU power domain is controlled by ANALOG configuration, not GPU/VPU PGC signals 1 Request Power Up sequence to start for GPU/VPU 0 no request
0 gpu_vpu_pdn_req	GPU/VPU Power Down request. Self-cleared bit. * Note: Power switch for GPU/VPU power domain is controlled by ANALOG configuration, not GPU/VPU PGC signals 1 Request Power Down sequence to start for GPU/VPU 0 no request

28.5.2 IRQ masking register 1 (GPC_IMR1)

IMR1 Register - masking of irq[63:32].

Address: 20D_C000h base + 8h offset = 20D_C008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR1 field descriptions

Field	Description
IMR1	IRQ[63:32] masking bits: 1-irq masked, 0-irq is not masked

28.5.3 IRQ masking register 2 (GPC_IMR2)

IMR2 Register - masking of irq[95:64].

Address: 20D_C000h base + Ch offset = 20D_C00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR2 field descriptions

Field	Description
IMR2	IRQ[95:64] masking bits: 1-irq masked, 0-irq is not masked

28.5.4 IRQ masking register 3 (GPC_IMR3)

IMR3 Register - masking of irq[127:96].

Address: 20D_C000h base + 10h offset = 20D_C010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR3																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR3 field descriptions

Field	Description
IMR3	IRQ[127:96] masking bits: 1-irq masked, 0-irq is not masked

28.5.5 IRQ masking register 4 (GPC_IMR4)

IMR4 Register - masking of irq[159:128].

Address: 20D_C000h base + 14h offset = 20D_C014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR4																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR4 field descriptions

Field	Description
IMR4	IRQ[159:128] masking bits: 1-irq masked, 0-irq is not masked

28.5.6 IRQ status resister 1 (GPC_ISR1)

ISR1 Register - status of irq [63:32].

Address: 20D_C000h base + 18h offset = 20D_C018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR1																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

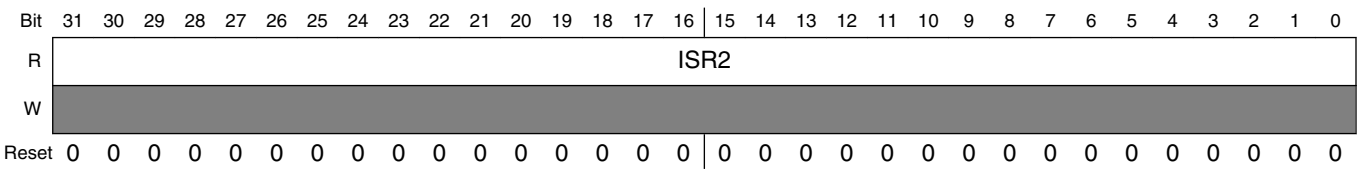
GPC_ISR1 field descriptions

Field	Description
ISR1	IRQ[63:32] status, read only

28.5.7 IRQ status resister 2 (GPC_ISR2)

ISR2 Register - status of irq [95:64].

Address: 20D_C000h base + 1Ch offset = 20D_C01Ch



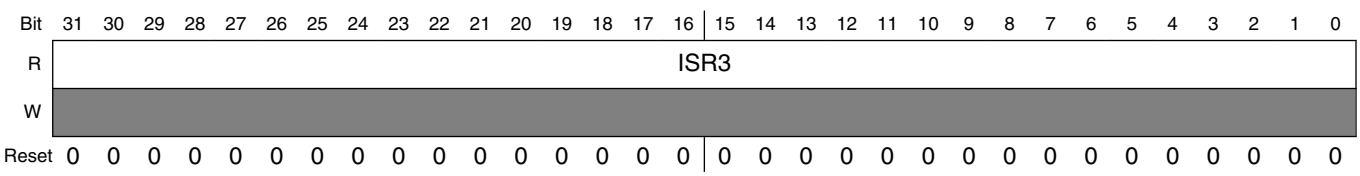
GPC_ISR2 field descriptions

Field	Description
ISR2	IRQ[95:64] status, read only

28.5.8 IRQ status resister 3 (GPC_ISR3)

ISR3 Register - status of irq [127:96].

Address: 20D_C000h base + 20h offset = 20D_C020h



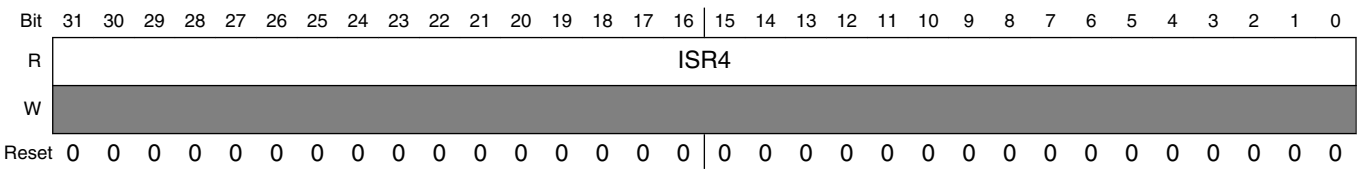
GPC_ISR3 field descriptions

Field	Description
ISR3	IRQ[127:96] status, read only

28.5.9 IRQ status resister 4 (GPC_ISR4)

ISR4 Register - status of irq [159:128].

Address: 20D_C000h base + 24h offset = 20D_C024h



GPC_ISR4 field descriptions

Field	Description
ISR4	IRQ[159:128] status, read only

28.6 PGC Memory Map/Register Definition

The PGC registers can be accessed only in supervisor mode.

Attempts to access registers when not in supervisor mode or attempts to access an unimplemented address location might trigger a bus transfer error. (The hardware asserts the signal ips_xfr_err if the PGC has been integrated with resp_sel tied low.) In this case, software should take appropriate action (such as ignore the error, log the error, or initiate a soft reset).

All PGC registers are byte-accessible.

NOTE

The base address of each PGC module instantiation is specified in the GPC module. Absolute address values will be calculated by [GPC base address] + [PGC CPU/GPU/DISPLAY Offset].

PGC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20D_C260	PGC GPU PGC Control Register (PGC_GPU_CTRL)	32	R/W	0000_0000h	28.6.1/1505

Table continues on the next page...

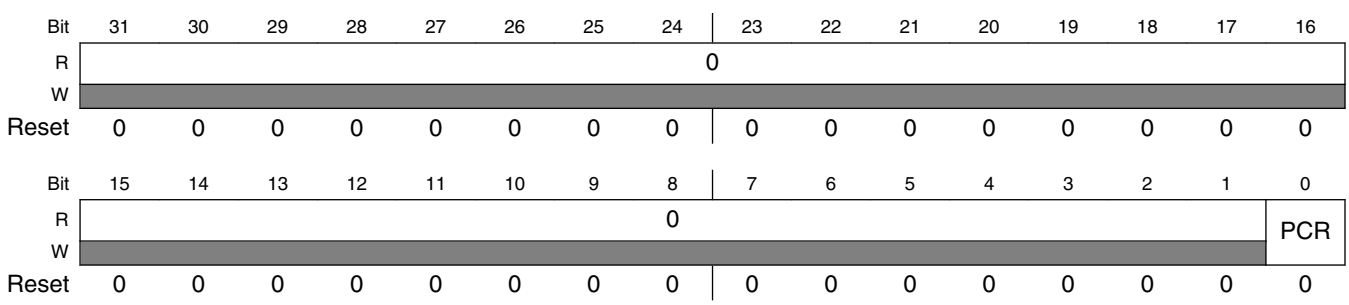
PGC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20D_C264	PGC GPU Power Up Sequence Control Register (PGC_GPU_PUPSCR)	32	R/W	0000_0F01h	28.6.2/1506
20D_C268	PGC GPU Pull Down Sequence Control Register (PGC_GPU_PDNSCR)	32	R/W	0000_0101h	28.6.3/1506
20D_C26C	PGC GPU Power Gating Controller Status Register (PGC_GPU_SR)	32	R/W	0000_0000h	28.6.4/1507
20D_C2A0	PGC CPU Control Register (PGC_CPU_CTRL)	32	R/W	0000_0000h	28.6.5/1508
20D_C2A4	PGC CPU Power Up Sequence Control Register (PGC_CPU_PUPSCR)	32	R/W	0000_0F01h	28.6.6/1508
20D_C2A8	PGC CPU Pull Down Sequence Control Register (PGC_CPU_PDNSCR)	32	R/W	0000_0101h	28.6.7/1509
20D_C2AC	PGC CPU Power Gating Controller Status Register (PGC_CPU_SR)	32	R/W	0000_0000h	28.6.8/1510

28.6.1 PGC GPU PGC Control Register (PGC_GPU_CTRL)

The PGCR enables the response to a power-down request.

Address: 20D_C000h base + 260h offset = 20D_C260h



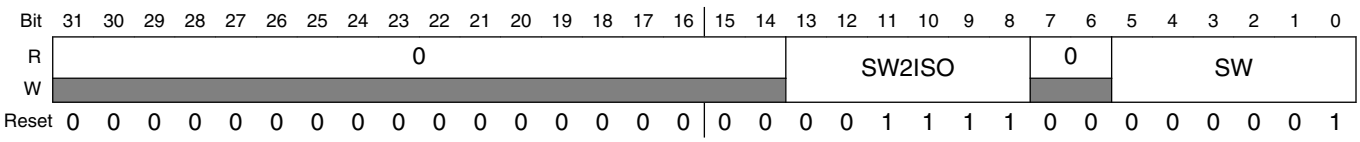
PGC_GPU_CTRL field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 PCR	Power Control **NOTE:** PCR must not change from power-down request (pdn_req) assertion until the target subsystem is completely powered up. 0 Do not switch off power even if pdn_req is asserted. 1 Switch off power when pdn_req is asserted.

28.6.2 PGC GPU Power Up Sequence Control Register (PGC_GPU_PUPSCR)

The PUPSCR contains the power-up timing parameters.

Address: 20D_C000h base + 264h offset = 20D_C264h



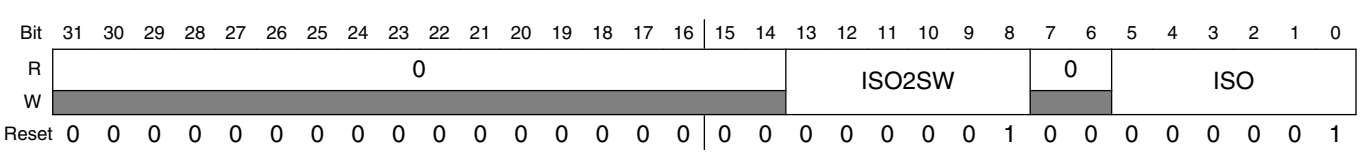
PGC_GPU_PUPSCR field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
13–8 SW2ISO	After asserting power toggle on/off signal (switch_b), the PGC waits a number of IPG clocks equal to the value of SW2ISO before negating isolation. **NOTE:** SW2ISO must not be programmed to zero.
7–6 Reserved	This read-only field is reserved and always has the value 0.
SW	After a power-up request (pup_req assertion), the PGC waits a number of IPG clocks equal to the value of SW before asserting power toggle on/off signal (switch_b). **NOTE:** SW must not be programmed to zero. **NOTE:** The PGC clock is generated from the IPG_CLK_ROOT. for frequency configuration of the IPG_CLK_ROOT. See Clock Controller Module (CCM).

28.6.3 PGC GPU Pull Down Sequence Control Register (PGC_GPU_PDNSCR)

The SR contains the status parameters.

Address: 20D_C000h base + 268h offset = 20D_C268h



PGC_GPU_PDNSCR field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
13–8 ISO2SW	After asserting isolation, the PGC waits a number of IPG clocks equal to the value of ISO2SW before negating power toggle on/off signal (switch_b). **NOTE:** ISO2SW must not be programmed to zero.
7–6 Reserved	This read-only field is reserved and always has the value 0.
ISO	After a power-down request (pdn_req assertion), the PGC waits a number of IPG clocks equal to the value of ISO before asserting isolation. **NOTE:** ISO must not be programmed to zero.

28.6.4 PGC GPU Power Gating Controller Status Register (PGC_GPU_SR)

The SR contains the status parameters.

Address: 20D_C000h base + 26Ch offset = 20D_C26Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															PSR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

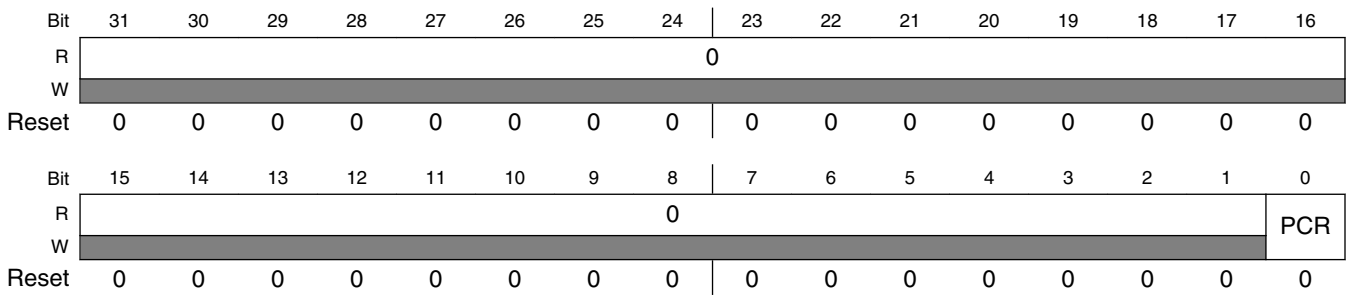
PGC_GPU_SR field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 PSR	Power status. When in functional (or software-controlled debug) mode, PGC hardware sets PSR as soon as any of the power control output changes its state to one. Write one to clear this bit. Software should clear this bit after power up; otherwise, PSR continues to reflect the power status of the initial power down. 0 The target subsystem was not powered down for the previous power-down request. 1 The target subsystem was powered down for the previous power-down request.

28.6.5 PGC CPU Control Register (PGC_CPU_CTRL)

The PGCR enables the response to a power-down request.

Address: 20D_C000h base + 2A0h offset = 20D_C2A0h



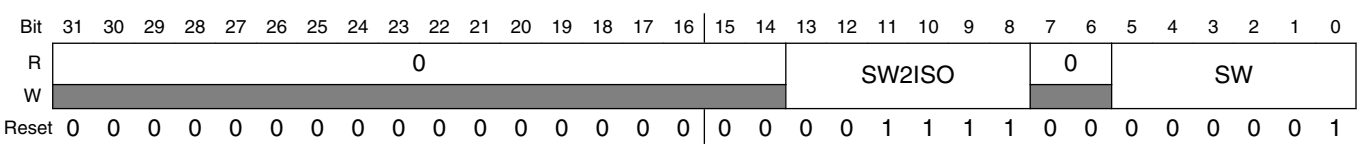
PGC_CPU_CTRL field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 PCR	Power Control **NOTE:** PCR must not change from power-down request (pdn_req) assertion until the target subsystem is completely powered up. 0 Do not switch off power even if pdn_req is asserted. 1 Switch off power when pdn_req is asserted.

28.6.6 PGC CPU Power Up Sequence Control Register (PGC_CPU_PUPSCR)

The PUPSCR contains the power-up timing parameters.

Address: 20D_C000h base + 2A4h offset = 20D_C2A4h



PGC_CPU_PUPSCR field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
13–8 SW2ISO	After asserting , the PGC waits a number of 32k clocks equal to the value of SW2ISO before negating isolation. **NOTE:** SW2ISO must not be programmed to zero. The SW2ISO value should be chosen such that the delay before negating isolation is greater than the LDO ramp-up time.
7–6 Reserved	This read-only field is reserved and always has the value 0.
SW	After a power-up request (pup_req assertion), the PGC waits a number of 32k clocks equal to the value of SW before asserting. **NOTE:** SW must not be programmed to zero.

28.6.7 PGC CPU Pull Down Sequence Control Register (PGC_CPU_PDNSCR)

The PDNSCR contains the power-down timing parameters.

Address: 20D_C000h base + 2A8h offset = 20D_C2A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																ISO2SW								0		ISO					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

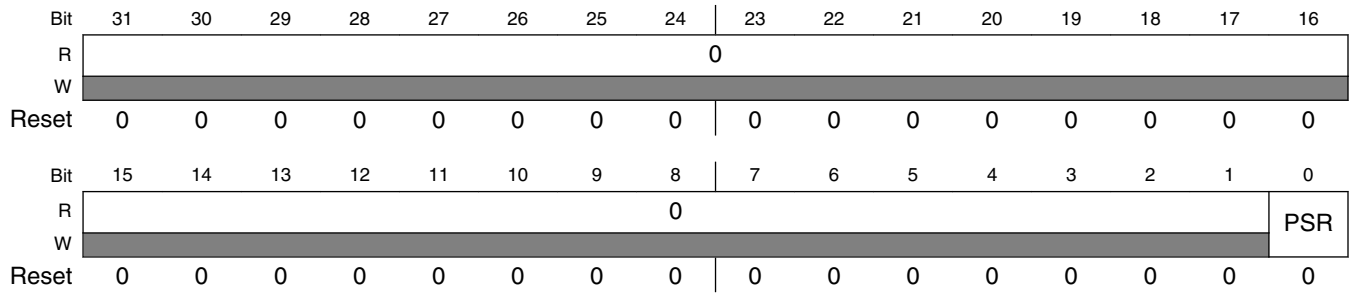
PGC_CPU_PDNSCR field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
13–8 ISO2SW	After asserting isolation, the PGC waits a number of 32k clocks equal to the value of ISO2SW before negating. **NOTE:** ISO2SW must not be programmed to zero.
7–6 Reserved	This read-only field is reserved and always has the value 0.
ISO	After a power-down request (pdn_req assertion), the PGC waits a number of 32k clocks equal to the value of ISO before asserting isolation. **NOTE:** ISO must not be programmed to zero.

28.6.8 PGC CPU Power Gating Controller Status Register (PGC_CPU_SR)

The SR contains the status parameters.

Address: 20D_C000h base + 2ACh offset = 20D_C2ACh



PGC_CPU_SR field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 PSR	Power status. When in functional (or software-controlled debug) mode, PGC hardware sets PSR as soon as any of the power control output changes its state to one (isolation or power off occurs). Write one to clear this bit. Software should clear this bit after power up; otherwise, PSR continues to reflect the power status of the initial power down. 0 The target subsystem was not powered down for the previous power-down request. 1 The target subsystem was powered down for the previous power-down request.

Chapter 29
General Purpose Input/Output (GPIO)

29.1 Overview

The GPIO general-purpose input/output peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs.

When configured as an output, it is possible to write to an internal register to control the state driven on the output pin. When configured as an input, it is possible to detect the state of the input by reading the state of an internal register. In addition, the GPIO peripheral can produce CORE interrupts.

The GPIO is one of the blocks controlling the IOMUX of the chip.

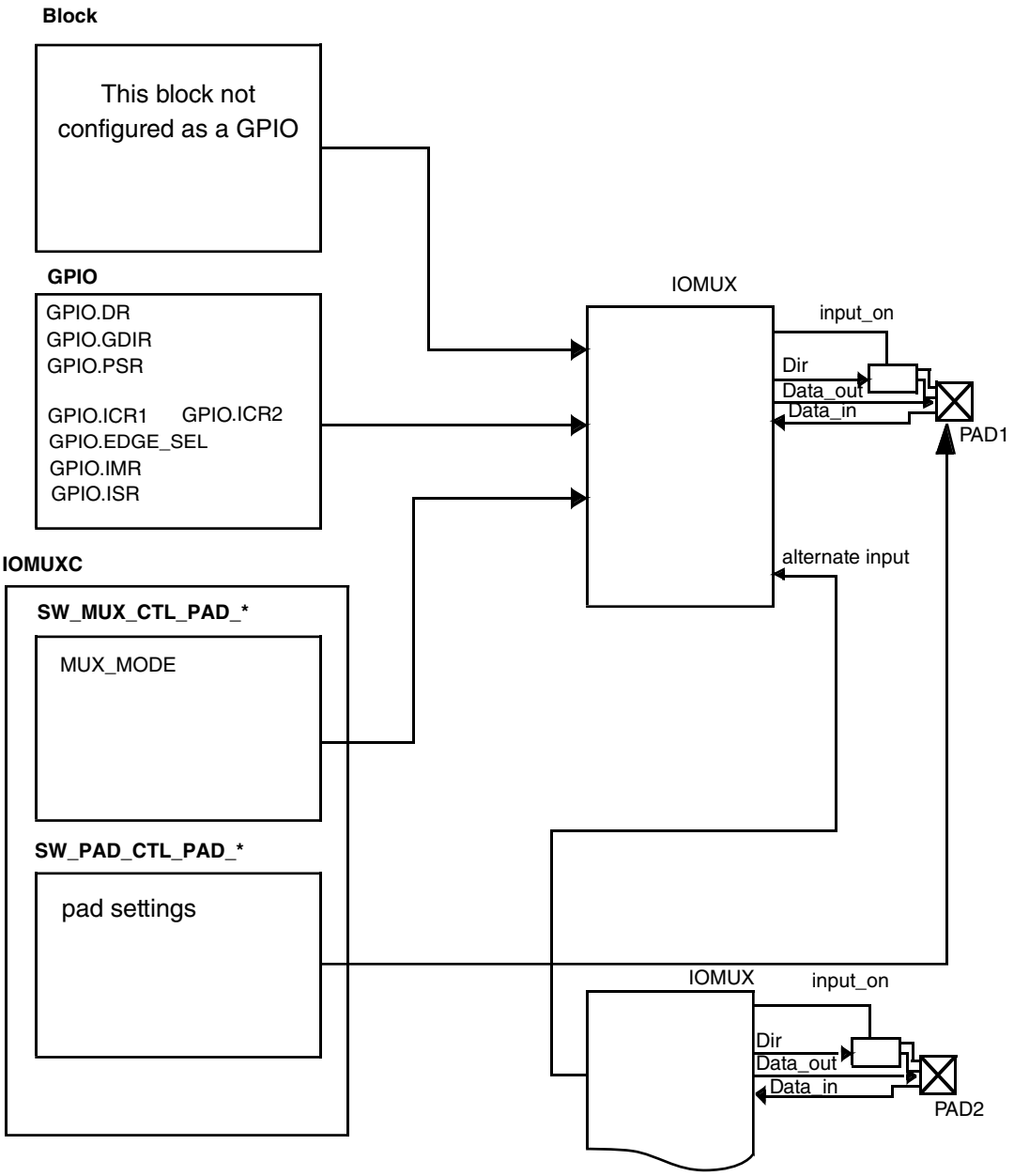


Figure 29-1. Chip IOMUX Scheme

The GPIO functionality is provided through eight registers, an edge-detect circuit, and interrupt generation logic.

The eight registers are:

- Data register (GPIO_DR)
- GPIO direction register (GPIO_GDIR)
- Pad sample register (GPIO_PSR)
- Interrupt control registers (GPIO_ICR1, GPIO_ICR2)

- Edge select register (GPIO_EDGE_SEL)
- Interrupt mask register (GPIO_IMR)
- Interrupt status register (GPIO_ISR)

These registers are described in detail in GPIO Memory Map/Register Definition.

Each GPIO input has a dedicated edge-detect circuit which can be configured through software to detect rising edges, falling edges, logic low-levels or logic high-levels on the input signals. The outputs of the edge detect circuits are optionally masked by setting the corresponding bit in the interrupt mask register (GPIO_IMR). These qualified outputs are OR'ed together to generate two one-bit interrupt lines:

- Combined interrupt indication for GPIOx signals 0 - 15
- Combined interrupt indication for GPIOx signals 16 - 31

In addition, GPIO1 provides visibility to each of its 8 low order interrupt sources (i.e. GPIO1 interrupt n, for n = 0 – 7). However, individual interrupt indications from other GPIOx are not available.

The GPIO edge detection is described further in Interrupt Control Unit.

The GPIO's overall functionality is described further in GPIO Functional Description.

29.1.1 Block Diagram

The GPIO subsystem contains multiple GPIO blocks, which can generate and control up to 32 signals for general purpose.

A block diagram of the GPIO is shown in Figure 29-2

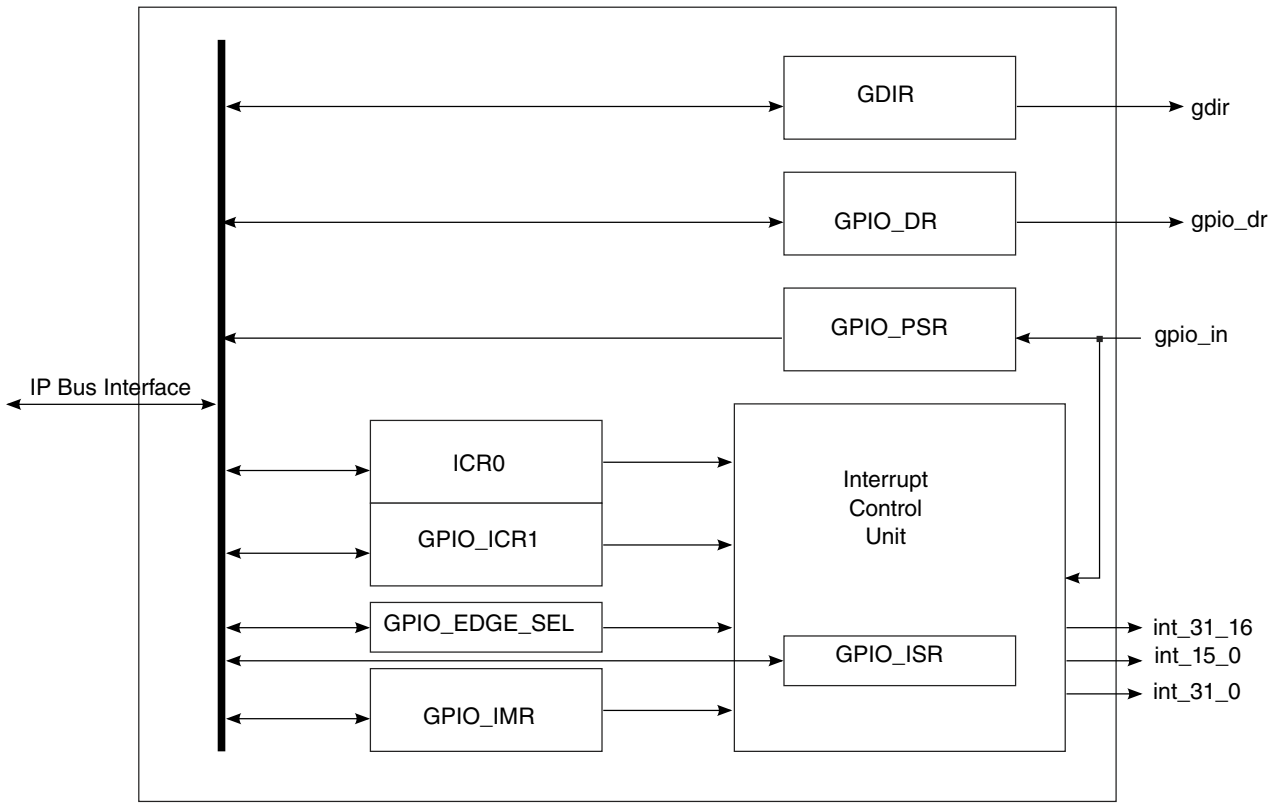


Figure 29-2. GPIO Block Diagram

29.1.2 Features

The GPIO includes the following features:

- General purpose input/output logic capabilities:
  - Drives specific data to output using the data register (GPIO_DR)
  - Controls the direction of the signal using the GPIO direction register (GPIO_GDIR)
  - Enables the core to sample the status of the corresponding inputs by reading the pad sample register (GPIO_PSR).
- GPIO interrupt capabilities:
  - Supports up to 32 interrupts
  - Identifies interrupt edges
  - Generates three active-high interrupts to the SoC interrupt controller

29.2 External Signals

The tables found here describe the external signals of GPIO.

Table 29-1. GPIO1 External Signals

Signal	Description	Pad	Mode	Direction
GPIO1_IO00	-	GPIO_0	ALT5	IO
GPIO1_IO01	-	GPIO_1	ALT5	IO
GPIO1_IO02	-	GPIO_2	ALT5	IO
GPIO1_IO03	-	GPIO_3	ALT5	IO
GPIO1_IO04	-	GPIO_4	ALT5	IO
GPIO1_IO05	-	GPIO_5	ALT5	IO
GPIO1_IO06	-	GPIO_6	ALT5	IO
GPIO1_IO07	-	GPIO_7	ALT5	IO
GPIO1_IO08	-	GPIO_8	ALT5	IO
GPIO1_IO09	-	GPIO_9	ALT5	IO
GPIO1_IO10	-	SD2_CLK	ALT5	IO
GPIO1_IO11	-	SD2_CMD	ALT5	IO
GPIO1_IO12	-	SD2_DAT3	ALT5	IO
GPIO1_IO13	-	SD2_DAT2	ALT5	IO
GPIO1_IO14	-	SD2_DAT1	ALT5	IO
GPIO1_IO15	-	SD2_DAT0	ALT5	IO
GPIO1_IO16	-	SD1_DAT0	ALT5	IO
GPIO1_IO17	-	SD1_DAT1	ALT5	IO
GPIO1_IO18	-	SD1_CMD	ALT5	IO
GPIO1_IO19	-	SD1_DAT2	ALT5	IO
GPIO1_IO20	-	SD1_CLK	ALT5	IO
GPIO1_IO21	-	SD1_DAT3	ALT5	IO
GPIO1_IO22	-	ENET_MDIO	ALT5	IO
GPIO1_IO23	-	ENET_REF_CLK	ALT5	IO
GPIO1_IO24	-	ENET_RX_ER	ALT5	IO
GPIO1_IO25	-	ENET_CRS_DV	ALT5	IO
GPIO1_IO26	-	ENET_RXD1	ALT5	IO
GPIO1_IO27	-	ENET_RXD0	ALT5	IO
GPIO1_IO28	-	ENET_TX_EN	ALT5	IO
GPIO1_IO29	-	ENET_TXD1	ALT5	IO
GPIO1_IO30	-	ENET_TXD0	ALT5	IO
GPIO1_IO31	-	ENET_MDC	ALT5	IO

Table 29-2. GPIO2 External Signals

Signal	Description	Pad	Mode	Direction
GPIO2_IO00	-	NANDF_D0	ALT5	IO
GPIO2_IO01	-	NANDF_D1	ALT5	IO
GPIO2_IO02	-	NANDF_D2	ALT5	IO
GPIO2_IO03	-	NANDF_D3	ALT5	IO
GPIO2_IO04	-	NANDF_D4	ALT5	IO
GPIO2_IO05	-	NANDF_D5	ALT5	IO
GPIO2_IO06	-	NANDF_D6	ALT5	IO
GPIO2_IO07	-	NANDF_D7	ALT5	IO
GPIO2_IO08	-	SD4_DAT0	ALT5	IO
GPIO2_IO09	-	SD4_DAT1	ALT5	IO
GPIO2_IO10	-	SD4_DAT2	ALT5	IO
GPIO2_IO11	-	SD4_DAT3	ALT5	IO
GPIO2_IO12	-	SD4_DAT4	ALT5	IO
GPIO2_IO13	-	SD4_DAT5	ALT5	IO
GPIO2_IO14	-	SD4_DAT6	ALT5	IO
GPIO2_IO15	-	SD4_DAT7	ALT5	IO
GPIO2_IO16	-	EIM_A22	ALT5	IO
GPIO2_IO17	-	EIM_A21	ALT5	IO
GPIO2_IO18	-	EIM_A20	ALT5	IO
GPIO2_IO19	-	EIM_A19	ALT5	IO
GPIO2_IO20	-	EIM_A18	ALT5	IO
GPIO2_IO21	-	EIM_A17	ALT5	IO
GPIO2_IO22	-	EIM_A16	ALT5	IO
GPIO2_IO23	-	EIM_CS0	ALT5	IO
GPIO2_IO24	-	EIM_CS1	ALT5	IO
GPIO2_IO25	-	EIM_OE	ALT5	IO
GPIO2_IO26	-	EIM_RW	ALT5	IO
GPIO2_IO27	-	EIM_LBA	ALT5	IO
GPIO2_IO28	-	EIM_EB0	ALT5	IO
GPIO2_IO29	-	EIM_EB1	ALT5	IO
GPIO2_IO30	-	EIM_EB2	ALT5	IO
GPIO2_IO31	-	EIM_EB3	ALT5	IO

Table 29-3. GPIO3 External Signals

Signal	Description	Pad	Mode	Direction
GPIO3_IO00	-	EIM_DA0	ALT5	IO
GPIO3_IO01	-	EIM_DA1	ALT5	IO
GPIO3_IO02	-	EIM_DA2	ALT5	IO

Table continues on the next page...

Table 29-3. GPIO3 External Signals (continued)

Signal	Description	Pad	Mode	Direction
GPIO3_IO03	-	EIM_DA3	ALT5	IO
GPIO3_IO04	-	EIM_DA4	ALT5	IO
GPIO3_IO05	-	EIM_DA5	ALT5	IO
GPIO3_IO06	-	EIM_DA6	ALT5	IO
GPIO3_IO07	-	EIM_DA7	ALT5	IO
GPIO3_IO08	-	EIM_DA8	ALT5	IO
GPIO3_IO09	-	EIM_DA9	ALT5	IO
GPIO3_IO10	-	EIM_DA10	ALT5	IO
GPIO3_IO11	-	EIM_DA11	ALT5	IO
GPIO3_IO12	-	EIM_DA12	ALT5	IO
GPIO3_IO13	-	EIM_DA13	ALT5	IO
GPIO3_IO14	-	EIM_DA14	ALT5	IO
GPIO3_IO15	-	EIM_DA15	ALT5	IO
GPIO3_IO16	-	EIM_D16	ALT5	IO
GPIO3_IO17	-	EIM_D17	ALT5	IO
GPIO3_IO18	-	EIM_D18	ALT5	IO
GPIO3_IO19	-	EIM_D19	ALT5	IO
GPIO3_IO20	-	EIM_D20	ALT5	IO
GPIO3_IO21	-	EIM_D21	ALT5	IO
GPIO3_IO22	-	EIM_D22	ALT5	IO
GPIO3_IO23	-	EIM_D23	ALT5	IO
GPIO3_IO24	-	EIM_D24	ALT5	IO
GPIO3_IO25	-	EIM_D25	ALT5	IO
GPIO3_IO26	-	EIM_D26	ALT5	IO
GPIO3_IO27	-	EIM_D27	ALT5	IO
GPIO3_IO28	-	EIM_D28	ALT5	IO
GPIO3_IO29	-	EIM_D29	ALT5	IO
GPIO3_IO30	-	EIM_D30	ALT5	IO
GPIO3_IO31	-	EIM_D31	ALT5	IO

Table 29-4. GPIO4 External Signals

Signal	Description	Pad	Mode	Direction
GPIO4_IO05	-	GPIO_19	ALT5	IO
GPIO4_IO06	-	KEY_COL0	ALT5	IO
GPIO4_IO07	-	KEY_ROW0	ALT5	IO
GPIO4_IO08	-	KEY_COL1	ALT5	IO
GPIO4_IO09	-	KEY_ROW1	ALT5	IO
GPIO4_IO10	-	KEY_COL2	ALT5	IO

Table continues on the next page...

Table 29-4. GPIO4 External Signals (continued)

Signal	Description	Pad	Mode	Direction
GPIO4_IO11	-	KEY_ROW2	ALT5	IO
GPIO4_IO12	-	KEY_COL3	ALT5	IO
GPIO4_IO13	-	KEY_ROW3	ALT5	IO
GPIO4_IO14	-	KEY_COL4	ALT5	IO
GPIO4_IO15	-	KEY_ROW4	ALT5	IO
GPIO4_IO16	-	DI0_DISP_CLK	ALT5	IO
GPIO4_IO17	-	DI0_PIN15	ALT5	IO
GPIO4_IO18	-	DI0_PIN2	ALT5	IO
GPIO4_IO19	-	DI0_PIN3	ALT5	IO
GPIO4_IO20	-	DI0_PIN4	ALT5	IO
GPIO4_IO21	-	DISP0_DAT0	ALT5	IO
GPIO4_IO22	-	DISP0_DAT1	ALT5	IO
GPIO4_IO23	-	DISP0_DAT2	ALT5	IO
GPIO4_IO24	-	DISP0_DAT3	ALT5	IO
GPIO4_IO25	-	DISP0_DAT4	ALT5	IO
GPIO4_IO26	-	DISP0_DAT5	ALT5	IO
GPIO4_IO27	-	DISP0_DAT6	ALT5	IO
GPIO4_IO28	-	DISP0_DAT7	ALT5	IO
GPIO4_IO29	-	DISP0_DAT8	ALT5	IO
GPIO4_IO30	-	DISP0_DAT9	ALT5	IO
GPIO4_IO31	-	DISP0_DAT10	ALT5	IO

Table 29-5. GPIO5 External Signals

Signal	Description	Pad	Mode	Direction
GPIO5_IO00	-	EIM_WAIT	ALT5	IO
GPIO5_IO02	-	EIM_A25	ALT5	IO
GPIO5_IO04	-	EIM_A24	ALT5	IO
GPIO5_IO05	-	DISP0_DAT11	ALT5	IO
GPIO5_IO06	-	DISP0_DAT12	ALT5	IO
GPIO5_IO07	-	DISP0_DAT13	ALT5	IO
GPIO5_IO08	-	DISP0_DAT14	ALT5	IO
GPIO5_IO09	-	DISP0_DAT15	ALT5	IO
GPIO5_IO10	-	DISP0_DAT16	ALT5	IO
GPIO5_IO11	-	DISP0_DAT17	ALT5	IO
GPIO5_IO12	-	DISP0_DAT18	ALT5	IO
GPIO5_IO13	-	DISP0_DAT19	ALT5	IO
GPIO5_IO14	-	DISP0_DAT20	ALT5	IO
GPIO5_IO15	-	DISP0_DAT21	ALT5	IO

Table continues on the next page...

Table 29-5. GPIO5 External Signals (continued)

Signal	Description	Pad	Mode	Direction
GPIO5_IO16	-	DISP0_DAT22	ALT5	IO
GPIO5_IO17	-	DISP0_DAT23	ALT5	IO
GPIO5_IO18	-	CSI0_PIXCLK	ALT5	IO
GPIO5_IO19	-	CSI0_MCLK	ALT5	IO
GPIO5_IO20	-	CSI0_DATA_EN	ALT5	IO
GPIO5_IO21	-	CSI0_VSYNC	ALT5	IO
GPIO5_IO22	-	CSI0_DAT4	ALT5	IO
GPIO5_IO23	-	CSI0_DAT5	ALT5	IO
GPIO5_IO24	-	CSI0_DAT6	ALT5	IO
GPIO5_IO25	-	CSI0_DAT7	ALT5	IO
GPIO5_IO26	-	CSI0_DAT8	ALT5	IO
GPIO5_IO27	-	CSI0_DAT9	ALT5	IO
GPIO5_IO28	-	CSI0_DAT10	ALT5	IO
GPIO5_IO29	-	CSI0_DAT11	ALT5	IO
GPIO5_IO30	-	CSI0_DAT12	ALT5	IO
GPIO5_IO31	-	CSI0_DAT13	ALT5	IO

Table 29-6. GPIO6 External Signals

Signal	Description	Pad	Mode	Direction
GPIO6_IO00	-	CSI0_DAT14	ALT5	IO
GPIO6_IO01	-	CSI0_DAT15	ALT5	IO
GPIO6_IO02	-	CSI0_DAT16	ALT5	IO
GPIO6_IO03	-	CSI0_DAT17	ALT5	IO
GPIO6_IO04	-	CSI0_DAT18	ALT5	IO
GPIO6_IO05	-	CSI0_DAT19	ALT5	IO
GPIO6_IO06	-	EIM_A23	ALT5	IO
GPIO6_IO07	-	NANDF_CLE	ALT5	IO
GPIO6_IO08	-	NANDF_ALE	ALT5	IO
GPIO6_IO09	-	NANDF_WP_B	ALT5	IO
GPIO6_IO10	-	NANDF_RB0	ALT5	IO
GPIO6_IO11	-	NANDF_CS0	ALT5	IO
GPIO6_IO14	-	NANDF_CS1	ALT5	IO
GPIO6_IO15	-	NANDF_CS2	ALT5	IO
GPIO6_IO16	-	NANDF_CS3	ALT5	IO
GPIO6_IO17	-	SD3_DAT7	ALT5	IO
GPIO6_IO18	-	SD3_DAT6	ALT5	IO
GPIO6_IO19	-	RGMII_TXC	ALT5	IO
GPIO6_IO20	-	RGMII_TD0	ALT5	IO

Table continues on the next page...

Table 29-6. GPIO6 External Signals (continued)

Signal	Description	Pad	Mode	Direction
GPIO6_IO21	-	RGMII_TD1	ALT5	IO
GPIO6_IO22	-	RGMII_TD2	ALT5	IO
GPIO6_IO23	-	RGMII_TD3	ALT5	IO
GPIO6_IO24	-	RGMII_RX_CTL	ALT5	IO
GPIO6_IO25	-	RGMII_RD0	ALT5	IO
GPIO6_IO26	-	RGMII_TX_CTL	ALT5	IO
GPIO6_IO27	-	RGMII_RD1	ALT5	IO
GPIO6_IO28	-	RGMII_RD2	ALT5	IO
GPIO6_IO29	-	RGMII_RD3	ALT5	IO
GPIO6_IO30	-	RGMII_RXC	ALT5	IO
GPIO6_IO31	-	EIM_BCLK	ALT5	IO

Table 29-7. GPIO7 External Signals

Signal	Description	Pad	Mode	Direction
GPIO7_IO00	-	SD3_DAT5	ALT5	IO
GPIO7_IO01	-	SD3_DAT4	ALT5	IO
GPIO7_IO02	-	SD3_CMD	ALT5	IO
GPIO7_IO03	-	SD3_CLK	ALT5	IO
GPIO7_IO04	-	SD3_DAT0	ALT5	IO
GPIO7_IO05	-	SD3_DAT1	ALT5	IO
GPIO7_IO06	-	SD3_DAT2	ALT5	IO
GPIO7_IO07	-	SD3_DAT3	ALT5	IO
GPIO7_IO08	-	SD3_RST	ALT5	IO
GPIO7_IO09	-	SD4_CMD	ALT5	IO
GPIO7_IO10	-	SD4_CLK	ALT5	IO
GPIO7_IO11	-	GPIO_16	ALT5	IO
GPIO7_IO12	-	GPIO_17	ALT5	IO
GPIO7_IO13	-	GPIO_18	ALT5	IO

29.3 Clocks

The table found here describes the clock sources for GPIO.

Please see the clock controller Module for clock setting, configuration and gating information.

Table 29-8. GPIO Clocks

Clock name	Clock Root	Description
ipg_clk_s	ipg_clk_root	Peripheral access clock

29.4 GPIO Functional Description

This section provides a complete functional description of the block.

29.4.1 GPIO Function

A GPIO signal can operate as a general-purpose input/output when the IOMUX is set to GPIO mode. Each GPIO signal may be independently configured as either an input or an output using the GPIO direction register (GPIO_GDIR).

When configured as an output (GPIO_GDIR bit = 1), the value in the data bit in the GPIO data register (GPIO_DR) is driven on the corresponding GPIO line. When a signal is configured as an input (GPIO_GDIR bit = 0), the state of the input can be read from the corresponding GPIO_PSR bit.

29.4.2 GPIO pad structure

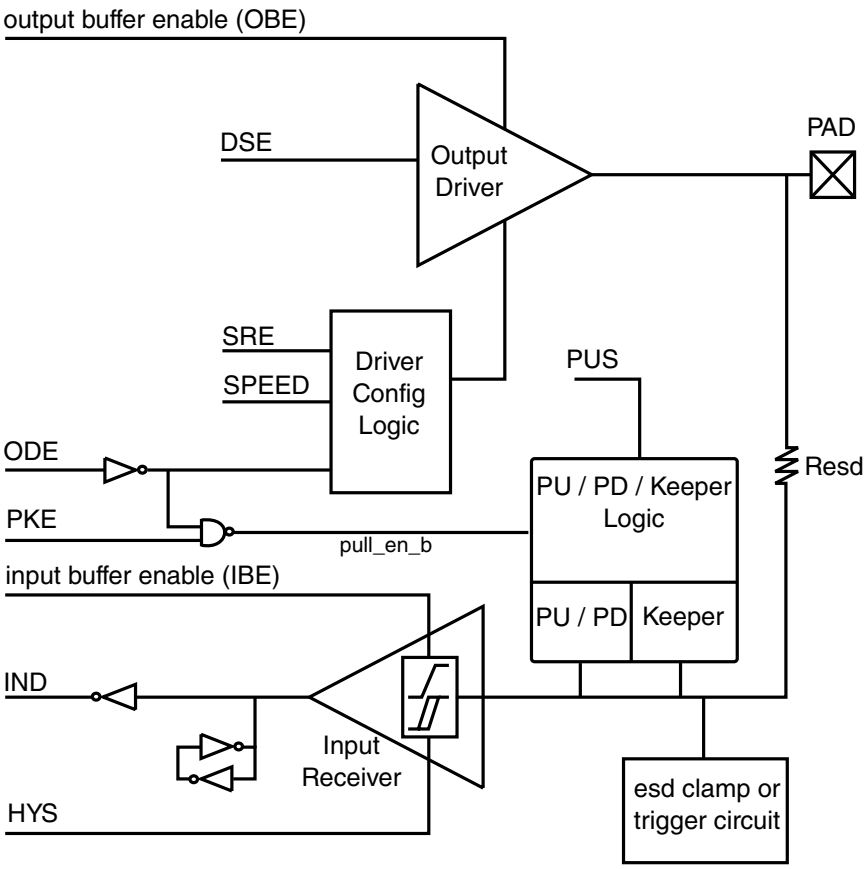


Figure 29-3. GPIO pad functional diagram

29.4.2.1 Input Driver

Input driver characteristics

- Selectable Schmitt trigger or CMOS input mode
- Keeper structure with buffer at the input receiver output to Core
- Receiver is tri-stated when I/O supply (OVDD) is powered down. (Keeper at receiver output keeps its previous state).

29.4.2.1.1 Schmitt trigger

The anti-jamming functionality of the Schmitt trigger is illustrated below.

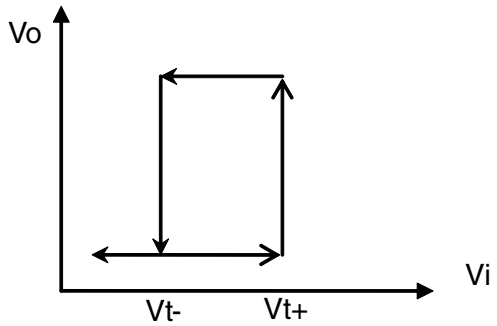


Figure 29-4. Schmitt trigger transfer characteristic

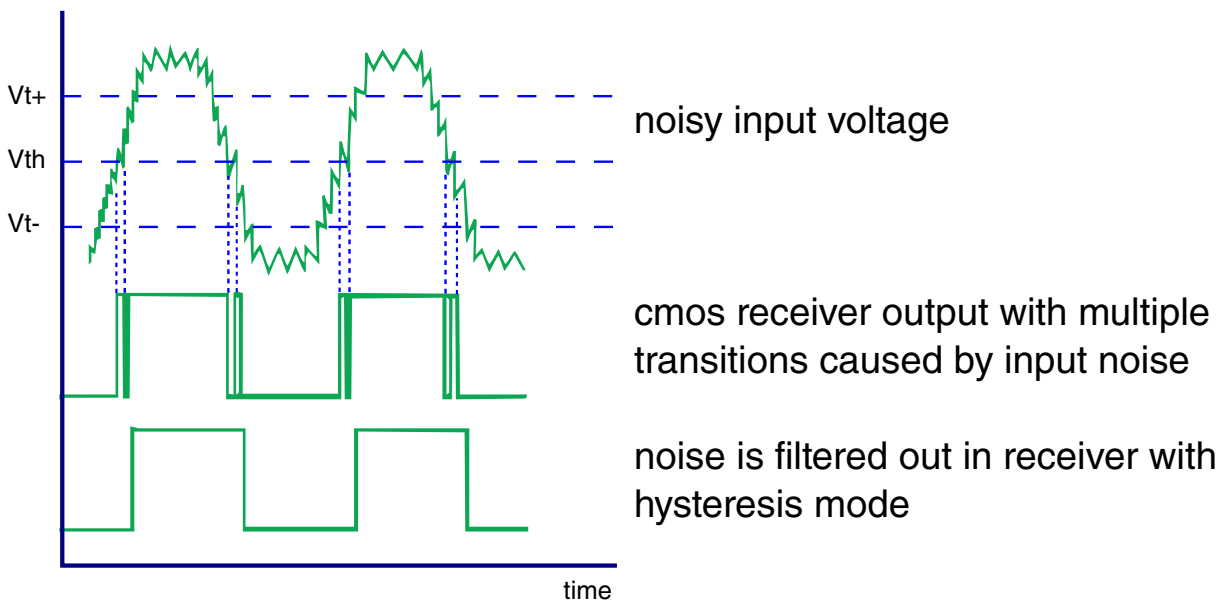


Figure 29-5. Receiver output in CMOS and hysteresis

29.4.2.1.2 Input keeper

A simple latch to hold the input value when OVDD is powered down, or the first inverter is tri-stated. Input buffer's keeper is always enabled for all the pads.

29.4.2.2 Output driver

Output driver characteristics

- Selectable CMOS or open-drain output type

- Selectable pull-keeper enable signal to enable/disable the pull-up/down and output keeper
- Selectable pull-up resistors of 22K, 47K, 100K and a pull-down resistor of 100KOhm. Unsilicided P+ poly resistor is used to limit resistance variation to within ± 20%.
- Internal pull-up, pull-down resistors, and pad keeper are disabled in output mode.
- Seven drive strengths in each operating mode
- Additional 2-bit slew rate control to select between 50, 100, and 200 MHz IO cell operation range with reduced switching noise

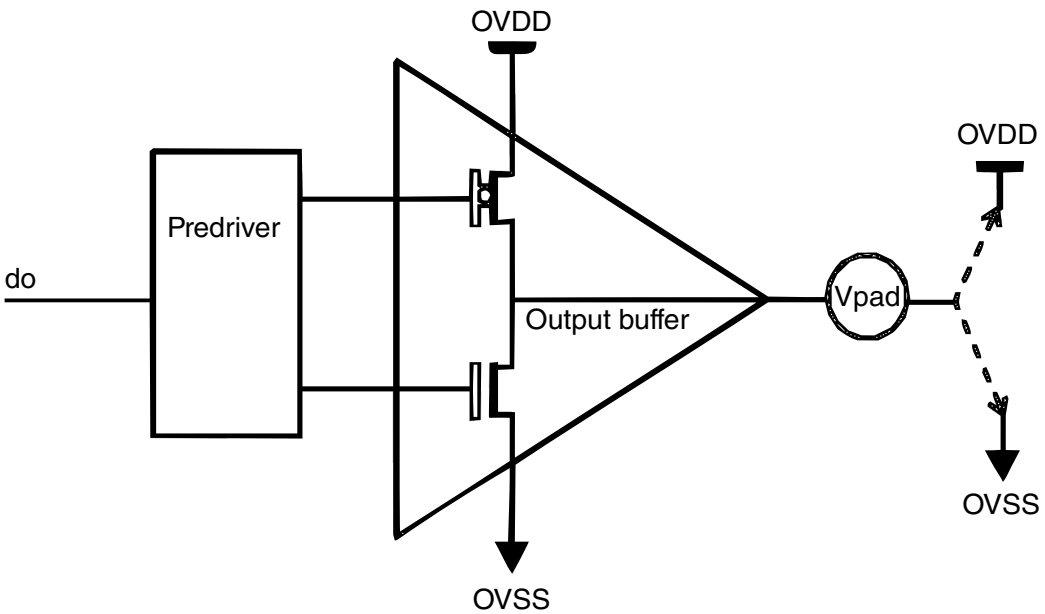


Figure 29-6. Output Driver Functional Diagram

29.4.2.2.1 Drive strength

Drive strength selection can be use to make the impedance matched and get better signal integrity.

29.4.2.2.2 Output keeper

A simple latch to hold the input value.

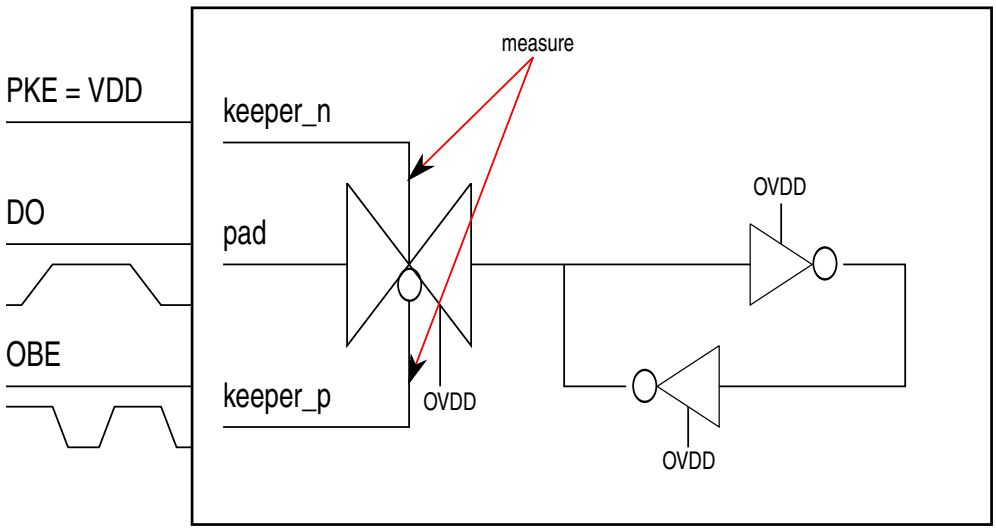


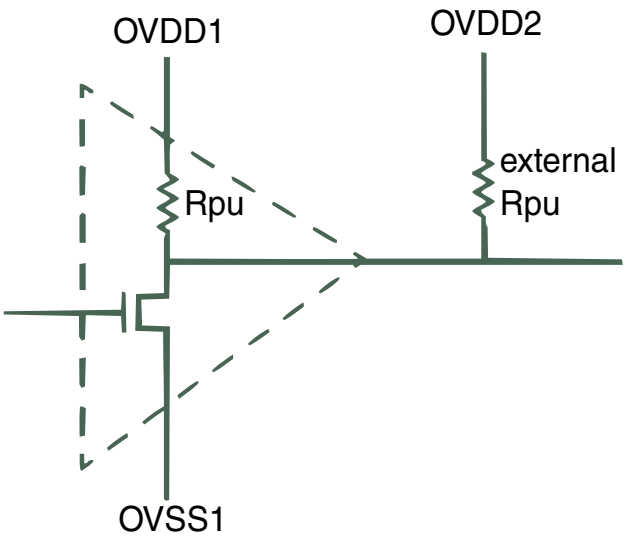
Figure 29-7. Keeper functional diagram

29.4.2.2.3 PU / PD / Keeper Logic

When Keeper is enabled, the pull-up and pull-down are disabled, and the output value of the pad depends on the Keeper. The output keeper is powered by OVDD. When the core VDD is powered down or the first inverter is tri-stated, the pad's state can be kept. Keeper and Pull can't be enabled together.

29.4.2.2.4 Open drain

Open drain is a circuit technique which allows multiple devices to communication over a single wire bi-directionally. Open drain drivers usually operate with an external or internal pull-up resistor that holds the signal line high until a device sinks enough current to pull the line low, usually used for a bus with multiple devices.



If internal pull-up resistor (Rpu) is used, output level will depend on OVDD1

If external Rpu is used, output level will depend on OVDD2

Figure 29-8. Output buffer in open drain mode

29.4.2.3 Operating Frequency

Table 29-9. IO Operating Frequency

DSE Setting	SRE / SPEED Setting	R_fixture (Ohm) OVDD = 1.8V nominal	R_fixture (Ohm) OVDD = 3.3V nominal	Operating Frequency (MHz)
1	0	400	220	50
	1			100
	10			100
	11			100
	100			50
	101			100
	110			100
	111			150
10	0	200	110	50
	1			100
	10			100
	11			100
	100			50
	101			100
	110			100
	111			150
11	0	132	73	50
	1			100

Table continues on the next page...

Table 29-9. IO Operating Frequency (continued)

DSE Setting	SRE / SPEED Setting	R_fixture (Ohm) OVDD = 1.8V nominal	R_fixture (Ohm) OVDD = 3.3V nominal	Operating Frequency (MHz)
	10			100
	11			100
	100			50
	101			100
	110			100
	111			150
100	0	104	58	50
	1			100
	10			100
	11			100
	100			50
	101			100
	110			100
	111			150
101	0	83	46	50
	1			100
	10			100
	11			100
	100			50
	101			100 (OVDD=1.8V nom) 150 (OVDD=3.3V nom)
	110			100 (OVDD=1.8V nom) 150 (OVDD=3.3V nom)
	111			150 (OVDD=1.8V nom) 200 (OVDD=3.3V nom)
110	0	70	38	50
	1			100
	10			100
	11			100
	100			50
	101			150
	110			150
	111			200
111	0	55	32	50
	1			100
	10			100
	11			100
	100			50

Table continues on the next page...

Table 29-9. IO Operating Frequency (continued)

DSE Setting	SRE / SPEED Setting	R_fixture (Ohm) OVDD = 1.8V nominal	R_fixture (Ohm) OVDD = 3.3V nominal	Operating Frequency (MHz)
	101			150
	110			150
	111			200

29.4.3 GPIO Programming

29.4.3.1 GPIO Read Mode

The programming sequence for reading input signals should be as follows:

1. Configure IOMUX to select GPIO mode (Via IOMUX Controller (IOMUXC)).
2. Configure GPIO direction register to input (GPIO_GDIR[GDIR] set to 0b).
3. Read value from data reg*ister*/pad status register.

A pseudocode description to read [input3:input0] values is as follows:

```
// SET INPUTS TO GPIO MODE.
write sw_mux_ctl_<input0>_<input1>_<input2>_<input3>, 32'h00000000
// SET GDIR TO INPUT.
write GDIR[31:4,input3_bit, input2_bit, input1_bit, input0_bit,] 32'hxxxxxxx0
// READ INPUT VALUE FROM DR.
read DR
// READ INPUT VALUE FROM PSR.
read PSR
```

NOTE

While the GPIO direction is set to input (GPIO_GDIR = 0), a read access to GPIO_DR does not return GPIO_DR data. Instead, it returns the GPIO_PSR data, which is the corresponding input signal value.

29.4.3.2 GPIO Write Mode

The programming sequence for driving output signals should be as follows:

1. Configure IOMUX to select GPIO mode (Via IOMUXC), also enable SION if need to read loopback pad value through PSR
2. Configure GPIO direction register to output (GPIO_GDIR[GDIR] set to 1b).
3. Write value to data register (GPIO_DR).

A pseudocode description to drive 4'b0101 on [output3:output0] is as follows:

```
// SET PADS TO GPIO MODE VIA IOMUX.
write sw_mux_ctl_pad_<output[0-3]>.mux_mode, <GPIO_MUX_MODE>
// Enable loopback so we can capture pad value into PSR in output mode
write sw_mux_ctl_pad_<output[0-3]>.sion, 1
// SET GDIR=1 TO OUTPUT BITS.
write GDIR[31:4,output3_bit,output2_bit, output1_bit, output0_bit,] 32'hxxxxxxxF
// WRITE OUTPUT VALUE=4'b0101 TO DR.
write DR, 32'hxxxxxxx5
// READ OUTPUT VALUE FROM PSR ONLY.
read_cmp PSR, 32'hxxxxxxx5
```

29.4.4 Interrupt Control Unit

In addition to the general-purpose input/output function, the edge-detect logic in the GPIO peripheral reflects whether a transition has occurred on a given GPIO signal that is configured as an input (GDIR bit = 0). The interrupt control registers (GPIO_ICR1 and GPIO_ICR2) may be used to independently configure the interrupt condition of each input signal (low-to-high transition, high-to-low transition, low, or high). For information about GPIO_ICR1 and GPIO_ICR2 settings, see GPIO Memory Map/Register Definition.

The interrupt control unit is built of 32 interrupt control subunits, where each subunit handles a single interrupt line.

29.5 GPIO Memory Map/Register Definition

There are eight 32-bit GPIO registers. All registers are accessible from the IP interface. Only 32-bit access is supported.

The GPIO memory map is shown in the following table.

GPIO memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
209_C000	GPIO data register (GPIO1_DR)	32	R/W	0000_0000h	29.5.1/1531
209_C004	GPIO direction register (GPIO1_GDIR)	32	R/W	0000_0000h	29.5.2/1532
209_C008	GPIO pad status register (GPIO1_PSR)	32	R	0000_0000h	29.5.3/1533
209_C00C	GPIO interrupt configuration register1 (GPIO1_ICR1)	32	R/W	0000_0000h	29.5.4/1533
209_C010	GPIO interrupt configuration register2 (GPIO1_ICR2)	32	R/W	0000_0000h	29.5.5/1537

Table continues on the next page...

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
209_C014	GPIO interrupt mask register (GPIO1_IMR)	32	R/W	0000_0000h	29.5.6/1540
209_C018	GPIO interrupt status register (GPIO1_ISR)	32	w1c	0000_0000h	29.5.7/1541
209_C01C	GPIO edge select register (GPIO1_EDGE_SEL)	32	R/W	0000_0000h	29.5.8/1542
20A_0000	GPIO data register (GPIO2_DR)	32	R/W	0000_0000h	29.5.1/1531
20A_0004	GPIO direction register (GPIO2_GDIR)	32	R/W	0000_0000h	29.5.2/1532
20A_0008	GPIO pad status register (GPIO2_PSR)	32	R	0000_0000h	29.5.3/1533
20A_000C	GPIO interrupt configuration register1 (GPIO2_ICR1)	32	R/W	0000_0000h	29.5.4/1533
20A_0010	GPIO interrupt configuration register2 (GPIO2_ICR2)	32	R/W	0000_0000h	29.5.5/1537
20A_0014	GPIO interrupt mask register (GPIO2_IMR)	32	R/W	0000_0000h	29.5.6/1540
20A_0018	GPIO interrupt status register (GPIO2_ISR)	32	w1c	0000_0000h	29.5.7/1541
20A_001C	GPIO edge select register (GPIO2_EDGE_SEL)	32	R/W	0000_0000h	29.5.8/1542
20A_4000	GPIO data register (GPIO3_DR)	32	R/W	0000_0000h	29.5.1/1531
20A_4004	GPIO direction register (GPIO3_GDIR)	32	R/W	0000_0000h	29.5.2/1532
20A_4008	GPIO pad status register (GPIO3_PSR)	32	R	0000_0000h	29.5.3/1533
20A_400C	GPIO interrupt configuration register1 (GPIO3_ICR1)	32	R/W	0000_0000h	29.5.4/1533
20A_4010	GPIO interrupt configuration register2 (GPIO3_ICR2)	32	R/W	0000_0000h	29.5.5/1537
20A_4014	GPIO interrupt mask register (GPIO3_IMR)	32	R/W	0000_0000h	29.5.6/1540
20A_4018	GPIO interrupt status register (GPIO3_ISR)	32	w1c	0000_0000h	29.5.7/1541
20A_401C	GPIO edge select register (GPIO3_EDGE_SEL)	32	R/W	0000_0000h	29.5.8/1542
20A_8000	GPIO data register (GPIO4_DR)	32	R/W	0000_0000h	29.5.1/1531
20A_8004	GPIO direction register (GPIO4_GDIR)	32	R/W	0000_0000h	29.5.2/1532
20A_8008	GPIO pad status register (GPIO4_PSR)	32	R	0000_0000h	29.5.3/1533
20A_800C	GPIO interrupt configuration register1 (GPIO4_ICR1)	32	R/W	0000_0000h	29.5.4/1533
20A_8010	GPIO interrupt configuration register2 (GPIO4_ICR2)	32	R/W	0000_0000h	29.5.5/1537
20A_8014	GPIO interrupt mask register (GPIO4_IMR)	32	R/W	0000_0000h	29.5.6/1540
20A_8018	GPIO interrupt status register (GPIO4_ISR)	32	w1c	0000_0000h	29.5.7/1541
20A_801C	GPIO edge select register (GPIO4_EDGE_SEL)	32	R/W	0000_0000h	29.5.8/1542
20A_C000	GPIO data register (GPIO5_DR)	32	R/W	0000_0000h	29.5.1/1531
20A_C004	GPIO direction register (GPIO5_GDIR)	32	R/W	0000_0000h	29.5.2/1532
20A_C008	GPIO pad status register (GPIO5_PSR)	32	R	0000_0000h	29.5.3/1533
20A_C00C	GPIO interrupt configuration register1 (GPIO5_ICR1)	32	R/W	0000_0000h	29.5.4/1533
20A_C010	GPIO interrupt configuration register2 (GPIO5_ICR2)	32	R/W	0000_0000h	29.5.5/1537
20A_C014	GPIO interrupt mask register (GPIO5_IMR)	32	R/W	0000_0000h	29.5.6/1540
20A_C018	GPIO interrupt status register (GPIO5_ISR)	32	w1c	0000_0000h	29.5.7/1541
20A_C01C	GPIO edge select register (GPIO5_EDGE_SEL)	32	R/W	0000_0000h	29.5.8/1542
20B_0000	GPIO data register (GPIO6_DR)	32	R/W	0000_0000h	29.5.1/1531
20B_0004	GPIO direction register (GPIO6_GDIR)	32	R/W	0000_0000h	29.5.2/1532
20B_0008	GPIO pad status register (GPIO6_PSR)	32	R	0000_0000h	29.5.3/1533

Table continues on the next page...

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20B_000C	GPIO interrupt configuration register1 (GPIO6_ICR1)	32	R/W	0000_0000h	29.5.4/1533
20B_0010	GPIO interrupt configuration register2 (GPIO6_ICR2)	32	R/W	0000_0000h	29.5.5/1537
20B_0014	GPIO interrupt mask register (GPIO6_IMR)	32	R/W	0000_0000h	29.5.6/1540
20B_0018	GPIO interrupt status register (GPIO6_ISR)	32	w1c	0000_0000h	29.5.7/1541
20B_001C	GPIO edge select register (GPIO6_EDGE_SEL)	32	R/W	0000_0000h	29.5.8/1542
20B_4000	GPIO data register (GPIO7_DR)	32	R/W	0000_0000h	29.5.1/1531
20B_4004	GPIO direction register (GPIO7_GDIR)	32	R/W	0000_0000h	29.5.2/1532
20B_4008	GPIO pad status register (GPIO7_PSR)	32	R	0000_0000h	29.5.3/1533
20B_400C	GPIO interrupt configuration register1 (GPIO7_ICR1)	32	R/W	0000_0000h	29.5.4/1533
20B_4010	GPIO interrupt configuration register2 (GPIO7_ICR2)	32	R/W	0000_0000h	29.5.5/1537
20B_4014	GPIO interrupt mask register (GPIO7_IMR)	32	R/W	0000_0000h	29.5.6/1540
20B_4018	GPIO interrupt status register (GPIO7_ISR)	32	w1c	0000_0000h	29.5.7/1541
20B_401C	GPIO edge select register (GPIO7_EDGE_SEL)	32	R/W	0000_0000h	29.5.8/1542

29.5.1 GPIO data register (GPIOx_DR)

The 32-bit GPIO_DR register stores data that is ready to be driven to the output lines. If the IOMUXC is in GPIO mode and a given GPIO direction bit is set, then the corresponding DR bit is driven to the output. If a given GPIO direction bit is cleared, then a read of GPIO_DR reflects the value of the corresponding signal.Two wait states are required in read access for synchronization.

The results of a read of a DR bit depends on the IOMUXC input mode settings and the corresponding GDIR bit as follows:

- If GDIR[*n*] is set and IOMUXC input mode is GPIO, then reading DR[*n*] returns the contents of DR[*n*].
- If GDIR[*n*] is cleared and IOMUXC input mode is GPIO, then reading DR[*n*] returns the corresponding input signal's value.
- If GDIR[*n*] is set and IOMUXC input mode is not GPIO, then reading DR[*n*] returns the contents of DR[*n*].
- If GDIR[*n*] is cleared and IOMUXC input mode is not GPIO, then reading DR[*n*] always returns zero.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_DR field descriptions

Field	Description
DR	Data bits. This register defines the value of the GPIO output when the signal is configured as an output (GDIR[n]=1). Writes to this register are stored in a register. Reading GPIO_DR returns the value stored in the register if the signal is configured as an output (GDIR[n]=1), or the input signal's value if configured as an input (GDIR[n]=0). **NOTE:** The I/O multiplexer must be configured to GPIO mode for the GPIO_DR value to connect with the signal. Reading the data register with the input path disabled always returns a zero value.

29.5.2 GPIO direction register (GPIOx_GDIR)

GPIO_GDIR functions as direction control when the IOMUXC is in GPIO mode. Each bit specifies the direction of a one-bit signal. The mapping of each DIR bit to a corresponding SoC signal is determined by the SoC's pin assignment and the IOMUX table. For more details consult the IOMUXC chapter.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GDIR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

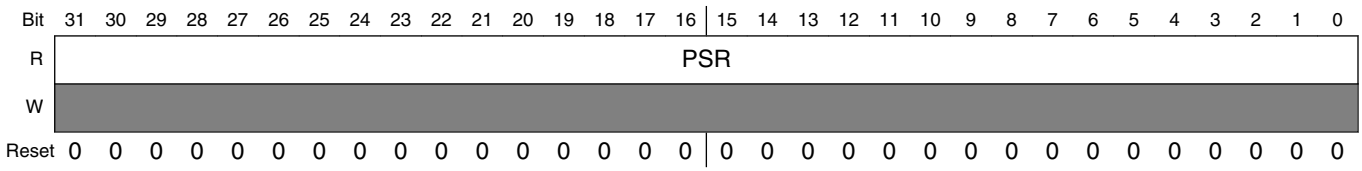
GPIOx_GDIR field descriptions

Field	Description
GDIR	GPIO direction bits. Bit n of this register defines the direction of the GPIO[n] signal. **NOTE:** GPIO_GDIR affects only the direction of the I/O signal when the corresponding bit in the I/O MUX is configured for GPIO. 0 **INPUT** — GPIO is configured as input. 1 **OUTPUT** — GPIO is configured as output.

29.5.3 GPIO pad status register (GPIOx_PSR)

GPIO_PSR is a read-only register. Each bit stores the value of the corresponding input signal (as configured in the IOMUX). This register is clocked with the ipg_clk_s clock, meaning that the input signal is sampled only when accessing this location. Two wait states are required any time this register is accessed for synchronization.

Address: Base address + 8h offset



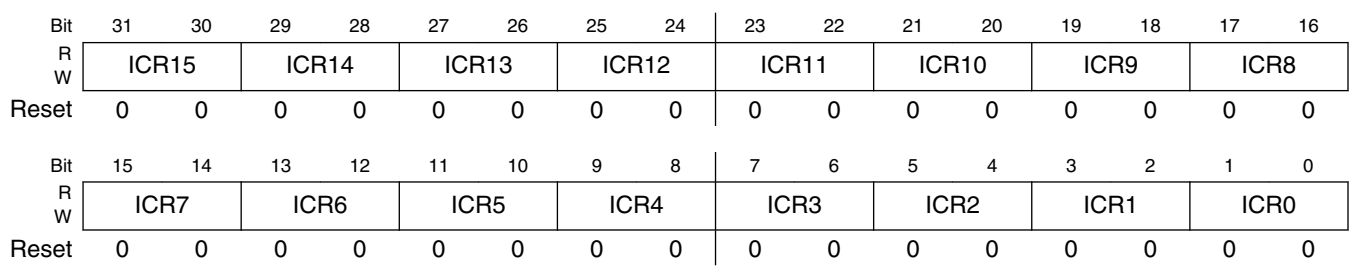
GPIOx_PSR field descriptions

Field	Description
PSR	GPIO pad status bits (status bits). Reading GPIO_PSR returns the state of the corresponding input signal. Settings: **NOTE:** The IOMUXC must be configured to GPIO mode for GPIO_PSR to reflect the state of the corresponding signal.

29.5.4 GPIO interrupt configuration register1 (GPIOx_ICR1)

GPIO_ICR1 contains 16 two-bit fields, where each field specifies the interrupt configuration for a different input signal.

Address: Base address + Ch offset



GPIOx_ICR1 field descriptions

Field	Description
31–30 ICR15	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 15.

Table continues on the next page...

GPIOx_ICR1 field descriptions (continued)

Field	Description
	Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
29–28 ICR14	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 14. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
27–26 ICR13	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 13. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
25–24 ICR12	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 12. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
23–22 ICR11	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 11. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
21–20 ICR10	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 10. Settings:

Table continues on the next page...

GPIOx_ICR1 field descriptions (continued)

Field	Description
	Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
19–18 ICR9	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 9. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
17–16 ICR8	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 8. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
15–14 ICR7	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 7. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
13–12 ICR6	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 6. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
11–10 ICR5	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 5. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows:

Table continues on the next page...

GPIOx_ICR1 field descriptions (continued)

Field	Description
	00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
9–8 ICR4	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 4. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
7–6 ICR3	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 3. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
5–4 ICR2	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 2. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
3–2 ICR1	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 1. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
ICR0	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive.

Table continues on the next page...

GPIOx_ICR1 field descriptions (continued)

Field	Description
	01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.

29.5.5 GPIO interrupt configuration register2 (GPIOx_ICR2)

GPIO_ICR2 contains 16 two-bit fields, where each field specifies the interrupt configuration for a different input signal.

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ICR31		ICR30		ICR29		ICR28		ICR27		ICR26		ICR25		ICR24	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ICR23		ICR22		ICR21		ICR20		ICR19		ICR18		ICR17		ICR16	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_ICR2 field descriptions

Field	Description
31–30 ICR31	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 31. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
29–28 ICR30	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 30. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
27–26 ICR29	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 29.

Table continues on the next page...

GPIOx_ICR2 field descriptions (continued)

Field	Description
	Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
25–24 ICR28	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 28. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
23–22 ICR27	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 27. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
21–20 ICR26	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 26. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
19–18 ICR25	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 25. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
17–16 ICR24	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 24. Settings:

Table continues on the next page...

GPIOx_ICR2 field descriptions (continued)

Field	Description
	Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
15–14 ICR23	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 23. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
13–12 ICR22	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 22. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
11–10 ICR21	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 21. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
9–8 ICR20	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 20. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
7–6 ICR19	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 19. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows:

Table continues on the next page...

GPIOx_ICR2 field descriptions (continued)

Field	Description
	00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
5–4 ICR18	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 18. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
3–2 ICR17	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 17. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
ICR16	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.

29.5.6 GPIO interrupt mask register (GPIOx_IMR)

GPIO_IMR contains masking bits for each interrupt line.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_IMR field descriptions

Field	Description
IMR	Interrupt Mask bits. This register is used to enable or disable the interrupt function on each of the 32 GPIO signals. Settings: Bit IMR[n] (n=0...31) controls interrupt n as follows: 0 **MASKED** — Interrupt n is disabled. 1 **UNMASKED** — Interrupt n is enabled.

29.5.7 GPIO interrupt status register (GPIOx_ISR)

The GPIO_ISR functions as an interrupt status indicator. Each bit indicates whether an interrupt condition has been met for the corresponding input signal. When an interrupt condition is met (as determined by the corresponding interrupt condition register field), the corresponding bit in this register is set. Two wait states are required in read access for synchronization. One wait state is required for reset.

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_ISR field descriptions

Field	Description
ISR	Interrupt status bits - Bit n of this register is asserted (active high) when the active condition (as determined by the corresponding ICR bit) is detected on the GPIO input and is waiting for service. The value of this register is independent of the value in GPIO_IMR. When the active condition has been detected, the corresponding bit remains set until cleared by software. Status flags are cleared by writing a 1 to the corresponding bit position.

29.5.8 GPIO edge select register (GPIOx_EDGE_SEL)

GPIO_EDGE_SEL may be used to override the ICR registers' configuration. If the GPIO_EDGE_SEL bit is set, then a rising edge or falling edge in the corresponding signal generates an interrupt. This register provides backward compatibility. On reset all bits are cleared (ICR is not overridden).

Address: Base address + 1Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPIO_EDGE_SEL																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_EDGE_SEL field descriptions

Field	Description
GPIO_EDGE_SEL	Edge select. When GPIO_EDGE_SEL[*n*] is set, the GPIO disregards the ICR[*n*] setting, and detects any edge on the corresponding input signal.

Chapter 30
General Purpose Media Interface (GPMI)

30.1 Overview

The GPMI controller is a flexible interface to supporting up to four NAND flash chip selects.

- ONFI 2.2, DDR Mode, Samsung / Toshiba Toggle NAND protocol compatible.
- Fully configurable address and command behavior, providing support for future devices not yet specified.

The GPMI resides on the APBH. The GPMI also provides an interface to the BCH module to allow direct parity processing.

Registers are clocked on the HCLK domain. The I/O and pin timing are clocked on a dedicated GPMICLK domain. GPMICLK can be set to maximize I/O performance.

The following figure shows a block diagram of the GPMI controller.

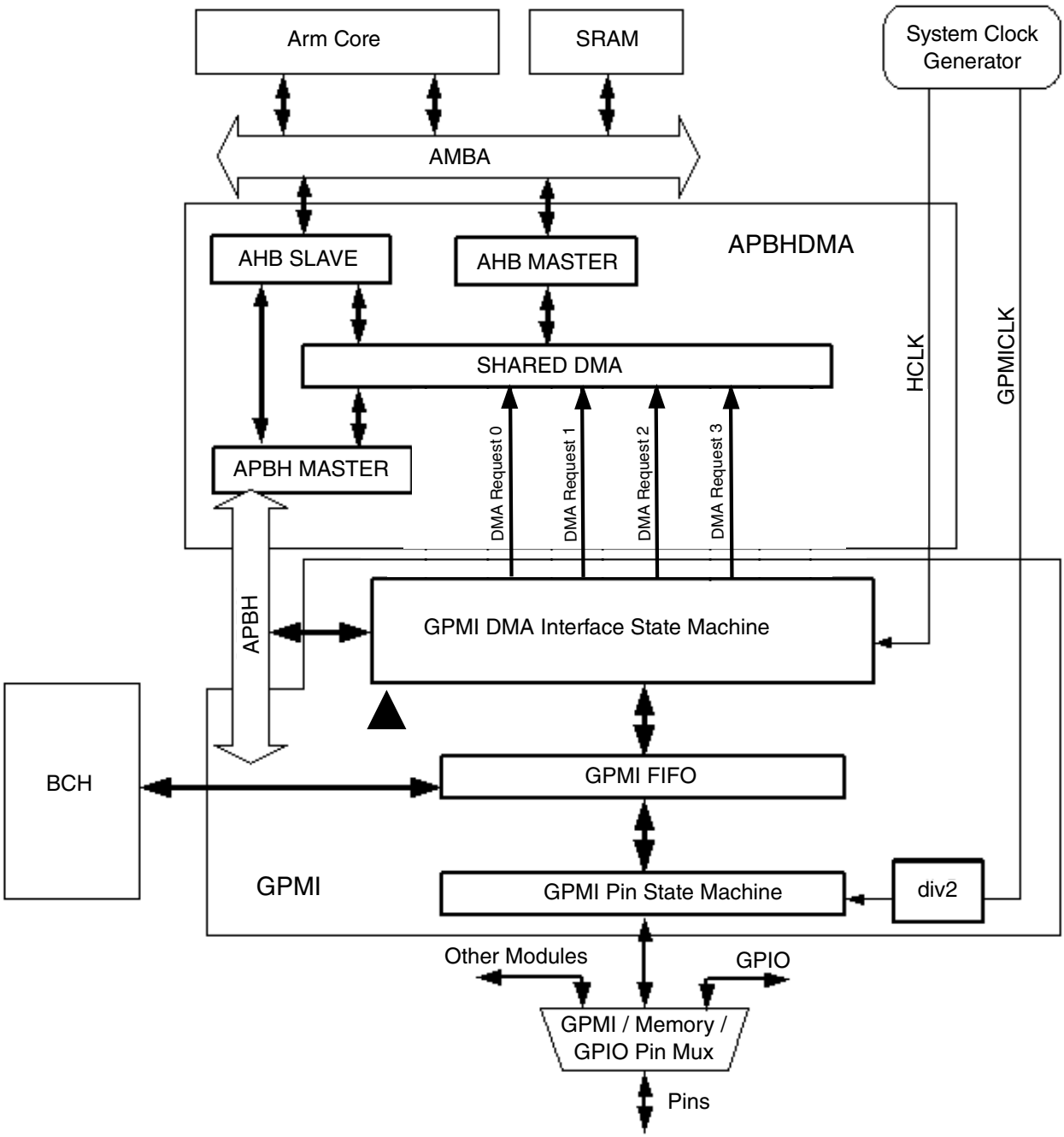


Figure 30-1. General-Purpose Media Interface Controller Block Diagram

30.2 External Signals

The table found here describes the external signals of GPMI.

Table 30-1. GPMI External Signals

Signal	Description	Pad	Mode	Direction
NAND_ALE	Address latch enable signal	NANDF_ALE	ALT0	O
NAND_CE0_B	Chip enable signal	NANDF_CS0	ALT0	O
NAND_CE1_B	Chip enable signal	NANDF_CS1	ALT0	O
NAND_CE2_B	Chip enable signal	NANDF_CS2	ALT0	O
NAND_CE3_B	Chip enable signal	NANDF_CS3	ALT0	O
NAND_CLE	Command latch enable signal	NANDF_CLE	ALT0	O
NAND_DATA00	Data signal	NANDF_D0	ALT0	I/O
NAND_DATA01	Data signal	NANDF_D1	ALT0	I/O
NAND_DATA02	Data signal	NANDF_D2	ALT0	I/O
NAND_DATA03	Data signal	NANDF_D3	ALT0	I/O
NAND_DATA04	Data signal	NANDF_D4	ALT0	I/O
NAND_DATA05	Data signal	NANDF_D5	ALT0	I/O
NAND_DATA06	Data signal	NANDF_D6	ALT0	I/O
NAND_DATA07	Data signal	NANDF_D7	ALT0	I/O
NAND_DQS	DQS signal	SD4_DAT0	ALT2	I/O
NAND_READY	Ready signal	NANDF_RB0	ALT0	I/O
NAND_RE_B	Read enable signal	SD4_CMD	ALT1	O
NAND_WE_B	Write enable signal	SD4_CLK	ALT1	O
NAND_WP_B	Wait polarity signal	NANDF_WP_B	ALT0	O

30.3 Clocks

The table found here describes the clock sources for GPMI.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 30-2. GPMI Clocks

Clock name	Clock Root	Description
bch_input_apb_clk	usdhc3_clk_root	BCH to APBH input clock
gpmi_bch_input_bch_clk	usdhc4_clk_root	BCH input clock
gpmi_bch_input_gpmi_io_clk	enfc_clk_root	GPMI IO input clock
gpmi_input_apb_clk	usdhc3_clk_root	GPMI to APBH clock

30.4 GPMI NAND Mode

The general-purpose media interface has several features to efficiently support NAND:

- Individual chip select pins and ganged ready/busy pin for up to four NANDs.
- Individual state machine and DMA channel for each chip select.
- Special command modes work with DMA controller to perform all normal NAND functions without CPU intervention.
- Configurable timing based on a dedicated clock allows optimal balance of high NAND performance and low system power.

GPMI and DMA have been designed to handle complex multi-page operations without CPU intervention. The DMA uses a linked descriptor function with branching capability to automatically handle all of the operations needed to read/write multiple pages:

- **Data/Register Read/Write**-The GPMI can be programmed to read or write multiple cycles to the NAND address, command or data registers.
- **Wait for NAND Ready**-The GPMI's Wait-for-Ready mode can monitor the ready/busy signal of a single NAND flash and signal the DMA when the device has become ready. It also has a time-out counter and can indicate to the DMA that a time-out error has occurred. The DMAs can conditionally branch to a different descriptor in the case of an error.
- **Check Status**-The Read-and-Compare mode allows the GPMI to check NAND status against a reference. If an error is found, the GPMI can instruct the DMA to branch to an alternate descriptor, which attempts to fix the problem or asserts a CPU IRQ.

30.4.1 Multiple NAND Support

The GPMI supports up to four NAND chip selects, with ganged ready/busy pins. Since they share a data bus and control lines, the GPMI can only actively communicate with a single NAND at a time. However, all NANDs can concurrently perform internal read, write, or erase operations. With fast NAND flash and software support for concurrent NAND operations, this architecture allows the total throughput to approach the data bus speed, which can be as high as 50 MB/s (8-bit bus running at 50 MHz single clock edge) in asynchronous mode and 200MB/s (8-bit bus running at 100MHz both clock edges) in Source Synchronous mode.

There are two options for controlling the four NAND chip selects via the DMA interface. The first option is the one to one mapping, where the each DMA channel is tied to its own NAND chip select. For example DMA channel 'n' accesses only NAND attached to chip select 'n'. The second option is the decoupled mode where a DMA channel can access any or all NAND chip selects connected to the GPMI. A DMA channel will signify the NAND chip select it wants to access by writing its chip select value in the GPMI_CTRL0[CS] field and setting the GPMI_CTRL1[DECOUPLE_CS] to 1. This option is useful if software chooses to use only one DMA channel to access all the attached NAND devices.

30.4.2 GPMI NAND Timing and Clocking

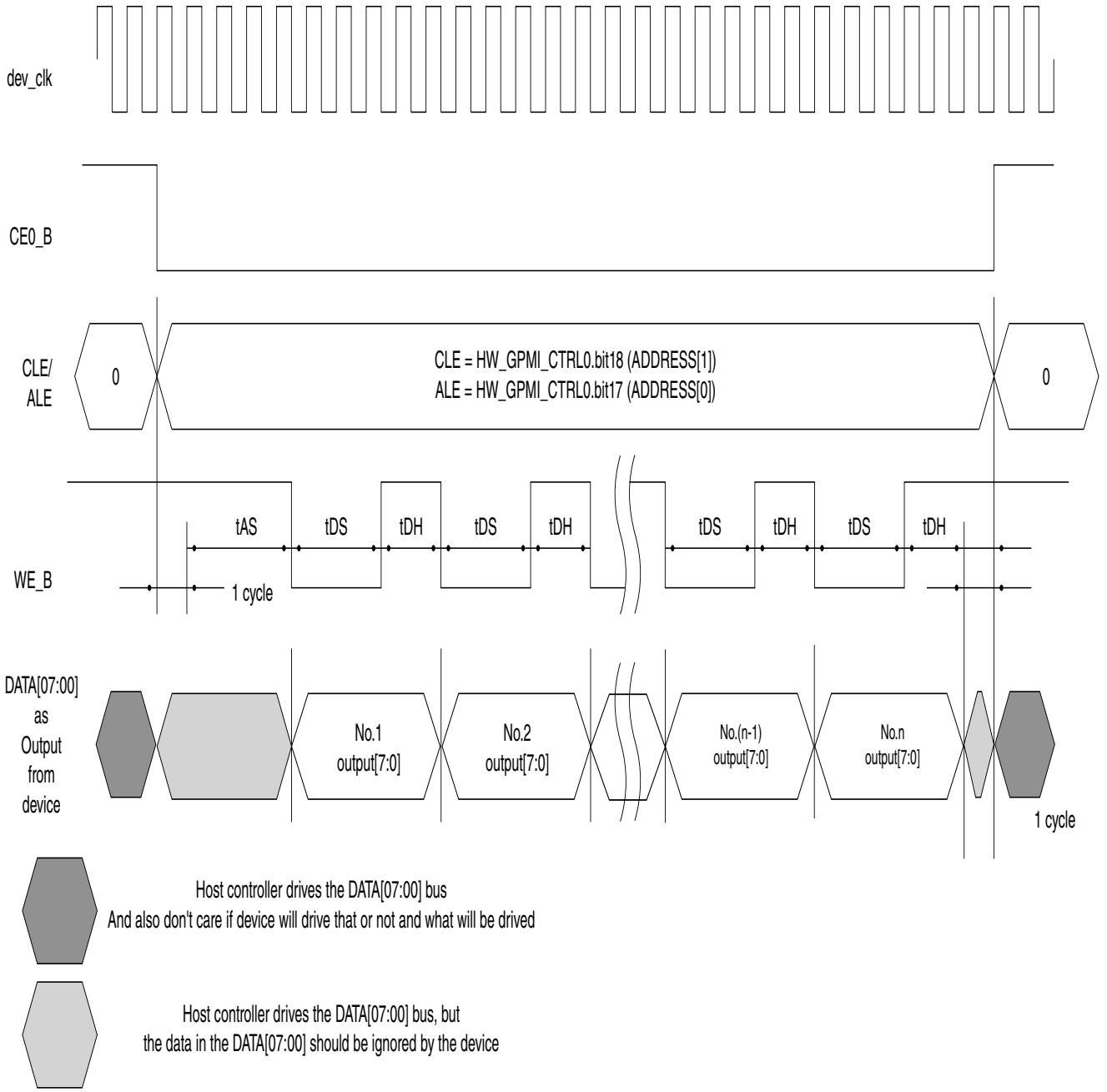
The dedicated clock, GPMICLK, is used as a timing reference for NAND flash I/O. Since various NANDs have different timing requirements, GPMICLK may need to be adjusted for each application.

While the actual pin timings are limited by the NAND chips used and the I/O pad configuration, the GPMI can support data bus speeds of up to 200 MHz x 8 bits. The actual read/write strobe timing parameters are adjusted as indicated in the register descriptions in Memory Map.

30.4.3 Basic NAND Timing

30.4.3.1 NAND Asynchronous Timing

Figure 30-3 and illustrates the operation of the output (from host to device) timing parameters in NAND ONFI asynchronous mode.



- tAS is configurable by programming HW_GPMI_TIMING0 Address_Setup: in this example, Address_Setup = 4, tAS is equal to 4 dev_clk cycles.
- tDS is configurable by programming HW_GPMI_TIMING0 Data_Setup; in this example, Data_Setup = 3, tDS is equal to 3 dev_clk cycles
- tDH is configuarble by programming HW_GPMI_TIMING0 Data_Hold: in this example, Data_Hold = 2, tDH is equal to 2 dev_clk cycles
- tAS/tDS/tDH will extend, if the output data is not ready in device fifo.

Figure 30-2. Asynchronous Mode Basic Write Timing Diagram (command write, address write, or data write

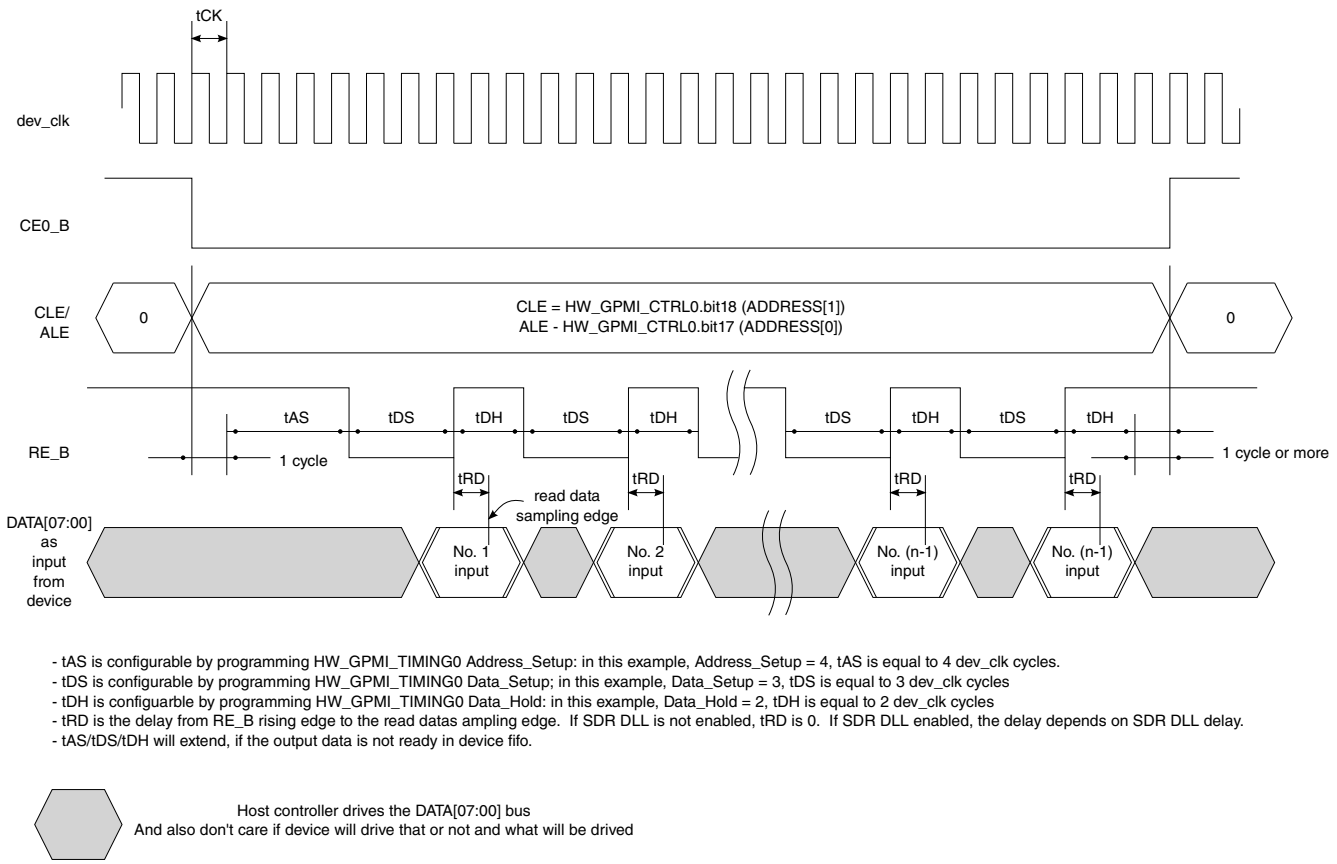


Figure 30-3. ONFI Asynchronous Mode Basic Read Timing Diagram (data read)

30.4.3.2 NAND Asynchronous EDO Mode Timing

In high-speed NANDS, the read data may not be valid until after the read strobe (NAND_RE_B) deasserts. This is the case when the minimum tDS is programmed to achieve higher bandwidth.

The GPMI implements a feedback read strobe to sample the read data. The feedback read strobe can be delayed to support fast nand EDO (Extended Data Out) timing where the read strobe may deassert before the read data is valid, and read data is valid for some time after read strobe. Nand EDO timings is applied typically for read cycle frequency above 33 MHz. See Figure 30-4.

The GPMI provides control over the amount of delay applied to the feedback read strobe. This delay depends on the maximum read access time (tREA) of the nand and the read pulse width (tRP) used to access the nand. tRP is specified by GPMI_TIMING0[DATA_SETUP] register. When (tREA + 4ns) is less than tRP, no

delay is required to sample to nand read data. (The 4ns provides adequate data setup time for the GPMI.) In this case set GPMI_CTRL1[HALF_PERIOD] = 0; GPMI_CTRL1[RDN_DELAY] = 0; GPMI_CTRL1[DLL_ENABLE] = 0.

When (tREA + 4ns) is greater than or equal to tRP, a delay of the feedback read strobe is required to sample to nand read data. This delay is equal to the difference between these two timings:

DELAY = tREA + 4ns - tRP.

Since the GPMI delay chain is limited to 12ns maximum, if DELAY > 12ns then increase tRP by increasing the value of GPMI_TIMING0[DATA_SETUP] until DELAY is less than or equal to 12ns.

The GPMI programming for this DELAY depends on the GPMICLK period. The GPMI DLL will not function properly if the GPMICLK period is greater than 24ns: disable the DLL if this is the case. If the GPMICLK period is greater than 12ns (and not greater than 24ns), set the GPMI_CTRL1[HALF_PERIOD]=1; This will cause the DLL reference period (RP) to be one-half of the GPMICLK period. If the GPMICLK period is 12ns or less then set the GPMI_CTRL1[HALF_PERIOD]=0; This will cause the DLL reference period (RP) to be equal to the GPMICLK period. DELAY is a multiple (0 to 1.875) of RP.

The GPMI_CTRL1[RDN_DELAY] is encoded as a 1-bit integer and 3-bit fraction delay factor. DELAY is a multiple of the delay factor and the reference period. See table below for details.

Table 30-3. RDN DELAY

HW_GPMI_CTRL1[RDN_DELAY]	Delay Factor
0	0.000
1	0.125
2	0.250
3	0.375
4	0.500
5	0.625
6	0.750
7	0.875
8	1.000
9	1.125
10	1.250
11	1.375
12	1.500
13	1.625

Table continues on the next page...

Table 30-3. RDN DELAY (continued)

HW_GPMI_CTRL1[RDN_DELAY]	Delay Factor
14	1.750
15	1.875

DELAY = DelayFactor x RP or DELAY = GPMI_CTRL1[RDN_DELAY] x 0.125 x RP.

Use this equation to calculate the value for GPMI_CTRL1[RDN_DELAY]. Then set GPMI_CTRL1[DLL_ENABLE]=1.

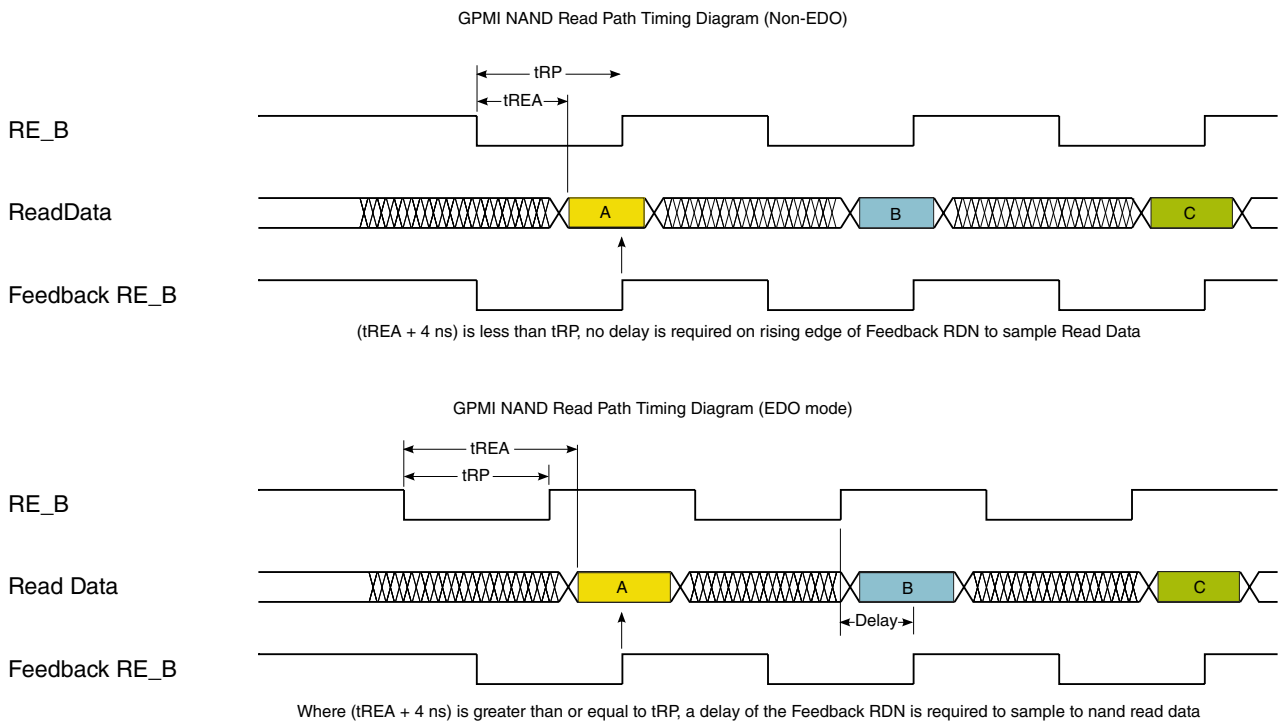


Figure 30-4. NAND Read Path Timing

For example, a NAND with tREAmax = 20 ns, tRPmin = 12 ns, and tRCmin = 25 ns (read cycle time) may be programmed as follows:

- GPMICLK clock frequency: Consider 480/6 = 80 MHz which is 12.5 ns clock period. This is too close to the minimum NAND spec if we program the data setup and hold to 1 GPMICLK cycle. Consider 480/7=68.57 MHz which is 14.58 ns clock period. With data setup and hold set to 1, we have a tRP of 14.58ns and a tRC of 29.16 ns (good margins).
- Since (tREA +4ns) is greater than tRP, required DELAY = tREA + 4ns - tRP = 20 + 4 - 14.58ns = 9.42 ns.

- GPMI_CTRL1[HALF_PERIOD] =1, since GPMICLK period is large than 12ns. So RP=GPMICLK period = 7.29ns.
- DELAY = GPMI_CTRL1[RDN_DELAY] x 0.125 x RP. 9.42 ns = GPMI_CTRL1[RDN_DELAY] x 0.125 x 7.29ns. GPMI_CTRL1[RDN_DELAY] = 10 (round of 10.33)

For ONFI spec EDO timing mode 4/5, It's recommended to have below settings.

EDO timing mode 5 :
GPMICLK=100MHz;GPMI_CTRL1[RDN_DELAY]=8;GPMI_CTRL1[HALF_PERIOD] =0;

EDO timing mode 4 :
GPMICLK=80MHz;GPMI_CTRL1[RDN_DELAY]=15;GPMI_CTRL1[HALF_PERIOD] =1;

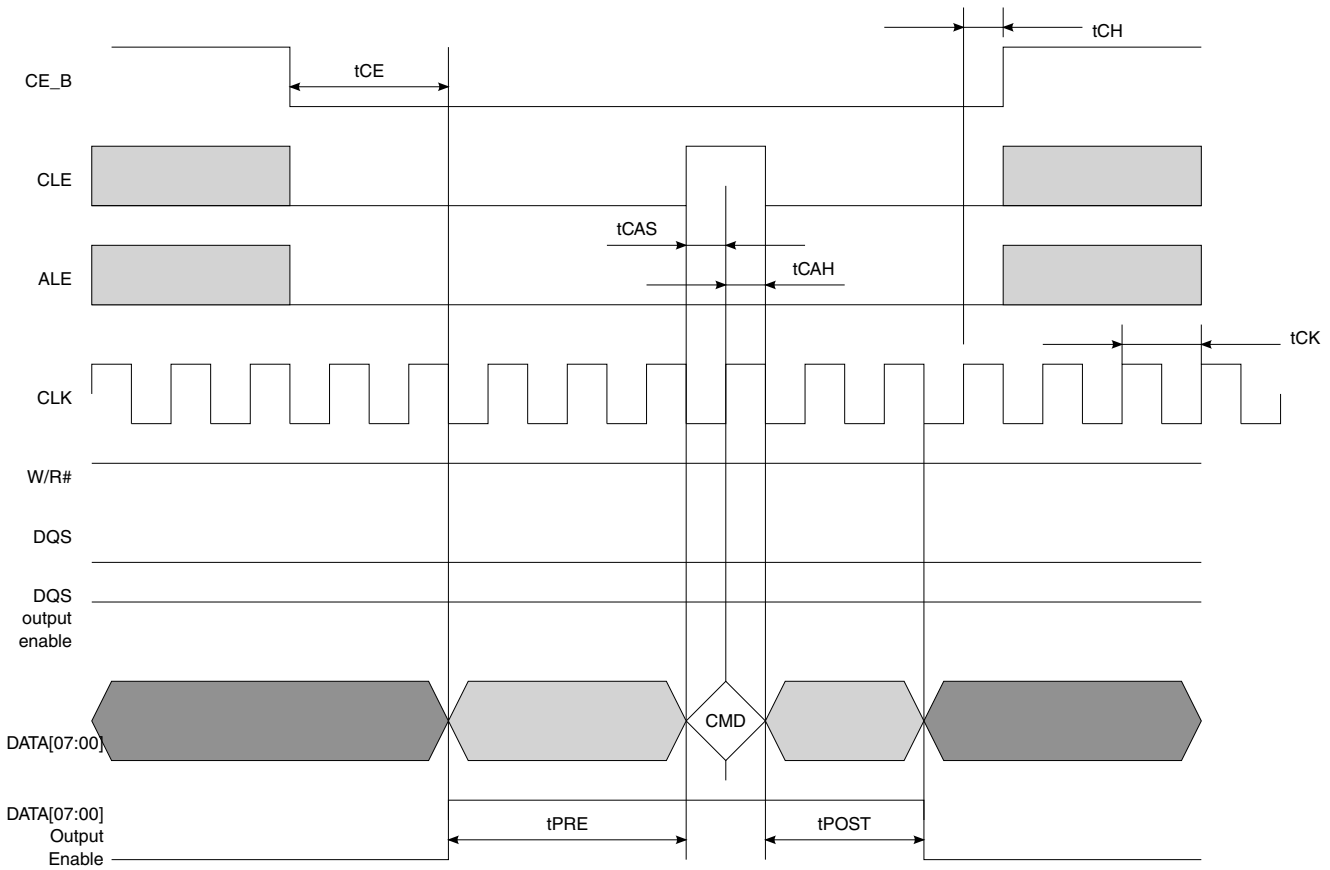
NOTE

It is recommended that the drive strength of NAND_RE_B and NAND_WE_B output pins be set to 8 mA. This will reduce the transition time under heavy loads. Low transition times will be important when NAND interface read and write cycle times are below 30 ns. The other GPMI pins may remain at 4 mA, since their frequency is only up to half that of NAND_RE_B and NAND_WE_B.

30.4.3.3 NAND ONFI Source Synchronous Mode Timing

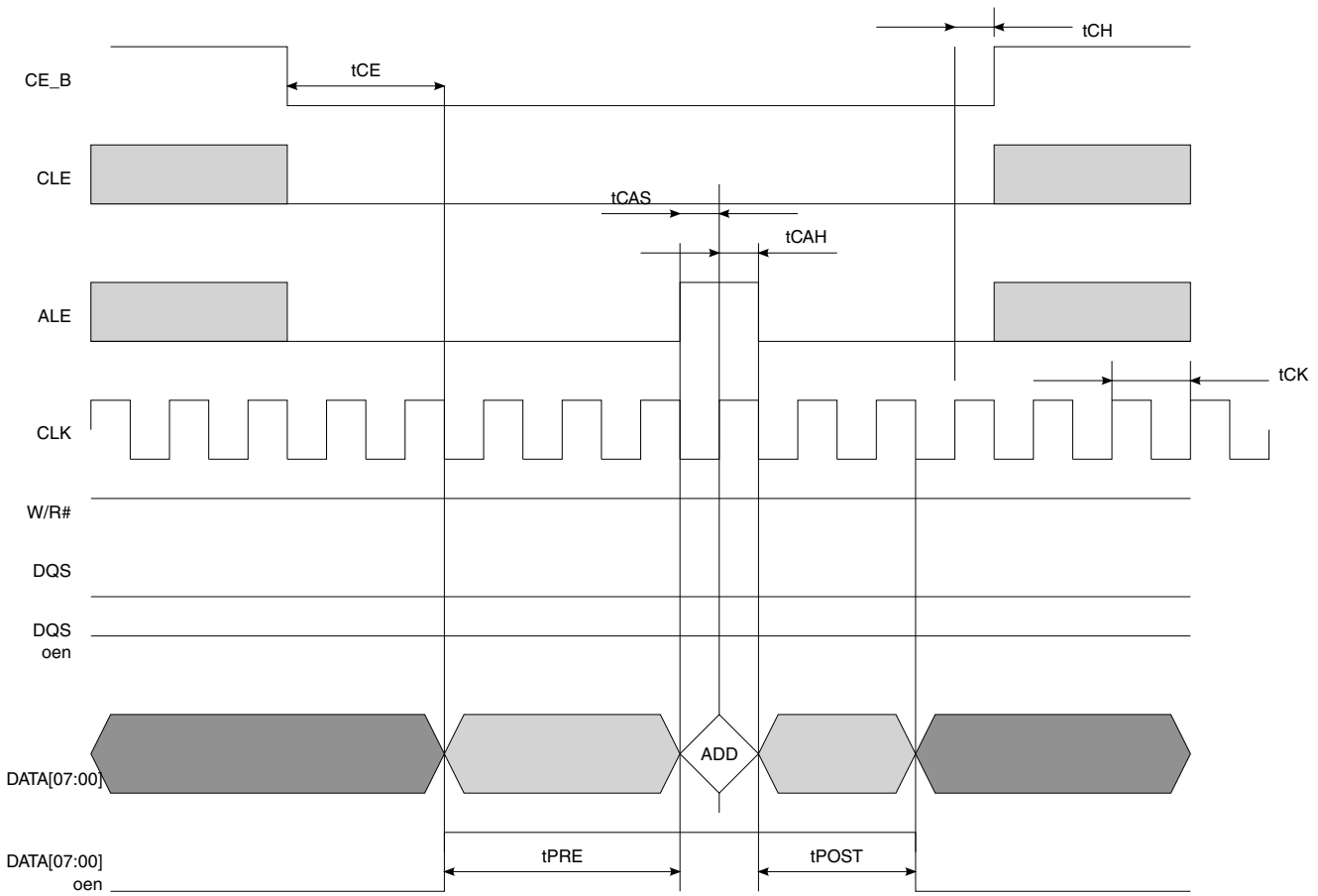
NOTE

In the following figures, CLK shares the same pin as WE_B in Async Mode. And W/R# shares the same pin as RE_B in Async Mode.



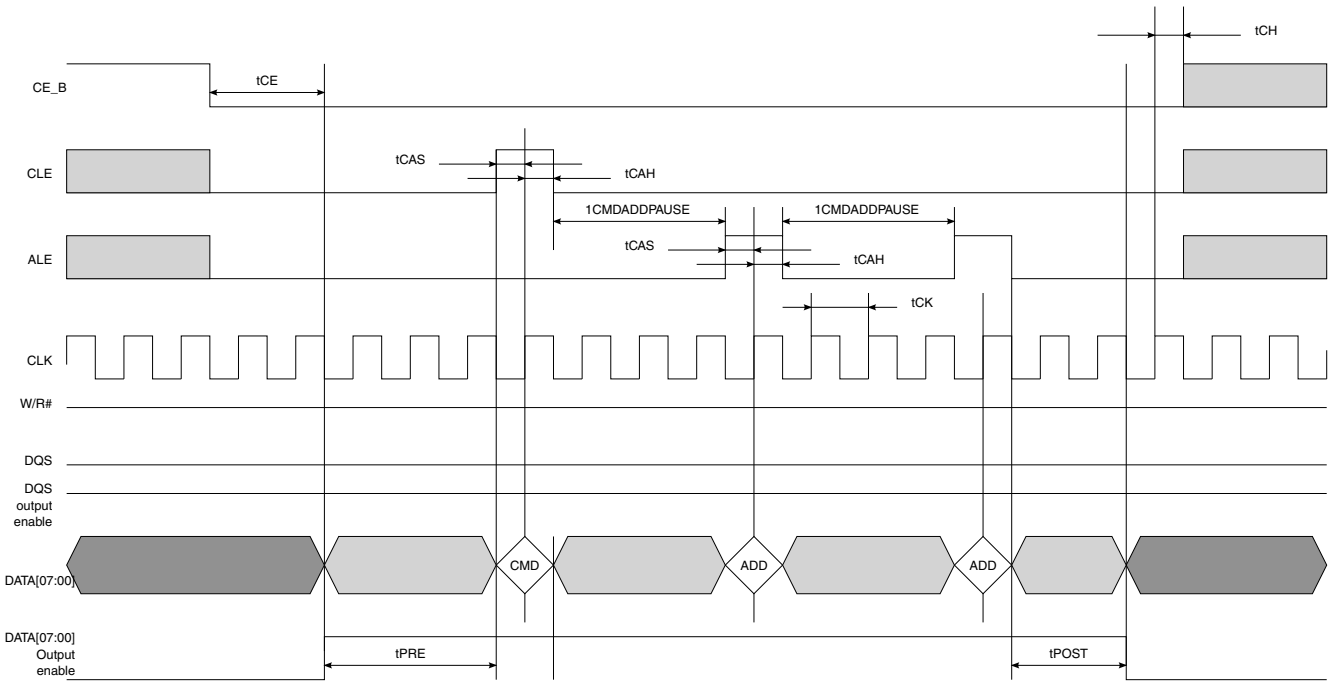
tCK is equal to device clock period
tCAS = 0.5 * tCK
tCAH = 0.5 * tCK
tCE = HW_GPMI_TIMING2.CE_DELAY * tCK
tCH = 0.5 * tCK
tPRE = HW_GPMI_TIMING2.PREAMBLE_DELAY * tCK
tPOST = HW_GPMI_TIMING2.POST_DELAY * tCK

Figure 30-5. ONFI Source Synchronous Mode Basic Command Write Timing Diagram



tCK is equal to device clock period
tCAS = 0.5 * tCK
tCAH = 0.5 * tCK
tCE = HW_GPMI_TIMING2.CE_DELAY * tCK
tCH = 0.5 * tCK
tPRE = HW_GPMI_TIMING2.PREAMBLE_DELAY * tCK
tPOST = HW_GPMI_TIMING2.POST_DELAY * tCK

Figure 30-6. ONFI Source Synchronous Mode Basic Address Write Timing Diagram

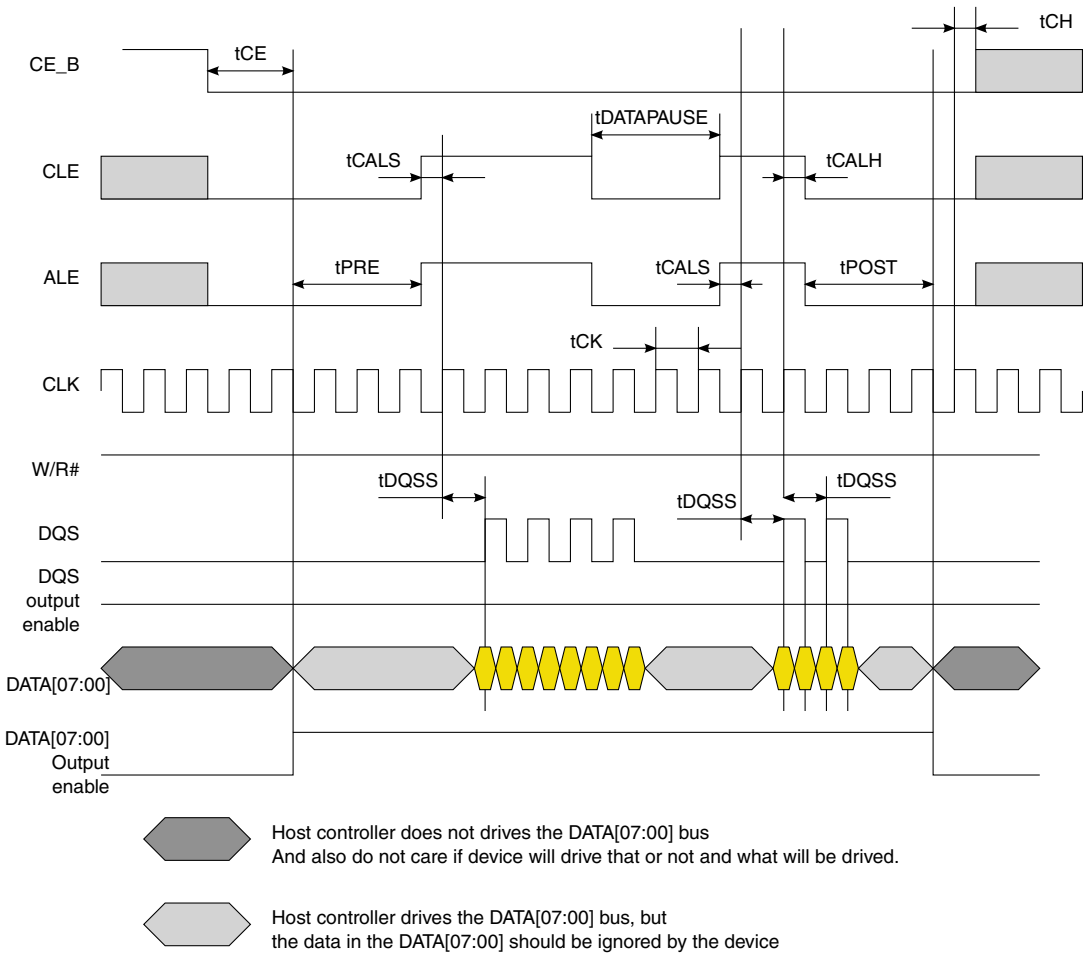


Host controller does not drives the DATA[07:00] bus
And also do not care if device will drive that or not and what will be drived.

Host controller drives the DATA[07:00] bus, but
the data in the DATA[07:00] should be ignored by the device

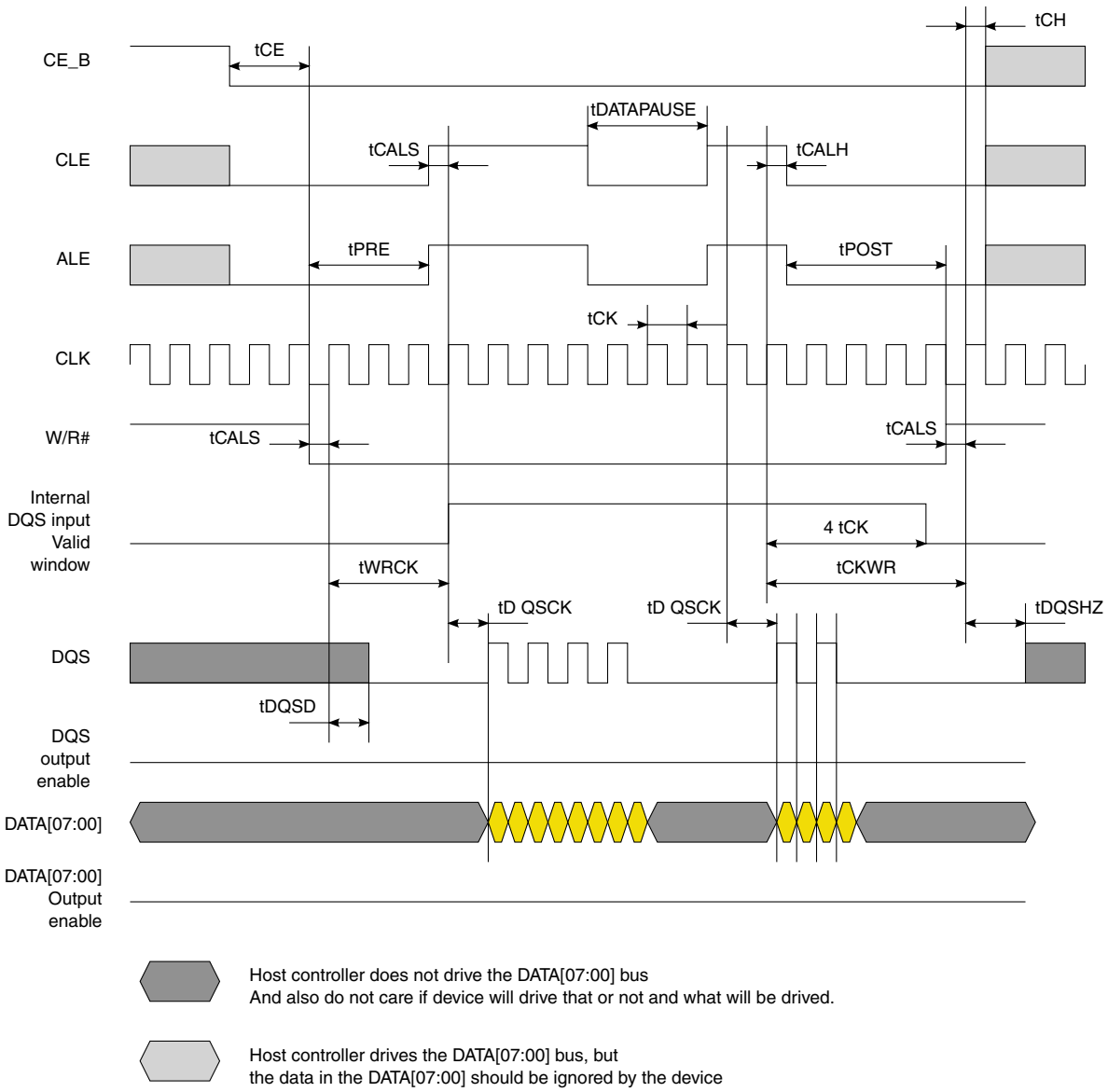
tCK is equal to device clock period
tCAS = 0.5 * tCK
tCAH = 0.5 *tCK
tCE = HW_GPMI_TIMING2.CE_DELAY * tCK
tCH = 0.5 * tCK
tPRE = HW_GPMI_TIMING2.PREAMBLE_DELAY * tCK
tPOST = HW_GPMI_TIMING2.POST_DELAY * tCK
tCMDADDPAUSE = HW_GPMI_TIMING2.CMDADD_PAUSE * tCK

Figure 30-7. ONFI Source Synchronous Mode Command + Address Write Timing Diagram



tCK is equal to device clock period
tCE = HW_GPI_TIMING2.CE_DELAY * tCK
tCH = 0.5 * tCK
tCALS = 0.5 * tCK
tCALH = 0.5 * tCK
tPRE = HW_GPMI_TIMING2.PREAMBLE_DELAY *tCK
tPOST = HW_GPMI_TIMING2.POST_DELAY * tCK
tDATAPAUSE = HW_GPMI_TIMING2.DATA_PAUSE * tCK
tDQSS = tCK

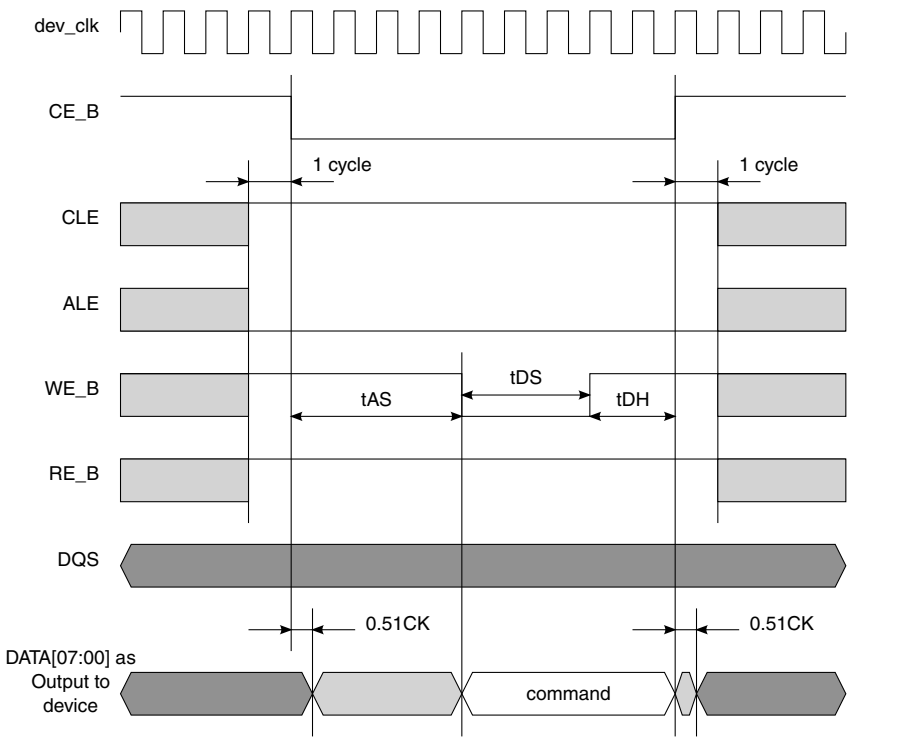
Figure 30-8. ONFI Source Synchronous Mode Data Write Timing Diagram



tCK is equal to device clock period
tCE = HW_GPI_TIMING2.CE_DELAY * tCK
tCH = 0.5 * tCK
tCALS = 0.5 * tCK
tCALH = 0.5 * tCK
tPRE = HW_GPMI_TIMING2.PREAMBLE_DELAY *tCK
tPOST = HW_GPMI_TIMING2.POST_DELAY * tCK
tDATAPAUSE = HW_GPMI_TIMING2.DATA_PAUSE * tCK
tWRCK/tCKWR/tDQSD/tDASCK/tDASHZ are device parameters

Figure 30-9. ONFI Source Synchronous Mode Data Read Timing Diagram

30.4.3.4 NAND Toggle Mode Timing

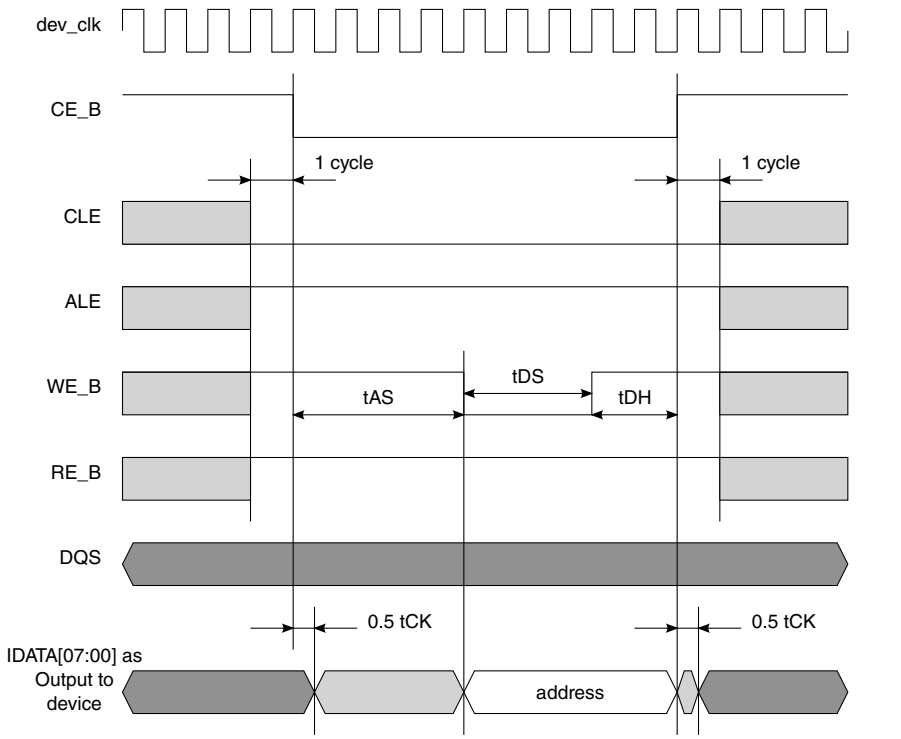


Host controller does not drive the DATA[07:00] bus
And also does not care if device will drive that or not and what will be drived.

Host controller drives the DATA[07:00] bus, but
the data in the DATA[07:00] should be ignored by the device

- tAS is configurable by programming HW_GPMI_TIMING0 Address_Setup;
- tDS is configurable by programming HW_GPMI_TIMING0 Data_Setup;
- tDH is configurable br programming HW_GPMI_TIMING Data_Hold;
- tAS/tDS/tDH will extend, if the output data is not ready in device fifo.

Figure 30-10. Samsung Toggle Mode Basic Command Write Timing Diagram



Host controller does not drive the DATA[07:00] bus
And also does not care if device will drive that or not and what will be drived.

Host controller drives the DATA[07:00] bus, but
the data in the DATA[07:00] should be ignored by the device

- tAS is configurable by programming HW_GPMI_TIMING0 Address_Setup;
- tDS is configurable by programming HW_GPMI_TIMING0 Data_Setup;
- tDH is configurable br programming HW_GPMI_TIMING Data_Hold;
- tAS/tDS/tDH will extend, if the output data is not ready in device fifo.

Figure 30-11. Samsung Toggle Mode Basic Address Write Timing Diagram

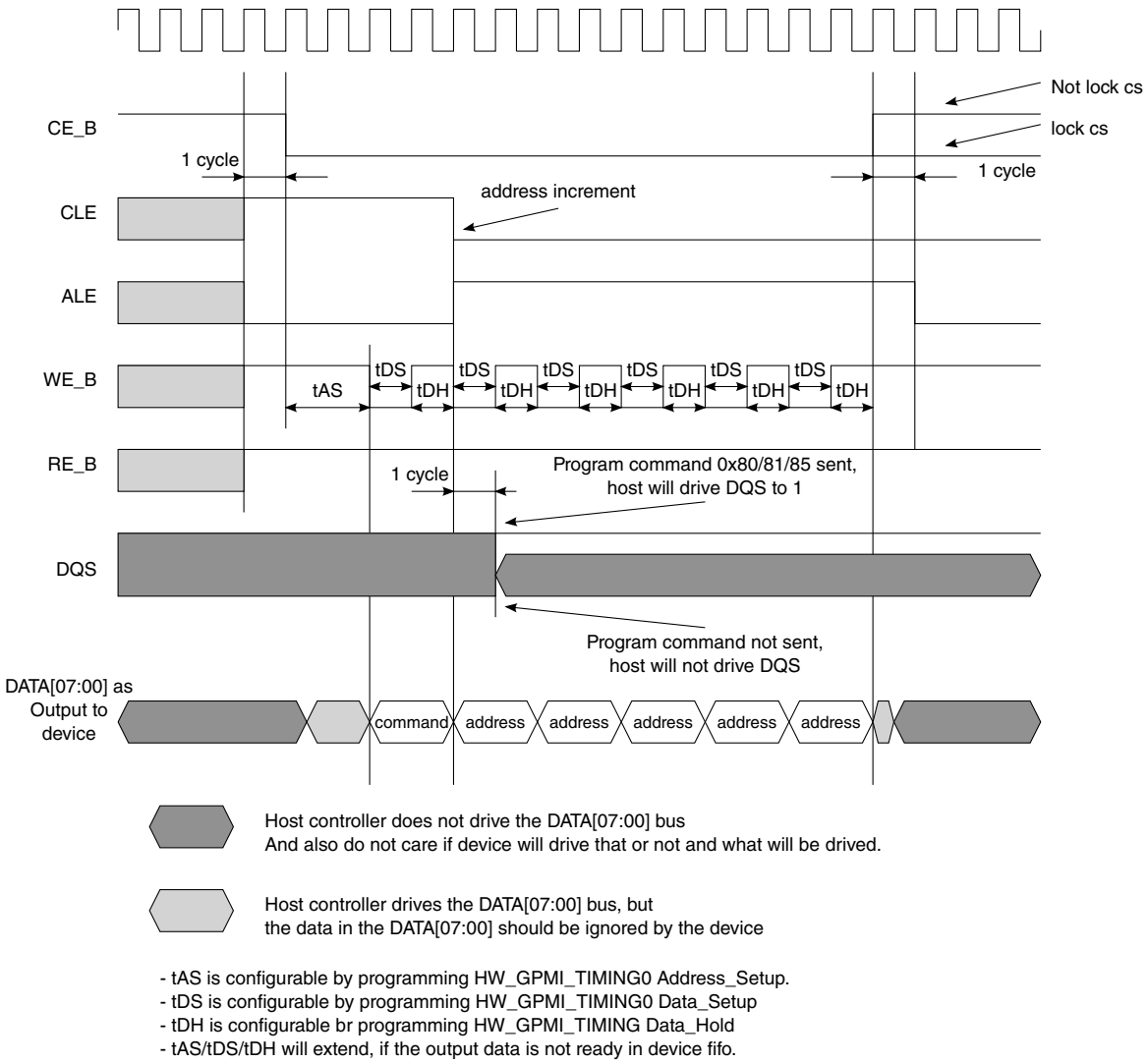


Figure 30-12. Samsung Toggle Mode Basic Command + Address Timing Diagram

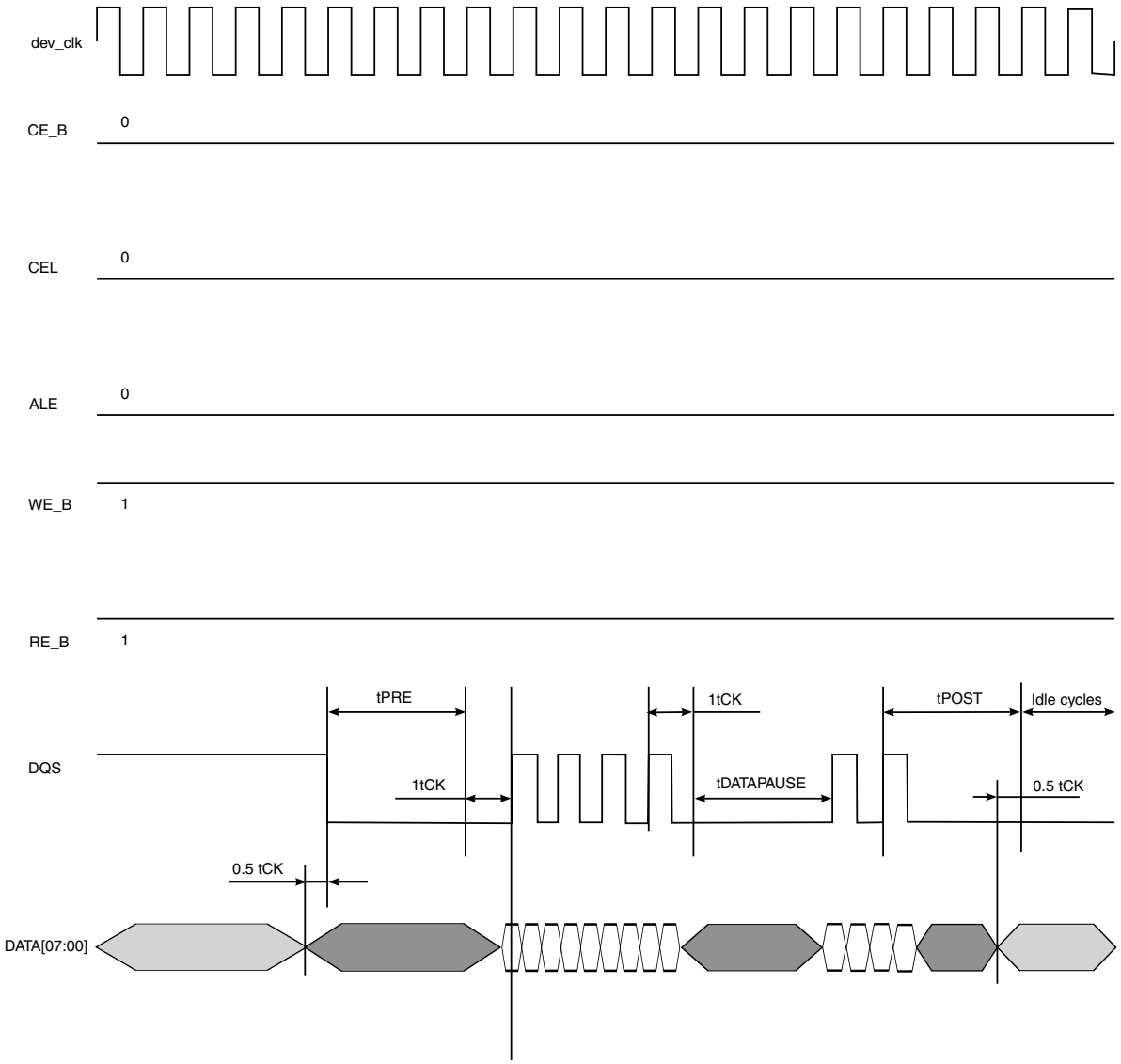
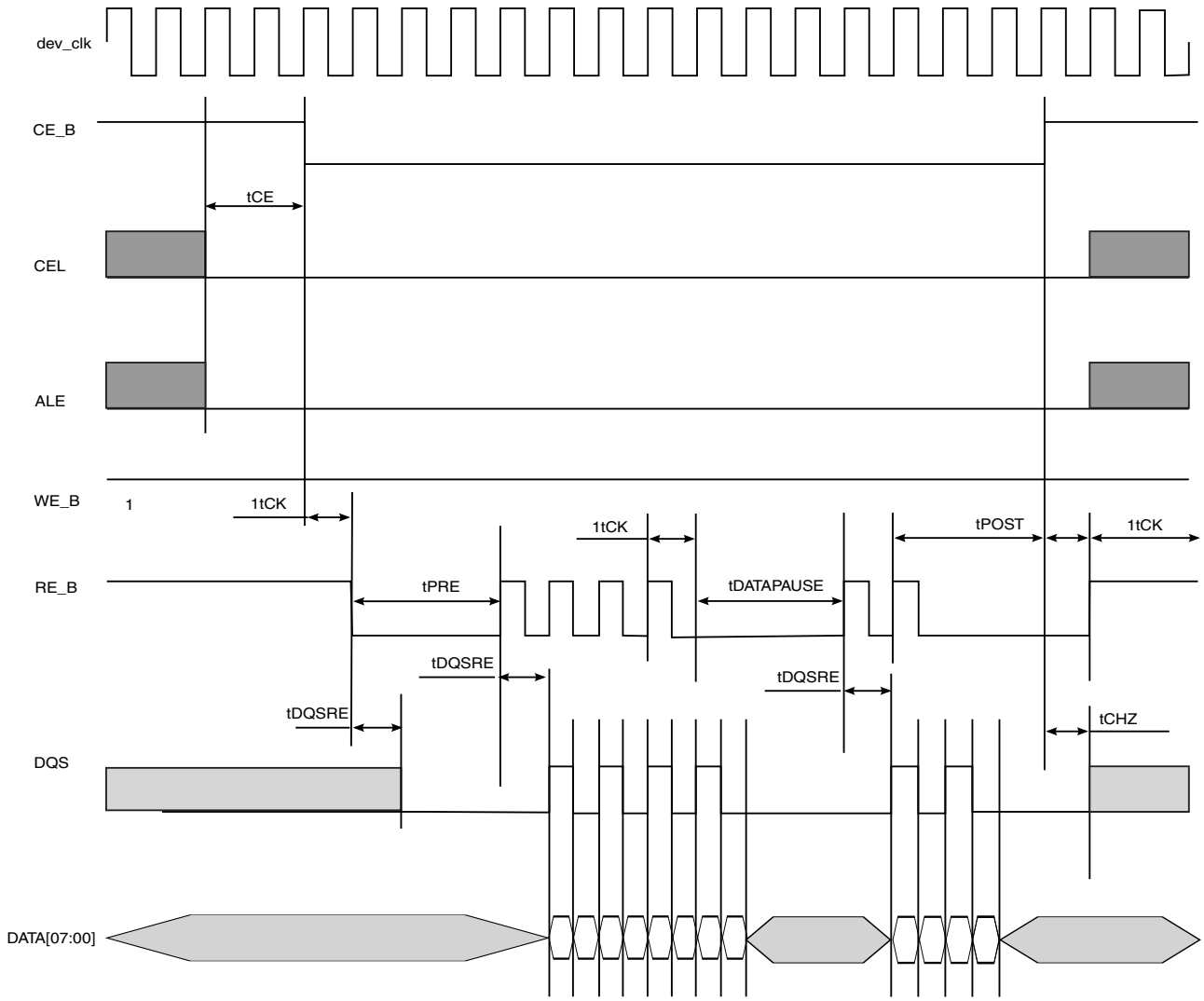


Figure 30-13. Toggle Mode Data Write Timing Diagram



Host controller does not drive the DATA[07:00] bus.
It also does not care if the device will drive or what will be driven.

Host controller drives the DATA[07:00] bus, but
the data in the DATA[07:00] should be ignored by device.

tCK is equal to device clock period

tCE = HW_GPMI_TIMIN G2.CE_DELAY* tCK

tPRE = HW_GPMI_TIMIN G2.PREAMBLE_DELAY* tCK

tPOST = HW_GPMI_TIMIN G2.POSTAMBLE_DELAY* tCK

tDATAPAUSE = HW_GPMI_TIMIN G2.DATA_PAUSE* tCK

Figure 30-14. Toggle Mode Data Read Timing Diagram

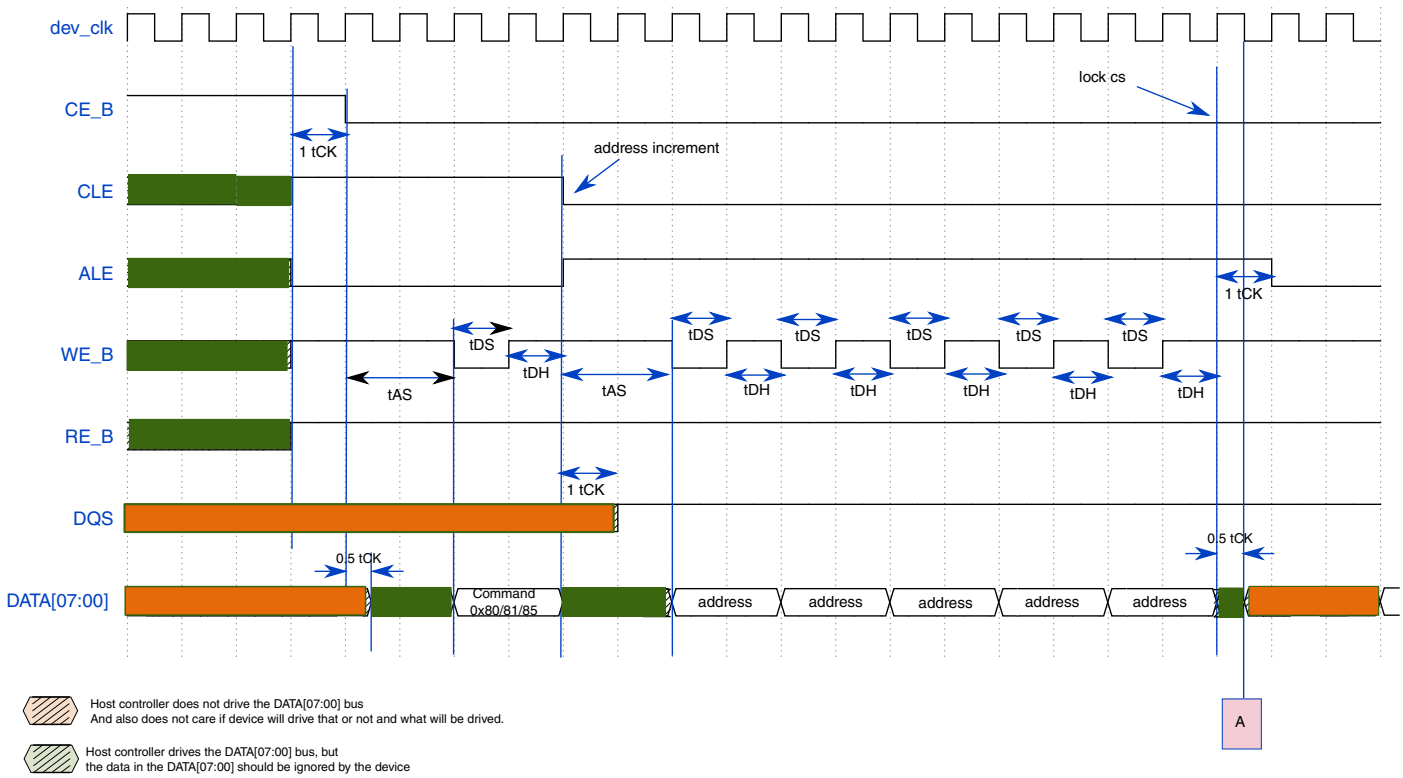


Figure 30-15. Toggle Mode Program Timing Diagram (A)

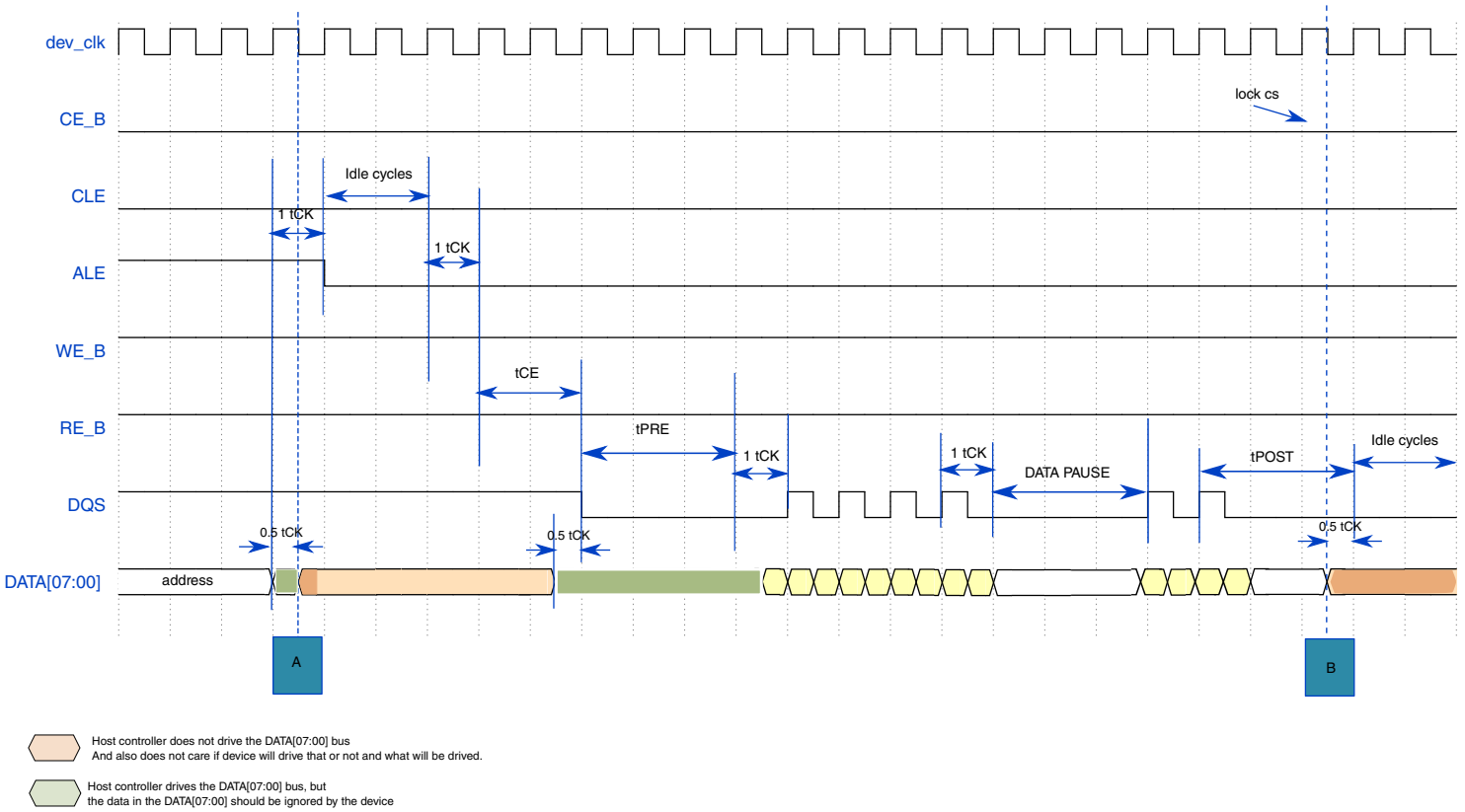


Figure 30-16. Toggle Mode Program Timing Diagram (B)

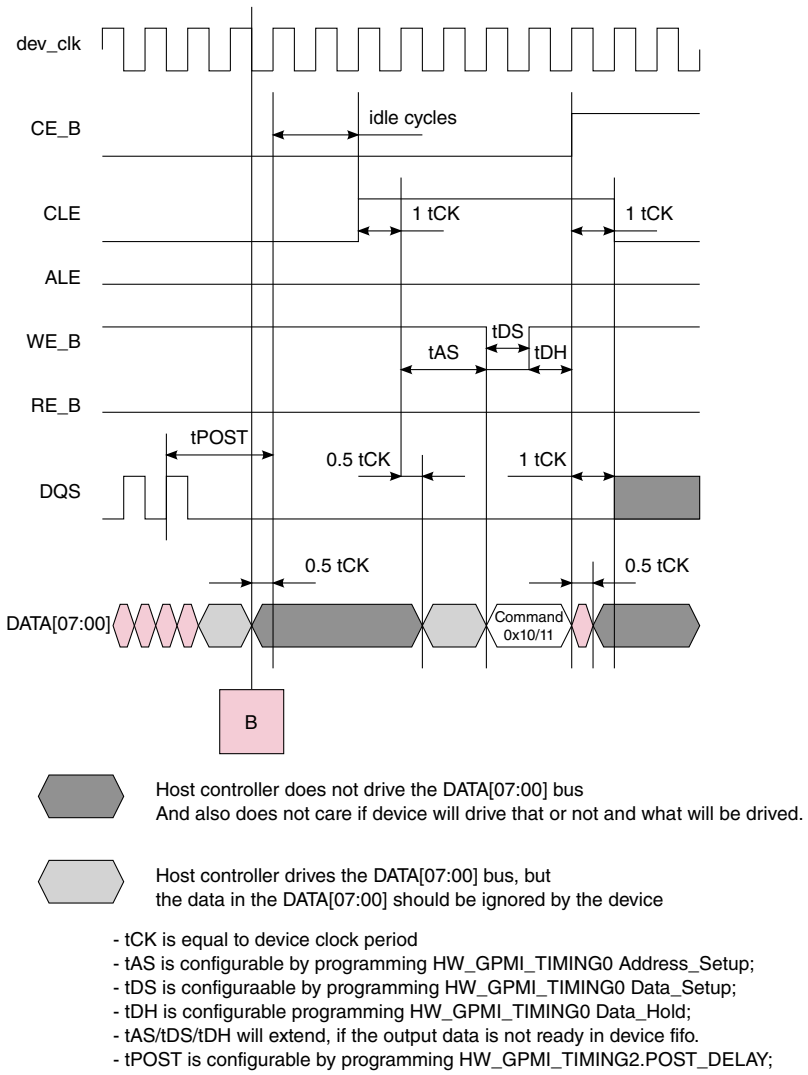


Figure 30-17. Toggle Mode Program Timing Diagram (C)

30.4.4 Hardware BCH Interface

The GPMI provides an interface to the BCH module. This reduces the SOC bus traffic and the software involvement.

.

When BCH ECC is enable, parity information is inserted on-the-fly during writes to 8-bit NAND devices. The BCH will supply payload and parity to the GPMI to write to the NAND. During NAND reads, parity is checked and ECC processing is performed after each read block. In this case the GPMI reads the NAND device and redirects the data and parity to the BCH module for ECC processing.

To program the BCH for NAND writes, remove the soft reset and clock gates from BCH_CTRL[SFTRST] and BCH_CTRL[CLKGATE]. The bulk of BCH programming is actually applied to the GPMI via PIO operations embedded in its DMA command structures. This has a subtle implication when writing to the GPMI ECC registers: access to the these registers must be written in progressive register order. Thus, to write to the GPMI_ECCCOUNT register, write first (in order) to registers GPMI_CTRL0, GPMI_COMPARE, and GPMI_ECCCTRL before writing to GPMI_ECCCOUNT. These additional register writes need to be accounted for in the CMDWORDS field of the respective DMA channel command register.

NOTE

Note that the GPMI_PAYLOAD and GPMI_AUXILIARY pointers need to be word-aligned for proper ECC operation. If those pointers are non-word-aligned, then the BCH engine will not operate properly and could possibly corrupt system memory in the adjoining memory regions.

30.5 Behavior During Reset

A soft reset (SFTRST) can take multiple clock periods to complete, so do NOT set CLKGATE when setting SFTRST. The reset process gates the clocks automatically.

30.6 GPMI Memory Map/Register Definition

The following registers provide control for programmable elements of the GPMI module.

NOTE

All ATA or UDMA features are not supported for the chip.

NOTE

GPMI does not support the Set feature command in Toggle mode. The NANDF_DQS output is only enabled in program operation for Toggle mode, but the Set feature command also needs to use the NANDF_DQS signal to write data to Toggle NAND flash. So the Set feature command in Toggle mode is not supported.

GPMI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
11_2000	GPMI Control Register 0 Description (GPMI_CTRL0)	32	R/W	C000_0000h	30.6.1/1568
11_2004	GPMI Control Register 0 Description (GPMI_CTRL0_SET)	32	R/W	C000_0000h	30.6.1/1568
11_2008	GPMI Control Register 0 Description (GPMI_CTRL0_CLR)	32	R/W	C000_0000h	30.6.1/1568
11_200C	GPMI Control Register 0 Description (GPMI_CTRL0_TOG)	32	R/W	C000_0000h	30.6.1/1568
11_2010	GPMI Compare Register Description (GPMI_COMPARE)	32	R/W	0000_0000h	30.6.2/1570
11_2020	GPMI Integrated ECC Control Register Description (GPMI_ECCCTRL)	32	R/W	0000_0000h	30.6.3/1571
11_2024	GPMI Integrated ECC Control Register Description (GPMI_ECCCTRL_SET)	32	R/W	0000_0000h	30.6.3/1571
11_2028	GPMI Integrated ECC Control Register Description (GPMI_ECCCTRL_CLR)	32	R/W	0000_0000h	30.6.3/1571
11_202C	GPMI Integrated ECC Control Register Description (GPMI_ECCCTRL_TOG)	32	R/W	0000_0000h	30.6.3/1571
11_2030	GPMI Integrated ECC Transfer Count Register Description (GPMI_ECCCOUNT)	32	R/W	0000_0000h	30.6.4/1572
11_2040	GPMI Payload Address Register Description (GPMI_PAYLOAD)	32	R/W	0000_0000h	30.6.5/1572
11_2050	GPMI Auxiliary Address Register Description (GPMI_AUXILIARY)	32	R/W	0000_0000h	30.6.6/1573
11_2060	GPMI Control Register 1 Description (GPMI_CTRL1)	32	R/W	0004_0004h	30.6.7/1574
11_2064	GPMI Control Register 1 Description (GPMI_CTRL1_SET)	32	R/W	0004_0004h	30.6.7/1574
11_2068	GPMI Control Register 1 Description (GPMI_CTRL1_CLR)	32	R/W	0004_0004h	30.6.7/1574
11_206C	GPMI Control Register 1 Description (GPMI_CTRL1_TOG)	32	R/W	0004_0004h	30.6.7/1574
11_2070	GPMI Timing Register 0 Description (GPMI_TIMING0)	32	R/W	0001_0203h	30.6.8/1577
11_2080	GPMI Timing Register 1 Description (GPMI_TIMING1)	32	R/W	0000_0000h	30.6.9/1577
11_2090	GPMI Timing Register 2 Description (GPMI_TIMING2)	32	R/W	0302_3336h	30.6.10/ 1578
11_20A0	GPMI DMA Data Transfer Register Description (GPMI_DATA)	32	R/W	0000_0000h	30.6.11/ 1579
11_20B0	GPMI Status Register Description (GPMI_STAT)	32	R	0000_0005h	30.6.12/ 1579
11_20C0	GPMI Debug Information Register Description (GPMI_DEBUG)	32	R	0000_0000h	30.6.13/ 1582
11_20D0	GPMI Version Register Description (GPMI_VERSION)	32	R	0501_0000h	30.6.14/ 1583
11_20E0	GPMI Debug2 Information Register Description (GPMI_DEBUG2)	32	R/W	0000_F100h	30.6.15/ 1583
11_20F0	GPMI Debug3 Information Register Description (GPMI_DEBUG3)	32	R	0000_0000h	30.6.16/ 1586
11_2100	GPMI Double Rate Read DLL Control Register Description (GPMI_READ_DDR_DLL_CTRL)	32	R/W	0000_0038h	30.6.17/ 1587
11_2110	GPMI Double Rate Write DLL Control Register Description (GPMI_WRITE_DDR_DLL_CTRL)	32	R/W	0000_0038h	30.6.18/ 1588

Table continues on the next page...

GPMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
11_2120	GPMI Double Rate Read DLL Status Register Description (GPMI_READ_DDR_DLL_STS)	32	R	0000_0000h	30.6.19/ 1590
11_2130	GPMI Double Rate Write DLL Status Register Description (GPMI_WRITE_DDR_DLL_STS)	32	R	0000_0000h	30.6.20/ 1591

30.6.1 GPMI Control Register 0 Description (GPMI_CTRL0*n*)

The GPMI control register 0 specifies the GPMI transaction to perform for the current command chain item.

Address: 11_2000h base + 0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SFTRST	CLKGATE	RUN	DEV_IRQ_EN	LOCK_CS	UDMA	COMMAND_MODE		WORD_LENGTH	CS			ADDRESS			ADDRESS_ INCREMENT
Reset	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	XFER_COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_CTRL0*n* field descriptions

Field	Description
31 SFTRST	Set to zero for normal operation. When this bit is set to one (default), then the entire block is held in its reset state. This will not work if the CLKGATE bit is already set to '1'. CLKGATE must be cleared to '0' before issuing a soft reset. Also the GPMICLK must be running for this to work properly. RUN = 0x0 Allow GPMI to operate normally. RESET = 0x1 Hold GPMI in reset.
30 CLKGATE	Set this bit zero for normal operation. Setting this bit to one (default),

Table continues on the next page...

GPMI_CTRL0*n* field descriptions (continued)

Field	Description
	gates all of the block level clocks off for miniminizing AC energy consumption. RUN = 0x0 Allow GPMI to operate normally. NO_CLKS = 0x1 Do not clock GPMI gates in order to minimize power consumption.
29 RUN	The GPMI is busy running a command whenever this bit is set to '1'. The GPMI is idle whenever this bit set to zero. This can be set to one by a CPU write. In addition, the DMA sets this bit each time a DMA command has finished its PIO transfer phase. IDLE = 0x0 The GPMI is idle. BUSY = 0x1 The GPMI is busy running a command.
28 DEV_IRQ_EN	When set to '1' and ATA_IRQ pin is asserted, the GPMI_IRQ output will assert.
27 LOCK_CS	For ATA/NAND mode: 0= Deassert chip select (CS) after RUN is complete. 1= Continue to assert chip select (CS) after RUN is complete. For Camera Mode: 0= Dont wait for VSYNC rising edge before capturing data. 1= Wait for VSYNC rising edge before capturing data (Camera mode only). DISABLED = 0x0 Deassert chip select (CS) after RUN is complete. ENABLED = 0x1 Continue to assert chip select (CS) after RUN is complete.
26 UDMA	DISABLED = 0x0 Use ATA-PIO mode on the external bus. ENABLED = 0x1 Use ATA-Ultra DMA mode on the external bus. 0 Use ATA-PIO mode on the external bus. 1 Use ATA-Ultra DMA mode on the external bus.
25–24 COMMAND_ MODE	WRITE = 0x0 Write mode. READ = 0x1 Read mode. READ_AND_COMPARE = 0x2 Read and Compare mode (setting sense flop). WAIT_FOR_READY = 0x3 Wait for Ready mode. For ATA WAIT_FOR_READY command set CS=01. 00 Write mode. 01 Read Mode. 10 Read and Compare Mode (setting sense flop). 11 Wait for Ready.
23 WORD_LENGTH	This bit should only be changed when RUN==0. Reserve = 0x0 Reserved. 8_BIT = 0x1 8-bit Data Bus mode. 0 Reserved. 1 8-bit Data Bus mode.
22–20 CS	Selects which chip select is active for this command. For ATA WAIT_FOR_READY command, this must be set to b01.
19–17 ADDRESS	Specifies the three address lines for ATA mode. In NAND mode, use A0 for CLE and A1 for ALE. NAND_DATA = 0x0 In NAND mode, this address is used to read and write data bytes. NAND_CLE = 0x1 In NAND mode, this address is used to write command bytes. NAND_ALE = 0x2 In NAND mode, this address is used to write address bytes.

Table continues on the next page...

GPMI_CTRL0*n* field descriptions (continued)

Field	Description
16 ADDRESS_ INCREMENT	In ATA mode, the address will increment with each cycle. In NAND mode, the address will increment once, after the first cycle (going from CLE to ALE). DISABLED = 0x0 Address does not increment. ENABLED = 0x1 Increment address. 0 Address does not increment. 1 Increment address.
XFER_COUNT	Number of bytes to transfer for this command. A value of zero will transfer 64K bytes.

30.6.2 GPMI Compare Register Description (GPMI_COMPARE)

The GPMI compare register specifies the expect data and the xor mask for comparing to the status values read from the device. This register is used by the Read and Compare command.

GPMI_COMPARE 0x010

Address: 11_2000h base + 10h offset = 11_2010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK																REFERENCE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

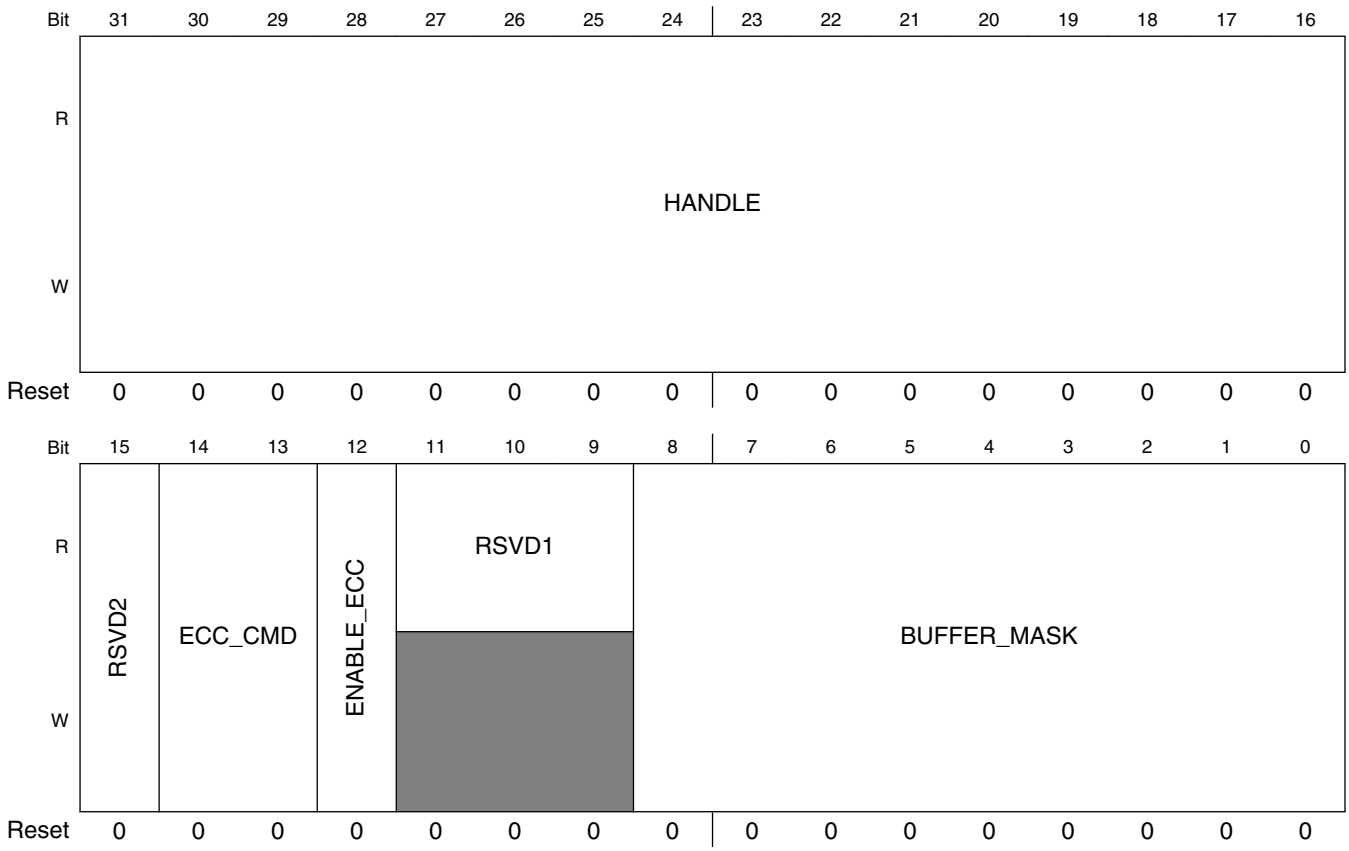
GPMI_COMPARE field descriptions

Field	Description
31–16 MASK	16-bit mask which is applied after the read data is XORed with the REFERENCE bit field.
REFERENCE	16-bit value which is XORed with data read from the NAND device.

30.6.3 GPMI Integrated ECC Control Register Description (GPMI_ECCCTRL*n*)

The GPMI ECC control register handles configuration of the integrated ECC accelerator.

Address: 11_2000h base + 20h offset + (4d × i), where i=0d to 3d



GPMI_ECCCTRL*n* field descriptions

Field	Description
31–16 HANDLE	This is a register available to software to attach an identifier to a transaction in progress. This handle will be available from the ECC register space when the completion interrupt occurs.
15 RSVD2	Always write zeroes to this bit field.
14–13 ECC_CMD	ECC Command information. DECODE = 0x0 Decode. ENCODE = 0x1 Encode. RESERVE2 = 0x2 Reserved. RESERVE3 = 0x3 Reserved.

Table continues on the next page...

GPMI_ECCCTRL*n* field descriptions (continued)

Field	Description
12 ENABLE_ECC	Enable ECC processing of GPMI transfers. ENABLE = 0x1 Use integrated ECC for read and write transfers. DISABLE = 0x0 Integrated ECC remains in idle.
11–9 RSVD1	Always write zeroes to this bit field.
BUFFER_MASK	ECC buffer information. The BCH error correction only allows two configurations of the buffer mask - software may either read just the first block on the flash page or the entire flash page. Write operations must be for the entire flash page. Invalid buffer mask values will cause the DMA descriptor command to be terminated. BCH_AUXONLY = 0x100 Set to request transfer from only the auxiliary buffer (block 0 on flash). BCH_PAGE = 0x1FF Set to request transfer to/from the entire page.

30.6.4 GPMI Integrated ECC Transfer Count Register Description (GPMI_ECCCOUNT)

The GPMI ECC Transfer Count Register contains the count of bytes that flow through the ECC subsystem.

GPMI_ECCCOUNT 0x030

Address: 11_2000h base + 30h offset = 11_2030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RSVD2																COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_ECCCOUNT field descriptions

Field	Description
31–16 RSVD2	Always write zeroes to this bit field.
COUNT	Number of bytes to pass through ECC. This is the GPMI transfer count plus the syndrome count that will be inserted into the stream by the ECC. In DMA2ECC_MODE this count must match the GPMI_CTRL0_XFER_COUNT. A value of zero will transfer 64K words.

30.6.5 GPMI Payload Address Register Description (GPMI_PAYLOAD)

The GPMI payload address register specifies the location of the data buffers in system memory. This value must be word aligned.

GPMI_PAYLOAD 0x040

Address: 11_2000h base + 40h offset = 11_2040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS														RSVD0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_PAYLOAD field descriptions

Field	Description
31–2 ADDRESS	Pointer to an array of one or more 512 byte payload buffers.
RSVD0	Always write zeroes to this bit field.

30.6.6 GPMI Auxiliary Address Register Description (GPMI_AUXILIARY)

The GPMI auxiliary address register specifies the location of the auxiliary buffers in system memory. This value must be word aligned.

GPMI_AUXILIARY 0x050

Address: 11_2000h base + 50h offset = 11_2050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS														RSVD0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_AUXILIARY field descriptions

Field	Description
31–2 ADDRESS	Pointer to ECC control structure and meta-data storage.
RSVD0	Always write zeroes to this bit field.

30.6.7 GPMI Control Register 1 Description (GPMI_CTRL1*n*)

The GPMI control register 1 specifies additional control fields that are not used on a per-transaction basis.

Address: 11_2000h base + 60h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DEV_CLK_STOP	SSYNC_CLK_STOP	WRITE_CLK_STOP	TOGGLE_MODE	GPMI_CLK_DIV2_EN	UPDATE_CS	SSYNCMODE	DECOUPLE_CS	WRN_DLY_SEL		RSVD1	TIMEOUT_IRQ_EN	GANGED_RDYBUSY	BCH_MODE	DLL_ENABLE	HALF_PERIOD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDN_DELAY				DMA2ECC_MODE	DEV_IRQ	TIMEOUT_IRQ	BURST_EN	ABORT_WAIT_REQUEST	ABORT_WAIT_FOR_READY_CHANNEL			DEV_RESET	ATA_IRQRDY_POLARITY	CAMERA_MODE	GPMI_MODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

GPMI_CTRL1*n* field descriptions

Field	Description
31 DEV_CLK_STOP	set this bit to 1 will stop gpmi io working clk.
30 SSYNC_CLK_ STOP	set this bit to 1 will stop the source synchronous mode clk.
29 WRITE_CLK_ STOP	In onfi source synchronous mode, host may save power during the data write cycles by holding the CLK signal high (i.e. stopping the CLK). The host may only stop the CLK during data write, by setting this bit to 1, if the device supports this feature as indicated in the parameter page.
28 TOGGLE_MODE	enable samsung toggle mode.
27 GPMI_CLK_ DIV2_EN	This bit should be reset to 0 in asynchronous mode. The frequence ratio of (device clock : ccm gpmi clock) will be (1 : 1). This bit should be set to 1, in source synchronous mode or toggle mode. The frequence raito of (device clock : ccm gpmi clock) will be (1 : 2). enable the gpmi clk divider. 0x0 internal factor-2 clock divider is disabled 0x1 internal factor-2 clock divider is enabled.
26 UPDATE_CS	force the CS value is be updated to external chip select pin, even GPMI is idle.
25 SSYNCMODE	source synchronouse mode 1 or asynchrous mode 0. ASYNC = 0x0 Asynchronous mode. SSYNC = 0x1 Source Synchronous mode.
24 DECOUPLE_CS	Decouple Chip Select from DMA Channel. Setting this bit to 1 will allow a DMA channel to specify any value in the CTRL0_CS register field. Software can use one DMA channel to access all 8 Nand devices.
23–22 WRN_DLY_SEL	Since the GPMI write strobe (WRN) is a fast clock pin, the delay on this signal can be programmed to match the load on this pin. 0 = 4ns to 8ns; 1 = 6ns to 10ns; 2 = 7ns to 12ns; 3 = no delay.
21 RSVD1	Always write zeroes to this bit field.
20 TIMEOUT_IRQ_ EN	Setting this bit to '1' will enable timeout IRQ for transfers in ATA mode only, and for WAIT_FOR_READY commands in both ATA and Nand mode. The Device_Busy_Timeout value is used for this timeout.
19 GANGED_ RDYBUSY	Set this bit to 1 will force all Nand RDY_BUSY inputs to be sourced from (tied to) RDY_BUSY0. This will free up all, except one, RDY_BUSY input pins.
18 BCH_MODE	This bit selects which error correction unit will access GPMI. This bit must always be set to '1', since only the BCH unit is available in this design.
17 DLL_ENABLE	Set this bit to 1 to enable the GPMI DLL. This is required for fast NAND reads (above 30 MHz read strobe). After setting this bit, wait 64 GPMI clock cycles for the DLL to lock before performing a NAND read.
16 HALF_PERIOD	Set this bit to 1 if the GPMI clock period is greater than 16ns for proper DLL operation. DLL_ENABLE must be zero while changing this field.
15–12 RDN_DELAY	This variable is a factor in the calculated delay to apply to the internal read strobe for correct read data sampling.

Table continues on the next page...

GPMI_CTRL1*n* field descriptions (continued)

Field	Description
	The applied delay (AD) is between 0 and 1.875 times the reference period (RP). RP is one half of the GPMI clock period if HALF_PERIOD=1 otherwise it is the full GPMI clock period. The equation is: AD = RDN_DELAY x 0.125 x RP. This value must not exceed 16ns. This variable is used to achieve faster NAND access. For example if the Read Strobe is asserted from time 0 to 13ns but the read access time is 20ns, then choose AD=12ns will cause the data to be sampled at time 25ns (13+12) giving a 5ns data setup time. If RP=13ns then RDN_DELAY = 12/(0.125 x 13ns) = 7.38 (0111b). DLL_ENABLE must be zero while changing this field.
11 DMA2ECC_MODE	This is mainly for testing HWECC without involving the Nand device. Setting this bit will cause DMA write data to redirected to HWECC module (instead of Nand Device) for encoding or decoding.
10 DEV_IRQ	This bit is set when an Interrupt is received from the ATA device. Write 0 to clear.
9 TIMEOUT_IRQ	This bit is set when a timeout occurs using the Device_Busy_Timeout value. Write 0 to clear.
8 BURST_EN	When set to 1 each DMA request will generate a 4-transfer burst on the APB bus.
7 ABORT_WAIT_REQUEST	Request to abort "wait for ready" command on channel indicated by ABORT_WAIT_FOR_READY_CHANNEL. Hardware will clear this bit when abort is done.
6–4 ABORT_WAIT_FOR_READY_CHANNEL	Abort a wait for ready command on selected channel. Set the ABORT_WAIT_REQUEST to kick of operation.
3 DEV_RESET	ENABLED = 0x0 NANDF_WP_B(WPN) pin is held low (asserted). DISABLED = 0x1 NANDF_WP_B(WPN) pin is held high (de-asserted). 0 NANDF_WP_B pin is held low (asserted). 1 NANDF_WP_B pin is held high (de-asserted).
2 ATA_IRQRDY_POLARITY	For ATA MODE: Note NAND_RDY_BUSY[3:2] are not affected by this bit. ACTIVELOW = 0x0 ATA IORDY and IRQ are active low, or NAND_RDY_BUSY[1:0] are active low ready. ACTIVEHIGH = 0x1 ATA IORDY and IRQ are active high, or NAND_RDY_BUSY[1:0] are active high ready. 0 External ATA IORDY and IRQ are active low. 1 External ATA IORDY and IRQ are active high. For NAND MODE: 0 External RDY_BUSY[1] and RDY_BUSY[0] pins are ready when low and busy when high. 1 External RDY_BUSY[1] and RDY_BUSY[0] pins are ready when high and busy when low.
1 CAMERA_MODE	When set to 1 and ATA UDMA is enabled the UDMA interface becomes a camera interface.
0 GPMI_MODE	ATA mode is only supported on channel zero. If ATA mode is selected, then only channel three is available for NAND use.

Table continues on the next page...

GPMI_CTRL1*n* field descriptions (continued)

Field	Description
	NAND = 0x0 NAND mode. ATA = 0x1 ATA mode. 0 NAND mode. 1 ATA mode.

30.6.8 GPMI Timing Register 0 Description (GPMI_TIMING0)

The GPMI timing register 0 specifies the timing parameters that are used by the cycle state machine to guarantee the various setup, hold and cycle times for the external media type.

GPMI_TIMING0 0x070

Address: 11_2000h base + 70h offset = 11_2070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R									ADDRESS_SETUP								DATA_HOLD								DATA_SETUP							
W	0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1

GPMI_TIMING0 field descriptions

Field	Description
31–24 RSVD1	Always write zeroes to this bit field.
23–16 ADDRESS_ SETUP	Number of GPMICLK cycles that the CE/ADDR signals are active before a strobe is asserted. A value of zero is interpreted as 0. For ATA PIO modes this is known in the ATA7 specification as "Address valid to DIOR-/DIOW- setup"
15–8 DATA_HOLD	Data bus hold time in GPMICLK cycles. Also the time that the data strobe is de-asserted in a cycle. A value of zero is interpreted as 256. For ATA PIO modes this is known in the ATA7 specification as "DIOR-/DIOW- recovery time"
DATA_SETUP	Data bus setup time in GPMICLK cycles. Also the time that the data strobe is asserted in a cycle. This value must be greater than 2 for ATA devices that use IORDY to extend transfer cycles. A value of zero is interpreted as 256. For ATA PIO modes this is known in the ATA7 specification as ""DIOR-/DIOW-"

30.6.9 GPMI Timing Register 1 Description (GPMI_TIMING1)

The GPMI timing register 1 specifies the timeouts used when monitoring the NAND READY pin or the ATA IRQ and IOWAIT signals.

GPMI_TIMING1 0x080

Address: 11_2000h base + 80h offset = 11_2080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEVICE_BUSY_TIMEOUT																RSVD1																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_TIMING1 field descriptions

Field	Description
31–16 DEVICE_BUSY_TIMEOUT	Timeout waiting for NAND Ready/Busy or ATA IRQ. Used in WAIT_FOR_READY mode. This value is the number of GPMI_CLK cycles multiplied by 4096.
RSVD1	Always write zeroes to this bit field.

30.6.10 GPMI Timing Register 2 Description (GPMI_TIMING2)

The GPMI timing register 2 specifies the double data rate timing parameters that are used by the cycle state machine to guarantee the various cs delay, pre-amble delay, post-amble delay, command/address delay, data delay, and read latency cycle times for the external media type.

GPMI_TIMING2 0x090

Address: 11_2000h base + 90h offset = 11_2090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD1					READ_LATENCY			RSVD0			CE_DELAY					PREAMBLE_DELAY				POSTAMBLE_DELAY				CMDADD_PAUSE				DATA_PAUSE				
W																																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	1	1	0

GPMI_TIMING2 field descriptions

Field	Description
31–27 RSVD1	Always write zeroes to this bit field.
26–24 READ_LATENCY	This field is for double data rate read latency configuration. others READ LATENCY is 3 000 READ LATENCY is 0 001 READ LATENCY is 1 010 READ LATENCY is 2 011 READ LATENCY is 3 100 READ LATENCY is 4 101 READ LATENCY is 5
23–21 RSVD0	Always write zeroes to this bit field.

Table continues on the next page...

GPMI_TIMING2 field descriptions (continued)

Field	Description
20–16 CE_DELAY	GPMI dealy from CEn assert to W/Rn changing edge. value of zero is interpreted as 32.
15–12 PREAMBLE_ DELAY	GPMI pre-amble delay in GPMICLK cycles. A value of zero is interpreted as 16.
11–8 POSTAMBLE_ DELAY	GPMI post-amble delay in GPMICLK cycles. A value of zero is interpreted as 16.
7–4 CMDADD_ PAUSE	GPMI delay time from command or addres pause to command or address resume in GPMICLK cycles. A value of zero is interpreted as 16.
DATA_PAUSE	GPMI delay time from data pause to data resume in GPMICLK cycles. A value of zero is interpreted as 16.

30.6.11 GPMI DMA Data Transfer Register Description (GPMI_DATA)

The GPMI DMA data transfer register is used by the DMA to read or write data to or from the ATA/NAND control state machine.

GPMI_DATA 0x0A0

Address: 11_2000h base + A0h offset = 11_20A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_DATA field descriptions

Field	Description
DATA	In 8-bit mode, one, two, three or four bytes can can be accessed to send the same number of bus cycles.

30.6.12 GPMI Status Register Description (GPMI_STAT)

The GPMI control and status register provides a read back path for various operational states of the GPMI controller.

GPMI_STAT 0x0B0

Address: 11_2000h base + B0h offset = 11_20B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	READY_BUSY								RDY_TIMEOUT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEV7_ERROR	DEV6_ERROR	DEV5_ERROR	DEV4_ERROR	DEV3_ERROR	DEV2_ERROR	DEV1_ERROR	DEV0_ERROR	RSVD1			ATA_IRQ	INVALID_BUFFER_MASK	FIFO_EMPTY	FIFO_FULL	PRESENT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

GPMI_STAT field descriptions

Field	Description
31–24 READY_BUSY	Read-only view of NAND Ready_Busy Input pins.
23–16 RDY_TIMEOUT	State of the RDY/BUSY Timeout Flags. When any bit is set to '1' in this field, it indicates that a time out has occurred while waiting for the ready state of the requested NAND device. Multiple bits may be set simultaneously. When GPMI_CTRL1_DECOUPLE_CS = 0, RDY_TIMEOUT[n] is associated with the NAND device on chip_select[n]. When GPMI_CTRL1_DECOUPLE_CS = 1, these flags become associated to a DMA channel instead of a NAND device. For example if DMA channel 6 sends a WAIT_FOR_READY command for NAND Device 2, and a timeout occured on READY_BUSY2, then READY_TIMEOUT[6] will be set instead of READY_TIMEOUT[2].
15 DEV7_ERROR	DMA channel 7 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 7. 1 An Error has occurred on ATA/NAND Device accessed by
14 DEV6_ERROR	DMA channel 6 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 6. 1 An Error has occurred on ATA/NAND Device accessed by
13 DEV5_ERROR	DMA channel 5 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 5. 1 An Error has occurred on ATA/NAND Device accessed by
12 DEV4_ERROR	DMA channel 4 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 4. 1 An Error has occurred on ATA/NAND Device accessed by
11 DEV3_ERROR	DMA channel 3 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 3. 1 An Error has occurred on ATA/NAND Device accessed by
10 DEV2_ERROR	DMA channel 2 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 2. 1 An Error has occurred on ATA/NAND Device accessed by
9 DEV1_ERROR	DMA channel 1 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 1. 1 An Error has occurred on ATA/NAND Device accessed by
8 DEV0_ERROR	DMA channel 0 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 0. 1 An Error has occurred on ATA/NAND Device accessed by
7–5 RSVD1	Always write zeroes to this bit field.

Table continues on the next page...

GPMI_STAT field descriptions (continued)

Field	Description
4 ATA_IRQ	Status of the ATA_IRQ input pin.
3 INVALID_ BUFFER_MASK	Buffer Mask Validity bit. 0 ECC Buffer Mask is not invalid. 1 ECC Buffer Mask is invalid.
2 FIFO_EMPTY	NOT_EMPTY = 0x0 FIFO is not empty. EMPTY = 0x1 FIFO is empty. 0 FIFO is not empty. 1 FIFO is empty.
1 FIFO_FULL	NOT_FULL = 0x0 FIFO is not full. FULL = 0x1 FIFO is full. 0 FIFO is not full. 1 FIFO is full.
0 PRESENT	UNAVAILABLE = 0x0 GPMI is not present in this product. AVAILABLE = 0x1 GPMI is present in this product. 0 GPMI is not present in this product. 1 GPMI is present is in this product.

30.6.13 GPMI Debug Information Register Description (GPMI_DEBUG)

The GPMI debug information register provides a read back path for diagnostics to determine the current operating state of the GPMI controller.

GPMI_DEBUG 0x0C0

Address: 11_2000h base + C0h offset = 11_20C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WAIT_FOR_READY_END								DMA_SENSE								DMAREQ								CMD_END							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_DEBUG field descriptions

Field	Description
31–24 WAIT_FOR_ READY_END	Read Only view of the Wait_For_Ready End toggle signals to DMA. One per channel

Table continues on the next page...

GPMI_DEBUG field descriptions (continued)

Field	Description
23–16 DMA_SENSE	Read-only view of sense state of the 8 DMA channels. A value of "1" in any bit position indicates that a read and compare command failed or a timeout occured for the corresponding channel.
15–8 DMAREQ	Read-only view of DMA request line for 8 DMA channels. A toggle on any bit position indicates a DMA request for the corresponding channel.
CMD_END	Read Only view of the Command End toggle signals to DMA. One per channel

30.6.14 GPMI Version Register Description (GPMI_VERSION)

This register reflects the version number for the GPMI.

GPMI_VERSION 0x0D0

Address: 11_2000h base + D0h offset = 11_20D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR								MINOR								STEP															
W																																
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_VERSION field descriptions

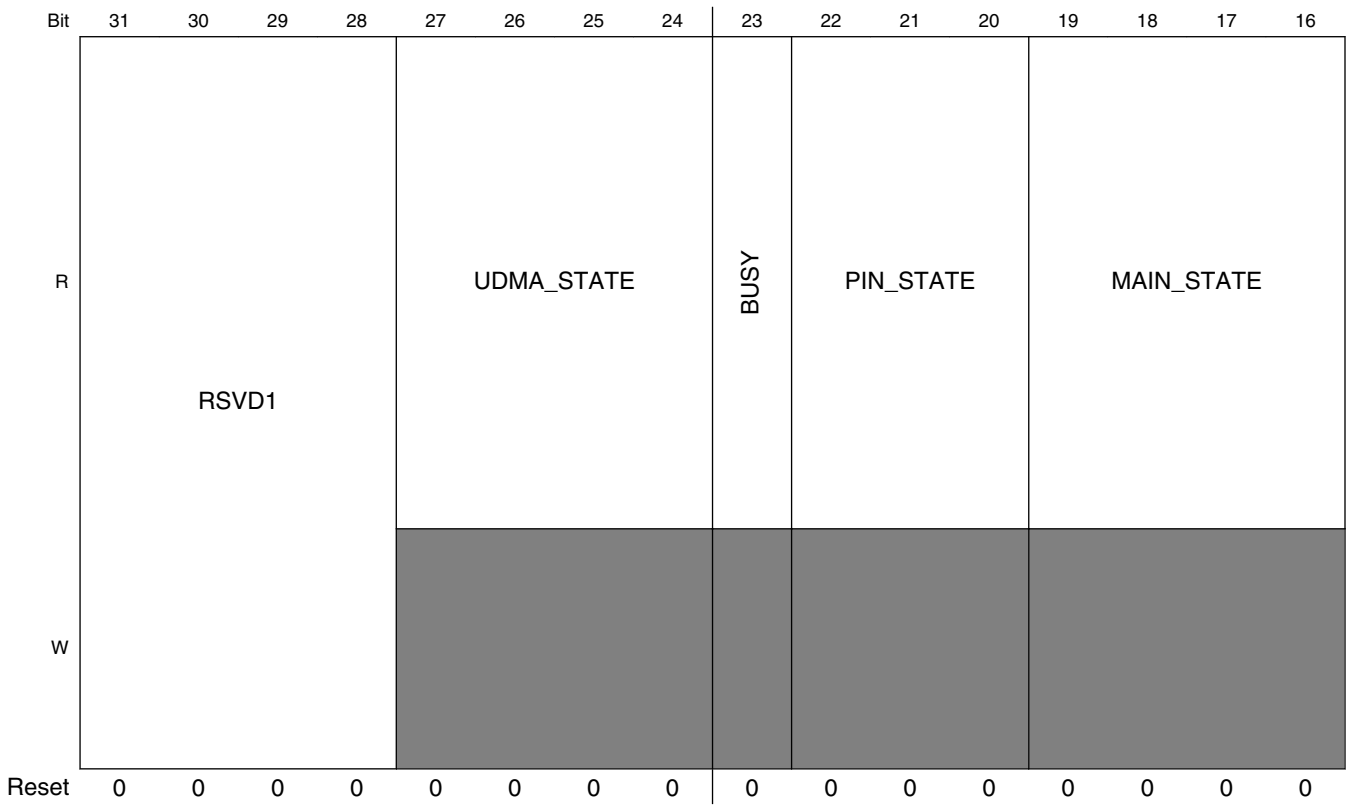
Field	Description
31–24 MAJOR	Fixed read-only value reflecting the MAJOR field of the RTL version.
23–16 MINOR	Fixed read-only value reflecting the MINOR field of the RTL version.
STEP	Fixed read-only value reflecting the stepping of the RTL version.

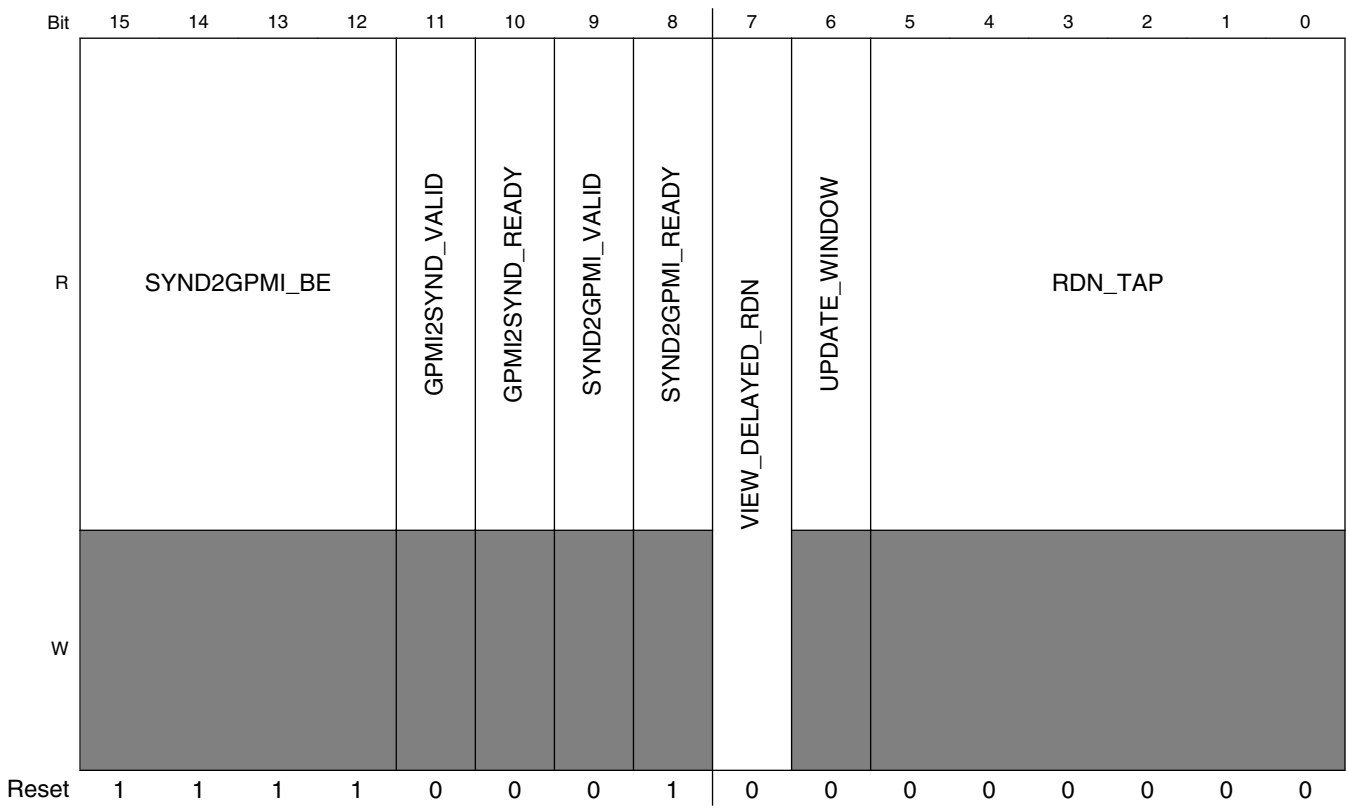
30.6.15 GPMI Debug2 Information Register Description (GPMI_DEBUG2)

The GPMI Debug2 information register provides a read back path for diagnostics to determine the current operating state of the GPMI controller.

GPMI_DEBUG2 0x0E0

Address: 11_2000h base + E0h offset = 11_20E0h





GPMI_DEBUG2 field descriptions

Field	Description
31–28 RSVD1	Always write zeroes to this bit field.
27–24 UDMA_STATE	USM_IDLE = 4'h0, idle USM_DMARQ = 4'h1, DMA req USM_ACK = 4'h2, DMA ACK USM_FIFO_E = 4'h3, Fifo empty USM_WPAUSE = 4'h4, WR DMA Paused by device USM_TSTRB = 4'h5, Toggle HSTROBE USM_CAPTUR = 4'h6, Capture Stage, (data sampled with DSTROBE is valid) USM_DATOUT = 4'h7, Change Burst DATAOUT USM_CRC = 4'h8, Source CRC to Device USM_WAIT_R = 4'h9, Waiting for DDMARDY- USM_END = 4'ha; Negate DMAACK (end of DMA) USM_WAIT_S = 4'hb, Waiting for DSTROBE USM_RPAUSE = 4'hc, Rd DMA Paused by Host USM_RSTOP = 4'hd, Rd DMA Stopped by Host USM_WTERM = 4'he, Wr DMA Termination State USM_RTERM = 4'hf, Rd DMA Termination state
23 BUSY	When asserted the GPMI is busy. Undefined results may occur if any registers are written when BUSY is asserted. DISABLED = 0x0 The GPMI is not busy. ENABLED = 0x1 The GPMI is busy.
22–20 PIN_STATE	parameter PSM_IDLE = 3'h0, PSM_BYTCNT = 3'h1, PSM_ADDR = 3'h2, PSM_STALL = 3'h3, PSM_STROBE = 3'h4, PSM_ATARDY = 3'h5, PSM_DHOLD = 3'h6, PSM_DONE = 3'h7. PSM_IDLE = 0x0 PSM_BYTCNT = 0x1 PSM_ADDR = 0x2 PSM_STALL = 0x3 PSM_STROBE = 0x4 PSM_ATARDY = 0x5 PSM_DHOLD = 0x6 PSM_DONE = 0x7
19–16 MAIN_STATE	parameter MSM_IDLE = 4'h0, MSM_BYTCNT = 4'h1, MSM_WAITFE = 4'h2, MSM_WAITFR = 4'h3, MSM_DMAREQ = 4'h4, MSM_DMAACK = 4'h5, MSM_WAITFF = 4'h6, MSM_LDFIFO = 4'h7, MSM_LDDMAR = 4'h8, MSM_RDCMP = 4'h9, MSM_DONE = 4'hA. MSM_IDLE = 0x0 MSM_BYTCNT = 0x1 MSM_WAITFE = 0x2 MSM_WAITFR = 0x3

Table continues on the next page...

GPMI_DEBUG2 field descriptions (continued)

Field	Description
	MSM_DMAREQ = 0x4 MSM_DMAACK = 0x5 MSM_WAITFF = 0x6 MSM_LDFIFO = 0x7 MSM_LDDMAR = 0x8 MSM_RDCMP = 0x9 MSM_DONE = 0xA
15–12 SYND2GPMI_BE	Data byte enable Input from BCH.
11 GPMI2SYND_VALID	Data handshake output to BCH.
10 GPMI2SYND_READY	Data handshake output to BCH.
9 SYND2GPMI_VALID	Data handshake Input from BCH.
8 SYND2GPMI_READY	Data handshake Input from BCH.
7 VIEW_DELAYED_RDN	Set to a 1 to select the delayed feedback RE_B to drive the GPMI_ADDR[0] (Nand CLE) pin. For debug purposes, this will allow you see if DLL is functioning properly.
6 UPDATE_WINDOW	A 1 indicates that the DLL is busy generating the required delay.
RDN_TAP	This is the DLL tap calculated by the DLL controller. The selects the amount of delay form the DLL chain.

30.6.16 GPMI Debug3 Information Register Description (GPMI_DEBUG3)

The GPMI Debug3 information register provides a read back path for diagnostics to determine the current operating state of the GPMI controller.

GPMI_DEBUG3 0x0F0

Address: 11_2000h base + F0h offset = 11_20F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	APB_WORD_CNTR																DEV_WORD_CNTR															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_DEBUG3 field descriptions

Field	Description
31–16 APB_WORD_ CNTR	Reflects the number of bytes remains to be transferred on the APB bus.
DEV_WORD_ CNTR	Reflects the number of bytes remains to be transferred on the ATA/Nand bus.

30.6.17 GPMI Double Rate Read DLL Control Register Description (GPMI_READ_DDR_DLL_CTRL)

GPMI DDR Read Delay Loop Lock Control Register. This register provides programmability in DDR mode for data input timing and data formats.

GPMI_READ_DDR_DLL_CTRL 0x100

Address: 11_2000h base + 100h offset = 11_2100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	REF_UPDATE_INT				SLV_UPDATE_INT								RSVD1		SLV_ OVERRIDE_ VAL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SLV_OVERRIDE_VAL						SLV_OVERRIDE	REFCLK_ON	GATE_UPDATE	SLV_DLY_TARGET				SLV_FORCE_UPD	RESET	ENABLE
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0

GPMI_READ_DDR_DLL_CTRL field descriptions

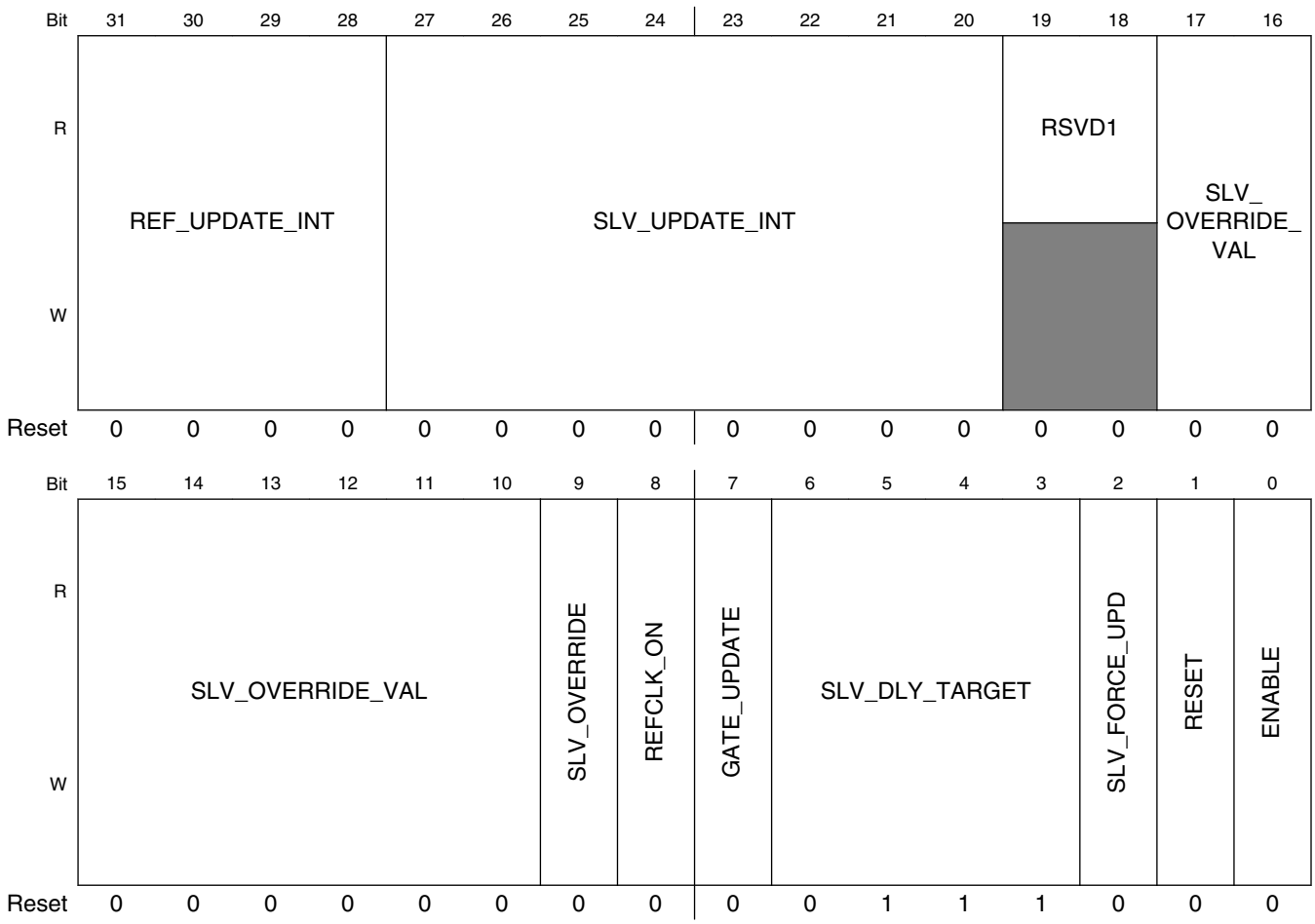
Field	Description
31–28 REF_UPDATE_INT	This field allows the user to add additional delay cycles to the DLL control loop (reference delay line control). By default, the DLL control loop shall update every two GPMICLK cycles. Programming this field results in a DLL control loop update interval of (2 + REF_UPDATE_INT) * GPMICLK. It should be noted that increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that may effect the delay (such as voltage and temperature)
27–20 SLV_UPDATE_INT	Setting a value greater than 0 in this field, shall over-ride the default slave delay-line update interval of 256 GPMICLK cycles. A value of 0 results in an update interval of 256 GPMICLK cycles (default setting). A value of 0x0f results in 15 cycles and so on. Note that software can always cause an update of the slave-delay line using the SLV_FORCE_UPDATE register. Note that the slave delay line will also update automatically when the reference DLL transitions to a locked state (from an un-locked state).
19–18 RSVD1	Reserved
17–10 SLV_OVERRIDE_VAL	When SLV_OVERRIDE=1 This field is used to select 1 of 256 physical taps manually. A value of 0 selects tap 1, and a value of 0x7f selects tap 256.
9 SLV_OVERRIDE	Set this bit to 1 to Enable manual override for slave delay chain using SLV_OVERRIDE_VAL; to set 0 to disable manual override. This feature does not require the DLL to tbe enabled using the ENABLE bit. In fact to reduce power, if SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE=0
8 REFCLK_ON	set this bit to 1 will turn on the reference clock
7 GATE_UPDATE	Setting this bit to 1, forces the slave delay line not update
6–3 SLV_DLY_TARGET	The delay target for the read clock is can be programmed in 1/16th increments of an GPMICLK half-period. So the input read-clock can be delayed relative input data from (GPMICLK/2)/16 to GPMICLK/2.
2 SLV_FORCE_UPD	Setting this bit to 1, forces the slave delay line to update to the DLL calibrated value immediately. The slave delay line shall update automatically based on the SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the slave-line update can only occur if SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered).
1 RESET	Setting this bit to 1 force a reset on DLL. This will cause the DLL to lose lock and re-calibrate to detect an GPMICLK half period phase shift. This signal is used by the DLL as edge-sensitive, so in order to create a subsequent reset, RESET must be taken low and then asserted again.
0 ENABLE	Set this bit to 1 to enable the DLL and delay chain; otherwise; set to 0 to bypasses DLL. Note that using the slave delay line override feature with SLV_OVERRIDE and SLV_OVERRIDE VAL, the DLL does not need to be enabled.

30.6.18 GPMI Double Rate Write DLL Control Register Description (GPMI_WRITE_DDR_DLL_CTRL)

GPMI DDR Write Delay Loop Lock Control Register. This register provides programmability in DDR mode for data output timing and data formats.

GPMI_WRITE_DDR_DLL_CTRL 0x110

Address: 11_2000h base + 110h offset = 11_2110h



GPMI_WRITE_DDR_DLL_CTRL field descriptions

Field	Description
31–28 REF_UPDATE_ INT	This field allows the user to add additional delay cycles to the DLL control loop (reference delay line control). By default, the DLL control loop shall update every two GPMICLK cycles. Programming this field results in a DLL control loop update interval of (2 + REF_UPDATE_INT) * GPMICLK. It should be noted that increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that may effect the delay (such as voltage and temperature)
27–20 SLV_UPDATE_ INT	Setting a value greater than 0 in this field, shall over-ride the default slave delay-line update interval of 256 GPMICLK cycles. A value of 0 results in an update interval of 256 GPMICLK cycles (default setting). A value of 0x0f results in 15 cycles and so on. Note that software can always cause an update of the slave-delay line using the SLV_FORCE_UPDATE register. Note that the slave delay line will also update automatically when the reference DLL transitions to a locked state (from an un-locked state).
19–18 RSVD1	Reserved
17–10 SLV_ OVERRIDE_VAL	When SLV_OVERRIDE=1 This field is used to select 1 of 256 physical taps manually. A value of 0 selects tap 1, and a value of 0x7f selects tap 256.
9 SLV_OVERRIDE	Set this bit to 1 to Enable manual override for slave delay chain using SLV_OVERRIDE_VAL; to set 0 to disable manual override. This feature does not require the DLL to tbe enabled using the ENABLE bit. In fact to reduce power, if SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE=0

Table continues on the next page...

GPMI_WRITE_DDR_DLL_CTRL field descriptions (continued)

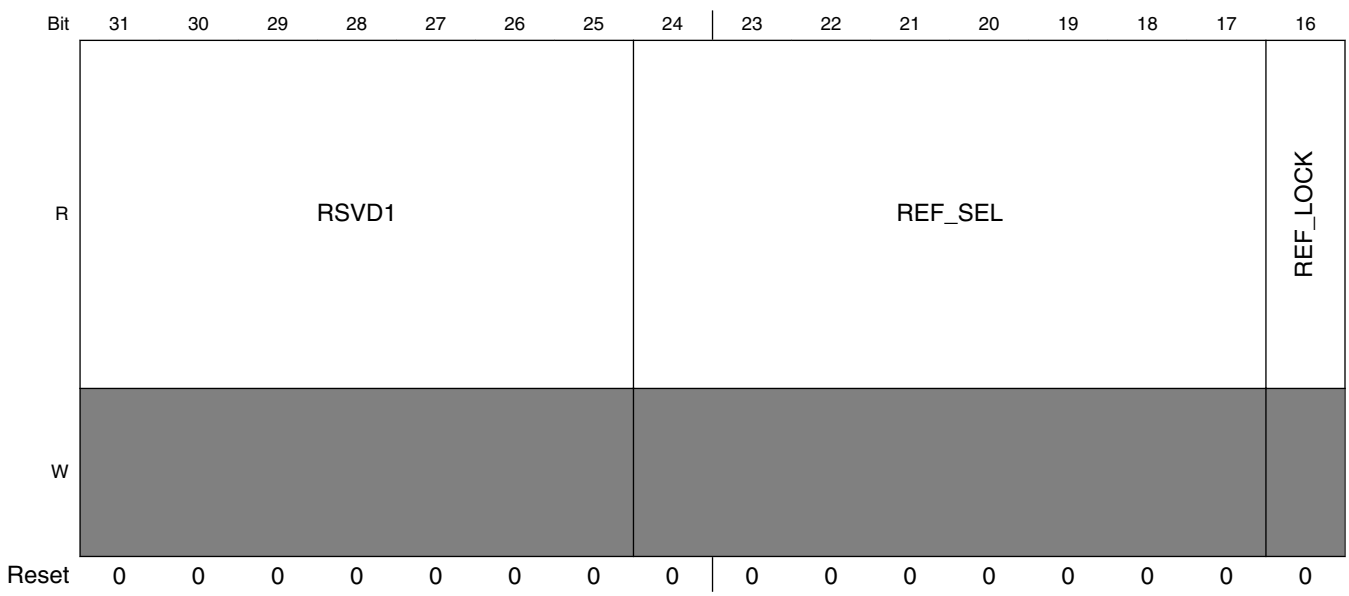
Field	Description
8 REFCLK_ON	set this bit to 1 will turn on the reference clock
7 GATE_UPDATE	Setting this bit to 1, forces the slave delay line not update
6–3 SLV_DLY_ TARGET	The delay target for the read clock can be programmed in 1/16th increments of an GPMICLK half-period. So the input read-clock can be delayed relative input data from (GPMICLK/2)/16 to GPMICLK/2.
2 SLV_FORCE_ UPD	Setting this bit to 1, forces the slave delay line to update to the DLL calibrated value immediately. The slave delay line shall update automatically based on the SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the slave-line update can only occur if SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered).
1 RESET	Setting this bit to 1 force a reset on DLL. This will cause the DLL to lose lock and re-calibrate to detect an GPMICLK half period phase shift. This signal is used by the DLL as edge-sensitive, so in order to create a subsequent reset, RESET must be taken low and then asserted again.
0 ENABLE	Set this bit to 1 to enable the DLL and delay chain; otherwise; set to 0 to bypasses DLL. Note that using the slave delay line override feature with SLV_OVERRIDE and SLV_OVERRIDE VAL, the DLL does not need to be enabled.

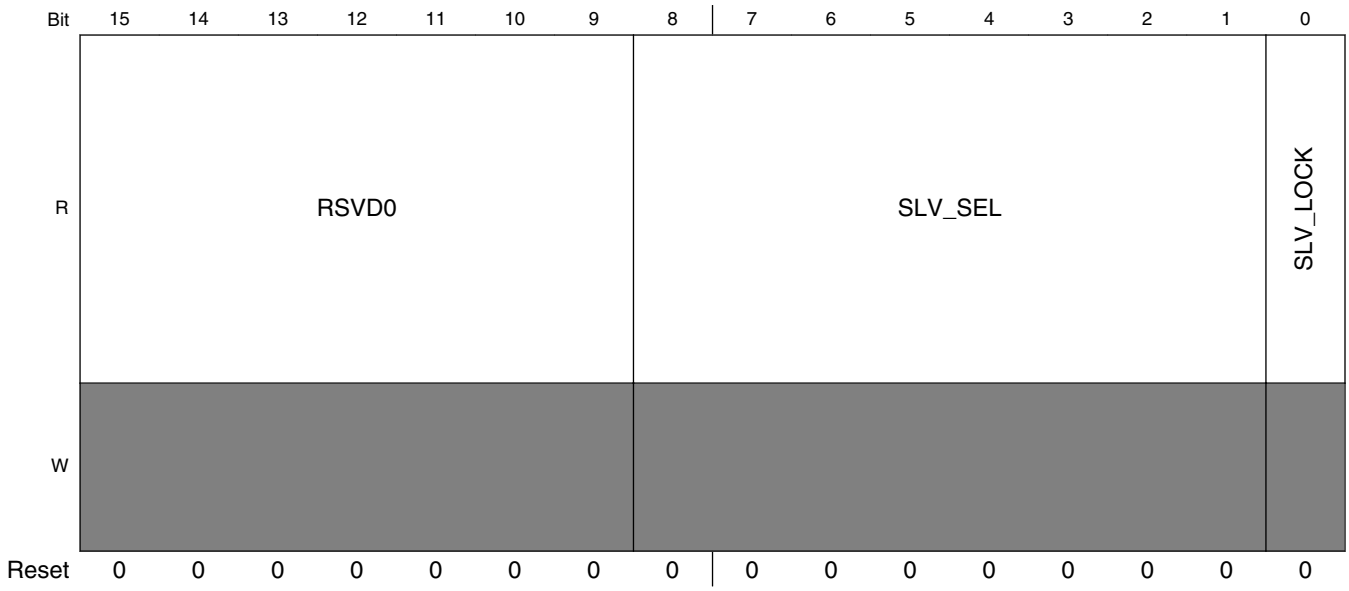
30.6.19 GPMI Double Rate Read DLL Status Register Description (GPMI_READ_DDR_DLL_STS)

GPMI Double Rate Read DLL Status Register, Read Only. GPMI DLL status fields are provided in this register.

GPMI_READ_DDR_DLL_STS 0x120

Address: 11_2000h base + 120h offset = 11_2120h





GPMI_READ_DDR_DLL_STS field descriptions

Field	Description
31–25 RSVD1	Reserved
24–17 REF_SEL	Reference delay line select status.
16 REF_LOCK	Reference DLL lock status. This signifies that the DLL has detected and locked to a half-phase GPMICLK shift, allowing the slave delay-line to perform programmed clock delays.
15–9 RSVD0	Reserved
8–1 SLV_SEL	Slave delay line select status
0 SLV_LOCK	Slave delay-line lock status. This signifies that a valid calibration has been set to the slave-delay line and that the slave-delay line is implementing the programmed delay value.

30.6.20 GPMI Double Rate Write DLL Status Register Description (GPMI_WRITE_DDR_DLL_STS)

GPMI Double Rate Write DLL Status Register, Read Only. GPMI DLL status fields are provided in this register.

GPMI_WRITE_DDR_DLL_STS 0x130

Address: 11_2000h base + 130h offset = 11_2130h



GPMI_WRITE_DDR_DLL_STS field descriptions

Field	Description
31–25 RSVD1	Reserved
24–17 REF_SEL	Reference delay line select status.
16 REF_LOCK	Reference DLL lock status. This signifies that the DLL has detected and locked to a half-phase GPMICLK shift, allowing the slave delay-line to perform programmed clock delays.
15–9 RSVD0	Reserved
8–1 SLV_SEL	Slave delay line select status

Table continues on the next page...

GPMI_WRITE_DDR_DLL_STS field descriptions (continued)

Field	Description
0 SLV_LOCK	Slave delay-line lock status. This signifies that a valid calibration has been set to the slave-delay line and that the slave-delay line is implementing the programmed delay value.

Chapter 31
General Purpose Timer (GPT)

31.1 Overview

This chapter describes the General Purpose Timer (GPT) module interface. It is also a reference for software driver programming. The GPT has a 32-bit up-counter. The timer counter value can be captured in a register using an event on an external pin. The capture trigger can be programmed to be a rising or/and falling edge. The GPT can also generate an event on the output compare pins and an interrupt when the timer reaches a programmed value. The GPT has a 12-bit prescaler, which provides a programmable clock frequency derived from multiple clock sources.

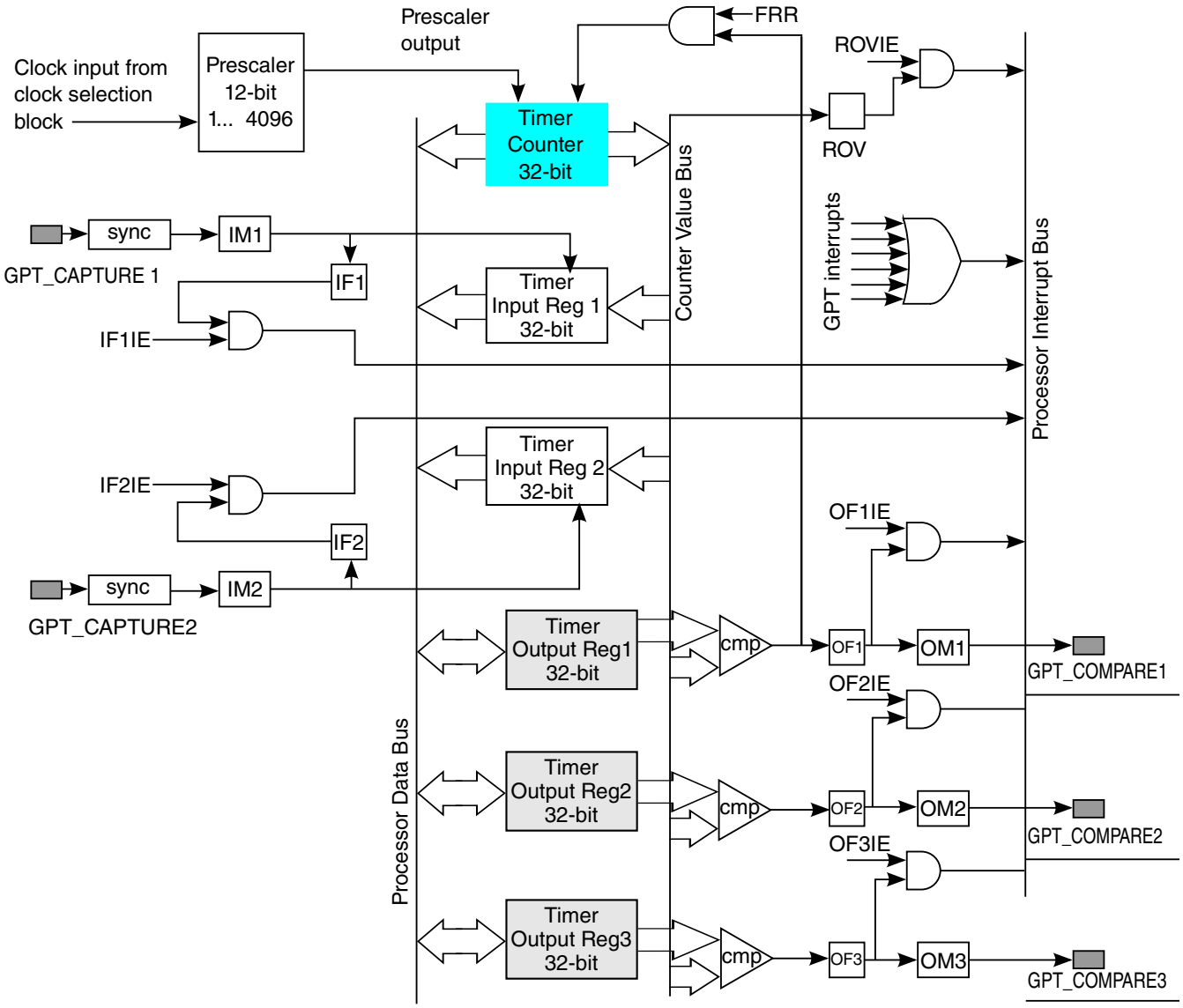


Figure 31-1. GPT Block Diagram

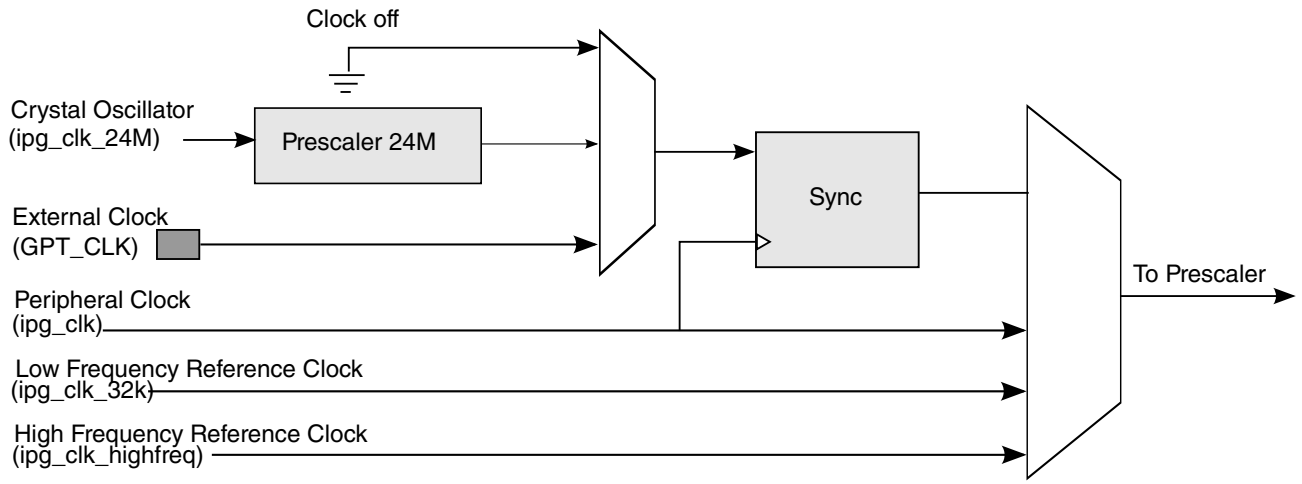


Figure 31-2. GPT Counter Clocks Diagram

31.1.1 Features

- One 32-bit up-counter with clock source selection, including external clock.
- Two input capture channels with a programmable trigger edge.
- Three output compare channels with a programmable output mode. A "forced compare" feature is also available.
- Can be programmed to be *active* in low power and debug modes.
- Interrupt generation at capture, compare, and rollover events.
- Restart or free-run modes for counter operations.

31.1.2 Modes and Operation

The GPT supports the modes described in the indicated sections:

- Operating Modes
  - Restart Mode
  - Free-Run Mode

31.2 External Signals

The GPT follows the IP Bus protocol for interfacing with the processor core. The GPT does not have *any interface signals with any other module inside the chip,* except for the clock and reset inputs (from the clock and reset controller module) and for the interrupt signals *to* the processor interrupt handler. There are functional and clock inputs, and functional output signals going outside the chip boundary.

The following table describes all block signals that connect off-chip.

Table 31-1. GPT External Signals

Signal	Description	Pad	Mode	Direction
GPT_CLK	Input pin for an external clock that the counter can be operated at.	SD1_CLK	ALT3	I
GPT_CAPTURE1	Input pin for a capture event for Input Capture Channel 1.	SD1_DAT0	ALT3	I
GPT_CAPTURE2	Input pin for a capture event for Input Capture Channel 2.	SD1_DAT1	ALT3	I
GPT_COMPARE1	Output pin that indicates a "compare event" occurrence in Output Compare Channel 1.	SD1_CMD	ALT3	O
GPT_COMPARE2	Output pin that indicates a "compare event" occurrence in Output Compare Channel 2.	SD1_DAT2	ALT2	O
GPT_COMPARE3	Output pin that indicates a "compare event" occurrence in Output Compare Channel 3.	SD1_DAT3	ALT2	O

There are six signals (three input, three output) in the GPT module that *can be* connected to the chip pads.

31.2.1 External Clock Input

The GPT counter can be operated using an external clock from outside the device, and this is the input pin used for that purpose. The external clock input (GPT_CLK) is treated as asynchronous to the peripheral clock (ipg_clk). To ensure proper operations of GPT, the external clock input frequency should be less than 1/4 of frequency of the peripheral clock (ipg_clk). Hysteresis characteristics on this pad will be required because this is a clock input.

31.2.2 Input Capture Trigger Signals

The GPT counter value can be stored in a register, triggered by an event from *outside the device*. A positive or/and negative edge on these signals GPT_CAPTURE1 , GPT_CAPTURE2 can trigger this capture event. These signals are treated as asynchronous to the peripheral clock (ipg_clk). Only those transitions which occur *at least a single clock cycle* (the clock selected to run the counter) *after the previous recorded transition* are guaranteed to trigger a capture event.

31.2.3 Output Compare Signals

The output compare signals: GPT_COMPARE1, GPT_COMPARE2, GPT_COMPARE3, indicate that output compare events have gone through a specified transition.

31.3 Clocks

The clock that is input to the prescaler can be selected from 4 clock sources. The following table describes the clock sources for GPT. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 31-2. GPT Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_32k	ckil_sync_clk_root	Low-frequency reference clock (32 kHz)
ipg_clk_highfreq	perclk_clk_root	High-frequency reference clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

- High-Frequency Clock (ipg_clk_highfreq)

  Provided by the Clock Controller Module (CCM), the High Frequency Clock is intended to be ON in Normal Power mode when the Peripheral Clock (ipg_clk) is turned OFF, thereby enabling the GPT to be operated using the High Frequency Clock *in Normal Power mode*. The CCM is expected to provide this clock *after* synchronizing it to the System Bus Clock (ahb_clk) in Normal functional mode; the CCM is also expected to switch to the *unsynchronized* version of the High Frequency Clock in a Low Power mode.

- Low-Reference Clock (ipg_clk_32k)

This 32 kHz Low Reference Clock (provided by the CCM) is intended to be ON in Low Power mode when the Peripheral Clock (ipg_clk) is turned OFF, thereby enabling the GPT to be operated using the Low Reference Clock in Low Power mode. The CCM is expected to provide the Low Reference Clock *after* synchronizing it to the System Bus Clock (ahb_clk) in Normal functional mode; the CCM is also expected to switch to the *unsynchronized* version of the Low Reference Clock in a Low Power mode.

- Peripheral Clock (ipg_clk)

  If the Peripheral Clock (ipg_clk) or the External Clock (GPT_CLK) is selected (CLKSRC=001 or 011) as Clock Source, then the Peripheral Clock will be ON in normal GPT operations. In Low Power modes, if the GPT is programmed to be disabled (STOPEN or WAITEN or DOZEN=0), then the Peripheral Clock (ipg_clk) can be switched OFF.

- External Clock (GPT_CLK)

  The External Clock comes from *outside the device* and can be selected to run the GPT counter. The External Clock is treated as *asynchronous to the Peripheral Clock,* (ipg_clk) and is synchronized to the Peripheral Clock (ipg_clk), *inside* the module. Therefore, the External Clock frequency is limited to < 1/4 frequency of the Peripheral Clock (ipg_clk), for proper GPT operations. Note that in Low Power modes, *if* the Peripheral Clock (ipg_clk) is not available, then the External Clock *cannot be used* to run the counter.

- Crystal Oscillator Clock (ipg_clk_24M)

  This 24 MHz Crystal Oscillator Clock (provided by the CCM) is intended to be used against frequency change of Peripheral Clock (ipg_clk) changes to provide a more accurate timer clock for operation system. The CCM is expected to provide the 24 MHz Crystal Oscillator Clock *without* synchronizing it to the System Bus Clock (ahb_clk) in Normal functional mode. Synchronization is done in GPT module. Before synchronization, the 24 MHz Crystal Oscillator Clock is divided by a 24 MHz clock prescaler, to make sure the clock frequency less than half of System Bus Clock (ahb_clk).

The clock input source is configured using the clock source field (CLKSRC, in the GPT_CR control register). The clock input to the prescaler can be disabled by programming the CLKSRC bits (of the GPT_CR control register) to 000. **The CLKSRC field value should be changed** *only after disabling the GPT* (by setting the EN bit in the GPT_CR to 0).

The PRESCALER field selects the divide ratio of the input clock that drives the main counter. The prescaler can divide the input clock by a value (from 1 to 4096) and can be changed *at any time*. A change in the value of the PRESCALER field *immediately affects* the output clock frequency.

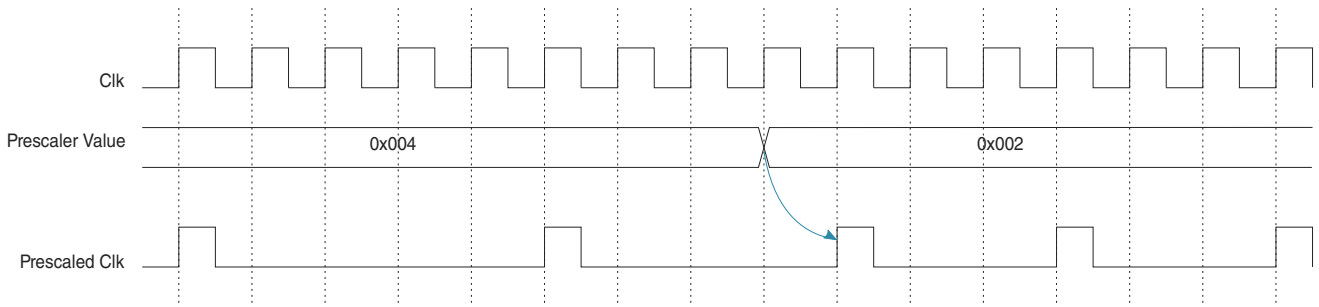


Figure 31-3. Prescaler Value Change Timing Diagram

31.4 Functional Description

This section provides a complete functional description of the GPT.

31.4.1 Operating Modes

The GPT counter can be programmed to work in either of two modes: Restart mode or Free-Run mode.

31.4.1.1 Restart Mode

In Restart mode (selectable through the GPT Control Register GPT_CR), when the counter reaches the compared value, the counter resets and starts again from 0x00000000. The Restart feature is associated only with Compare Channel 1.

Any write access to the Compare register of Channel 1 will reset the GPT counter. This is done to avoid possibly missing a compare event when compare value is changed from a higher value to lower value while counting is proceeding.

For the other two compare channels, when the compare event occurs the counter is *not reset*.

31.4.1.2 Free-Run Mode

In Free-Run mode, when compare events occur for all 3 channels, the counter is *not reset*; instead the counter continues to count until 0xffffffff, and then rolls over (to 0x00000000).

31.4.2 Operation

The General Purpose Timer (GPT) has a single counter (GPT_CNT) that is a 32-bit free-running *up-counter,* which starts counting *after it is enabled by software* (EN=1). The counter's clock source is the output of the prescaler labelled "Prescaler output" in Figure 31-1.

- If the GPT timer is disabled (EN=0), then the Main Counter *and* Prescaler Counter freeze their current count values. The ENMOD bit determines the value of the GPT counter when the EN bit is set and the Counter is enabled again.
  - If the ENMOD bit is set (=1), then the Main Counter and Prescaler Counter values are reset to 0, when GPT is enabled (EN=1).
  - If ENMOD bit is programmed to 0, then the Main Counter and Prescaler Counter restart counting from their frozen values, when GPT is enabled again (EN=1).
- If GPT is programmed to be disabled in a low power mode (STOP/WAIT), then the Main Counter and Prescaler Counter freeze at their current count values *when* GPT enters low power mode. When GPT exits a low power mode, the Main Counter and Prescaler Counter start counting from their frozen values *regardless* of the ENMOD bit value. Note that the GPT_CNT can be read *at any time* by the processor, and that *both* Input Capture Channels use the *same* counter (GPT_CNT).
- A hardware reset resets all the GPT registers to their respective reset values. All registers except the Output Compare Registers (OCR1, OCR2, OCR3) obtain a value of 0x0. The Compare registers are reset to 0xFFFF_FFFF.
- The software reset (SWR bit in the GPT_CR control register) resets *all* of the register bits *except* the EN, ENMOD, STOPEN, WAITEN, and DBGEN bits. The state of these bits is not affected by a software reset. Note that a software reset can be given *while the GPT is disabled*.

31.4.2.1 Input Capture

There are two Input Capture Channels, and each Input Capture Channel has a dedicated capture pin, capture register and input edge detection/selection logic. Each input capture function has an associated status flag, and can cause the processor to make an interrupt service request.

When a selected edge transition occurs on an Input Capture pin, the contents of the GPT_CNT is captured on the corresponding capture register and the appropriate interrupt status flag is set. An interrupt request can be generated when the transition is detected *if* its corresponding enable bit is set (in the Interrupt Register). The capture can be programmed to occur on the input pin's rising edge, falling edge, on both rising and falling edges, or the capture can be disabled. The events are synchronized with the clock that was selected to run the counter. Only those transitions that occur at least one clock cycle (clock selected to run the counter) *after* the previous recorded transition will be guaranteed to trigger a capture event. There can be up to one clock cycle of uncertainty in the latching of the input transition. The Input Capture registers can be read *at any time* without affecting their values.

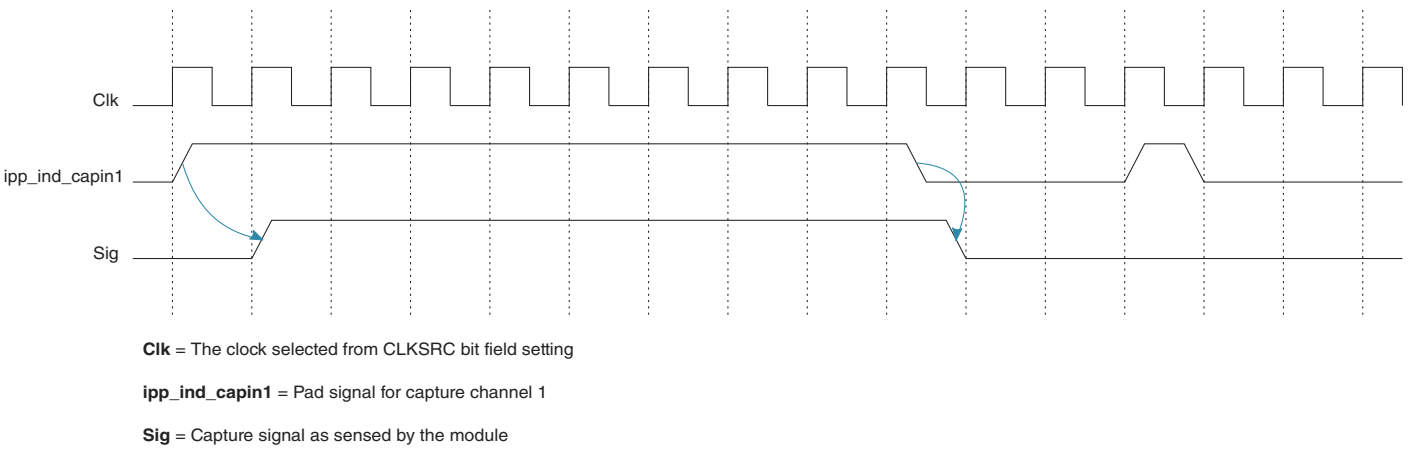


Figure 31-4. Input Capture Event Timing

31.4.2.2 Output Compare

The three Output Compare Channels *use the same counter* (GPT_CNT) as the Input Capture Channels. When the programmed content of an Output Compare register matches the value in GPT_CNT, an output compare status flag is set and an interrupt is generated (if the corresponding bit is set in the interrupt register). Consequently, the Output Compare timer pin will be set, cleared, toggled, not affected at all or provide an active-low pulse for one input clock period (subject to the restriction on the maximum frequency allowed on the pad) according to the mode bits (that were programmed).

There is also a "forced-compare" feature that allows the software to generate a compare event when required, *without the condition of the counter value that is equal to the compare value*. The action taken as a result of a forced compare is the same as when an output compare match occurs, *except that the status flags are not set and no interrupt can be generated*. Forced channels take programmed action immediately after the write to the force-compare bits. These bits are self-negating and always read as zeros.

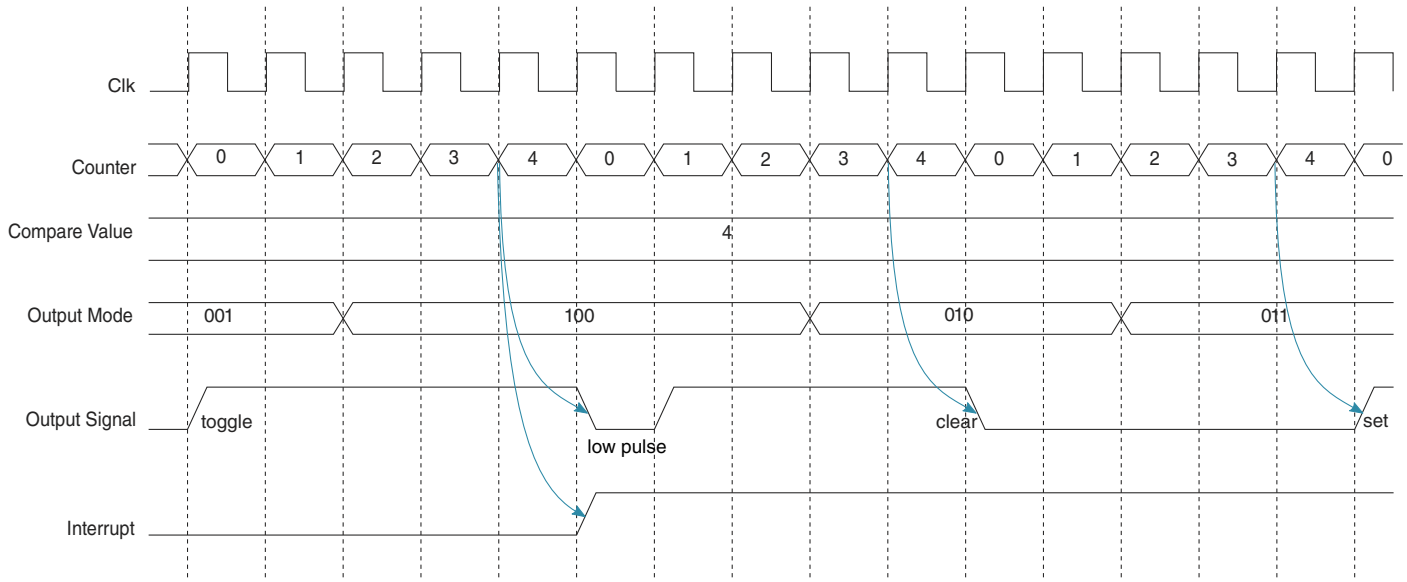


Figure 31-5. Output Compare and Interrupt Timing

31.4.2.3 Interrupts

There are 6 different interrupts that are generated by the GPT. If the selected clock for running the counter is available, then *all interrupts can be generated in Low Power and Debug modes*.

- Rollover Interrupt

  The Rollover Interrupt is generated when the GPT counter reaches 0xffffffff, then resets to 0x00000000 and continues counting. The Rollover Interrupt is enabled by the ROVIE bit in the GPT_IR register; the associated status bit is the ROV bit in the GPT_SR register.

- Input Capture Interrupt 1, 2

  After a capture event occurs, the associated Input Capture Channel generates an interrupt. The "capture event" interrupts are enabled by the IF2IE and IF1IE bits (in the GPT_IR register); the associated status bits are IF2 and IF1 (in the GPT_SR register). The capture of the counter value because of a capture event is *not affected by a pending capture interrupt*. The Capture register is updated with a new counter value when a capture event occurs, regardless of whether that Capture Channels' interrupt has been serviced or not.

- Output Compare Interrupt 1, 2, 3

After a compare event occurs, the associated Output Compare Channel generates an interrupt. The "compare event" interrupts are enabled by the OF3IE, OF2IE, and OF1IE bits (in the GPT_IR register); the associated status bits are OF3, OF2, and OF1 (in the GPT_SR register). A "forced compare" does not generate an interrupt.

A *cumulative* interrupt line is also present, which is asserted whenever any of the above interrupts are posted. The cumulative interrupt line has *no* associated enables or status bits.

31.4.2.4 Low Power Mode Behavior

In Low Power modes, if the clock from the selected clock source is available (except for the External Clock (GPT_CLK), which can be used *only if* the Peripheral Clock (ipg_clk) is available), the counter will continue to run depending on whether the control bit for that mode is set. If the clock is not present or if the corresponding low power bit in the GPT_CR control register is 0, the Main Counter and the Prescaler Counter freeze at their current values and resume counting (from their frozen values) when the Low Power mode is exited.

31.4.2.5 Debug Mode Behavior

In Debug mode, the modules in the device have the option of continuing to run or be halted.

- If the DBGEN bit is set, then the GPT timer will continue to run in Debug mode.
- If the DBGEN bit is not set (in the GPT_CR control register), then the GPT timer is halted.

31.5 Initialization/ Application Information

31.5.1 Selecting the Clock Source

The CLKSRC field in the GPT_CR register selects the clock source. The CLKSRC field value should be changed only after disabling the GPT (EN=0).

The software sequence to be followed while changing clock source is:

1. Disable GPT by setting EN=0 in GPT_CR register.
2. Disable GPT interrupt register (GPT_IR).

3. Configure Output Mode to unconnected/ disconnected—Write zeros in OM3, OM2, and OM1 in GPT_CR
4. Disable Input Capture Modes—Write zeros in IM1 and IM2 in GPT_CR
5. Change clock source CLKSRC to the desired value in GPT_CR register.
6. Assert the SWR bit in GPT_CR register.
7. Clear GPT status register (GPT_SR) (i.e., w1c).
8. Set ENMOD=1 in GPT_CR register, to bring GPT counter to 0x00000000.
9. Enable GPT (EN=1) in GPT_CR register.
10. Enable GPT interrupt register (GPT_IR).

31.6 GPT Memory Map/Register Definition

The GPT has 10 user-accessible 32-bit registers, which are used to configure, operate, and monitor the state of the GPT.

An IP bus write access to the GPT Control Register (GPT_CR) and the GPT Output Compare Register1 (GPT_OCR1) results in *one cycle of wait state*, while other valid IP bus accesses incur 0 wait states.

Irrespective of the Response Select signal value, a Write access to the GPT Status Registers (Read-only registers GPT_ICR1, GPT_ICR2, GPT_CNT) will generate a bus exception.

- If the Response Select signal is driven Low, then the Read/Write access to the *unimplemented* address space of GPT (*ips_addr* is greater than or equal to $BASE + $028) will generate a bus exception.
- If the Response Select is driven High, then the Read/Write access to the unimplemented address space of GPT will *not* generate any error response (like a bus exception).

GPT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
209_8000	GPT Control Register (GPT_CR)	32	R/W	0000_0000h	31.6.1/1607
209_8004	GPT Prescaler Register (GPT_PR)	32	R/W	0000_0000h	31.6.2/1611
209_8008	GPT Status Register (GPT_SR)	32	R/W	0000_0000h	31.6.3/1612
209_800C	GPT Interrupt Register (GPT_IR)	32	R/W	0000_0000h	31.6.4/1613
209_8010	GPT Output Compare Register 1 (GPT_OCR1)	32	R/W	FFFF_FFFFh	31.6.5/1614
209_8014	GPT Output Compare Register 2 (GPT_OCR2)	32	R/W	FFFF_FFFFh	31.6.6/1615

Table continues on the next page...

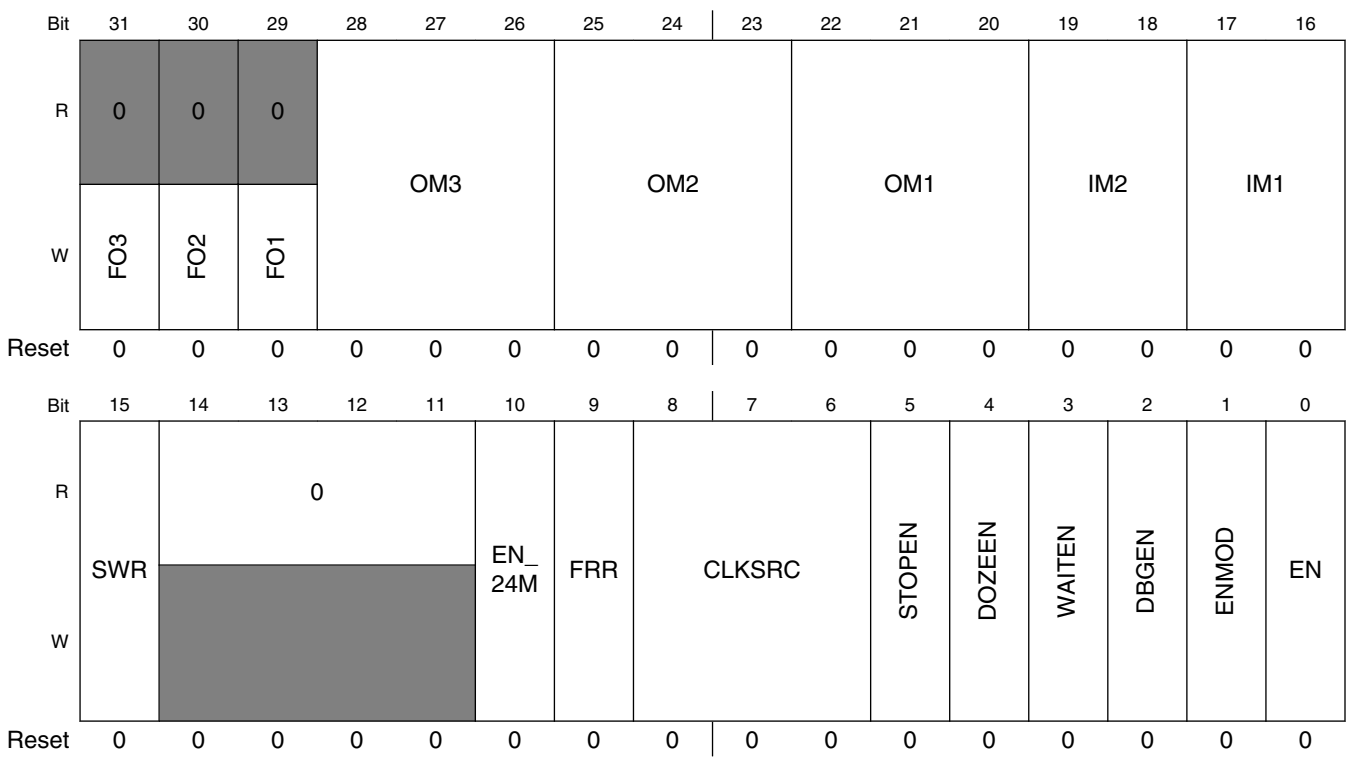
GPT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
209_8018	GPT Output Compare Register 3 (GPT_OCR3)	32	R/W	FFFF_FFFFh	31.6.7/1615
209_801C	GPT Input Capture Register 1 (GPT_ICR1)	32	R	0000_0000h	31.6.8/1616
209_8020	GPT Input Capture Register 2 (GPT_ICR2)	32	R	0000_0000h	31.6.9/1616
209_8024	GPT Counter Register (GPT_CNT)	32	R	0000_0000h	31.6.10/ 1617

31.6.1 GPT Control Register (GPT_CR)

The GPT Control Register (GPT_CR) is used to program and configure GPT operations. An IP Bus Write to the GPT Control Register occurs after one cycle of wait state, while an IP Bus Read occurs after 0 wait states.

Address: 209_8000h base + 0h offset = 209_8000h



GPT_CR field descriptions

Field	Description
31 FO3	FO3 Force Output Compare Channel 3 FO2 Force Output Compare Channel 2

Table continues on the next page...

GPT_CR field descriptions (continued)

Field	Description
	FO1 Force Output Compare Channel 1 The FO*n* bit causes the pin action *programmed* for the timer Output Compare *n* pin (according to the OM*n* bits in this register). • The OF*n* flag (OF3, OF2, OF1) in the status register is **not affected**. • This bit is self-negating and always read as zero. 0 Writing a 0 has no effect. 1 Causes the programmed pin action on the timer Output Compare *n* pin; the OF*n* flag is not set.
30 FO2	See F03
29 FO1	See F03
28–26 OM3	OM3 (bits 28-26) controls the Output Compare Channel 3 operating mode. OM2 (bits 25-23) controls the Output Compare Channel 2 operating mode. OM1 (bits 22-20) controls the Output Compare Channel 1 operating mode. The OM*n* bits specify the response that a compare event will generate on the output pin of Output Compare Channel *n*. • The toggle, clear, and set options cause a change on the output pin *only* if a compare event occurs. • When OM*n* is programmed as 1xx (active low pulse), the output pin is set to one immediately on the next input clock; a low pulse (that is an input clock in width) occurs when there is a compare event. Note that here, "input clock" refers to the clock selected by the CLKSRC bits of the GPT Control Register. 000 Output disconnected. No response on pin. 001 Toggle output pin 010 Clear output pin 011 Set output pin 1xx Generate an active low pulse (that is one input clock wide) on the output pin.
25–23 OM2	See OM3
22–20 OM1	See OM3
19–18 IM2	IM2 (bits 19-18, Input Capture Channel 2 operating mode) IM1 (bits 17-16, Input Capture Channel 1 operating mode) The IM*n* bit field determines the transition on the input pin (for Input capture channel *n*), which will trigger a capture event. 00 capture disabled 01 capture on rising edge only 10 capture on falling edge only 11 capture on both edges
17–16 IM1	See IM2
15 SWR	Software reset. This is the software reset of the GPT module. It is a self-clearing bit. • The SWR bit is set when the module is in reset state.

Table continues on the next page...

GPT_CR field descriptions (continued)

Field	Description
	• The SWR bit is cleared when the reset procedure finishes. • Setting the SWR bit resets **all of the registers** to their default reset values, except for the EN, ENMOD, STOPEN, DOZEEN, WAITEN, and DBGEN bits in the GPT Control Register (this control register). 0 GPT is not in reset state 1 GPT is in reset state
14–11 Reserved	This read-only field is reserved and always has the value 0.
10 EN_24M	Enable 24 MHz clock input from crystal. • A hardware reset resets the EN_24M bit. • A software reset *does not affect* the EN_24M bit. 0 24M clock disabled 1 24M clock enabled
9 FRR	Free-Run or Restart mode. The FFR bit determines the behavior of the GPT when a compare event in channel 1 occurs. • In Restart mode, after a compare event, the counter resets to 0x00000000 and resumes counting (after the occurrence of a compare event). • In Free-Run mode, after a compare event, the counter continues counting until 0xFFFFFFFF and then rolls over to 0. 0 Restart mode 1 Free-Run mode
8–6 CLKSRC	Clock Source select. The CLKSRC bits select which clock will go to the prescaler (and subsequently be used to run the GPT counter). • The CLKSRC bit field value should only be changed after disabling the GPT by clearing the EN bit in this register (GPT_CR). • A software reset does not affect the CLKSRC bit. 000 No clock 001 Peripheral Clock (ipg_clk) 010 High Frequency Reference Clock (ipg_clk_highfreq) 011 External Clock 100 Low Frequency Reference Clock (ipg_clk_32k) 101 Crystal oscillator as Reference Clock (ipg_clk_24M) others Reserved
5 STOPEN	GPT Stop Mode enable. The STOPEN read/write control bit enables GPT operation *during Stop mode*. • A hardware reset resets the STOPEN bit. • A software reset *does not affect* the STOPEN bit. 0 GPT is disabled in Stop mode. 1 GPT is enabled in Stop mode.
4 DOZEEN	GPT Doze Mode Enable. • A hardware reset resets the DOZEEN bit. • A software reset *does not affect* the DOZEEN bit.

Table continues on the next page...

GPT_CR field descriptions (continued)

Field	Description
	0 GPT is disabled in doze mode. 1 GPT is enabled in doze mode.
3 WAITEN	GPT Wait Mode enable. The WAITEN read/write control bit enables GPT operation *during Wait mode*. • A hardware reset resets the WAITEN bit. • A software reset *does not affect* the WAITEN bit. 0 GPT is disabled in wait mode. 1 GPT is enabled in wait mode.
2 DBGEN	GPT debug mode enable. The DBGEN read/write control bit enables GPT operation *during Debug mode*. • A hardware reset resets the DBGEN bit. • A software reset *does not affect* the DBGEN bit. 0 GPT is disabled in debug mode. 1 GPT is enabled in debug mode.
1 ENMOD	GPT Enable mode. When the GPT is disabled (EN=0), then both the Main Counter and Prescaler Counter *freeze their current count values*. The ENMOD bit determines the value of the GPT counter when Counter is enabled again (if the EN bit is set). • If the ENMOD bit is 1, then the Main Counter and Prescaler Counter values are reset to 0 after GPT is enabled (EN=1). • If the ENMOD bit is 0, then the Main Counter and Prescaler Counter restart counting *from their frozen values* after GPT is enabled (EN=1). • If GPT is programmed to be disabled in a low power mode (STOP/WAIT), then the Main Counter and Prescaler Counter *freeze at their current count values* when the GPT enters low power mode. • When GPT exits low power mode, the Main Counter and Prescaler Counter start counting from their frozen values, regardless of the ENMOD bit value. • Setting the SWR bit will clear the Main Counter and Prescaler Counter values, regardless of the value of EN or ENMOD bits. • A hardware reset resets the ENMOD bit. • A software reset *does not affect* the ENMOD bit. 0 GPT counter will retain its value when it is disabled. 1 GPT counter value is reset to 0 when it is disabled.
0 EN	GPT Enable. The EN bit is the GPT module enable bit. **Before setting the EN bit**, we recommend that *all registers be properly programmed*. • A hardware reset resets the EN bit. • A software reset *does not affect* the EN bit. 0 GPT is disabled. 1 GPT is enabled.

31.6.2 GPT Prescaler Register (GPT_PR)

The GPT Prescaler Register (GPT_PR) contains bits that determine the *divide value* of the clock that runs the counter.

Address: 209_8000h base + 4h offset = 209_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PRESCALER24M				PRESCALER											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

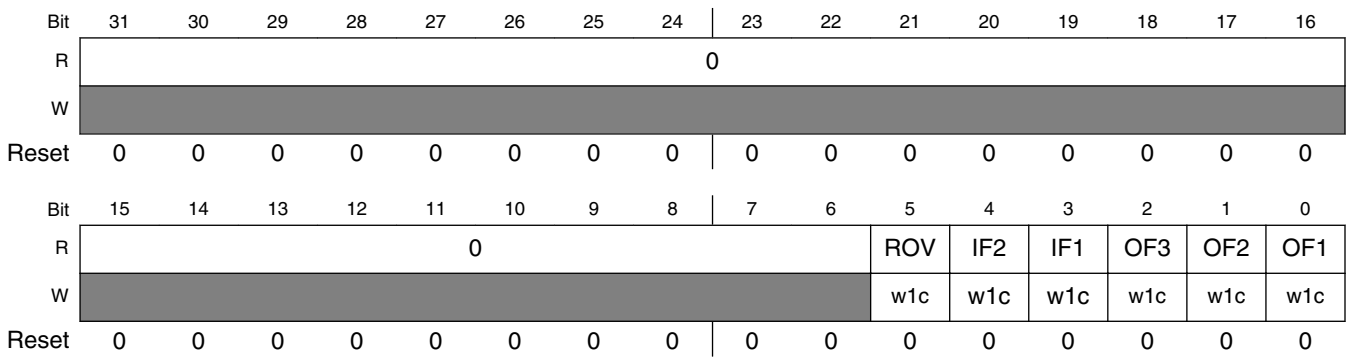
GPT_PR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–12 PRESCALER24M	Prescaler bits. 24M crystal clock is divided by [PRESCALER24M + 1] before selected by the CLKSRC field. If 24M crystal clock is not selected, this feild takes no effect. 0x0 Divide by 1 0x1 Divide by 2 ... ... 0xF Divide by 16
PRESCALER	Prescaler bits. The clock selected by the CLKSRC field is divided by [PRESCALER + 1], and then used to run the counter. • A change in the value of the PRESCALER bits cause the Prescaler counter to reset and a new count period to start immediately. • See Figure 31-3 for the timing diagram. 0x000 Divide by 1 0x001 Divide by 2 ... ... 0xFFF Divide by 4096

31.6.3 GPT Status Register (GPT_SR)

The GPT Status Register (GPT_SR) contains bits that indicate that a counter has rolled over, and if any event has occurred on the Input Capture and Output Compare channels. The bits are cleared by writing a 1 to them.

Address: 209_8000h base + 8h offset = 209_8008h



GPT_SR field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.
5 ROV	Rollover Flag. The ROV bit indicates that the counter has reached its *maximum possible value* and *rolled over* to 0 (from which the counter continues counting). The ROV bit is only set if the counter has reached 0xFFFFFFFF in both Restart and Free-Run modes. 0 Rollover has not occurred. 1 Rollover has occurred.
4 IF2	IF2 Input capture 2 Flag IF1 Input capture 1 Flag The IF*n* bit indicates that a capture event has occurred on Input Capture channel *n*. 0 Capture event has not occurred. 1 Capture event has occurred.
3 IF1	See IF2
2 OF3	OF3 Output Compare 3 Flag OF2 Output Compare 2 Flag OF1 Output Compare 1 Flag The OF*n* bit indicates that a compare event has occurred on Output Compare channel *n*.

Table continues on the next page...

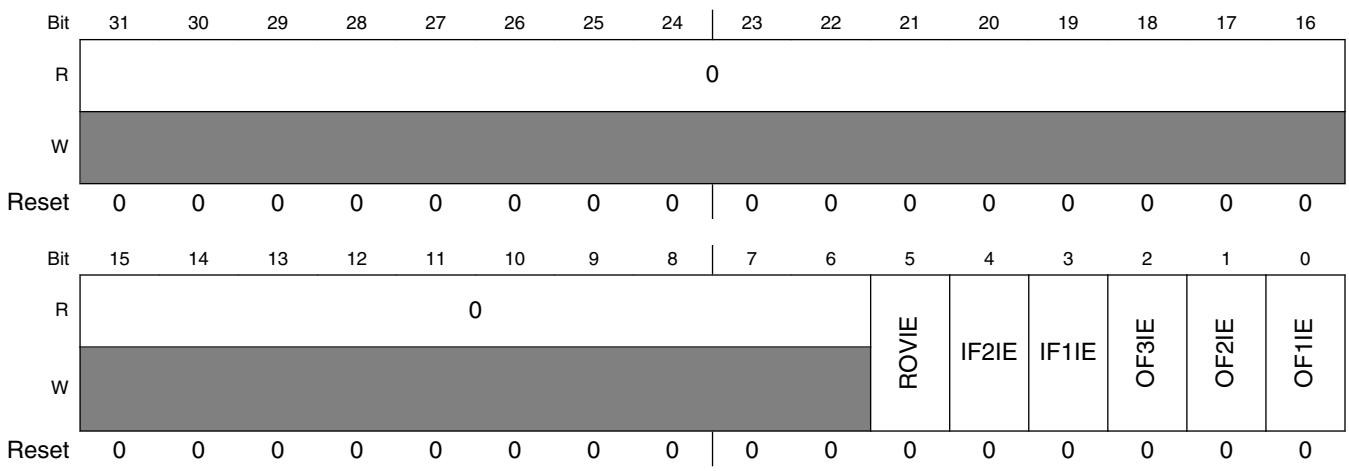
GPT_SR field descriptions (continued)

Field	Description
	0 Compare event has not occurred. 1 Compare event has occurred.
1 OF2	See OF3
0 OF1	See OF3

31.6.4 GPT Interrupt Register (GPT_IR)

The GPT Interrupt Register (GPT_IR) contains bits that control whether interrupts are generated after rollover, input capture and output compare events.

Address: 209_8000h base + Ch offset = 209_800Ch



GPT_IR field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.
5 ROVIE	Rollover Interrupt Enable. The ROVIE bit controls the Rollover interrupt. 0 Rollover interrupt is disabled. 1 Rollover interrupt enabled.
4 IF2IE	IF2IE Input capture 2 Interrupt Enable IF1IE Input capture 1 Interrupt Enable The IF*n*IE bit controls the IF*n*IE Input Capture *n* Interrupt Enable.

Table continues on the next page...

GPT_IR field descriptions (continued)

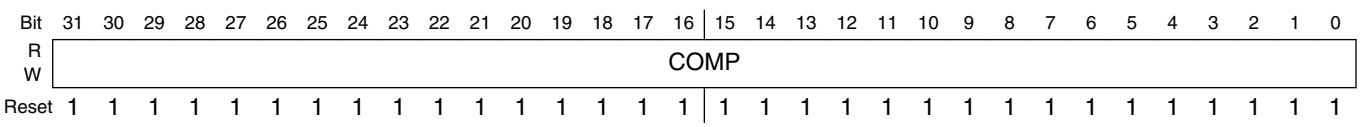
Field	Description
	0 IF2IE Input Capture *n* Interrupt Enable is disabled. 1 IF2IE Input Capture *n* Interrupt Enable is enabled.
3 IF1IE	See IF2IE
2 OF3IE	OF3IE Output Compare 3 Interrupt Enable OF2IE Output Compare 2 Interrupt Enable OF1IE Output Compare 1 Interrupt Enable The OF*n*IE bit controls the Output Compare Channel *n* interrupt. 0 Output Compare Channel *n* interrupt is disabled. 1 Output Compare Channel *n* interrupt is enabled.
1 OF2IE	See OF3IE
0 OF1IE	See OF3IE

31.6.5 GPT Output Compare Register 1 (GPT_OCR1)

The GPT Compare Register 1 (GPT_OCR1) holds the value that determines when a compare event will be generated on Output Compare Channel 1. Any write access to the Compare register of Channel 1 while in Restart mode (FRR=0) will reset the GPT counter.

An IP Bus Write access to the GPT Output Compare Register1 (GPT_OCR1) occurs *after* one cycle of wait state; an IP Bus Read access occurs *immediately* (0 wait states).

Address: 209_8000h base + 10h offset = 209_8010h



GPT_OCR1 field descriptions

Field	Description
COMP	Compare Value. When the counter value equals the COMP bit field value, a compare event is generated on Output Compare Channel 1.

31.6.6 GPT Output Compare Register 2 (GPT_OCR2)

The GPT Compare Register 2 (GPT_OCR2) holds the value that determines when a compare event will be generated on Output Compare Channel 2.

Address: 209_8000h base + 14h offset = 209_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMP																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

GPT_OCR2 field descriptions

Field	Description
COMP	Compare Value. When the counter value equals the COMP bit field value, a compare event is generated on Output Compare Channel 2.

31.6.7 GPT Output Compare Register 3 (GPT_OCR3)

The GPT Compare Register 3 (GPT_OCR3) holds the value that determines when a compare event will be generated on Output Compare Channel 3.

Address: 209_8000h base + 18h offset = 209_8018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMP																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

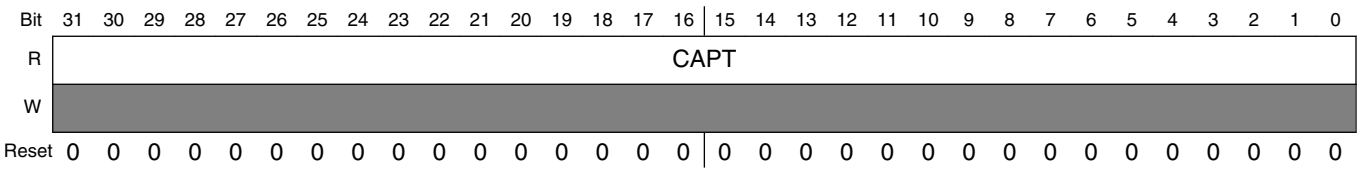
GPT_OCR3 field descriptions

Field	Description
COMP	Compare Value. When the counter value equals the COMP bit field value, a compare event is generated on Output Compare Channel 3.

31.6.8 GPT Input Capture Register 1 (GPT_ICR1)

The GPT Input Capture Register 1 (GPT_ICR1) is a read-only register that holds the value *that was in the counter during the last capture event* on Input Capture Channel 1.

Address: 209_8000h base + 1Ch offset = 209_801Ch



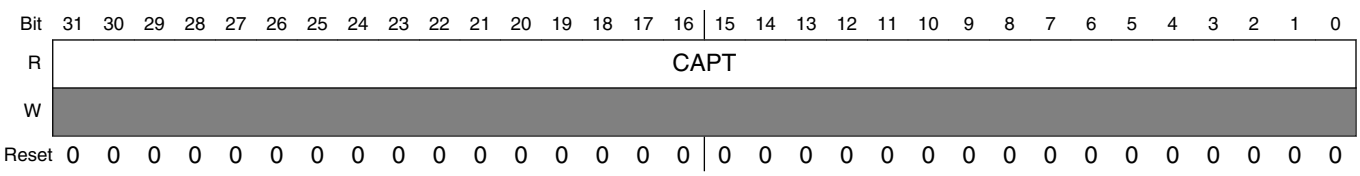
GPT_ICR1 field descriptions

Field	Description
CAPT	Capture Value. After a capture event on Input Capture Channel 1 occurs, the current value of the counter is loaded into GPT Input Capture Register 1.

31.6.9 GPT Input Capture Register 2 (GPT_ICR2)

The GPT Input capture Register 2 (GPT_ICR2) is a read-only register which holds the value that was in the counter during the last capture event on input capture channel 2.

Address: 209_8000h base + 20h offset = 209_8020h



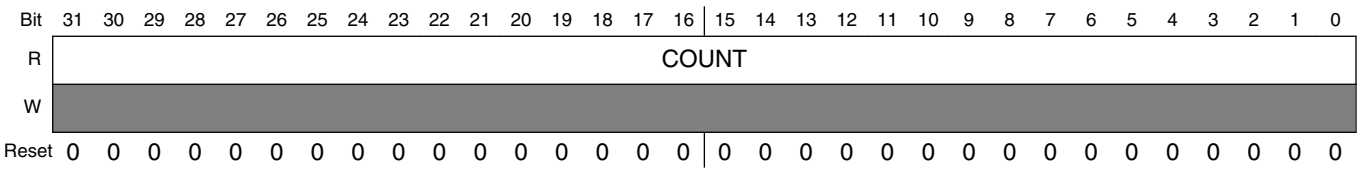
GPT_ICR2 field descriptions

Field	Description
CAPT	Capture Value. After a capture event on Input Capture Channel 2 occurs, the current value of the counter is loaded into GPT Input Capture Register 2.

31.6.10 GPT Counter Register (GPT_CNT)

The GPT Counter Register (GPT_CNT) is the main counter's register. GPT_CNT is a read-only register and can be read *without affecting the counting process* of the GPT.

Address: 209_8000h base + 24h offset = 209_8024h



GPT_CNT field descriptions

Field	Description
COUNT	Counter Value. The COUNT bits show the current count value of the GPT counter.

Chapter 32
2D Graphics Processing Unit (GPU2D)

32.1 Overview

The 2D Graphics Processing Unit (GPU2D) is a high-performance multi-pipe 2D graphics core that accelerates the 2D graphics display on a variety of devices. The GPU2D supports a wide-range of addressable screen sizes and resolutions.

The R2D GPU Hardware acceleration is brought to numerous 2D applications including graphical user interfaces (GUI), menu displays, flash animation, and gaming.

32.2 GPU2D Block Diagram

32.2.1 R2D GPU

The R2D Graphics Processing Unit (GPU) defines a high-performance 2D raster graphics core that accelerates the 2D graphics display.

R2D GPU supports acceleration of the following graphics APIs:

- DirectFB (Linux / Linux Embedded)
- GDI / DirectDraw (Windows Embedded Compact 7 / Embedded CE 6)
- Android

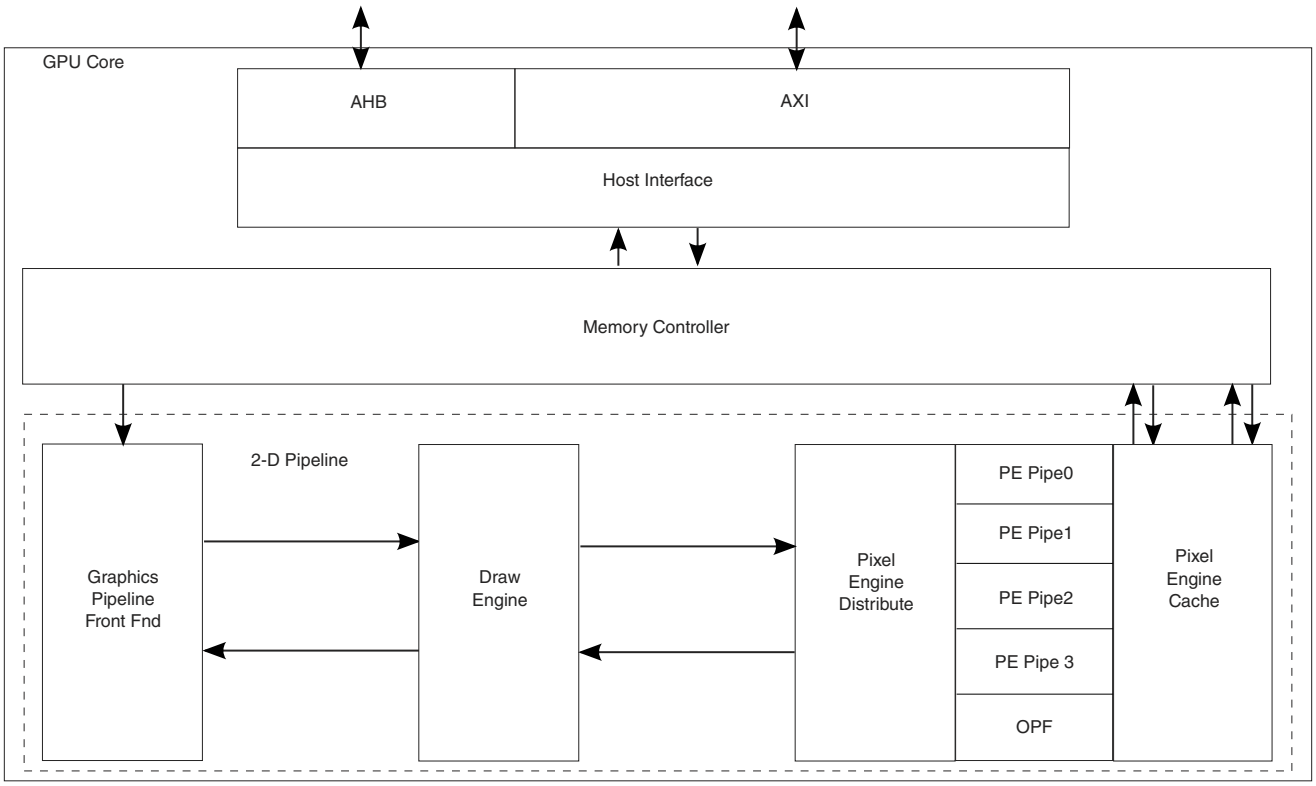


Figure 32-1. R2D GPU Block Diagram

32.3 GPU2D Features

The following sections describe the functional features of the R2D GPU.

32.3.1 Full Featured R2D GPU Pipeline

- Bit BLT
- Stretch BLT
- Rectangle fill and clear
- Line drawing
- Filter BLT
- Mono expansion for text rendering
- ROP2, ROP3, and ROP4
- Alpha blending, including Java 2 Porter-Duff compositing blending rules
- 32K x 32K coordinate system
- 90 / 180 / 270 degree rotation
- Transparency by monochrome mask, chroma key, or pattern mask

32.4 GPU2D OPERATIONS

32.4.1 R2D GPU Operations

Information detailing the R2D GPU operations can be found in this section.

32.4.1.1 Line

The LINE operation draws a line. Coordinates for two points are given: start point and end point. The end point is not drawn. Lines are rendered using the Bresenham algorithm. The Bresenham algorithm has the advantage of using integer arithmetic and has no accumulation of rounding errors.

In the case of line, only ROP2 and ROP4 are supported. It operates on pattern and destination. The pattern should have a transparency mask in order to use ROP4.

Clipping is supported for lines on a per pixel basis.

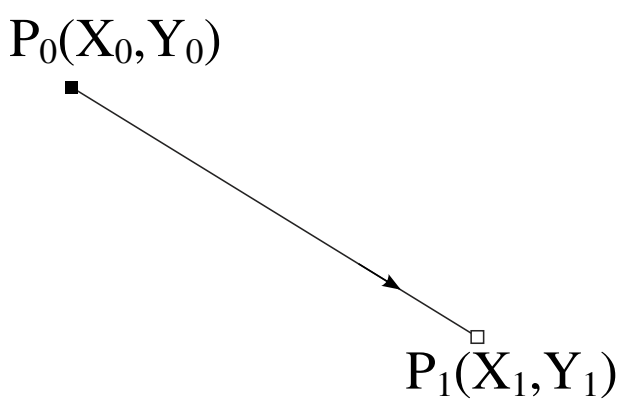


Figure 32-2. Line

32.4.1.2 Rectangle Fill and Clear

Rectangle fill creates a rectangle area with a given color or a pattern fill. Essentially rectangle fill is a pattern fill, where an 8x8 pattern is initialized with the specified color. It supports ROP2 and ROP4 with the pattern and destination as its inputs. If ROP4 is used, the pattern should have a transparency mask.

Clear is similar to Rectangle Fill except that it does not use a pattern. A 32-bit clear value with 4-bit byte mask is used to fill the entire rectangle area.

Both Rectangle Fill and Clear support clipping, which is performed on a per primitive basis.

32.4.1.3 BitBLT

Bit blit (BitBLT) transfers data from one area of a memory (source) to another area of the memory (destination).

The source and destination can be from the same or different memory locations. Both source and destination must be described by a rectangular area. The source and destination rectangles can be of the same size (most bit blits are of this nature) or of different sizes, in which case, the operation becomes a stretch or shrink blit.

BitBLT supports ROP2, ROP3, and ROP4 which includes source, destination and pattern, and an optional transparency color. Clipping can be performed on a primitive basis.

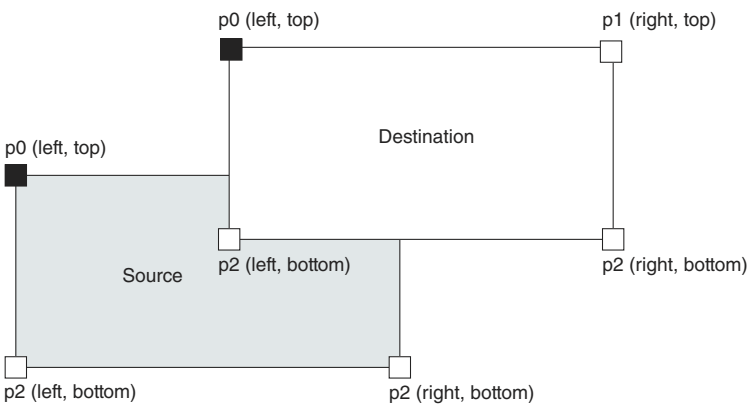


Figure 32-3. BitBLT

The graphics engine supports the following source and destination formats for data bit blits and filter blits. In addition to these source and destination RGB formats, their swizzle formats (ARGB, RGBA, ABGR, BGRA) are also supported.

Table 32-1. BitBLT Formats

Formats	Source Image	Destination Image
A1R5G5B5	Yes	Yes
A4R4G4B4	Yes	Yes
X1R5G5B5	Yes	Yes

Table continues on the next page...

Table 32-1. BitBLT Formats (continued)

Formats	Source Image	Destination Image
X4R4G4B4	Yes	Yes
R5G6B5	Yes	Yes
A8R8G8B8	Yes	Yes
X8R8G8B8	Yes	Yes
A8	Yes	Yes
1-bit monochrome	Yes	No
8-bit color index	No	No

32.4.1.4 Stretch BLT

The stretch blit (Stretch BLT) primitive performs a bit blt operation with stretch or shrink. The modified Bresenham algorithm is used to generate corresponding coordinates for fast stretching. The stretch factor is specified in a 15.0 fixed-point format. Stretch blit is not allowed to overlap, therefore no part of source and destination can share any piece of memory. Non-stretch blits can overlap. For stretch blit, clipping is performed on a per pixel basis.

32.4.1.5 Monochrome Expansion and Mask BLT

Monochrome expansion and mask blit are different operations, although both use the bit stream from command buffer and both can be the source for ROP4 source selection. This means that each output pixel can be a combination of source, pattern, monochrome mask (for masked blits) and destination.

32.4.1.5.1 Monochrome expansion

For monochrome expansion, the bit from the stream is used to switch on/off a solid color that is defined in a register. This mechanism enables the use of just one bit per pixel to represent colors. In effect, the MONO EXPANSION primitive increases color representation from one bit per pixel to multiple bits per pixel. A typical application for mono color is font drawing.

Monochrome expansion does not support overlapping of the source and destination. It is the responsibility of the driver to make sure that the command will never be executed with overlapping source and destination.

32.4.1.5.2 Mask BLT

For Mask BLT, the bit from the stream is used to toggle on/off a color in the source frame buffer. Mask BLT takes its color source from memory and its monochrome mask from the command stream. Clipping is supported and is performed on a per pixel basis.

32.4.1.6 Filter BLT

Filter blit performs high quality scaling, up or down, using an FIR re-sampling filter with up to 9 taps. The sub-pixel coordinates (locations between the pixel grids) are generated by the drawing engine. The filter block in the drawing engine uses the sub-pixel information to select the appropriate filter kernel. R2D GPU processes 1 pixel every cycle when performing filter blit.

A stretch- or shrink-factor of 15.16 fixed-point format is supported. To generate a single destination pixel requires 9 source pixels. An image is scaled in two passes, one for X-dimension (HOR_FILTER_BLT) and the other for Y-dimension (VER_FILTER_BLT). The software sets up the filter kernel/coefficient table and the kernel size, as well as a temporary buffer for storing intermediate results. After the first pass is completed, intermediate results are sent back to memory, and then the second pass starts to scale the first-pass image. Because of this two-step procedure, the throughput of FILTER BLT is lower than that of STRETCH BLT. Also the Filter Kernel Table may need to be reloaded, and some cycles are consumed in calculating the stepping parameters.

When the stretch or shrink factor is 1, the filterBlit works as a bitBlit copy. It can be used as format converter in that case, for instance, YUV to RGB converter. To use as a format converter, only one pass (HOR_FILTER_BLT or VER_FILTER_BLT) is needed. To optimize the memory bandwidth, when using filterBlit to do YUV to RGB filtering, the temporary target buffer format can be specified as YUY2 to process Y-dimension filtering (VER_FILTER_BLT). This is to avoid converting YUV to A8R8G8B8 in the 1st vertical pass to reduce the memory bandwidth and increase the pixel processing rate. This is the only special case that GPU may use YUY2 as target format.

The Filter BLT primitive supports the following source and destination image formats:

Table 32-2. Filter BLT Formats

Formats	Source Image	Destination Image
A1R5G5B5	Yes	Yes
A4R4G4B4	Yes	Yes
A8R8G8B8	Yes	Yes
R5G6B5	Yes	Yes
X1R5G5B5	Yes	Yes

Table continues on the next page...

Table 32-2. Filter BLT Formats (continued)

Formats		Source Image	Destination Image
X4R4G4B4		Yes	Yes
X8R8G8B8		Yes	Yes
YUV	NV12 (4:2:0, 2 planes)	Yes	No
	NV16 (4:2:2, 2 planes)	Yes	No
	UYVY (4:2:2, interleave)	Yes	No
	YUY2 (4:2:2, interleave)	Yes	No
	YV12 (4:2:0, 3 planes)	Yes	No
	8-bit color index	Yes	No
	A8	Yes	Yes

32.4.1.7 R2D Performance of different operations

Table 32-3. Performance of different operations without rotation

Primitive	Peak Performance	Source/destination overlap	Clipping
Line	1 pixel / cycle	N/A	Done on pixel basis
Rectangle	2.4 pixel / cycle	N/A	Done on primitive basis
Clear	2.4 pixel / cycle	N/A	Done on primitive basis
Blit	2.4 pixel / cycle	Overlap is allowed	Done on primitive basis
Stretch blit	1 pixel / cycle	No overlap allowed	Done on pixel basis
Monochrome expansion	1.6 pixel / cycle	No overlap allowed	Done on pixel basis
Filter blit	1 pixel / cycle	N/A	N/A

Table 32-4. Performance of different operations with rotation

Primitive	Performance	Source/destination overlap	Clipping
Line	1 pixel / cycle	N/A	Done on pixel basis
Rectangle	2 pixel / cycle	N/A	Done on primitive basis
Clear	2 pixel / cycle	N/A	Done on primitive basis
Blit	2 pixel / cycle	Overlap is allowed	Done on primitive basis
Stretch blit	1 pixel / cycle	No overlap allowed	Done on pixel basis
Monochrome expansion	1 pixel / cycle	No overlap allowed	Done on pixel basis
Filter blit	1 pixel / cycle	N/A	N/A

NOTE

The performance decreases for filter blit operations which require two passes.

32.4.1.8 Rotation

- 90° / 180° / 270° / X-Flip / Y-Flip/ Mirror rotation is supported for all primitives.

32.4.1.9 Transparency Mode

For monochrome expansion:

- Opaque
- Conditional transparency: Transparent if the current pixel matches the specified value.

For blits:

- Opaque
- Masked transparency: Transparent if the mask for the current pixel or pattern is zero.
- Source Conditional transparency: Transparent if the source pixel is within the specified value range.
- Destination Conditional transparency: Transparent if the destination pixel is not within the specified value range.

32.4.1.10 Clipping

One clipping rectangle is supported for all bit blit primitives.

32.4.1.11 R2D GPU Data Formats

The graphics engine supports the following source and destination formats for data bit blits and filter blits. In addition to these source and destination RGB formats, their swizzle formats (ARGB, RGBA, ABGR, BGRA) are also supported. For YUV formats, the GPU supports their U/V swap formats.

Table 32-5. Data Formats

Format	Bit Blit Input	Bit Blit Output	Stretch Blit Input	Stretch Blit Dst	Filter Blit Input	Filter Blit Dst	OPF Input	OPF Dst
A1R5G5B5	Y	Y	Y	Y	Y	Y	Y	Y
A4R4G4B4	Y	Y	Y	Y	Y	Y	Y	Y
A8R8G8B8	Y	Y	Y	Y	Y	Y	Y	Y
X1R5G5B5	Y	Y	Y	Y	Y	Y	Y	Y

Table continues on the next page...

Table 32-5. Data Formats (continued)

Format	Bit Blit Input	Bit Blit Output	Stretch Blit Input	Stretch Blit Dst	Filter Blit Input	Filter Blit Dst	OPF Input	OPF Dst
X4R4G4B4	Y	Y	Y	Y	Y	Y	Y	Y
X8R8G8B8	Y	Y	Y	Y	Y	Y	Y	Y
R5G6B5	Y	Y	Y	Y	Y	Y	Y	Y
A8	Y	Y	Y	Y	Y	Y	Y	Y
YUY2 (4:2:2)	Y	N	Y	N	Y	N	Y	N
UYVY (4:2:2)	Y	N	Y	N	Y	N	Y	N
YV12 (4:2:0)	Y	N	Y	N	Y	N	Y	N
NV12	Y	N	Y	N	Y	N	Y	N
NV16	Y	N	Y	N	Y	N	Y	N
8-bit color index	Y	N	Y	N	Y	N	N	N
1-bit monochrome	Y	N	N	N	N	N	N	N

32.4.1.12 ARGB Data Conversion of R2D GPU

The pixels read from source or destination will be expanded into A8R8G8B8 format to maintain lossless pixel operations. The resulting pixels will be converted into the destination format.

32.4.1.13 YUV to RGB Conversion of R2D GPU

YUV data can be converted into 8-bit per component RGB format at the output of the cache only. Once converted, there is no way back to YUV format. GPU supports BT.601 and BT.709 YUV to RGB color conversion standards.

In BT.601, the YUV to RGB conversion is done using the following approximation:

16 <= Y <= 235
16 <= U <= 240
16 <= V <= 240

A = Y - 16
B = U - 128
C = V - 128

R = clip((298*A + 410*C + 128) >> 8)
G = clip((298*A - 101*B - 209*C + 128) >> 8)
B = clip((298*A + 519*B + 128) >> 8)

The Y, U and V components are clamped prior to the conversion.

Y is clamped between 16 and 235, inclusively.
U and V are clamped between 16 and 240, inclusively.

In BT.709, the R, G, B equations are slightly changed to
R = clip((298*A + 461*C + 128) >> 8)
G = clip((298*A - 55*B - 137*C + 128) >> 8)
B = clip((298*A + 543*B + 128) >> 8)

32.4.1.14 Color Index Input Conversion Support of R2D GPU

Color index is supported for source data only. A look-up table with 256 entries is provided for indexing the data. The table is fully programmable. The conversion is done when pixels are read out of the cache.

32.4.1.15 Source/Destination Pre-multiply and De-Multiply Support

GPU supports source pre-multiply source alpha or global alpha, or source global color for global colorizing. On destination, the GPU supports destination pre-multiply destination alpha, destination de-multiply alpha.

32.4.1.16 Alpha Blending

The GPU supports alpha blending together with ROP. The alpha blending function is performed on ROP function result source.

The general alpha blending equations are:
Cd = Fs * Cs' + Fd * Cd'
Ad = Fs * As'' + Fd * Ad''

Where

- Cs' is the source color component (adjusted for NPM if necessary)
- Cd' is the destination color component (adjusted for NPM if necessary)
- As'' is the modified source alpha component
- Ad'' is the modified destination alpha component
- Fs is fraction of the source that contributes to the final value
- Fd is fraction of the destination that contributes to the final value

The blending is done in 5 logical stages (not real implementation stages):

1. Transparent/opaque conversion

- In this stage, the incoming alpha (source or destination independently) can be inverted if needed to match the internal alpha rule. Internally, an alpha of 0 means transparent, while an alpha of "0xFF" means opaque. External content might follow the opposite rule. The output of the block is either As (Ad for destination) or 1-As (1-Ad for destination).

2. Global value substitution
   - A global alpha value from a register can be used to substitute or scale the incoming alpha. An incoming alpha As can pass-through, be directly substituted by Ags (global alpha) or scaled by the global alpha value (As * Ags). The source and destination have distinct global alpha values.
3. Blending factor generation
   - At this stage, the blending factors are generated (refer to table below). Each alpha can take the values 0, 1, A or 1-A depending on the blending mode.
4. Final blending
   - This is the final stage which implements blending equations.

The fractions take the values described in the following table, depending on the blending mode.

Table 32-6. Blending Modes Fractions Description

Blending Mode	Fs	Fd
Clear	0	0
SRC	1	0
DST	0	1
SRC_OVER	1	1 - As"
DST_OVER	1 - Ad"	1
SRC_IN	Ad"	0
DST_IN	0	As"
SRC_OUT	1 - Ad"	0
DST_OUT	0	1 - As"
SRC_ATOP	Ad"	1 - As"
DST_ATOP	1 - Ad"	As"
XOR	1 - Ad"	1 - As"

To control the blending modes, the following register fields are used:

- 1 bit for transparent/opaque conversion for source alpha

- 1 bit for transparent/opaque conversion for destination alpha
- 2 bits for source alpha modifications, to specify the 3 cases (As, Ags, As*Ags)
- 2 bits for destination alpha modifications, to specify the 3 cases (Ad, Agd, Ad*Agd)
- 4 bits to select between the 12 blending modes
- 8 bits for global source alpha
- 8 bits for global destination alpha

Alpha blending is supported on bit blit and filter blit primitives.

32.4.1.17 GPU Cache Management

The software cache flush is supported to flush the GPU cache to memory. Auto-flush of GPU cache is also supported. The software sets up the auto-flush interval, and hardware will do the cache flush automatically at programmable intervals.

Chapter 33
3D Graphics Processing Unit (GPU3D)

33.1 Overview

The GPU3D is a high-performance core that delivers hardware acceleration for 3D graphics display.

Addressable screen sizes range from the smallest cell phones to HD 1080p displays. It provides high performance, high quality graphics, low power consumption, and the smallest silicon footprint.

GPU3D accelerates numerous 3D graphics applications, including graphical user interfaces (GUI), menu displays, flash animation, and gaming. This module supports the following graphics APIs:

- OpenGL ES 3.0
- OpenGL ES 2.0
- OpenGL ES 1.1
- OpenVG 1.1
- EGL 1.4

33.2 GPU3D Block Diagram

The main functional units of the GPU3D are shown in the figure below and their description is as follows:

- Host Interface: Allows GPU3D to communicate with external memory and the CPU through AXI or AHB bus. In this block data crosses clock domain boundaries.
- Memory Controller: Internal memory management unit that controls the block-to-host memory request interface.
- Graphic Pipeline Front End: Inserts high level primitives and commands into the graphics pipeline.

- Ultra-threaded Unified Shader: SIMD processor that performs as both vertex shader and fragment shader. When used as a vertex shader it performs geometry transformations and lighting computations. When used as a fragment shader, it applies texture data and computes color values for each pixel. GPU3D has four such shaders.
- 3D Rendering Engine: Converts triangles and lines into pixels. Computes slopes of color attributes and texture coordinates. Performs clipping.
- Texture Engine: Retrieves texture information from memory upon request by the fragment shader. Performs interpolation and filtering, and transfers the computed value to the fragment shader or the vertex shader. The texture unit can process 2 pixels or 2 vertices/cycle.
- Pixel Engine/Resolve: Pixel engine does alpha blending and visible surface determination. Resolve does tiling and de-tiling as well as FSAA filtering. The pixel engine can process 2 pixels/cycle.

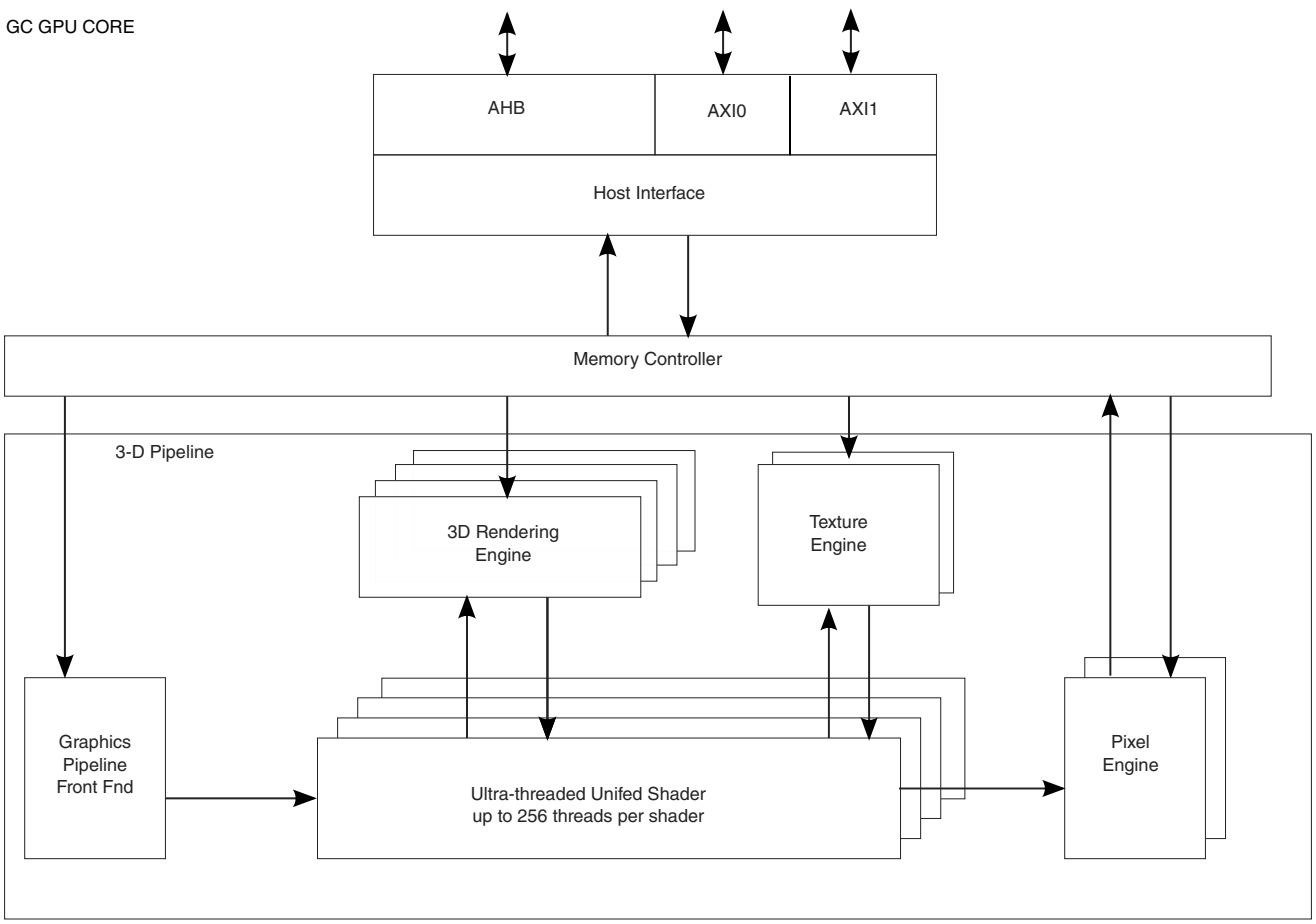


Figure 33-1. GPU3D Top Level Block Diagram

There are two main clocks for the processing control: shader clk for the shader part, and core clk for the remaining part. The core clk and shader clk frequency are programmable in CCM module (as the clk frequency for GPU3D_CORE_CLK_ROOT, GPU3D_SHADER_CLK_ROOT).

The GPU3D provides architectural support for the following features:

Table 33-1. GPU3D Architectural Support

Feature	GPU Support
Primary API	OpenGL ES1.1, 2.0, and 3.0
Additional API's	OpenVG1.1
Other Graphics Support	EGL1.4
Drivers	OpenGLES1.1,2.0 and 3.0
	OpenVG1.1
	EGL1.4
Operating Systems	Windows CE
	Linux Embedded
Z (depth)	Early Z support included
Stencil	Early stencil support included
Shader Languages	GLSL ES 1.0
Shader model compatability	Shader model 3.0
Shader types and execution units	One programmable Scalable Ultra-threaded Unified Shaders (SIMD4:transcendental,ctl-flow,tx-load) one instruction issue per shader per clock; IEEE 32-bit floating-point pipeline supports long shader instructions
FSAA anti-aliasing mechanisms	High quality MSAA 4x MSAA 16x for OpenVG
Code and data memory location restrictions	Unrestricted; arbitrary memory reads and writes
Physical address	32 bits
MMU description	32-bit virtual address, 4 kB pages, error reporting outside of address space
TLB	4 cache lines per requestor
Resource locks with CPU	Semaphore lock
Max memory latency without a performance hit	128 GPU cycles

33.3 GPU3D Hardware Features

The GPU3D block has the following hardware features:

- OpenGL ES 2.0 compliance, including extensions; OpenGL ES 1.1; OpenVG 1.1
- IEEE 32-bit floating-point pipeline

- Ultra-threaded, unified vertex and fragment shaders
- Low bandwidth at both high and low data rates
- Low CPU loading
- Up to 12 programmable elements per vertex
- Dependent texture operation with high-performance
- Alpha blending
- Depth and stencil compare
- Support for 8 fragment shader simultaneous textures
- Support for 4 vertex shader simultaneous textures
- Point sampling, bi-linear sampling, tri-linear filtering, and cubic textures
- Resolve and fast clear
- 8k x 8k texture size and 8k x 8k rendering target
- 4 Vertex DMA streams
- 2 texture units and 2 pixel units for higher pixel processing rate
- Supports YUV format in display output (YUYV 4:2:0)

Table 33-2. Unified vertex-fragment shader features

Feature	GPU Support
Shader type and execution units	Unified shader, SIMD4, SFP32 Trans
Swizzle capabilities	Full 32-bit word level swizzle in a 128-bit vector
GPR's per shader	Up to 512 general purpose registers, 128 bits each
Uniform registers	Vertex Shader: 168 registers, 128 bits each Fragment Shader: 64 registers, 128 bits each
FP denorm and rounding options	Denorms are set to zero. Supports rounding to zero.
Maximum number of data input attributes	Maximum of 12 vertex shader input elements; Maximum of 8 fragment shader input elements;
Maximum number of instructions	256 for vertex and 256 for fragment shaders
Maximum number of vertex streams	4
Maximum number of threads in flight per shader core	256
Subroutines	4 levels
Conditional branch support	GT, LT, EQ, GE, LE, NE
Shader instruction rate	1-cycle throughput for all shader instructions
Floating-point instruction precision	Transcendental: 22 bits SIMD4 (vector): 23.5 bits
Fragment shader video	Supports video texture
Control flow instructions	Yes

Table 33-3. Vertex Processing Features

Feature	GPU Support
Vx D3D, OGL ES formats supported	BYTE, UBYTE, SHORT, USHORT, INT, UINT, DEC, UDEC, FLOAT, FLOAT16, D3DCOLOR, FIXED16DOT16
Vertex data size limits	256 bytes
Pre shader cache	1 kB
Post shader cache	8 vertices

Table 33-4. Primitive Processing Features

Feature	GPU Support
Primitives supported	triangle strip, fan, and list; line strip line loop and list; point list
Vertex/primitive geometry input index sizes	8-bit ,16-bit and 32-bit indices
Setup parameters available to fragment shader	8 vec4 parameters; all available to fragment shader
AA line support	Yes

Table 33-5. Texture Processing Features

<table>
<tr><th>Feature</th><th>GPU Support</th></tr>
<tr><td>Fixed-point input texture formats</td><td>A8, L8, I8, A8L8, ARGB4, XRGB4, ARGB8, XRGB8, ABGR8, XBGR8, R5G6B5, A1RGB5, X1RGB5, YV12, YUY2, UYVY, D16, D24S8, A8_OES, DXT1, DXT2, DXT3, DXT4, DXT5, ETC1; all fixed-point formats are filtered.
<table>
<tr><td>Bits</td><td>Format</td><td>Alpha</td><td>R</td><td>G</td><td>B</td></tr>
<tr><td>16</td><td>ARGB4444</td><td>4</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>16</td><td>XRGB4444</td><td>4 don't care</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>16</td><td>ARGB1555</td><td>1</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>16</td><td>XRGB1555</td><td>1 don't care</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>16</td><td>RGB565</td><td>0</td><td>5</td><td>6</td><td>5</td></tr>
<tr><td>32</td><td>ARGB8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>XRGB8888</td><td>8 don't care</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>ABGR8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>XBGR8888</td><td>8 don't care</td><td>8</td><td>8</td><td>8</td></tr>
</table>
<table>
<tr><td>Planes</td><td>Format</td><td>Mode</td><td>Y</td><td>U</td><td>V</td><td>UV</td><td>YUYV</td><td>UYVY</td></tr>
<tr><td>3</td><td>YV12</td><td>4:2:0</td><td>1</td><td>1</td><td>1</td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>NV12</td><td>4:2:0</td><td>1</td><td></td><td></td><td>1</td><td></td><td></td></tr>
<tr><td>1</td><td>YUY2</td><td>4:2:2</td><td></td><td></td><td></td><td></td><td>1</td><td></td></tr>
<tr><td>1</td><td>UYVY</td><td>4:2:2</td><td></td><td></td><td></td><td></td><td></td><td>1</td></tr>
</table></td></tr>
<tr><td>Texture compression</td><td>4 bits and 8 bits per texel</td></tr>
</table>

Table continues on the next page...

Table 33-5. Texture Processing Features (continued)

Feature	GPU Support
Compressed texture formats	DXT1, DXT2, DXT3, DXT4, DXT5, ETC1 All compressed formats are filtered.
Texture size maximum	8k x 8k
Addressing modes	Wrap, mirror, clamp
Mipmap support	14 mipmap levels; programmable LOD biasing & replacement
Shadow texture	Depth texture PCF filtering
Texture cache organization	Tiled, 4x4 texels
Texture cache size	32 cache lines, with 64 bytes per cache line; 2 KB texture cache total
Texture coordinate fraction bits	5 bits
Texture sampler units	12 samplers, indexable
Textures per fragment maximum	8 texture samplers
Dependent texture operation	High performance; unlimited dependent texture reads
Dependent tx per fragment max, relative sampling	No limit
Texture repeat max	256
Texture types	2D, cube map, 1D, projected, depth, bump map, displacement map
Texture filters	Point sample, bi-linear, tri-linear
Texture component mapping: D3D, OGL, ES options	Supports both D3D and OES options
Texture size types	Power-of-2, Non-square

33.3.1 Rasterization

Table 33-6. Rasterization features

Feature	GPU Support
Interpolant attributes limit	8
Render target size	8K x 8K
Clipping window support	Clipping rectangle supported
Early Z	Yes

33.3.2 Fragment Processing

Table 33-7. Fragment Processing Features

Feature	GPU Support
FSAA mechanisms	High quality MSAA 4x, MSAA 16x for OpenVG

Table continues on the next page...

Table 33-7. Fragment Processing Features (continued)

<table>
<tr><td>Fragment color, alpha, Z, stencil precision</td><td>
<table>
<tr><th>Bits</th><th>Format</th><th>Alpha</th><th>R</th><th>B</th><th>G</th></tr>
<tr><td>16</td><td>ARGB4444</td><td>4</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>16</td><td>XRGB4444</td><td>4
don't care</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>16</td><td>ARGB1555</td><td>1</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>16</td><td>XRGB1555</td><td>1
don't care</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>16</td><td>RGB565</td><td>0</td><td>5</td><td>6</td><td>5</td></tr>
<tr><td>32</td><td>ARGB8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>XRGB8888</td><td>8
don't care</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>ABGR8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>XBGR8888</td><td>8
don't care</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>RGB10_a2ui</td><td>2</td><td>10</td><td>10</td><td>10</td></tr>
<tr><th>Bits</th><th>Format</th><th>Depth</th><th colspan="3">Stencil</th></tr>
<tr><td>16</td><td>D16</td><td>16</td><td colspan="3">0</td></tr>
<tr><td>32</td><td>D2488</td><td>24</td><td colspan="3">8</td></tr>
</table>
</td></tr>
<tr><td>Fragment storage</td><td>16-bit color and z, 32-bit color and z for each fragment. Lossless compression, no storage reduction.</td></tr>
<tr><td>Fragment cache</td><td>16 cache lines for color 16 cache lines for Z 64 bytes per cache line
32 cache lines for Z</td></tr>
<tr><td>Alpha support</td><td>Individual fragment alpha masking</td></tr>
</table>

33.3.3 Dest/Alpha Blending

Table 33-8. Dest/Alpha Blending Features

<table>
<tr><th>Feature</th><th>GPU Support</th></tr>
<tr><td>Destination color formats</td><td>
<table>
<tr><th>Bits</th><th>Format</th><th>Alpha</th><th>R</th><th>B</th><th>G</th></tr>
<tr><td>16</td><td>ARGB4444</td><td>4</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>16</td><td>ARGB1555</td><td>1</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>16</td><td>RGB565</td><td>0</td><td>5</td><td>6</td><td>5</td></tr>
<tr><td>32</td><td>ARGB8888</td><td></td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>ABGR8888</td><td></td><td>8</td><td>8</td><td>8</td></tr>
</table>
</td></tr>
<tr><td>Blend modes</td><td>Porter-Duff blending modes</td></tr>
<tr><td>Render target dithering support</td><td>Yes</td></tr>
</table>

33.3.4 Z/Stencil Buffer

Table 33-9. Z/Stencil Buffer Features

Feature	GPU Support
Z/stencil formats	16-bit Z; 24-bit Z plus 8-bit stencil, with lossless compression support
Z/stencil buffer	16 cache lines; 64 bytes per line;
Stencil compression	Yes

33.3.5 Render Target

Feature	GPU Support
Formats	16-bit and 32-bit, with lossless compression support

Table continues on the next page...

RT buffer cache	16 cache lines; 64 bytes per line; RT caches are fully set associative.

33.4 Usage Mode

The GPU3D should be programmed through the NXP provided driver. NXP does not provide support for software that directly programs the GPU3D registers. APIs for programming the GPU3D through the software driver are described in separate driver documentation.

Chapter 34
HDMI Transmitter (HDMI)

34.1 Overview

34.1.1 HDMI Operational Model Overview

The High Definition Multimedia Interface (HDMI) is a wired digital interconnect that replaces the analog TV out or VGA out.

HDMI is capable of transferring uncompressed video, audio, and data using a single cable. The video pixel rates are typically from 25 MHz up to 270 MHz (and 3D video modes), but HDMI can support higher rates up to 270 MHz. It can support S/PDIF (IEC60958 L-PCM and IEC61937 compressed non-linear PCM: AC-3, MPEG-1/-2 Audio, DTS®, MPEG-2/-4 AAC, ATRAC, WMA, MAT) and Parallel HBR (high bit rate) audio interface, enabling the support of Dolby® True-HD and DTS-HD Master Audio. HDMI has the capability of automatically setting the display format configuration (intelligent link).

HDMI include a content protection system called HDCP (High-bandwidth Data Content Protection). The HDMI connections can be used to connect DVD recorders, set-top boxes, and game consoles to flat panel televisions and an AV amplifier that can act as repeater/router.

HDMI system architecture consists of sources (transmitter) and sinks (receiver). As shown in the figure below, the HDMI cable and connectors carry four differential pairs that make up the TMDS data and clock channels. These channels are used to carry video, audio, and auxiliary data. In addition, HDMI carries a VESA Data Display Channel (DDC). The DDC is used for configuration and status exchange between a single source and a single sink. The optional CEC protocol provides high-level control functions between all of the various audiovisual products in a user's environment.

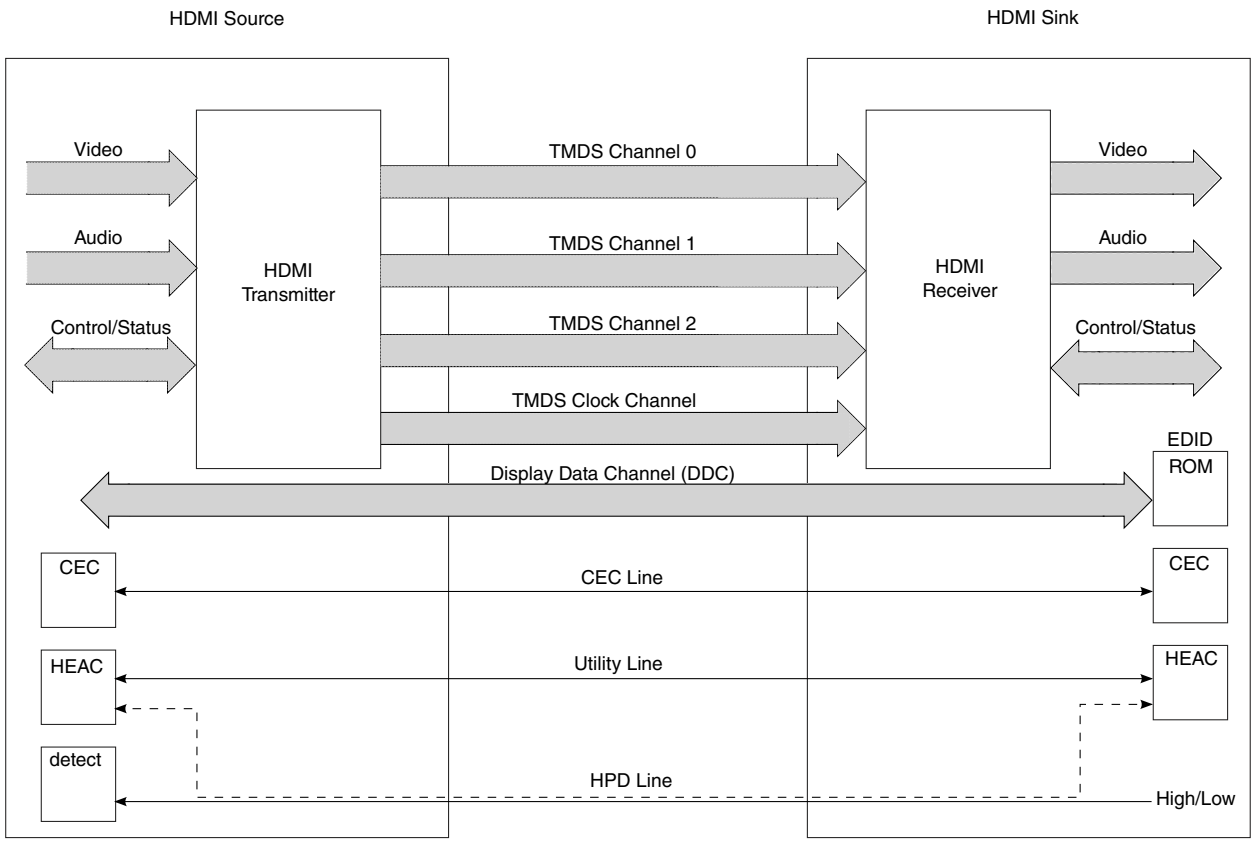


Figure 34-1. HDMI Block Diagram

Audio, video, and auxiliary data is transmitted across the three TMDS data channels. A TMDS clock running at 1x (24-bit true color mode), the video pixel rate is transmitted on the TMDS clock channel and used by the receiver as a frequency reference for data recovery on the three TMDS data channels. Video data can have a pixel size of 24bits . Video at the default 24-bit color depth is carried at a TMDS clock rate equal to the pixel clock rate. Higher color depths are carried using a correspondingly higher TMDS clock rate. Video formats with TMDS rates below 25MHz (such as, 13.5MHz for 480i/NTSC) can be transmitted using a pixel-repetition scheme. The video pixels can be encoded in either RGB, YCBCR 4:4:4, or YCBCR 4:2:2 formats.

HDMI uses a packet structure to transmit audio and auxiliary data across the TMDS channels. To attain the highest reliability required of audio and control data, this data is protected with a BCH error correction code and is encoded using a special error reduction code to produce the transmitted 10-bit word.

Basic audio functionality consists of a single IEC 60958 L-PCM audio stream (two audio channels) at sample rates of 32 KHz, 44.1 KHz, or 48 KHz, which can accommodate any normal stereo stream. Optionally, HDMI can carry audio at sample rates up to 192KHz

and with three to eight audio channels. HDMI can also carry an IEC 61937 compressed (such as, surround sound) audio stream at bit rates up to 24.576 Mbps. For bit rates above 6.144 Mbps, compressed audio streams conforming to IEC 61937 are carried using HBR Audio Stream Packets. Each packet carries four IEC 60958 frames, which corresponds to (4x2x16 =) 128 contiguous bits of an IEC 61937 stream.

The source uses the DDC to read the sink's Enhanced Extended Display Identification Data (E-EDID) to obtain the sink's configuration and/or capabilities.

The HDMI TX Controller schedules the three periods: Video Data Period, Data Island period, and Control period. During the Video Data Period, the active pixels of an active video line are transmitted. During the Data Island period, audio and auxiliary data are transmitted using a series of packets. The Control period is used when no video, audio, or auxiliary data needs to be transmitted. A Control Period is required between any two periods that are not Control Periods.

An example of each period placement is shown in the figure below.

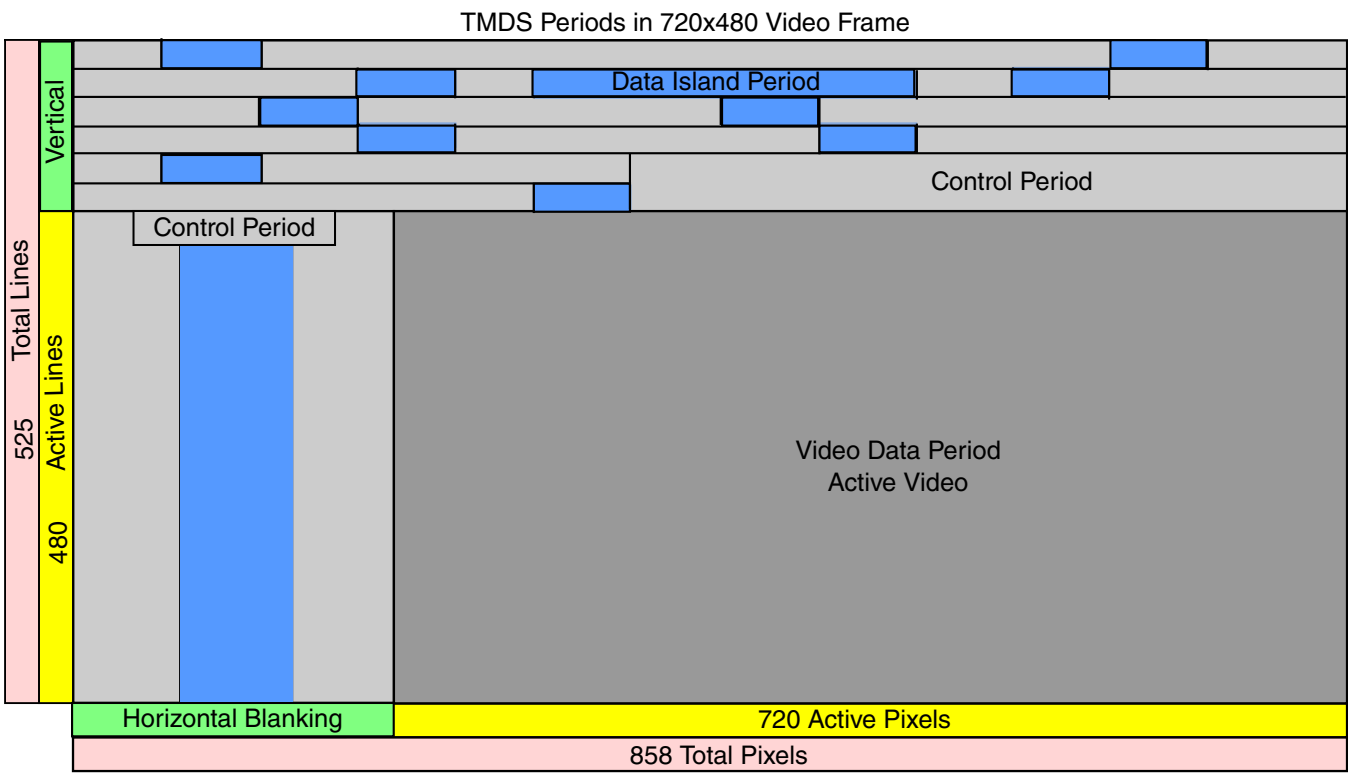


Figure 34-2. TMDS Periods in 720x480p Video Frame

34.1.1.1 Interfaces

HDMI TX has the following interfaces:

- HDCP interface
- External ROM interface for key storage
- External RAM interface for revocation
- Random number generator interface
- Video input interface
- RGB 4:4:4
- YCbCr4:2:2
- YCbCr4:4:4
- Digital audio input interface
- AHB audio DMA
- System interface
- AMBA AHB
- Scan test interface
- HDMI TX PHY interface
- CEC interface

34.1.1.2 Features

HDMI TX includes the following features:

- Supported video formats:
- All CEA-861-E video formats up to 1080p at 60Hz and 720p/1080i at 120Hz
- Supported colorimetry:
- 24bit RGB 4:4:4
- 24bit YCbCr 4:4:4
- 16bit YCbCr 4:2:2
- xvYCC601
- xvYCC709
- Integrated color space converter:
- RGB(4:4:4) to/from YCbCr(4:4:4 or 4:2:2)
- Optional HDMI 1.4a supported video formats:
- HDMI 1.4a 3D video modes with up to 270MHz (TMDS clock)
- Optional HDMI 1.4a supported colorimetry:
- sYCC601
- Adobe RGB
- Adobe YCC601
- Optional HDMI 1.4a supported Infoframes:
- Audio InfoFrame packet extension to support LFE playback level information
- AVI infoFrame packet extension to support YCC Quantization range (Limited Range, Full
- Range)

- AVI infoFrame packet extension to support Content type (Graphics, Photo, Cinema, Game)
- Supported Audio formats:
- Up to four I^2S interface for eight-channel Linear-PCM audio
- S/PDIF interface for linear and non-linear PCM formats:
  - AC-3
  - MPEG-1/-2 Audio
  - DTS
  - MPEG- 2/-4 AAC
  - ATRAC
  - WMA
  - MAT
- Parallel audio interface for High-Bit Rate (HBR) Audio:
  - Dolby® True-HD
  - DTS®-HD Master Audio
  - Generic Parallel Audio interface
- AHB DMA Audio interface
- Up to 192 KHz IEC60958 audio sampling rate
- Pixel clock from 13.5MHz up to 270 MHz
- Option to remove pixel repetition clock (prepclk) from HDMI TX interface for an easy integration with third-party HDMI TX PHYs
- Flexible synchronous enable per clock domain to set functional power down modes
- Register access:
- AMBA AHB
- I^2C DDC, EDID block read mode
- Advanced PHY testability
- Integrated CEC hardware engine

34.2 External Signals

See HDMI PHY for external signal information.

34.3 Clocks

The table found here describes the clock sources for HDMI.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 34-1. HDMI Clocks

Clock name	Clock Root	Description
iahbclk	ahb_clk_root	Bus clock
icecclk	ckil_sync_clk_root	CEC low-frequency clock (32kHZ)
ihclk	ahb_clk_root	Module clock
isfrclk	video_27m_clk_root	Internal SFR clock (video clock 27MHz)

34.4 Functional Description

This section describes the functional architecture of the HDMI TX controller.

34.4.1 HDMI TX Functional Overview

The HDMI TX provides a variety of standard audio, video, and system interfaces.

It includes an high-bandwidth data content protection (HDCP) encryption engine for HDMI receiver authentication, revocation, and data encryption.

Figure below illustrates the top level diagram of the HDMI TX solution.

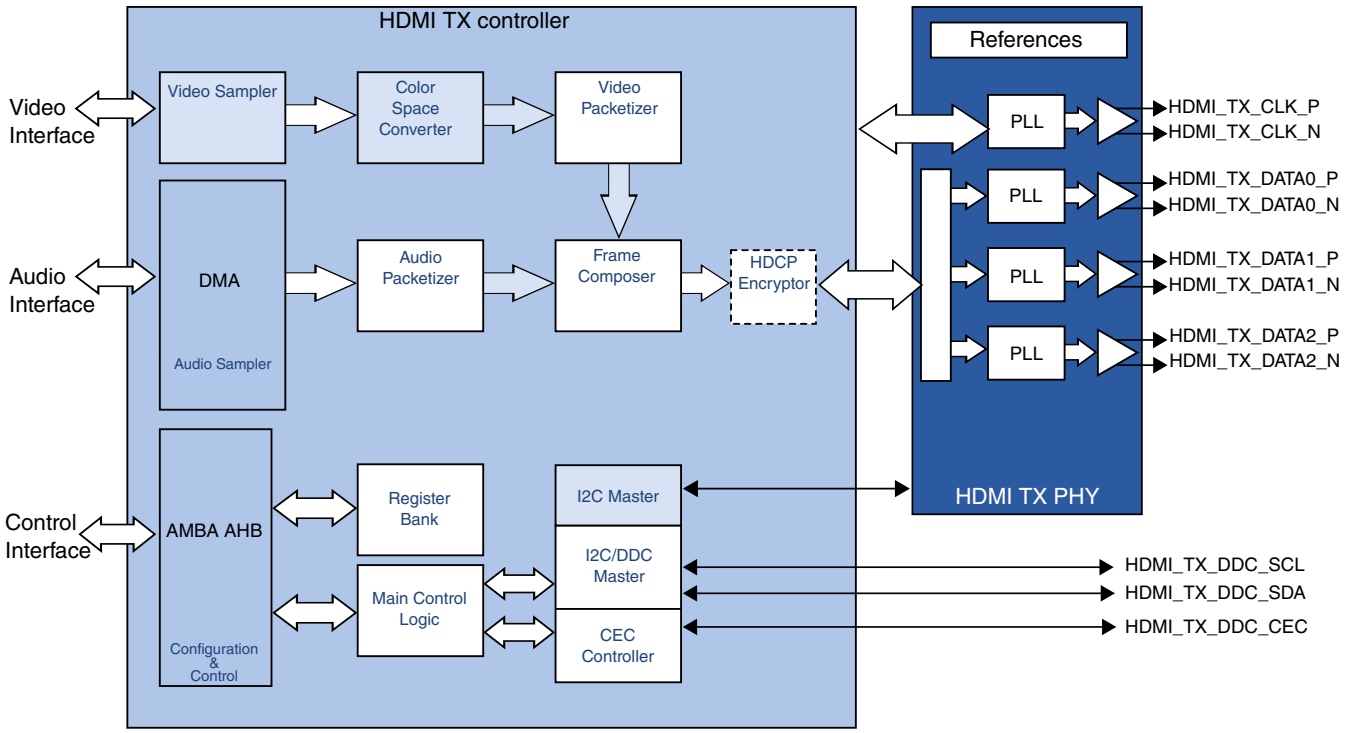


Figure 34-3. HDMI TX Top Level Block Diagram

The HDCP encryption engine is responsible for HDMI receiver authentication, revocation, and data encryption.

The input video stream can be either RGB 4:4:4, YcbCr 4:2:2, or YcbCr 4:44 in single data rate (SDR) bus formats as described in Table 34-2. The video mode's timing format must follow the CEA-861-E specification. An embedded color space conversion allows the pixel color format to be converted on the HDMI source side to match the best with the HDMI sink capabilities. 24

The input audio stream can be provided through audio AHB DMA interface;

Finally, HDMI_TX can output video in full HD with up to 48-bit color mode and inserts high fidelity audio up to eight-channels over low resolution video formats by performing automatic pixel repetition over the input video stream.

34.4.2 Video Pixel Sampler

The Video pixel sampler block is responsible for the video data synchronization, according to the video data input mapping defined by the Color Depth (Deep Color) and format configuration.

Optionally, for YCbCr 4:2:2 format, data mapping can be performed to conform to ITU. 601 and ITU.656 standards but without the support of embedded synchronizers.

The video pixel sampler registers base address is 0x0200.

34.4.2.1 HDMI Transmitter Controller Databook Functional Description

Table 34-2. Input Data Mapping

Video Input Format		ivdata[47:0] mapping																								
Color Space	Color Depth	47-46	45-44	43-42	41-40	39-38	37-35	35-34	33-32	31-30	29-28	27-26	25-24	23-22	21-20	19-18	17-16	15-14	13-12	11-10	9-8	7-6	5-4	3-2	1-0
RGB 4:4:4	8-bit	R[7:0]								G[7:0]								B[7:0]							
YCbCr 4:4:4	8-bit	Cb[7:0]								Y[7:0]								Cr[7:0]							
YCbCr 4:2:2	8-bit	Cb[7:0]								Y[7:0]															
		Cr[7:0]								Y[7:0]															

For each video timing format, there is a specific timing parameters defined in the CEA-861-E specification.

The following timing diagram is an example for the video mode format 1 (640x480p @ 59.94/60 Hz): Data Enable = idataen, HSYNC = ihsync, VSYNC = ivsync.

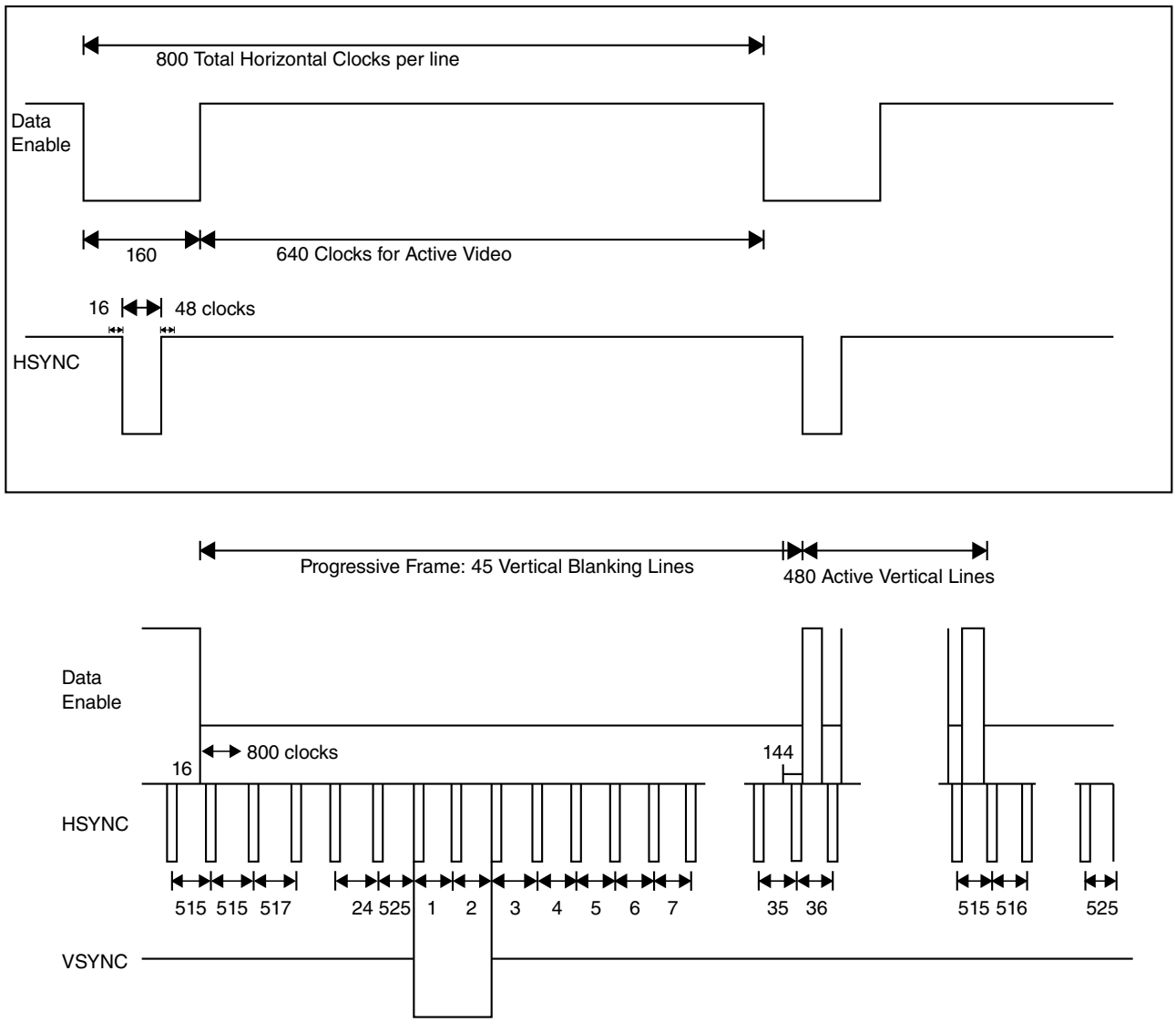


Figure 34-4. Timing Parameters for 640x480p @ 59.94/60 Hz

For a complete list of timing parameters and diagrams, refer to the CEA-861-E specification. The SDR video sample input format is illustrated in the figure below.

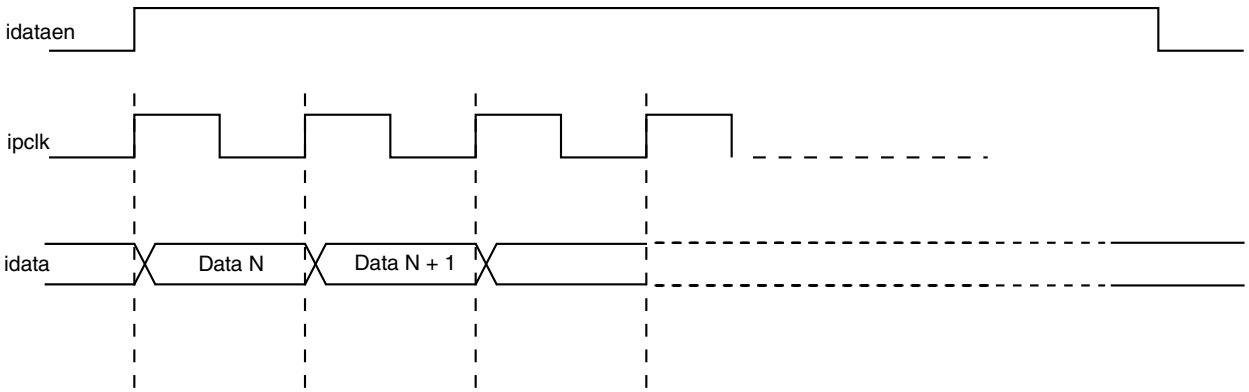


Figure 34-5. Video Sample Timing Interface for RGB and YCbCr SDR Format

34.4.3 Supported Video Mode

The table below shows examples of the supported video modes.

Table 34-3. Video Modes

				2D	3D Structure							
VideoMode	Mode	H x V Active Resolution (pixel)	Refresh Rate (Hz)	2D Pixel Rate (Mp/s)	Frame Packing Pixel Rate (Mp/s)	Field Alt. Pixel Rate (Mp/s)	Line Alt. Pixel Rate (Mp/s)	Side-by-Side (full) Pixel Rate (Mp/s)	L+depth Pixel Rate (Mp/s)	L +depth + graphics+ graphics-depth Pixel Rate (Mp/s)	Side-by-Side (Half) Pixel Rate (Mp/s)	Top-and-Bottom Pixel Rate (Mp/s)
					ALL CEA	Interlaced Only	Progr. Only	ALL CEA	Progr. Only	Progr. Only	ALL CEA	ALL CEA
Primary HDMI Video Format Timings (CEA-861-E)												

Table continues on the next page...

Table 34-3. Video Modes (continued)

1	640x480p (EDTV)	640 x 480	59.94	25.18	50.35		50.35	50.35	50.35	100.70	25.18	25.18
1			60.00	25.2	50.40		50.40	50.40	50.40	100.80	25.2	25.2
19	1280x720p (HDTV)	1280 x 720	50.00	74.25	148.50		148.50	148.50	148.50	297.00	74.25	74.25
4			59.94	74.18	148.35		148.35	148.35	148.35	296.70	74.18	74.18
4			60.00	74.25	148.50		148.50	148.50	148.50	297.00	74.25	74.25
20	1920x1080i (HDTV)	1920 x 1080	50.00	74.25	148.50	148.50		148.50			74.25	74.25
5			59.94	74.18	148.35	148.35		148.35			74.18	74.18
5			60.00	74.25	148.50	148.50		148.50			74.25	74.25
2.3	720x480p (EDTV)	720 x 480	59.94	27.00	54.00		54.00	54.00	54.00	108.00	27.00	27.00
2.3			60.00	27.03	54.05		54.05	54.05	54.05	108.11	27.03	27.03
6.7	720(1440)x480i (SDTV)	1440 x 480	59.94	27.00	54.00	54.00		54.00			27.00	27.00
6.7			60.00	27.03	54.05	54.05		54.05			27.03	27.03
17.18	720x576p (EDTV)	720 x 576	50.00	27.00	54.00			54.00			27.00	27.00
21.22	720(1440)x576i (SDTV)	1440 x 576	50.00	27.00	54.00	54.00		54.00			27.00	27.00
Secondary HDMI video format timings (CEA-861-E)												
8.9	720(1440)x240p	1440 x 240	59.94	27.00	54.00		54.00	54.00	54.00	108.00	27.00	27.00
8.9			60.00	27.03	54.05		54.05	54.05	54.05	108.11	27.03	27.03
10.11	1440(2880)x480i	2880 x 480	59.94	54.00	108.00	108.00		108.00			54.00	54.00
10.11			60.00	54.05	108.11	108.11		108.11			54.05	54.05
12.13	1440(2880)x240p	2880 x 240	59.94	54.00	108.00		108.00	108.00	108.00	216.00	54.00	54.00
12.13			60.00	54.05	108.11		108.11	108.11	108.11	216.22	54.05	54.05
32	1920x1080p (HDTV)	1920 x 1080	23.98	74.18	148.35		148.35	148.35	148.35	296.70	74.18	74.18
32			24.00	74.25	148.50		148.50	148.50	148.50	297.00	74.25	74.25
33			25.00	74.25	148.50		148.50	148.50	148.50	297.00	74.25	74.25
34			29.97	74.18	148.35		148.35	148.35	148.35	296.70	74.18	74.18
34			30.00	74.25	148.50		148.50	148.50	148.50	297.00	74.25	74.25
31			50.00	148.5	297.00		297.00	297.00	297.00		148.5	148.5
16			59.94	148.35	296.70		296.70	296.70	296.70		148.35	148.35
16			60.00	148.5	297.00		297.00	297.00	297.00		148.5	148.5
64			100.00	297.00							297.00	297.00

Table continues on the next page...

Table 34-3. Video Modes (continued)

63			120.00	297.00							297.00	297.00
40	1920x1080i (HDTV)	1920 x 1080	100.00	148.5	297.00	297.00		297.00			148.5	148.5
46			119.88	148.35	296.70	296.70		296.70			148.35	148.35
46			120.00	148.5	297.00	297.00		297.00			148.5	148.5
60	1280x720p (HDTV)	1280 x 720	24.00	59.40	118.80		118.80	118.80	118.80	237.60	59.40	59.40
61			25.00	74.25	148.50		148.50	148.50	148.50	297.00	74.25	74.25
62			30.00	74.25	148.50		148.50	148.50	148.50	297.00	74.25	74.25
41			100.00	148.5	297.00		297.00	297.00	297.00		148.5	148.5
47			119.88	148.35	296.70		296.70	296.70	296.70		148.35	148.35
47			120.00	148.5	297.00		297.00	297.00	297.00		148.5	148.5
29, 30	1440x576p (EDTV)	1440 x 576	50.00	54.00	108.00		108.00	108.00	108.00	216.00	54.00	54.00
37.38	2880x576p (EDTV)	2880 x 576	50.00	108.00	216.00		216.00	216.00	216.00		108.00	108.00
14.15	1440x480p (EDTV)	1440 x 480	59.94	54.00	108.00		108.00	108.00	108.00	216.00	54.00	54.00
14.15			60.00	54.05	108.11		108.11	108.11	108.11	216.22	54.05	54.05
35.36	2880x480p (EDTV)	2880 x 480	59.94	54.00	108.00		108.00	108.00	108.00	216.00	54.00	54.00
35.36			60.00	54.05	108.11		108.11	108.11	108.11	216.22	54.05	54.05
48.49	720x480p (EDTV)	720 x 480	119.88	54.00	108.00		108.00	108.00	108.00	216.00	54.00	54.00
48.49			120.00	54.05	108.11		108.11	108.11	108.11	216.22	54.05	54.05
56.57			239.76	108.00	216.00		216.00	216.00	216.00		108.00	108.00
56.57			240.00	108.11	216.22		216.22	216.22	216.22		108.11	108.11
50.51	720(1440)x480i (SDTV)	1440 x 480	119.88	54.00	108.00	108.00		108.00			54.00	54.00
50.51			120.00	54.05	108.11	108.11		108.11			54.05	54.05
58.59			239.76	108.00	216.00	216.00		216.00			108.00	108.00
58.59			240.00	108.11	216.22	216.22		216.22			108.11	108.11
42.43	720x576p (EDTV)	720 x 576	100.00	54.00	108.00		108.00	108.00	108.00	216.00	54.00	54.00

Table continues on the next page...

Table 34-3. Video Modes (continued)

52.53			200.00	108.00	216.00		216.00	216.00	216.00		108.00	108.00
44.45	720(1440)x576i (SDTV)	1440 x 576	100.00	54.00	108.00	108.00		108.00			54.00	54.00
54.55			200.00	108.00	216.00	216.00		216.00			108.00	108.00
23.24	720(1440)x288p	1440 x 288	50.00	27.00	54.00		54.00	54.00	54.00	108.00	27.00	27.00
25.26	720(1440)x576i	1440 x 576	50.00	54.00	108.00	108.00		108.00			54.00	54.00
27.28	1440(2880)x288p	2880 x 288	50.00	54.00	108.00		108.00	108.00	108.00	216.00	54.00	54.00
39	1920x1080i	1920 x 1080	50.00	72.00	144.00	144.00		144.00			72.00	72.00

34.4.4 Video Packetizer

This block is responsible for:

- Pixel repetition (if not already performed in the input video stream and needed by the user)
- 10-bit, 12-bit, and 16-bit packing when in deep color modes
- YCC 422 remapping according to the HDMI 1.4a specification
- Clock rate transformation from pixel or repetition clock to the final TMDS clock domain (by means of FIFOs)

The figure below depicts a functional diagram of the Video Packetizer block.

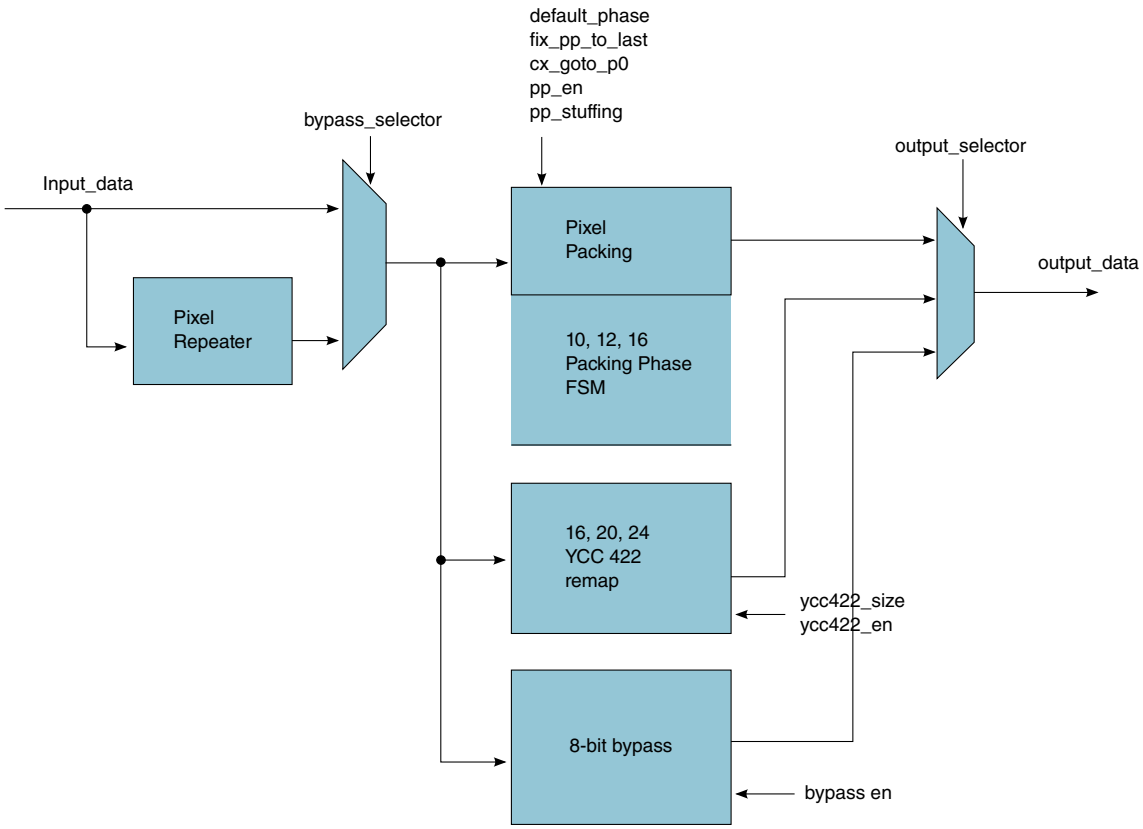


Figure 34-6. Video Packetizer Functional Diagram

34.4.5 Color Space Conversion

This block is responsible for carrying out the following video color space conversion functions:

- RGB to/from YCbCr
- 4:2:2 to/from 4:4:4 up (pixel repetition or linear interpolation)/down-converter
- Limited to/from full quantization range conversion

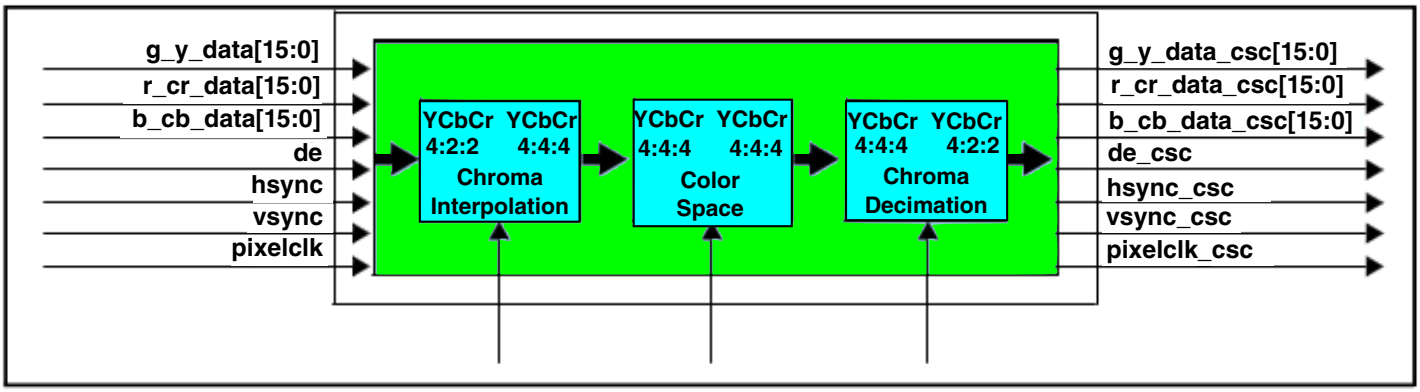


Figure 34-7. Color Space Converter Simplified Block Diagram

The Color Space Converter (CSC) supports all the timings reported in the CEA-861-D specification and the following pixel modes:

- RGB444 and YCbCr444: 24, 30, 36, and 48 bits
- YCbCr422: 16, 20, and 24 bits

The color space conversion matrix is ruled by the following equations listed below. The color space conversion registers base address is 0x4100.

$$out_1 = (X_1 \times in_1 /4096 + X_2 \times in_2 /4096 + X_3 \times in_3 /4096 + X_4) \times 2^{scale}$$

$$out_2 = (Y_1 \times in_1 /4096 + Y_2 \times in_2 /4096 + Y_3 \times in_3 /4096 + Y_4) \times 2^{scale}$$

$$out_3 = (Z_1 \times in_1 /4096 + Z_2 \times in_2 /4096 + Z_3 \times in_3 /4096 + Z_4) \times 2^{scale}$$

34.4.6 Audio Interfaces

The supported audio input interfaces are:

- AHB Direct Memory Access (DMA)

No lipsync support is available inside the HDMI TX. If necessary, this feature can be performed at the system audio processor side. From the HDMI TX, no audio/video delay or skew is added.

The audio sampler registers base address is 0x3100.

34.4.6.1 CTS Calculation

Because there is no audio clock carried through the HDMI link, only the pixel clock is used.

The CTS/N has to be set by software with value taken in the following table. Table below shows the CTS and N value for the supported standard. All other TMDS clocks are not supported; the TMDS clocks divided or multiplied by 1,001 coefficients are not supported.

Table 34-4. N and CTS for 8-Bit Color Depth

	TMDS Clock (MHz)											
	25.2		27		54		74.25		148.5		297	
Fs (kHZ)	N	CTS	N	CTS	N	CTS	N	CTS	N	CTS	N	CTS
32	4096	25200	4096	27000	4096	54000	4096	74250	4096	148500	3072	222750
44.1	6272	28000	6272	30000	6272	60000	6272	82500	6272	165000	4704	247500
48	6144	25200	6144	27000	6144	54000	6144	74250	6144	148500	5120	247500
88.2	12544	28000	12544	30000	12544	60000	12544	82500	12544	165000	9408	247500
96	12288	25200	12288	27000	12288	54000	12288	74250	12288	148500	10240	247500
176.4	25088	28000	25088	30000	25088	60000	25088	82500	25088	165000	18816	247500
192	24576	25200	24576	27000	24576	54000	24576	74250	24576	148500	20480	247500
768	Used for HBR audio only. N and CTS configured for Fs=192kHZ (1/4th ACR value per spec)											

To support the deep color mode and/or 3D video modes, the TMDS clock is multiplied by 4, 2, 1.5, or 1.25, depending on the mode. In this case, the CTS value must also follow the same ratio.

34.4.6.2 Audio DMA Interface

This audio direct memory access (DMA) interface is intended for advanced systems running 32-bit CPU SoC solutions.

This module provides a direct audio DMA interface, which is useful in systems where a DSP handles audio processing. In these systems, sending the incoming audio samples directly to the memory provides a cleaner architecture to the SoC, without the overhead of converting several audio standards.

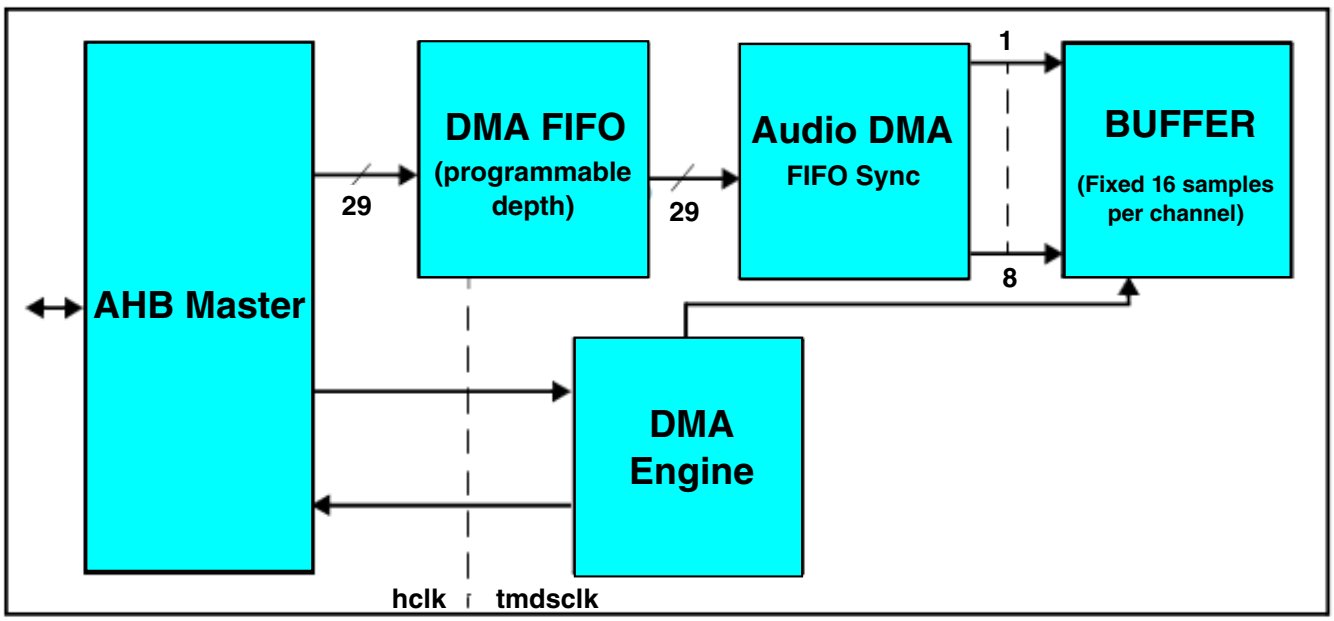


Figure 34-8. Audio DMA Block Diagram

The audio DMA block combines an AHB master interface with a FIFO to perform direct memory access to audio samples stored in a system memory.

The DMA engine is configurable through programmable software registers to perform autonomous burst reading on a configured memory range.

34.4.6.2.1 AHB Master

The AHB master is compliant with the AMBA AHB Specification, Revision 2.0 from ARM.

It has the following features:

- Multi-master capable operation
- 32-bit data transfer
- OKAY, ERROR, RETRY, and SPLIT slave responses
- Rescheduling of burst requirements
- IDLE, NONSEQ, and SEQ transfer types
- Incremental burst modes: unspecified lengths (upper limit is 1 kB boundary) and INCR, INCR4, INCR8, and INCR16 fixed-beat bursts
- Master burst lock mechanism

- Bus access granting
- The following features are not supported:
- Write transaction
- Protection control
- BUSY transfer type
- Wrapping burst

Data Organization in System Memory

The AHB master block fetches the samples from system memory. The Audio Samples are organized according to the channel allocation.

For example, channel 0, 1, 3, 5 are enabled (0 and 1 are always enabled). The Audio Samples must be organized in the system memory like the following:

Table 34-5. Audio Sample Arrangement in System Memory

Position	Sample	Channel
0	n-1	0
1	n-1	1
2	n-1	3
3	n-1	5
4	n-1	0
5	n-1	1
6	n-1	3
7	n-1	5
...	...	...

Table 34-6. Data Arrangement in System Memory for L-PCM (24 bits)

Bit	Description
28	B - IEC B bit
27	P - Parity bit
26	C - Channel Status bit
25	U - User Data bit
24	V - Validity Bit
[23:0]	Audio Sample Data

Table 34-7. Data Arrangement in System Memory for L-PCM (16 bits) and NL-PCM (16 bits)

Bit	Description
28	B - IEC B bit
27	P - Parity bit
26	C - Channel Status bit

Table continues on the next page...

Table 34-7. Data Arrangement in System Memory for L-PCM (16 bits) and NL-PCM (16 bits) (continued)

25	U - User Data bit
24	V - Validity Bit
[23:8]	Audio Sample Data
[7:0]	0x00

34.4.6.2.2 DMA Engine

The DMA engine is responsible for requesting burst transfers to the AHB master, taking into account the FIFO threshold and register settings.

34.4.6.2.2.1 Functional Behavior

The engine:

- Arbitrates read requests to start the burst in the initial address with the size sufficient to fill the FIFO (the size of the FIFO is a parameter in the audio DMA core).
- After this first request, the DMA engine performs subsequent burst requests (incrementing accordingly ohaddr[31:0] and determining correct ohburst[2:0]) towards final_addr[31:0] configured at the register bank and taking into account the AUDIO_FIFO_DEPTH parameter and fifo_threshold[7:0] configuration.
- In the burst mode (INCR4, INCR8, INCR16), the operation stops at the end of the burst.
- Stops operation upon ERROR slave response, signaling ointerror interrupt and staterror signal
- Continues burst transaction:
- Upon RETRY/SPLIT slave response, signaling ointretrysplit interrupt and statretrysplit signal
- Upon losing ownership (no ihgrant) as consequence of arbiter action, signaling ointlostownership interrupt and statlostownership signal
- Decides through register configuration which burst method (unspecified length incrementing or fixed beat incrementing) to use in the read transfers
- Issues ointdone interrupt when it reaches final address reading or is stopped upon user request
- Automatically starts new burst requests until the final_addr[31:0] is reached
- The DMA engine is either stopped by the user or an error/fail condition appears at the slave response.
- Takes into account that an incrementing burst can be of any length (if unspecified INCR type), but upper limit is set because the address must not cross a 1kB boundary

- A maximum theoretical length of a burst is 1024. The burst size must be declared on the mburstlength_addr[10:0].
- Has INCR with an unspecified burst as the default operation burst mode

34.4.6.2.2.2 DMA Operation

Normal operation of the DMA engine is as follows:

1. The enable_hlock, incr_type[1:0], burst_mode, fifo_threshold[7:0], initial_addr[31:0], and final_addr[31:0] are configured according to desired DMA operation.

   NOTE

   Configured values have to follow these rules:

   The number of memory positions (between initial_addr and final_addr) has to be a multiple of theactive audio channels.

   The final_addr[31:0] signal is always bigger than the initial_addr[31:0].

2. To start the audio DMA operation, a '1' is written to data_buffer_ready.
3. The DMA engine starts the operation. The first burst transfer is:

```
        ohaddr[31:0] = initial_addr[31:0];
        ohburst[2:0] = INCR;
        mburstlength[10:0] = ((initial_addr[31:0] + AUDIO_FIFO_DEPTH) <=
final_addr[31:0]) ?
        ((AUDIO_FIFO_DEPTH < 1024) ? AUDIO_FIFO_DEPTH : 1024) : (final_addr[31:0] -
initial_addr[31:0]);
```

4. While DMA is reading samples from the AHB master and writing samples to the Audio FIFO, a datafetch request from the internal frame composer block might happen at the Audio FIFO interface, diminishing the number of samples in the FIFO.
5. When the number of samples in the Audio FIFO is lower than the configured fifo_threshold[7:0], the DMA engine requests a new burst request to the AHB master interface with:

```
ohaddr[15:0] = last address in step 5);

ohburst[2:0] = INCR;

mburstlength[10:0] = AUDIO_FIFO_DEPTH - fifo_threshold[7:0];
```

6. Steps 4) and 5) continue until the final_addr[31:0] is reached.

NOTE

In the last burst request, the DMA engine calculates the mburstlength[10:0] such that the last requested read position is the final_addr[31:0].

7. After completion of the DMA operation, the DMA engine issues the ointdone interrupt signaling end of operation.

Variations of the DMA engine's behavior occur when fixed-beat, incremental bursts are used by INCR4/INCR8/INCR16 burst selects. When this forcing mode is used, you must correctly configure the FIFO's threshold such that the last of the consecutive INCRx (with x = 4, 8, or 16) bursts correctly fill the FIFO at last burst received. Note there is a re-alignment at the 1k boundary.

The following are exceptions to the described DMA behavior:

1. When a user requests end stop_dma_transaction, the DMA engine stops at the end of the current burst operation and signals its completion with an ointdone interrupt.
2. When the AHB slave sends an error response, the DMA engine stops the current operation and signals ointerror interrupt.

Rules for Configuration of Address Registers:

1. Configure the last 2 bits of initial_addr with 0.

```
For example, 32'h0000_0000.
```

2. Configure the last 2 bits of final_addr with non-zero values.

```
For example
32'hxxx_xxx3 or
32'hxxx_xxx7 or
32'hxxx_xxxB or
32'hxxx_xxxF
Where x= any value
```

3. The number of samples is calculated by using the following formula:

```
Number of samples = (final_addr - initial_addr + 1) / 4
Therefore, final_addr = (Number of samples x 4) + initial_addr -1
If a defined length burst is used, align initial_addr, final_addr and
fifo_threshold with the value. If
the burst is not aligned, DMA uses AHB INCR transfers when required.
```

```
Then the number of samples = 100 (a multiple of 5)
```

4. The threshold must be

```
Greater than the selected FIFO DEPTH;

Greater than the number of channels enabled in channel allocation.
```

Due to the limit of audio DMA design, some registers configuration are updated by SDMA (Smart Direct Memory Access). SDMA need to do these items below:

1. clear the audio DMA done request(actually it's an interrupt); set offset 0x00120109 -- bit2.
2. configure next audio DMA start address(offset 0x00123604~0x00123607) and next stop address(offset 0x00123608~0x0012360b); Start address and stop address are provided by S/W and it's variable.
3. Set offset 0x00123601 – bit0 "1" to start DMA;

34.4.6.2.2.3 Transfer Data, Package, and Word

One transfer data can be composed of several transfer packages, and one transfer package can be composed of one or several transfer bursts. One transfer burst can be composed of several transfer words.

The figure below shows the transfer data structure for a fixed-beat, incremental burst.

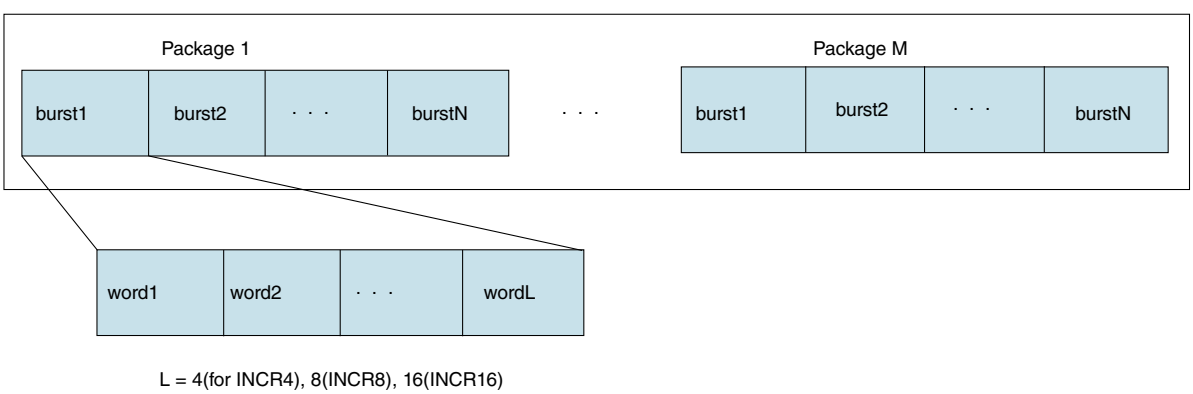


Figure 34-9. Transfer Data Constitution for Fixed-Beat, Incremental Burst

The figure below depicts the transfer data structure for an unspecified burst length.

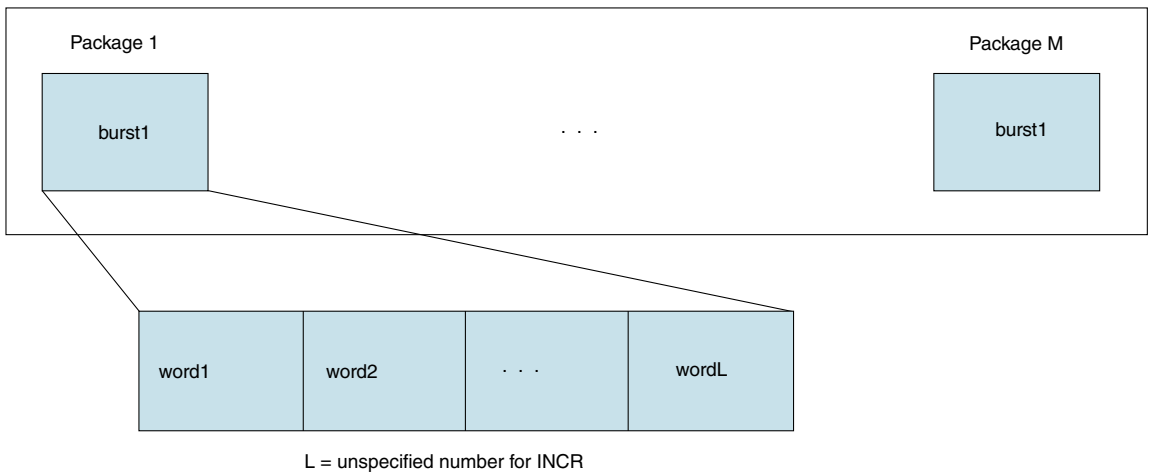


Figure 34-10. Transfer Data Constitution for Unspecified Burst Length

The figure below illustrates the DMA state machine.

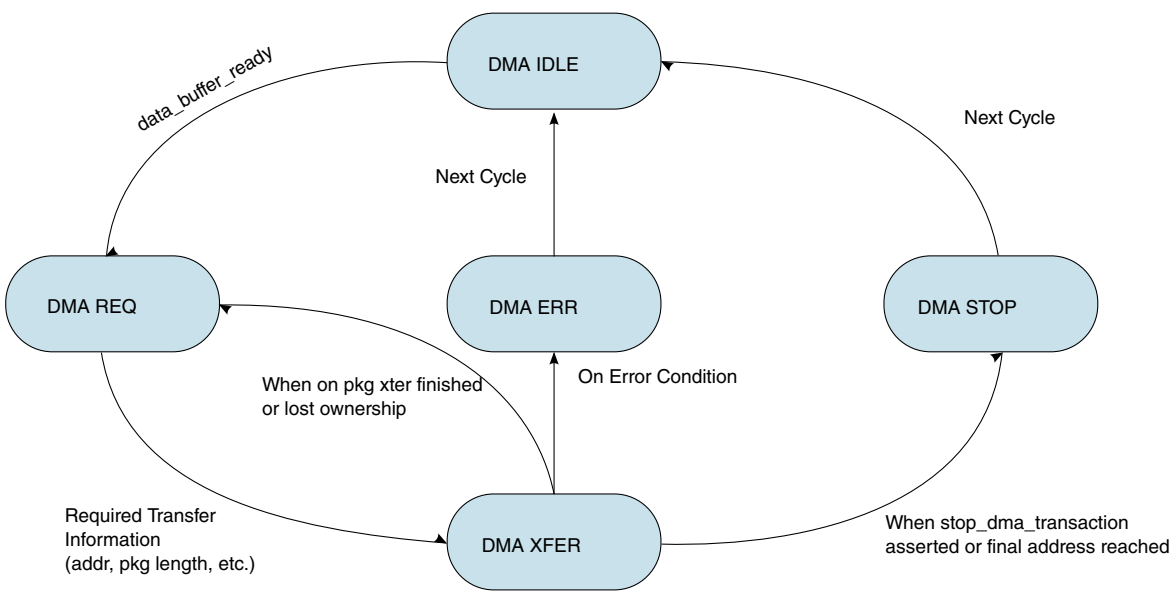


Figure 34-11. DMA FSM Diagram

1. When the operation request is written into the data_buffer_ready, the state switches from IDLE to DMAREQ.
2. When enough transfer information is ready, the state changes to DMAXFER.

3. When one package is completed-indicated by "OnePackOver"-or a lost ownership occurs, the state jumps back to DMAREQ, so the new transaction information can be calculated.
4. When the DMA is ready again, it jumps to DMAXFER.

The previous steps continue until you request an end operation or until the whole operation is completed.

34.4.6.2.3 Audio FIFO

This block contains a FIFO with a configurable depth.

The statthrfiffoempty flag is a version of the FIFO empty, that is active whenever the amount of samples in the FIFO is smaller than four samples. The AHB_DMA_THRSLD and the statthrfifoempty indicators are different. The AHB_DMA_THRSLD defines the occupation of the FIFO, while statthrfifoempty helps the HDMI TX's Frame Composer in the audio packet composition (required when non-linear audio is being packed, in which case four pair of samples are needed to compose one packet).

34.4.6.2.4 FIFO Occupancy/FIFO Almost Empty Flags

The FIFO depth is configured by setting the AUDIO_FIFO_DEPTH parameter in coreConsultant. The FIFO occupancy is calculated from the pointer values.

Also, the most significant pointer bits are used to perform the following calculation:

```
        occupancy = wrptr[n-1:0] - rdptr_previous[n-1:0], if wrptr > rdptr_previous

        occupancy = AUDIO_FIFO_DEPTH + wrptr[n-1:0] - rdptr_previous[n-1:0], if wrptr <
rdptr_previous
```

If occupancy is less than 4, then statthrfifoempty is active.

An example of full/empty flags and FIFO occupancy is shown in the figure below.

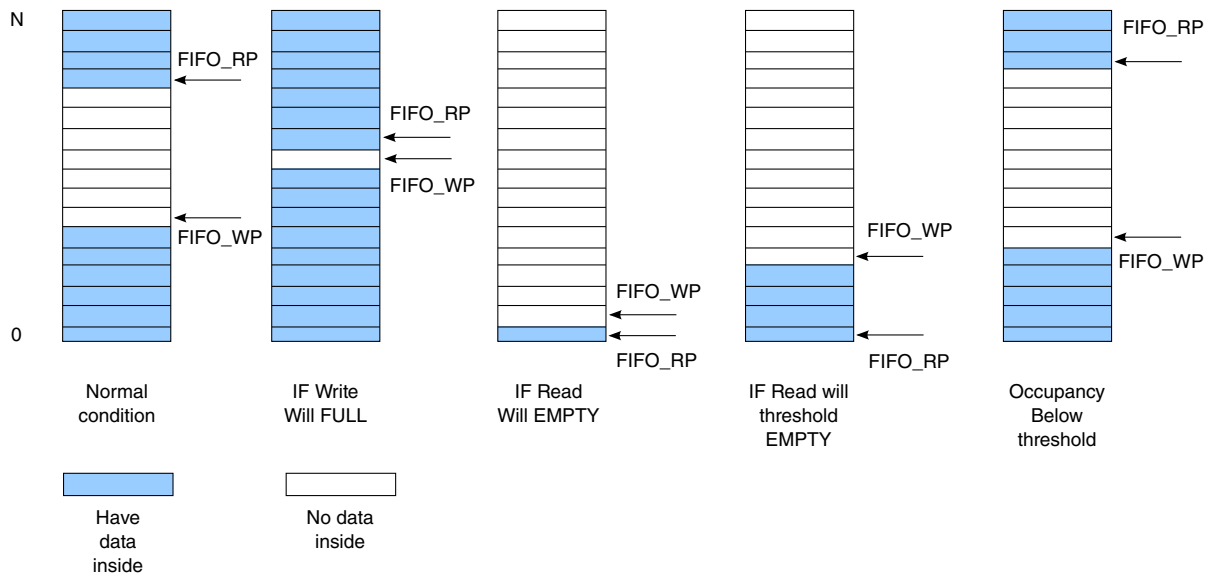


Figure 34-12. Audio FIFO Status Indication

34.4.7 Supported Audio Formats

The HDMI TX has several audio interfaces and each of them has different audio format support capabilities.

Table below represents the audio interface and format dependencies.

Table 34-8. Supported Audio Formats

Audio Input Interface	Audio Format Supported
Audio DMA	Up to eight channels L-PCM/NL-PCM and HBR audio, allowing all audio formats listed to support one single audio DMA interface.

34.4.8 Frame Composer

This block is responsible for assembling video, audio, and data packets in a consistent frame that are streamed to the HDCP cipher and then finally to the HDMI TX PHY.

The HDMI 1.3a standard precisely describes the packet insertion timing and distribution that must be followed to correctly compose an HDMI TMDS (transition minimized differential signaling) stream. In this context, there are data island packets that-when available (ready for insertion in output stream)-have higher priority over others. Two packet descriptor queues are responsible for prioritizing packet insertion.

The higher priority packets are described in Table below. These packets are inserted in the output stream as soon as data to compose them is available (see the HDMI 1.3a standard).

Table 34-9. High Priority Data Island Packets

Packet	Description
Audio Clock Regeneration (ACR)	Indicates to sink device the N/CTS values that should be used in the ACR process
Audio Sample (AUDS)	Transports L-PCM and IEC 61937 compressed audio
General Control (GCP)	Indicates Color Depth, Pixel Packing phase, and AV mute information to sink device

The packets described in tables below can be considered as low priority packets-even though they have precise timing insertion-because their insertion timing is large (for example, one per frame or one per two frames without specific location for some of the packet types and on user request transmit for others).

Table 34-10. Low Priority Data Island Packets

Packet	Description
Audio Content Protection (ACP)	Used to convey content-related information about the active audio stream transmitted
Audio InfoFrame (AUDI)	Indicates characteristics of the active audio stream by using IEC 60958 channel status bits, IEC 61937 burst info, and/or stream data (if present).
Null (NULL)	Ignored by sink devices.
International Standard Recording Code (ISRC1/ISRC2)	See HDMI 1.3a section 5.3.8.
Vendor Specific (VSD) InfoFrame	According to CEA-861-E standard.
AVI infoFrame (AVI)	Video information from source to sink.
Source Data Product Descriptor (SPD) infoFrame	Name and product type of the source device. MPEG (MPEG) Source InfoFrame packets (optional, implementation discouraged by CEA-861-E Section 6.7). Describes several aspects of the compressed video stream that were used to produce the uncompressed video.

The Frame Composer distributes and assembles the data island packets according to the module register bank configuration. The block allows extended control periods to appear with a certain programmed spacing. The Frame Composer uses two packet buffers that allow a packet to be composed while another is being sent to the output HDMI stream.

Packet requests are inserted into the packet queues by a data island flexible scheduler. The HDMI 1.3a specification requires that packet distribution and insertion timing correctly compose an output HDMI TMDS stream. In this context, there are data island packets that are sent on data availability, while others are sent once per frame or once per two frames. and finally others that are sent on user request. Classification of the packets according to this insertion timing is described in the table below.

Table 34-11. Packet Classification

Packet	Classification
Audio Clock Regeneration (ACR)	Sent on data availability.
Audio Sample (AUDS)	Sent on data availability (precede ACR if present).
Audio Content Protection (ACP)	On user request or automatic insertion.
Audio InfoFrame (AUDI)	Once per two frames.
Null (NULL)	On user request or automatic insertion to fill Data Island period.
General Control (GCP)	Once per frame.
International Standard Recording Code (ISRC1/ ISRC2)	On user request.
Vendor Specific (VSD) InfoFrame	On user request or automatic insertion.
AVI infoFrame (AVI)	Once per frame.
Source Data Product Descriptor (SPD) infoFrame	On user request or automatic insertion.

The Data Island Scheduler (DIS) handles packet distribution in the Frame Composer. The DIS is a round- robin (RDRB) state machine that is able to schedule packet insertion on an input video frame or line basis. The DIS is fully configurable and can schedule any packet type to be inserted at a given input video frame rate or input video line rate.

While determining packet distribution on an input video frame or line basis, the DIS schedules the packets to be inserted in the output HDMI stream by inserting the packet descriptor in the corresponding packet priority queue, according to packet priority classification.

After the packet descriptor has been inserted in the packet priority queues, the Data Island Packer (DIP) is responsible for assembling and sequencing the packets for output HDMI stream insertion.

Dedicated ECC generators and checksum byte-wide sum hardware generate the BCH ECC parity codes and infoFrames checksums for all the data islands packets.

The content of GCP, ISRC1/2, VSD, AVI, SPD, and MPEG packets are configured through the registers bank starting at address 0x1000.

34.4.9 HDCP Encryption Engine

HDCP is designed to protect the transmission of audio-visual content between an HDCP Transmitter and an HDCP Receiver.

The system also allows for HDCP Repeaters that support downstream HDCP-protected interface ports. The HDCP system allows up to seven levels of HDCP Repeaters and as many as 128 total HDCP devices, including HDCP Repeaters, to be attached to an HDCP-protected interface port.

NOTE

This feature must be configured and requires a separate license. Contact your NXP representative for more information on HDCP.

The authentication protocol enables the HDCP Transmitter to verify that a given HDCP Receiver is licensed. With the legitimacy of the Receiver determined, encrypted HDCP content is transmitted between the two based on shared secrets established during authentication. In the event that legitimate devices are compromised to permit unauthorized use of the content, renewability allows an HDCP Transmitter to identify such compromised devices and prevent the transmission of the content.

The implemented HDCP functionality is compliant with HDCP revision 1.4. The HDCP transmitter implements the three layers of the HDCP cipher, including LFSR and other functions required to generate the encryption key bytes that are then XORed with the data.

34.4.10 EDID/HDCP I^2C E-DDC Interface

The E-DDC channel is a dedicated I2C master interface that allows the read of sink E-EDID based on system needs.

Data read from sink E-EDID can be then transferred to the I2C Master register bank, starting at address 0x7E00.

This block reads the E-EDID (and all its segments according to user configuration) and, after completion, it warns the CPU of data availability. This block also arbitrates the I2C master interface to allow the HDMI TX's HDCP authentication protocol to be performed through this interface. Sink HDCP links (with addresses 0x74 and/or 0x76) should be present at these lines to enable HDCP-compliant behavior.

The interface is shared with the DDC channel of the HDMI controller through multiplexers and is I2C compliant.

34.4.10.1 I²C Master Interface Normal Mode

This operation implements a single read or write operation using the Special Function Register configuration.

The I²C data transfer protocol used is the 7-bit addressed, as defined in Section 9 of the I²C-bus Specification, version 2.1.

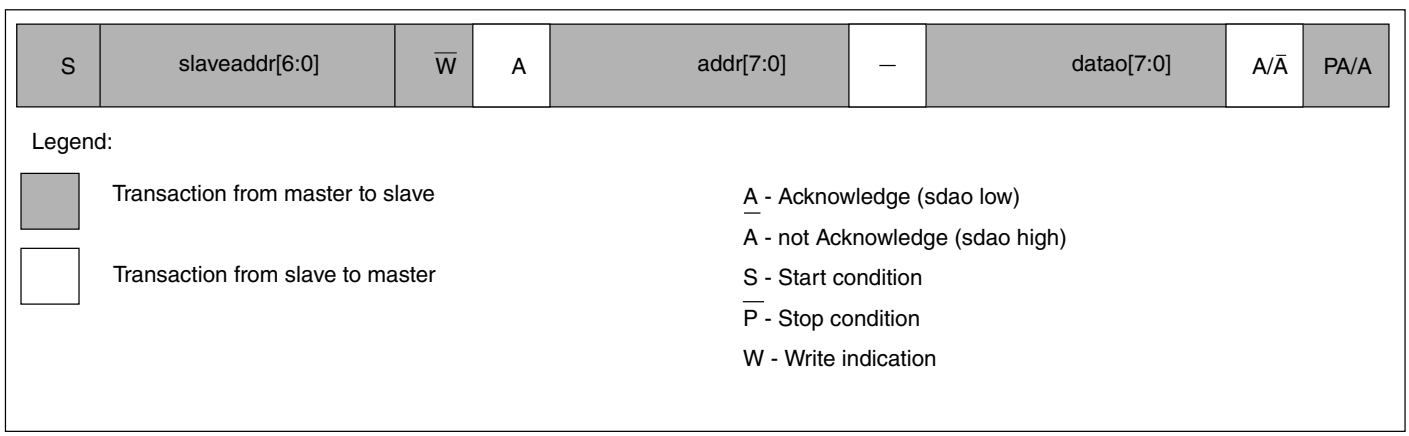


Figure 34-13. Data Write Transaction

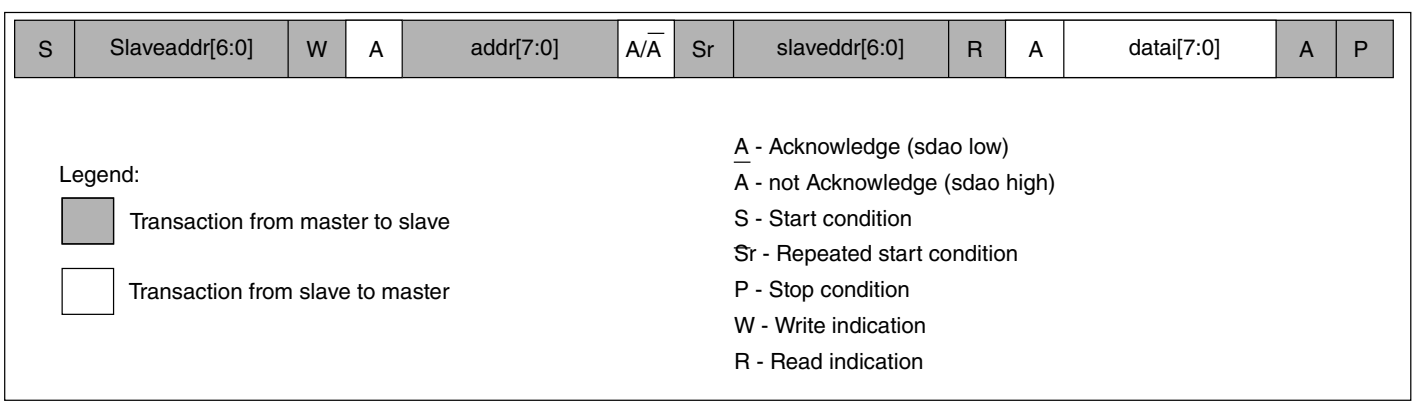


Figure 34-14. Data Read Transaction

34.4.10.2 I²C Master Interface Extended Read Mode

This I2C extended read mode operation implements a segment pointer-based read operation using the Special Register configuration.

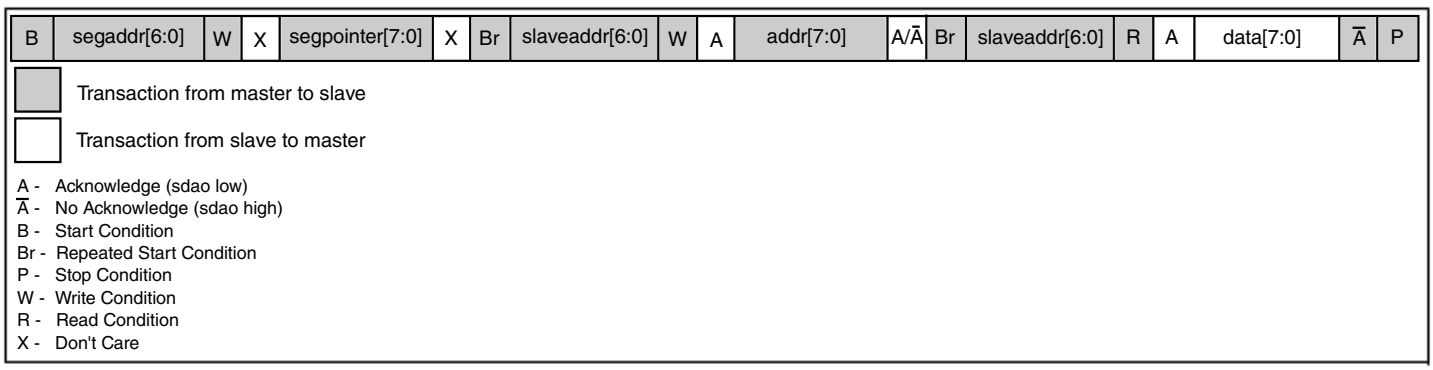


Figure 34-15. Extended Data Read Operation

34.4.11 System Configuration Interfaces

The internal register set is distributed with the HDMI TX. The system interface (the interface that connects to the processor bus) bridges these registers using a simple standard interface.

The system interface is AMBA AHB,.

The AHB bridges the bus to the internal SFR bus.

34.4.11.1 AMBA AHB Slave Interface

The AMBA AHB slave interface is compatible with the AMBA Specification, revision 2.0.

The AHB slave interface is used for register configuration implementing only a simple transaction mode and single master slave operation.

The HDMI TX does not support protection control, burst transfers, or split transactions.

34.4.12 CEC Hardware Engine

Consumer Electronics Control (CEC) is a protocol that provides high-level control functions between all of the various audiovisual products in a user's environment.

It is an optional feature in the HDMI 1.3a Specification. It uses only one bidirectional line for transmission and reception.

All transactions on the CEC line consist of an initiator and one or more followers. The initiator is responsible for sending the message structure and the data. The follower is the recipient of any data and is responsible for setting any acknowledgement bits.

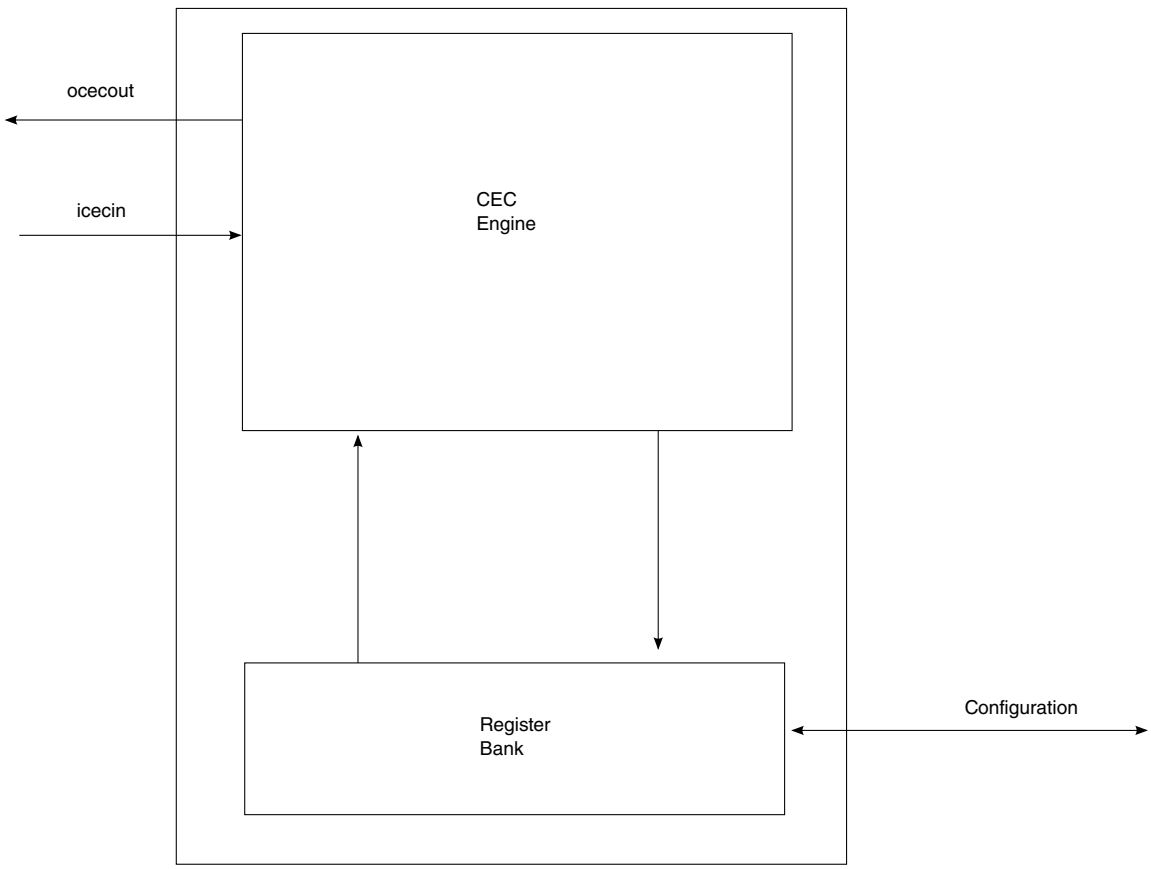


Figure 34-16. CEC Engine Simplified Block Diagram

There are two operation modes for a CEC controller.

- Initiator Mode
- In this mode, the CEC controller sends messages out and waits for a follower to feedback. The CEC controller works in this mode when it starts to send a frame. After the transmission is done, it automatically returns to the follower mode (no software control involved).

- Follower Mode
- In this mode, the CEC controller receives messages and feeds back the initiator with appropriate signals. The CEC controller always works in the follower mode whenever it is not transmitting any data.

For correct CEC controller interface operation, initial reset is required in order to set internal registers to a known state. After this reset, the interface is in an IDLE state, waiting for a read or write request coming from the register configuration.

A specific CEC API is provided that implements all necessary low-level register configuration to send and receive CEC messages. For more information, see the CEC API documentation.

The CEC engine registers base address is 0x7D00. For more information about these registers, see HDMI Memory Map/Register Definition.

For more information about CEC, see *Consumer Electronics Control (CEC) Application Note*.

34.5 HDMI Memory Map/Register Definition

All registers are addressable on 32-bit boundaries; each unused bit or address location is reserved for future use and read back as 0.

.

HDMI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_0000	Design Identification Register (HDMI_DESIGN_ID)	8	R	00h	34.5.1/1688
12_0001	Revision Identification Register (HDMI_REVISION_ID)	8	R	00h	34.5.2/1689
12_0002	Product Identification Register 0 (HDMI_PRODUCT_ID0)	8	R	00h	34.5.3/1690
12_0003	Product Identification Register 1 (HDMI_PRODUCT_ID1)	8	R	00h	34.5.4/1690
12_0004	Configuration Identification Register 0 (HDMI_CONFIG0_ID)	8	R	00h	34.5.5/1691
12_0005	Configuration Identification Register 1 (HDMI_CONFIG1_ID)	8	R	00h	34.5.6/1692
12_0006	Configuration Identification Register 2 (HDMI_CONFIG2_ID)	8	R	00h	34.5.7/1693
12_0007	Configuration Identification Register 3 (HDMI_CONFIG3_ID)	8	R	00h	34.5.8/1693
12_0100	Frame Composer Interrupt Status Register 0 (HDMI_IH_FC_STAT0)	8	w1c	00h	34.5.9/1694

Table continues on the next page...

HDMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_0101	Frame Composer Interrupt Status Register 1 (HDMI_IH_FC_STAT1)	8	w1c	00h	34.5.10/ 1695
12_0102	Frame Composer Interrupt Status Register 2 (HDMI_IH_FC_STAT2)	8	w1c	00h	34.5.11/ 1696
12_0103	Audio Sampler Interrupt Status Register (HDMI_IH_AS_STAT0)	8	w1c	00h	34.5.12/ 1697
12_0104	PHY Interface Interrupt Status Register (HDMI_IH_PHY_STAT0)	8	w1c	00h	34.5.13/ 1698
12_0105	E-DDC I2C Master Interrupt Status Register (HDMI_IH_I2CM_STAT0)	8	w1c	00h	34.5.14/ 1699
12_0106	CEC Interrupt Status Register (HDMI_IH_CEC_STAT0)	8	w1c	00h	34.5.15/ 1700
12_0107	Video Packetizer Interrupt Status Register (HDMI_IH_VP_STAT0)	8	w1c	00h	34.5.16/ 1701
12_0108	PHY GEN2 I2C Master Interrupt Status Register (HDMI_IH_I2CMPHY_STAT0)	8	w1c	00h	34.5.17/ 1702
12_0109	AHB Audio DMA Interrupt Status Register (HDMI_IH_AHBDMAAUD_STAT0)	8	w1c	00h	34.5.18/ 1702
12_0180	Frame Composer Interrupt Mute Control Register 0 (HDMI_IH_MUTE_FC_STAT0)	8	R/W	00h	34.5.19/ 1704
12_0181	Frame Composer Interrupt Mute Control Register 1 (HDMI_IH_MUTE_FC_STAT1)	8	R/W	00h	34.5.20/ 1705
12_0182	Frame Composer Interrupt Mute Control Register 2 (HDMI_IH_MUTE_FC_STAT2)	8	R/W	00h	34.5.21/ 1706
12_0183	Audio Sampler Interrupt Mute Control Register 0 (HDMI_IH_MUTE_AS_STAT0)	8	R/W	00h	34.5.22/ 1706
12_0184	PHY Interface Interrupt Mute Control Register (HDMI_IH_MUTE_PHY_STAT0)	8	R/W	00h	34.5.23/ 1707
12_0185	E-DDC I2C Master Interrupt Mute Control Register (HDMI_IH_MUTE_I2CM_STAT0)	8	R/W	00h	34.5.24/ 1708
12_0186	CEC Interrupt Mute Control Register (HDMI_IH_MUTE_CEC_STAT0)	8	R/W	00h	34.5.25/ 1708
12_0187	Video Packetizer Interrupt Mute Control Register (HDMI_IH_MUTE_VP_STAT0)	8	R/W	00h	34.5.26/ 1709
12_0188	PHY GEN 2 I2C Master Interrupt Mute Control Register (HDMI_IH_MUTE_I2CMPHY_STAT0)	8	R/W	00h	34.5.27/ 1710
12_0189	AHB Audio DMA Interrupt Mute Control Register (HDMI_IH_MUTE_AHBDMAAUD_STAT0)	8	R/W	00h	34.5.28/ 1711
12_01FF	Global Interrupt Mute Control Register (HDMI_IH_MUTE)	8	R/W	03h	34.5.29/ 1712
12_0200	Video Input Mapping and Internal Data Enable Configuration Register (HDMI_TX_INVID0)	8	R/W	01h	34.5.30/ 1712
12_0201	Video Input Stuffing Enable Register (HDMI_TX_INSTUFFING)	8	R/W	00h	34.5.31/ 1713

Table continues on the next page...

HDMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_0202	Video Input GY Data Channel Stuffing Register 0 (HDMI_TX_GYDATA0)	8	R/W	00h	34.5.32/ 1714
12_0203	Video Input GY Data Channel Stuffing Register 1 (HDMI_TX_GYDATA1)	8	R/W	00h	34.5.33/ 1715
12_0204	Video Input RCR Data Channel Stuffing Register 0 (HDMI_TX_RCRDATA0)	8	R/W	00h	34.5.34/ 1715
12_0205	Video Input RCR Data Channel Stuffing Register 1 (HDMI_TX_RCRDATA1)	8	R/W	00h	34.5.35/ 1716
12_0206	Video Input RCB Data Channel Stuffing Register 0 (HDMI_TX_BCBDATA0)	8	R/W	00h	34.5.36/ 1716
12_0207	Video Input RCB Data Channel Stuffing Register 1 (HDMI_TX_BCBDATA1)	8	R/W	00h	34.5.37/ 1717
12_0800	Video Packetizer Packing Phase Status Register (HDMI_VP_STATUS)	8	R	00h	34.5.38/ 1717
12_0801	Video Packetizer Pixel Repetition and Color Depth Register (HDMI_VP_PR_CD)	8	R/W	00h	34.5.39/ 1718
12_0802	Video Packetizer Stuffing and Default Packing Phase Register (HDMI_VP_STUFF)	8	R/W	00h	34.5.40/ 1719
12_0803	Video Packetizer YCC422 Remapping Register (HDMI_VP_REMAP)	8	R/W	00h	34.5.41/ 1720
12_0804	Video Packetizer Output, Bypass, and Enable Configuration Register (HDMI_VP_CONF)	8	R/W	46h	34.5.42/ 1721
12_0807	Video Packetizer Interrupt Mask Register (HDMI_VP_MASK)	8	R/W	00h	34.5.43/ 1721
12_1000	Frame Composer Input Video Configuration and HDCP Keepout Register (HDMI_FC_INVIDCONF)	8	R/W	70h	34.5.44/ 1722
12_1001	Frame Composer Input Video HActive Pixels Register 0 (HDMI_FC_INHACTIV0)	8	R/W	00h	34.5.45/ 1724
12_1002	Frame Composer Input Video HActive Pixels Register 1 (HDMI_FC_INHACTIV1)	8	R/W	00h	34.5.46/ 1724
12_1003	Frame Composer Input Video HBlank Pixels Register 0 (HDMI_FC_INHBLANK0)	8	R/W	00h	34.5.47/ 1725
12_1004	Frame Composer Input Video HBlank Pixels Register 1 (HDMI_FC_INHBLANK1)	8	R/W	00h	34.5.48/ 1725
12_1005	Frame Composer Input Video VActive Pixels Register 0 (HDMI_FC_INVACTIV0)	8	R/W	00h	34.5.49/ 1726
12_1006	Frame Composer Input Video VActive Pixels Register 1 (HDMI_FC_INVACTIV1)	8	R/W	00h	34.5.50/ 1727
12_1007	Frame Composer Input Video VBlank Pixels Register (HDMI_FC_INVBLANK)	8	R/W	00h	34.5.51/ 1727
12_1008	Frame Composer Input Video HSync Front Porch Register 0 (HDMI_FC_HSYNCINDELAY0)	8	R/W	00h	34.5.52/ 1728
12_1009	Frame Composer Input Video HSync Front Porch Register 1 (HDMI_FC_HSYNCINDELAY1)	8	R/W	00h	34.5.53/ 1728

Table continues on the next page...

HDMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_100A	Frame Composer Input Video HSync Width Register 0 (HDMI_FC_HSYNCINWIDTH0)	8	R/W	00h	34.5.54/ 1729
12_100B	Frame Composer Input Video HSync Width Register 1 (HDMI_FC_HSYNCINWIDTH1)	8	R/W	00h	34.5.55/ 1730
12_100C	Frame Composer Input Video VSync Front Porch Register (HDMI_FC_VSYNCINDELAY)	8	R/W	00h	34.5.56/ 1730
12_100D	Frame Composer Input Video VSync Width Register (HDMI_FC_VSYNCINWIDTH)	8	R/W	00h	34.5.57/ 1731
12_100E	Frame Composer Input Video Refresh Rate Register 0 (HDMI_FC_INFREQ0)	8	R/W	00h	34.5.58/ 1731
12_100F	Frame Composer Input Video Refresh Rate Register 1 (HDMI_FC_INFREQ1)	8	R/W	00h	34.5.59/ 1732
12_1010	Frame Composer Input Video Refresh Rate Register 2 (HDMI_FC_INFREQ2)	8	R/W	00h	34.5.60/ 1732
12_1011	Frame Composer Control Period Duration Register (HDMI_FC_CTRLDUR)	8	R/W	00h	34.5.61/ 1733
12_1012	Frame Composer Extended Control Period Duration Register (HDMI_FC_EXCTRLDUR)	8	R/W	00h	34.5.62/ 1734
12_1013	Frame Composer Extended Control Period Maximum Spacing Register (HDMI_FC_EXCTRLSPAC)	8	R/W	00h	34.5.63/ 1734
12_1014	Frame Composer Channel 0 Non-Preamble Data Register (HDMI_FC_CH0PREAM)	8	R/W	00h	34.5.64/ 1735
12_1015	Frame Composer Channel 1 Non-Preamble Data Register (HDMI_FC_CH1PREAM)	8	R/W	00h	34.5.65/ 1735
12_1016	Frame Composer Channel 2 Non-Preamble Data Register (HDMI_FC_CH2PREAM)	8	R/W	00h	34.5.66/ 1736
12_1017	Frame Composer AVI Configuration Register 3 (HDMI_FC_AVICONF3)	8	R/W	00h	34.5.67/ 1736
12_1018	Frame Composer GCP Packet Configuration Register (HDMI_FC_GCP)	8	R/W	00h	34.5.68/ 1737
12_1019	Frame Composer AVI Packet Configuration Register 0 (HDMI_FC_AVICONF0)	8	R/W	00h	34.5.69/ 1737
12_101A	Frame Composer AVI Packet Configuration Register 1 (HDMI_FC_AVICONF1)	8	R/W	00h	34.5.70/ 1739
12_101B	FC_AVICONFFrame Composer AVI Packet Configuration Register 2 (HDMI_FC_AVICONF2)	8	R/W	00h	34.5.71/ 1739
12_101C	Frame Composer AVI Packet VIC Register (HDMI_FC_AVIVID)	8	R/W	00h	34.5.72/ 1740
12_101D	Frame Composer AVI Packet End of Top Bar Register 0 (HDMI_FC_AVIETB0)	8	R/W	00h	34.5.73/ 1740
12_101E	Frame Composer AVI Packet End of Top Bar Register 1 (HDMI_FC_AVIETB1)	8	R/W	00h	34.5.74/ 1741
12_101F	Frame Composer AVI Packet Start of Bottom Bar Register 0 (HDMI_FC_AVISBB0)	8	R/W	00h	34.5.75/ 1741

Table continues on the next page...

HDMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_1020	Frame Composer AVI Packet Start of Bottom Bar Register 1 (HDMI_FC_AVISBB1)	8	R/W	00h	34.5.76/ 1742
12_1021	Frame Composer AVI Packet End of Left Bar Register 0 (HDMI_FC_AVIELB0)	8	R/W	00h	34.5.77/ 1742
12_1022	Frame Composer AVI Packet End of Left Bar Register 1 (HDMI_FC_AVIELB1)	8	R/W	00h	34.5.78/ 1743
12_1023	Frame Composer AVI Packet Start of Right Bar Register 0 (HDMI_FC_AVISRB0)	8	R/W	00h	34.5.79/ 1743
12_1024	Frame Composer AVI Packet Start of Right Bar Register 1 (HDMI_FC_AVISRB1)	8	R/W	00h	34.5.80/ 1744
12_1025	Frame Composer AUD Packet Configuration Register 0 (HDMI_FC_AUDICONF0)	8	R/W	00h	34.5.81/ 1744
12_1026	Frame Composer AUD Packet Configuration Register 1 (HDMI_FC_AUDICONF1)	8	R/W	00h	34.5.82/ 1745
12_1027	Frame Composer AUD Packet Configuration Register 2 (HDMI_FC_AUDICONF2)	8	R/W	00h	34.5.83/ 1746
12_1028	Frame Composer AUD Packet Configuration Register 3 (HDMI_FC_AUDICONF3)	8	R/W	00h	34.5.84/ 1746
12_1029	Frame Composer VSI Packet Data IEEE Register 0 (HDMI_FC_VSDIEEEID0)	8	R/W	00h	34.5.85/ 1746
12_102A	Frame Composer VSI Packet Data Size Register (HDMI_FC_VSDSIZE)	8	R/W	1Bh	34.5.86/ 1747
12_1030	Frame Composer VSI Packet Data IEEE Register 1 (HDMI_FC_VSDIEEEID1)	8	R/W	00h	34.5.87/ 1748
12_1031	Frame Composer VSI Packet Data IEEE Register 2 (HDMI_FC_VSDIEEEID2)	8	R/W	00h	34.5.88/ 1748
12_1032	Frame Composer VSI Packet Data IEEE Register 0 (HDMI_FC_VSDPAYLOAD0)	8	R/W	00h	34.5.89/ 1749
12_1033	Frame Composer VSI Packet Data IEEE Register 1 (HDMI_FC_VSDPAYLOAD1)	8	R/W	00h	34.5.90/ 1749
12_1034	Frame Composer VSI Packet Data IEEE Register 2 (HDMI_FC_VSDPAYLOAD2)	8	R/W	00h	34.5.91/ 1750
12_1035	Frame Composer VSI Packet Data IEEE Register 3 (HDMI_FC_VSDPAYLOAD3)	8	R/W	00h	34.5.92/ 1750
12_1036	Frame Composer VSI Packet Data IEEE Register 4 (HDMI_FC_VSDPAYLOAD4)	8	R/W	00h	34.5.93/ 1751
12_1037	Frame Composer VSI Packet Data IEEE Register 5 (HDMI_FC_VSDPAYLOAD5)	8	R/W	00h	34.5.94/ 1751
12_1038	Frame Composer VSI Packet Data IEEE Register 6 (HDMI_FC_VSDPAYLOAD6)	8	R/W	00h	34.5.95/ 1752
12_1039	Frame Composer VSI Packet Data IEEE Register 7 (HDMI_FC_VSDPAYLOAD7)	8	R/W	00h	34.5.96/ 1752
12_103A	Frame Composer VSI Packet Data IEEE Register 8 (HDMI_FC_VSDPAYLOAD8)	8	R/W	00h	34.5.97/ 1753

Table continues on the next page...

HDMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_103B	Frame Composer VSI Packet Data IEEE Register 9 (HDMI_FC_VSDPAYLOAD9)	8	R/W	00h	34.5.98/ 1753
12_103C	Frame Composer VSI Packet Data IEEE Register 10 (HDMI_FC_VSDPAYLOAD10)	8	R/W	00h	34.5.99/ 1754
12_103D	Frame Composer VSI Packet Data IEEE Register 11 (HDMI_FC_VSDPAYLOAD11)	8	R/W	00h	34.5.100/ 1754
12_103E	Frame Composer VSI Packet Data IEEE Register 12 (HDMI_FC_VSDPAYLOAD12)	8	R/W	00h	34.5.101/ 1755
12_103F	Frame Composer VSI Packet Data IEEE Register 13 (HDMI_FC_VSDPAYLOAD13)	8	R/W	00h	34.5.102/ 1755
12_1040	Frame Composer VSI Packet Data IEEE Register 14 (HDMI_FC_VSDPAYLOAD14)	8	R/W	00h	34.5.103/ 1756
12_1041	Frame Composer VSI Packet Data IEEE Register 15 (HDMI_FC_VSDPAYLOAD15)	8	R/W	00h	34.5.104/ 1756
12_1042	Frame Composer VSI Packet Data IEEE Register 16 (HDMI_FC_VSDPAYLOAD16)	8	R/W	00h	34.5.105/ 1757
12_1043	Frame Composer VSI Packet Data IEEE Register 17 (HDMI_FC_VSDPAYLOAD17)	8	R/W	00h	34.5.106/ 1757
12_1044	Frame Composer VSI Packet Data IEEE Register 18 (HDMI_FC_VSDPAYLOAD18)	8	R/W	00h	34.5.107/ 1758
12_1045	Frame Composer VSI Packet Data IEEE Register 19 (HDMI_FC_VSDPAYLOAD19)	8	R/W	00h	34.5.108/ 1758
12_1046	Frame Composer VSI Packet Data IEEE Register 20 (HDMI_FC_VSDPAYLOAD20)	8	R/W	00h	34.5.109/ 1759
12_1047	Frame Composer VSI Packet Data IEEE Register 21 (HDMI_FC_VSDPAYLOAD21)	8	R/W	00h	34.5.110/ 1759
12_1048	Frame Composer VSI Packet Data IEEE Register 22 (HDMI_FC_VSDPAYLOAD22)	8	R/W	00h	34.5.111/ 1760
12_1049	Frame Composer VSI Packet Data IEEE Register 23 (HDMI_FC_VSDPAYLOAD23)	8	R/W	00h	34.5.112/ 1760
12_104A	Frame Composer SPD Packet Data Vendor Name Register 0 (HDMI_FC_SPDVENDORNAME0)	8	R/W	00h	34.5.113/ 1761
12_1052	Frame Composer SPD Packet Data Product Name Register 0 (HDMI_FC_SPDPRODUCTNAME0)	8	R/W	00h	34.5.114/ 1761
12_1062	Frame Composer SPD Packet Data Source Product Descriptor Register (HDMI_FC_SPDDEVICEINF)	8	R/W	00h	34.5.115/ 1762
12_1063	Frame Composer Audio Sample Flat and Layout Configuration Register (HDMI_FC_AUDSCONF)	8	R/W	00h	34.5.116/ 1762
12_1064	Frame Composer Audio Packet Sample Present Status Register (HDMI_FC_AUDSSTAT)	8	R	00h	34.5.117/ 1763
12_1073	Frame Composer Number of High Priority Packets Attended Configuration Register (HDMI_FC_CTRLQHIGH)	8	R/W	0Fh	34.5.118/ 1764
12_1074	Frame Composer Number of Low Priority Packets Attended Configuration Register (HDMI_FC_CTRLQLOW)	8	R/W	03h	34.5.119/ 1764

Table continues on the next page...

HDMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_1075	Frame Composer ACP Packet Type Configuration Register 0 (HDMI_FC_ACP0)	8	R/W	00h	34.5.120/ 1765
12_1091	Frame Composer ACP Packet Type Configuration Register 1 (HDMI_FC_ACP1)	8	R/W	00h	34.5.121/ 1765
12_1092	FC_ISCR1_Frame Composer Packet Status, Valid, and Continue Configuration Register (HDMI_FC_ISCR1_0)	8	R/W	00h	34.5.122/ 1766
12_1093	Frame Composer ISCR1 Packet Body Register 1 (HDMI_FC_ISCR1_1)	8	R/W	00h	34.5.123/ 1766
12_10A3	Frame Composer ISCR2 Packet Body Register 0 (HDMI_FC_ISCR2_0)	8	R/W	00h	34.5.124/ 1767
12_10B3	Frame Composer Data Island Auto Packet Scheduling Register 0 (HDMI_FC_DATAUTO0)	8	R/W	00h	34.5.125/ 1768
12_10B4	Frame Composer Data Island Auto Packet Scheduling Register 1 (HDMI_FC_DATAUTO1)	8	R/W	00h	34.5.126/ 1768
12_10B5	Frame Composer Data Island Auto Packet Scheduling Register 2 (HDMI_FC_DATAUTO2)	8	R/W	00h	34.5.127/ 1769
12_10B6	Frame Composer Data Island Manual Packet Request Register (HDMI_FC_DATMAN)	8	W	00h	34.5.128/ 1770
12_10B7	Frame Composer Data Island Auto Packet Scheduling Register 3 (HDMI_FC_DATAUTO3)	8	R/W	0Fh	34.5.129/ 1770
12_10B8	Frame Composer Round Robin ACR Packet Insertion Register 0 (HDMI_FC_RDRB0)	8	R/W	00h	34.5.130/ 1771
12_10B9	Frame Composer Round Robin ACR Packet Insertion Register 1 (HDMI_FC_RDRB1)	8	R/W	00h	34.5.131/ 1772
12_10BA	Frame Composer Round Robin ACR Packet Insertion Register 2 (HDMI_FC_RDRB2)	8	R/W	00h	34.5.132/ 1772
12_10BB	Frame Composer Round Robin ACR Packet Insertion Register 3 (HDMI_FC_RDRB3)	8	R/W	00h	34.5.133/ 1773
12_10BC	Frame Composer Round Robin ACR Packet Insertion Register 4 (HDMI_FC_RDRB4)	8	R/W	00h	34.5.134/ 1773
12_10BD	Frame Composer Round Robin ACR Packet Insertion Register 5 (HDMI_FC_RDRB5)	8	R/W	00h	34.5.135/ 1774
12_10BE	Frame Composer Round Robin ACR Packet Insertion Register 6 (HDMI_FC_RDRB6)	8	R/W	00h	34.5.136/ 1775
12_10BF	Frame Composer Round Robin ACR Packet Insertion Register 7 (HDMI_FC_RDRB7)	8	R/W	00h	34.5.137/ 1775
12_10D2	Frame Composer Packet Interrupt Mask Register 0 (HDMI_FC_MASK0)	8	R/W	25h	34.5.138/ 1776
12_10D6	Frame Composer Packet Interrupt Mask Register 1 (HDMI_FC_MASK1)	8	R/W	00h	34.5.139/ 1777
12_10E0	Frame Composer Pixel Repetition Configuration Register (HDMI_FC_PRCONF)	8	R/W	10h	34.5.140/ 1777
12_1100	Frame Composer GMD Packet Status Register (HDMI_FC_GMD_STAT)	8	R	00h	34.5.141/ 1779

Table continues on the next page...

HDMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_1101	Frame Composer GMD Packet Enable Register (HDMI_FC_GMD_EN)	8	R/W	00h	34.5.142/ 1780
12_1102	Frame Composer GMD Packet Update Register (HDMI_FC_GMD_UP)	8	W	00h	34.5.143/ 1780
12_1103	Frame Composer GMD Packet Schedule Configuration Register (HDMI_FC_GMD_CONF)	8	R/W	10h	34.5.144/ 1781
12_1104	Frame Composer GMD Packet Profile and Gamut Sequence Configuration Register (HDMI_FC_GMD_HB)	8	R/W	00h	34.5.145/ 1782
12_1105	Frame Composer GMD Packet Body Register 0 (HDMI_FC_GMD_PB0)	8	R/W	00h	34.5.146/ 1782
12_1106	Frame Composer GMD Packet Body Register 1 (HDMI_FC_GMD_PB1)	8	R/W	00h	34.5.147/ 1783
12_1107	Frame Composer GMD Packet Body Register 2 (HDMI_FC_GMD_PB2)	8	R/W	00h	34.5.148/ 1783
12_1108	Frame Composer GMD Packet Body Register 3 (HDMI_FC_GMD_PB3)	8	R/W	00h	34.5.149/ 1784
12_1109	Frame Composer GMD Packet Body Register 4 (HDMI_FC_GMD_PB4)	8	R/W	00h	34.5.150/ 1784
12_110A	Frame Composer GMD Packet Body Register 5 (HDMI_FC_GMD_PB5)	8	R/W	00h	34.5.151/ 1785
12_110B	Frame Composer GMD Packet Body Register 6 (HDMI_FC_GMD_PB6)	8	R/W	00h	34.5.152/ 1785
12_110C	Frame Composer GMD Packet Body Register 7 (HDMI_FC_GMD_PB7)	8	R/W	00h	34.5.153/ 1786
12_110D	Frame Composer GMD Packet Body Register 8 (HDMI_FC_GMD_PB8)	8	R/W	00h	34.5.154/ 1786
12_110E	Frame Composer GMD Packet Body Register 9 (HDMI_FC_GMD_PB9)	8	R/W	00h	34.5.155/ 1787
12_110F	Frame Composer GMD Packet Body Register 10 (HDMI_FC_GMD_PB10)	8	R/W	00h	34.5.156/ 1787
12_1110	Frame Composer GMD Packet Body Register 11 (HDMI_FC_GMD_PB11)	8	R/W	00h	34.5.157/ 1788
12_1111	Frame Composer GMD Packet Body Register 12 (HDMI_FC_GMD_PB12)	8	R/W	00h	34.5.158/ 1788
12_1112	Frame Composer GMD Packet Body Register 13 (HDMI_FC_GMD_PB13)	8	R/W	00h	34.5.159/ 1789
12_1113	Frame Composer GMD Packet Body Register 14 (HDMI_FC_GMD_PB14)	8	R/W	00h	34.5.160/ 1789
12_1114	Frame Composer GMD Packet Body Register 15 (HDMI_FC_GMD_PB15)	8	R/W	00h	34.5.161/ 1790
12_1115	Frame Composer GMD Packet Body Register 16 (HDMI_FC_GMD_PB16)	8	R/W	00h	34.5.162/ 1790
12_1116	Frame Composer GMD Packet Body Register 17 (HDMI_FC_GMD_PB17)	8	R/W	00h	34.5.163/ 1791

Table continues on the next page...

HDMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_1117	Frame Composer GMD Packet Body Register 18 (HDMI_FC_GMD_PB18)	8	R/W	00h	34.5.164/ 1791
12_1118	Frame Composer GMD Packet Body Register 19 (HDMI_FC_GMD_PB19)	8	R/W	00h	34.5.165/ 1792
12_1119	Frame Composer GMD Packet Body Register 20 (HDMI_FC_GMD_PB20)	8	R/W	00h	34.5.166/ 1792
12_111A	Frame Composer GMD Packet Body Register 21 (HDMI_FC_GMD_PB21)	8	R/W	00h	34.5.167/ 1793
12_111B	Frame Composer GMD Packet Body Register 22 (HDMI_FC_GMD_PB22)	8	R/W	00h	34.5.168/ 1793
12_111C	Frame Composer GMD Packet Body Register 23 (HDMI_FC_GMD_PB23)	8	R/W	00h	34.5.169/ 1794
12_111D	Frame Composer GMD Packet Body Register 24 (HDMI_FC_GMD_PB24)	8	R/W	00h	34.5.170/ 1794
12_111E	Frame Composer GMD Packet Body Register 25 (HDMI_FC_GMD_PB25)	8	R/W	00h	34.5.171/ 1795
12_111F	Frame Composer GMD Packet Body Register 26 (HDMI_FC_GMD_PB26)	8	R/W	00h	34.5.172/ 1795
12_1120	Frame Composer GMD Packet Body Register 27 (HDMI_FC_GMD_PB27)	8	R/W	00h	34.5.173/ 1796
12_1200	Frame Composer Video/Audio Force Enable Register (HDMI_FC_DBGFORCE)	8	R/W	00h	34.5.174/ 1796
12_1201	Frame Composer Audio Channel 0 Register 0 (HDMI_FC_DBGAUD0CH0)	8	R/W	00h	34.5.175/ 1797
12_1202	Frame Composer Audio Channel 0 Register 1 (HDMI_FC_DBGAUD1CH0)	8	R/W	00h	34.5.176/ 1798
12_1203	Frame Composer Audio Channel 0 Register 2 (HDMI_FC_DBGAUD2CH0)	8	R/W	00h	34.5.177/ 1798
12_1204	Frame Composer Audio Channel 1 Register 0 (HDMI_FC_DBGAUD0CH1)	8	R/W	00h	34.5.178/ 1799
12_1205	Frame Composer Audio Channel 1 Register 1 (HDMI_FC_DBGAUD1CH1)	8	R/W	00h	34.5.179/ 1799
12_1206	Frame Composer Audio Channel 1 Register 2 (HDMI_FC_DBGAUD2CH1)	8	R/W	00h	34.5.180/ 1800
12_1207	Frame Composer Debug Audio Channel 2 Register 0 (HDMI_FC_DBGAUD0CH2)	8	R/W	00h	34.5.181/ 1800
12_1208	Frame Composer Debug Audio Channel 2 Register 1 (HDMI_FC_DBGAUD1CH2)	8	R/W	00h	34.5.182/ 1801
12_1209	Frame Composer Audio Channel 2 Register 2 (HDMI_FC_DBGAUD2CH2)	8	R/W	00h	34.5.183/ 1801
12_120A	Frame Composer Audio Channel 3 Register 0 (HDMI_FC_DBGAUD0CH3)	8	R/W	00h	34.5.184/ 1802
12_120B	Frame Composer Audio Channel 3 Register 1 (HDMI_FC_DBGAUD1CH3)	8	R/W	00h	34.5.185/ 1802

Table continues on the next page...

HDMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_120C	Frame Composer Audio Channel 3 Register 2 (HDMI_FC_DBGAUD2CH3)	8	R/W	00h	34.5.186/ 1803
12_120D	Frame Composer Audio Channel 4 Register 0 (HDMI_FC_DBGAUD0CH4)	8	R/W	00h	34.5.187/ 1803
12_120E	Frame Composer Audio Channel 4 Register 1 (HDMI_FC_DBGAUD1CH4)	8	R/W	00h	34.5.188/ 1804
12_120F	Frame Composer Audio Channel 4 Register 2 (HDMI_FC_DBGAUD2CH4)	8	R/W	00h	34.5.189/ 1804
12_1210	Frame Composer Audio Channel 5 Register 0 (HDMI_FC_DBGAUD0CH5)	8	R/W	00h	34.5.190/ 1805
12_1211	Frame Composer Audio Channel 5 Register 1 (HDMI_FC_DBGAUD1CH5)	8	R/W	00h	34.5.191/ 1805
12_1212	Frame Composer Audio Channel 5 Register 2 (HDMI_FC_DBGAUD2CH5)	8	R/W	00h	34.5.192/ 1806
12_1213	Frame Composer Audio Channel 6 Register 0 (HDMI_FC_DBGAUD0CH6)	8	R/W	00h	34.5.193/ 1806
12_1214	Frame Composer Audio Channel 6 Register 1 (HDMI_FC_DBGAUD1CH6)	8	R/W	00h	34.5.194/ 1807
12_1215	Frame Composer Audio Channel 6 Register 2 (HDMI_FC_DBGAUD2CH6)	8	R/W	00h	34.5.195/ 1807
12_1216	Frame Composer Audio Channel 7 Register 1 (HDMI_FC_DBGAUD0CH7)	8	R/W	00h	34.5.196/ 1808
12_1217	Frame Composer Audio Channel 7 Register 0 (HDMI_FC_DBGAUD1CH7)	8	R/W	00h	34.5.197/ 1808
12_1218	Frame Composer Audio Channel 7 Register 2 (HDMI_FC_DBGAUD2CH7)	8	R/W	00h	34.5.198/ 1809
12_1219	Frame Composer TMDS Channel 0 Register (HDMI_FC_DBGTMDS0)	8	R/W	00h	34.5.199/ 1809
12_121A	Frame Composer TMDS Channel 1 Register (HDMI_FC_DBGTMDS1)	8	R/W	00h	34.5.200/ 1810
12_121B	Frame Composer TMDS Channel 2 Register (HDMI_FC_DBGTMDS2)	8	R/W	00h	34.5.201/ 1810
12_3000	PHY Configuration Register (HDMI_PHY_CONF0)	8	R/W	06h	34.5.202/ 1811
12_3001	PHY Test Interface Register 0 (HDMI_PHY_TST0)	8	R/W	00h	34.5.203/ 1812
12_3002	PHY Test Interface Register 1 (HDMI_PHY_TST1)	8	R/W	00h	34.5.204/ 1812
12_3003	PHY Test Interface Register 2 (HDMI_PHY_TST2)	8	R	00h	34.5.205/ 1813
12_3004	PHY RXSENSE, PLL lock, and HPD Status Register (HDMI_PHY_STAT0)	8	R	00h	34.5.206/ 1813
12_3005	PHY RXSENSE, PLL lock, and HPD Interrupt Register (HDMI_PHY_INT0)	8	R	00h	34.5.207/ 1814

Table continues on the next page...

HDMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_3006	PHY RXSENSE, PLL lock, and HPD Mask Register (HDMI_PHY_MASK0)	8	R/W	00h	34.5.208/ 1815
12_3007	PHY RXSENSE, PLL lock and HPD Polarity Register (HDMI_PHY_POL0)	8	R/W	F3h	34.5.209/ 1816
12_3020	PHY I2C Slave Address Configuration Register (HDMI_PHY_I2CM_SLAVE_ADDR)	8	R/W	00h	34.5.210/ 1817
12_3021	PHY I2C Address Configuration Register (HDMI_PHY_I2CM_ADDRESS_ADDR)	8	R/W	00h	34.5.211/ 1817
12_3022	PHY I2C Data Write Register 1 (HDMI_PHY_I2CM_DATAO_1_ADDR)	8	R/W	00h	34.5.212/ 1818
12_3023	PHY I2C Data Write Register 0 (HDMI_PHY_I2CM_DATAO_0_ADDR)	8	R/W	00h	34.5.213/ 1819
12_3024	PHY I2C Data Read Register 1 (HDMI_PHY_I2CM_DATAI_1_ADDR)	8	R	00h	34.5.214/ 1819
12_3025	PHY I2C Data Read Register 0 (HDMI_PHY_I2CM_DATAI_0_ADDR)	8	R/W	00h	34.5.215/ 1820
12_3026	PHY I2C Read/Write Operation (HDMI_PHY_I2CM_OPERATION_ADDR)	8	W	00h	34.5.216/ 1820
12_3027	PHY I2C Done Interrupt Register (HDMI_PHY_I2CM_INT_ADDR)	8	R/W	08h	34.5.217/ 1821
12_3028	PHY I2C Done Interrupt Register (HDMI_PHY_I2CM_CTLINT_ADDR)	8	R/W	88h	34.5.218/ 1822
12_3029	PHY I2C Speed Control Register (HDMI_PHY_I2CM_DIV_ADDR)	8	R/W	0Bh	34.5.219/ 1823
12_302A	PHY I2C Software Reset Register (HDMI_PHY_I2CM_SOFTRSTZ_ADDR)	8	R/W	01h	34.5.220/ 1823
12_302B	PHY I2C Slow Speed SCL High Level Control Register 1 (HDMI_PHY_I2CM_SS_SCL_HCNT_1_ADDR)	8	R/W	00h	34.5.221/ 1824
12_302C	PHY I2C Slow Speed SCL High Level Control Register 0 (HDMI_PHY_I2CM_SS_SCL_HCNT_0_ADDR)	8	R/W	6Ch	34.5.222/ 1825
12_302D	PHY I2C Slow Speed SCL Low Level Control Register 1 (HDMI_PHY_I2CM_SS_SCL_LCNT_1_ADDR)	8	R/W	00h	34.5.223/ 1825
12_302E	PHY I2C Slow Speed SCL Low Level Control Register 0 (HDMI_PHY_I2CM_SS_SCL_LCNT_0_ADDR)	8	R/W	7Fh	34.5.224/ 1826
12_302F	PHY I2C Fast Speed SCL High Level Control Register 1 (HDMI_PHY_I2CM_FS_SCL_HCNT_1_ADDR)	8	R/W	00h	34.5.225/ 1826
12_3030	PHY I2C Fast Speed SCL High Level Control Register 0 (HDMI_PHY_I2CM_FS_SCL_HCNT_0_ADDR)	8	R/W	11h	34.5.226/ 1827
12_3031	PHY I2C Fast Speed SCL Low Level Control Register 1 (HDMI_PHY_I2CM_FS_SCL_LCNT_1_ADDR)	8	R/W	00h	34.5.227/ 1827
12_3032	PHY I2C Fast Speed SCL Low Level Control Register 0 (HDMI_PHY_I2CM_FS_SCL_LCNT_0_ADDR)	8	R/W	24h	34.5.228/ 1828
12_3200	Audio Clock Regenerator N Value Register 1 (HDMI_AUD_N1)	8	R/W	00h	34.5.229/ 1828

Table continues on the next page...

HDMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_3201	Audio Clock Regenerator N Value Register 2 (HDMI_AUD_N2)	8	R/W	00h	34.5.230/ 1829
12_3202	Audio Clock Regenerator N Value Register 3 (HDMI_AUD_N3)	8	R/W	00h	34.5.231/ 1829
12_3203	AUD_CTS1 (HDMI_AUD_CTS1)	8	R/W	00h	34.5.232/ 1830
12_3204	AUD_CTS2 (HDMI_AUD_CTS2)	8	R/W	00h	34.5.233/ 1830
12_3205	AUD_CTS3 (HDMI_AUD_CTS3)	8	R/W	00h	34.5.234/ 1831
12_3600	Audio DMA Start Register (HDMI_AHB_DMA_CONF0)	8	R/W	00h	34.5.235/ 1831
12_3601	AHB_DMA_START (HDMI_AHB_DMA_START)	8	R/W	00h	34.5.236/ 1832
12_3602	Audio DMA Stop Register (HDMI_AHB_DMA_STOP)	8	R/W	00h	34.5.237/ 1833
12_3603	Audio DMA FIFO Threshold Register (HDMI_AHB_DMA_THRSLD)	8	R/W	0100h	34.5.238/ 1834
12_3604	Audio DMA Start Address Register 0 (HDMI_AHB_DMA_STRADDR0)	8	R/W	00h	34.5.239/ 1834
12_3605	Audio DMA Start Address Register 1 (HDMI_AHB_DMA_STRADDR1)	8	R/W	00h	34.5.240/ 1835
12_3606	Audio DMA Start Address Register 2 (HDMI_AHB_DMA_STRADDR2)	8	R/W	00h	34.5.241/ 1835
12_3607	Audio DMA Start Address Register 3 (HDMI_AHB_DMA_STRADDR3)	8	R/W	00h	34.5.242/ 1836
12_3608	Audio DMA Stop Address Register 0 (HDMI_AHB_DMA_STPADDR0)	8	R/W	00h	34.5.243/ 1836
12_3609	Audio DMA Stop Address Register 1 (HDMI_AHB_DMA_STPADDR1)	8	R/W	00h	34.5.244/ 1837
12_360A	Audio DMA Stop Address Register 2 (HDMI_AHB_DMA_STPADDR2)	8	R/W	00h	34.5.245/ 1837
12_360B	Audio DMA Stop Address Register 3 (HDMI_AHB_DMA_STPADDR3)	8	R/W	00h	34.5.246/ 1838
12_360C	Audio DMA Burst Start Address Register 0 (HDMI_AHB_DMA_BSTADDR0)	8	R	00h	34.5.247/ 1838
12_360D	Audio DMA Burst Start Address Register 1 (HDMI_AHB_DMA_BSTADDR1)	8	R	00h	34.5.248/ 1839
12_360E	Audio DMA Burst Start Address Register 2 (HDMI_AHB_DMA_BSTADDR2)	8	R	00h	34.5.249/ 1839
12_360F	Audio DMA Burst Start Address Register 3 (HDMI_AHB_DMA_BSTADDR3)	8	R	00h	34.5.250/ 1839
12_3610	Audio DMA Burst Length Register 0 (HDMI_AHB_DMA_MBLENGTH0)	8	R	00h	34.5.251/ 1840

Table continues on the next page...

HDMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_3611	Audio DMA Burst Length Register 1 (HDMI_AHB_DMA_MBLENGTH1)	8	R	00h	34.5.252/ 1841
12_3612	Audio DMA Interrupt Status Register (HDMI_AHB_DMA_STAT)	8	R	00h	34.5.253/ 1841
12_3613	Audio DMA Interrupt Register (HDMI_AHB_DMA_INT)	8	R	00h	34.5.254/ 1842
12_3614	Audio DMA Mask Interrupt Register (HDMI_AHB_DMA_MASK)	8	R/W	1111_0111h	34.5.255/ 1843
12_3615	Audio DMA Polarity Interrupt Register (HDMI_AHB_DMA_POL)	8	R/W	1111_0111h	34.5.256/ 1844
12_3616	Audio DMA Channel Enable Configuration Register 1 (HDMI_AHB_DMA_CONF1)	8	R/W	00h	34.5.257/ 1845
12_3617	Audio DMA Buffer Interrupt Status Register (HDMI_AHB_DMA_BUFFSTAT)	8	R	00h	34.5.258/ 1846
12_3618	Audio DMA Buffer Interrupt Register (HDMI_AHB_DMA_BUFFINT)	8	R	00h	34.5.259/ 1847
12_3619	Audio DMA Buffer Mask Interrupt Register (HDMI_AHB_DMA_BUFFMASK)	8	R/W	11h	34.5.260/ 1848
12_361A	Audio DMA Buffer Polarity Interrupt Register (HDMI_AHB_DMA_BUFFPOL)	8	R/W	11h	34.5.261/ 1848
12_4001	Main Controller Synchronous Clock Domain Disable Register (HDMI_MC_CLKDIS)	8	R/W	00h	34.5.262/ 1849
12_4002	Main Controller Software Reset Register (HDMI_MC_SWRSTZREQ)	8	R/W	FFh	34.5.263/ 1850
12_4004	Main Controller Feed Through Control Register (HDMI_MC_FLOWCTRL)	8	R/W	00h	34.5.264/ 1851
12_4005	Main Controller PHY Reset Register (HDMI_MC_PHYRSTZ)	8	R/W	00h	34.5.265/ 1851
12_4006	Main Controller Clock Present Register (HDMI_MC_LOCKONCLOCK)	8	w1c	00h	34.5.266/ 1852
12_4007	Main Controller HEAC PHY Reset Register (HDMI_MC_HEACPHY_RST)	8	R/W	00h	34.5.267/ 1853
12_4100	Color Space Converter Interpolation and Decimation Configuration Register (HDMI_CSC_CFG)	8	R/W	00h	34.5.268/ 1853
12_4101	Color Space Converter Scale and Deep Color Configuration Register (HDMI_CSC_SCALE)	8	R/W	01h	34.5.269/ 1854
12_4102	CSC_COEF_A1_MSB (HDMI_CSC_COEF_A1_MSB)	8	R/W	20h	34.5.270/ 1855
12_4103	CSC_COEF_A1_LSB (HDMI_CSC_COEF_A1_LSB)	8	R/W	00h	34.5.271/ 1855
12_4104	CSC_COEF_A2_MSB (HDMI_CSC_COEF_A2_MSB)	8	R/W	00h	34.5.272/ 1856
12_4105	CSC_COEF_A2_LSB (HDMI_CSC_COEF_A2_LSB)	8	R/W	00h	34.5.273/ 1856

Table continues on the next page...

HDMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_4106	CSC_COEF_A3_MSB (HDMI_CSC_COEF_A3_MSB)	8	R/W	00h	34.5.274/ 1857
12_4107	CSC_COEF_A3_LSB (HDMI_CSC_COEF_A3_LSB)	8	R/W	00h	34.5.275/ 1857
12_4108	CSC_COEF_A4_MSB (HDMI_CSC_COEF_A4_MSB)	8	R/W	00h	34.5.276/ 1858
12_4109	CSC_COEF_A4_LSB (HDMI_CSC_COEF_A4_LSB)	8	R/W	00h	34.5.277/ 1858
12_410A	CSC_COEF_B1_MSB (HDMI_CSC_COEF_B1_MSB)	8	R/W	00h	34.5.278/ 1859
12_410B	CSC_COEF_B1_LSB (HDMI_CSC_COEF_B1_LSB)	8	R/W	00h	34.5.279/ 1859
12_410C	CSC_COEF_B2_MSB (HDMI_CSC_COEF_B2_MSB)	8	R/W	20h	34.5.280/ 1860
12_410D	CSC_COEF_B2_LSB (HDMI_CSC_COEF_B2_LSB)	8	R/W	00h	34.5.281/ 1860
12_410E	CSC_COEF_B3_MSB (HDMI_CSC_COEF_B3_MSB)	8	R/W	00h	34.5.282/ 1861
12_410F	CSC_COEF_B3_LSB (HDMI_CSC_COEF_B3_LSB)	8	R/W	00h	34.5.283/ 1861
12_4110	CSC_COEF_B4_MSB (HDMI_CSC_COEF_B4_MSB)	8	R/W	00h	34.5.284/ 1862
12_4111	CSC_COEF_B4_LSB (HDMI_CSC_COEF_B4_LSB)	8	R/W	00h	34.5.285/ 1862
12_4112	CSC_COEF_C1_MSB (HDMI_CSC_COEF_C1_MSB)	8	R/W	00h	34.5.286/ 1863
12_4113	CSC_COEF_C1_LSB (HDMI_CSC_COEF_C1_LSB)	8	R/W	00h	34.5.287/ 1863
12_4114	CSC_COEF_C2_MSB (HDMI_CSC_COEF_C2_MSB)	8	R/W	00h	34.5.288/ 1864
12_4115	CSC_COEF_C2_LSB (HDMI_CSC_COEF_C2_LSB)	8	R/W	00h	34.5.289/ 1864
12_4116	CSC_COEF_C3_MSB (HDMI_CSC_COEF_C3_MSB)	8	R/W	20h	34.5.290/ 1865
12_4117	CSC_COEF_C3_LSB (HDMI_CSC_COEF_C3_LSB)	8	R/W	00h	34.5.291/ 1865
12_4118	CSC_COEFC4_MSB (HDMI_CSC_COEFC4_MSB)	8	R/W	00h	34.5.292/ 1866
12_4119	CSC_COEFC4_LSB (HDMI_CSC_COEFC4_LSB)	8	R/W	00h	34.5.293/ 1866
12_7D00	CEC_CTRL (HDMI_CEC_CTRL)	8	R/W	02h	34.5.294/ 1867
12_7D02	CEC_MASK (HDMI_CEC_MASK)	8	R/W	00h	34.5.295/ 1868

Table continues on the next page...

HDMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_7D05	CEC_ADDR_L (HDMI_CEC_ADDR_L)	8	R/W	00h	34.5.296/ 1869
12_7D06	CEC_ADDR_H (HDMI_CEC_ADDR_H)	8	R/W	80h	34.5.297/ 1870
12_7D07	CEC_TX_CNT (HDMI_CEC_TX_CNT)	8	R/W	00h	34.5.298/ 1871
12_7D08	CEC_RX_CNT (HDMI_CEC_RX_CNT)	8	R	00h	34.5.299/ 1872
12_7D10	CEC_TX_DATA (HDMI_CEC_TX_DATA0)	8	R/W	00h	34.5.300/ 1873
12_7D11	CEC_TX_DATA (HDMI_CEC_TX_DATA1)	8	R/W	00h	34.5.300/ 1873
12_7D12	CEC_TX_DATA (HDMI_CEC_TX_DATA2)	8	R/W	00h	34.5.300/ 1873
12_7D13	CEC_TX_DATA (HDMI_CEC_TX_DATA3)	8	R/W	00h	34.5.300/ 1873
12_7D14	CEC_TX_DATA (HDMI_CEC_TX_DATA4)	8	R/W	00h	34.5.300/ 1873
12_7D15	CEC_TX_DATA (HDMI_CEC_TX_DATA5)	8	R/W	00h	34.5.300/ 1873
12_7D16	CEC_TX_DATA (HDMI_CEC_TX_DATA6)	8	R/W	00h	34.5.300/ 1873
12_7D17	CEC_TX_DATA (HDMI_CEC_TX_DATA7)	8	R/W	00h	34.5.300/ 1873
12_7D18	CEC_TX_DATA (HDMI_CEC_TX_DATA8)	8	R/W	00h	34.5.300/ 1873
12_7D19	CEC_TX_DATA (HDMI_CEC_TX_DATA9)	8	R/W	00h	34.5.300/ 1873
12_7D1A	CEC_TX_DATA (HDMI_CEC_TX_DATA10)	8	R/W	00h	34.5.300/ 1873
12_7D1B	CEC_TX_DATA (HDMI_CEC_TX_DATA11)	8	R/W	00h	34.5.300/ 1873
12_7D1C	CEC_TX_DATA (HDMI_CEC_TX_DATA12)	8	R/W	00h	34.5.300/ 1873
12_7D1D	CEC_TX_DATA (HDMI_CEC_TX_DATA13)	8	R/W	00h	34.5.300/ 1873
12_7D1E	CEC_TX_DATA (HDMI_CEC_TX_DATA14)	8	R/W	00h	34.5.300/ 1873
12_7D1F	CEC_TX_DATA (HDMI_CEC_TX_DATA15)	8	R/W	00h	34.5.300/ 1873
12_7D20	CEC_RX_DATA (HDMI_CEC_RX_DATA0)	8	R	00h	34.5.301/ 1873
12_7D21	CEC_RX_DATA (HDMI_CEC_RX_DATA1)	8	R	00h	34.5.301/ 1873

Table continues on the next page...

HDMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_7D22	CEC_RX_DATA (HDMI_CEC_RX_DATA2)	8	R	00h	34.5.301/ 1873
12_7D23	CEC_RX_DATA (HDMI_CEC_RX_DATA3)	8	R	00h	34.5.301/ 1873
12_7D24	CEC_RX_DATA (HDMI_CEC_RX_DATA4)	8	R	00h	34.5.301/ 1873
12_7D25	CEC_RX_DATA (HDMI_CEC_RX_DATA5)	8	R	00h	34.5.301/ 1873
12_7D26	CEC_RX_DATA (HDMI_CEC_RX_DATA6)	8	R	00h	34.5.301/ 1873
12_7D27	CEC_RX_DATA (HDMI_CEC_RX_DATA7)	8	R	00h	34.5.301/ 1873
12_7D28	CEC_RX_DATA (HDMI_CEC_RX_DATA8)	8	R	00h	34.5.301/ 1873
12_7D29	CEC_RX_DATA (HDMI_CEC_RX_DATA9)	8	R	00h	34.5.301/ 1873
12_7D2A	CEC_RX_DATA (HDMI_CEC_RX_DATA10)	8	R	00h	34.5.301/ 1873
12_7D2B	CEC_RX_DATA (HDMI_CEC_RX_DATA11)	8	R	00h	34.5.301/ 1873
12_7D2C	CEC_RX_DATA (HDMI_CEC_RX_DATA12)	8	R	00h	34.5.301/ 1873
12_7D2D	CEC_RX_DATA (HDMI_CEC_RX_DATA13)	8	R	00h	34.5.301/ 1873
12_7D2E	CEC_RX_DATA (HDMI_CEC_RX_DATA14)	8	R	00h	34.5.301/ 1873
12_7D2F	CEC_RX_DATA (HDMI_CEC_RX_DATA15)	8	R	00h	34.5.301/ 1873
12_7D30	CEC_LOCK (HDMI_CEC_LOCK)	8	R/W	00h	34.5.302/ 1874
12_7D31	CEC_WKUPCTRL (HDMI_CEC_WKUPCTRL)	8	R/W	FFh	34.5.303/ 1874
12_7E00	I2CM_SLAVE (HDMI_I2CM_SLAVE)	8	R/W	00h	34.5.304/ 1875
12_7E01	I2CM_ADDRESS (HDMI_I2CM_ADDRESS)	8	R/W	00h	34.5.305/ 1876
12_7E02	I2CM_DATAO (HDMI_I2CM_DATAO)	8	R/W	00h	34.5.306/ 1876
12_7E03	I2CM_DATAI (HDMI_I2CM_DATAI)	8	R	00h	34.5.307/ 1877
12_7E04	I2CM_OPERATION (HDMI_I2CM_OPERATION)	8	W	00h	34.5.308/ 1877
12_7E05	I2CM_INT (HDMI_I2CM_INT)	8	R/W	08h	34.5.309/ 1878

Table continues on the next page...

HDMI memory map (continued)

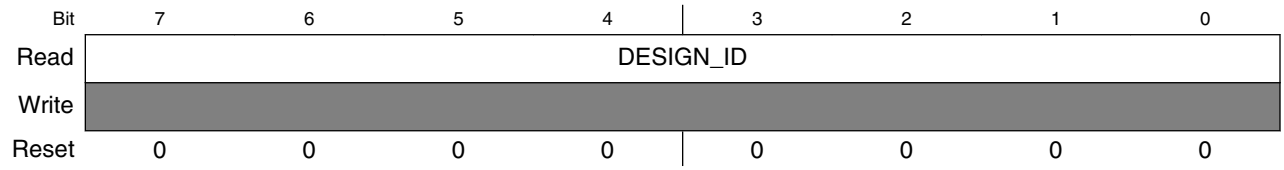
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
12_7E06	I2CM_CTLINT (HDMI_I2CM_CTLINT)	8	R/W	88h	34.5.310/ 1879
12_7E07	I2CM_DIV (HDMI_I2CM_DIV)	8	R/W	0Bh	34.5.311/ 1879
12_7E08	I2CM_SEGADDR (HDMI_I2CM_SEGADDR)	8	R/W	00h	34.5.312/ 1880
12_7E09	I2CM_SOFTRSTZ (HDMI_I2CM_SOFTRSTZ)	8	R/W	01h	34.5.313/ 1881
12_7E0A	I2CM_SEGPTR (HDMI_I2CM_SEGPTR)	8	R/W	00h	34.5.314/ 1881
12_7E0B	I2CM_SS_SCL_HCNT_1_ADDR (HDMI_I2CM_SS_SCL_HCNT_1_ADDR)	8	R/W	00h	34.5.315/ 1882
12_7E0C	I2CM_SS_SCL_HCNT_0_ADDR (HDMI_I2CM_SS_SCL_HCNT_0_ADDR)	8	R/W	6Ch	34.5.316/ 1882
12_7E0D	I2CM_SS_SCL_LCNT_1_ADDR (HDMI_I2CM_SS_SCL_LCNT_1_ADDR)	8	R/W	00h	34.5.317/ 1883
12_7E0E	I2CM_SS_SCL_LCNT_0_ADDR (HDMI_I2CM_SS_SCL_LCNT_0_ADDR)	8	R/W	7Fh	34.5.318/ 1883
12_7E0F	I2CM_FS_SCL_HCNT_1_ADDR (HDMI_I2CM_FS_SCL_HCNT_1_ADDR)	8	R/W	00h	34.5.319/ 1884
12_7E10	I2CM_FS_SCL_HCNT_0_ADDR (HDMI_I2CM_FS_SCL_HCNT_0_ADDR)	8	R/W	11h	34.5.320/ 1884
12_7E11	I2CM_FS_SCL_LCNT_1_ADDR (HDMI_I2CM_FS_SCL_LCNT_1_ADDR)	8	R/W	00h	34.5.321/ 1885
12_7E12	I2CM_FS_SCL_LCNT_0_ADDR (HDMI_I2CM_FS_SCL_LCNT_0_ADDR)	8	R/W	24h	34.5.322/ 1885
12_7F00	BASE_POINTER_ADDR (HDMI_BASE_POINTER_ADDR)	8	R/W	00h	34.5.323/ 1886

34.5.1 Design Identification Register (HDMI_DESIGN_ID)

The following are the registers used to identify the HDMI TX controller.

- Name: Design Identification Register
- Address Offset: 0x0000
- Size: 8 bits
- Value after Reset: Implementation Dependent
- Access: Read

Address: 12_0000h base + 0h offset = 12_0000h



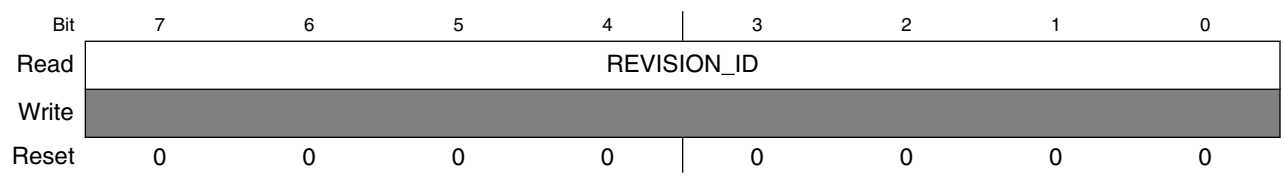
HDMI_DESIGN_ID field descriptions

Field	Description
DESIGN_ID	This is a 1 byte design ID code fixed by NXP that Identifies the main revision of the HDMI TX controller. For example, HDMI TX 1.30a, DESIGN_ID = 11h; REVISION_ID = 0Ah

34.5.2 Revision Identification Register (HDMI_REVISION_ID)

- Name: Revision Identification Register
- Address Offset: 0x0001
- Size: 8 bits
- Value after Reset: Implementation Dependent
- Access: Read

Address: 12_0000h base + 1h offset = 12_0001h



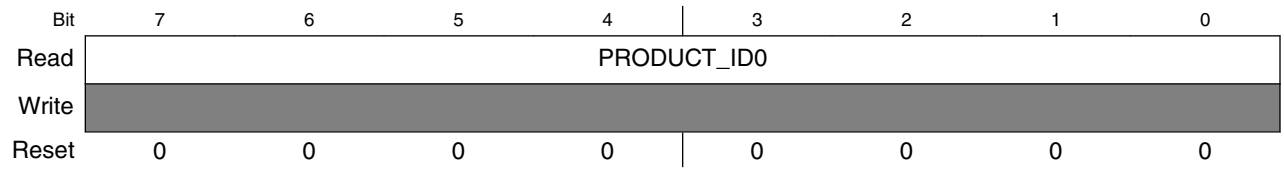
HDMI_REVISION_ID field descriptions

Field	Description
REVISION_ID	This is a one byte revision ID code fixed by NXP that Identifies the main revision of the HDMI TX controller. For example, HDMI TX 1.30a, DESIGN_ID = 12h; REVISION_ID = 0Ah

34.5.3 Product Identification Register 0 (HDMI_PRODUCT_ID0)

- Name: Product Identification Register 0
- Address Offset: 0x0002
- Size: 8 bits
- Value after Reset: Implementation Dependent
- Access: Read

Address: 12_0000h base + 2h offset = 12_0002h



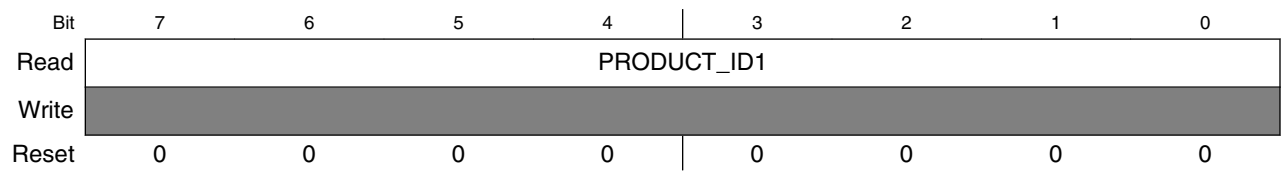
HDMI_PRODUCT_ID0 field descriptions

Field	Description
PRODUCT_ID0	This one byte fixed code identifies the NXP product line ("A0h" for HDMI TX products).

34.5.4 Product Identification Register 1 (HDMI_PRODUCT_ID1)

- Name: Product Identification Register 1
- Address Offset: 0x0003
- Size: 8 bits
- Value after Reset: Implementation Dependent
- Access: Read

Address: 12_0000h base + 3h offset = 12_0003h



HDMI_PRODUCT_ID1 field descriptions

Field	Description
PRODUCT_ID1	This one byte fixed code identifies the NXP product line according to: 01h HDMI TX Controller C1h HDMI TX Controller with HDCP encryption engine

34.5.5 Configuration Identification Register 0 (HDMI_CONFIG0_ID)

- Name: Configuration Identification Register 0
- Address Offset: 0x0004
- Size: 8 bits
- Value after Reset: Implementation Dependent
- Access: Read

Address: 12_0000h base + 4h offset = 12_0004h

Bit	7	6	5	4	3	2	1	0
Read	prepen	audhbr	audspdif	audi2s	hdmi14	csc	cec	hdcp
Write								
Reset	0	0	0	0	0	0	0	0

HDMI_CONFIG0_ID field descriptions

Field	Description
7 prepen	Indicates if it is possible to use internal pixel repetition
6 audhbr	Indicates if HBR interface is present
5 audspdif	Indicates if SPDIF interface is present
4 audi2s	Indicates if I2S interface is present
3 hdmi14	Indicates if HDMI 1.4 features are present
2 csc	Indicates if Color Space Conversion block is present
1 cec	Indicates if CEC is present
0 hdcp	Indicates if HDCP is present

34.5.6 Configuration Identification Register 1 (HDMI_CONFIG1_ID)

- Name: Configuration Identification Register 1
- Address Offset: 0x0005
- Size: 8 bits
- Value after Reset: Implementation Dependent
- Access: Read

Address: 12_0000h base + 5h offset = 12_0005h

Bit	7	6	5	4	3	2	1	0
Read	Reserved			confsfrdir	confi2c	confocp	confapb	confahb
Write								
Reset	0	0	0	0	0	0	0	0

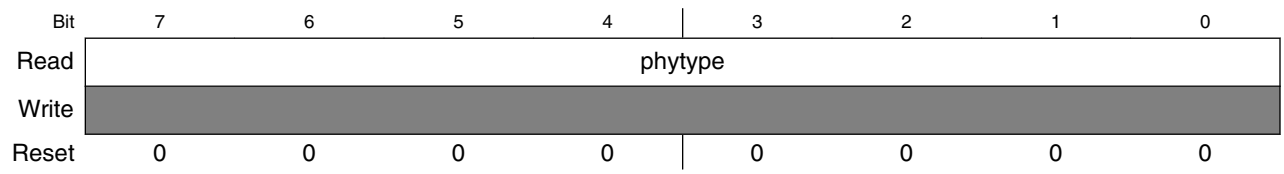
HDMI_CONFIG1_ID field descriptions

Field	Description
7–5 -	This field is reserved.
4 confsfrdir	Indicates that configuration interface is SFR interface
3 confi2c	Indicates that configuration interface is I2C interface
2 confocp	Indicates that configuration interface is OCP interface
1 confapb	Indicates that configuration interface is APB interface
0 confahb	Indicates that configuration interface is AHB interface

34.5.7 Configuration Identification Register 2 (HDMI_CONFIG2_ID)

- Name: Configuration Identification Register 2
- Address Offset: 0x0006
- Size: 8 bits
- Value after Reset: Implementation Dependent
- Access: Read

Address: 12_0000h base + 6h offset = 12_0006h



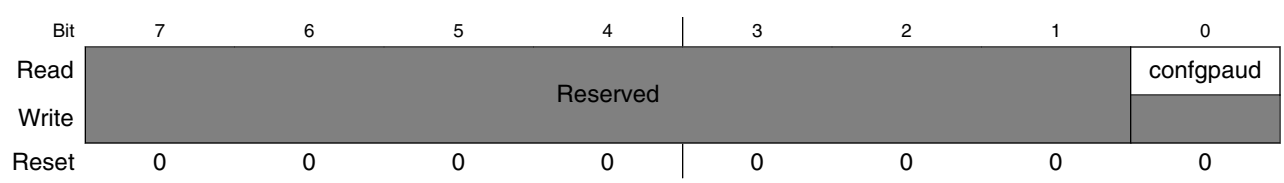
HDMI_CONFIG2_ID field descriptions

Field	Description
phytype	Indicates the type of PHY interface selected: 00h Legacy PHY (HDMI TX PHY) F2h PHY_Gen2 (HDMI 3D TX PHY) E2h PHY_Gen2 (HDMI 3D TX PHY) + HEAC PHY

34.5.8 Configuration Identification Register 3 (HDMI_CONFIG3_ID)

- Name: Configuration Identification Register 3
- Address Offset: 0x0007
- Size: 8 bits
- Value after Reset: Implementation Dependent
- Access: Read

Address: 12_0000h base + 7h offset = 12_0007h



HDMI_CONFIG3_ID field descriptions

Field	Description
7–1 -	This field is reserved.
0 confgpaud	Indicates that configuration interface is Generic Parallel Audio (GPAUD) interface

34.5.9 Frame Composer Interrupt Status Register 0 (HDMI_IH_FC_STAT0)

This section describes clear on write (1 to corresponding bit) status registers, which contain the following active-high, sticky bit interrupts.

HDMI TX introduces a new set of sticky bit mute control registers (IH_MUTE_FC_STAT0 to IH_MUTE_AHBDMAAUD_STAT0) that correspond to the interrupt registers. You can ignore a sticky bit interrupt by setting the corresponding mute control register bit to 1. This puts the global interrupt line on a higher priority than the sticky bit interrupt.

- Address Offset: 0x0100
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Clear on Write

Address: 12_0000h base + 100h offset = 12_0100h

Bit	7	6	5	4	3	2	1	0
Read	AUDI	ACP	HBR	DST	OBA	AUDS	ACR	NULL
Write	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

HDMI_IH_FC_STAT0 field descriptions

Field	Description
7 AUDI	Active after successful transmission of an Audio InfoFrame packet.
6 ACP	Active after successful transmission of an Audio Content Protection packet.
5 HBR	Active after successful transmission of an Audio HBR packet.
4 DST	Reserved
3 OBA	Reserved

Table continues on the next page...

HDMI_IH_FC_STAT0 field descriptions (continued)

Field	Description
2 AUDS	Active after successful transmission of an Audio Sample packet. Due to high number of audio sample packets transmitted, this interrupt is by default masked at frame composer.
1 ACR	Active after successful transmission of an Audio Clock Regeneration (N/CTS transmission) packet.
0 NULL	Active after successful transmission of an Null packet. Due to high number of audio sample packets transmitted, this interrupt is by default masked at frame composer.

34.5.10 Frame Composer Interrupt Status Register 1 (HDMI_IH_FC_STAT1)

- Address Offset: 0x0101
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Clear on Write

Address: 12_0000h base + 101h offset = 12_0101h

Bit	7	6	5	4	3	2	1	0
Read	GMD	ISCR1	ISCR2	VSD	SPD	MPEG	AVI	GCP
Write	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

HDMI_IH_FC_STAT1 field descriptions

Field	Description
7 GMD	Active after successful transmission of an Gamut metadata packet.
6 ISCR1	Active after successful transmission of an International Standard Recording Code 1 packet.
5 ISCR2	Active after successful transmission of an International Standard Recording Code 2 packet.
4 VSD	Active after successful transmission of an Vendor Specific Data infoFrame packet.
3 SPD	Active after successful transmission of an Source Product Descriptor infoFrame packet.
2 MPEG	Reserved
1 AVI	Active after successful transmission of an AVI infoFrame packet.
0 GCP	Active after successful transmission of an General Control Packet.

34.5.11 Frame Composer Interrupt Status Register 2 (HDMI_IH_FC_STAT2)

- Address Offset: 0x0102
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Clear on Write

Address: 12_0000h base + 102h offset = 12_0102h

Bit	7	6	5	4	3	2	1	0
Read	Reserved						LowPriority_ overflow	HighPriority _overflow
Write							w1c	w1c
Reset	0	0	0	0	0	0	0	0

HDMI_IH_FC_STAT2 field descriptions

Field	Description
7–2 -	This field is reserved. Reserved
1 LowPriority_ overflow	Frame Composer low priority packet queue descriptor overflow indication.
0 HighPriority_ overflow	Frame Composer high priority packet queue descriptor overflow indication.

34.5.12 Audio Sampler Interrupt Status Register (HDMI_IH_AS_STAT0)

- Address Offset: 0x0103
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Clear on Write

Address: 12_0000h base + 103h offset = 12_0103h

Bit	7	6	5	4	3	2	1	0
Read	Reserved					Aud_fifo_ underflow_ thr	Aud_fifo_ underflow	Aud_fifo_ overflow
Write						w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

HDMI_IH_AS_STAT0 field descriptions

Field	Description
7–3 -	This field is reserved. Reserved
2 Aud_fifo_ underflow_thr	Audio Sampler audio FIFO empty threshold (four samples) indication. Only valid in HBR audio.
1 Aud_fifo_ underflow	Audio Sampler audio FIFO empty indication.
0 Aud_fifo_ overflow	Audio Sampler audio FIFO full indication.

34.5.13 PHY Interface Interrupt Status Register (HDMI_IH_PHY_STAT0)

- Address Offset: 0x0104
- Size: 8 bits
- Value after Reset: 0x00
- Access: Clear on Write/Read

Address: 12_0000h base + 104h offset = 12_0104h

Bit	7	6	5	4
Read	Reserved		RX_SENSE3	RX_SENSE2
Write			w1c	w1c
Reset	0	0	0	0

Bit	3	2	1	0
Read	RX_SENSE1	RX_SENSE0	TX_PHY_LOCK	HDP
Write	w1c	w1c	w1c	w1c
Reset	0	0	0	0

HDMI_IH_PHY_STAT0 field descriptions

Field	Description
7–6 -	This field is reserved. Reserved
5 RX_SENSE3	TX PHY RX_SENSE indication for driver 3. You may need to mask or change polarity of this interrupt after it has become active.
4 RX_SENSE2	TX PHY RX_SENSE indication for driver 2. You may need to mask or change polarity of this interrupt after it has become active.
3 RX_SENSE1	TX PHY RX_SENSE indication for driver 1. You may need to mask or change polarity of this interrupt after it has become active.
2 RX_SENSE0	TX PHY RX_SENSE indication for driver 0. You may need to mask or change polarity of this interrupt after it has become active.
1 TX_PHY_LOCK	TX PHY PLL lock indication. Please refer to PHY datasheet for more information. You may need to mask or change polarity of this interrupt after it has become active.
0 HDP	HDMI Hot Plug Detect indication. You may need to mask or change polarity of this interrupt after it has become active.

34.5.14 E-DDC I2C Master Interrupt Status Register (HDMI_IH_I2CM_STAT0)

- Address Offset: 0x0105
- Size: 8 bits
- Value after Reset: 0x00
- Access: Clear on Write/Read

Address: 12_0000h base + 105h offset = 12_0105h

Bit	7	6	5	4
Read	Reserved			
Write				
Reset	0	0	0	0

Bit	3	2	1	0
Read	Reserved		I2Cmasterdone	I2CMASTER_ERROR
Write			w1c	w1c
Reset	0	0	0	0

HDMI_IH_I2CM_STAT0 field descriptions

Field	Description
7–2 -	This field is reserved. Reserved
1 I2Cmasterdone	I2C Master done indication
0 I2CMASTER_ERROR	I2C Master error indication

34.5.15 CEC Interrupt Status Register (HDMI_IH_CEC_STAT0)

- Address Offset: 0x0106
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Clear on Write

Address: 12_0000h base + 106h offset = 12_0106h

Bit	7	6	5	4	3	2	1	0
Read	Reserved	WAKEUP	ERROR_ FOLLOW	ERROR_ INITIATOR	ARB_LOST	NACK	EOM	DONE
Write		w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

HDMI_IH_CEC_STAT0 field descriptions

Field	Description
7 -	This field is reserved. Reserved
6 WAKEUP	CEC Wake-up indication
5 ERROR_ FOLLOW	CEC Error_follow indication
4 ERROR_ INITIATOR	CEC Error_follow indication
3 ARB_LOST	CEC Arb_Lost indication
2 NACK	CEC Nack indication
1 EOM	CEC End of Message Indication
0 DONE	CEC Done Indication

34.5.16 Video Packetizer Interrupt Status Register (HDMI_IH_VP_STAT0)

- Address Offset: 0x0107
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Clear on Write

Address: 12_0000h base + 107h offset = 12_0107h

Bit	7	6	5	4
Read	fifofullrepet	fifoemptyrepet	fifofullpp	fifoemptypp
Write	w1c	w1c	w1c	w1c
Reset	0	0	0	0

Bit	3	2	1	0
Read	fifofullremap	fifoemptyremap	fifofullbyp	fifoemptybyp
Write	w1c	w1c	w1c	w1c
Reset	0	0	0	0

HDMI_IH_VP_STAT0 field descriptions

Field	Description
7 fifofullrepet	Video packetizer pixel repeater FIFO full interrupt
6 fifoemptyrepet	Video packetizer pixel repeater FIFO empty interrupt
5 fifofullpp	Video packetizer pixel packing FIFO full interrupt
4 fifoemptypp	Video packetizer pixel packing FIFO empty interrupt
3 fifofullremap	Video packetizer pixel YCC 422 re-mapper FIFO full interrupt
2 fifoemptyremap	Video packetizer pixel YCC 422 re-mapper FIFO empty interrupt
1 fifofullbyp	Video packetizer 8-bit bypass fifo full interrupt
0 fifoemptybyp	Video packetizer 8-bit bypass fifo empty interrupt

34.5.17 PHY GEN2 I2C Master Interrupt Status Register (HDMI_IH_I2CMPHY_STAT0)

This clear on write (1 to corresponding bit) register contains the following active high sticky bit interrupts. That I2C Master PHY is the I2C Master block used to access the PHY I2C Slave.

- Address Offset: 0x0108
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Clear on Write

Address: 12_0000h base + 108h offset = 12_0108h

Bit	7	6	5	4
Read	Reserved			
Write				
Reset	0	0	0	0

Bit	3	2	1	0
Read	Reserved		i2cmphydone	i2cmphyerror
Write			w1c	w1c
Reset	0	0	0	0

HDMI_IH_I2CMPHY_STAT0 field descriptions

Field	Description
7–2 -	This field is reserved. Reserved
1 i2cmphydone	I2C Master PHY done indication
0 i2cmphyerror	I2C Master PHY error indication

34.5.18 AHB Audio DMA Interrupt Status Register (HDMI_IH_AHBDMAAUD_STAT0)

Address Offset: 0x0109

Size: 8 bits

Value after Reset: 0x00

Access: Read/Clear on Write

Address: 12_0000h base + 109h offset = 12_0109h

Bit	7	6	5	4
Read	Reserved		ahbdmaaud_interror	ahbdmaaud_ intlostownership
Write			w1c	w1c
Reset	0	0	0	0

Bit	3	2	1	0
Read	ahbdmaaud_intretrysplit	ahbdmaaud_intdone	ahbdmaaud_intbufffull	ahbdmaaud_ intbuffempty
Write	w1c	w1c	w1c	w1c
Reset	0	0	0	0

HDMI_IH_AHBDMAAUD_STAT0 field descriptions

Field	Description
7–6 -	This field is reserved. Reserved
5 ahbdmaaud_interror	AHB audio DMA error interrupt
4 ahbdmaaud_ intlostownership	AHB audio DMA lost ownership interrupt
3 ahbdmaaud_ intretrysplit	AHB audio DMA RETRY/SPLIT interrupt
2 ahbdmaaud_intdone	AHB audio DMA done interrupt
1 ahbdmaaud_ intbufffull	AHB audio DMA Buffer full interrupt
0 ahbdmaaud_ intbuffempty	AHB audio DMA Buffer empty interrupt

34.5.19 Frame Composer Interrupt Mute Control Register 0 (HDMI_IH_MUTE_FC_STAT0)

- Address Offset: 0x0180
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 180h offset = 12_0180h

Bit	7	6	5	4	3	2	1	0
Read Write	AUDI	ACP	HBR	DST	OBA	AUDS	ACR	NULL
Reset	0	0	0	0	0	0	0	0

HDMI_IH_MUTE_FC_STAT0 field descriptions

Field	Description
7 AUDI	When set to 1, mutes IH_ FC_STAT0[7]
6 ACP	When set to 1, mutes IH_ FC_STAT0[6]
5 HBR	When set to 1, mutes IH_ FC_STAT0[5]
4 DST	When set to 1, mutes IH_ FC_STAT0[4]
3 OBA	When set to 1, mutes IH_ FC_STAT0[3]
2 AUDS	When set to 1, mutes IH_ FC_STAT0[2]
1 ACR	When set to 1, mutes IH_ FC_STAT0[1]
0 NULL	When set to 1, mutes IH_ FC_STAT0[0]

34.5.20 Frame Composer Interrupt Mute Control Register 1 (HDMI_IH_MUTE_FC_STAT1)

- Address Offset: 0x0181
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 181h offset = 12_0181h

Bit	7	6	5	4	3	2	1	0
Read Write	GMD	ISCR1	ISCR2	VSD	SPD	MPEG	AVI	GCP
Reset	0	0	0	0	0	0	0	0

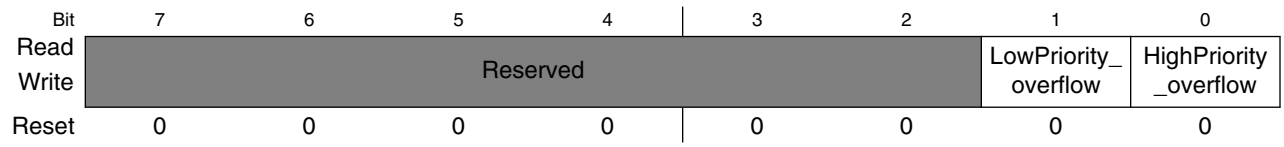
HDMI_IH_MUTE_FC_STAT1 field descriptions

Field	Description
7 GMD	When set to 1, mutes IH_ FC_STAT1[7]
6 ISCR1	When set to 1, mutes IH_ FC_STAT1[6]
5 ISCR2	When set to 1, mutes IH_ FC_STAT1[5]
4 VSD	When set to 1, mutes IH_ FC_STAT1[4]
3 SPD	When set to 1, mutes IH_ FC_STAT1[3]
2 MPEG	When set to 1, mutes IH_ FC_STAT1[2]
1 AVI	When set to 1, mutes IH_ FC_STAT1[1]
0 GCP	When set to 1, mutes IH_ FC_STAT1[0]

34.5.21 Frame Composer Interrupt Mute Control Register 2 (HDMI_IH_MUTE_FC_STAT2)

- Address Offset: 0x0182
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 182h offset = 12_0182h



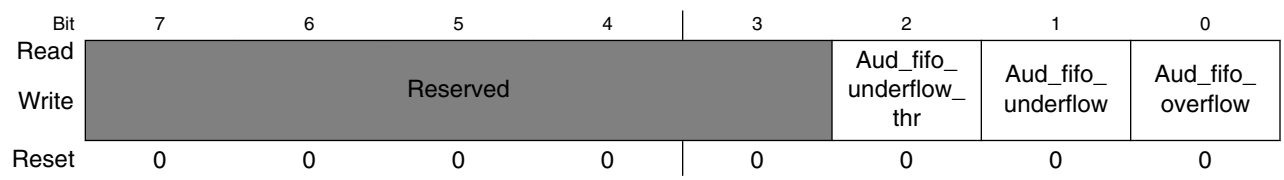
HDMI_IH_MUTE_FC_STAT2 field descriptions

Field	Description
7–2 -	This field is reserved. Reserved
1 LowPriority_ overflow	When set to 1, mutes IH_ FC_STAT2[1]
0 HighPriority_ overflow	When set to 1, mutes IH_ FC_STAT2[0]

34.5.22 Audio Sampler Interrupt Mute Control Register 0 (HDMI_IH_MUTE_AS_STAT0)

- Address Offset: 0x0183
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 183h offset = 12_0183h



HDMI_IH_MUTE_AS_STAT0 field descriptions

Field	Description
7–3 -	This field is reserved. Reserved
2 Aud_fifo_ underflow_thr	When set to 1, mutes IH_ AS_STAT0[2]
1 Aud_fifo_ underflow	When set to 1, mutes IH_ AS_STAT0[1]
0 Aud_fifo_ overflow	When set to 1, mutes IH_ AS_STAT0[0]

34.5.23 PHY Interface Interrupt Mute Control Register (HDMI_IH_MUTE_PHY_STAT0)

- Address Offset: 0x0184
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 184h offset = 12_0184h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved		RX_ SENSE3	RX_ SENSE2	RX_ SENSE1	RX_ SENSE0	TX_PHY_ LOCK	HDP
Reset	0	0	0	0	0	0	0	0

HDMI_IH_MUTE_PHY_STAT0 field descriptions

Field	Description
7–6 -	This field is reserved. Reserved
5 RX_SENSE3	When set to 1, mutes IH_ PHY_STAT0[5]
4 RX_SENSE2	When set to 1, mutes IH_ PHY_STAT0[4]
3 RX_SENSE1	When set to 1, mutes IH_ PHY_STAT0[3]
2 RX_SENSE0	When set to 1, mutes IH_ PHY_STAT0[2]
1 TX_PHY_LOCK	When set to 1, mutes IH_ PHY_STAT0[1]
0 HDP	When set to 1, mutes IH_ PHY_STAT0[0]

34.5.24 E-DDC I2C Master Interrupt Mute Control Register (HDMI_IH_MUTE_I2CM_STAT0)

- Address Offset: 0x0185
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 185h offset = 12_0185h

Bit	7	6	5	4
Read Write	Reserved			
Reset	0	0	0	0

Bit	3	2	1	0
Read Write	Reserved		I2Cmasterdone	I2CMASTER_ERROR
Reset	0	0	0	0

HDMI_IH_MUTE_I2CM_STAT0 field descriptions

Field	Description
7–2 -	This field is reserved. Reserved
1 I2Cmasterdone	When set to 1, mutes IH_ I2CM_STAT0[1]
0 I2CMASTER_ERROR	When set to 1, mutes IH_ I2CM_STAT0[0]

34.5.25 CEC Interrupt Mute Control Register (HDMI_IH_MUTE_CEC_STAT0)

- Address Offset: 0x0186
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 186h offset = 12_0186h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved	WAKEUP	ERROR_ FOLLOW	ERROR_ INITIATOR	ARB_LOST	NACK	EOM	DONE
Reset	0	0	0	0	0	0	0	0

HDMI_IH_MUTE_CEC_STAT0 field descriptions

Field	Description
7 -	This field is reserved. Reserved
6 WAKEUP	When set to 1, mutes IH_ CEC_STAT0[6]
5 ERROR_ FOLLOW	When set to 1, mutes IH_ CEC_STAT0[5]
4 ERROR_ INITIATOR	When set to 1, mutes IH_ CEC_STAT0[4]
3 ARB_LOST	When set to 1, mutes IH_ CEC_STAT0[3]
2 NACK	When set to 1, mutes IH_ CEC_STAT0[2]
1 EOM	When set to 1, mutes IH_ CEC_STAT0[1]
0 DONE	When set to 1, mutes IH_ CEC_STAT0[0]

34.5.26 Video Packetizer Interrupt Mute Control Register (HDMI_IH_MUTE_VP_STAT0)

- Address Offset: 0x0187
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 187h offset = 12_0187h

Bit	7	6	5	4
Read Write	fifofullrepet	fifoemptyrepet	fifofullpp	fifoemptypp
Reset	0	0	0	0

Bit	3	2	1	0
Read Write	fifofullremap	fifoemptyremap	fifofullbyp	fifoemptybyp
Reset	0	0	0	0

HDMI_IH_MUTE_VP_STAT0 field descriptions

Field	Description
7 fifofullrepet	When set to 1, mutes IH_ VP_STAT0[7]

Table continues on the next page...

HDMI_IH_MUTE_VP_STAT0 field descriptions (continued)

Field	Description
6 fifoemptyrepet	When set to 1, mutes IH_ VP_STAT0[6]
5 fifofullpp	When set to 1, mutes IH_ VP_STAT0[5]
4 fifoemptypp	When set to 1, mutes IH_ VP_STAT0[4]
3 fifofullremap	When set to 1, mutes IH_ VP_STAT0[3]
2 fifoemptyremap	When set to 1, mutes IH_ VP_STAT0[2]
1 fifofullbyp	When set to 1, mutes IH_ VP_STAT0[1]
0 fifoemptybyp	When set to 1, mutes IH_ VP_STAT0[0]

34.5.27 PHY GEN 2 I2C Master Interrupt Mute Control Register (HDMI_IH_MUTE_I2CMPHY_STAT0)

- Address Offset: 0x0188
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 188h offset = 12_0188h

Bit	7	6	5	4
Read Write	Reserved			
Reset	0	0	0	0

Bit	3	2	1	0
Read Write	Reserved		i2cmphydone	i2cmphyerror
Reset	0	0	0	0

HDMI_IH_MUTE_I2CMPHY_STAT0 field descriptions

Field	Description
7–2 -	This field is reserved. Reserved
1 i2cmphydone	When set to 1, mutes IH_ I2CMPHY_STAT0[1]
0 i2cmphyerror	When set to 1, mutes IH_ I2CMPHY_STAT0[0]

34.5.28 AHB Audio DMA Interrupt Mute Control Register (HDMI_IH_MUTE_AHBDMAAUD_STAT0)

- Address Offset: 0x0189
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 189h offset = 12_0189h

Bit	7	6	5	4
Read Write	Reserved		ahbdmaaud_interror	ahbdmaaud_ intlostownership
Reset	0	0	0	0

Bit	3	2	1	0
Read Write	ahbdmaaud_intretrysplit	ahbdmaaud_intdone	ahbdmaaud_intbufffull	ahbdmaaud_ intbuffempty
Reset	0	0	0	0

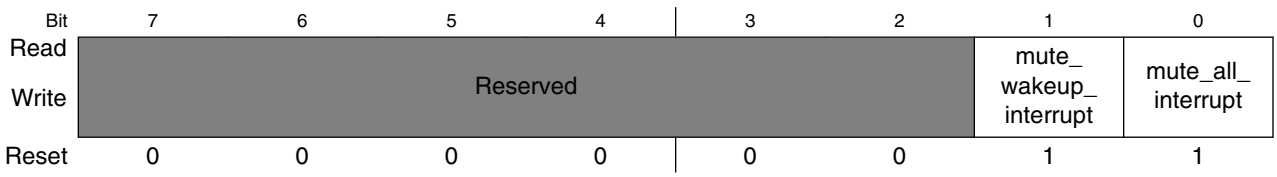
HDMI_IH_MUTE_AHBDMAAUD_STAT0 field descriptions

Field	Description
7–6 -	This field is reserved. Reserved
5 ahbdmaaud_interror	When set to 1, mutes IH_AHBDMAAUD_STAT0[5]
4 ahbdmaaud_ intlostownership	When set to 1, mutes IH_AHBDMAAUD_STAT0[4]
3 ahbdmaaud_ intretrysplit	When set to 1, mutes IH_AHBDMAAUD_STAT0[3]
2 ahbdmaaud_intdone	When set to 1, mutes IH_AHBDMAAUD_STAT0[2]
1 ahbdmaaud_ intbufffull	When set to 1, mutes IH_AHBDMAAUD_STAT0[1]
0 ahbdmaaud_ intbuffempty	When set to 1, mutes IH_AHBDMAAUD_STAT0[0]

34.5.29 Global Interrupt Mute Control Register (HDMI_IH_MUTE)

- Address Offset: 0x01FF
- Size: 8 bits
- Value after Reset: 0x03
- Access: Read/Write

Address: 12_0000h base + 1FFh offset = 12_01FFh



HDMI_IH_MUTE field descriptions

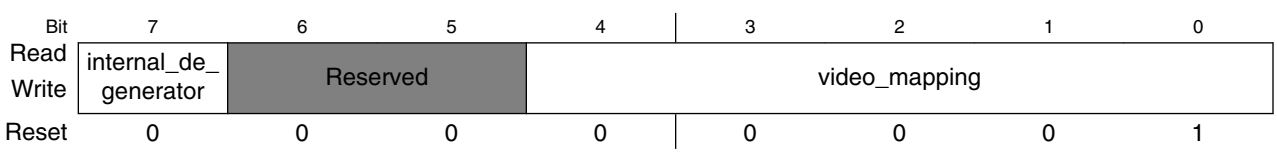
Field	Description
7–2 -	This field is reserved.
1 mute_wakeup_ interrupt	When set to 1, mutes the wake-up interrupt line. The sticky bit interrupt continues with its state; only the wake up interrupt line is muted.
0 mute_all_ interrupt	When set to 1, mutes the main interrupt line (where all interrupts are ORed). The sticky bit interrupts continue with their state; only the main interrupt line will be muted.

34.5.30 Video Input Mapping and Internal Data Enable Configuration Register (HDMI_TX_INVID0)

This registers contains the input video mapping code as defined in Table 2-1.

- Address Offset: 0x0200
- Size: 8 bits
- Value after Reset: 0x01
- Access: Read/Write

Address: 12_0000h base + 200h offset = 12_0200h



HDMI_TX_INVID0 field descriptions

Field	Description
7 internal_de_ generator	Internal data enable (DE) generator enable. If data enable is not available for the input video the user may set this bit to one to activate the internal data enable generator. **NOTE:** This feature only works for input video modes that have native repetition (such as, all CEA videos). No desired pixel repetition can be used with this feature because these configurations only affect the Frame Composer and not this block.
6–5 -	This field is reserved. Reserved
video_mapping	video_mapping

34.5.31 Video Input Stuffing Enable Register (HDMI_TX_INSTUFFING)

This register enables the stuffing mechanism of the Video Sampler module in order to correctly perform Color Space Conversion of the ITU.601 standard YCC video. In this case, when "de" is low, the output video components gydata[15:0], rcrdata[15:0], and bcbdata[15:0] can be configured.

- Address Offset: 0x0201
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 201h offset = 12_0201h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved					BCBDATA_ STUFFING	RCRDATA_ STUFFING	GYDATA_ STUFFING
Reset	0	0	0	0	0	0	0	0

HDMI_TX_INSTUFFING field descriptions

Field	Description
7–3 -	This field is reserved. Reserved
2 BCBDATA_ STUFFING	BCBDATA stuffing bit 0 When the dataen signal is low, the value in the bcbdata[15:0] output is the one sampled from the corresponding input data. 1 When the dataen signal is low, the value in the bcbdata[15:0] output is given by the values in register TX_BCBDTA0 and TX_BCBDATA1.
1 RCRDATA_ STUFFING	RCRDATA stuffing bit

Table continues on the next page...

HDMI_TX_INSTUFFING field descriptions (continued)

Field	Description
	0 When the dataen signal is low, the value in the rcrdata[15:0] output is the one sampled from the corresponding input data. 1 When the dataen signal is low, the value in the rcrdata[15:0] output is given by the values in TX_RCRDTA0 and TX_RCRDATA1 registers.
0 GYDATA_ STUFFING	GYDATA stuffing bit 0 when the dataen signal is low, the value in the gydata[15:0] output is the one sampled from the corresponding input data. 1 When the dataen signal is low, the value in the gydata[15:0] output is given by the values in TX_GYDTA0 and TX_GYDATA1 registers.

34.5.32 Video Input GY Data Channel Stuffing Register 0 (HDMI_TX_GYDATA0)

- Address Offset: 0x0202
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 202h offset = 12_0202h

Bit	7	6	5	4	3	2	1	0
Read Write	gydata							
Reset	0	0	0	0	0	0	0	0

HDMI_TX_GYDATA0 field descriptions

Field	Description
gydata	gydata[7:0].This register defines the value of gydata[7:0] when TX_INSTUFFING[0] (gydata_stuffing) is set to 1b.

34.5.33 Video Input GY Data Channel Stuffing Register 1 (HDMI_TX_GYDATA1)

- Address Offset: 0x0203
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 203h offset = 12_0203h

Bit	7	6	5	4	3	2	1	0
Read Write	gydata							
Reset	0	0	0	0	0	0	0	0

HDMI_TX_GYDATA1 field descriptions

Field	Description
gydata	gydata[15:8].This register defines the value of gydata[15:8] when TX_INSTUFFING[0] (gydata_stuffing) is set to 1b.

34.5.34 Video Input RCR Data Channel Stuffing Register 0 (HDMI_TX_RCRDATA0)

- Address Offset: 0x0204
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 204h offset = 12_0204h

Bit	7	6	5	4	3	2	1	0
Read Write	rcrdata							
Reset	0	0	0	0	0	0	0	0

HDMI_TX_RCRDATA0 field descriptions

Field	Description
rcrdata	rcrdata[7:0]. This register defines the value of rcrydata[7:0] when TX_INSTUFFING[1] (rcrdata_stuffing) is set to 1b.

34.5.35 Video Input RCR Data Channel Stuffing Register 1 (HDMI_TX_RCRDATA1)

- Address Offset: 0x0205
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 205h offset = 12_0205h

Bit	7	6	5	4	3	2	1	0
Read Write	rcrdata							
Reset	0	0	0	0	0	0	0	0

HDMI_TX_RCRDATA1 field descriptions

Field	Description
rcrdata	rcrdata[15:8]. This register defines the value of rcrydata[15:8] when TX_INSTUFFING[1] (rcrdata_stuffing) is set to 1b.

34.5.36 Video Input RCB Data Channel Stuffing Register 0 (HDMI_TX_BCBDATA0)

- Address Offset: 0x0206
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 206h offset = 12_0206h

Bit	7	6	5	4	3	2	1	0
Read Write	bcbdata							
Reset	0	0	0	0	0	0	0	0

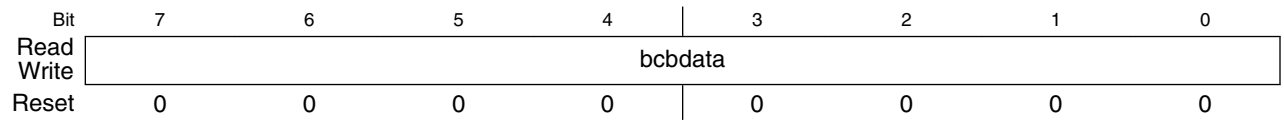
HDMI_TX_BCBDATA0 field descriptions

Field	Description
bcbdata	bcbdata[7:0]. This register defines the value of bcbdata[7:0] when TX_INSTUFFING[2] (bcbdata_stuffing) is set to 1b.

34.5.37 Video Input RCB Data Channel Stuffing Register 1 (HDMI_TX_BCBDATA1)

- Address Offset: 0x0207
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 207h offset = 12_0207h



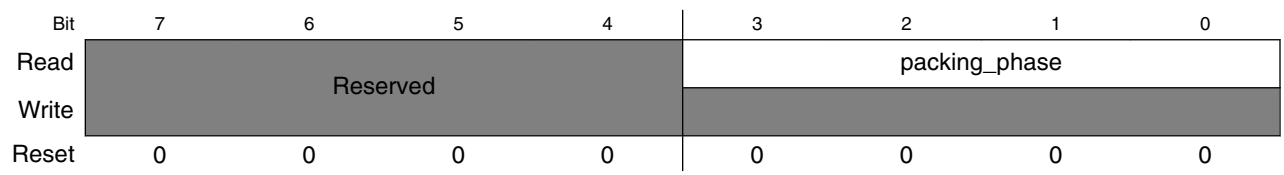
HDMI_TX_BCBDATA1 field descriptions

Field	Description
bcbdata	bcbdata[15:8]. This register defines the value of bcbdata[15:8] when TX_INSTUFFING[2] (bcbdata_stuffing) is set to 1b.

34.5.38 Video Packetizer Packing Phase Status Register (HDMI_VP_STATUS)

- Address Offset: 0x0800
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 800h offset = 12_0800h



HDMI_VP_STATUS field descriptions

Field	Description
7–4 -	This field is reserved. Reserved

Table continues on the next page...

HDMI_VP_STATUS field descriptions (continued)

Field	Description
packing_phase	Read only register that holds the "packing phase" output by the Video packetizer block. For more information about "packing" video data, refer to the HDMI1.4a specification. The register is updated at tmds clock rate.

34.5.39 Video Packetizer Pixel Repetition and Color Depth Register (HDMI_VP_PR_CD)

This register configures the Color Depth of the input video and Pixel repetition to apply to video.

- Address Offset: 0x0801
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 801h offset = 12_0801h

Bit	7	6	5	4	3	2	1	0
Read Write	color_depth[3:0]				desired_pr_factor[3:0]			
Reset	0	0	0	0	0	0	0	0

HDMI_VP_PR_CD field descriptions

Field	Description
7–4 color_depth[3:0]	Color depth configuration: other Reserved. Not used. 0000 24 bits per pixel video (8 bit per component). 8-bit packing mode. 0100 24 bits per pixel video (8 bit per component). 8-bit packing mode. 0101 30 bits per pixel video (10 bit per component). 10-bit packing mode. 0110 36 bits per pixel video (12 bit per component). 12-bit packing mode. 0111 48 bits per pixel video (16 bit per component). 16-bit packing mode.
desired_pr_ factor[3:0]	Desired pixel repetition factor configuration. The configured value sets H13T PHY PLL to multiply pixel clock by the factor in order to obtain the desired repetition clock. For the CEA modes some are already defined with pixel repetition in the input video. So for CEA modes this shall be always 0. Shall only be used if the user wants to do pixel repetition using H13TCTRL core. other Reserved. Not used. 0000 No pixel repetition (pixel sent only once) 0001 Pixel sent 2 times (pixel repeated once) 0010 Pixel sent 3 times 0011 Pixel sent 4 times 0100 Pixel sent 5 times 0101 Pixel sent 6 times

Table continues on the next page...

HDMI_VP_PR_CD field descriptions (continued)

Field	Description
	0110 Pixel sent 7 times 0111 Pixel sent 8 times 1000 Pixel sent 9 times 1001 Pixel sent 10 times

34.5.40 Video Packetizer Stuffing and Default Packing Phase Register (HDMI_VP_STUFF)

This register controls the Pixel repetition, pixel packing and YCC422 stuffing.

- Address Offset: 0x0802
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 802h offset = 12_0802h

Bit	7	6	5	4
Read Write	Reserved		idefault_phase	ifix_pp_to_last
Reset	0	0	0	0

Bit	3	2	1	0
Read Write	icx_goto_p0_st	ycc422_stuffing	pp_stuffing	pr_stuffing
Reset	0	0	0	0

HDMI_VP_STUFF field descriptions

Field	Description
7–6 -	This field is reserved. Reserved
5 idefault_phase	Controls the default phase packing machine used according to: "If the transmitted video format has timing such that the phase of the first pixel of every Video Data Period corresponds to pixel packing phase 0 (for example, 10P0, 12P0, 16P0), the Source may set the Default_Phase bit in the GCP. The Sink may use this bit to optimize it's filtering or handling of the PP field." (HDMI specification version 1.4a) This means that for10 bit mode the Htotal must be dividable by 4 and for 12 bit mode the Htotal must be dividable by 2.
4 ifix_pp_to_last	Reserved. Controls packing machine strategy.
3 icx_goto_p0_st	Reserved. Controls packing machine strategy.
2 ycc422_stuffing	YCC 422 remap stuffing control. For horizontal blanking: 0 YCC 422 remap block in direct mode (input blanking data goes directly to output). 1 YCC 422 remap block in stuffing mode. When "de" goes to low the outputs are fixed to 0x00.

Table continues on the next page...

HDMI_VP_STUFF field descriptions (continued)

Field	Description
1 pp_stuffing	Pixel packing stuffing control 0 Pixel packing block in direct mode (input blanking data goes directly to output). 1 Pixel packing block in stuffing mode. When "de_rep" goes to low the outputs are fixed to 0x00.
0 pr_stuffing	Pixel repeater stuffing control 0 Pixel repeater block in direct mode (input blanking data goes directly to output). 1 Pixel repeater block in stuffing mode. When "de" goes to low the outputs are fixed to 0x00.

34.5.41 Video Packetizer YCC422 Remapping Register (HDMI_VP_REMAP)

This register controls YCC422 remap of the Video Packetizer. For more information about YCC422 remap refer to HDMI 1.4a specification.

- Address Offset: 0x0803
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 803h offset = 12_0803h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved						ycc422_size[1:0]	
Reset	0	0	0	0	0	0	0	0

HDMI_VP_REMAP field descriptions

Field	Description
7–2 -	This field is reserved. Reserved
ycc422_size[1:0]	YCC 422 remap input video size: 00 YCC 422 16-bit input video (8 bits per component). 01 YCC 422 20-bit input video (10 bits per component). 10 YCC 422 24-bit input video (12 bits per component). 11 Reserved. Not used.

34.5.42 Video Packetizer Output, Bypass, and Enable Configuration Register (HDMI_VP_CONF)

This register controls the Video Packetizer output selection, bypass select, YCC422 enable, Pixel repeater, and pixel packing enabling.

- Address Offset: 0x0804
- Size: 8 bits
- Value after Reset: 0x46
- Access: Read/Write

Address: 12_0000h base + 804h offset = 12_0804h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved	bypass_en	pp_en	pr_en	ycc422_en	BYPASS_ SELECT	output_selector[1:0]	
Reset	0	1	0	0	0	1	1	0

HDMI_VP_CONF field descriptions

Field	Description
7 -	This field is reserved. Reserved
6 bypass_en	Bypass enable. Disabling forces bypass module to output always zeros.
5 pp_en	Pixel packing enable. Disabling forces bypass module to output always zeros.
4 pr_en	Pixel repeater enable. Disabling forces bypass module to output always zeros.
3 ycc422_en	YCC 422 select enable. Disabling forces bypass module to output always zeros.
2 BYPASS_ SELECT	Bypass select bit 0 Data from pixel repeater block. 1 Data from input of video packetizer block.
output_ selector[1:0]	Video packetizer output selection. 00 Data from pixel packing block. 01 Data from YCC 422 remap block. 10 Data from 8-bit bypass block. 11 Data from 8-bit bypass block.

34.5.43 Video Packetizer Interrupt Mask Register (HDMI_VP_MASK)

Mask register for generation of VP_INT interrupts.

- Address Offset: 0x0807
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 807h offset = 12_0807h

Bit	7	6	5	4	3	2	1	0
Read Write	VPMASK7	VPMASK6	VPMASK5	VPMASK4	VPMASK3	VPMASK2	VPMASK1	VPMASK0
Reset	0	0	0	0	0	0	0	0

HDMI_VP_MASK field descriptions

Field	Description
7 VPMASK7	Mask bit for VP_INT[7] interrupt bit.
6 VPMASK6	Mask bit for VP_INT[6] interrupt bit.
5 VPMASK5	Mask bit for VP_INT[5] interrupt bit.
4 VPMASK4	Mask bit for VP_INT[4] interrupt bit.
3 VPMASK3	Mask bit for VP_INT[3] interrupt bit.
2 VPMASK2	Mask bit for VP_INT[2] interrupt bit.
1 VPMASK1	Mask bit for VP_INT[1] interrupt bit.
0 VPMASK0	Mask bit for VP_INT[0] interrupt bit.

34.5.44 Frame Composer Input Video Configuration and HDCP Keepout Register (HDMI_FC_INVIDCONF)

This register configures the Interlaced/progressive, Vblank variation and polarity of all video synchronism of the input video signal.

- Address Offset: 0x1000
- Size: 8 bits
- Value after Reset: 0x70
- Access: Read/Write

Address: 12_0000h base + 1000h offset = 12_1000h

Bit	7	6	5	4
Read Write	Reserved	vsync_in_polarity	hsync_in_polarity	de_in_polarity
Reset	0	1	1	1

Bit	3	2	1	0
Read Write	DVI_mode	Reserved	r_v_blank_in_osc	in_I_P
Reset	0	0	0	0

HDMI_FC_INVIDCONF field descriptions

Field	Description
7 -	This field is reserved. Reserved
6 vsync_in_polarity	Vsync input polarity 1 Active high 0 Active low
5 hsync_in_polarity	Hsync input polarity 1 Active high 0 Active low
4 de_in_polarity	Data enable input polarity 1 Active high 0 Active low
3 DVI_mode	Active low 0 DVI mode selected 1 HDMI mode selected
2 -	This field is reserved. Reserved
1 r_v_blank_in_osc	Used for CEA861-D modes with fractional Vblank (for example, modes 5, 6, 7, 10, 11, 20, 21, and 22. For more modes, refer to CEA861-D specification. 1 Active high
0 in_I_P	Input video mode: 1 Interlaced 0 Progressive

34.5.45 Frame Composer Input Video HActive Pixels Register 0 (HDMI_FC_INHACTIV0)

- Address Offset: 0x1001
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1001h offset = 12_1001h

Bit	7	6	5	4	3	2	1	0
Read Write	H_in_activ[7:0]							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_INHACTIV0 field descriptions

Field	Description
H_in_activ[7:0]	Input video Horizontal active pixel region width. Number of Horizontal active pixels [0...8191].

34.5.46 Frame Composer Input Video HActive Pixels Register 1 (HDMI_FC_INHACTIV1)

- Address Offset: 0x1002
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1002h offset = 12_1002h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved			H_in_activ[12:8]				
Reset	0	0	0	0	0	0	0	0

HDMI_FC_INHACTIV1 field descriptions

Field	Description
7–5 -	This field is reserved. Reserved
H_in_activ[12:8]	Input video Horizontal active pixel region width. Dependencies: Value after Reset: 0000b

Table continues on the next page...

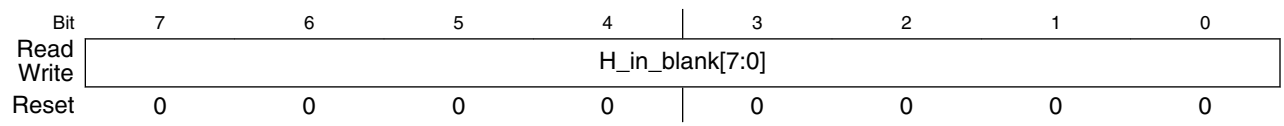
HDMI_FC_INHACTIV1 field descriptions (continued)

Field	Description
	• the higher bit of Horizontal active pixels; Number of Horizontal active pixels [0...8191].

34.5.47 Frame Composer Input Video HBlank Pixels Register 0 (HDMI_FC_INHBLANK0)

- Address Offset: 0x1003
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1003h offset = 12_1003h



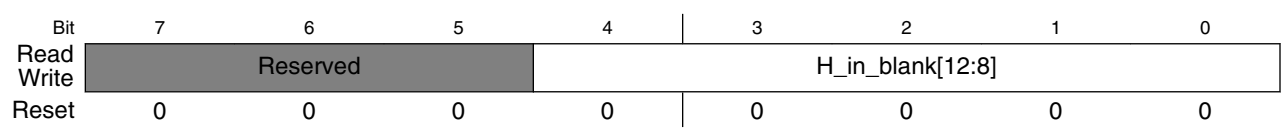
HDMI_FC_INHBLANK0 field descriptions

Field	Description
H_in_blank[7:0]	Input video Horizontal blanking pixel region width. Number of Horizontal blanking pixels [0...4095].

34.5.48 Frame Composer Input Video HBlank Pixels Register 1 (HDMI_FC_INHBLANK1)

- Address Offset: 0x1004
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1004h offset = 12_1004h



HDMI_FC_INHBLANK1 field descriptions

Field	Description
7–5 -	This field is reserved. Reserved
H_in_blank[12:8]	Input video Horizontal blanking pixel region width. Dependencies: Value after Reset: 0000b • the higher bits of Horizontal blanking pixels; Number of Horizontal blanking pixels [0...8191].

34.5.49 Frame Composer Input Video VActive Pixels Register 0 (HDMI_FC_INVACTIV0)

- Address Offset: 0x1005
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1005h offset = 12_1005h

Bit	7	6	5	4	3	2	1	0
Read Write	V_in_activ[7:0]							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_INVACTIV0 field descriptions

Field	Description
V_in_activ[7:0]	Input video Vertical active pixel region width. Number of Vertical active lines [0...4095].

34.5.50 Frame Composer Input Video VActive Pixels Register 1 (HDMI_FC_INVACTIV1)

- Address Offset: 0x1006
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1006h offset = 12_1006h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				V_in_activ[12:8]			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_INVACTIV1 field descriptions

Field	Description
7–5 -	This field is reserved. Reserved
V_in_activ[12:8]	Input video Vertical active pixel region width. Dependencies: Value after Reset: 0000b • the higher 5 bits of Vertical active line; Number of Vertical active lines [0...8191].

34.5.51 Frame Composer Input Video VBlank Pixels Register (HDMI_FC_INVBLANK)

- Address Offset: 0x1007
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1007h offset = 12_1007h

Bit	7	6	5	4	3	2	1	0
Read Write	V_in_blank[7:0]							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_INVBLANK field descriptions

Field	Description
V_in_blank[7:0]	Input video Vertical blanking pixel region width. Number of Vertical blanking lines [0...255]. Value after Reset: 0x00

34.5.52 Frame Composer Input Video HSync Front Porch Register 0 (HDMI_FC_HSYNCINDELAY0)

- Address Offset: 0x1008
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1008h offset = 12_1008h

Bit	7	6	5	4	3	2	1	0
Read Write	H_in_delay[7:0]							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_HSYNCINDELAY0 field descriptions

Field	Description
H_in_delay[7:0]	Input video Hsync active edge delay. Integer number of pixel clock cycles from "de" non active edge of the last "de" valid period [0...4095].

34.5.53 Frame Composer Input Video HSync Front Porch Register 1 (HDMI_FC_HSYNCINDELAY1)

- Address Offset: 0x1009
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1009h offset = 12_1009h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				H_in_delay[12:8]			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_HSYNCINDELAY1 field descriptions

Field	Description
7–5 -	This field is reserved. Reserved
H_in_delay[12:8]	Input video Hsync active edge delay. Dependencies: Value after Reset: 0000b • the higher 5 bits of delay; Integer number of pixel clock cycles from "de" non active edge of the last "de" valid period [0...8191].

34.5.54 Frame Composer Input Video HSync Width Register 0 (HDMI_FC_HSYNCINWIDTH0)

- Address Offset: 0x100A
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 100Ah offset = 12_100Ah

Bit	7	6	5	4	3	2	1	0
Read Write	H_in_width[7:0]							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_HSYNCINWIDTH0 field descriptions

Field	Description
H_in_width[7:0]	Input video Hsync active pulse width. Integer number of pixel clock cycles [0...511].

34.5.55 Frame Composer Input Video HSync Width Register 1 (HDMI_FC_HSYNCINWIDTH1)

- Address Offset: 0x100B
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 100Bh offset = 12_100Bh

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved						H_in_width[9:8]	
Reset	0	0	0	0	0	0	0	0

HDMI_FC_HSYNCINWIDTH1 field descriptions

Field	Description
7–2 -	This field is reserved. Reserved
H_in_width[9:8]	Input video Hsync active pulse width. Dependencies: Value after Reset after Reset: 0b • Integer number of pixel clock cycles [0...1024].

34.5.56 Frame Composer Input Video VSync Front Porch Register (HDMI_FC_VSYNCINDELAY)

- Address Offset: 0x100C
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 100Ch offset = 12_100Ch

Bit	7	6	5	4	3	2	1	0
Read Write	V_in_delay[7..0]							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSYNCINDELAY field descriptions

Field	Description
V_in_delay[7..0]	Input video Vsync active edge delay. Integer number of Hsync pulses from "de" non active edge of the last "de" valid period. [0...255].

34.5.57 Frame Composer Input Video VSync Width Register (HDMI_FC_VSYNCINWIDTH)

- Address Offset: 0x100D
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 100Dh offset = 12_100Dh

Bit	7	6	5	4	3	2	1	0
Read / Write	Reserved				V_in_width[5..0]			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSYNCINWIDTH field descriptions

Field	Description
7–6 -	This field is reserved. Reserved
V_in_width[5..0]	Value after Reset: 000000b Input video Vsync active pulse width: Integer number of pixel clock cycles [0...63].

34.5.58 Frame Composer Input Video Refresh Rate Register 0 (HDMI_FC_INFREQ0)

- Address Offset: 0x100E
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 100Eh offset = 12_100Eh

Bit	7	6	5	4	3	2	1	0
Read / Write	infreq[7:0]							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_INFREQ0 field descriptions

Field	Description
infreq[7:0]	Video refresh rate in Hz*1E3 format. This registers are provided for debug and informative purposes. No data is written to this registers by the H13TCTRL and the data here written by software is not used in any way by the H13TCTRL.

34.5.59 Frame Composer Input Video Refresh Rate Register 1 (HDMI_FC_INFREQ1)

- Address Offset: 0x100F
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 100Fh offset = 12_100Fh

Bit	7	6	5	4	3	2	1	0
Read Write	infreq[7:0]							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_INFREQ1 field descriptions

Field	Description
infreq[7:0]	Video refresh rate in Hz*1E3 format. This registers are provided for debug and informative purposes. No data is written to this registers by the H13TCTRL and the data here written by software is not used in any way by the H13TCTRL.

34.5.60 Frame Composer Input Video Refresh Rate Register 2 (HDMI_FC_INFREQ2)

- Address Offset: 0x1010
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1010h offset = 12_1010h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				infreq[19:16]			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_INFREQ2 field descriptions

Field	Description
7–4 -	This field is reserved. Reserved
infreq[19:16]	Video refresh rate in Hz*1E3 format. This registers are provided for debug and informative purposes. No data is written to this registers by the H13TCTRL and the data here written by software is not used in any way by the H13TCTRL. Value after Reset: 0000b

34.5.61 Frame Composer Control Period Duration Register (HDMI_FC_CTRLDUR)

- Address Offset: 0x1011
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1011h offset = 12_1011h

Bit	7	6	5	4	3	2	1	0
Read Write	ctrlperiodduration							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_CTRLDUR field descriptions

Field	Description
ctrlperiodduration	Configuration of the control period minimum duration (min. of 12 pixel clock cycles, refer to HDMI 1.4a specification). Integer number of pixel clocks cycles [0..255].

34.5.62 Frame Composer Extended Control Period Duration Register (HDMI_FC_EXCTRLDUR)

- Address Offset: 0x1012
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1012h offset = 12_1012h

Bit	7	6	5	4	3	2	1	0
Read Write	exctrlperiodduration							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_EXCTRLDUR field descriptions

Field	Description
exctrlperiodduration	Configuration of the extended control period minimum duration (min. of 32 pixel clock cycles, see HDMI 1.4a specification). Integer number of pixel clocks cycles [0..255].

34.5.63 Frame Composer Extended Control Period Maximum Spacing Register (HDMI_FC_EXCTRLSPAC)

- Address Offset: 0x1013
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1013h offset = 12_1013h

Bit	7	6	5	4	3	2	1	0
Read Write	exctrlperiodspacing							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_EXCTRLSPAC field descriptions

Field	Description
exctrlperiodspacing	Configuration of the maximum spacing between consecutive extended control periods (max of 50msec, see HDMI 1.4a specification): generated spacing = (1/freq tmds clock)*256*256*extctrlperiodspacing

34.5.64 Frame Composer Channel 0 Non-Preamble Data Register (HDMI_FC_CH0PREAM)

- Address Offset: 0x1014
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1014h offset = 12_1014h

Bit	7	6	5	4	3	2	1	0
Read Write	ch0_preamble_filter							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_CH0PREAM field descriptions

Field	Description
ch0_preamble_filter	When in control mode, configures 8-bits that are going to fill the channel 0 data lines not used to transmit the preamble (for more clarifications refer to HDMI 1.4a specification).

34.5.65 Frame Composer Channel 1 Non-Preamble Data Register (HDMI_FC_CH1PREAM)

- Address Offset: 0x1015
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1015h offset = 12_1015h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved		ch1_preamble_filter					
Reset	0	0	0	0	0	0	0	0

HDMI_FC_CH1PREAM field descriptions

Field	Description
7–6 -	This field is reserved. Reserved
ch1_preamble_filter	When in control mode, configures 6-bits that are going to fill the channel 1 data lines not used to transmit the preamble (for more clarifications refer to HDMI 1.4a specification).

34.5.66 Frame Composer Channel 2 Non-Preamble Data Register (HDMI_FC_CH2PREAM)

- Address Offset: 0x1016
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1016h offset = 12_1016h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				ch2_preamble_filter			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_CH2PREAM field descriptions

Field	Description
7–6 -	This field is reserved. Reserved
ch2_preamble_ filter	When in control mode, configures 6-bits that are going to fill the channel 2 data lines not used to transmit the preamble (for more clarifications, see HDMI 1.4a specification).

34.5.67 Frame Composer AVI Configuration Register 3 (HDMI_FC_AVICONF3)

- Address Offset: 0x1017
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

configuration of Quantization range and IT content type.

Address: 12_0000h base + 1017h offset = 12_1017h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				YQ1_YQ0_YCC		CN1_CN0	
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AVICONF3 field descriptions

Field	Description
7–4 -	This field is reserved. Reserved

Table continues on the next page...

HDMI_FC_AVICONF3 field descriptions (continued)

Field	Description
3–2 YQ1_YQ0_YCC	Quantization range according to CEA specification.
CN1_CN0	IT content type according to CEA specification

34.5.68 Frame Composer GCP Packet Configuration Register (HDMI_FC_GCP)

Configures the General Control Packet A/V mute indicators and the default phase.

- Address Offset: 0x1018
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1018h offset = 12_1018h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved					default_ phase	set_avmute	clear_ avmute
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GCP field descriptions

Field	Description
7–3 -	This field is reserved. Reserved
2 default_phase	Value of "default_phase" in the GCP packet. This data should be equal to the default phase used at Video packetizer packing machine. Value after Reset: 0b
1 set_avmute	Value of "set_avmute" in the GCP packet. Value after Reset: 0b
0 clear_avmute	Value of "clear_avmute" in the GCP packet. Value after Reset: 0b

34.5.69 Frame Composer AVI Packet Configuration Register 0 (HDMI_FC_AVICONF0)

Configures the following contents of the AVI infoFrame:

- RGB/YCC indication
- Bar information

- Scan information
- Active format present
- Progressive/Interlaced indicator
- Active aspect ratio
- Picture aspect ratio
- Colorimetry
- IT content
- Extended colorimetry
- Quantization range
- Non-uniform picture scaling

For more information, refer to HDMI 1.4a and CEA - 861D specifications.

- Address Offset: 0x1019
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1019h offset = 12_1019h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_AVICONF0_MISC	FC_AVICONF0_ACTIVE_FORMAT	FC_AVICONF0_SCAN		FC_AVICONF0_BAR		FC_AVICONF0_RGB_YCC	
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AVICONF0 field descriptions

Field	Description
7 FC_AVICONF0_MISC	Frame composer AVI packet configuration bit
6 FC_AVICONF0_ACTIVE_FORMAT	Active format present
5–4 FC_AVICONF0_SCAN	Scan information
3–2 FC_AVICONF0_BAR	Bar information
FC_AVICONF0_RGB_YCC	RGB/YCC indication Value after Reset: 0b

34.5.70 Frame Composer AVI Packet Configuration Register 1 (HDMI_FC_AVICONF1)

- Address Offset: 0x101A
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 101Ah offset = 12_101Ah

Bit	7	6	5	4	3	2	1	0
Read Write	FC_AVICONF0_COLOR		FC_AVICONF1_ PICTURE_AR		FC_AVICONF1_ACTIVE_AR			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AVICONF1 field descriptions

Field	Description
7–6 FC_AVICONF0_ COLOR	Colorimetry
5–4 FC_AVICONF1_ PICTURE_AR	Picture aspect ratio
FC_AVICONF1_ ACTIVE_AR	Active aspect ratio Value after Reset: 0b

34.5.71 FC_AVICONFFrame Composer AVI Packet Configuration Register 2 (HDMI_FC_AVICONF2)

- Address Offset: 0x101B
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 101Bh offset = 12_101Bh

Bit	7	6	5	4	3	2	1	0
Read Write	FC_ AVICONF2_ IT	FC_AVICONF2_EXT_COLOR			-		FC_AVICONF2_SCALE	
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AVICONF2 field descriptions

Field	Description
7 FC_AVICONF2_IT	IT content
6–4 FC_AVICONF2_EXT_COLOR	Extended colorimetry
3–2 -	Quantization range
FC_AVICONF2_SCALE	Non-uniform picture scaling Value after Reset: 0b

34.5.72 Frame Composer AVI Packet VIC Register (HDMI_FC_AVIVID)

Configures the AVI infoFrame Video Identification code. For more information, refer to the CEA-861-E specification.

- Address Offset: 0x101C
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 101Ch offset = 12_101Ch

Bit	7	6	5	4	3	2	1	0
Read Write	FC_AVIVID							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AVIVID field descriptions

Field	Description
FC_AVIVID	the AVI infoFrame Video Identification code.

34.5.73 Frame Composer AVI Packet End of Top Bar Register 0 (HDMI_FC_AVIETB0)

These registers define the AVI infoFrame End of Top Bar value. For more information, refer to CEA-861-E specification.

- Address Offset: 0x101D
- Size: 8 bits

- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 101Dh offset = 12_101Dh

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AVIETB0 field descriptions

Field	Description
-	Line number of end of top bar (lower 8 bits)

34.5.74 Frame Composer AVI Packet End of Top Bar Register 1 (HDMI_FC_AVIETB1)

- Address Offset: 0x101E
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 101Eh offset = 12_101Eh

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AVIETB1 field descriptions

Field	Description
-	Line number of end of top bar (upper 8 bits)

34.5.75 Frame Composer AVI Packet Start of Bottom Bar Register 0 (HDMI_FC_AVISBB0)

These registers define the AVI infoFrame Start of Bottom Bar value. For more information, refer to CEA-861D specification.

- Address Offset: 0x101F
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 101Fh offset = 12_101Fh

Bit	7	6	5	4	3	2	1	0
Read Write	FC_AVISBB0							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AVISBB0 field descriptions

Field	Description
FC_AVISBB0	Line number of Start of Bottom Bar (lower 8 bits)

34.5.76 Frame Composer AVI Packet Start of Bottom Bar Register 1 (HDMI_FC_AVISBB1)

- Address Offset: 0x1020
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1020h offset = 12_1020h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_AVISBB1							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AVISBB1 field descriptions

Field	Description
FC_AVISBB1	Line number of Start of Bottom Bar (upper 8 bits)

34.5.77 Frame Composer AVI Packet End of Left Bar Register 0 (HDMI_FC_AVIELB0)

These registers define the AVI infoFrame End of Left Bar value. For more information, refer to CEA-861D specification.

- Address Offset: 0x1021
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1021h offset = 12_1021h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_AVIELB0							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AVIELB0 field descriptions

Field	Description
FC_AVIELB0	Pixel number of end of left Bar (lower 8 bits)

34.5.78 Frame Composer AVI Packet End of Left Bar Register 1 (HDMI_FC_AVIELB1)

- Address Offset: 0x1022
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1022h offset = 12_1022h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_AVIELB1							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AVIELB1 field descriptions

Field	Description
FC_AVIELB1	Pixel number of end of left Bar (lower 8 bits)

34.5.79 Frame Composer AVI Packet Start of Right Bar Register 0 (HDMI_FC_AVISRB0)

These registers define the AVI infoFrame Start of Right Bar value. For more information, refer to CEA-861D specification.

- Address Offset: 0x1023
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1023h offset = 12_1023h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_AVISRB0							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AVISRB0 field descriptions

Field	Description
FC_AVISRB0	Pixel number of start of right Bar (lower 8 bits)

34.5.80 Frame Composer AVI Packet Start of Right Bar Register 1 (HDMI_FC_AVISRB1)

- Address Offset: 0x1024
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1024h offset = 12_1024h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_AVISRB1							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AVISRB1 field descriptions

Field	Description
FC_AVISRB1	Pixel number of start of right Bar (upper 8 bits)

34.5.81 Frame Composer AUD Packet Configuration Register 0 (HDMI_FC_AUDICONF0)

These registers configure the following contents of the AUDIO infoFrame:

- Coding type
- Channel count
- Sampling frequency
- Sampling size
- Channel allocation
- Audio level shift value
- Down mix enable

For more information, refer to CEA-861D specification.

- Address Offset: 0x1025 to 0x1028
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For the FC_AUDICONF0 register, bits [6:5] correspond to LFEPBL1, LFEPBL0 LFE playback level as compared to the other channels (from HDMI 1.4a specification).

Address: 12_0000h base + 1025h offset = 12_1025h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved	CC[2:0]			CT[3:0]			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AUDICONF0 field descriptions

Field	Description
7 -	This field is reserved. Reserved
6–4 CC[2:0]	Channel count
CT[3:0]	Coding Type

34.5.82 Frame Composer AUD Packet Configuration Register 1 (HDMI_FC_AUDICONF1)

Address: 12_0000h base + 1026h offset = 12_1026h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved		SS[1:0]		Reserved	SF[2:0]		
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AUDICONF1 field descriptions

Field	Description
7–6 -	This field is reserved. Reserved
5–4 SS[1:0]	Sampling size
3 -	This field is reserved. Reserved
SF[2:0]	Sampling frequency

34.5.83 Frame Composer AUD Packet Configuration Register 2 (HDMI_FC_AUDICONF2)

Address: 12_0000h base + 1027h offset = 12_1027h

Bit	7	6	5	4	3	2	1	0
Read Write	CA[7:0]							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AUDICONF2 field descriptions

Field	Description
CA[7:0]	Channel allocation

34.5.84 Frame Composer AUD Packet Configuration Register 3 (HDMI_FC_AUDICONF3)

Address: 12_0000h base + 1028h offset = 12_1028h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved	LFEPBL[1:0]		DM_INH	LSV[3:0]			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AUDICONF3 field descriptions

Field	Description
7 -	This field is reserved. Reserved
6–5 LFEPBL[1:0]	LFE playback information
4 DM_INH	Down mix enable
LSV[3:0]	Level shift value (for down mixing)

34.5.85 Frame Composer VSI Packet Data IEEE Register 0 (HDMI_FC_VSDIEEEID0)

These registers configure the Vendor Specific infoFrame IEEE registration identifier. For more information, refer to CEA-861D specification.

- Address Offset: 0x1029
- Size: 8 bits

- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1029h offset = 12_1029h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDIEEEID0 field descriptions

Field	Description
-	the Vendor Specific infoFrame IEEE registration identifier byte 0

34.5.86 Frame Composer VSI Packet Data Size Register (HDMI_FC_VSDSIZE)

- Address Offset: 0x102A
- Size: 8 bits
- Value after Reset: 0x1B
- Access: Read/Write

configuration of Packet size.

Address: 12_0000h base + 102Ah offset = 12_102Ah

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved			VSDSIZE				
Reset	0	0	0	1	1	0	1	1

HDMI_FC_VSDSIZE field descriptions

Field	Description
7–5 -	This field is reserved. Reserved
VSDSIZE	Packet size as described in HDMI Vendor Specific InfoFrame (from HDMI specification).

34.5.87 Frame Composer VSI Packet Data IEEE Register 1 (HDMI_FC_VSDIEEEID1)

- Address Offset: 0x102a
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1030h offset = 12_1030h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDIEEEID1 field descriptions

Field	Description
-	the Vendor Specific infoFrame IEEE registration identifier byte 1

34.5.88 Frame Composer VSI Packet Data IEEE Register 2 (HDMI_FC_VSDIEEEID2)

- Address Offset: 0x102b
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1031h offset = 12_1031h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDIEEEID2 field descriptions

Field	Description
-	the Vendor Specific infoFrame IEEE registration identifier byte 2

34.5.89 Frame Composer VSI Packet Data IEEE Register 0 (HDMI_FC_VSDPAYLOAD0)

These registers configure the Vendor Specific infoFrame 24 bytes specific payload. For more information, refer to CEA-861D specification.

- Address Offset: 0x1032
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1032h offset = 12_1032h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD0 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte0

34.5.90 Frame Composer VSI Packet Data IEEE Register 1 (HDMI_FC_VSDPAYLOAD1)

- Address Offset: 0x1033
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1033h offset = 12_1033h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD1 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte1

34.5.91 Frame Composer VSI Packet Data IEEE Register 2 (HDMI_FC_VSDPAYLOAD2)

- Address Offset: 0x1034
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1034h offset = 12_1034h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD2 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte2

34.5.92 Frame Composer VSI Packet Data IEEE Register 3 (HDMI_FC_VSDPAYLOAD3)

- Address Offset: 0x1035
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1035h offset = 12_1035h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD3 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte3

34.5.93 Frame Composer VSI Packet Data IEEE Register 4 (HDMI_FC_VSDPAYLOAD4)

- Address Offset: 0x1036
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1036h offset = 12_1036h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD4 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte4

34.5.94 Frame Composer VSI Packet Data IEEE Register 5 (HDMI_FC_VSDPAYLOAD5)

- Address Offset: 0x1037
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1037h offset = 12_1037h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD5 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte5

34.5.95 Frame Composer VSI Packet Data IEEE Register 6 (HDMI_FC_VSDPAYLOAD6)

- Address Offset: 0x1038
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1038h offset = 12_1038h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD6 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte6

34.5.96 Frame Composer VSI Packet Data IEEE Register 7 (HDMI_FC_VSDPAYLOAD7)

- Address Offset: 0x1039
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1039h offset = 12_1039h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD7 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte7

34.5.97 Frame Composer VSI Packet Data IEEE Register 8 (HDMI_FC_VSDPAYLOAD8)

- Address Offset: 0x103a
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 103Ah offset = 12_103Ah

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD8 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte8

34.5.98 Frame Composer VSI Packet Data IEEE Register 9 (HDMI_FC_VSDPAYLOAD9)

- Address Offset: 0x103b
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 103Bh offset = 12_103Bh

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD9 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte9

34.5.99 Frame Composer VSI Packet Data IEEE Register 10 (HDMI_FC_VSDPAYLOAD10)

- Address Offset: 0x103c
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 103Ch offset = 12_103Ch

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD10 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte10

34.5.100 Frame Composer VSI Packet Data IEEE Register 11 (HDMI_FC_VSDPAYLOAD11)

- Address Offset: 0x103d
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 103Dh offset = 12_103Dh

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD11 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte11

34.5.101 Frame Composer VSI Packet Data IEEE Register 12 (HDMI_FC_VSDPAYLOAD12)

- Address Offset: 0x103e
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 103Eh offset = 12_103Eh

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD12 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte12

34.5.102 Frame Composer VSI Packet Data IEEE Register 13 (HDMI_FC_VSDPAYLOAD13)

- Address Offset: 0x103f
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 103Fh offset = 12_103Fh

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD13 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte13

34.5.103 Frame Composer VSI Packet Data IEEE Register 14 (HDMI_FC_VSDPAYLOAD14)

- Address Offset: 0x1040
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1040h offset = 12_1040h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD14 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte14

34.5.104 Frame Composer VSI Packet Data IEEE Register 15 (HDMI_FC_VSDPAYLOAD15)

- Address Offset: 0x1041
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1041h offset = 12_1041h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD15 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte15

34.5.105 Frame Composer VSI Packet Data IEEE Register 16 (HDMI_FC_VSDPAYLOAD16)

- Address Offset: 0x1042
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1042h offset = 12_1042h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD16 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte16

34.5.106 Frame Composer VSI Packet Data IEEE Register 17 (HDMI_FC_VSDPAYLOAD17)

- Address Offset: 0x1043
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1043h offset = 12_1043h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD17 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte17

34.5.107 Frame Composer VSI Packet Data IEEE Register 18 (HDMI_FC_VSDPAYLOAD18)

- Address Offset: 0x1044
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1044h offset = 12_1044h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD18 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte18

34.5.108 Frame Composer VSI Packet Data IEEE Register 19 (HDMI_FC_VSDPAYLOAD19)

- Address Offset: 0x1045
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1045h offset = 12_1045h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

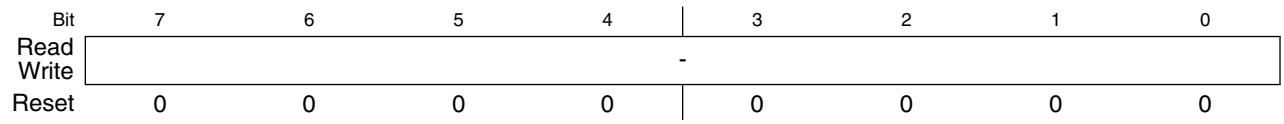
HDMI_FC_VSDPAYLOAD19 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte19

34.5.109 Frame Composer VSI Packet Data IEEE Register 20 (HDMI_FC_VSDPAYLOAD20)

- Address Offset: 0x1046
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1046h offset = 12_1046h



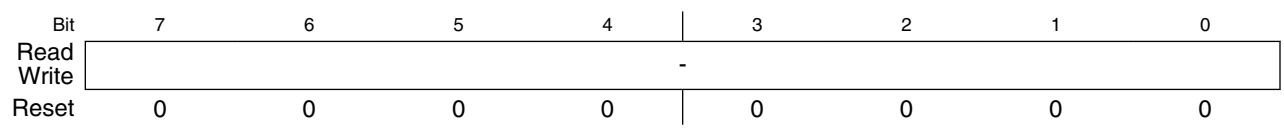
HDMI_FC_VSDPAYLOAD20 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte20

34.5.110 Frame Composer VSI Packet Data IEEE Register 21 (HDMI_FC_VSDPAYLOAD21)

- Address Offset: 0x1047
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1047h offset = 12_1047h



HDMI_FC_VSDPAYLOAD21 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte21

34.5.111 Frame Composer VSI Packet Data IEEE Register 22 (HDMI_FC_VSDPAYLOAD22)

- Address Offset: 0x1048
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1048h offset = 12_1048h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD22 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte22

34.5.112 Frame Composer VSI Packet Data IEEE Register 23 (HDMI_FC_VSDPAYLOAD23)

- Address Offset: 0x1049
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1049h offset = 12_1049h

Bit	7	6	5	4	3	2	1	0
Read Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_VSDPAYLOAD23 field descriptions

Field	Description
-	the Vendor Specific infoFrame 24 bytes specific payload byte23

34.5.113 Frame Composer SPD Packet Data Vendor Name Register 0 (HDMI_FC_SPDVENDORNAME0)

These registers configure the Source Product Descriptor infoFrame 8 bytes Vendor name. For more information, refer to CEA-861D specification.

- Address Offset: 0x104A to 0x1051
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 104Ah offset = 12_104Ah

Bit	7	6	5	4	3	2	1	0
Read Write	vendor_name							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_SPDVENDORNAME0 field descriptions

Field	Description
vendor_name	Vendor name

34.5.114 Frame Composer SPD Packet Data Product Name Register 0 (HDMI_FC_SPDPRODUCTNAME0)

These registers configure the Source Product Descriptor infoFrame 16 bytes Product name. For more information, refer to CEA-861D specification.

- Address Offset: 0x1052 to 0x1061
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1052h offset = 12_1052h

Bit	7	6	5	4	3	2	1	0
Read Write	product_name							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_SPDPRODUCTNAME0 field descriptions

Field	Description
product_name	Product name

34.5.115 Frame Composer SPD Packet Data Source Product Descriptor Register (HDMI_FC_SPDDEVICEINF)

This register configures Source Product Descriptor infoFrame description device field. For more information, refer to CEA-861D specification.

- Address Offset: 0x1062
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1062h offset = 12_1062h

Bit	7	6	5	4	3	2	1	0
Read Write	product_descriptor							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_SPDDEVICEINF field descriptions

Field	Description
product_ descriptor	Product descriptor

34.5.116 Frame Composer Audio Sample Flat and Layout Configuration Register (HDMI_FC_AUDSCONF)

Configures the Audio sample packet sample flat and layout configuration. For more information, refer to HDMI 1.4a specification.

- Address Offset: 0x1063
- Size: 8 bits
- Value after Reset: 0x00
- AEccess: Read/Write

Address: 12_0000h base + 1063h offset = 12_1063h

Bit	7	6	5	4	3	2	1	0
Read Write	aud_packet_sampfit[3:0]				Reserved			aud_ packet_ layout
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AUDSCONF field descriptions

Field	Description
7–4 aud_packet_ sampfit[3:0]	Set the audio packet sample flat value to be sent on the packet.
3–1 -	This field is reserved. Reserved
0 aud_packet_ layout	Set the audio packet layout to be sent in the packet: 1 layout 1 0 layout 0

34.5.117 Frame Composer Audio Packet Sample Present Status Register (HDMI_FC_AUDSSTAT)

Shows the data sample present indication of the last Audio sample packet sent by the HDMI TX Controller. For more information, refer to HDMI 1.4a specification.

- Address Offset: 0x1064
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 1064h offset = 12_1064h

Bit	7	6	5	4	3	2	1	0
Read	Reserved				packet_sampprs[3:0]			
Write								
Reset	0	0	0	0	0	0	0	0

HDMI_FC_AUDSSTAT field descriptions

Field	Description
7–4 -	This field is reserved. Reserved
packet_ sampprs[3:0]	Shows the data sample present indication of the last Audio sample packet sent by the HDMI TX Controller. This register information is at tmds clock rate.

34.5.118 Frame Composer Number of High Priority Packets Attended Configuration Register (HDMI_FC_CTRLQHIGH)

- Address Offset: 0x1073
- Size: 8 bits
- Value after Reset: 0x0F
- Access: Read/Write

Address: 12_0000h base + 1073h offset = 12_1073h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				onhighattended[4:0]			
Reset	0	0	0	0	1	1	1	1

HDMI_FC_CTRLQHIGH field descriptions

Field	Description
7–5 -	This field is reserved. Reserved
onhighattended[4:0]	Configures the number of high priority packets or audio sample packets consecutively attended before checking low priority queue status. Integer number [0..31]

34.5.119 Frame Composer Number of Low Priority Packets Attended Configuration Register (HDMI_FC_CTRLQLOW)

- Address Offset: 0x1074
- Size: 8 bits
- Value after Reset: 0x03
- Access: Read/Write

Address: 12_0000h base + 1074h offset = 12_1074h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				onlowattended[4:0]			
Reset	0	0	0	0	0	0	1	1

HDMI_FC_CTRLQLOW field descriptions

Field	Description
7–5 -	This field is reserved. Reserved
onlowattended[4:0]	Configures the number of low priority packets or null packets consecutively attended before checking high priority queue status or audio sample availability. Integer number [0..31]

34.5.120 Frame Composer ACP Packet Type Configuration Register 0 (HDMI_FC_ACP0)

Configures the following contents of the ACP packet. For more information, refer to the HDMI 1.4 specification.

- Address Offset: 0x1075
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1075h offset = 12_1075h

Bit	7	6	5	4	3	2	1	0
Read Write	acptype[7:0]							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_ACP0 field descriptions

Field	Description
acptype[7:0]	Configures the ACP packet type.

34.5.121 Frame Composer ACP Packet Type Configuration Register 1 (HDMI_FC_ACP1)

Configures the following contents of the Audio Content Packet (ACP) body:

- Address Offset: 0x1091 to 0x1082
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1091h offset = 12_1091h

Bit	7	6	5	4	3	2	1	0
Read Write	Audio_contentpacket							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_ACP1 field descriptions

Field	Description
Audio_ contentpacket	Audio content packet

34.5.122 FC_ISCR1_Frame Composer Packet Status, Valid, and Continue Configuration Register (HDMI_FC_ISCR1_0)

Configures the following contents of the ISRC1 packet:

- Address Offset: 0x1092
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, see the HDMI 1.4 specification.

Address: 12_0000h base + 1092h offset = 12_1092h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved			isrc_status[2:0]			isrc_valid	isrc_cont
Reset	0	0	0	0	0	0	0	0

HDMI_FC_ISCR1_0 field descriptions

Field	Description
7–5 -	This field is reserved. Reserved
4–2 isrc_status[2:0]	Status of ISRC1.
1 isrc_valid	Valid of ISRC1.
0 isrc_cont	Indication of ISRC2.

34.5.123 Frame Composer ISCR1 Packet Body Register 1 (HDMI_FC_ISCR1_1)

Configures the following contents of the ISRC1 packet:

- ISRC1 packet body
- Address Offset: 0x10A2 to 0x1093
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, see the HDMI 1.4 specification.

Address: 12_0000h base + 1093h offset = 12_1093h

Bit	7	6	5	4	3	2	1	0
Read Write	isrc1							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_ISCR1_1 field descriptions

Field	Description
isrc1	Configures the contents of the ISRC1 packet:

34.5.124 Frame Composer ISCR2 Packet Body Register 0 (HDMI_FC_ISCR2_0)

Configures the following contents of the ISRC2 packet:

- ISRC2 packet body
- Address Offset: 0x10B2 to 0x10A3
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, see the HDMI 1.4 specification.

Address: 12_0000h base + 10A3h offset = 12_10A3h

Bit	7	6	5	4	3	2	1	0
Read Write	isrc2							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_ISCR2_0 field descriptions

Field	Description
isrc2	Configures the contents of the ISRC1 packet:

34.5.125 Frame Composer Data Island Auto Packet Scheduling Register 0 (HDMI_FC_DATAUTO0)

Configures the Frame Composer RDRB(1)/Manual(0) data island packet insertion for SPD, VSD, ISRC2, ISRC1 and ACP packets. On RDRB mode the described packet scheduling is controlled by registers FC_DATAUTO1 and FC_DATAUTO2, while in Manual mode register FC_DATMAN requests to FC the insertion of the requested packet.

- Address Offset: 0x10B3
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 10B3h offset = 12_10B3h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved			spd_auto	vsd_auto	iscr2_auto	iscr1_auto	acp_auto
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DATAUTO0 field descriptions

Field	Description
7–5 -	This field is reserved. Reserved
4 spd_auto	Enables SPD automatic packet scheduling
3 vsd_auto	Enables VSD automatic packet scheduling
2 iscr2_auto	Enables ISRC2 automatic packet scheduling
1 iscr1_auto	Enables ISRC1 automatic packet scheduling
0 acp_auto	Enables ACP automatic packet scheduling

34.5.126 Frame Composer Data Island Auto Packet Scheduling Register 1 (HDMI_FC_DATAUTO1)

Configures the Frame Composer (FC) RDRB frame interpolation for SPD, VSD, ISRC2, ISRC1 and ACP packet insertion on data island when FC is on RDRB mode for the listed packets.

- Address Offset: 0x10B4

- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 10B4h offset = 12_10B4h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				AUTO_FRAME_INTERPOLATION			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DATAUTO1 field descriptions

Field	Description
7–4 -	This field is reserved. Reserved
AUTO_FRAME_ INTERPOLATION	Packet frame interpolation, for automatic packet scheduling

34.5.127 Frame Composer Data Island Auto Packet Scheduling Register 2 (HDMI_FC_DATAUTO2)

Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for SPD, VSD, ISRC2, ISRC1 and ACP packet insertion on data island when FC is on RDRB mode for the listed packets.

- Address Offset: 0x10B5
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 10B5h offset = 12_10B5h

Bit	7	6	5	4	3	2	1	0
Read Write	AUTO_FRAME_PACKETS				AUTO_LINE_SPACING			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DATAUTO2 field descriptions

Field	Description
7–4 AUTO_FRAME_ PACKETS	Packets per frame, for automatic packet scheduling
AUTO_LINE_ SPACING	Packets line spacing, for automatic packet scheduling

34.5.128 Frame Composer Data Island Manual Packet Request Register (HDMI_FC_DATMAN)

Requests to the Frame Composer the data island packet insertion for NULL, SPD, VSD, ISRC2, ISRC1 and ACP packets when FC_DATAUTO0 bit is in manual mode for the packet requested.

- Address Offset: 0x10B6
- Size: 8 bits
- Value after Reset: 0x00
- Access: Write

Address: 12_0000h base + 10B6h offset = 12_10B6h

Bit	7	6	5	4	3	2	1	0
Read	Reserved							
Write			null_tx	spd_tx	vsd_tx	iscr2_tx	isr1_tx	acp_tx
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DATMAN field descriptions

Field	Description
7–6 -	This field is reserved. Reserved
5 null_tx	Null packet
4 spd_tx	SPD packet
3 vsd_tx	VSD packet
2 iscr2_tx	ISRC2 packet
1 isr1_tx	ISRC1 packet
0 acp_tx	ACP packet

34.5.129 Frame Composer Data Island Auto Packet Scheduling Register 3 (HDMI_FC_DATAUTO3)

Configures the Frame Composer Automatic(1)/RDRB(0) data island packet insertion for AVI, GCP, AUDI and ACR packets. In Automatic mode, the packet will be inserted on Vblanking when first line with active Vsync appears.

- Address Offset: 0x10B7
- Size: 8 bits
- Value after Reset: 0x0F
- Access: Read/Write

Address: 12_0000h base + 10B7h offset = 12_10B7h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				avi_auto	gcp_auto	audi_auto	acr_auto
Reset	0	0	0	0	1	1	1	1

HDMI_FC_DATAUTO3 field descriptions

Field	Description
7–4 -	This field is reserved. Reserved
3 avi_auto	Enable AVI packet insertion
2 gcp_auto	Enable GCP packet insertion
1 audi_auto	Enable AUDI packet insertion
0 acr_auto	Enable ACR packet insertion

34.5.130 Frame Composer Round Robin ACR Packet Insertion Register 0 (HDMI_FC_RDRB0)

Configures the Frame Composer (FC) RDRB frame interpolation for ACR packet insertion on data island when FC is on RDRB mode for this packet.

- Address Offset: 0x10B8
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 10B8h offset = 12_10B8h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				ACRframeinterpolation			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_RDRB0 field descriptions

Field	Description
7–4 -	This field is reserved. Reserved
ACRframeinterpolation	ACR frame interpolation

34.5.131 Frame Composer Round Robin ACR Packet Insertion Register 1 (HDMI_FC_RDRB1)

Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the ACR packet insertion on data island when FC is on RDRB mode this packet.

- Address Offset: 0x10B9
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 10B9h offset = 12_10B9h

Bit	7	6	5	4	3	2	1	0
Read Write	ACRpacketsinframe				ACRpacketlinespacing			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_RDRB1 field descriptions

Field	Description
7–4 ACRpacketsinframe	ACR packets in frame
ACRpacketlinespacing	ACR packet line spacing

34.5.132 Frame Composer Round Robin ACR Packet Insertion Register 2 (HDMI_FC_RDRB2)

Configures the Frame Composer (FC) RDRB frame interpolation for AUDI packet insertion on data island when FC is on RDRB mode for this packet.

- Address Offset: 0x10BA
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 10BAh offset = 12_10BAh

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				AUDIframeinterpolation			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_RDRB2 field descriptions

Field	Description
7–4 -	This field is reserved. Reserved
AUDIframeinterpolation	Audio frame interpolation

34.5.133 Frame Composer Round Robin ACR Packet Insertion Register 3 (HDMI_FC_RDRB3)

Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the AUDI packet insertion on data island when FC is on RDRB mode this packet.

- Address Offset: 0x10BB
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 10BBh offset = 12_10BBh

Bit	7	6	5	4	3	2	1	0
Read Write	AUDIpacketsinframe				AUDIpacketlinespacing			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_RDRB3 field descriptions

Field	Description
7–4 AUDIpacketsinframe	Audio packets per frame
AUDIpacketlinespacing	Audio packets line spacing

34.5.134 Frame Composer Round Robin ACR Packet Insertion Register 4 (HDMI_FC_RDRB4)

Configures the Frame Composer (FC) RDRB frame interpolation for GCP packet insertion on data island when FC is on RDRB mode for this packet.

- Address Offset: 0x10BC

- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 10BCh offset = 12_10BCh

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				GCPframeinterpolation			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_RDRB4 field descriptions

Field	Description
7–4 -	This field is reserved. Reserved
GCPframeinterpolation	GCP packets line spacing

34.5.135 Frame Composer Round Robin ACR Packet Insertion Register 5 (HDMI_FC_RDRB5)

Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the GCP packet insertion on data island when FC is on RDRB mode this packet.

- Address Offset: 0x10BD
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 10BDh offset = 12_10BDh

Bit	7	6	5	4	3	2	1	0
Read Write	GCPpacketsinframe				GCPpacketlinespacing			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_RDRB5 field descriptions

Field	Description
7–4 GCPpacketsinframe	GCP packets per frame
GCPpacketlinespacing	GCP packets line spacing

34.5.136 Frame Composer Round Robin ACR Packet Insertion Register 6 (HDMI_FC_RDRB6)

Configures the Frame Composer (FC) RDRB frame interpolation for AVI packet insertion on data island when FC is on RDRB mode for this packet.

- Address Offset: 0x10BE
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 10BEh offset = 12_10BEh

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				AVIframeinterpolation			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_RDRB6 field descriptions

Field	Description
7–4 -	This field is reserved. Reserved
AVIframeinterpolation	GCP packets line spacing

34.5.137 Frame Composer Round Robin ACR Packet Insertion Register 7 (HDMI_FC_RDRB7)

Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the AVI packet insertion on data island when FC is on RDRB mode this packet.

- Address Offset: 0x10BF
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 10BFh offset = 12_10BFh

Bit	7	6	5	4	3	2	1	0
Read Write	AVIpacketsinframe				AVIpacketlinespacing			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_RDRB7 field descriptions

Field	Description
7–4 AVIpacketsinframe	AVI packets per frame
AVIpacketlinespacing	AVI packets line spacing

34.5.138 Frame Composer Packet Interrupt Mask Register 0 (HDMI_FC_MASK0)

Mask register for generation of FC_INT0 interrupts.

- Address Offset: 0x10D2
- Size: 8 bits
- Value after Reset: 0x25
- Access: Read/Write

Address: 12_0000h base + 10D2h offset = 12_10D2h

Bit	7	6	5	4	3	2	1	0
Read Write	AUDI	ACP	HBR	Reserved		AUDS	ACR	NULL
Reset	0	0	1	0	0	1	0	1

HDMI_FC_MASK0 field descriptions

Field	Description
7 AUDI	Mask bit for FC_INT0.AUDI interrupt bit Value after Reset: 0b
6 ACP	Mask bit for FC_INT0.ACP interrupt bit Value after Reset: 0b
5 HBR	Mask bit for FC_INT0.HBR interrupt bit Value after Reset: 0b
4–3 -	This field is reserved. Reserved
2 AUDS	Mask bit for FC_INT0.AUDS interrupt bit Value after Reset: 0b
1 ACR	Mask bit for FC_INT0.ACR interrupt bit Value after Reset: 0b
0 NULL	Mask bit for FC_INT0.NULL interrupt bit Value after Reset: 0b

34.5.139 Frame Composer Packet Interrupt Mask Register 1 (HDMI_FC_MASK1)

Mask register for generation of FC_INT1 interrupts.

- Address Offset: 0x10D6
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 10D6h offset = 12_10D6h

Bit	7	6	5	4	3	2	1	0
Read Write	GMD	ISCR1	ISCR2	VSD	SPD	Reserved	AVI	GCP
Reset	0	0	0	0	0	0	0	0

HDMI_FC_MASK1 field descriptions

Field	Description
7 GMD	Mask bit for FC_INT1.GMD interrupt bit
6 ISCR1	Mask bit for FC_INT1.ISRC1 interrupt bit
5 ISCR2	Mask bit for FC_INT1.ISRC2 interrupt bit
4 VSD	Mask bit for FC_INT1.VSD interrupt bit
3 SPD	Mask bit for FC_INT1.SPD interrupt bit
2 -	This field is reserved. Reserved
1 AVI	Mask bit for FC_INT1.AVI interrupt bit
0 GCP	Mask bit for FC_INT1.GCP interrupt bit

34.5.140 Frame Composer Pixel Repetition Configuration Register (HDMI_FC_PRCONF)

Defines the Pixel Repetition ratio factor of the input and output video signal.

- Address Offset: 0x10E0
- Size: 8 bits
- Value after Reset: 0x10
- Access: Read/Write

Address: 12_0000h base + 10E0h offset = 12_10E0h

Bit	7	6	5	4	3	2	1	0
Read Write	incoming_pr_factor[3:0]				output_pr_factor[3:0]			
Reset	0	0	0	1	0	0	0	0

HDMI_FC_PRCONF field descriptions

Field	Description
7–4 incoming_pr_factor[3:0]	Configures the input video pixel repetition. A plus 1 factor should be added in this register configuration. For CEA modes this value should be extracted from the CEA spec for the video mode being inputted. **NOTE:** When working in YCC422 video the actual repetition of the stream will be Incoming_pr_factor * (desired_pr_factor + 1). This calculation is done internally in the H13TCTRL and no HW overflow protection is available. Care must be taken to avoid this result passes the maximum number of 10 pixels repeated since no HDMI support is available for this in the spec and the H13TPHY does not support this higher repetition values. other: Reserved. Not used. 0000 No action. Shall not be used. 0001 No pixel repetition (pixel sent only once). 0010 Pixel sent twice (pixel repeated once). 0011 Pixel sent 3 times. 0100 Pixel sent 4 times. 0101 Pixel sent 5 times. 0110 Pixel sent 6 times. 0111 Pixel sent 7 times. 1000 Pixel sent 8 times. 1001 Pixel sent 9 times. 1010 Pixel sent 10 times.
output_pr_factor[3:0]	Configures the video pixel repetition ratio to be sent on the AVI infoFrame. This value must be valid according to HDMI spec. The output_pr_factor = incoming_pr_factor(without the + 1 factor) * desired_pr_factor. other: Reserved. Not used. 0000 No action. Shall not be used. 0001 Pixel sent twice (pixel repeated once). 0010 Pixel sent 3 times. 0011 Pixel sent 4 times. 0100 Pixel sent 5 times. 0101 Pixel sent 6 times. 0110 Pixel sent 7 times. 0111 Pixel sent 8 times. 1000 Pixel sent 9 times. 1001 Pixel sent 10 times.

34.5.141 Frame Composer GMD Packet Status Register (HDMI_FC_GMD_STAT)

Gamut metadata packet status bit information for no_current_gmd, next_gmd_field, gmd_packet_sequence and current_gamut_seq_num. For more information, refer to the HDMI 1.4a specification.

- Address Offset: 0x1100
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 1100h offset = 12_1100h

Bit	7	6	5	4
Read	igmdno_crnt_gbd	igmddnext_field	igmdpacket_seq[1:0]	
Write				
Reset	0	0	0	0

Bit	3	2	1	0
Read	igmdcurrent_gamut_seq_num[3:0]			
Write				
Reset	0	0	0	0

HDMI_FC_GMD_STAT field descriptions

Field	Description
7 igmdno_crnt_gbd	Gamut scheduling: No current gamut data
6 igmddnext_field	Gamut scheduling: Gamut Next field
5–4 igmdpacket_seq[1:0]	Gamut scheduling: Gamut packet sequence
igmdcurrent_gamut_seq_num[3:0]	Gamut scheduling: Current Gamut packet sequence number

34.5.142 Frame Composer GMD Packet Enable Register (HDMI_FC_GMD_EN)

This register enables Gamut metadata (GMD) packet transmission. Packets are inserted in the incoming frame, starting in the line where active Vsync indication starts. After enable of GMD packets the outgoing packet is sent with no_current_gmd active indication until update GMD request is performed in the controller.

- Address Offset: 0x1101
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1101h offset = 12_1101h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved							gmdenabletx
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_EN field descriptions

Field	Description
7–1 -	This field is reserved. Reserved
0 gmdenabletx	Gamut Metadata packet transmission enable (1b).

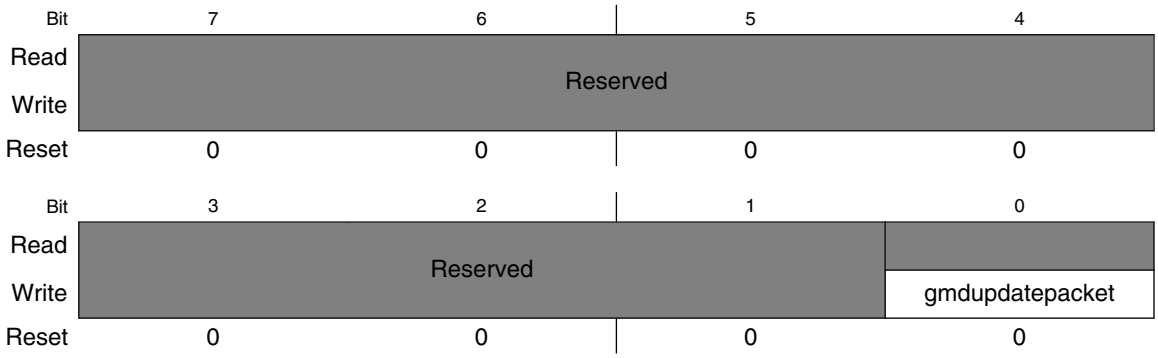
34.5.143 Frame Composer GMD Packet Update Register (HDMI_FC_GMD_UP)

This register performs an GMD packet content update according to the configured packet body (FC_GMD_PB0 to FC_GMD_PB27) and packet header (FC_GMD_HB). This active high auto clear register reflects the body and header configurations on the GMD packets sent arbitrating the current_gamut_seq_num, gmd_packet_sequence and next_gmd_field bits on packet to correctly indicate to source the Gamut change to be performed. After enable GMD packets the first update request is also responsible for deactivating the no_current_gmd indication bit. Attention packet update request must only be done after correct configuration of GMD packet body and header registers. Correct affected_gamut_seq_num and gmd_profile configuration is user responsibility and must convey with HDMI 1.4a standard gamut rules.

- Address Offset: 0x1102

- Size: 8 bits
- Value after Reset: 0x00
- Access: Write

Address: 12_0000h base + 1102h offset = 12_1102h



HDMI_FC_GMD_UP field descriptions

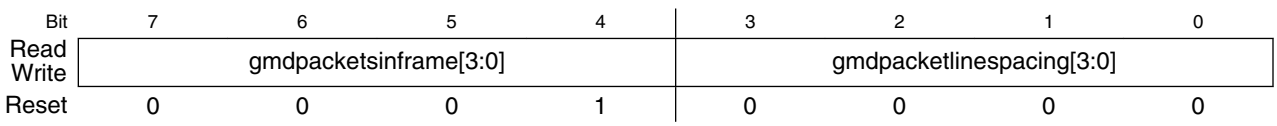
Field	Description
7–1 -	This field is reserved. Reserved
0 gmdupdatepacket	Gamut Metadata packet update.

34.5.144 Frame Composer GMD Packet Schedule Configuration Register (HDMI_FC_GMD_CONF)

This register configures the number of GMD packets to be inserted per frame (starting always in the line where the active Vsync appears) and the line spacing between the transmitted GMD packets. Note that for profile P0 (refer to HDMI 1.4a spec) this register should only indicate one GMD packet to be inserted per video field.

- Address Offset: 0x1103
- Size: 8 bits
- Value after Reset: 0x10
- Access: Read/Write

Address: 12_0000h base + 1103h offset = 12_1103h



HDMI_FC_GMD_CONF field descriptions

Field	Description
7–4 gmdpacketsinframe[3:0]	Number of GMD packets per frame or video field (profile P0)
gmdpacketlinespacing[3:0]	Number of line spacing between the transmitted GMD packets

34.5.145 Frame Composer GMD Packet Profile and Gamut Sequence Configuration Register (HDMI_FC_GMD_HB)

This register configures the GMD packet header affected_gamut_seq_num and gmd_profile bits. For more information, refer to the HDMI 1.4a specification.

- Address Offset: 0x1104
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1104h offset = 12_1104h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved	gmdgbd_profile			gmdaffected_gamut_seq_num			
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_HB field descriptions

Field	Description
7 -	This field is reserved. Reserved
6–4 gmdgbd_profile	GMD profile bits
gmdaffected_ gamut_seq_num	Affected gamut sequence number

34.5.146 Frame Composer GMD Packet Body Register 0 (HDMI_FC_GMD_PB0)

Configures the following contents of the GMD packet:

- GMD packet body byte0
- Address Offset: 0x1105
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1105h offset = 12_1105h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB0							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB0 field descriptions

Field	Description
FC_GMD_PB0	Gamut Metadata packet byte0

34.5.147 Frame Composer GMD Packet Body Register 1 (HDMI_FC_GMD_PB1)

- GMD packet body byte1
- Address Offset: 0x1106
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1106h offset = 12_1106h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB1							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB1 field descriptions

Field	Description
FC_GMD_PB1	Gamut Metadata packet byte1

34.5.148 Frame Composer GMD Packet Body Register 2 (HDMI_FC_GMD_PB2)

- GMD packet body byte2
- Address Offset: 0x1107
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1107h offset = 12_1107h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB2							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB2 field descriptions

Field	Description
FC_GMD_PB2	Gamut Metadata packet byte2

34.5.149 Frame Composer GMD Packet Body Register 3 (HDMI_FC_GMD_PB3)

- GMD packet body byte3
- Address Offset: 0x1108
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1108h offset = 12_1108h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB3							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB3 field descriptions

Field	Description
FC_GMD_PB3	Gamut Metadata packet byte3

34.5.150 Frame Composer GMD Packet Body Register 4 (HDMI_FC_GMD_PB4)

- GMD packet body byte4
- Address Offset: 0x1109
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1109h offset = 12_1109h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB4							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB4 field descriptions

Field	Description
FC_GMD_PB4	Gamut Metadata packet byte4

34.5.151 Frame Composer GMD Packet Body Register 5 (HDMI_FC_GMD_PB5)

- GMD packet body byte5
- Address Offset: 0x110a
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 110Ah offset = 12_110Ah

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB5							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB5 field descriptions

Field	Description
FC_GMD_PB5	Gamut Metadata packet byte5

34.5.152 Frame Composer GMD Packet Body Register 6 (HDMI_FC_GMD_PB6)

- GMD packet body byte6
- Address Offset: 0x110b
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 110Bh offset = 12_110Bh

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB6							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB6 field descriptions

Field	Description
FC_GMD_PB6	Gamut Metadata packet byte6

34.5.153 Frame Composer GMD Packet Body Register 7 (HDMI_FC_GMD_PB7)

- GMD packet body byte7
- Address Offset: 0x110c
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 110Ch offset = 12_110Ch

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB2							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB7 field descriptions

Field	Description
FC_GMD_PB2	Gamut Metadata packet byte7

34.5.154 Frame Composer GMD Packet Body Register 8 (HDMI_FC_GMD_PB8)

- GMD packet body byte8
- Address Offset: 0x110d
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 110Dh offset = 12_110Dh

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB8							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB8 field descriptions

Field	Description
FC_GMD_PB8	Gamut Metadata packet byte8

34.5.155 Frame Composer GMD Packet Body Register 9 (HDMI_FC_GMD_PB9)

- GMD packet body byte9
- Address Offset: 0x110e
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 110Eh offset = 12_110Eh

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB9							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB9 field descriptions

Field	Description
FC_GMD_PB9	Gamut Metadata packet byte9

34.5.156 Frame Composer GMD Packet Body Register 10 (HDMI_FC_GMD_PB10)

- GMD packet body byte10
- Address Offset: 0x110f
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 110Fh offset = 12_110Fh

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB10							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB10 field descriptions

Field	Description
FC_GMD_PB10	Gamut Metadata packet byte10

34.5.157 Frame Composer GMD Packet Body Register 11 (HDMI_FC_GMD_PB11)

- GMD packet body byte11
- Address Offset: 0x1110
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1110h offset = 12_1110h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB11							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB11 field descriptions

Field	Description
FC_GMD_PB11	Gamut Metadata packet byte11

34.5.158 Frame Composer GMD Packet Body Register 12 (HDMI_FC_GMD_PB12)

- GMD packet body byte12
- Address Offset: 0x1111
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1111h offset = 12_1111h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB12							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB12 field descriptions

Field	Description
FC_GMD_PB12	Gamut Metadata packet byte12

34.5.159 Frame Composer GMD Packet Body Register 13 (HDMI_FC_GMD_PB13)

- GMD packet body byte13
- Address Offset: 0x1112
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1112h offset = 12_1112h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB13							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB13 field descriptions

Field	Description
FC_GMD_PB13	Gamut Metadata packet byte13

34.5.160 Frame Composer GMD Packet Body Register 14 (HDMI_FC_GMD_PB14)

- GMD packet body byte14
- Address Offset: 0x1113
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1113h offset = 12_1113h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB14							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB14 field descriptions

Field	Description
FC_GMD_PB14	Gamut Metadata packet byte14

34.5.161 Frame Composer GMD Packet Body Register 15 (HDMI_FC_GMD_PB15)

- GMD packet body byte15
- Address Offset: 0x1114
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1114h offset = 12_1114h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB15							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB15 field descriptions

Field	Description
FC_GMD_PB15	Gamut Metadata packet byte15

34.5.162 Frame Composer GMD Packet Body Register 16 (HDMI_FC_GMD_PB16)

- GMD packet body byte16
- Address Offset: 0x1115
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1115h offset = 12_1115h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB16							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB16 field descriptions

Field	Description
FC_GMD_PB16	Gamut Metadata packet byte16

34.5.163 Frame Composer GMD Packet Body Register 17 (HDMI_FC_GMD_PB17)

- GMD packet body byte17
- Address Offset: 0x1116
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1116h offset = 12_1116h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB17							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB17 field descriptions

Field	Description
FC_GMD_PB17	Gamut Metadata packet byte17

34.5.164 Frame Composer GMD Packet Body Register 18 (HDMI_FC_GMD_PB18)

- GMD packet body byte18
- Address Offset: 0x1117
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1117h offset = 12_1117h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB18							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB18 field descriptions

Field	Description
FC_GMD_PB18	Gamut Metadata packet byte18

34.5.165 Frame Composer GMD Packet Body Register 19 (HDMI_FC_GMD_PB19)

- GMD packet body byte19
- Address Offset: 0x1118
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1118h offset = 12_1118h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB18							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB19 field descriptions

Field	Description
FC_GMD_PB18	Gamut Metadata packet byte18

34.5.166 Frame Composer GMD Packet Body Register 20 (HDMI_FC_GMD_PB20)

- GMD packet body byte20
- Address Offset: 0x1119
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1119h offset = 12_1119h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB20							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB20 field descriptions

Field	Description
FC_GMD_PB20	Gamut Metadata packet byte20

34.5.167 Frame Composer GMD Packet Body Register 21 (HDMI_FC_GMD_PB21)

- GMD packet body byte21
- Address Offset: 0x111a
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 111Ah offset = 12_111Ah

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB21							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB21 field descriptions

Field	Description
FC_GMD_PB21	Gamut Metadata packet byte21

34.5.168 Frame Composer GMD Packet Body Register 22 (HDMI_FC_GMD_PB22)

- GMD packet body byte22
- Address Offset: 0x111b
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 111Bh offset = 12_111Bh

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB22							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB22 field descriptions

Field	Description
FC_GMD_PB22	Gamut Metadata packet byte22

34.5.169 Frame Composer GMD Packet Body Register 23 (HDMI_FC_GMD_PB23)

- GMD packet body byte23
- Address Offset: 0x111c
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 111Ch offset = 12_111Ch

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB23							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB23 field descriptions

Field	Description
FC_GMD_PB23	Gamut Metadata packet byte23

34.5.170 Frame Composer GMD Packet Body Register 24 (HDMI_FC_GMD_PB24)

- GMD packet body byte24
- Address Offset: 0x111d
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 111Dh offset = 12_111Dh

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB24							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB24 field descriptions

Field	Description
FC_GMD_PB24	Gamut Metadata packet byte24

34.5.171 Frame Composer GMD Packet Body Register 25 (HDMI_FC_GMD_PB25)

- GMD packet body byte25
- Address Offset: 0x111e
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 111Eh offset = 12_111Eh

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB25							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB25 field descriptions

Field	Description
FC_GMD_PB25	Gamut Metadata packet byte25

34.5.172 Frame Composer GMD Packet Body Register 26 (HDMI_FC_GMD_PB26)

- GMD packet body byte26
- Address Offset: 0x111f
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 111Fh offset = 12_111Fh

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB26							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB26 field descriptions

Field	Description
FC_GMD_PB26	Gamut Metadata packet byte26

34.5.173 Frame Composer GMD Packet Body Register 27 (HDMI_FC_GMD_PB27)

- GMD packet body byte27
- Address Offset: 0x1120
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

For more information, refer to the HDMI 1.4a specification.

Address: 12_0000h base + 1120h offset = 12_1120h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_GMD_PB27							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_GMD_PB27 field descriptions

Field	Description
FC_GMD_PB27	Gamut Metadata packet byte27

34.5.174 Frame Composer Video/Audio Force Enable Register (HDMI_FC_DBGFORCE)

This register allows to force the controller to output audio and video data the values configured in the FC_DBGAUD and FC_DBGTMDS registers.

- Address Offset: 0x1200
- Size: 8 bits

- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1200h offset = 12_1200h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved			forceaudio	Reserved			forcevideo
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGFORCE field descriptions

Field	Description
7–5 -	This field is reserved. Reserved
4 forceaudio	Force fixed audio output with FC_DBGAUDxCHx registers contain.
3–1 -	This field is reserved. Reserved
0 forcevideo	Force fixed video output with FC_DBGTMDSx registers contain.

34.5.175 Frame Composer Audio Channel 0 Register 0 (HDMI_FC_DBGAUD0CH0)

Configures the audio fixed data to be used in channel 0 when in fixed audio selection.

- Address Offset: 0x1201
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1201h offset = 12_1201h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD0CH0							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD0CH0 field descriptions

Field	Description
FC_ DBGAUD0CH0	the audio fixed data byte0 to be used in channel 0 when in fixed audio selection

34.5.176 Frame Composer Audio Channel 0 Register 1 (HDMI_FC_DBGAUD1CH0)

- Address Offset: 0x1202
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1202h offset = 12_1202h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD1CH0							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD1CH0 field descriptions

Field	Description
FC_ DBGAUD1CH0	the audio fixed data byte1 to be used in channel 0 when in fixed audio selection

34.5.177 Frame Composer Audio Channel 0 Register 2 (HDMI_FC_DBGAUD2CH0)

- Address Offset: 0x1203
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1203h offset = 12_1203h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD2CH0							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD2CH0 field descriptions

Field	Description
FC_ DBGAUD2CH0	the audio fixed data byte2 to be used in channel 0 when in fixed audio selection

34.5.178 Frame Composer Audio Channel 1 Register 0 (HDMI_FC_DBGAUD0CH1)

Configures the audio fixed data to be used in channel 0 when in fixed audio selection.

- Address Offset: 0x1204
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1204h offset = 12_1204h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD0CH1							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD0CH1 field descriptions

Field	Description
FC_ DBGAUD0CH1	the audio fixed data byte2 to be used in channel 0 when in fixed audio selection

34.5.179 Frame Composer Audio Channel 1 Register 1 (HDMI_FC_DBGAUD1CH1)

- Address Offset: 0x1205
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1205h offset = 12_1205h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD1CH1							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD1CH1 field descriptions

Field	Description
FC_ DBGAUD1CH1	the audio fixed data byte1 to be used in channel 1 when in fixed audio selection

34.5.180 Frame Composer Audio Channel 1 Register 2 (HDMI_FC_DBGAUD2CH1)

- Address Offset: 0x1206
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1206h offset = 12_1206h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD2CH1							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD2CH1 field descriptions

Field	Description
FC_ DBGAUD2CH1	the audio fixed data byte2 to be used in channel 1 when in fixed audio selection

34.5.181 Frame Composer Debug Audio Channel 2 Register 0 (HDMI_FC_DBGAUD0CH2)

Configures the audio fixed data to be used in channel 0 when in fixed audio selection.

- Address Offset: 0x1207
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1207h offset = 12_1207h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD0CH2							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD0CH2 field descriptions

Field	Description
FC_ DBGAUD0CH2	the audio fixed data byte0 to be used in channel 2 when in fixed audio selection

34.5.182 Frame Composer Debug Audio Channel 2 Register 1 (HDMI_FC_DBGAUD1CH2)

- Address Offset: 0x1208
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1208h offset = 12_1208h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD1CH2							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD1CH2 field descriptions

Field	Description
FC_ DBGAUD1CH2	the audio fixed data byte1 to be used in channel 2 when in fixed audio selection

34.5.183 Frame Composer Audio Channel 2 Register 2 (HDMI_FC_DBGAUD2CH2)

- Address Offset: 0x1209
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1209h offset = 12_1209h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD2CH2							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD2CH2 field descriptions

Field	Description
FC_ DBGAUD2CH2	the audio fixed data byte2 to be used in channel 2 when in fixed audio selection

34.5.184 Frame Composer Audio Channel 3 Register 0 (HDMI_FC_DBGAUD0CH3)

Configures the audio fixed data to be used in channel 0 when in fixed audio selection.

- Address Offset: 0x120A
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 120Ah offset = 12_120Ah

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD0CH3							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD0CH3 field descriptions

Field	Description
FC_ DBGAUD0CH3	the audio fixed data byte0 to be used in channel 3 when in fixed audio selection

34.5.185 Frame Composer Audio Channel 3 Register 1 (HDMI_FC_DBGAUD1CH3)

- Address Offset: 0x120B
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 120Bh offset = 12_120Bh

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD1CH3							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD1CH3 field descriptions

Field	Description
FC_ DBGAUD1CH3	the audio fixed data byte1 to be used in channel 3 when in fixed audio selection

34.5.186 Frame Composer Audio Channel 3 Register 2 (HDMI_FC_DBGAUD2CH3)

- Address Offset: 0x120C
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 120Ch offset = 12_120Ch

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD2CH3							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD2CH3 field descriptions

Field	Description
FC_ DBGAUD2CH3	the audio fixed data byte2 to be used in channel 3 when in fixed audio selection

34.5.187 Frame Composer Audio Channel 4 Register 0 (HDMI_FC_DBGAUD0CH4)

Configures the audio fixed data to be used in channel 0 when in fixed audio selection.

- Address Offset: 0x120D
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 120Dh offset = 12_120Dh

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD0CH4							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD0CH4 field descriptions

Field	Description
FC_ DBGAUD0CH4	the audio fixed data byte0 to be used in channel 4 when in fixed audio selection

34.5.188 Frame Composer Audio Channel 4 Register 1 (HDMI_FC_DBGAUD1CH4)

- Address Offset: 0x120E
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 120Eh offset = 12_120Eh

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD1CH4							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD1CH4 field descriptions

Field	Description
FC_ DBGAUD1CH4	the audio fixed data byte1 to be used in channel 4 when in fixed audio selection

34.5.189 Frame Composer Audio Channel 4 Register 2 (HDMI_FC_DBGAUD2CH4)

- Address Offset: 0x120F
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 120Fh offset = 12_120Fh

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD2CH4							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD2CH4 field descriptions

Field	Description
FC_ DBGAUD2CH4	the audio fixed data byte2 to be used in channel 4 when in fixed audio selection

34.5.190 Frame Composer Audio Channel 5 Register 0 (HDMI_FC_DBGAUD0CH5)

Configures the audio fixed data to be used in channel 0 when in fixed audio selection.

- Address Offset: 0x1210
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1210h offset = 12_1210h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD0CH5							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD0CH5 field descriptions

Field	Description
FC_ DBGAUD0CH5	the audio fixed data byte0 to be used in channel 5 when in fixed audio selection

34.5.191 Frame Composer Audio Channel 5 Register 1 (HDMI_FC_DBGAUD1CH5)

- Address Offset: 0x1211
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1211h offset = 12_1211h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD1CH5							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD1CH5 field descriptions

Field	Description
FC_ DBGAUD1CH5	the audio fixed data byte1 to be used in channel 5 when in fixed audio selection

34.5.192 Frame Composer Audio Channel 5 Register 2 (HDMI_FC_DBGAUD2CH5)

- Address Offset: 0x1212
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1212h offset = 12_1212h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD2CH5							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD2CH5 field descriptions

Field	Description
FC_ DBGAUD2CH5	the audio fixed data byte2 to be used in channel 5 when in fixed audio selection

34.5.193 Frame Composer Audio Channel 6 Register 0 (HDMI_FC_DBGAUD0CH6)

Configures the audio fixed data to be used in channel 0 when in fixed audio selection.

- Address Offset: 0x1213
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1213h offset = 12_1213h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD0CH6							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD0CH6 field descriptions

Field	Description
FC_ DBGAUD0CH6	The audio fixed data byte0 to be used in channel 6 when in fixed audio selection

34.5.194 Frame Composer Audio Channel 6 Register 1 (HDMI_FC_DBGAUD1CH6)

- Address Offset: 0x1214
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1214h offset = 12_1214h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD1CH6							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD1CH6 field descriptions

Field	Description
FC_ DBGAUD1CH6	the audio fixed data byte1 to be used in channel 6 when in fixed audio selection

34.5.195 Frame Composer Audio Channel 6 Register 2 (HDMI_FC_DBGAUD2CH6)

- Address Offset: 0x1215
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1215h offset = 12_1215h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD2CH6							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD2CH6 field descriptions

Field	Description
FC_ DBGAUD2CH6	the audio fixed data byte2 to be used in channel 6 when in fixed audio selection

34.5.196 Frame Composer Audio Channel 7 Register 1 (HDMI_FC_DBGAUD0CH7)

Configures the audio fixed data to be used in channel 7 when in fixed audio selection.

- Address Offset: 0x1216
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1216h offset = 12_1216h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD0CH7							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD0CH7 field descriptions

Field	Description
FC_ DBGAUD0CH7	the audio fixed data byte0 to be used in channel 7 when in fixed audio selection

34.5.197 Frame Composer Audio Channel 7 Register 0 (HDMI_FC_DBGAUD1CH7)

- Address Offset: 0x1217
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1217h offset = 12_1217h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD1CH7							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD1CH7 field descriptions

Field	Description
FC_ DBGAUD1CH7	the audio fixed data byte1 to be used in channel 0 when in fixed audio selection

34.5.198 Frame Composer Audio Channel 7 Register 2 (HDMI_FC_DBGAUD2CH7)

- Address Offset: 0x1218
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1218h offset = 12_1218h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGAUD2CH7							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGAUD2CH7 field descriptions

Field	Description
FC_ DBGAUD2CH7	the audio fixed data byte2 to be used in channel 0 when in fixed audio selection

34.5.199 Frame Composer TMDS Channel 0 Register (HDMI_FC_DBGTMDS0)

Configures the video fixed data to be used in tmds channel 0 when in fixed video selection. This equals to set B pixel component value in RGB video or Cb pixel component value in YCbCr.

- Address Offset: 0x1219
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 1219h offset = 12_1219h

Bit	7	6	5	4	3	2	1	0
Read Write	FC_DBGTMDS0							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGTMDS0 field descriptions

Field	Description
FC_DBGTMDS0	set B pixel component value in RGB video or Cb pixel component value in YCbCr

34.5.200 Frame Composer TMDS Channel 1 Register (HDMI_FC_DBGTMDS1)

Configures the video fixed data to be used in tmds channel 1 when in fixed video selection. This equals to set G pixel component value in RGB video or Y pixel component value in YCbCr.

- Address Offset: 0x121A
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 121Ah offset = 12_121Ah

Bit	7	6	5	4	3	2	1	0
Read / Write	FC_DBGTMDS1							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGTMDS1 field descriptions

Field	Description
FC_DBGTMDS1	set G pixel component value in RGB video or Y pixel component value in YCbCr

34.5.201 Frame Composer TMDS Channel 2 Register (HDMI_FC_DBGTMDS2)

Configures the video fixed data to be used in tmds channel 2 when in fixed video selection. This equals to set R pixel component value in RGB video or Cr pixel component value in YCbCr.

- Address Offset: 0x121B
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 121Bh offset = 12_121Bh

Bit	7	6	5	4	3	2	1	0
Read / Write	FC_DBGTMDS2							
Reset	0	0	0	0	0	0	0	0

HDMI_FC_DBGTMDS2 field descriptions

Field	Description
FC_DBGTMDS2	set R pixel component value in RGB video or Cr pixel component value in YCbCr

34.5.202 PHY Configuration Register (HDMI_PHY_CONF0)

This register holds the power down, data enable polarity and interface control of the HDMI Source PHY control. For more information, refer to the DesignWare Cores HDMI TX PHY Databook.

- Address Offset: 0x3000
- Size: 8 bits
- Value after Reset: 0x06
- Access: Read/Write

Address: 12_0000h base + 3000h offset = 12_3000h

Bit	7	6	5	4
Read Write	PDZ	ENTMDS	sparectrl	gen2_pddq
Reset	0	0	0	0

Bit	3	2	1	0
Read Write	gen2_txpwron	gen2_enhpdrxsense	seldataenpol	seldipif
Reset	0	1	1	0

HDMI_PHY_CONF0 field descriptions

Field	Description
7 PDZ	Power-down enable (active low 0b). Value after Reset: 0b
6 ENTMDS	Enable TMDS drivers, bias, and TMDS digital logic. Value after Reset: 0b
5 sparectrl	Reserved. Spare pin control. Value after Reset: 0b
4 gen2_pddq	PHY_Gen2 PDDQ signal Value after Reset: 0b
3 gen2_txpwron	PHY_Gen2 TXPWRON signal Value after Reset: 0b
2 gen2_ enhpdrxsense	PHY_Gen2 ENHPDRXSENSE signal Value after Reset: 1b
1 seldataenpol	Select data enable polarity. Value after Reset: 1b

Table continues on the next page...

HDMI_PHY_CONF0 field descriptions (continued)

Field	Description
0 seldipif	Select interface control. Value after Reset: 0b

34.5.203 PHY Test Interface Register 0 (HDMI_PHY_TST0)

PHY TX mapped text interface (control). For more information, refer to the DesignWare Cores HDMI TX PHY Databook.

- Address Offset: 0x3001
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3001h offset = 12_3001h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved		testclr	testen	Reserved			testclk
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_TST0 field descriptions

Field	Description
7–6 -	This field is reserved. Reserved
5 testclr	Enable TMDS drivers, bias and tmds digital logic. Value after Reset: 0b
4 testen	Reserved. Spare control pins. Value after Reset: 0b
3–1 -	This field is reserved. Reserved
0 testclk	Test clock signal. Value after Reset: 0b

34.5.204 PHY Test Interface Register 1 (HDMI_PHY_TST1)

PHY TX mapped text interface (data in). For more information, refer to the DesignWare Cores HDMI TX PHY Databook.

- Address Offset: 0x3002
- Size: 8 bits

- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3002h offset = 12_3002h

Bit	7	6	5	4	3	2	1	0
Read Write	testdin[7:0]							
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_TST1 field descriptions

Field	Description
testdin[7:0]	Test data input.

34.5.205 PHY Test Interface Register 2 (HDMI_PHY_TST2)

PHY TX mapped text interface (data out). For more information, refer to the DesignWare Cores HDMI TX PHY Databook.

- Address Offset: 0x3003
- Size: 8 bits
- Value after Reset: N/A
- Access: Read

Address: 12_0000h base + 3003h offset = 12_3003h

Bit	7	6	5	4	3	2	1	0
Read	testdout[7:0]							
Write								
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_TST2 field descriptions

Field	Description
testdout[7:0]	Test data output.

34.5.206 PHY RXSENSE, PLL lock, and HPD Status Register (HDMI_PHY_STAT0)

This register contains the following active high packet sent status indications. For more information, see Overview

- Address Offset: 0x3004
- Size: 8 bits

- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 3004h offset = 12_3004h

Bit	7	6	5	4
Read	RX_SENSE3	RX_SENSE2	RX_SENSE1	RX_SENSE0
Write				
Reset	0	0	0	0

Bit	3	2	1	0
Read	Reserved		HPD	TX_PHY_LOCK
Write				
Reset	0	0	0	0

HDMI_PHY_STAT0 field descriptions

Field	Description
7 RX_SENSE3	Status bit. TX PHY RX_SENSE indication for TMDS CLK driver. User may need to mask or change polarity of this interrupt after it has became active.
6 RX_SENSE2	Status bit. TX PHY RX_SENSE indication for TMDS channel 2 driver. User may need to mask or change polarity of this interrupt after it has became active.
5 RX_SENSE1	Status bit. TX PHY RX_SENSE indication for TMDS channel 1 driver. User may need to mask or change polarity of this interrupt after it has became active.
4 RX_SENSE0	Status bit. TX PHY RX_SENSE indication for TMDS channel 0 driver. User may need to mask or change polarity of this interrupt after it has became active.
3–2 -	This field is reserved. Reserved
1 HPD	Status bit. HDMI Hot Plug Detect indication. User may need to mask or change polarity of this interrupt after it has became active.
0 TX_PHY_LOCK	Status bit. TX PHY PLL lock indication. Please refer to PHY datasheet for more information. User may need to mask or change polarity of this interrupt after it has became active.

34.5.207 PHY RXSENSE, PLL lock, and HPD Interrupt Register (HDMI_PHY_INT0)

This register contains the interrupt indication of the PHY_STAT0 status interrupts. Interrupt generation is accomplished in the following way:

interrupt = (mask == 1'b0) && (polarity == status);

All this interrupts are forwarded to the Interrupt Handler sticky bit registers and after ORed to a single main interrupt line to micro controller. Assertion of this interrupt implies that data related with the corresponding packet has been sent through the HDMI interface.

- Address Offset: 0x3005

- Size: 8 bits
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 3005h offset = 12_3005h

Bit	7	6	5	4
Read	RX_SENSE3	RX_SENSE2	RX_SENSE1	RX_SENSE0
Write				
Reset	0	0	0	0

Bit	3	2	1	0
Read	Reserved		HPD	TX_PHY_LOCK
Write				
Reset	0	0	0	0

HDMI_PHY_INT0 field descriptions

Field	Description
7 RX_SENSE3	Interrupt indication bit TX PHY RX_SENSE indication interrupt for TMDS CLK driver.
6 RX_SENSE2	Interrupt indication bit TX PHY RX_SENSE indication interrupt for TMDS channel 2 driver.
5 RX_SENSE1	Interrupt indication bit TX PHY RX_SENSE indication interrupt for TMDS channel 1 driver.
4 RX_SENSE0	Interrupt indication bit TX PHY RX_SENSE indication interrupt for TMDS channel 0 driver.
3–2 -	This field is reserved. Reserved
1 HPD	Interrupt indication bit HDMI Hot Plug Detect indication interrupt.
0 TX_PHY_LOCK	Interrupt indication bit TX PHY PLL lock indication interrupt.

34.5.208 PHY RXSENSE, PLL lock, and HPD Mask Register (HDMI_PHY_MASK0)

Mask register for generation of PHY_INT0 interrupts.

- Address Offset: 0x3006
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3006h offset = 12_3006h

Bit	7	6	5	4	3	2	1	0
Read Write	RX_ SENSE3	RX_ SENSE2	RX_ SENSE1	RX_ SENSE0	Reserved		HPD	TX_PHY_ LOCK
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_MASK0 field descriptions

Field	Description
7 RX_SENSE3	Mask bit for PHY_INT0.RX_SENSE3 interrupt bit
6 RX_SENSE2	Mask bit for PHY_INT0.RX_SENSE2 interrupt bit
5 RX_SENSE1	Mask bit for PHY_INT0.RX_SENSE1 interrupt bit
4 RX_SENSE0	Mask bit for PHY_INT0.RX_SENSE0 interrupt bit
3–2 -	This field is reserved. Reserved
1 HPD	Mask bit for PHY_INT0.HPD interrupt bit
0 TX_PHY_LOCK	Mask bit for PHY_INT0.TX_PHY_LOCK interrupt bit

34.5.209 PHY RXSENSE, PLL lock and HPD Polarity Register (HDMI_PHY_POL0)

Polarity register for generation of PHY_INT0 interrupts.

- Address Offset: 0x3007
- Size: 8 bits
- Value after Reset: 0xF3
- Access: Read/Write

Address: 12_0000h base + 3007h offset = 12_3007h

Bit	7	6	5	4	3	2	1	0
Read Write	RX_ SENSE3	RX_ SENSE2	RX_ SENSE1	RX_ SENSE0	Reserved		HPD	TX_PHY_ LOCK
Reset	1	1	1	1	0	0	1	1

HDMI_PHY_POL0 field descriptions

Field	Description
7 RX_SENSE3	Polarity bit for PHY_INT0.RX_SENSE3 interrupt bit

Table continues on the next page...

HDMI_PHY_POL0 field descriptions (continued)

Field	Description
6 RX_SENSE2	Polarity bit for PHY_INT0.RX_SENSE2 interrupt bit
5 RX_SENSE1	Polarity bit for PHY_INT0.RX_SENSE1 interrupt bit
4 RX_SENSE0	Polarity bit for PHY_INT0.RX_SENSE0 interrupt bit
3–2 -	This field is reserved. Reserved
1 HPD	Polarity bit for PHY_INT0.HPD interrupt bit
0 TX_PHY_LOCK	Polarity bit for PHY_INT0.TX_PHY_LOCK interrupt bit

34.5.210 PHY I2C Slave Address Configuration Register (HDMI_PHY_I2CM_SLAVE_ADDR)

This register writes the slave address of the I2C Master PHY.

- Address Offset: 0x3020
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3020h offset = 12_3020h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved	-						
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_I2CM_SLAVE_ADDR field descriptions

Field	Description
7 -	This field is reserved. Reserved
-	Slave address to be sent during read and write operations. The PHY Gen2 slave address is: 7'h69 The HEAC PHY slave address is: 7'h49

34.5.211 PHY I2C Address Configuration Register (HDMI_PHY_I2CM_ADDRESS_ADDR)

This register writes the address for read and writer operations.

- Address Offset: 0x3021
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3021h offset = 12_3021h

Bit	7	6	5	4	3	2	1	0
Read Write	address							
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_I2CM_ADDRESS_ADDR field descriptions

Field	Description
address	Register address for read and write operations.

34.5.212 PHY I2C Data Write Register 1 (HDMI_PHY_I2CM_DATAO_1_ADDR)

- Address Offset: 0x3022
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3022h offset = 12_3022h

Bit	7	6	5	4	3	2	1	0
Read Write	datao[15:8]							
Reset	0	0	0	0	0	0	0	0

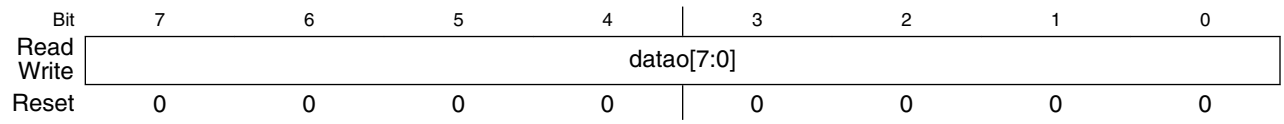
HDMI_PHY_I2CM_DATAO_1_ADDR field descriptions

Field	Description
datao[15:8]	MSB's of data to be written on register pointed by address [7:0].

34.5.213 PHY I2C Data Write Register 0 (HDMI_PHY_I2CM_DATAO_0_ADDR)

- Address Offset: 0x3023
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3023h offset = 12_3023h



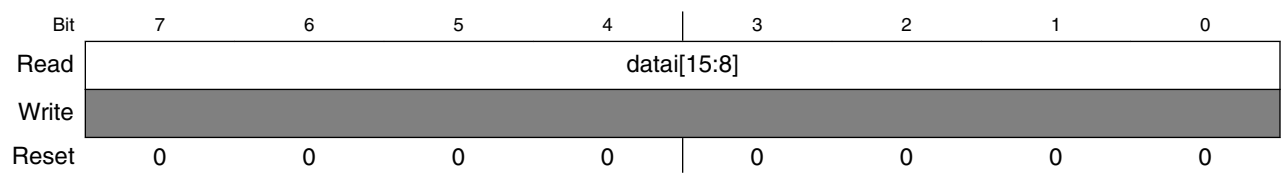
HDMI_PHY_I2CM_DATAO_0_ADDR field descriptions

Field	Description
datao[7:0]	LSB's of data to be written on register pointed by address [7:0].

34.5.214 PHY I2C Data Read Register 1 (HDMI_PHY_I2CM_DATAI_1_ADDR)

- Address Offset: 0x3024
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 3024h offset = 12_3024h



HDMI_PHY_I2CM_DATAI_1_ADDR field descriptions

Field	Description
datai[15:8]	MSB's of data read from the register pointed by address [7:0].

34.5.215 PHY I2C Data Read Register 0 (HDMI_PHY_I2CM_DATAI_0_ADDR)

- Address Offset: 0x3025
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 3025h offset = 12_3025h

Bit	7	6	5	4	3	2	1	0
Read Write	datai[7:0]							
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_I2CM_DATAI_0_ADDR field descriptions

Field	Description
datai[7:0]	LSB's of data read from the register pointed by address [7:0].

34.5.216 PHY I2C Read/Write Operation (HDMI_PHY_I2CM_OPERATION_ADDR)

This register requests read and write operations from the I2C Master PHY. This register can only be written; reading this register always results in 00h. Writing 1'b1 simultaneously to read and write requests is considered a read request.

- Address Offset: 0x3026
- Size: 8 bits
- Value after Reset: 0x00
- Access: Write

Address: 12_0000h base + 3026h offset = 12_3026h

Bit	7	6	5	4	3	2	1	0
Read	Reserved				Reserved			
Write				write				read
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_I2CM_OPERATION_ADDR field descriptions

Field	Description
7–5 -	This field is reserved. Reserved
4 write	Write operation request
3–1 -	This field is reserved. Reserved
0 read	Read operation request.

34.5.217 PHY I2C Done Interrupt Register (HDMI_PHY_I2CM_INT_ADDR)

This register contains and configures I2C master PHY done interrupt.

- Address Offset: 0x3027
- Size: 8 bits
- Value after Reset: 0x08
- Access: Read/Write

Address: 12_0000h base + 3027h offset = 12_3027h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				done_pol	done_mask	done_ interrupt	done_status
Reset	0	0	0	0	1	0	0	0

HDMI_PHY_I2CM_INT_ADDR field descriptions

Field	Description
7–4 -	This field is reserved. Reserved
3 done_pol	Done interrupt polarity configuration Value after Reset: 1b
2 done_mask	Done interrupt mask signal Value after Reset: 0b
1 done_interrupt	Operation done interrupt bit.{done_interrupt =(done_mask==0b)&& (done_status==done_pol)}. Value after Reset: 0b
0 done_status	Operation done status bit.Marks the end of a rd or write operation. Value after Reset: 0b

34.5.218 PHY I2C Done Interrupt Register (HDMI_PHY_I2CM_CTLINT_ADDR)

This register contains and configures the I2C master PHY error interrupts.

- Address Offset: 0x3028
- Size: 8 bits
- Value after Reset: 0x88
- Access: Read/Write

Address: 12_0000h base + 3028h offset = 12_3028h

Bit	7	6	5	4
Read Write	nack_pol	nack_mask	nack_interrupt	nack_status
Reset	1	0	0	0

Bit	3	2	1	0
Read Write	arbitration_pol	arbitration_mask	arbitration_interrupt	arbitration_status
Reset	1	0	0	0

HDMI_PHY_I2CM_CTLINT_ADDR field descriptions

Field	Description
7 nack_pol	Not acknowledge error interrupt polarity configuration. Value after Reset: 1b
6 nack_mask	Not acknowledge error interrupt mask signal Value after Reset: 0b
5 nack_interrupt	Not acknowledge error interrupt bit.{nack_interrupt = nack_mask==0b) && (nack_status==nack_pol)}. Value after Reset: 0b
4 nack_status	Not acknowledge error status bit.Error on I2C not acknowledge. Value after Reset: 0b
3 arbitration_pol	Arbitration error interrupt polarity configuration. Value after Reset: 1b
2 arbitration_mask	Arbitration error interrupt mask signal. Value after Reset: 0b
1 arbitration_ interrupt	Arbitration error interrupt bit.{arbitration_interrupt = (arbitration_mask==0b)&& (arbitration_status==arbitration_pol)}. Value after Reset: 0b
0 arbitration_status	Arbitration error status bit. Error on master I2C protocol arbitration. Value after Reset: 0b

34.5.219 PHY I2C Speed Control Register (HDMI_PHY_I2CM_DIV_ADDR)

This register wets the I2C Master PHY to work in either Fast or Standard mode.

- Address Offset: 0x3029
- Size: 8 bits
- Value after Reset: 0x0B
- Access: Read/Write

Address: 12_0000h base + 3029h offset = 12_3029h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				fast_mode			
Reset	0	0	0	0	1	0	1	1

HDMI_PHY_I2CM_DIV_ADDR field descriptions

Field	Description
7–4 -	This field is reserved. Reserved
fast_mode	Sets the I2C Master to work in Fast Mode or Standard Mode (x implies that it can take any value) Value after Reset: 1011b 1xxxb Fast Mode 0xxxb Standard Mode

34.5.220 PHY I2C Software Reset Register (HDMI_PHY_I2CM_SOFTRSTZ_ADDR)

This register sets the I2C Master PHY software reset.

- Address Offset: 0x302A
- Size: 8 bits
- Value after Reset: 0x01
- Access: Read/Write

The following *CNT registers must be set before any I2C bus transaction can take place to ensure proper I/O timing.

The following are the I2C Master SCL clock settings:

- SS: Standard Speed

- FS: Fast Speed
- HCNT: SCL High Level counter
- LCNT: SCL Low Level counter

Address: 12_0000h base + 302Ah offset = 12_302Ah

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved							i2c_softrst
Reset	0	0	0	0	0	0	0	1

HDMI_PHY_I2CM_SOFTRSTZ_ADDR field descriptions

Field	Description
7–1 -	This field is reserved. Reserved
0 i2c_softrst	I2C Master PHY Software Reset. Active by writing a zero and auto cleared to one in the following cycle. Value after Reset: 1b

34.5.221 PHY I2C Slow Speed SCL High Level Control Register 1 (HDMI_PHY_I2CM_SS_SCL_HCNT_1_ADDR)

- Address Offset: 0x302B
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 302Bh offset = 12_302Bh

Bit	7	6	5	4	3	2	1	0
Read Write	i2cmp_ss_scl_hcnt[15:8]							
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_I2CM_SS_SCL_HCNT_1_ADDR field descriptions

Field	Description
i2cmp_ss_scl_ hcnt[15:8]	Value after Reset: 8'h00

34.5.222 PHY I2C Slow Speed SCL High Level Control Register 0 (HDMI_PHY_I2CM_SS_SCL_HCNT_0_ADDR)

- Address Offset: 0x302C
- Size: 8 bits
- Value after Reset: 0x6C
- Access: Read/Write

Address: 12_0000h base + 302Ch offset = 12_302Ch

Bit	7	6	5	4	3	2	1	0
Read Write	i2cmp_ss_scl_hcnt[7:0]							
Reset	0	1	1	0	1	1	0	0

HDMI_PHY_I2CM_SS_SCL_HCNT_0_ADDR field descriptions

Field	Description
i2cmp_ss_scl_ hcnt[7:0]	Value after Reset: 8'h6C

34.5.223 PHY I2C Slow Speed SCL Low Level Control Register 1 (HDMI_PHY_I2CM_SS_SCL_LCNT_1_ADDR)

- Address Offset: 0x302D
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 302Dh offset = 12_302Dh

Bit	7	6	5	4	3	2	1	0
Read Write	i2cmp_ss_scl_lcnt[15:8]							
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_I2CM_SS_SCL_LCNT_1_ADDR field descriptions

Field	Description
i2cmp_ss_scl_ lcnt[15:8]	Value after Reset: 8'h00

34.5.224 PHY I2C Slow Speed SCL Low Level Control Register 0 (HDMI_PHY_I2CM_SS_SCL_LCNT_0_ADDR)

- Address Offset: 0x302E
- Size: 8 bits
- Value after Reset: 0x7F
- Access: Read/Write

Address: 12_0000h base + 302Eh offset = 12_302Eh

Bit	7	6	5	4	3	2	1	0
Read Write	i2cmp_ss_scl_lcnt[7:0]							
Reset	0	1	1	1	1	1	1	1

HDMI_PHY_I2CM_SS_SCL_LCNT_0_ADDR field descriptions

Field	Description
i2cmp_ss_scl_ lcnt[7:0]	Value after Reset: 8'h7F

34.5.225 PHY I2C Fast Speed SCL High Level Control Register 1 (HDMI_PHY_I2CM_FS_SCL_HCNT_1_ADDR)

- Address Offset: 0x302F
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 302Fh offset = 12_302Fh

Bit	7	6	5	4	3	2	1	0
Read Write	i2cmp_fs_scl_hcnt[15:8]							
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_I2CM_FS_SCL_HCNT_1_ADDR field descriptions

Field	Description
i2cmp_fs_scl_ hcnt[15:8]	Value after Reset: 8'h00

34.5.226 PHY I2C Fast Speed SCL High Level Control Register 0 (HDMI_PHY_I2CM_FS_SCL_HCNT_0_ADDR)

- Address Offset: 0x3030
- Size: 8 bits
- Value after Reset: 0x11
- Access: Read/Write

Address: 12_0000h base + 3030h offset = 12_3030h

Bit	7	6	5	4	3	2	1	0
Read Write	i2cmp_fs_scl_hcnt[7:0]							
Reset	0	0	0	1	0	0	0	1

HDMI_PHY_I2CM_FS_SCL_HCNT_0_ADDR field descriptions

Field	Description
i2cmp_fs_scl_ hcnt[7:0]	Value after Reset: 8'h11

34.5.227 PHY I2C Fast Speed SCL Low Level Control Register 1 (HDMI_PHY_I2CM_FS_SCL_LCNT_1_ADDR)

- Address Offset: 0x3031
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3031h offset = 12_3031h

Bit	7	6	5	4	3	2	1	0
Read Write	i2cmp_fs_scl_lcnt[15:8]							
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_I2CM_FS_SCL_LCNT_1_ADDR field descriptions

Field	Description
i2cmp_fs_scl_ lcnt[15:8]	Value after Reset: 8'h00

34.5.228 PHY I2C Fast Speed SCL Low Level Control Register 0 (HDMI_PHY_I2CM_FS_SCL_LCNT_0_ADDR)

- Address Offset: 0x3032
- Size: 8 bits
- Value after Reset: 0x24
- Access: Read/Write

Address: 12_0000h base + 3032h offset = 12_3032h

Bit	7	6	5	4	3	2	1	0
Read Write	i2cmp_fs_scl_lcnt[7:0]							
Reset	0	0	1	0	0	1	0	0

HDMI_PHY_I2CM_FS_SCL_LCNT_0_ADDR field descriptions

Field	Description
i2cmp_fs_scl_ lcnt[7:0]	Value after Reset: 8'h24

34.5.229 Audio Clock Regenerator N Value Register 1 (HDMI_AUD_N1)

For N expected values, refer to the HDMI 1.4a specification.

- Address Offset: 0x3200
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3200h offset = 12_3200h

Bit	7	6	5	4	3	2	1	0
Read Write	AudN[7:0]							
Reset	0	0	0	0	0	0	0	0

HDMI_AUD_N1 field descriptions

Field	Description
AudN[7:0]	HDMI Audio Clock Regenerator N value

34.5.230 Audio Clock Regenerator N Value Register 2 (HDMI_AUD_N2)

For N expected values, refer to the HDMI 1.4a specification.

- Address Offset: 0x3201
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3201h offset = 12_3201h

Bit	7	6	5	4	3	2	1	0
Read Write	AudN[15:8]							
Reset	0	0	0	0	0	0	0	0

HDMI_AUD_N2 field descriptions

Field	Description
AudN[15:8]	HDMI Audio Clock Regenerator N value

34.5.231 Audio Clock Regenerator N Value Register 3 (HDMI_AUD_N3)

For N expected values, refer to the HDMI 1.4a specification.

- Address Offset: 0x3202
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3202h offset = 12_3202h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				AudN[19:16]			
Reset	0	0	0	0	0	0	0	0

HDMI_AUD_N3 field descriptions

Field	Description
7–4 -	This field is reserved. Reserved
AudN[19:16]	HDMI Audio Clock Regenerator N value

34.5.232 AUD_CTS1 (HDMI_AUD_CTS1)

For CTS expected values, refer to the HDMI 1.4a specification.

- Address Offset: 0x3203
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3203h offset = 12_3203h

Bit	7	6	5	4	3	2	1	0
Read Write	audCTS[7:0]							
Reset	0	0	0	0	0	0	0	0

HDMI_AUD_CTS1 field descriptions

Field	Description
audCTS[7:0]	HDMI Audio Clock Regenerator CTS calculated value. This value can be manually set using the CTS_manual (AUD_CTS3) mechanism.

34.5.233 AUD_CTS2 (HDMI_AUD_CTS2)

For CTS expected values, refer to the HDMI 1.4a specification.

- Address Offset: 0x3204
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3204h offset = 12_3204h

Bit	7	6	5	4	3	2	1	0
Read Write	audCTS[15:8]							
Reset	0	0	0	0	0	0	0	0

HDMI_AUD_CTS2 field descriptions

Field	Description
audCTS[15:8]	HDMI Audio Clock Regenerator CTS calculated value. This value can be manually set using the CTS_manual (AUD_CTS3) mechanism.

34.5.234 AUD_CTS3 (HDMI_AUD_CTS3)

For CTS expected values, refer to the HDMI 1.4a specification.

- Address Offset: 0x3205
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3205h offset = 12_3205h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				audCTS[19:16]			
Reset	0	0	0	0	0	0	0	0

HDMI_AUD_CTS3 field descriptions

Field	Description
7–4 -	This field is reserved. Reserved
audCTS[19:16]	HDMI Audio Clock Regenerator CTS calculated value. This value can be manually set using the CTS_manual (AUD_CTS3) mechanism.

34.5.235 Audio DMA Start Register (HDMI_AHB_DMA_CONF0)

This register contains the software reset bit for the audio FIFOs. It also configures operating modes of the AHB master.

- Address Offset: 0x3600
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3600h offset = 12_3600h

Bit	7	6	5	4	3	2	1	0
Read Write	sw_fifo_rst	Reserved		hbr	enable_ hlock	incr_type[1:0]		burst_mode
Reset	0	0	0	0	0	0	0	0

HDMI_AHB_DMA_CONF0 field descriptions

Field	Description
7 sw_fifo_rst	This is the software reset bit for the audio and FIFOs clear. Writing 0'b does not result in any action.

Table continues on the next page...

HDMI_AHB_DMA_CONF0 field descriptions (continued)

Field	Description
	Writing 1'b to this register resets all audio FIFOs. Reading from this register always returns 0'b.
6–5 -	This field is reserved. Reserved
4 hbr	HBR packets enable. The HDMI TX sends the HBR packets. This bit is enabled when the audio frequency is higher than 192 KHz. If this bit is enabled, the number of channels configured in AHB_DMA_CONF1 is always 8.
3 enable_hlock	Enable request of locked burst AHB mechanism. 1 Enables the usage of ohlock for master request to arbiter of a locked complete burst.\ 0 Disables request of locked burst AHB mechanism
2–1 incr_type[1:0]	Forced size burst mode. 00 Corresponds to INCR4 fixed four beat incremental AHB burst mode. Only valid when burst_mode is high. 01 Corresponds to INCR8 fixed eight beat incremental AHB burst mode. Only valid when burst_mode is high. 10 Corresponds to INCR16 fixed 16 beat incremental AHB burst mode. Only valid when burst_mode is high. 11 Corresponds to INCR16 fixed 16 beat incremental AHB burst mode. Only valid when burst_mode is high.
0 burst_mode	Burst mode bit 1 Forces the burst mode to be fixed beat incremental burst mode designated by the incr_type[1:0] signal. 0 Normal operation is unspecified length incremental burst. It corresponds to INCR AHB burst mode.

34.5.236 AHB_DMA_START (HDMI_AHB_DMA_START)

The data_ buffer_ready bit field signals the AHB audio DMA to start accessing system memory in order to fetch data samples to store in the FIFO. After the operation starts, a new request for a DMA start is ignored until the DMA is stopped or it reaches the end address. Only in one of these situations will a new start request be acknowledged.

The first DMA burst request after data_buffer_ready configuration uses the initial_addr[31:0] as the ohaddr[31:0] and the MBURSTLENGTH[10:0] = AUDIO_FIFO_DEPTH if AUDIO_FIFO_DEPTH < 1024 or MBURSTLENGTH[10:0] = 1024 if AUDIO_FIFO_DEPTH >= 1024.

- Address Offset: 0x3601
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3601h offset = 12_3601h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved							data_buffer_ready
Reset	0	0	0	0	0	0	0	0

HDMI_AHB_DMA_START field descriptions

Field	Description
7–1 -	This field is reserved. Reserved
0 data_buffer_ready	Data buffer ready

34.5.237 Audio DMA Stop Register (HDMI_AHB_DMA_STOP)

The stop_dma_transaction bit field signals the AHB audio DMA to stop current memory access. After it stops, if a new start DMA operation is requested, the DMA engine restarts the memory access assuming the initial_addr[31:0] is programmed at AHB_DMA_STRADDR0 to AHB_DMA_STRADDR3.

- Address Offset: 0x3602
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3602h offset = 12_3602h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved							stop_dma_transaction
Reset	0	0	0	0	0	0	0	0

HDMI_AHB_DMA_STOP field descriptions

Field	Description
7–1 -	This field is reserved. Reserved
0 stop_dma_transaction	Stop DMA transaction

34.5.238 Audio DMA FIFO Threshold Register (HDMI_AHB_DMA_THRSLD)

This register defines the FIFO medium threshold occupation value.

After the AHB master successfully completes a burst transaction, the FIFO may stay remain full until the data fetch interface requests samples. The sample request from the FIFO using the data fetch mechanism drops the number of samples stored in the audio FIFO.

As soon as the number of samples in the FIFO drops lower than the fifo_threshold[7:0], the DMA engine requests a new burst of samples to the AHB master with a size (MBURSTLENGTH[10:0]) equal to AUDIO_FIFO_DEPTH minus fifo_threshold[7:0].

Therefore, the fifo_threshold[7:0] is the medium number of samples that should be available in the audio FIFO across the DMA operation.

- Address Offset: 0x3603
- Size: 8 bits
- Value after Reset: 0x04
- Access: Read/Write

Address: 12_0000h base + 3603h offset = 12_3603h

Bit	7	6	5	4	3	2	1	0
Read Write	fifo_threshold[7]							
Reset	0	0	0	1	0	0	0	0

HDMI_AHB_DMA_THRSLD field descriptions

Field	Description
fifo_threshold[7]	FIFO medium threshold occupation value

34.5.239 Audio DMA Start Address Register 0 (HDMI_AHB_DMA_STRADDR0)

These registers define the initial_addr[31:0] used to initiate the DMA burst read transactions upon data_buffer_ready configuration.

- Address Offset: 0x3604 to 0x3607
- Size: 8 bits per register
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3604h offset = 12_3604h

Bit	7	6	5	4	3	2	1	0
Read Write	initial_addr[7]							
Reset	0	0	0	0	0	0	0	0

HDMI_AHB_DMA_STRADDR0 field descriptions

Field	Description
initial_addr[7]	Defines init_addr[7:0] for bits 7-0 to initiate DMA burst transactions

34.5.240 Audio DMA Start Address Register 1 (HDMI_AHB_DMA_STRADDR1)

Address: 12_0000h base + 3605h offset = 12_3605h

Bit	7	6	5	4	3	2	1	0
Read Write	initial_addr[15]							
Reset	0	0	0	0	0	0	0	0

HDMI_AHB_DMA_STRADDR1 field descriptions

Field	Description
initial_addr[15]	Defines init_addr[15:8] for bits 7-0 to initiate DMA burst transactions

34.5.241 Audio DMA Start Address Register 2 (HDMI_AHB_DMA_STRADDR2)

Address: 12_0000h base + 3606h offset = 12_3606h

Bit	7	6	5	4	3	2	1	0
Read Write	initial_addr[23]							
Reset	0	0	0	0	0	0	0	0

HDMI_AHB_DMA_STRADDR2 field descriptions

Field	Description
initial_addr[23]	Defines init_addr[23:16] for bits 7-0 to initiate DMA burst transactions

34.5.242 Audio DMA Start Address Register 3 (HDMI_AHB_DMA_STRADDR3)

Address: 12_0000h base + 3607h offset = 12_3607h

Bit	7	6	5	4	3	2	1	0
Read Write	initial_addr[31]							
Reset	0	0	0	0	0	0	0	0

HDMI_AHB_DMA_STRADDR3 field descriptions

Field	Description
initial_addr[31]	Defines init_addr[31:24] for bits 7-0 to initiate DMA burst transactions

34.5.243 Audio DMA Stop Address Register 0 (HDMI_AHB_DMA_STPADDR0)

This registers define the final_addr[31:0] used as the final point to the DMA burst read transactions.

Upon data_buffer_ready configuration, the DMA engine starts requesting burst reads from the external system memory. Each burst read can have a maximum theoretical length of 1024 words (due to the AMBA AHB specification restriction). As an example, if the first burst transaction of the AHB audio DMA has a length of 16, then the second burst starts at address ohaddr[31:0] = initial_addr[31:0] + 16 and has a length of MBURSTLENGTH[10:0] = AUDIO_FIFO_DEPTH - fifo_threshold[7:0].

The DMA engine is responsible for incrementing the burst starting address and defining its corresponding burst length to reach the final_addr[31:0] address. The last burst request issued by the DMA engine takes into account that it should only request data until the final_addr[31:0] address (included) and for that should calculate the correct burst length.

After reaching the final_addr[31:0] address, the done interrupt is active to signal completion of DMA operation.

- Address Offset: 0x3608 to 0x360B
- Size: 8 bits per register
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3608h offset = 12_3608h

Bit	7	6	5	4	3	2	1	0
Read Write	final_addr[7]							
Reset	0	0	0	0	0	0	0	0

HDMI_AHB_DMA_STPADDR0 field descriptions

Field	Description
final_addr[7]	Defines final_addr[7:0] for bits 7-0 to initiate DMA burst transactions

34.5.244 Audio DMA Stop Address Register 1 (HDMI_AHB_DMA_STPADDR1)

Address: 12_0000h base + 3609h offset = 12_3609h

Bit	7	6	5	4	3	2	1	0
Read Write	final_addr[15]							
Reset	0	0	0	0	0	0	0	0

HDMI_AHB_DMA_STPADDR1 field descriptions

Field	Description
final_addr[15]	Defines final_addr[15:8] for bits 7-0 to initiate DMA burst transactions

34.5.245 Audio DMA Stop Address Register 2 (HDMI_AHB_DMA_STPADDR2)

Address: 12_0000h base + 360Ah offset = 12_360Ah

Bit	7	6	5	4	3	2	1	0
Read Write	final_addr[23]							
Reset	0	0	0	0	0	0	0	0

HDMI_AHB_DMA_STPADDR2 field descriptions

Field	Description
final_addr[23]	Defines final_addr[23:16] for bits 7-0 to initiate DMA burst transactions

34.5.246 Audio DMA Stop Address Register 3 (HDMI_AHB_DMA_STPADDR3)

Address: 12_0000h base + 360Bh offset = 12_360Bh

Bit	7	6	5	4	3	2	1	0
Read Write	final_addr[31]							
Reset	0	0	0	0	0	0	0	0

HDMI_AHB_DMA_STPADDR3 field descriptions

Field	Description
final_addr[31]	Defines final_addr[31:24] for bits 7-0 to initiate DMA burst transactions

34.5.247 Audio DMA Burst Start Address Register 0 (HDMI_AHB_DMA_BSTADDR0)

This read-only register composes the start address of the current burst operation. As an example, if the first burst transaction of the AHB audio DMA as a length of 16, then the second burst should start at address ohaddr[31:0] = initial_addr[31:0] + 16. While this burst is being executed, burst_start_addr[31:0] = haddr[31:0] = initial_addr[31:0] + 16.

- Address Offset: 0x360C to 0x360F
- Size: 8 bits per register
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 360Ch offset = 12_360Ch

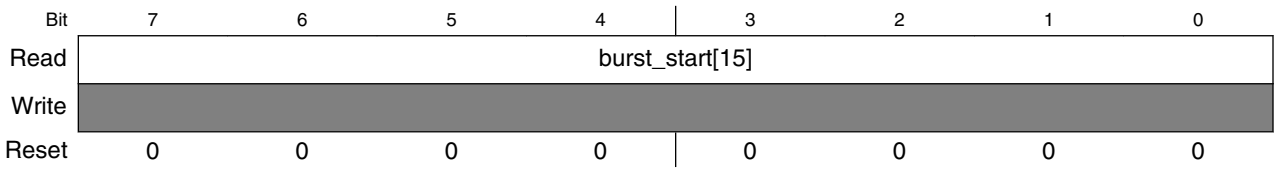
Bit	7	6	5	4	3	2	1	0
Read	burst_start[7]							
Write								
Reset	0	0	0	0	0	0	0	0

HDMI_AHB_DMA_BSTADDR0 field descriptions

Field	Description
burst_start[7]	Start address for the current burst operation

34.5.248 Audio DMA Burst Start Address Register 1 (HDMI_AHB_DMA_BSTADDR1)

Address: 12_0000h base + 360Dh offset = 12_360Dh

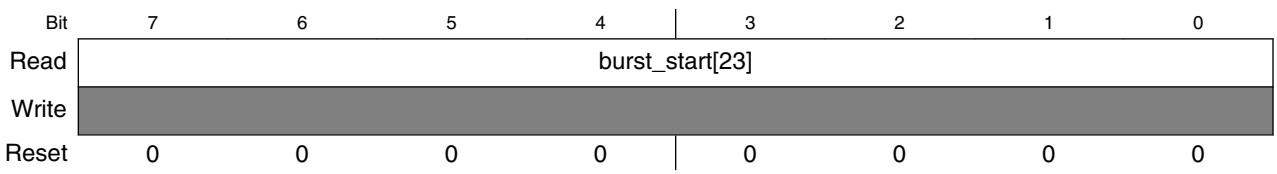


HDMI_AHB_DMA_BSTADDR1 field descriptions

Field	Description
burst_start[15]	Start address for the current burst operation

34.5.249 Audio DMA Burst Start Address Register 2 (HDMI_AHB_DMA_BSTADDR2)

Address: 12_0000h base + 360Eh offset = 12_360Eh

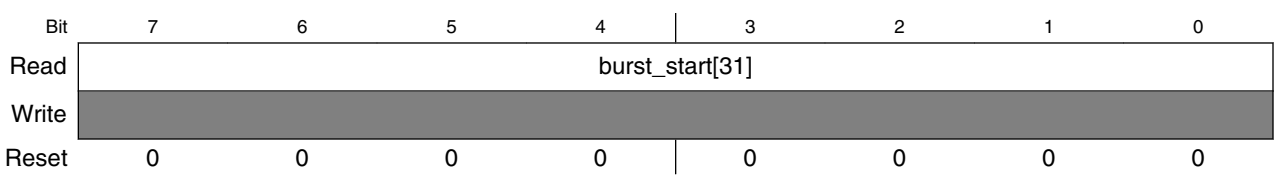


HDMI_AHB_DMA_BSTADDR2 field descriptions

Field	Description
burst_start[23]	Start address for the current burst operation

34.5.250 Audio DMA Burst Start Address Register 3 (HDMI_AHB_DMA_BSTADDR3)

Address: 12_0000h base + 360Fh offset = 12_360Fh



HDMI_AHB_DMA_BSTADDR3 field descriptions

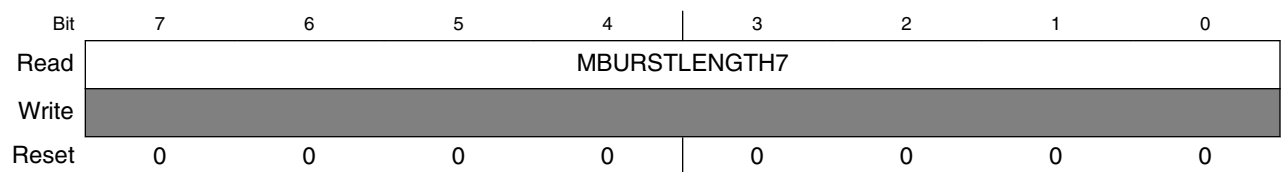
Field	Description
burst_start[31]	Start address for the current burst operation

34.5.251 Audio DMA Burst Length Register 0 (HDMI_AHB_DMA_MBLENGTH0)

These registers hold the length of the current burst operation. As an example, if the first burst transaction of the AHB audio DMA is a length of 8, then the second burst should start at address ohaddr[31:0] = initial_addr[31:0] + 8. It will also have length MBURSTLENGTH[10:0] = AUDIO_FIFO_DEPTH - fifo_threshold[7:0] while this burst is being executed, MBURSTLENGTH[10:0] = AUDIO_FIFO_DEPTH - fifo_threshold[7:0].

- Address Offset: 0x3610 to 0x3611
- Size: 8 bits per register
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 3610h offset = 12_3610h



HDMI_AHB_DMA_MBLENGTH0 field descriptions

Field	Description
MBURSTLENGTH7	Requested burst length

34.5.252 Audio DMA Burst Length Register 1 (HDMI_AHB_DMA_MBLENGTH1)

Address: 12_0000h base + 3611h offset = 12_3611h

Bit	7	6	5	4
Read	Reserved			
Write				
Reset	0	0	0	0

Bit	3	2	1	0
Read	Reserved	MBURSTLENGTH10	MBURSTLENGTH9	MBURSTLENGTH8
Write				
Reset	0	0	0	0

HDMI_AHB_DMA_MBLENGTH1 field descriptions

Field	Description
7–3 -	This field is reserved. Reserved
2 MBURSTLENGTH10	Requested burst length
1 MBURSTLENGTH9	Requested burst length
0 MBURSTLENGTH8	Requested burst length

34.5.253 Audio DMA Interrupt Status Register (HDMI_AHB_DMA_STAT)

This register contains the status bits of the following interrupts:

- Address Offset: 0x3612
- Size: 8 bits per register
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 3612h offset = 12_3612h

Bit	7	6	5	4
Read	statdone	statretrysplit	statlostownership	staterror
Write				
Reset	0	0	0	0

Bit	3	2	1	0
Read	Reserved	statthrfifoempty	statfifofull	statfifoempty
Write				
Reset	0	0	0	0

HDMI_AHB_DMA_STAT field descriptions

Field	Description
7 statdone	Status of DMA end of operation interrupt. Active when DMA engine reaches final_addr[15:0] or when stop DMA operation is activated.
6 statretrysplit	Status of retry/split interrupt. Active when AHB master receives a RETRY or SPLIT response from slave.
5 statlostownership	Status of master lost ownership when in burst transfer. Active when AHB master loses BUS ownership within the course of a burst transfer.
4 staterror	Status of error interrupt. Active when slave indicates error through the isresp[1:0].
3 -	This field is reserved. Reserved
2 statthrfifoempty	Status of audio FIFO empty when audio FIFO has less than four samples.
1 statfifofull	Status of audio FIFO full interrupt.
0 statfifoempty	Status of audio FIFO empty interrupt.

34.5.254 Audio DMA Interrupt Register (HDMI_AHB_DMA_INT)

This register contains the interrupt bits of the following interrupts:

- Address Offset: 0x3613
- Size: 8 bits per register
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 3613h offset = 12_3613h

Bit	7	6	5	4
Read	intdone	intretrysplit	intlostownership	interror
Write				
Reset	0	0	0	0

Bit	3	2	1	0
Read	Reserved	intthrfifoempty	intfifofull	intfifoempty
Write				
Reset	0	0	0	0

HDMI_AHB_DMA_INT field descriptions

Field	Description
7 intdone	DMA end of operation interrupt. Active when DMA engine reaches final_addr[15:0] or when stop DMA operation is activated.
6 intretrysplit	Retry/split interrupt. Active when AHB master receives a RETRY or SPLIT response from slave.
5 intlostownership	Master lost ownership interrupt when in burst transfer. Active when AHB master loses BUS ownership within the course of a burst transfer.
4 interror	Error interrupt. Active when slave indicates error through the isresp[1:0].
3 -	This field is reserved. Reserved
2 intthrfifoempty	Audio FIFO empty interrupt when audio FIFO has less than four samples.
1 intfifofull	Audio FIFO full interrupt.
0 intfifoempty	Audio FIFO empty interrupt.

34.5.255 Audio DMA Mask Interrupt Register (HDMI_AHB_DMA_MASK)

Mask for each of the interrupts present in the AHB audio DMA module. For usage information, see Audio DMA Interrupt Register (HDMI_AHB_DMA_INT) ."

- Address Offset: 0x3614
- Size: 8 bits per register
- Value after Reset: 0xF7
- Access: Read/Write

Address: 12_0000h base + 3614h offset = 12_3614h

Bit	7	6	5	4
Read Write	done_mask	retrysplit_mask	lostownership_mask	error_mask
Reset	0	0	0	1

Bit	3	2	1	0
Read Write	Reserved	fifo_thrempty_mask	fifo_full_mask	fifo_empty_mask
Reset	0	0	0	1

HDMI_AHB_DMA_MASK field descriptions

Field	Description
7 done_mask	DMA end of operation interrupt mask. Active when DMA engine reaches final_addr[15:0] or when stop DMA operation is activated.

Table continues on the next page...

HDMI_AHB_DMA_MASK field descriptions (continued)

Field	Description
6 retrysplit_mask	Retry/split interrupt mask. Active when AHB master receives a RETRY or SPLIT response from slave.
5 lostownership_ mask	Master lost ownership interrupt mask when in burst transfer. Active when AHB master loses BUS ownership within the course of a burst transfer.
4 error_mask	Error interrupt mask. Active when slave indicates error through the isresp[1:0].
3 -	This field is reserved. Reserved
2 fifo_thrempty_ mask	Audio FIFO empty interrupt mask when audio FIFO has less than four samples.
1 fifo_full_mask	Audio FIFO full interrupt mask.
0 fifo_empty_mask	Audio FIFO empty interrupt mask.

34.5.256 Audio DMA Polarity Interrupt Register (HDMI_AHB_DMA_POL)

Polarity for each of the interrupts present in the AHB audio DMA module. For usage information, see Audio DMA Interrupt Register (HDMI_AHB_DMA_INT) ."

- Address Offset: 0x3615
- Size: 8 bits per register
- Value after Reset: 0xF7
- Access: Read/Write

Address: 12_0000h base + 3615h offset = 12_3615h

Bit	7	6	5	4
Read Write	done_polarity	retrysplit_polarity	lostownership_polarity	error_polarity
Reset	0	0	0	1

Bit	3	2	1	0
Read Write	Reserved	fifo_thrfifoempty_ polarity	fifo_full_polarity	fifo_empty_polarity
Reset	0	0	0	1

HDMI_AHB_DMA_POL field descriptions

Field	Description
7 done_polarity	DMA end of operation interrupt mask. Active when DMA engine reaches final_addr[15:0] or when stop DMA operation is activated.

Table continues on the next page...

HDMI_AHB_DMA_POL field descriptions (continued)

Field	Description
6 retrysplit_polarity	Retry/split interrupt mask. Active when AHB master receives a RETRY or SPLIT response from slave.
5 lostownership_ polarity	Master lost ownership interrupt mask when in burst transfer. Active when AHB master loses BUS ownership within the course of a burst transfer.
4 error_polarity	Error interrupt mask. Active when slave indicates error through the isresp[1:0].
3 -	This field is reserved. Reserved
2 fifo_thrfifoempty_ polarity	Audio FIFO empty interrupt mask when audio FIFO has less than four samples.
1 fifo_full_polarity	Audio FIFO full interrupt mask.
0 fifo_empty_ polarity	Audio FIFO empty interrupt mask.

34.5.257 Audio DMA Channel Enable Configuration Register 1 (HDMI_AHB_DMA_CONF1)

- Address Offset: 0x3616
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 3616h offset = 12_3616h

Bit	7	6	5	4	3	2	1	0
Read Write	CH_IN_EN7	CH_IN_EN6	CH_IN_EN5	CH_IN_EN4	CH_IN_EN3	CH_IN_EN2	CH_IN_EN1	CH_IN_EN0
Reset	0	0	0	0	0	0	0	0

HDMI_AHB_DMA_CONF1 field descriptions

Field	Description
7 CH_IN_EN7	Channel 7 enable bit 1 Channel enabled 0 Channel disabled
6 CH_IN_EN6	Channel 6 enable bit 1 Channel enabled 0 Channel disabled

Table continues on the next page...

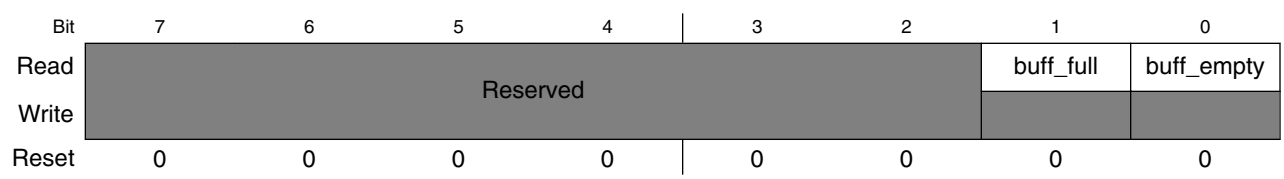
HDMI_AHB_DMA_CONF1 field descriptions (continued)

Field	Description
5 CH_IN_EN5	Channel 5 enable bit 1 Channel enabled 0 Channel disabled
4 CH_IN_EN4	Channel 4 enable bit 1 Channel enabled 0 Channel disabled
3 CH_IN_EN3	Channel 3 enable bit 1 Channel enabled 0 Channel disabled
2 CH_IN_EN2	Channel 2 enable bit 1 Channel enabled 0 Channel disabled
1 CH_IN_EN1	Channel 1 is always enabled.
0 CH_IN_EN0	Channel 0 is always enabled.

34.5.258 Audio DMA Buffer Interrupt Status Register (HDMI_AHB_DMA_BUFFSTAT)

- Address Offset: 0x3617
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 3617h offset = 12_3617h



HDMI_AHB_DMA_BUFFSTAT field descriptions

Field	Description
7–2 -	This field is reserved. Reserved
1 buff_full	Buffer full flag status

Table continues on the next page...

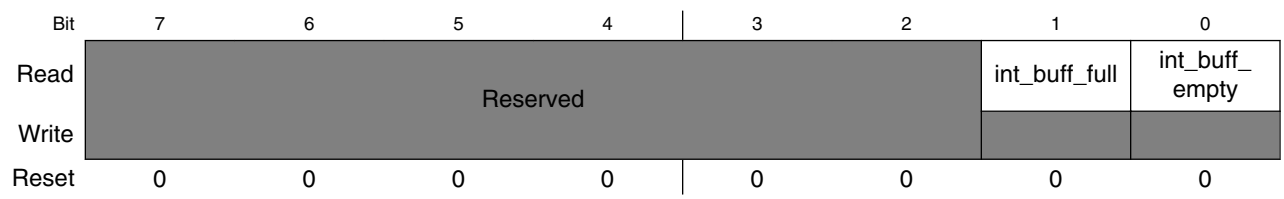
HDMI_AHB_DMA_BUFFSTAT field descriptions (continued)

Field	Description
0 buff_empty	Buffer empty flag status

34.5.259 Audio DMA Buffer Interrupt Register (HDMI_AHB_DMA_BUFFINT)

- Address Offset: 0x3618
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 3618h offset = 12_3618h



HDMI_AHB_DMA_BUFFINT field descriptions

Field	Description
7–2 -	This field is reserved. Reserved
1 int_buff_full	Buffer full flag interrupt
0 int_buff_empty	Buffer empty flag interrupt

34.5.260 Audio DMA Buffer Mask Interrupt Register (HDMI_AHB_DMA_BUFFMASK)

- Address Offset: 0x3619
- Size: 8 bits
- Value after Reset: 0x03
- Access: Read/Write

Address: 12_0000h base + 3619h offset = 12_3619h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved						int_buff_full	int_buff_ empty
Reset	0	0	0	1	0	0	0	1

HDMI_AHB_DMA_BUFFMASK field descriptions

Field	Description
7–2 -	This field is reserved. Reserved
1 int_buff_full	Buffer full flag mask
0 int_buff_empty	Buffer empty flag mask

34.5.261 Audio DMA Buffer Polarity Interrupt Register (HDMI_AHB_DMA_BUFFPOL)

- Address Offset: 0x361A
- Size: 8 bits
- Value after Reset: 0x03
- Access: Read/Write

Address: 12_0000h base + 361Ah offset = 12_361Ah

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved						int_buff_full	int_buff_ empty
Reset	0	0	0	1	0	0	0	1

HDMI_AHB_DMA_BUFFPOL field descriptions

Field	Description
7–2 -	This field is reserved. Reserved
1 int_buff_full	Buffer full flag polarity
0 int_buff_empty	Buffer empty flag polarity

34.5.262 Main Controller Synchronous Clock Domain Disable Register (HDMI_MC_CLKDIS)

Main controller synchronous disable control per clock domain. Upon release of synchronous disable the corresponding sw reset NRZ request signal, to that domain, is toggled asking to the output for a synchronized active low reset to be generated to that domain.

- Address Offset: 0x4001
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4001h offset = 12_4001h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved	Reserved	cecclk_ disable	cscclk_ disable	audclk_ disable	prepclk_ disable	tmdsclk_ disable	pixelclk_ disable
Reset	0	0	0	0	0	0	0	0

HDMI_MC_CLKDIS field descriptions

Field	Description
7 -	This field is reserved. Reserved
6 hdcpclk_disable	This field is reserved. Reserved
5 cecclk_disable	CEC Engine clock synchronous disable signal.
4 cscclk_disable	Color Space Converter clock synchronous disable signal.
3 audclk_disable	Audio Sampler clock synchronous disable signal.
2 prepclk_disable	Pixel Repetition clock synchronous disable signal.

Table continues on the next page...

HDMI_MC_CLKDIS field descriptions (continued)

Field	Description
1 tmdsclk_disable	TMDS clock synchronous disable signal.
0 pixelclk_disable	Pixel clock synchronous disable signal.

34.5.263 Main Controller Software Reset Register (HDMI_MC_SWRSTZREQ)

Main controller software reset request per clock domain. Writing zero to a bit of this register results in an NRZ signal toggle at sfrclk rate to an output signal that indicates a software reset request. This toggle must be used to generate a synchronized reset to de corresponding domain, with at least 1 clock cycle. Register defaults back to 0xFF.

- Address Offset: 0x4002
- Size: 8 bits
- Value after Reset: 0xFF
- Access: Read/Write

Address: 12_0000h base + 4002h offset = 12_4002h

Bit	7	6	5	4
Read Write	Reserved	cecswrst_req	Reserved	
Reset	1	1	1	1

Bit	3	2	1	0
Read Write	Reserved	prepswrst_req	tmdsswrst_req	pixelswrst_req
Reset	1	1	1	1

HDMI_MC_SWRSTZREQ field descriptions

Field	Description
7 -	This field is reserved. Reserved
6 cecswrst_req	CEC software reset request. Defaults back to 1b after reset request.
5–3 -	This field is reserved. Reserved
2 prepswrst_req	Pixel Repetition clock synchronous disable signal.
1 tmdsswrst_req	TMDS software reset request. Defaults back to 1b after reset request.
0 pixelswrst_req	Pixel software reset request. Defaults back to 1b after reset request.

34.5.264 Main Controller Feed Through Control Register (HDMI_MC_FLOWCTRL)

- Address Offset: 0x4004
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4004h offset = 12_4004h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved							Feed_ through_off
Reset	0	0	0	0	0	0	0	0

HDMI_MC_FLOWCTRL field descriptions

Field	Description
7–1 -	This field is reserved. Reserved
0 Feed_through_off	Video path Feed Through enable bit: 1 Color Space Converter is in the video data path. 0 Color Space Converter is bypassed (not in the video data path).

34.5.265 Main Controller PHY Reset Register (HDMI_MC_PHYRSTZ)

- Address Offset: 0x4005
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4005h offset = 12_4005h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved							phyrstz
Reset	0	0	0	0	0	0	0	0

HDMI_MC_PHYRSTZ field descriptions

Field	Description
7–1 -	This field is reserved. Reserved
0 phyrstz	HDMI Source PHY active low reset control.

34.5.266 Main Controller Clock Present Register (HDMI_MC_LOCKONCLOCK)

- Address Offset: 0x4006
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Clear on Write

Address: 12_0000h base + 4006h offset = 12_4006h

Bit	7	6	5	4	3	2	1	0
Read	Reserved	pclk	tclktclk	prepclk	Reserved			cecclk
Write		w1c	w1c	w1c				w1c
Reset	0	0	0	0	0	0	0	0

HDMI_MC_LOCKONCLOCK field descriptions

Field	Description
7 -	This field is reserved. Reserved.
6 pclk	Pixel clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position.
5 tclktclk	TMDS clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position
4 prepclk	Pixel repetition clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position.
3–1 -	This field is reserved. Reserved.
0 cecclk	CEC clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position.

34.5.267 Main Controller HEAC PHY Reset Register (HDMI_MC_HEACPHY_RST)

- Address Offset: 0x4007
- Size: 8 bits
- Value after Reset: N/A
- Access: Read/Write

Address: 12_0000h base + 4007h offset = 12_4007h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved							heacphyrst
Reset	0	0	0	0	0	0	0	0

HDMI_MC_HEACPHY_RST field descriptions

Field	Description
7–1 -	This field is reserved. Reserved
0 heacphyrst	HEAC PHY reset (active high)

34.5.268 Color Space Converter Interpolation and Decimation Configuration Register (HDMI_CSC_CFG)

Color Space Conversion configuration register. Configures YCC422 to YCC444 interpolation mode and YCC444 to YCC422 decimation mode.

- Address Offset: 0x4100
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4100h offset = 12_4100h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved		INTMODE		Reserved		DECMODE	
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_CFG field descriptions

Field	Description
7–6 -	This field is reserved. Reserved
5–4 INTMODE	Chroma interpolation configuration: 00 interpolation disabled 01 $H_u(z) = 1 + z^{-1}$ 10 $H_u(z)=1/2 + Z^{-1} +1/2\ z^{-2}$ 11 interpolation disabled.
3–2 -	This field is reserved. Reserved
DECMODE	Chroma decimation configuration: DECMODE[1:0] Chroma decimation 00 decimation disabled 01 H d? z?=1 10 H d? z?=1/ 4?1/ 2 z?1?1/4 z?2 11 H d? z?Þ211=?5?12 z?2?22 z?4?39 z?6?65 z?8?109 z?10?204 z?12?648 z?14?1024 z?15?648 z?16?204 z?18?109 z?20?65 z?22?39 z?24?22 z?26?12 z?28?5 z?30 00 decimation disabled 01 $H_d(z) = 1$ 10 $H_d(Z)=1/4 + 1/2z^{-1} 1+1/4z^{-2}$ $11H_d(z)x2^{11} =-5+12z^{-2} +22z^{-4} +39z^{-8} +109z^{-10} -204z^{-12} +648z^{-14} +1024z^{-15} +648z^{-16} -204z^{-18} +109z^{-20} -65z^{-22} +39z^{-24} -22z^{-26} +12z^{-28} -5z^{-30}$

34.5.269 Color Space Converter Scale and Deep Color Configuration Register (HDMI_CSC_SCALE)

- Address Offset: 0x4101
- Size: 8 bits
- Value after Reset: 0x01
- Access: Read/Write

$$\begin{bmatrix} G \\ R \\ B \end{bmatrix} = 2^{\text{csc scale} - 12} \times \begin{bmatrix} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \\ C_1 & C_2 & C_3 \end{bmatrix} + 2^{\text{csc scale}} \times \begin{bmatrix} A_4 \\ B_4 \\ C_4 \end{bmatrix}$$

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = 2^{\text{cscscale} - 12} \times \begin{bmatrix} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \\ C_1 & C_2 & C_3 \end{bmatrix} + 2^{\text{cscscale}} \times \begin{bmatrix} A_4 \\ B_4 \\ C_4 \end{bmatrix}$$

Figure 34-17. CSC Conversion Functions

Address: 12_0000h base + 4101h offset = 12_4101h

Bit	7	6	5	4	3	2	1	0
Read Write	csc_colorde_pth[3:0]				Reserved		-	
Reset	0	0	0	0	0	0	0	1

HDMI_CSC_SCALE field descriptions

Field	Description
7–4 csc_colorde_pth[3:0]	Color space converter color depth configuration: Other: Reserved. 0000 24 bit per pixel video (8 bit per component). 0100 24 bit per pixel video (8 bit per component). 0101 30 bit per pixel video (10 bit per component). 0110 36 bit per pixel video (12 bit per component). 0111 48 bit per pixel video (16 bit per component).
3–2 -	This field is reserved. Reserved
-	Defines the cscscale[1:0] scale factor to apply to all coefficients in Color Space Conversion. This scale factor is expressed in the number of left shifts to apply to each of the coefficients, ranging from 0 to 2.

34.5.270 CSC_COEF_A1_MSB (HDMI_CSC_COEF_A1_MSB)

Color Space Conversion A1 coefficient.

- Address Offset: 0x4102
- Size: 8 bits
- Value after Reset: 0x20
- Access: Read/Write

Address: 12_0000h base + 4102h offset = 12_4102h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_A1_MSB							
Reset	0	0	1	0	0	0	0	0

HDMI_CSC_COEF_A1_MSB field descriptions

Field	Description
CSC_COEF_A1_MSB	Color Space Conversion A1 MSB coefficient.

34.5.271 CSC_COEF_A1_LSB (HDMI_CSC_COEF_A1_LSB)

Color Space Conversion A1 coefficient.

- Address Offset: 0x4103
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4103h offset = 12_4103h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_A1_LSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_A1_LSB field descriptions

Field	Description
CSC_COEF_A1_LSB	Color Space Conversion A1 LSB coefficient

34.5.272 CSC_COEF_A2_MSB (HDMI_CSC_COEF_A2_MSB)

Color Space Conversion A2 coefficient.

- Address Offset: 0x4104
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4104h offset = 12_4104h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_A2_MSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_A2_MSB field descriptions

Field	Description
CSC_COEF_A2_MSB	Color Space Conversion A2 MSB coefficient.

34.5.273 CSC_COEF_A2_LSB (HDMI_CSC_COEF_A2_LSB)

Color Space Conversion A2 coefficient.

- Address Offset: 0x4105
- Size: 8 bits

- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4105h offset = 12_4105h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_A2_LSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_A2_LSB field descriptions

Field	Description
CSC_COEF_A2_LSB	Color Space Conversion A2 LSB coefficient.

34.5.274 CSC_COEF_A3_MSB (HDMI_CSC_COEF_A3_MSB)

Color Space Conversion A3 coefficient.

- Address Offset: 0x4106
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4106h offset = 12_4106h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_A3_MSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_A3_MSB field descriptions

Field	Description
CSC_COEF_A3_MSB	Color Space Conversion A3 MSB coefficient.

34.5.275 CSC_COEF_A3_LSB (HDMI_CSC_COEF_A3_LSB)

Color Space Conversion A3 coefficient.

- Address Offset: 0x4107
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4107h offset = 12_4107h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_A3_LSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_A3_LSB field descriptions

Field	Description
CSC_COEF_A3_LSB	Color Space Conversion A3 LSB coefficient.

34.5.276 CSC_COEF_A4_MSB (HDMI_CSC_COEF_A4_MSB)

Color Space Conversion A4 coefficient.

- Address Offset: 0x4108
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4108h offset = 12_4108h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_A4_MSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_A4_MSB field descriptions

Field	Description
CSC_COEF_A4_MSB	Color Space Conversion A4 MSB coefficient.

34.5.277 CSC_COEF_A4_LSB (HDMI_CSC_COEF_A4_LSB)

Color Space Conversion A4 coefficient.

- Address Offset: 0x4109
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4109h offset = 12_4109h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_A4_LSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_A4_LSB field descriptions

Field	Description
CSC_COEF_A4_LSB	Color Space Conversion A4 LSB coefficient.

34.5.278 CSC_COEF_B1_MSB (HDMI_CSC_COEF_B1_MSB)

Color Space Conversion B1 coefficient.

- Address Offset: 0x410A
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 410Ah offset = 12_410Ah

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_B1_MSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_B1_MSB field descriptions

Field	Description
CSC_COEF_B1_MSB	Color Space Conversion B1 MSB coefficient.

34.5.279 CSC_COEF_B1_LSB (HDMI_CSC_COEF_B1_LSB)

Color Space Conversion B1 coefficient.

- Address Offset: 0x410B
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 410Bh offset = 12_410Bh

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_B1_LSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_B1_LSB field descriptions

Field	Description
CSC_COEF_B1_LSB	Color Space Conversion B1 LSB coefficient.

34.5.280 CSC_COEF_B2_MSB (HDMI_CSC_COEF_B2_MSB)

Color Space Conversion B2 coefficient.

- Address Offset: 0x410C
- Size: 8 bits
- Value after Reset: 0x20
- Access: Read/Write

Address: 12_0000h base + 410Ch offset = 12_410Ch

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_B2_MSB							
Reset	0	0	1	0	0	0	0	0

HDMI_CSC_COEF_B2_MSB field descriptions

Field	Description
CSC_COEF_B2_MSB	Color Space Conversion B2 MSB coefficient.

34.5.281 CSC_COEF_B2_LSB (HDMI_CSC_COEF_B2_LSB)

Color Space Conversion B2 coefficient.

- Address Offset: 0x410D
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 410Dh offset = 12_410Dh

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_B2_LSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_B2_LSB field descriptions

Field	Description
CSC_COEF_B2_LSB	Color Space Conversion B2 LSB coefficient.

34.5.282 CSC_COEF_B3_MSB (HDMI_CSC_COEF_B3_MSB)

Color Space Conversion B3 coefficient.

- Address Offset: 0x410E
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 410Eh offset = 12_410Eh

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_B3_MSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_B3_MSB field descriptions

Field	Description
CSC_COEF_B3_MSB	Color Space Conversion B3 MSB coefficient.

34.5.283 CSC_COEF_B3_LSB (HDMI_CSC_COEF_B3_LSB)

Color Space Conversion B3 coefficient.

- Address Offset: 0x410F
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 410Fh offset = 12_410Fh

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_B3_LSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_B3_LSB field descriptions

Field	Description
CSC_COEF_B3_LSB	Color Space Conversion B3 LSB coefficient.

34.5.284 CSC_COEF_B4_MSB (HDMI_CSC_COEF_B4_MSB)

Color Space Conversion B4 coefficient.

- Address Offset: 0x4110
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4110h offset = 12_4110h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_B4_MSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_B4_MSB field descriptions

Field	Description
CSC_COEF_B4_MSB	Color Space Conversion B4 MSB coefficient.

34.5.285 CSC_COEF_B4_LSB (HDMI_CSC_COEF_B4_LSB)

Color Space Conversion B4 coefficient.

- Address Offset: 0x4111
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4111h offset = 12_4111h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_B4_LSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_B4_LSB field descriptions

Field	Description
CSC_COEF_B4_LSB	Color Space Conversion B4 LSB coefficient.

34.5.286 CSC_COEF_C1_MSB (HDMI_CSC_COEF_C1_MSB)

Color Space Conversion C1 coefficient.

- Address Offset: 0x4112
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4112h offset = 12_4112h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_C1_MSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_C1_MSB field descriptions

Field	Description
CSC_COEF_C1_MSB	Color Space Conversion C1 MSB coefficient.

34.5.287 CSC_COEF_C1_LSB (HDMI_CSC_COEF_C1_LSB)

Color Space Conversion C1 coefficient.

- Address Offset: 0x4113
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4113h offset = 12_4113h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_C1_LSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_C1_LSB field descriptions

Field	Description
CSC_COEF_C1_LSB	Color Space Conversion C1 LSB coefficient.

34.5.288 CSC_COEF_C2_MSB (HDMI_CSC_COEF_C2_MSB)

Color Space Conversion C2 coefficient.

- Address Offset: 0x4114
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4114h offset = 12_4114h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_C2_MSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_C2_MSB field descriptions

Field	Description
CSC_COEF_C2_MSB	Color Space Conversion C2 MSB coefficient.

34.5.289 CSC_COEF_C2_LSB (HDMI_CSC_COEF_C2_LSB)

Color Space Conversion C2 coefficient.

- Address Offset: 0x4115
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4115h offset = 12_4115h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_C2_LSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_C2_LSB field descriptions

Field	Description
CSC_COEF_C2_LSB	Color Space Conversion C2 LSB coefficient.

34.5.290 CSC_COEF_C3_MSB (HDMI_CSC_COEF_C3_MSB)

Color Space Conversion C3 coefficient.

- Address Offset: 0x4116
- Size: 8 bits
- Value after Reset: 0x20
- Access: Read/Write

Address: 12_0000h base + 4116h offset = 12_4116h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_C3_MSB							
Reset	0	0	1	0	0	0	0	0

HDMI_CSC_COEF_C3_MSB field descriptions

Field	Description
CSC_COEF_C3_MSB	Color Space Conversion C3 MSB coefficient.

34.5.291 CSC_COEF_C3_LSB (HDMI_CSC_COEF_C3_LSB)

Color Space Conversion C3 coefficient.

- Address Offset: 0x4117
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4117h offset = 12_4117h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEF_C3_LSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEF_C3_LSB field descriptions

Field	Description
CSC_COEF_C3_LSB	Color Space Conversion C3 LSB coefficient.

34.5.292 CSC_COEFC4_MSB (HDMI_CSC_COEFC4_MSB)

Color Space Conversion C4 coefficient.

- Address Offset: 0x4118
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4118h offset = 12_4118h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEFC4_MSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEFC4_MSB field descriptions

Field	Description
CSC_COEFC4_MSB	Color Space Conversion C4 MSB coefficient.

34.5.293 CSC_COEFC4_LSB (HDMI_CSC_COEFC4_LSB)

Color Space Conversion C4 coefficient.

- Address Offset: 0x4119
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 4119h offset = 12_4119h

Bit	7	6	5	4	3	2	1	0
Read Write	CSC_COEFC4_LSB							
Reset	0	0	0	0	0	0	0	0

HDMI_CSC_COEFC4_LSB field descriptions

Field	Description
CSC_COEFC4_ LSB	Color Space Conversion C4 LSB coefficient.

34.5.294 CEC_CTRL (HDMI_CEC_CTRL)

CEC registers control the CEC feature that is implemented in HDMI TX. They perform various functions like controlling, monitoring, and buffering data for the transmitter and the receiver.

This register handles the main control of the CEC initiator.

- Address Offset: 0x7D00
- Size: 8 bits
- Value after Reset: 0x02
- Access: Read/Write

Address: 12_0000h base + 7D00h offset = 12_7D00h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved			STANDBY	BC_NACK	FRAME_TYP		SEND
Reset	0	0	0	0	0	0	1	0

HDMI_CEC_CTRL field descriptions

Field	Description
7–5 -	This field is reserved. Reserved
4 STANDBY	Standby bit 0 CEC controller responds the ACK to all messages. 1 CEC controller responds with ACK to all ping messages (only when the EOM is received) and responds with NACK to all other messages, generating wake-up status for selected opcodes. Attention that the NACK will only be posted on the last block of a frame.
3 BC_NACK	Broadcast NACK bit 0 Reset by software to ACK the received broadcast message. 1 Set by software to NACK the received broadcast message. This bit holds till software resets. The broadcasts will be answered with 1'b0. It means the follower reject the message.

Table continues on the next page...

HDMI_CEC_CTRL field descriptions (continued)

Field	Description
2–1 FRAME_TYP	Frame Type bit 00 Signal Free Time = 3-bit periods. Previous attempt to send frame is unsuccessful. 01 Signal Free Time = 5-bit periods. New initiator wants to send a frame. 10 Signal Free Time = 7-bit periods. Present initiator wants to send another frame immediately after its previous frame. (spec CEC 9.1) 11 Illegal value. If software write this value, hardware will set the value to the default 2'b01.
0 SEND	Send bit 0 Reset to 0 by hardware when the CEC transmission is done (no matter successful or failed). It can also work as an indicator checked by software to see whether the transmission is finished. 1 Set by software to trigger CEC sending a frame as an initiator. This bit keeps at 1 while the transmission is going on.

34.5.295 CEC_MASK (HDMI_CEC_MASK)

This read/write register masks/unmasks the interrupt events. When the bit is set to 1 (masked), the corresponding event will not trigger an interrupt signal at the system interface. When the bit is reset to 0, the interrupt event is unmasked.

- Address Offset: 0x7D02
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 7D02h offset = 12_7D02h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved	WAKEUP_MASK	ERROR_FOLL_MASK	ERROR_INIT_MASK	ARB_LOST_MASK	NACK_MASK	EOM_MASK	DONE_MASK
Reset	0	0	0	0	0	0	0	0

HDMI_CEC_MASK field descriptions

Field	Description
7 -	This field is reserved. Reserved
6 WAKEUP_MASK	Follower wake-up signal mask
5 ERROR_FOLL_MASK	An error is notified by a follower. Abnormal logic data bit error (for follower).

Table continues on the next page...

HDMI_CEC_MASK field descriptions (continued)

Field	Description
4 ERROR_INIT_ MASK	An error is detected on cec line (for initiator only).
3 ARB_LOST_ MASK	The initiator losses the CEC line arbitration to a second initiator. (specification CEC 9).
2 NACK_MASK	A frame is not acknowledged in a directly addressed message. Or a frame is negatively acknowledged in a broadcast message (for initiator only).
1 EOM_MASK	EOM is detected so that the received data is ready in the receiver data buffer (for follower only).
0 DONE_MASK	The current transmission is successful (for initiator only).

34.5.296 CEC_ADDR_L (HDMI_CEC_ADDR_L)

CEC_ADDR_L and CEC_ADDR_H registers indicate the logical address(es) allocated to the CEC device. The logical address mappings are shown in CEC_ADDR_L (HDMI_CEC_ADDR_L) and CEC_ADDR_H (HDMI_CEC_ADDR_H) . This register is written by software when the logical allocation is finished. Bit value 1 means the corresponding logical address is allocated to this device. Bit value 0 means the corresponding logical address is not allocated to this device.

- Address Offset: 0x7D05
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 7D05h offset = 12_7D05h

Bit	7	6	5	4
Read Write	CEC_ADDR_L7	CEC_ADDR_L6	CEC_ADDR_L5	CEC_ADDR_L4
Reset	0	0	0	0

Bit	3	2	1	0
Read Write	CEC_ADDR_L3	CEC_ADDR_L2	CEC_ADDR_L1	CEC_ADDR_L0
Reset	0	0	0	0

HDMI_CEC_ADDR_L field descriptions

Field	Description
7 CEC_ADDR_L7	Logical address 7 - Tuner 3

Table continues on the next page...

HDMI_CEC_ADDR_L field descriptions (continued)

Field	Description
6 CEC_ADDR_L6	Logical address 6 - Tuner 2
5 CEC_ADDR_L5	Logical address 5 - Audio System
4 CEC_ADDR_L4	Logical address 4 - Playback Device 1
3 CEC_ADDR_L3	Logical address 3 - Tuner 1
2 CEC_ADDR_L2	Logical address 2 - Recording Device 2
1 CEC_ADDR_L1	Logical address 1 - Recording Device 1
0 CEC_ADDR_L0	Logical address 0 - Device TV

34.5.297 CEC_ADDR_H (HDMI_CEC_ADDR_H)

CEC_ADDR_L and CEC_ADDR_H registers indicate the logical address(es) allocated to the CEC device. The logical address mappings are shown in CEC_ADDR_L (HDMI_CEC_ADDR_L) and CEC_ADDR_H (HDMI_CEC_ADDR_H) . This register is written by software when the logical allocation is finished. Bit value 1 means the corresponding logical address is allocated to this device. Bit value 0 means the corresponding logical address is not allocated to this device.

- Address Offset: 0x7D06
- Size: 8 bits
- Value after Reset: 0x80
- Access: Read/Write

Address: 12_0000h base + 7D06h offset = 12_7D06h

Bit	7	6	5	4
Read Write	CEC_ADDR_H7	CEC_ADDR_H6	CEC_ADDR_H5	CEC_ADDR_H4
Reset	1	0	0	0

Bit	3	2	1	0
Read Write	CEC_ADDR_H3	CCEC_ADDR_H2	CEC_ADDR_H1	CEC_ADDR_H0
Reset	0	0	0	0

HDMI_CEC_ADDR_H field descriptions

Field	Description
7 CEC_ADDR_H7	Logical address 15 - Unregistered (as initiator address), Broadcast (as destination address)
6 CEC_ADDR_H6	Logical address 14 - Free use
5 CEC_ADDR_H5	Logical address 13 - Reserved
4 CEC_ADDR_H4	Logical address 12 - Reserved
3 CEC_ADDR_H3	Logical address 11 - Playback Device 3
2 CCEC_ADDR_H2	Logical address 10 - Tuner 4
1 CEC_ADDR_H1	Logical address 9 - Playback Device 3
0 CEC_ADDR_H0	Logical address 8 - Playback Device 2

34.5.298 CEC_TX_CNT (HDMI_CEC_TX_CNT)

This register indicates the size of the frame in bytes (including header and data blocks), which are available in the transmitter data buffer.

When the value is zero, the CEC controller ignores the send command triggered by software. When the transmission is done (no matter success or not), the current value is held until it is overwritten by software.

- Address Offset: 0x7D07
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 7D07h offset = 12_7D07h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				CEC_TX_CNT			
Reset	0	0	0	0	0	0	0	0

HDMI_CEC_TX_CNT field descriptions

Field	Description
7–5 -	This field is reserved. Reserved

Table continues on the next page...

HDMI_CEC_TX_CNT field descriptions (continued)

Field	Description
CEC_TX_CNT	CEC Transmitter Counter register: Value after Reset: 5'b00000 0 No data needs to be transmitted. 1 Frame size is 1 byte. 16 Frame size is 16 byte.

34.5.299 CEC_RX_CNT (HDMI_CEC_RX_CNT)

This register indicates the size of the frame in bytes (including header and data blocks), which are available in the receiver data buffer.

Only after the whole receiving process is finished successfully, the counter is refreshed to the value which indicates the total number of data bytes in the Receiver Data Register.

- Address Offset: 0x7d08
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 7D08h offset = 12_7D08h

Bit	7	6	5	4	3	2	1	0
Read	-			CEC_RX_CNT				
Write	-							
Reset	0	0	0	0	0	0	0	0

HDMI_CEC_RX_CNT field descriptions

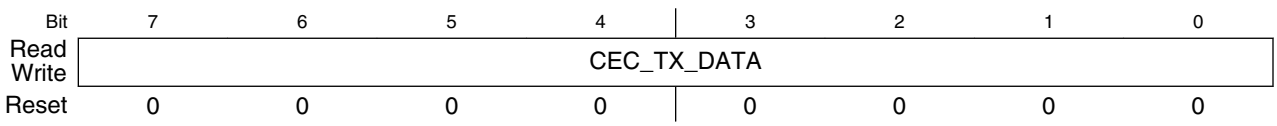
Field	Description
7–5 -	
CEC_RX_CNT	CEC Receiver Counter register. Value after Reset: 5‘b00000 0 No data received 1 1-byte data is received. 16 16-byte data is received.

34.5.300 CEC_TX_DATA (HDMI_CEC_TX_DATA*n*)

These registers (8 bit each) are the buffers used for storing the data waiting for transmission(including header and data blocks).

- Address Offset: 0x7D10 .. 0x7D1F
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 7D10h offset + (1d × i), where i=0d to 15d



HDMI_CEC_TX_DATA*n* field descriptions

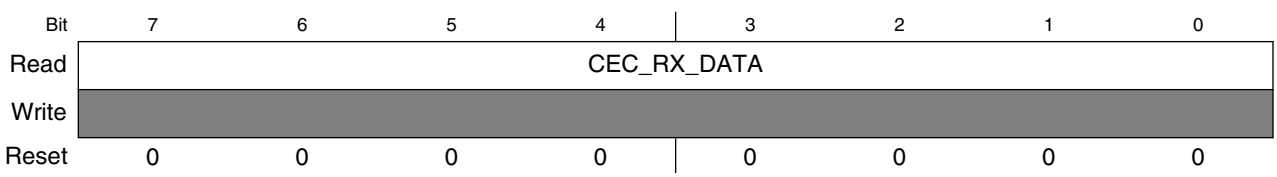
Field	Description
CEC_TX_DATA	Header block in CEC_TX_DATA0 Data blockn in CEC_TX_DATAn

34.5.301 CEC_RX_DATA (HDMI_CEC_RX_DATA*n*)

These registers (8 bit each) are the buffers used for storing the received data (including header and data blocks).

- Address Offset: 0x7D20 .. 0x7D2F
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 7D20h offset + (1d × i), where i=0d to 15d



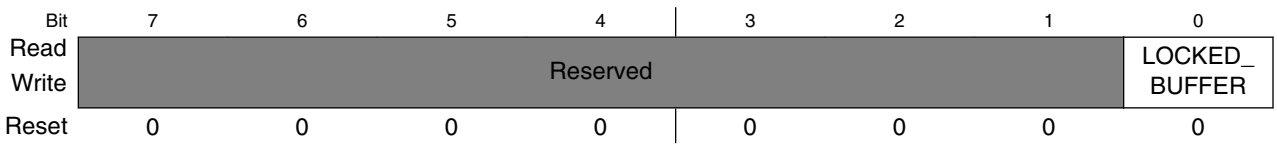
HDMI_CEC_RX_DATA*n* field descriptions

Field	Description
CEC_RX_DATA	Header block in CEC_RX_DATA0 Data blockn in CEC_RX_DATAn

34.5.302 CEC_LOCK (HDMI_CEC_LOCK)

- Address Offset: 0x7D30
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 7D30h offset = 12_7D30h



HDMI_CEC_LOCK field descriptions

Field	Description
7–1 -	This field is reserved. Reserved
0 LOCKED_ BUFFER	When a frame is received, this bit would be active. The CEC controller answers to all the messages with NACK until the CPU writes it to '0'.

34.5.303 CEC_WKUPCTRL (HDMI_CEC_WKUPCTRL)

- Address Offset: 0x7D31
- Size: 8 bits
- Value after Reset: 0xFF
- Access: Read/Write

After receiving a message in the CEC_RX_DATA1 (OPCODE) registers, the CEC engine verifies the message opcode[7:0] against one of the previously defined values to generate the wake-up status:

Wakeupstatus is 1 when:

received opcode is 0x04 and opcode0x04en is 1 or received opcode is 0x0D and opcode0x0Den is 1 or received opcode is 0x41 and opcode0x41en is 1 or received opcode is 0x42 and opcode0x42en is 1 or received opcode is 0x44 and opcode0x44en is 1 or received opcode is 0x70 and opcode0x70en is 1 or received opcode is 0x82 and opcode0x82en is 1 or received opcode is 0x86 and opcode0x86en is 1

Wakeupstatus is 0 when none of the previous conditions are true.

This formula means that the wake-up status (on CEC_STAT[6] register) is only '1' if the opcode[7:0] received is equal to one of the defined values and the corresponding enable bit of that defined value is set to '1'.

Address: 12_0000h base + 7D31h offset = 12_7D31h

Bit	7	6	5	4
Read Write	OPCODE0x86en	OPCODE0x82en	OPCODE0x70en	OPCODE0x44en
Reset	1	1	1	1

Bit	3	2	1	0
Read Write	OPCODE0x42en	OPCODE0x41en	OPCODE0x0Den	OPCODE0x04en
Reset	1	1	1	1

HDMI_CEC_WKUPCTRL field descriptions

Field	Description
7 OPCODE0x86en	OPCODE 0x86 wake up enable
6 OPCODE0x82en	OPCODE 0x82 wake up enable
5 OPCODE0x70en	OPCODE 0x70 wake up enable
4 OPCODE0x44en	OPCODE 0x44 wake up enable
3 OPCODE0x42en	OPCODE 0x42 wake up enable
2 OPCODE0x41en	OPCODE 0x41 wake up enable
1 OPCODE0x0Den	OPCODE 0x0D wake up enable
0 OPCODE0x04en	OPCODE 0x04 wake up enable

34.5.304 I2CM_SLAVE (HDMI_I2CM_SLAVE)

I2C Master Registers (E-DDC) registers are responsible for the Master's coordination with the Slave, by coordinating the Slave address, data identification, transaction status, acknowledgement, and reset functions.

- Address Offset: 0x7E00
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 7E00h offset = 12_7E00h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved	slaveaddr[6:0]						
Reset	0	0	0	0	0	0	0	0

HDMI_I2CM_SLAVE field descriptions

Field	Description
7 -	This field is reserved. Reserved
slaveaddr[6:0]	Slave address to be sent during read and write normal operations.

34.5.305 I2CM_ADDRESS (HDMI_I2CM_ADDRESS)

- Address Offset: 0x7E01
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 7E01h offset = 12_7E01h

Bit	7	6	5	4	3	2	1	0
Read Write	address[7:0]							
Reset	0	0	0	0	0	0	0	0

HDMI_I2CM_ADDRESS field descriptions

Field	Description
address[7:0]	Register address for read and write operations.

34.5.306 I2CM_DATAO (HDMI_I2CM_DATAO)

- Address Offset: 0x7E02
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 7E02h offset = 12_7E02h

Bit	7	6	5	4	3	2	1	0
Read Write	datao[7:0]							
Reset	0	0	0	0	0	0	0	0

HDMI_I2CM_DATAO field descriptions

Field	Description
datao[7:0]	Data to be written on register pointed by address[7:0].

34.5.307 I2CM_DATAI (HDMI_I2CM_DATAI)

- Address Offset: 0x7E03
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read

Address: 12_0000h base + 7E03h offset = 12_7E03h

Bit	7	6	5	4	3	2	1	0
Read	datai[7:0]							
Write								
Reset	0	0	0	0	0	0	0	0

HDMI_I2CM_DATAI field descriptions

Field	Description
datai[7:0]	Data read from register pointed by address[7:0].

34.5.308 I2CM_OPERATION (HDMI_I2CM_OPERATION)

Read and write operation request. This register can only be written, reading this register will always result in 00h. Writing 1'b1 simultaneously to rd, rd_ext and wr requests is considered as a read (rd) request.

- Address Offset: 0x7E04
- Size: 8 bits
- Value after Reset: 0x00
- Access: Write

Address: 12_0000h base + 7E04h offset = 12_7E04h

Bit	7	6	5	4	3	2	1	0
Read	Reserved				Reserved			
Write				wr			rd_ext	rd
Reset	0	0	0	0	0	0	0	0

HDMI_I2CM_OPERATION field descriptions

Field	Description
7–5 -	This field is reserved. Reserved
4 wr	Write operation request.
3–2 -	This field is reserved. Reserved
1 rd_ext	After writing 1'b1 to rd_ext bit a extended data read operation is started (E- DDC read operation).
0 rd	Read operation request.

34.5.309 I2CM_INT (HDMI_I2CM_INT)

This register contains and configures I2C master done interrupt.

- Address Offset: 0x7E05
- Size: 8 bits
- Value after Reset: 0x08
- Access: Read/Write

Address: 12_0000h base + 7E05h offset = 12_7E05h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				done_pol	done_mask	done_ interrupt	done_status
Reset	0	0	0	0	1	0	0	0

HDMI_I2CM_INT field descriptions

Field	Description
7–4 -	This field is reserved. Reserved
3 done_pol	Done interrupt polarity configuration.
2 done_mask	Done interrupt mask signal.
1 done_interrupt	Operation done interrupt bit. Only lasts for 1 SFR clock cycle and is auto cleaned after it. {done_interrupt = (done_mask==0b) && (done_status==done_pol)}.
0 done_status	Operation done status bit. Marks the end of a rd or write operation.

34.5.310 I2CM_CTLINT (HDMI_I2CM_CTLINT)

This register contains and configures I2C master arbitration error and not acknowledge error interrupt.

- Address Offset: 0x7E06
- Size: 8 bits
- Value after Reset: 0x88
- Access: Read/Write

Address: 12_0000h base + 7E06h offset = 12_7E06h

Bit	7	6	5	4
Read Write	nack_pol	nack_mask	nack_interrupt	nack_status
Reset	1	0	0	0

Bit	3	2	1	0
Read Write	arbitration_pol	arbitration_mask	arbitration_interrupt	arbitration_status
Reset	1	0	0	0

HDMI_I2CM_CTLINT field descriptions

Field	Description
7 nack_pol	Not acknowledge error interrupt polarity configuration.
6 nack_mask	Not acknowledge error interrupt mask signal.
5 nack_interrupt	Not acknowledge error interrupt bit. Only lasts for 1 SFR clock cycle and is auto cleaned after it. {nack_interrupt = (nack_mask==0b) && (nack_status==nack_pol)}.
4 nack_status	Not acknowledge error status bit. Error on I2C not acknowledge.
3 arbitration_pol	Arbitration error interrupt polarity configuration.
2 arbitration_mask	Arbitration error interrupt mask signal.
1 arbitration_ interrupt	Arbitration error interrupt bit. Only lasts for 1 SFR clock cycle and is auto cleaned after it. {arbitration_interrupt = (arbitration_mask==0b) && (arbitration_status==arbitration_pol)}.
0 arbitration_status	Arbitration error status bit. Error on master I2C protocol arbitration.

34.5.311 I2CM_DIV (HDMI_I2CM_DIV)

This register configures the division relation between master and scl clock.

- Address Offset: 0x7E07
- Size: 8 bits
- Value after Reset: 0x0B
- Access: Read/Write

Address: 12_0000h base + 7E07h offset = 12_7E07h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				fast_std_ mode	Reserved		
Reset	0	0	0	0	1	0	1	1

HDMI_I2CM_DIV field descriptions

Field	Description
7–4 -	This field is reserved. Reserved
3 fast_std_mode	Sets the I2C Master to work in Fast Mode or Standard Mode: 1 Fast Mode 0 Standard Mode
-	This field is reserved. Reserved

34.5.312 I2CM_SEGADDR (HDMI_I2CM_SEGADDR)

This register configures the segment address for extended RD/WR destination.

- Address Offset: 0x7E08
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 7E08h offset = 12_7E08h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved	SEGADDR						
Reset	0	0	0	0	0	0	0	0

HDMI_I2CM_SEGADDR field descriptions

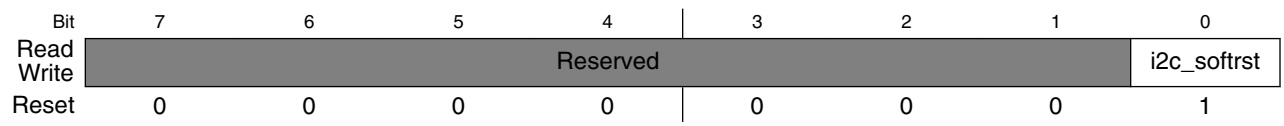
Field	Description
7 -	This field is reserved. Reserved
SEGADDR	E-DDC Extended read segment address

34.5.313 I2CM_SOFTRSTZ (HDMI_I2CM_SOFTRSTZ)

This register resets the I2C master.

- Address Offset: 0x7E09
- Size: 8 bits
- Value after Reset: 0x01
- Access: Read/Write

Address: 12_0000h base + 7E09h offset = 12_7E09h



HDMI_I2CM_SOFTRSTZ field descriptions

Field	Description
7–1 -	This field is reserved. Reserved
0 i2c_softrst	I2C Master Software Reset. Active by writing a zero and auto cleared to one in the following cycle. Value after Reset: 1b

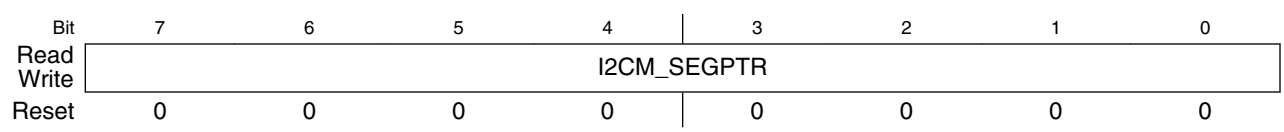
34.5.314 I2CM_SEGPTR (HDMI_I2CM_SEGPTR)

This register configures the segment pointer for extended RD/WR request.

- Address Offset: 0x7E0A
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

The following *CNT registers must be set before any I2C bus transaction can take place to ensure proper I/O timing.

Address: 12_0000h base + 7E0Ah offset = 12_7E0Ah



HDMI_I2CM_SEGPTR field descriptions

Field	Description
I2CM_SEGPTR	I2CM_SEGPTR is used for EDID reading operations, particularly for the Extended Data Read Operation (See I²C Master Interface Extended Read Mode ") which is used for Enhanced DDC. This is all described in the VESA Enhanced Display Data Channel Standard v1.1 spec. (addresses A0h/A1h pairs and a segment pointer - 60h).

34.5.315 I2CM_SS_SCL_HCNT_1_ADDR (HDMI_I2CM_SS_SCL_HCNT_1_ADDR)

- Address Offset: 0x7E0B
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 7E0Bh offset = 12_7E0Bh

Bit	7	6	5	4	3	2	1	0
Read / Write	i2cmp_ss_scl_hcnt[15:8]							
Reset	0	0	0	0	0	0	0	0

HDMI_I2CM_SS_SCL_HCNT_1_ADDR field descriptions

Field	Description
i2cmp_ss_scl_hcnt[15:8]	Value after Reset: 8'h00

34.5.316 I2CM_SS_SCL_HCNT_0_ADDR (HDMI_I2CM_SS_SCL_HCNT_0_ADDR)

- Address Offset: 0x7E0C
- Size: 8 bits
- Value after Reset: 0x6C
- Access: Read/Write

Address: 12_0000h base + 7E0Ch offset = 12_7E0Ch

Bit	7	6	5	4	3	2	1	0
Read / Write	i2cmp_ss_scl_hcnt[7:0]							
Reset	0	1	1	0	1	1	0	0

HDMI_I2CM_SS_SCL_HCNT_0_ADDR field descriptions

Field	Description
i2cmp_ss_scl_hcnt[7:0]	Value after Reset: 8'h6C

34.5.317 I2CM_SS_SCL_LCNT_1_ADDR (HDMI_I2CM_SS_SCL_LCNT_1_ADDR)

- Address Offset: 0x7E0D
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 7E0Dh offset = 12_7E0Dh

Bit	7	6	5	4	3	2	1	0
Read Write	i2cmp_ss_scl_lcnt[15:8]							
Reset	0	0	0	0	0	0	0	0

HDMI_I2CM_SS_SCL_LCNT_1_ADDR field descriptions

Field	Description
i2cmp_ss_scl_lcnt[15:8]	Value after Reset: 8'h00

34.5.318 I2CM_SS_SCL_LCNT_0_ADDR (HDMI_I2CM_SS_SCL_LCNT_0_ADDR)

- Address Offset: 0x7E0E
- Size: 8 bits
- Value after Reset: 0x7F
- Access: Read/Write

Address: 12_0000h base + 7E0Eh offset = 12_7E0Eh

Bit	7	6	5	4	3	2	1	0
Read Write	i2cmp_ss_scl_lcnt[7:0]							
Reset	0	1	1	1	1	1	1	1

HDMI_I2CM_SS_SCL_LCNT_0_ADDR field descriptions

Field	Description
i2cmp_ss_scl_lcnt[7:0]	Value after Reset: 8'h7F

34.5.319 I2CM_FS_SCL_HCNT_1_ADDR (HDMI_I2CM_FS_SCL_HCNT_1_ADDR)

- Address Offset: 0x7E0F
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 7E0Fh offset = 12_7E0Fh

Bit	7	6	5	4	3	2	1	0
Read / Write	i2cmp_fs_scl_hcnt[15:8]							
Reset	0	0	0	0	0	0	0	0

HDMI_I2CM_FS_SCL_HCNT_1_ADDR field descriptions

Field	Description
i2cmp_fs_scl_hcnt[15:8]	Value after Reset: 8'h00

34.5.320 I2CM_FS_SCL_HCNT_0_ADDR (HDMI_I2CM_FS_SCL_HCNT_0_ADDR)

- Address Offset: 0x7E10
- Size: 8 bits
- Value after Reset: 0x11
- Access: Read/Write

Address: 12_0000h base + 7E10h offset = 12_7E10h

Bit	7	6	5	4	3	2	1	0
Read / Write	i2cmp_fs_scl_hcnt[7:0]							
Reset	0	0	0	1	0	0	0	1

HDMI_I2CM_FS_SCL_HCNT_0_ADDR field descriptions

Field	Description
i2cmp_fs_scl_hcnt[7:0]	Value after Reset: 8'h11

34.5.321 I2CM_FS_SCL_LCNT_1_ADDR (HDMI_I2CM_FS_SCL_LCNT_1_ADDR)

- Address Offset: 0x7E11
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 7E11h offset = 12_7E11h

Bit	7	6	5	4	3	2	1	0
Read Write	i2cmp_fs_scl_lcnt[15:8]							
Reset	0	0	0	0	0	0	0	0

HDMI_I2CM_FS_SCL_LCNT_1_ADDR field descriptions

Field	Description
i2cmp_fs_scl_lcnt[15:8]	Value after Reset: 8'h00

34.5.322 I2CM_FS_SCL_LCNT_0_ADDR (HDMI_I2CM_FS_SCL_LCNT_0_ADDR)

- Address Offset: 0x7E12
- Size: 8 bits
- Value after Reset: 0x24
- Access: Read/Write

Address: 12_0000h base + 7E12h offset = 12_7E12h

Bit	7	6	5	4	3	2	1	0
Read Write	i2cmp_fs_scl_lcnt[7:0]							
Reset	0	0	1	0	0	1	0	0

HDMI_I2CM_FS_SCL_LCNT_0_ADDR field descriptions

Field	Description
i2cmp_fs_scl_ lcnt[7:0]	Value after Reset: 8'h24

34.5.323 BASE_POINTER_ADDR (HDMI_BASE_POINTER_ADDR)

The I2C Slave Registers allow register memory pagination, and function in the incremental burst operation mode that increases the data throughput when consecutive addressed registers need to be read or write.

The I2C base pointer operation mode is a aimed to allow register memory pagination. As long as this operational mode is enabled the value written to this register will be used as the seven most significant bits of the internal Special Function Register address interface (sfraddr[14:8]) for all read or write operations. I2C data transfer protocol used shall be the 7-bit addressed as defined in the section 9 of the I2C-bus Specification, version 2.1.

- Address Offset: 0x7F00
- Size: 8 bits
- Value after Reset: 0x00
- Access: Read/Write

Address: 12_0000h base + 7F00h offset = 12_7F00h

Bit	7	6	5	4	3	2	1	0
Read Write	en_base_ pointer_addr	base_pointer_base_addr[6:0]						
Reset	0	0	0	0	0	0	0	0

HDMI_BASE_POINTER_ADDR field descriptions

Field	Description
7 en_base_ pointer_addr	Enables the base pointer operation mode.
base_pointer_ base_addr[6:0]	Defines the base address for base pointer operation mode. They represent the address bits [14:8]

Chapter 35
HDMI 3D Tx PHY (HDMI_PHY)

35.1 Overview

35.1.1 General Description

The HDMI 3D Tx PHY (HDMI_PHY) is the physical layer of a single-link HDMI transmitter interface and comprises three line drivers for data transmission and an additional line driver for clock transmission. The HDMI_PHY is designed to perform the serialization and transmission of video data and control information through an HDMI interface. It interfaces with an HDMI Tx link controller through a common graphic controller interface supporting 30- or 60-bit data transfers. A clock line driver is used for reference clock transmission.

35.1.2 Applications

The HDMI_PHY is targeted to digital video/audio transmission for high resolution display applications, supporting major display formats up to 1080 i/p in DTV applications and QXGA in graphic display applications, with true-color or deep-color resolutions. The HDMI_PHY supports 3D video formats. At maximum pixel rate, the HDMI channel bit rate is 3.4 Gbps, providing a maximum throughput of 9.2 Gbps.

The HDMI_PHY is an ideal solution for implementing the physical layer of a high-reliability digital video/audio interface based on HDMI technology, with reduced line count and minimum cable-wire and EMI shielding requirements. The HDMI_PHY delivers the bandwidth needed in every time mode that demands high resolution panels, while keeping clock sources at a low frequency.

The figure below shows a typical application for HDMI_PHY.

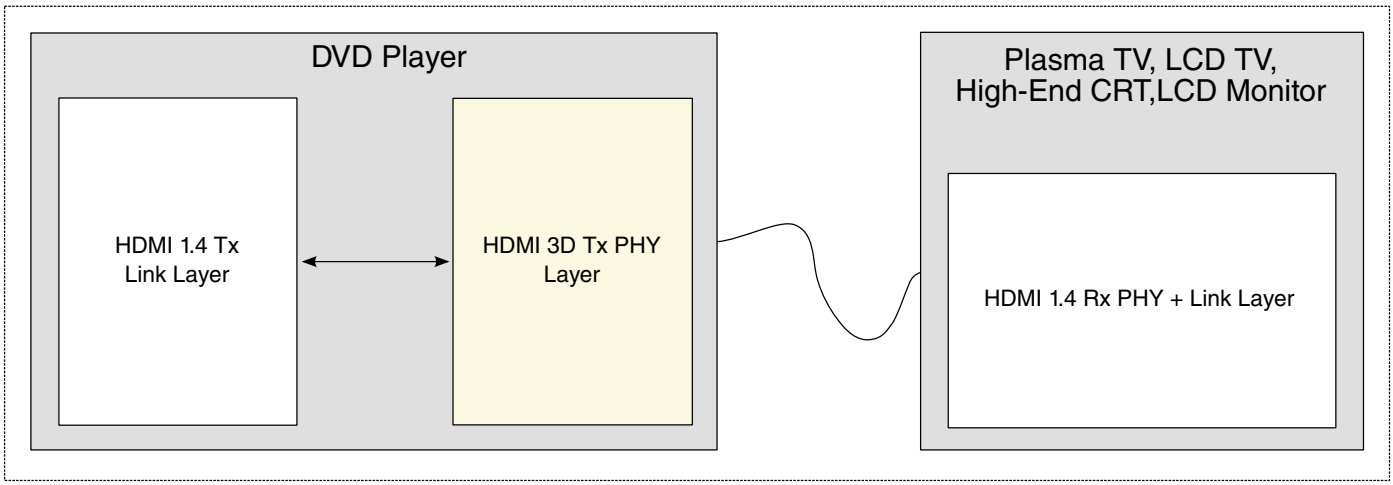


Figure 35-1. Typical Application for HDMI 3D Tx PHY

35.1.3 Standards Compliance

The HDMI_PHY is fully compliant with all the required specifications of the following standards:

- *High-Definition Multimedia Interface Specification*, Version 1.4a
- *Digital Visual Interface*, Revision 1.0
- *HDMI Compliance Test Specification*, Version 1.4a

35.1.4 Features

The HDMI_PHY provides the following features:

- Support for up to 720p at 100 Hz and 720i at 200 Hz or 1080p at 60 Hz and 1080i/720i at 120 Hz HDTV display resolutions and up to QXGA graphic display resolutions
- Support for 3D video formats
- Link controller flexible interface with 30- or 60-bit SDR data access
- Up to 9.2 Gbps aggregate bandwidth
- Driver with pre-emphasis and edge rate control for extra-long cable support
- Programmable source terminations
- HPD input analog comparator
- Rx sensing
- 13.5-270 MHz input reference clock

- 50% duty-cycle output clock
- Embedded Tx scope
- Embedded A/D converter and analog testbus for ATE testing
- Built-in pattern generator
- Small core area
- Low power consumption

35.1.5 HDMI 3D Tx PHY System-Level Overview

This section provides a system-level overview of the HDMI 3D Tx PHY.

35.1.5.1 System-Level Block Diagram

The figure below shows a system-level block diagram of the HDMI Tx System.

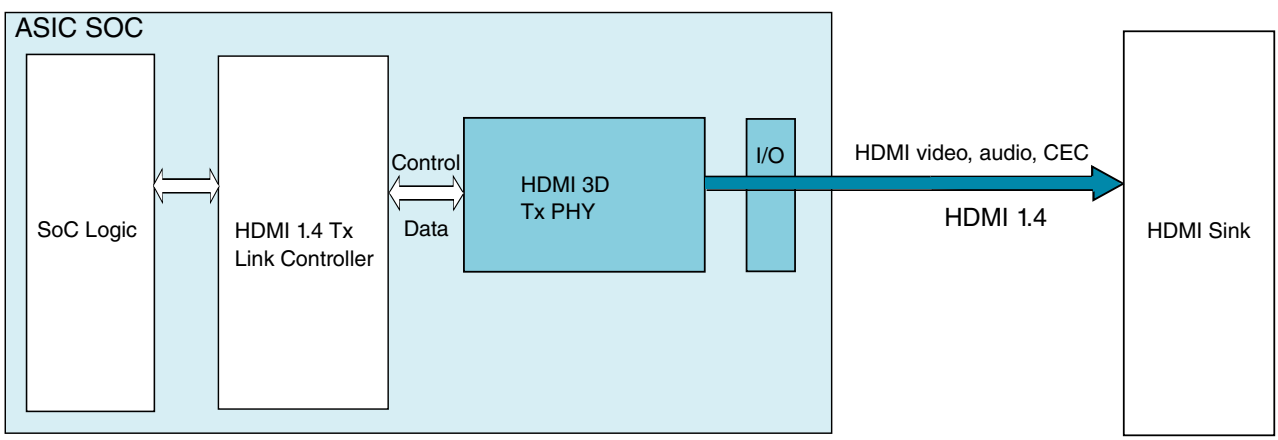


Figure 35-2. HDMI Tx System-Level Block Diagram

The HDMI_PHY macro interfaces with the HDMI Tx link controller, which provides control signals and video data to be transmitted through the TMDS lanes. Analog I/O's interface with the external world.

35.1.5.2 HDMI 3D Tx PHY

The HDMI 3D Tx PHY is the physical layer of an HDMI/DVI-compliant digital transmitter (source).

For a functional block diagram and a functional description of the HDMI 3D Tx PHY, see Functional Description.

35.1.5.2.1 Interfaces

The HDMI interface on the HDMI_PHY block is defined by the TMDS differential lanes and the HPD/DDCCEC connections. HDMI video data is provided through three differential TMDS pairs for data and one TMDS differential pair for clock. HPD/DDC_CEC enables detection of the HDMI sink. These signals interface the external world and connect to the HDMI connector.

On the system interface, the HDMI_PHY connects to the HDMI Tx link controller. This block provides an input reference clock to the PHY (pixel clock). Based on the defined video mode, a pixel repetition clock and a TMDS word clock are output to the controller. The controller then generates video data on this TMDS word clock and outputs both data and clock to the HDMI_PHY. Data is then serialized and sent to the HDMI interface.

In addition, the system side includes one standard I^2C interface for configuration and testability of the PHY. Other signals correspond to control logic and are either input to the PHY or provided by the PHY to the controller block for observability.

35.1.5.3 HDMI 1.4 I/O Pads

HDMI 3D Tx PHY provides a set of analog ESD pads for use.

35.1.5.3.1 I/O Pads Description

HDMI 3D Tx PHY I/O pads are provided as follows:

- Pad order: VPH, VP, GD, VPH, VP, GD, VPH, VP, GD, HDMI_TX_CLK_P, HDMI_TX_CLK_N, GD, HDMI_TX_DATA0_N, HDMI_TX_DATA0_P, GD, HDMI_TX_DATA1_N, HDMI_TX_DATA1_P, GD, HDMI_TX_DATA2_N, HDMI_TX_DATA2_P, GD, VP, VPH, REXT, HDMI_TX_HPD, HDMI_TX_DDC_CEC

As shown in the figure below, for maximum performance, HDMI 3D Tx PHY I/O pads contain decoupling capacitance between VP and GD and a filter between VP and VP_FILT (one for each VP_FILT signal).

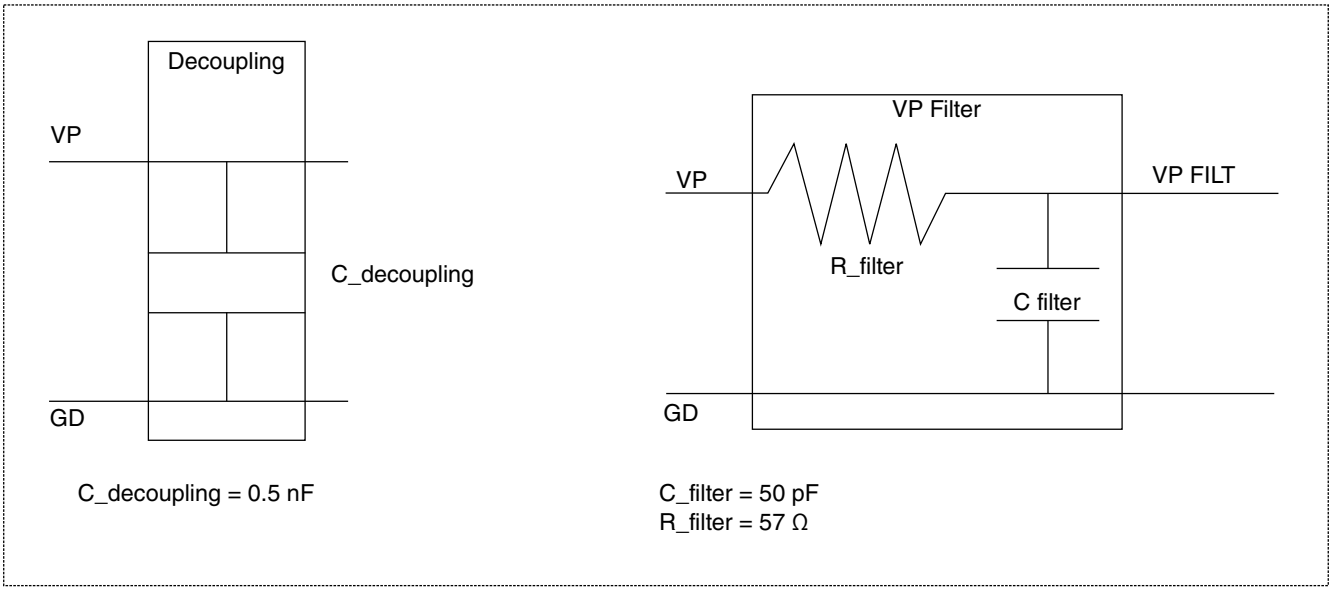


Figure 35-3. I/O Pads: Decoupling Capacitance Between VP and GD, Filter Between VP and VP_FILT

35.2 External Signals

The table found here describes the external signals of HDMI.

Table 35-1. HDMI External Signals

Signal	Description	Pad	Mode	Direction
HDMI_TX_CEC_LINE (CEC_LINE)	CEC line between source and sink	EIM_A25	ALT6	I/O
		KEY_ROW2	ALT6	
HDMI_TX_CLK_N (CLK_N)	Negative Clock Signal	HDMI_CLKM	No Muxing	I
HDMI_TX_CLK_P (CLK_P)	Positive Clock Signal	HDMI_CLKP	No Muxing	I
HDMI_TX_DATA0_N (DATA0_N)	Negative Data Signal 0	HDMI_D0M	No Muxing	I/O
HDMI_TX_DATA0_P (DATA0_P)	Positive Data Signal 0	HDMI_D0P	No Muxing	I/O
HDMI_TX_DATA1_N (DATA1_N)	Negative Data Signal 1	HDMI_D1M	No Muxing	I/O
HDMI_TX_DATA1_P (DATA1_P)	Positive Data Signal 1	HDMI_D1P	No Muxing	I/O
HDMI_TX_DATA2_N (DATA2_N)	Negative Data Signal 2	HDMI_D2M	No Muxing	I/O

Table continues on the next page...

Table 35-1. HDMI External Signals (continued)

Signal	Description	Pad	Mode	Direction
HDMI_TX_DATA2_P (DATA0_P)	Positive Data Signal 2	HDMI_D2P	No Muxing	I/O
HDMI_TX_DDC_CEC (DDC_CEC)	CEC Signal	HDMI_DDCCEC	No Muxing	I/O
HDMI_TX_DDC_SCL (DDC_SCL)	SCL Signal	EIM_EB2	ALT4	I/O
		KEY_COL3	ALT2	
HDMI_TX_DDC_SDA (DDC_SDA)	SDA Signal	EIM_D16	ALT4	I/O
		KEY_ROW3	ALT2	
HDMI_TX_HPD (HPD)	HPD Signal	HDMI_HPD	No Muxing	I/O

35.2.1 Top-Level I/O Diagram

The figure below shows the HDMI 3D Tx PHY top-level signals.

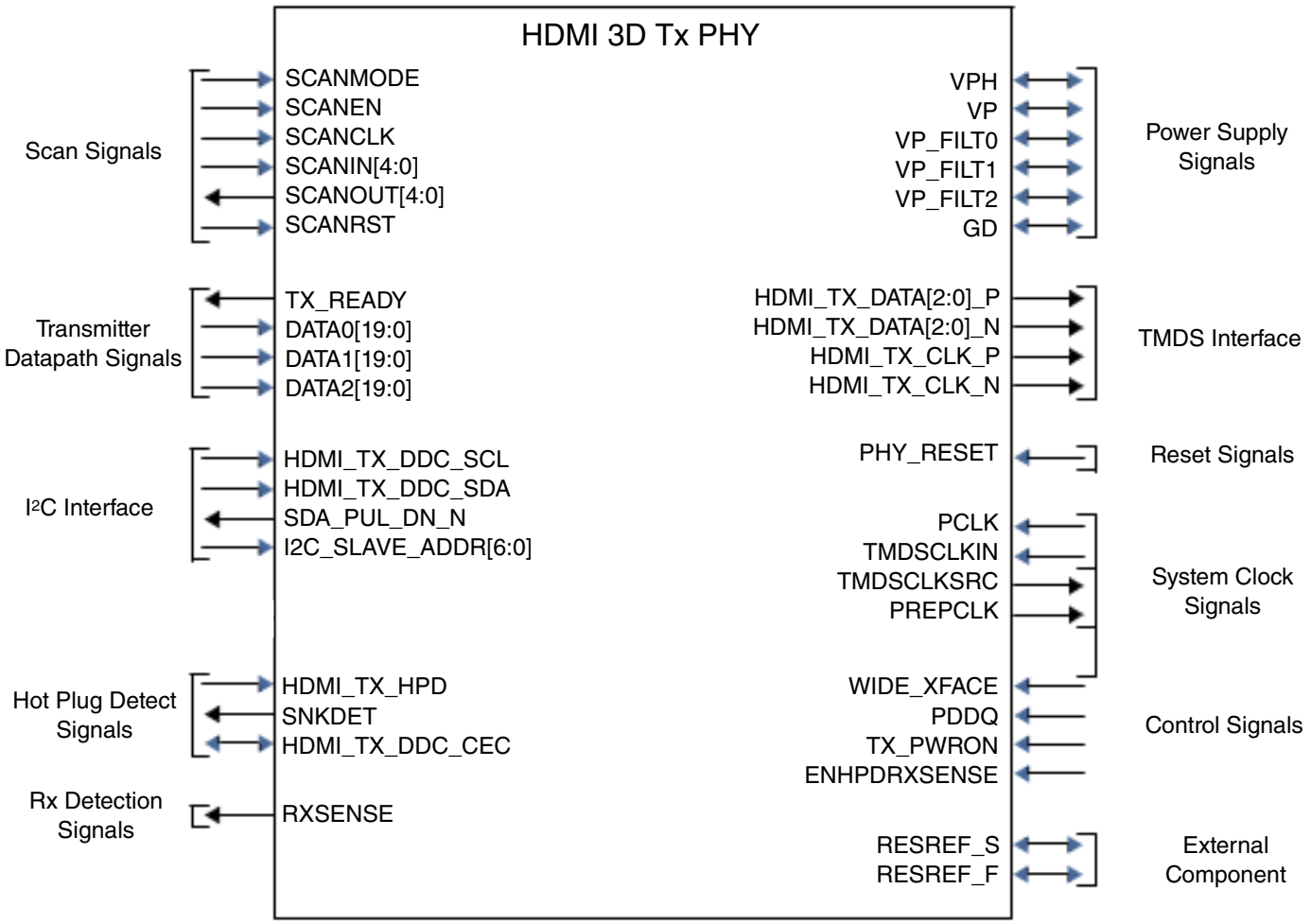


Figure 35-4. Top-Level I/O Diagram

35.2.2 Top -Level Signal Descriptions

This section describes the top-level signals.

In addition to describing the function of each signal, the signal descriptions include the following information:

- **Voltage range**: Describes the voltage range expected on this pin.
- **Synchronous**: Indicates that the signal is asserted or deasserted with respect to a clock edge.
- **Asynchronous**: Indicates that the signal is not asserted or deasserted with respect to a clock edge.

35.2.2.1 TMDS Interface

The table below describes the Transmission Minimized Differential Signaling (TMDS) interface signals.

Table 35-2. TMDS Interface

Signal	I/O	Description
HDMI_TX_DATA[2:0]_P	O	**Function**: Positive TMDS differential line driver data output for data channels 0,1, and 2. **Voltage range**: For the permitted voltage range, refer to *High-Definition Interface Specification*, Version 1.4, "Electrical Specification" section. **Active state**: N/A **Synchronous to**: N/A
HDMI_TX_DATA[2:0]_N	O	**Function**: Negative TMDS differential line driver data output for data channels 0,1, and 2. **Voltage range**: For the permitted voltage range, refer to *High-Definition Interface Specification*, Version 1.4, "Electrical Specification" section. **Active state**: N/A **Synchronous to**: N/A
HDMI_TX_CLK_P	O	**Function**: Positive TMDS differential line driver clock output **Voltage range**: For the permitted voltage range, refer to *High-Definition Multimedia Interface Specification*, Version 1.4, "Electrical Specification" section. **Active state**: N/A **Synchronous to**: N/A
HDMI_TX_CLK_N	O	**Function**: Negative TMDS differential line driver clock output **Voltagerange**: For the permitted voltage range, refer to *High-DefinitionMultimediaInterfaceSpecification*, Version 1.4, "Electrical Specification" section. **Activestate**: N/A **Synchronousto**: N/A

35.2.2.2 Reset Signals

The table blow describes the reset signals.

Table 35-3. Reset Signals

Signal	I/O	Description
PHY_RESET	I	**Function**: PHY reset. This signal places the digital section of the macro into a reset state. **Voltagerange**: 0-VP

Table 35-3. Reset Signals

Signal	I/O	Description
		Activestate: High **Synchronousto**: Asynchronous

35.2.2.3 External Component

The table below describes the top-level signals that connect to an external component.

Table 35-4. External Component

Signal	I/O	Description
RESREF_S	I/O	**Function**: 1600- precision resistor to ground. During resistor tuning, current is forced through the external resistor using RESREF_F, while the induced voltage is sensed on RESREF_S. **Voltage range**: N/A **Active state**: N/A **Synchronous to**: Asynchronous
RESREF_F	I/O	**Function**: Reference resistor connection **Voltage range**: N/A **Active state**: N/A **Synchronous to**: Asynchronous

35.3 Functional Description

This section describes the functional architecture of the HDMI 3D Tx PHY.

35.3.1 Functional Overview

The figure found in this section shows a functional block diagram of the HDMI_PHY.

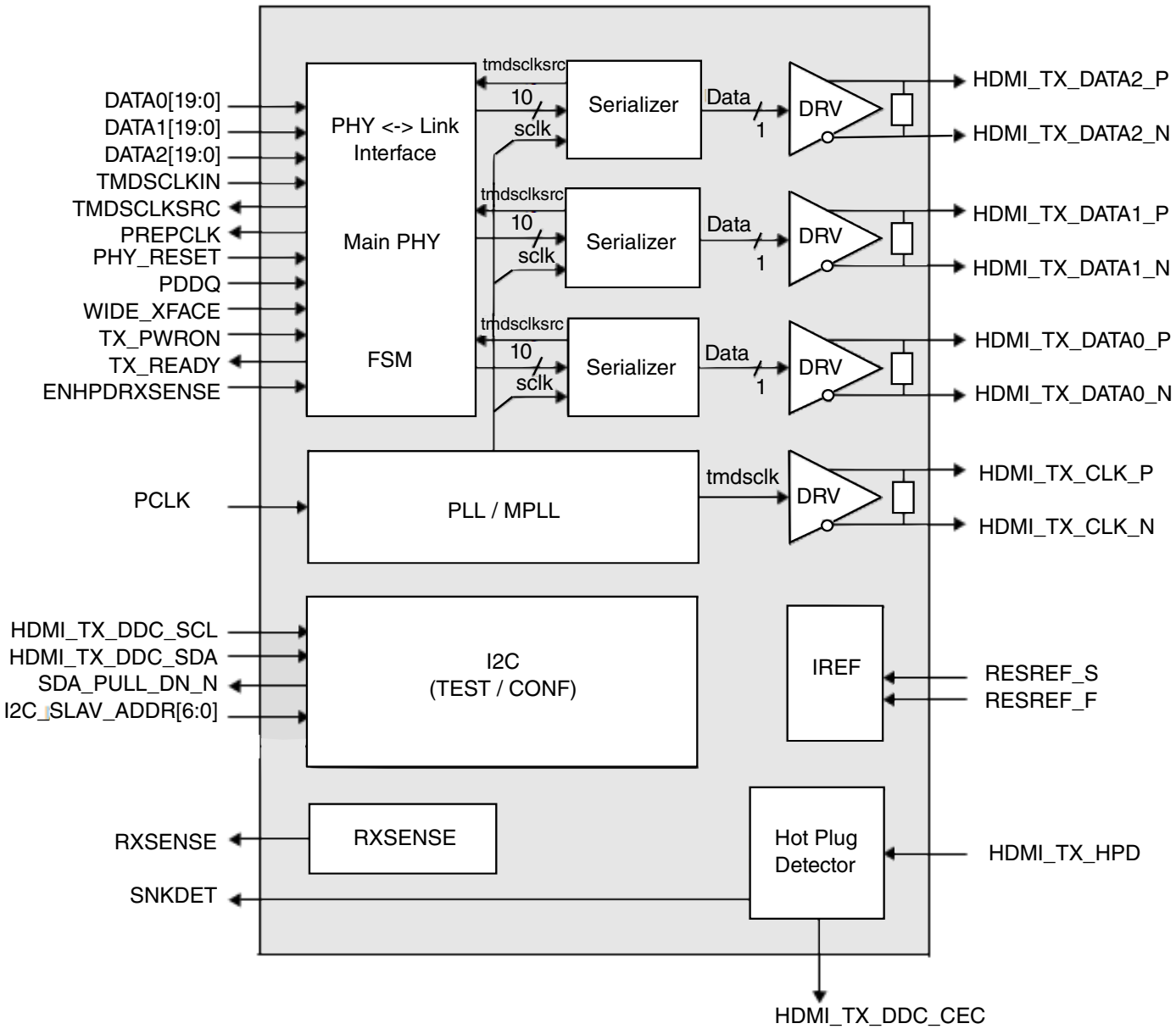


Figure 35-5. HDMI 3D Tx PHY Functional Block Diagram

The HDMI_PHY is the physical layer of an HDMI/DVI-compliant digital transmitter (source), capable of encoding and transmitting high-speed data streams carrying RGB video, audio, and control information. The HDMI Link Controller interface conforms to a 30- or 60-bit synchronous data interface (DATA0[19:0],DATA1[19:0],DATA2[19:0]).

The PHY includes two PLLs that synthesize the high-speed serial bit clock (required by the transmitter) from a reference pixel clock with a frequency of 13.5-270 MHz, for a transmitter capable of transmitting up to 9.2 Gbps using three lanes.

The HDMI_PHY drives audio and video across three TMDS data channels. Each serial TMDS link has a data rate range of 0.25-3.4 Gbps. The HDMI_PHY also drives the TMDS clock at 1/10th of the serial data rate with a frequency range of 25-270 MHz.

Within the HDMI_PHY, additional support blocks exist: the Bandgap block (for blocks biasing), the Resistor Calibration block, the ADC, and the analog test bus.

The HDMI_PHY accepts an input pixel clock (PCLK) with a frequency range of 13.5-270 MHz and accepts three channels of parallel data (TXDATA0[19:0], TXDATA1[19:0], TXDATA2[19:0]) that is synchronous with the TMDSCLKIN input clock. The TMDSCLKIN clock has a frequency range of 12.5-270MHz.

The TMDSCLKSRC clock output should be used by the HDMI transmitter controller as the source for TMDSCLKIN. The relation between TMDSCLKSRC and TMDSCLKIN should be constant between resets.

The HDMI_PHY's status and configuration is accessed through its I^2C interface. In Scan mode of operation, SCANCLK is bypassed to all PHY output clocks.

PLL/MPLL Operation

The PLL/MPLL can be configured in Coherent mode or Non-Coherent mode (default). In Coherent mode, the TMDS clock is the MPLL feedback clock, which is coherent with the MPLL's high-speed output clock, because both clocks are shaped by the MPLL response. In Non-Coherent mode, the TMDS clock is the MPLL reference clock, which is not coherent with the MPLL's high-speed output clock.

Driver Operation

The driver differential source termination, edge rate, pre-emphasis, and voltage level can be configured for maximum performance.

35.3.2 Operating Modes

This section describes various operating modes of the HDMI_PHY.

The HDMI_PHY can be placed in two different operating modes: Power-Down and Active.

The PHY_RESET signal is used to place the digital section of the IP in a well-defined state. Reset is active when the PHY_RESET signal is asserted high. PHY_RESET assertion also clears the configuration registers. Through the control registers, assertion of a soft reset to the macro is possible. This soft reset clears all system FSMs except I^2C and control registers.

For each separate video mode in which the HDMI_PHY is set to transmit, due to different operating frequency, color depth and pixel repetition that characterizes each one, you might need to configure the PHY block for correct operation and optimized performance. It is recommended that PHY configuration through the I^2C interface be done while the PHY is in Power-down mode. Configuration involves programming the PLL/MPLL internal blocks as well as the analog drivers' source termination value and edge rate control, for example. This programming is done through the I^2C interface. For more information about the I^2C interface, see I2C Interface Signals, for information about the PLL/MPLL configurations that must be made for each video mode of operation, see PLL/MPLL Generic Configuration Settings , and for information about the driver configurations, see Control Registers and Appendix A: Driver Voltage Level Configuration.

35.3.2.1 Power-Down Mode

This mode is the lowest power consumption mode. The PHY enters this mode when the TX_PWRON and ENHPDRXSENSE signals are set to 1'b0 and the PDDQ and PHY_RESET signals are set to 1'b1.

This mode is characterized by having all analog blocks disabled and digital logic quiet. Current consumption in this mode corresponds to the analog blocks standby current and digital logic leakage current.

To enable the HDMI 3D Tx PHY to enter Power-down mode, after TX_PWRON is set to 1'b0, the PCLK input must continue toggling until the TX_READY output signal is set to 1'b0 (indicating that the PHY has been correctly set in Power-down mode). If this sequence is not followed, TX_READY is not set to 1'b0, and the PHY remains in Active mode.

During Power-down mode, the ENHPDRXSENSE control signal can be set to 1'b1, enabling the HPD and RXSENSE-related circuitry. This setting enables the link controller to observe when an HDMI Rx is connected to the Tx and to power up the HDMI 3D Tx PHY after detecting the Rx.

During Power-down mode, if ENHPDRXSENSE is enabled, it is recommended that source terminations on both data and clock lines are enabled by setting tx_rescal[6:0] and ck_rescal[6:0] to 7'b1111111 on registers

0x04 and 0x05, respectively. Because the analog driver within the macro is disabled, enabling the source terminations will help pull-down the floating TMDS data/clock lines to a voltage level that will guarantee proper RXSENSE operation and output signaling when the Rx connects/disconnects from the PHY.

35.3.2.2 Active Mode

Active mode is the normal operational mode. To start processing received data from the link controller interface, the macro must first reach this state. The PHY is ready to transmit when TX_READY is asserted high.

In this state, the interface signals (DATA0[19:0], DATA1[19:0], DATA2[19:0]) are sent to the HDMI interface. Data is transferred according to the HDMI specification.

After entering Active mode, the HDMI Tx macro starts serially transmitting data received from the link controller to the TMDS interface. The way in which input data is encoded (depends on whether data corresponds to Video Data Period, Data Island Period, or Control Periods) depends on the HDMI protocol sequence. Data encoding is performed by the Link controller layer.

While operating in Active mode, the HDMI_PHY operates in one of the following modes (as defined in the HDMI specification, and as shown in the figure below):

- Video Data Period
- Data Island Period
- Control Period

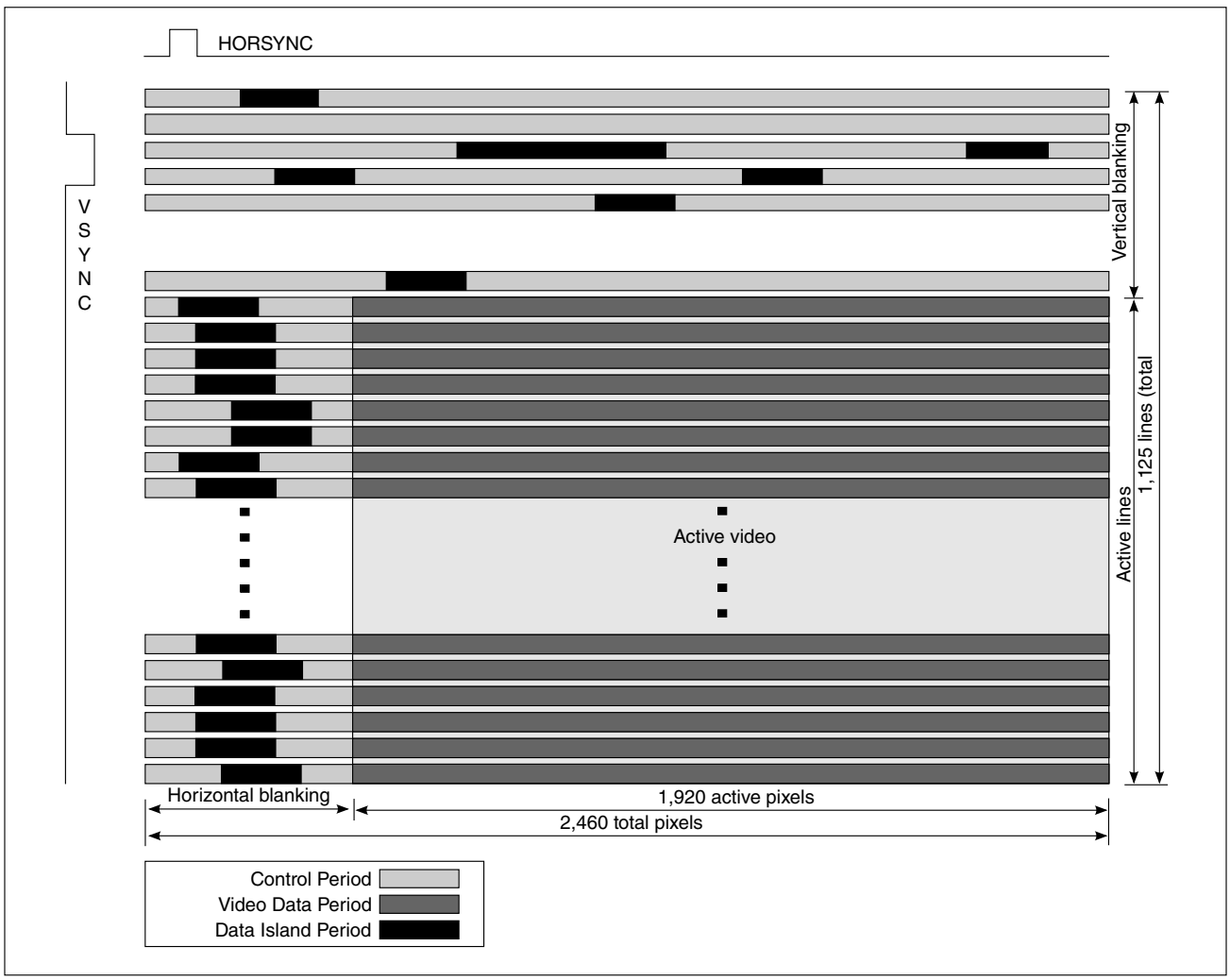


Figure 35-6. Active Mode

Video Data Period is used for the transmission of active pixel data. Transition Minimized (TMDS) data coding (8B10B) is used for 8-bit data transfers. In Deep-color modes, pixel data words are first packed in 8-bit multiple data groups. These groups are then fragmented into 8-bit length words for TMDS encoding.

Data Island Period is used for audio or other auxiliary data information transmission. Data is transferred in packet format using a similar Transition Minimized coding, in this case a TMDS Error Reduction Coding (TERC4). This is a 4B10B code.

Finally, Control Period is used when non-active video/audio or other auxiliary information needs to be transmitted. Only control information is transmitted in this mode. In Control Periods, data is transferred using a Transition Maximized (also TMDS) data

coding, in a form of 2B10B code. This property is very useful to enable proper word alignment on the sink (receiver) side. A Control Period is always entered between any other two periods that are not Control Periods.

35.3.2.2.1 Wide Interface Mode

To support the full HDMI bandwidth requirement, each of the three channels accepts 9-bit data and clock at a maximum rate of 340 MHz. In this case, the WIDE_XFACE input is set to 1'b0.

When WIDE_XFACE is set to 1'b1, each channel accepts 20-bit data and clock at a maximum rate of 170 MHz. This mode is meant to ease interface timing while maintaining the maximum HDMI bandwidth requirement.

When WIDE_XFACE is set to 1'b1, the TMDSCLKSRC clock frequency is not reduced to half simply through the WIDE_XFACE control. The frequency can be reduced to half through the I^2C interface, register 0x1E, cko_word_div_enb control bit. This control is independent from the WIDE_XFACE selection.

35.3.2.3 Power Sequence

The HDMI_PHY power sequence implementation controls the PLL/MPLL start of operation, resistor calibration, and clock alignment.

As shown in the figure below, during power-up, the Power Sequence finite state machine (FSM) advances through the PLL and MPLL power-up states by loading a counter at the beginning of each state (pwr_cnt) and advancing when the counter (pwr_cnt) decrements to 0. After the Resistor Calibration FSM is enabled in the RES_CAL state, the FSM waits for the assertion of rcal_adc_done before advancing. After the Clock Alignment FSM is enabled in the TX_CLK_ALIGN state, the Power Sequence FSM waits for the assertion of tx_ck_align_done before advancing to the TX_READY state. The TX_READY state asserts the TX_READY output signal, which indicates that the HDMI_PHY is ready to transmit the TMDS clock and data and that normal operation can begin.

If TX_PWRON is deasserted in the TX_READY state, the Power Sequence FSM moves to the DISABLE_TX state. The power-down sequence is initiated. This state disables the TMDS clock and data outputs. Next, the PWR_DN_PLL resets the PLL and MPLL blocks. Finally, the FSM returns to the PWR_DN state and waits for TX_PWRON assertion to begin the power-up sequence.

The HDMI_PHY enters Power-down mode only when the TX_READY output is deasserted. To enable the power sequence to reach this state, when TX_PWRON is set to 1'b0 while the HDMI_PHY is in the "TX_READY" state, the PCLK input must continue

toggling until TX_READY is deasserted. This sequence is required to enable the Power Sequence FSM to continue its execution and move from TX_READY to DISABLE_TX, later to PWR_DN_PLL, and finally to PWR_DN that corresponds to the power-down state.

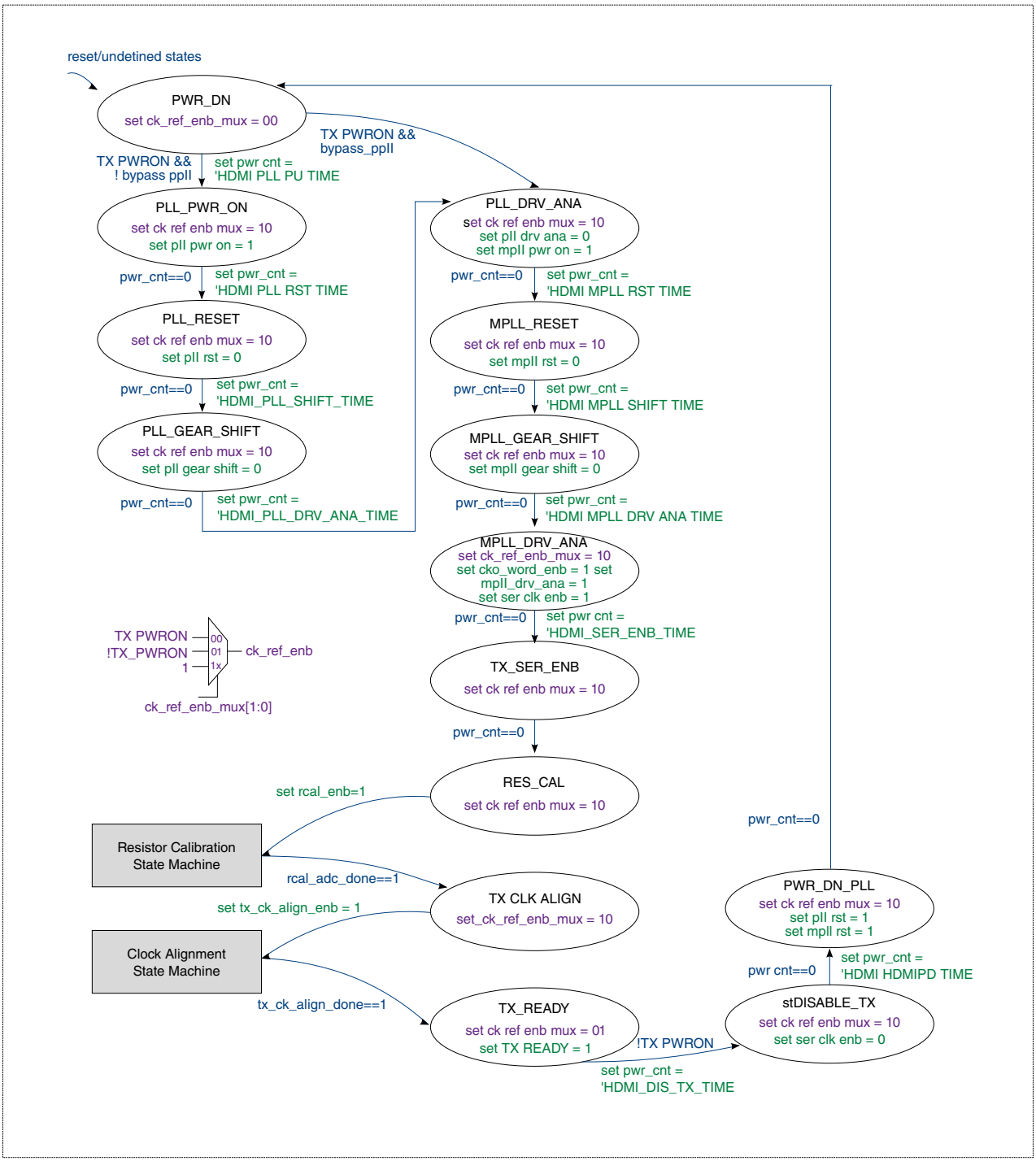


Figure 35-7. Power Sequence Finite State Machine

The following signals can be overridden through the control registers: bypass_ppll, pll_pwr_on, pll_rst, pll_gear_shift, pll_drv_ana, mpll_rst, mpll_gear_shift, cko_word_enb, mpll_drv_ana, ser_ckl_enb, rcal_enb, tx_clk_align_enb, ser_clk_enb, and ck_ref_enb.

The rcal_adc_done and tx_ck_align_done signals can be observed through the control registers or through the digital test bus (dtb[1:0], controlled through control registers).

The TX_PWRON and TX_READY signals belong to the HDMI 3D Tx PHY macro interface.

The power sequence can also be started through control registers, using the tx_pwron0, tx_pwron1, tx_pwron2, and ck_pwron signals.

TX_READY can also be overridden through control registers.

The clock source that clocks the Power Sequence FSM is controlled by the refclk_enb signal. This signal is asserted when TX_PWRON is asserted or when the state machine is in a state other than PWR_DN. This signal can be overridden through the control registers.

NOTE

Due to a potential different state on some analog nodes, it is recommended that the power-up sequence be run twice (after the first TX_READY assertion, the PHY can be powered down, then powered up again).

The clock alignment procedure can be done differently for each of the power-up sequence runs. For the first run, the clock alignment can be set to be performed channel-by-channel (tx_ck_align_mode = 1'b1, register 0x1C); for the second power-up, the clock alignment can be set to be performed for all channels at the same time (tx_ck_align_mode = 1'b0, register 0x1C) to ensure inter-pair skew close to 0 UI.

35.3.2.3.1 PLL/MPLL

During power-up, the PLL block and (optionally) MPLL block receive the pwron, rst, gear_shift, and drv_ana signals, as shown in the figure below. The minimum and maximum timing for these signals is listed in the table below. The power sequence then initiates the resistor calibration and clock alignment steps.

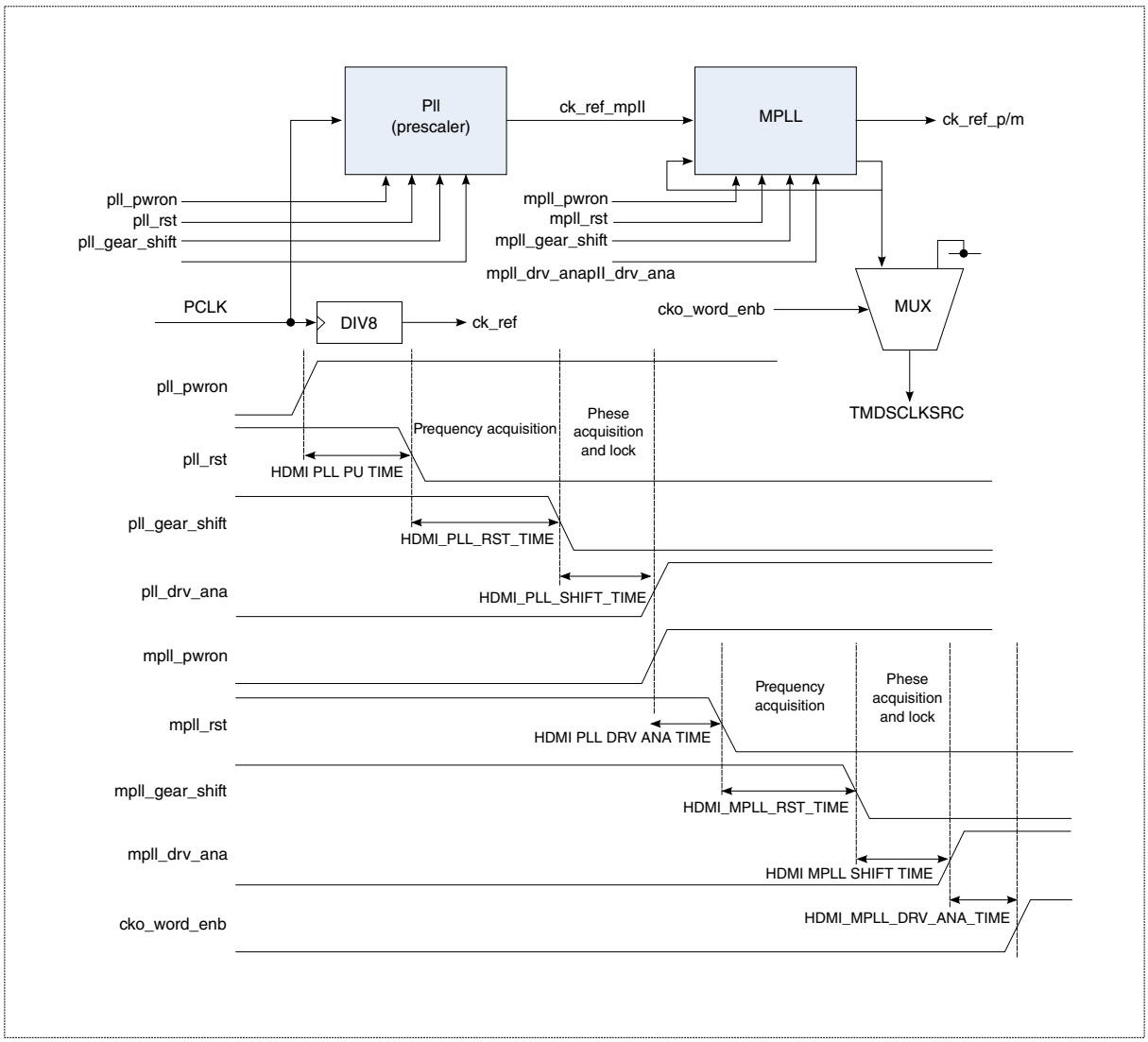


Figure 35-8. PLL/MPLL Power-Up Sequence

The following signals can be overridden through the control registers: pll_pwron, pll_rst, pll_gear_shift, pll_drv_ana, mpll_pwron, mpll_rst, mpll_gear_shift, mpll_drv_ana, and cko_word_enb.

ck_ref, ck_ref_p/m, and ck_ref_mpll are internal signals, while PCLK and TMDSCLKSRC belong to the HDMI 3D Tx PHY macro interface.

The table below provides the PLL/MPLL power-up times.

Table 35-5. PLL/MPLL Power-Up Time

Count Name	Count Value	Minimum Time (µs) (T(refclk) = 0.00294 µs)	Maximum Time (µs) (T(refclk = 0.07407 µs)
HDMI_PLL_PU_TIME	125	2.294	74.074
HDMI_PLL_RST_TIME	1250	22.940	740.740
HDMI_PLL_SHIFT_TIME	50	0.918	29.630
HDMI_PLL_DRV_ANA_TIME	125	2.294	74.074
HDMI_MPLL_RST_TIME	2500	45.880	1481.480
HDMI_MPLL_SHIFT_TIME	1000	18.352	592.592
HDMI_MPLL_DRV_ANA_TIME	50	0.918	29.630
HDMI_SER_ENB_TIME	50	0.918	29.630
Total PLL and MPLL Power-up Time		**94.513**	**3,051.849**

35.3.2.3.2 Resistor, ADC Calibration

Users can set the single-ended source termination resistance via the d_tx_term[2:0] control register bits, according to the table below.

Table 35-6. Single-Ended Source Termination Resistance Settings

N	d_tx_term[2:0]	R(Ω)
0	000	50
1	001	56.14
2	010	66.67
3	011	80
4	100	100
5	101	133.33
6	110	200
7	111	Open

To accurately set the specified termination, the resistor calibration measures the actual resistance of the external resistor. The resistance is measured using successive approximations of a 7-bit termination value by enabling each bit (one-hot) from MSB to LSB.

Users select the termination value by setting the d_tx_term[2:0] control register bits. The HDMI 3D Tx PHY includes differential source termination on the drivers.

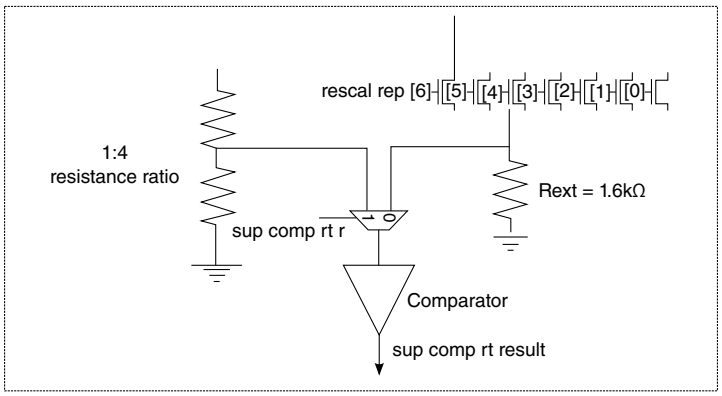


Figure 35-9. External Resistor Measurement

The exact value of Rext is determined to create an exact termination.

The analog support block applies a current to the external resistor and compares the resultant voltage. The result (sup_comp_rt_result) is registered by the Calibration block.

The source termination is then set according to the following equation:

tx_rescal = rescal_rep x (1 - 0.125 x *N*)

Instead of subtracting by (1/8th x rescal_rep x *N*), rescal_rep is initially multiplied by 8 (left shift by 3 bits), then tx_rescal is subtracted *N* number of times. *N* sets clk_cnt, which sets the number of subtraction loops, as shown in the table below. This table represents the clk_cnt required to trigger tx_rescal_done. The final result is divided by 8 (right shift by 3 bits) and rounded up if necessary.

Table 35-7. clk_cnt Necessary to Trigger tx_rescal_done

N/d_tx_term	clk_cnt to Trigger tx_rescal_done
0	<all>
1	0
2	1
3	2
4	3
5	4
6	5
7	<all>

The figure below shows the FSM for the resistor and ADC calibration.

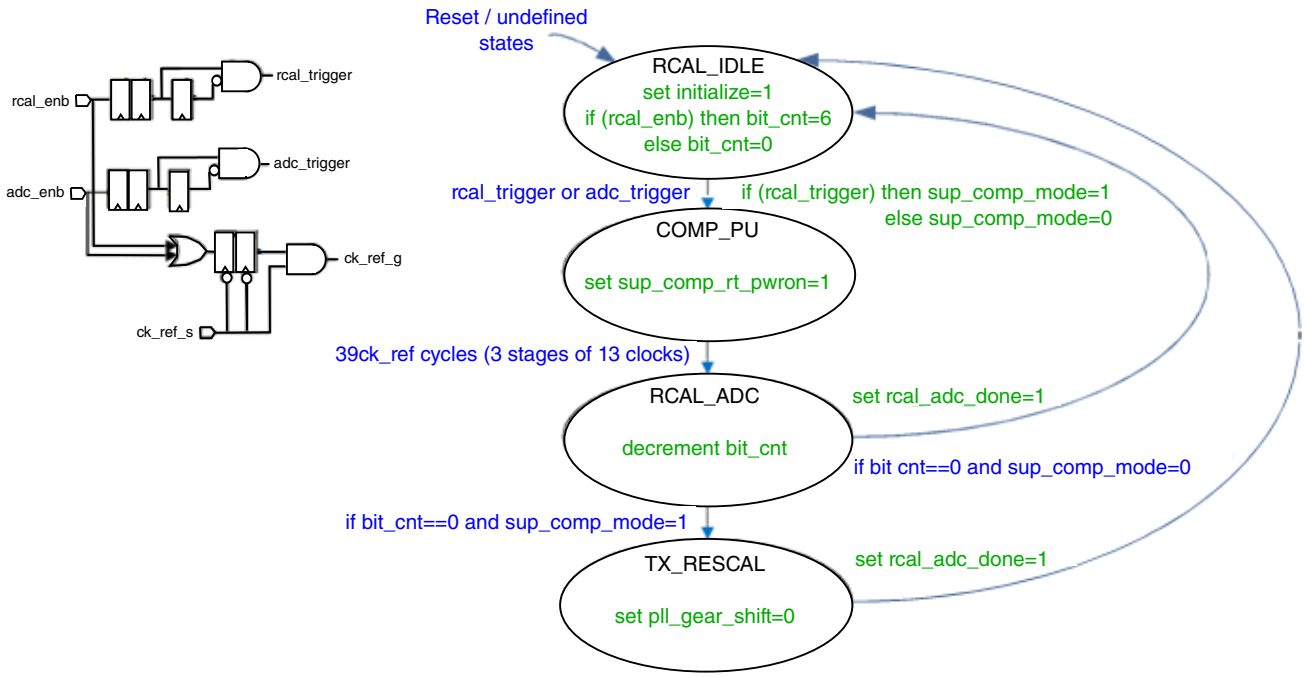


Figure 35-10. Finite State Machine for the Resistor and ADC Calibration

The rcal_enb signal is asserted by the Power Sequence FSM, but this signal can be overridden through the control registers.

The rescal_rep, sup_comp_mode, sup_comp_rt_r, sup_comp_rt_result, tx_rescal, sup_comp_rt_pwron, and adc_enb signals can be overridden through the control registers.

The rcal_adc_done signal can be observed through the control registers or through the digital test bus (dtb[1:0], controlled through control registers).

35.3.2.3.3 Clock Alignment

There are two clocks in the transmit clock path, TMDSCLKIN (from the parallel HDMI controller interface), and internal ck_tx_out (a divided-down clock, generated from the MPLL clock). These two clocks must be properly synchronized, aligning them before transmitting data through the HDMI 3D Tx PHY.

The alignment results in the internal tx_ser_clk (buffered version of TMDSCLKIN) being delayed in the range of 0.6-0.8 UI with respect to the TMDSCLKIN clock.

The Clock Alignment module adjusts the alignment between TMDSCLKIN and tx_ser_clk by generating internal clock kill signals called tx_ser_clk_kill[2:0]. On the rising edge of each tx_ser_clk_kill[2:0] signal, the corresponding tx_ser_clk[2:0] is delayed by 0.2 UI.

The clock alignment enable input, tx_ck_align_enb, is generated by the power sequence, but this input can also be overridden through the control registers.

Because it is assumed that channel 1 (the middle channel) has the median delays among the three channels, the state machine will compare TMDSCLKIN and tx_ser_clk[1] only when the register bit, tx_ck_align_mode, is low (default). In this case, tx_ser_clk_kill[2:0] is asserted simultaneously. When tx_ck_align_mode is high, each channel is aligned individually. Channel 2 is aligned first, followed by channel 0 and lastly channel 1. The tx_ck_align_mode signal is controlled through the control registers.

The internal clock kill signals, tx_ser_clk_kill2, tx_ser_clk_kill1, and tx_ser_clk_kill0, are generated by the Clock Alignment state machine, but these signals can be overridden through the control registers.

35.3.2.4 Color Depth and Color Mode Selection

This section describes HDMI 3D Tx PHY color depth and color mode selection.

35.3.2.4.1 Pixel Repetition Clock Generation Selection

For small displays operating a TMDS clock below 25 MHz or applications that require an increase in available bandwidth for audio and auxiliary information transmission, the HDMI_PHY can generate pixel repetition clocks.

Pixel repetition signals are dedicated to configure pixel repetition rates. These signals affect only the relation between PCLK and PREPCLK.

PCLK 1 / (pixel repetition) x PREPCLK

35.3.3 Configuration and Test Mode

This section describes HDMI 3D Tx PHY configuration and test mode operation.

The HDMI 3D Tx PHY can be configured and set in test mode through the supported I^2C interface.

To avoid transient periods in the macro operation during reconfiguration procedures, it is highly recommended that the I^2C interface be used while the HDMI 3D Tx PHY is in Power-down mode.

Control Registers describes all the registers and corresponding fields that are available through the I^2C interface.

35.3.3.1 Power-Up Configuration

Before powering up the HDMI 3D Tx PHY, configure the PHY for proper operation and performance.

Configuring the PHY through the I^2C interface is as follows.

NOTE

The recommended settings for the PLL/MPLL mode of operation and driver configurations provided in this section are presented as the default value that should be considered and initially configured. Unless stated otherwise, these recommended settings apply to all operating frequencies.

PLL/MPLL Mode of Operation

Configuration summary: Configure PLL/MPLL mode of operation as Single or Two-PLL in Coherent or Non-Coherent mode. For more information about these modes of operation, see Power-Up Requirements.

Table 35-8. PLL/MPLL Mode of Operation

Fields	Bits	Default
Register0x13		
bypass_ppll	11	1'b0
Register0x17		
cko_sel	2-1	2'b11

PLL/MPLL Clock Dividers and Analog Configuration

Configuration summary: Configured for each video mode. For information about the PLL and MPLL divider settings, see Table B-2 .

Table 35-9. PLL/MPLL Clock Dividers and Analog Configuration

Fields	Bits
Register 0x06	
prep_div[1:0]	14-13
mpll_n_ctrl[1:0]	8-7
pll_n_ctrl[1:0]	6-5
pixel_rep[2:0]	4-2
clr_dpth[1:0]	1-0

Table continues on the next page...

Table 35-9. PLL/MPLL Clock Dividers and Analog Configuration (continued)

Fields	Bits
Register 0x10	
mpll_prop_cntrl[2:0]	11-9
mpll_int_cntrl[2:0]	8-6
pll_prop_cntrl[2:0]	5-3
pll_int_cntrl[2:0]	2-0
Register 0x15	
pll_gmp_cntrl[1:0]	3-2
mpll_gmp_cntrl[1:0]	1-0

Driver Edge Rate Control

Configuration summary: This bus controls the slew rate of the clock and data output drivers.

Table 35-10. Driver Edge Rate Control

Fields	Bits	Default
Register 0x06		
tx_edgerate[1:0]	12-11	2'b00
ck_edgerate[1:0]	10-9	2'b00

Driver Single-Ended Source Termination

Configuration summary: This bus controls the driver single-ended source termination.

Table 35-11. Driver Single-Ended Source Termination

d-tx_term	R (ohm)
000	50
001	56.14
010	66.67
011	80
100	100
101	133.33
110	200
111	Open circuit

Fields	Bits	Default
Register 0x19		
d_tx_term[2:0]	2-0	3'b101

Driver Voltage Level

Configuration summary: For information about the driver voltage level settings, see Appendix A: Driver Voltage Level Configuration. Driver voltage level configuration is dependent on driver differential source termination and driver pre-emphasis settings.

Table 35-12. Driver Voltage Level

Fields	Bits	Default
Register 0x0E		
sup_tx_lvl[4:0]	9-5	25-148.5 MHz: 5'b10000 222.75-297 MHz: 5'b01111 340 MHz: 5'b01110
sup_ck_lvl[4:0]	4-0	25-148.5 MHz: 5'b10000 222.75-297 MHz: 5'b01111 340 MHz: 5'b01110

Driver Pre-Emphasis

Configuration summary: Enables trailer drivers to enable pre-emphasis operation. Pre-emphasis is achieved by reducing the drive level of a non-transition bit with respect to a transition bit. The following table summarizes the total output current configured with pre-emphasis (I is current of unit current source).

Table 35-13. Driver Pre-Emphasis

Total Output Current	tx_symon	tx_traon	tx_trbon
0 * I	0	0	0
36 * I	1	0	0
36 * I + 3 * I	1	0	1
36 * I + 6 * I	1	1	0
36 * I + 6 * I + 3 * I	1	1	1

The amount of pre-emphasis is programmed through the tx_traon and tx_trbon signals in the control register, address 0x09 as follows.

Field	Bit	Default

Table continues on the next page...

tx_trbon	1	25-148.5 MHz: 1'b0 222.75-340 MHz: 1'b1
tx_traon	2	1'b0

35.4 System-Level Implementation

35.4.1 System Operation

This section describes HDMI 3D Tx PHY system operation including power-up/power-down and power- on reset.

35.4.1.1 Powering Up and Powering Down

To set the HDMI_PHY in Power-down mode, set the TX_PWRON signal to 1'b0 and the PDDQ signal to 1'b1. To power up the HDMI 3D Tx PHY and place it in Active mode, set TX_PWRON to 1'b1 and PDDQ to 1'b0.

Any configuration programmed on the HDMI_PHY must be done in Power-down mode. To configure the PHY through the I^2C interface, set PHY_RESET to 1'b0 (to move the digital core from a reset state and to enable programming).

The figure below shows the power-down and power-up sequences.

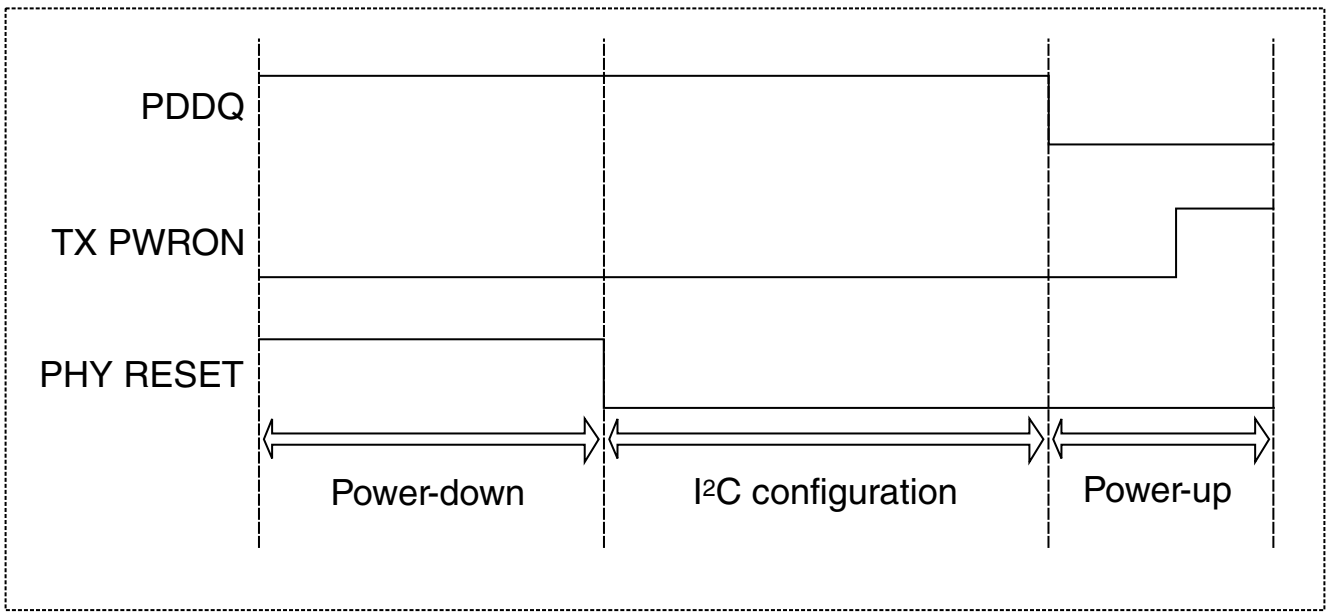


Figure 35-11. Power-Down and Power-Up Sequence

The power-up sequence starts by advancing one finite state machine (FSM) through the Phase-Locked Loop (PLL) and Multiplexed Phase-Locked Loop (MPLL) power-up states. Afterwards, resistor calibration is performed and clock alignment is enabled. When clock alignment is completed, the TX_READY signal is set to 1'b1 to indicate that the PHY is ready to transmit the TMDS clock and that data transmission and normal operation can start.

In Power-down mode, TMDS clock and data lines are disabled.

Power-Up Configuration

Before powering up the HDMI_ PHY, configure the PHY for proper operation and performance. The following PHY characteristics can be configured:

- PLL/MPLL mode of operation: Configure PLL/MPLL mode of operation as Single or Two-PLL in Coherent or Non-Coherent mode.
- PLL/MPLL clock dividers and analog configuration: Configure for each video mode.
- Driver edge rate control: Slew rate of clock and data output drivers
- Driver differential source termination: Differential source termination of clock and data output drivers
- Driver voltage level: Voltage reference level for clock and data channels
- Driver pre-emphasis: Enable and control pre-emphasis operation

For more information about this configuration and the associated control registers and signals, see Power-Up Configuration.

35.4.1.2 Active Mode Requirements

During normal operation, the RXSENSE circuitry must always be enabled (ENHPDRXSENSE asserted high).

During normal operation, if ENHPDRXSENSE is deactivated (asserted low), the PHY's analog portion is turned off.

35.4.1.3 Power-Up Requirements

Before setting the TX_PWRON signal high and the PDDQ signal low to power up the HDMI 3D Tx PHY, the PCLK and TMDSCLKIN input reference clocks must be stable and within their specified parameters, and power supply rails must be stable and within the specifications.

35.4.1.4 Power Supply Sequence When the HDMI 3D Tx PHY is Not Used

The HDMI 3D Tx PHY supports power collapsing.

There is no constraint on the power supply sequence when the HDMI 3D Tx PHY is not used. However, it is recommended that you short the data/clock lines to ground (or leave them floating). Activity at the data/clock lines or shorting the data/clock lines to the 3v3 rail (from Rx) should be avoided.

35.4.1.5 Power-Down Requirements

To enable the HDMI 3D Tx PHY to enter Power-down mode, after TX_PWRON is set to 1'b0, the PCLK input must continue toggling until the TX_READY output signal is set to 1'b0 (indicating that the PHY has been correctly set in Power-down mode). This sequence is required to enable the power-sequence machine to continue its execution and move from its states until the machine reaches the power-down state.

If this sequence is not followed, TX_READY is not set to 1'b0, and the HDMI 3D Tx PHY remains in Active mode.

35.5 Reference Clock

This chapter describes the reference clock that the HDMI_PHY supports.

The HDMI_PHY system requires an input clock signal that must be applied to the PCLK input with the following constraints:

- Minimum PCLK period/frequency (pixel repetition): 74 ns/13.5 MHz
- Minimum PCLK period/frequency (no pixel repetition): 39.7 ns/24.175 MHz
- PCLK duty cycle: 40-60%
- Maximum PCLK long-term RMS jitter: 100 ps

35.6 Control Registers

35.6.1 Control Registers Module Design Architecture

The Control Registers module is designed to provide the HDMI 3D Tx PHY with a file register component (36 words where each word comprises 16 bits).

Each register is designed to correspond to one of the following access types.

- **Read-only**: The HDMI controller can only read from this register. No writing by the HDMI controller is permitted.
- **Read/write**: The HDMI controller can read from and write to this register.
- **Read/write/override**: The HDMI controller can read from and write to this register. However, if the MSB of the register is set to 0, the value read by the HDMI controller will not be the value stored in the register; instead, the value will be the internal value of the HDMI PHY. In particular, this event occurs for 0x14, 0x18, 0x0A, and 0x0F where the value is read from the I^2C interface when PHY_RESET and TX_PWRON are low.
- **Read/write/asynchronous set-on-done**: The HDMI controller can read from and write to this register synchronously. However, three internal modules (Resistance Calibration, Clock Alignment, and Tx Scope) can write to this register asynchronously.

35.7 HDMI_PHY Memory Map/Register Definition

NOTE

HDMI_PHY registers are accessed through the I2C Master Interface of the HDMI Controller. See HDMI Transmitter (HDMI) for more information.

HDMI_PHY memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Power Control (HDMI_PHY_PWRCTRL)	16	R/W	0000h	35.7.1/1918
1	Serializer Divider Control (HDMI_PHY_SERDIVCTRL)	16	R/W	0000h	35.7.2/1920
2	Serializer Clock Control (HDMI_PHY_SERCKCTRL)	16	R/W	0000h	35.7.3/1920
3	Serializer Clock Kill Control (HDMI_PHY_SERCKKILLCTRL)	16	R/W	0000h	35.7.4/1921
4	Transmitter and Resistance Calibration Control (HDMI_PHY_TXRESCTRL)	16	R/W	0000h	35.7.5/1922
5	Clock Calibration Control (HDMI_PHY_CKCALCTRL)	16	R/W	0000h	35.7.6/1923
6	Color Depth, Pixel Repetition, Clock Divider for PLL and MPLL, and Edge Rate Control (HDMI_PHY_CPCE_CTRL)	16	R/W	0400h	35.7.7/1924
7	Tx and Clock Measure Control (HDMI_PHY_TXCLKMEASCTRL)	16	R/W	0000h	35.7.8/1926
8	Tx Measure Control (HDMI_PHY_TXMEASCTRL)	16	R/W	0000h	35.7.9/1927
9	Clock Symbol and Transmitter Control (HDMI_PHY_CKSYMTXCTRL)	16	R/W	0000h	35.7.10/ 1929
A	Comparator Sequence Control (HDMI_PHY_CMPSEQCTRL)	16	R/W	0000h	35.7.11/ 1930
B	Comparator Power Control (HDMI_PHY_CMPPWRCTRL)	16	R/W	0000h	35.7.12/ 1931
C	Comparator Mode Control (HDMI_PHY_CMPMODECTRL)	16	R/W	0000h	35.7.13/ 1931
D	Measure Control (HDMI_PHY_MEASCTRL)	16	R/W	0000h	35.7.14/ 1932
E	Voltage Level Control (HDMI_PHY_VLEVCTRL)	16	R/W	0000h	35.7.15/ 1933
F	Digital-to-Analog Control (HDMI_PHY_D2ACTRL)	16	R/W	0000h	35.7.16/ 1934
10	Current Control (HDMI_PHY_CURRCTRL)	16	R/W	08ABh	35.7.17/ 1935
11	Drive Analog Control (HDMI_PHY_DRVANACTRL)	16	R/W	0003h	35.7.18/ 1935
12	PLL Measure Control (HDMI_PHY_PLLMEASCTRL)	16	R/W	0000h	35.7.19/ 1936
13	PLL Phase and Bypass Control (HDMI_PHY_PLLPHBYCTRL)	16	R/W	0000h	35.7.20/ 1938

Table continues on the next page...

HDMI_PHY memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
14	Gear Shift, Reset Mode, and Power State Control (HDMI_PHY_GRP_CTRL)	16	R/W	0000h	35.7.21/ 1939
15	Gmp Control (HDMI_PHY_GMPCTRL)	16	R/W	0000h	35.7.22/ 1940
16	MPLL Measure Control (HDMI_PHY_MPLLMEASCTRL)	16	R/W	0000h	35.7.23/ 1941
17	MPLL and PLL Phase, Scope Clock Select, and MUX Clock Control (HDMI_PHY_MSM_CTRL)	16	R/W	0000h	35.7.24/ 1943
18	Scope, Comparator Result and Power Bad Status (HDMI_PHY_SCRPB_STATUS)	16	R	0000h	35.7.25/ 1944
19	Transmission Termination (HDMI_PHY_TXTERM)	16	R/W	0007h	35.7.26/ 1946
1A	Power Sequence, TX Clock Alignment, Resistance Calibration, Pattern Generator Skip Bit, and TMDS Encoder Enable (HDMI_PHY_PTRPT_ENBL)	16	R/W	0000h	35.7.27/ 1947
1B	Pattern Generator Mode (HDMI_PHY_PATTERNGEN)	16	R/W	0000h	35.7.28/ 1949
1C	The Soft-Reset and DAC Enable, Clock Alignment and PG Mode (HDMI_PHY_SDCAP_MODE)	16	R/W	0000h	35.7.29/ 1950
1D	Scope Mode register (HDMI_PHY_SCOPEMODE)	16	R/W	0000h	35.7.30/ 1952
1E	Digital Transmission Mode (HDMI_PHY_DIGTXMODE)	16	R/W	0000h	35.7.31/ 1953
1F	Scope, Transmission Clock Alignment, and Resistance Calibration Set-on-Done Status (HDMI_PHY_STR_STATUS)	16	R/W	0000h	35.7.32/ 1957
20	Scope Counter on Channel 0 (HDMI_PHY_SCOPECNT0)	16	R	0000h	35.7.33/ 1959
21	Scope Counter on Channel 1 (HDMI_PHY_SCOPECNT1)	16	R	0000h	35.7.34/ 1959
22	Scope Counter on Channel 2 (HDMI_PHY_SCOPECNT2)	16	R	0000h	35.7.35/ 1960
23	Scope Counter on Clock Channel (HDMI_PHY_SCOPECNTCLK)	16	R	0000h	35.7.36/ 1960
24	Scope Sample Count MSB, Scope Sample Repetition (HDMI_PHY_SCOPESAMPLE)	16	R/W	13C0h	35.7.37/ 1961
25	Scope Counter MSB Channel 0 and Channel 1 (HDMI_PHY_SCOPECNTMSB01)	16	R	0000h	35.7.38/ 1962
26	Scope Counter MSB Channel 2 and Clock Channel (HDMI_PHY_SCOPECNTMSB2CK)	16	R	0000h	35.7.39/ 1962

35.7.1 Power Control (HDMI_PHY_PWRCTRL)

Register name: PWRCTRL

Access type: Read/write/override

Address: 0x00

Value at reset: 0x0000

Address: 0h base + 0h offset = 0h

Bit	15	14	13	12	11	10	9	8
Read Write	Override	Reserved						
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved			tx_pwron	tx_pwron0	tx_pwron1	tx_pwron2	ck_pwron
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_PWRCTRL field descriptions

Field	Description
15 Override	If the Override bit is set to 1, the working value is the Override bit value, not the registered value.
14–5 -	This field is reserved. Reserved
4 tx_pwron	Transmitter Power-On This bit enables users to power down the entire PHY through the I2C interface. 0 Power off the PHY, if the Override bit is 1. 1 Power on the PHY, if the Override bit is 1.
3 tx_pwron0	Transmitter Power-On 0 This bit powers on or powers off the transmitter driver for channel 0. 0 Power off the transmitter driver for the first channel, if the Override bit is 0. 1 Power on the transmitter driver for the first channel, if the Override bit is 0.
2 tx_pwron1	Transmitter Power-On 1 This bit powers on or powers off the transmitter driver for channel 1. 0 Power off the transmitter driver for the second channel, if the Override bit is 0. 1 Power on the transmitter driver for the second channel, if the Override bit is 0.
1 tx_pwron2	Transmitter Power-On 2 This bit powers on or powers off the transmitter driver for channel 2. 0 Power off the transmitter driver for the third channel, if the Override bit is 0. 1 Power on the transmitter driver for the third channel, if the Override bit is 0.
0 ck_pwron	Clock Power-On This bit powers on or powers off the clock driver. 0 Power off the clock driver, if the Override bit is 0. 1 Power on the clock driver, if the Override bit is 0.

35.7.2 Serializer Divider Control (HDMI_PHY_SERDIVCTRL)

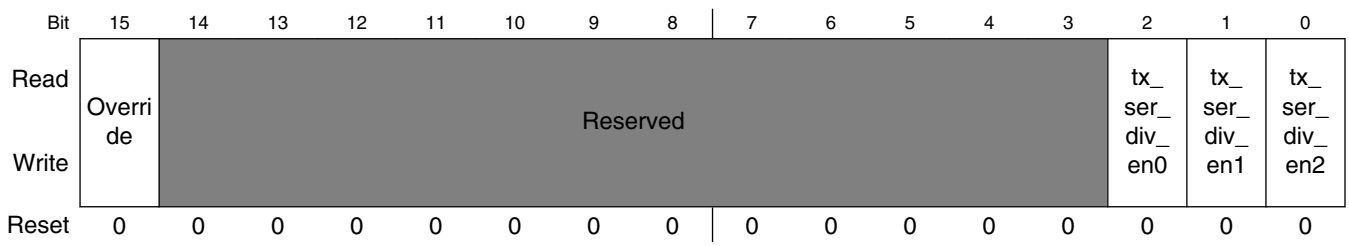
Register name: SERDIVCTRL

Access type: Read/write/override

Address: 0x01

Value at reset: 0x0000

Address: 0h base + 1h offset = 1h



HDMI_PHY_SERDIVCTRL field descriptions

Field	Description
15 Override	If the Override bit is set to 1, the working value is the Override bit value, not the registered value.
14–3 -	This field is reserved. Reserved
2 tx_ser_div_en0	Transmitter Serializer Divider Enable 0 This bit enables or disables the low-speed clock in serializer 0. 0 Disable the low-speed clock in the first serializer, if the Override bit is 0. 1 Enable the low-speed clock in the first serializer, if the Override bit is 0.
1 tx_ser_div_en1	Transmitter Serializer Divider Enable 1 This bit enables or disables the low-speed clock in serializer 1. 0 Disable the low-speed clock in the second serializer, if the Override bit is 0. 1 Enable the low-speed clock in the second serializer, if the Override bit is 0.
0 tx_ser_div_en2	Transmitter Serializer Divider Enable 2 This bit enables or disables the low-speed clock in serializer 2. 0 Disable the low-speed clock in the third serializer, if the Override bit is 0. 1 Enable the low-speed clock in the third serializer, if the Override bit is 0.

35.7.3 Serializer Clock Control (HDMI_PHY_SERCKCTRL)

Register name: SERCKCTRL

Access type: Read/write/override

Address: 0x02

Value at reset: 0x0000

Address: 0h base + 2h offset = 2h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Override	Reserved												tx_ser_clk_en0	tx_ser_clk_en1	tx_ser_clk_en2
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HDMI_PHY_SERCKCTRL field descriptions

Field	Description
15 Override	If the Override bit is set to 1, the working value is the Override bit value, not the registered value.
14–3 -	This field is reserved. Reserved
2 tx_ser_clk_en0	Transmitter Serializer Clock Enable 0 This bit enables or disables the high-speed clock in serializer 0. 0 Disable the high-speed clock in the first serializer, if the Override bit is 0. 1 Enable the high-speed clock in the first serializer, if the Override bit is 0.
1 tx_ser_clk_en1	Transmitter Serializer Clock Enable 1 This bit enables or disables the high-speed clock in serializer 1. 0 Disable the high-speed clock in the second serializer, if the Override bit is 0. 1 Enable the high-speed clock in the second serializer, if the Override bit is 0.
0 tx_ser_clk_en2	Transmitter Serializer Clock Enable 2 This bit enables or disables the high-speed clock in serializer two. 0 Disable the high-speed clock in the third serializer, if the Override bit is 0. 1 Enable the high-speed clock in the third serializer, if the Override bit is 0.

35.7.4 Serializer Clock Kill Control (HDMI_PHY_SERCKKILLCTRL)

Register name: SERCKKILLCTRL

Access type: Read/write/override

Address: 0x03

Value at reset: 0x0000

Address: 0h base + 3h offset = 3h

Bit	15	14	13	12	11	10	9	8
Read / Write	Override	Reserved						
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read / Write	Reserved					tx_ser_clk_kill0	tx_ser_clk_kill1	tx_ser_clk_kill2
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_SERCKKILLCTRL field descriptions

Field	Description
15 Override	If the Override bit is set to 1, the working value is the Override bit value, not the registered value.
14–3 -	This field is reserved. Reserved
2 tx_ser_clk_kill0	Transmitter Serializer Clock Kill 0 This bit is used to delay the tx_ck_out0 sampling clock by a time $t_{delay,}$ where t_{delay} equals one period of refclk (340 MHz). This delay equals 0.2 UI of the tx_ck_out0 clock. 0 No effect, if the Override bit is 0. 1 Delay the tx_ck_out0 clock by 0.2 UI, if the Override bit is 0.
1 tx_ser_clk_kill1	Transmitter Serializer Clock Kill 1 This bit is used to delay the tx_ck_out1 sampling clock by a time $t_{delay,}$ where t_{delay} equals one period of refclk (340 MHz). This delay equals 0.2 UI of the tx_ck_out1 clock. 0 No effect, if the Override bit is 0. 1 Delay the tx_ck_out1 clock by 0.2 UI, if the Override bit is 0.
0 tx_ser_clk_kill2	Transmitter Serializer Clock Kill 2 This bit is used to delay the tx_ck_out2 sampling clock by a time $t_{delay,}$ where t_{delay} equals one period of refclk (340 MHz). This delay equals 0.2 UI of the tx_ck_out2 clock. 0 No effect, if the Override bit is 0. 1 Delay the tx_ck_out2 clock by 0.2 UI, if the Override bit is 0.

35.7.5 Transmitter and Resistance Calibration Control (HDMI_PHY_TXRESCTRL)

Register name: TXRESCALCTRL

Access type: Read/write/override

Address: 0x04

Value at reset: 0x0000

Address: 0h base + 4h offset = 4h

Bit	15	14	13	12	11	10	9	8
Read Write	Override	Reserved	tx_rescal[6:0]					
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	tx_ rescal[6:0]	rescal_rep[6:0]						
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_TXRESCTRL field descriptions

Field	Description
15 Override	If the Override bit is set to 1, the working value is the Override bit value, not the registered value.
14 - Reserved	This field is reserved.
13–7 tx_rescal[6:0]	Transmitter Resistance Calibration This bus controls the parallel termination of the transmitter drivers and sets termination to a value based on the calibration algorithm performed in the support resistance calibration module. tx_rescal[6:0] provides 64 termination levels; the specific values are defined from lab test results.
rescal_rep[6:0]	Resistance Calibration Replica This bus controls the bias voltage of the transmitter driver. rescal_rep[6:0] provides 64 voltage levels; the specific values are defined from lab test results.

35.7.6 Clock Calibration Control (HDMI_PHY_CKCALCTRL)

Register name: CKCALCTRL

Access type: Read/write/override

Address: 0x05

Value at reset: 0x0000

Address: 0h base + 5h offset = 5h

Bit	15	14	13	12	11	10	9	8
Read Write	Override	Reserved						
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved	ck_rescal[6:0]						
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_CKCALCTRL field descriptions

Field	Description
15 Override	If the Override bit is set to 1, the working value is the Override bit value, not the registered value.
14–7 -	This field is reserved. Reserved
ck_rescal[6:0]	Clock Resistance Calibration This bus controls the termination of the clock driver and sets the termination to a value based on the calibration algorithm performed in the support resistance calibration module. ck_rescal[6:0] provides 64 termination levels; the specific values are defined from lab test results.

35.7.7 Color Depth, Pixel Repetition, Clock Divider for PLL and MPLL, and Edge Rate Control (HDMI_PHY_CPCE_CTRL)

Register name: -

Access type: Read/write

Address: 0x06

Value at reset: 0x0400

Address: 0h base + 6h offset = 6h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved	prep_div[1:0]		tx_edgerate[1:0]		ck_edgerate[1:0]		mpll_n_ cntrl[1:0]
Reset	0	0	0	0	0	1	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	mpll_n_ cntrl[1:0]	pll_n_cntrl[1:0]		pixel_rep[2:0]			clr_dpth[1:0]	
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_CPCE_CTRL field descriptions

Field	Description
15 -	This field is reserved. Reserved
14–13 prep_div[1:0]	Digital Pixel Repetition Divider Controls the ratio by which the internal TMDS clock is divided to generate PREPCLK. 00 Divide by 1 (8 bit). 01 Divide by 1.25 (10 bits). 10 Divide by 1.5 (12 bits). 11 Divide by 2 (16 bits).
12–11 tx_edgerate[1:0]	Transmitter Edge Rate

Table continues on the next page...

HDMI_PHY_CPCE_CTRL field descriptions (continued)

Field	Description
	This bus controls the slew rate of the transmitter output driver. 0 tx_edgerate[0]: Slow edges 1 tx_edgerate[0]: Fast edges 0 tx_edgerate[1]: Disable edge rate override. 1 tx_edgerate[1]: Enable edge rate override.
10–9 ck_edgerate[1:0]	Clock Edge Rate This bus controls the slew rate of the clock output driver. 0 ck_edgerate[0]: Slow edges 1 ck_edgerate[0]: Fast edges 0 ck_edgerate[1]: Disable edge rate override. 1 ck_edgerate[1]: Enable edge rate override.
8–7 mpll_n_cntrl[1:0]	Programmable Divider Control This bus controls the programmable divider modulus, which are set based on the ck_ref_mpll_p/m (TMDS rate) input reference frequency to keep the ring oscillator within the required range (925 MHz through 1.85 GHz in MPLL). 00 N = 1 (for TMDS rate of 184.1-370 MHz) 01 N = 2 (for TMDS rate of 92.51-185 MHz) 10 N = 4 (for TMDS rate of 45.26-92.5 MHz) 11 N = 8 (for TMDS rate up to 45.25 MHz)
6–5 pll_n_cntrl[1:0]	Programmable Divider Control This bus controls the programmable divider modules, which are set based on the refclk_p/m (pixel rate) input reference frequency to keep the ring oscillator within the required range (740 MHz through 1.48 GHz in PLL). 00 N = 1 (for TMDS rate of 184.1-370 MHz) 01 N = 2 (for TMDS rate of 92.51-185 MHz) 10 N = 4 (for TMDS rate of 45.26-92.5 MHz) 11 N = 8 (for TMDS rate (not the pixel rate) up to 45.25 MHz)
4–2 pixel_rep[2:0]	Pixel Repetition This bus controls another factor by which to divide the input frequency (refclk) by the output TMDS rate (ck_ref_mpll_p/m). FTMDS/Fin = (clr_depth[1:0] x pixel_rep[2]) / (pixel_rep[1:0]) 0 **Pixel_rep[2]** Divide by 1. 1 **Pixel_rep[2]**Divide by 2. 00 **Pixel_rep[1:0]**Divide by 4. 01 **Pixel_rep[1:0]**Divide by 2. 1x **Pixel_rep[1:0]**Divide by 1.
clr_dpth[1:0]	Color Depth This bus controls the factor by which to divide the reference clock (PCLK) by the output TMDS rate (ck_ref_mpll_p/m). 00 Divide by 4. 01 Divide by 5.

Table continues on the next page...

HDMI_PHY_CPCE_CTRL field descriptions (continued)

Field	Description
	10 Divide by 6. 11 Divide by 8.

35.7.8 Tx and Clock Measure Control (HDMI_PHY_TXCLKMEASCTRL)

Register name: TXCKMEASCTRL

Access type: Read/write

Address: 0x07

Value at reset: 0x0000

Table 35-14. tx_meas_iv2[7:0]

Effective Bit	Register Value	Description
tx_meas_iv2[0]	00000001	Connect V_{cm_p} (common DC voltage of the positive side of scope) to the analog test bus.
tx_meas_iv2[1]	00000010	Connect V_{cm_m} (common DC voltage of the negative side of scope) to the analog test bus.
tx_meas_iv2[2]	00000100	Connect V_{bg3by4_reg} (output node of tx_vreg_vbgby2 block) to the analog test bus.
tx_meas_iv2[3]	00001000	Connect t_{x_vref} (reference voltage of tx_biasgen block) to the analog test bus.
tx_meas_iv2[4]	00010000	Connect V_{rep_fb} (feedback voltage of the replica circuit of tx_biasgen block) to the analog test bus.
tx_meas_iv2[5]	00100000	In the event that bleed current is too large, this bit can be used to force vb closer to gnd and disable/reduce the bleed current. (This bit is a debug feature, which does not pull vb to gnd properly.)
tx_meas_iv2[6]	01000000	Connect V_p (low power supply) to the analog test bus.
tx_meas_iv2[7]	10000000	Connect V_{cm} (common DC voltage of the scope) to the analog test bus.

Table 35-15. ck_meas_iv[7:0]

Effective Bit	Register Value	Description
ck_meas_iv[0]	00000001	Connect V_{cm_p} (common DC voltage of the positive side of scope) to the analog test bus.
ck_meas_iv[1]	00000010	Connect V_{cm_m} (common DC voltage of the negative side of scope) to the analog test bus.
ck_meas_iv[2]	00000100	Connect V_{bg3by4_reg} (output node of tx_vreg_vbgby2 block) to the analog test bus.

Table continues on the next page...

Table 35-15. ck_meas_iv[7:0] (continued)

ck_meas_iv[3]	00001000	Connect t_{x_vref} (reference voltage of tx_biasgen block) to the analog test bus.
ck_meas_iv[4]	00010000	Connect V_{rep_fb} (feedback voltage of the replica circuit of tx_biasgen block) to the analog test bus.
ck_meas_iv[5]	00100000	In the event that bleed current is too large, this bit can be used to force v_b closer to gnd and disable/reduce the bleed current. (This bit is a debug feature, which does not pull v_b to gnd properly.)
ck_meas_iv[6]	01000000	Connect V_p (low power supply) to the analog test bus.
ck_meas_iv[7]	10000000	Connect V_{cm} (common DC voltage of scope) to the analog test bus.

Address: 0h base + 7h offset = 7h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	tx_meas_iv2[7:0]								ck_meas_iv[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HDMI_PHY_TXCLKMEASCTRL field descriptions

Field	Description
15–8 tx_meas_iv2[7:0]	Transmitter Measure Internal Voltage 2 This bus is used to test specific voltages for third transmitter lane/channel by applying voltages on the atb_sense port based on configured currents, as described in the tx_meas_iv2[7:0] table.
ck_meas_iv[7:0]	Clock Measure Internal Voltage This bus is used to test specific voltages for the clock lane/channel by applying voltages on the atb_sense port based on configured currents, as described in the ck_meas_iv[7:0] table.

35.7.9 Tx Measure Control (HDMI_PHY_TXMEASCTRL)

Register name: TXMEASCTRL

Access type: Read/write

Address: 0x08

Value at reset: 0x0000

Table 35-16. tx_meas_iv1[7:0]

Effective Bit	Register Value	Description
tx_meas_iv1[0]	00000001	Connect V_{cm_p} (common DC voltage of positive side of scope) to the analog test bus.
tx_meas_iv1[1]	00000010	Connect V_{cm_m} (common DC voltage of negative side of scope) to the analog test bus.
tx_meas_iv1[2]	00000100	Connect V_{bg3by4_reg} (output node of tx_vreg_vbgby2 block) to the analog test bus.

Table continues on the next page...

Table 35-16. tx_meas_iv1[7:0] (continued)

tx_meas_iv1[3]	00001000	Connect t_{x_vref} (reference voltage of tx_biasgen block) to the analog test bus.
tx_meas_iv1[4]	00010000	Connect V_{rep_fb} (feedback voltage of replica circuit of tx_biasgen block) to the analog test bus.
tx_meas_iv1[5]	00100000	In case bleed current is too large, this bit can be used to force v_b closer to gnd and disable/reduce the bleed current. (This bit is a debug feature, which does not pull v_b to gnd properly.)
tx_meas_iv1[6]	01000000	Connect V_p (low power supply) to the analog test bus.
tx_meas_iv1[7]	10000000	Connect V_{cm} (common DC voltage of scope) to the analog test bus.

Table 35-17. tx_meas_iv0[7:0]

Effective Bit	Register Value	Description
tx_meas_iv0[0]	00000001	Connect V_{cm_p} (common DC voltage of positive side of scope) to the analog test bus.
tx_meas_iv0[1]	00000010	Connect V_{cm_m} (common DC voltage of negative side of scope) to the analog test bus.
tx_meas_iv0[2]	00000100	Connect V_{bg3by4_reg} (output node of tx_vreg_vbgby2 block) to the analog test bus.
tx_meas_iv0[3]	00001000	Connect t_{x_vref} (reference voltage of tx_biasgen block) to the analog test bus
tx_meas_iv0[4]	00010000	Connect V_{rep_fb} (feedback voltage of replica circuit of tx_biasgen block) to the analog test bus.
tx_meas_iv0[5]	00100000	In the event that bleed current is too large, this bit can be used to force v_b closer to gnd and disable/reduce the bleed current. (This bit is a debug feature, which does not pull v_b to gnd properly.)
tx_meas_iv0[6]	01000000	Connect V_p (low power supply) to the analog test bus.
tx_meas_iv0[7]	10000000	Connect V_{cm} (common DC voltage of scope) to the analog test bus.

Address: 0h base + 8h offset = 8h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	tx_meas_iv0[7:0]								tx_meas_iv1[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HDMI_PHY_TXMEASCTRL field descriptions

Field	Description
15–8 tx_meas_iv0[7:0]	Transmitter Measure Internal Voltage 0 This bus is used to test specific voltages for the first transmitter lane/channel by applying voltages on the atb_sense port based on configured currents, as described in the tx_meas_iv0[7:0] table.
tx_meas_iv1[7:0]	Transmitter Measure Internal Voltage 1 This bus is used to test specific voltages for the second transmitter lane/channel by applying voltages on the atb_sense port based on configured currents, as described in the tx_meas_iv1[7:0] table.

35.7.10 Clock Symbol and Transmitter Control (HDMI_PHY_CKSYMTXCTRL)

Register name: CKSYMTXCTRL

Access type: Read/write/override

Address: 0x09

Value at reset: 0x0009

Address: 0h base + 9h offset = 9h

Bit	15	14	13	12	11	10	9	8
Read Write	Override	Reserved						
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				tx_symon	tx_traon	tx_trbon	ck_symon
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_CKSYMTXCTRL field descriptions

Field	Description
15 Override	Writing a 0 to the override bit causes the register to set the value 0x0F, regardless of the other bit settings. Setting the override bit to 1 causes tx_symon, tx_traon, tx_trbon, and ck_symon bits to work per the register settings. These bits must be set at the same time the override bit is set to 1, or while the override bit remains at 1.
14–4 -	This field is reserved. Reserved
3 tx_symon	Transmitter Symbol On This bit enables the transmitter symbol driver(s), To enable the transmitter driver(s), the tx_pwron bit for each channel must be high. 0 Disable the transmitter symbol driver(s). 1 Enable the transmitter symbol driver(s).
2 tx_traon	Transmitter Trailer A On This bit enables the transmitter trailer A driver(s). To enable the transmitter trailer A driver(s) and to enable pre-emphasis, the tx_pwron bit for each channel must be high. 0 Disable the transmitter trailer A driver(s). 1 Enable the transmitter trailer A driver(s).
1 tx_trbon	Transmitter Trailer B On This bit enables the transmitter trailer B driver(s). To enable the transmitter trailer B driver(s) and to enable pre-emphasis, the tx_pwron bit for each channel must be high. 0 Disable the transmitter trailer B driver(s). 1 Enable the transmitter trailer B driver(s).

Table continues on the next page...

HDMI_PHY_CKSYMTXCTRL field descriptions (continued)

Field	Description
0 ck_symon	Clock Symbol On This bit enables the clock symbol driver. To enable the clock driver, the ck_powon bit must be high. In addition, there is no pre-emphasis enable for the clock driver. 0 Disable the clock symbol driver. 1 Enable the clock symbol driver.

35.7.11 Comparator Sequence Control (HDMI_PHY_CMPSEQCTRL)

Register name: CMPSEQCTRL

Access type: Read/write/override

Address: 0x0A

Value at reset: 0x0000

Address: 0h base + Ah offset = Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Override	Reserved														sup_comp_rt_r
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HDMI_PHY_CMPSEQCTRL field descriptions

Field	Description
15 Override	If the Override bit is set to 1, the working value is the Override bit value, not the registered value.
14–1 -	This field is reserved. Reserved
0 sup_comp_rt_r	Support Comparator Resistance Termination This bit controls the comparator sequence. 0 Latch the first input, if the Override bit is 0. 1 Latch the second input, then set the comparator's output pin by the result of comparison, if the Override bit is 0.

35.7.12 Comparator Power Control (HDMI_PHY_CMPPWRCTRL)

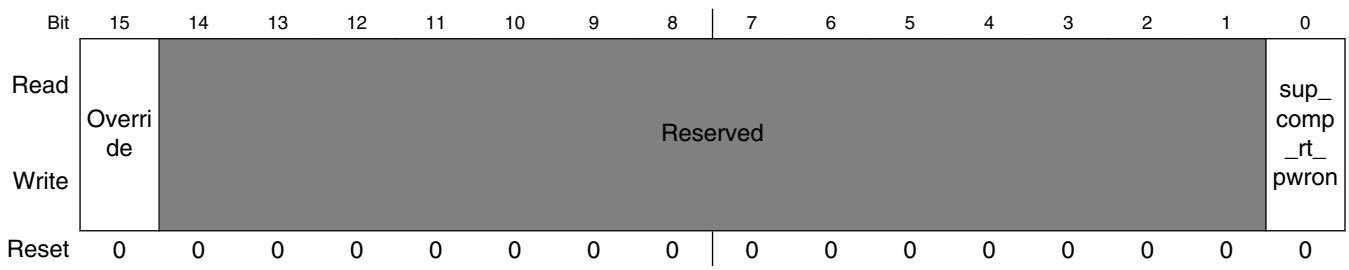
Register name: CMPPWRCTRL

Access type: Read/write/override

Address: 0x0B

Value at reset: 0x0000

Address: 0h base + Bh offset = Bh



HDMI_PHY_CMPPWRCTRL field descriptions

Field	Description
15 Override	If the Override bit is set to 1, the working value is the Override bit value, not the registered value.
14–1 -	This field is reserved. Reserved
0 sup_comp_rt_pwron	Support Comparator Resistance Termination Power-On This bit powers on the Comparator module. 0 Power off the Comparator module and connect the comparator's output to ground, if the Override bit is 0. 1 Power on the Comparator module, if the Override bit is 0.

35.7.13 Comparator Mode Control (HDMI_PHY_CMPMODECTRL)

Register name: CMPMODECTRL

Access type: Read/write/override

Address: 0x0C

Value at reset: 0x0000

Address: 0h base + Ch offset = Ch

Bit	15	14	13	12	11	10	9	8
Read / Write	Override	Reserved						
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read / Write	Reserved							sup_comp_mode
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_CMPMODECTRL field descriptions

Field	Description
15 Override	If the Override bit is set to 1, the working value is the Override bit value, not the registered value.
14–1 -	This field is reserved. Reserved
0 sup_comp_mode	Support Comparator Mode This bit selects the comparator mode. 0 Testing mode (ADC mode) 1 Calibration mode

35.7.14 Measure Control (HDMI_PHY_MEASCTRL)

Register name: MEASCTRL

Access type: Read/write

Address: 0x0D

Value at reset: 0x0000

NOTE

Two or more of the previous register bits must not be set to 1 simultaneously; doing so can lead to a hardware problem.

Address: 0h base + Dh offset = Dh

Bit	15	14	13	12	11	10	9	8
Read / Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read / Write	Reserved				sup_atb_on_rext	sup_por_meas_iv[1:0]		sup_dac_on_atb
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_MEASCTRL field descriptions

Field	Description
15–4 -	This field is reserved. Reserved
3 sup_atb_on_rext	Support Analog Test Bus On External Calibration Resistance This bit connects or disconnects the V_{rext} node to/from the analog test bus. 0 Disconnect the V_{rext} node from the analog test bus. 1 Connect the V_{rext} node to the analog test bus.
2–1 sup_por_meas_iv[1:0]	Support Power Measure Internal Voltage This bus connects or disconnects a single output signal on the analog test bus to measure the voltage of two nodes of the support power block. 01 Connect V_{be} (Bipolar transistor voltage) to the analog test bus. 10 Connect V_{bg} (Band-gap voltage) to the analog test bus.
0 sup_dac_on_atb	Support Digital-to-Analog Converter On Analog Test Bus This bit connects or disconnects the DAC's output on the analog test bus to test the performance of the DAC through Integral Non-Linearity (INL) and Differential Non-Linearity (DNL). 0 Disconnect the DAC's output from the analog test bus. 1 Connect the DAC's output to the analog test bus.

35.7.15 Voltage Level Control (HDMI_PHY_VLEVCTRL)

Register name: VLEVCTRL

Access type: Read/write

Address: 0x0E

Value at reset: 0x0000

Address: 0h base + Eh offset = Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	Reserved						sup_tx_lvl[4:0]					sup_ck_lvl[4:0]				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HDMI_PHY_VLEVCTRL field descriptions

Field	Description
15–10 -	This field is reserved. Reserved
9–5 sup_tx_lvl[4:0]	Support Transmitter Level This bus controls the reference voltage level of the three transmitter channel modules. This voltage reference has a direct relation with the output signal voltage level. For more information about the driver voltage level configuration, see Appendix A: Driver Voltage Level Configuration.

Table continues on the next page...

HDMI_PHY_VLEVCTRL field descriptions (continued)

Field	Description
sup_ck_lvl[4:0]	Support Clock Level This bus controls the reference voltage level of the Clock Channel module. This voltage reference has a direct relation with the output signal voltage level. For more information about the driver voltage level configuration, see Appendix A: Driver Voltage Level Configuration.

35.7.16 Digital-to-Analog Control (HDMI_PHY_D2ACTRL)

Register name: D2ACTRL

Access type: Read/write/override

Address: 0x0F

Value at reset: 0x0000

Address: 0h base + Fh offset = Fh

Bit	15	14	13	12	11	10	9	8
Read Write	Override	Reserved				sup_dac_n[7:0]		
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	sup_dac_n[7:0]					sup_dac_th_n[2:0]		
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_D2ACTRL field descriptions

Field	Description
15 Override	If the Override bit is set to 1, the working value is the Override bit value, not the registered value.
14–11 -	This field is reserved. Reserved
10–3 sup_dac_n[7:0]	Support Analog-to-Digital Inverted This bus represents the LSB's of the 9-bit DAC value.
sup_dac_th_n[2:0]	Support Digital-to-Analog Thermometer Inverted This bus value is inverted and defined in the thermometer code to represent the binary code of the two MSB's of the 9-bit DAC value. **Note**: To increase the stability of the DAC block, the two MSB's of the 9-bit DAC value are represented in thermometer code, not in binary code. The MSB's of the 9-bit DAC value is split into two thermometer-code bits. A transition from 0 to 1 of the 9-bit DAC value's MSB is represented by "00" -> "01" -> "11" in thermometer code.

35.7.17 Current Control (HDMI_PHY_CURRCTRL)

Register name: CURRCTRL

Access type: Read/write

Address: 0x10

Value at reset: 0x08AB

Address: 0h base + 10h offset = 10h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	Reserved				mpll_prop_cntrl[2:0]			mpll_int_cntrl[2:0]			pll_prop_cntrl[2:0]			pll_int_cntrl[2:0]		
Reset	0	0	0	0	1	0	0	0	1	0	1	0	1	0	1	1

HDMI_PHY_CURRCTRL field descriptions

Field	Description
15–12 -	This field is reserved. Reserved
11–9 mpll_prop_ cntrl[2:0]	MPLL Proportional Control This bus controls the MPLL charge pump proportional current. Eight levels of charge pump proportional current value are possible. The specific values are defined in PLL/MPLL Generic Configuration Settings. Default (reset) value of pll_prop_cntrl[1:0] is 100.
8–6 mpll_int_cntrl[2:0]	MPLL Integral Control This bus controls the charge pump integral current. Eight levels of charge pump integral current value are possible. The specific values are defined in PLL/MPLL Generic Configuration Settings. Default (reset) value of pll_int_cntrl[1:0] is 100.
5–3 pll_prop_ cntrl[2:0]	PLL Proportional Control This bus controls the PLL charge pump proportional current. Eight levels of charge pump proportional current value are possible. The specific values are defined in PLL/MPLL Generic Configuration Settings. Default (reset) value of pll_int_cntrl is 011.
pll_int_cntrl[2:0]	PLL Charge Pump Integral Control This bus controls the PLL charge pump integral current. Eight levels of charge pump integral current value are possible. The specific values are defined in PLL/MPLL Generic Configuration Settings. Default (reset) value of pll_int_cntrl[2:0] is 100.

35.7.18 Drive Analog Control (HDMI_PHY_DRVANACTRL)

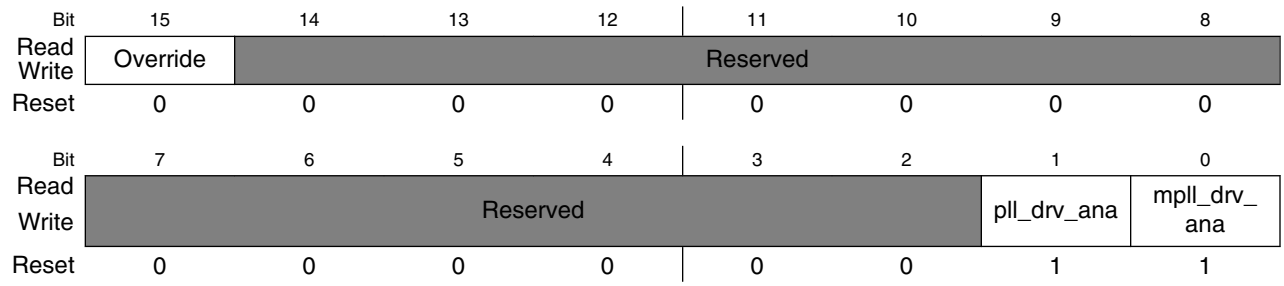
Register name: DRVANACTRL

Access type: Read/write/override

Address: 0x11

Value at reset: 0x0003

Address: 0h base + 11h offset = 11h



HDMI_PHY_DRVANACTRL field descriptions

Field	Description
15 Override	If the Override bit is set to 1, the working value is the Override bit value, not the registered value.
14–2 -	This field is reserved. Reserved
1 pll_drv_ana	PLL Drive Analog This bit enables or disables driving the ck_ref_mpll_p/m clocks to the MPLL module. Default (reset) value of pll_drv_ana is 1. 0 Set ck_ref_mpll low, if the Override bit is 0. 1 Enable 25-340 MHz output clock (ck_ref_mpll_p/m) to be driven to MPLL, if the Override bit is 0.
0 mpll_drv_ana	MPLL Drive Analog This bit enables or disables driving the ck_ref_p/m clocks to all transmitters (tx_topa). Default (reset) value of mpll_drv_ana is 1. 0 Set ck_ref_p high; ck_ref_m low, if the Override bit is 0. 1 Enable ck_ref_mpll output clock (125-1700 MHz ck_ref_p/m) to be driven to all transmitters (tx_topa), if the Override bit is 0.

35.7.19 PLL Measure Control (HDMI_PHY_PLLMEASCTRL)

Register name: PLLMEASCTRL

Access type: Read/write

Address: 0x12

Value at reset: 0x0000

NOTE

With the exception of pll_atb_sense_sel and pll_meas_iv[9], two or more of the previous register bits must not be set to 1 simultaneously; doing so can lead to a hardware problem.

The table below describes the pll_meas_iv[10:0] bit settings.

Table 35-18. pll_meas_iv[10:0]

Effective Bit	Pll_meas_iv[10:0] Value	Description
Pll_meas_iv[0]	00000000001	Not used.
Pll_meas_iv[1]	00000000010	Not used in pll_top. Connected to fast_tech pins of PLL and MPLL in hdmi_topa.
Pll_meas_iv[2]	00000000100	Connects VP supply voltage to the atb_sense line.
Pll_meas_iv[3]	00000001000	Connects vp_cp to the atb_sense line.
Pll_meas_iv[4]	00000010000	Connects internal supply voltage of the VCO (ivco) to the atb_sense line.
Pll_meas_iv[5]	00000100000	Connects vp_cko voltage to the atb_sense line.
Pll_meas_iv[6]	00001000000	Connects node vpsf to the atb_sense line.
Pll_meas_iv[7]	00010000000	Connects vref to atb_sense line.
Pll_meas_iv[8]	00100000000	Connects vcntrl to atb_sense line.
Pll_meas_iv[9]	01000000000	Enables the phase mixer and pll_cko_pm_p/m.
Pll_meas_iv[10]	10000000000	Measures the voltage corresponding to the output phase of the clr_dpth divider (fb_clk) with respect to refclk.

Address: 0h base + 12h offset = 12h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved			pll_atb_ sense_sel	pll_meas_iv[10:0]			
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	pll_meas_iv[10:0]							pll_meas_gd
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_PLLMEASCTRL field descriptions

Field	Description
15–13 -	This field is reserved. Reserved
12 pll_atb_sense_ sel	PLL Analog Test Bus Sense Select This bit enables or disables internal signals of the PLL to be connected to the analog test bus. Without setting this bit, no measurements can be made on the atb_sense line. Default (reset) value of pll_atb_sense_sel is 0. 0 Disable the ability to measure internal DC signals on the atb_sense line in the PLL. 1 Enable the ability to measure internal DC signals on the atb_sense line in the PLL.

Table continues on the next page...

HDMI_PHY_PLLMEASCTRL field descriptions (continued)

Field	Description
11–1 pll_meas_iv[10:0]	PLL Measure Internal Voltage This bus enables or disables measuring various PLL node voltages and branch currents. For information about the bit settings, see the pll_meas_iv[10:0] table.
0 pll_meas_gd	PLL Measure Ground This bit connects or disconnects the ground signal to the atb_sense (analog test bus) bus. 0 Disconnect the ground signal from the atb_sense bus. 1 Connect the ground signal to the atb_sense bus.

35.7.20 PLL Phase and Bypass Control (HDMI_PHY_PLLPHBYCTRL)

Register name: PLLPHBYCTRL

Access type: Read/write

Address: 0x13

Value at reset: 0x0000

Address: 0h base + 13h offset = 13h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Reserved				bypass_ppll	pll_ph_sel_ck	pll_ph_sel[9:0]									
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HDMI_PHY_PLLPHBYCTRL field descriptions

Field	Description
15–12 -	This field is reserved. Reserved
11 bypass_ppll	Bypass Pre-PLL This bit enables or disables bypassing the pre-PLL. 0 Disable bypassing pll_top by forcing refclk_mpll_ref low. 1 Enable bypassing pll_top by buffering refclk to refclk_mpll_ref.
10 pll_ph_sel_ck	PLL Phase Select Clock This bit enables or disables latching the ph_sel[9:0] into a 9-bit DAC used in the phase mixer.

Table continues on the next page...

HDMI_PHY_PLLPHBYCTRL field descriptions (continued)

Field	Description
	0 Disable latching the pll_ph_sel[9:0] into the 9-bit DAC. 1 Enable latching the pll_ph_sel[9:0] into the 9-bit DAC.
pll_ph_sel[9:0]	PLL Phase Select This bus is a control word for the PLL's phase mixer that enables the phase of pll_cko_pm_p/m to be varied ± 0.5 UI of the VCO frequency, which is 740-1,480 MHz.

35.7.21 Gear Shift, Reset Mode, and Power State Control (HDMI_PHY_GRP_CTRL)

Register name: -

Access type: Read/write/override

Address: 0x14

Value at reset: 0x0000

Address: 0h base + 14h offset = 14h

Bit	15	14	13	12	11	10	9	8
Read Write	Override	Reserved						
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved		pll_pwr_on	pll_rst	mpll_pwr_on	mpll_rst	pll_gear_shift	mpll_gear_shift
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_GRP_CTRL field descriptions

Field	Description
15 Override	If the Override bit is set to 1, the working value is the Override bit value, not the registered value.
14–6 -	This field is reserved. Reserved
5 pll_pwr_on	PLL Power-On This bit is used to power on/off the PLL module. Note: If the Override bit is set to 1, the working value is the Override bit value, not the registered value. 0 Power off PLL and draw minimal current, if the Override bit is 0. 1 Power on PLL and enable it to operate normally, if the Override bit is 0.
4 pll_rst	PLL Reset This bit is used to place the MPLL in Reset mode.

Table continues on the next page...

HDMI_PHY_GRP_CTRL field descriptions (continued)

Field	Description
	0 Enable PLL to operate normally, if the Override bit is 0. 1 Place the PLL in Reset mode, if the Override bit is 0.
3 mpll_pwr_on	MPLL Power-On This bit is used to power-on/off the MPLL module. 0 Power on MPLL and set all output clocks to DC levels, if the Override bit is 0. 1 Power off MPLL and enable it to operate normally, if the Override bit is 0.
2 mpll_rst	MPLL Reset This bit is used to place the MPLL in Reset mode. 0 Enable MPLL to operate normally, if the Override bit is 0. 1 Place the MPLL in Reset mode, if the Override bit is 0.
1 pll_gear_shift	PLL Gear Shift This bit enables or disables Rapid Locking mode, where the pll_gear_shift bit is asserted for 25 µs when coming out of reset, then deasserted before clocks are valid. 0 Disable Rapid Locking mode, if the Override bit is 0. 1 Enable Rapid Locking mode, if the Override bit is 0.
0 mpll_gear_shift	MPLL Gear Shift This bit enables or disables Rapid Locking mode, where the mpll_gear_shift bit is asserted for 25 µs when coming out of reset, then deasserted before clocks are valid. 0 Disable Rapid Locking mode, if the Override bit is 0. 1 Enable Rapid Locking mode, if the Override bit is 0.

35.7.22 Gmp Control (HDMI_PHY_GMPCTRL)

Register name: GMPCTRL

Access type: Read/write

Address: 0x15

Value at reset: 0x0000

Address: 0h base + 15h offset = 15h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	Reserved												pll_gmp_cntrl[1:0]		mpll_gmp_cntrl[1:0]	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HDMI_PHY_GMPCTRL field descriptions

Field	Description
15–4 -	This field is reserved. Reserved
3–2 pll_gmp_cntrl[1:0]	PLL gmp Control This bus controls the effective loop-filter resistance (equal) to increase or decrease PLL bandwidth and to compensate for changes in the Divider module (n_cntrl). 00 TMDS rate up to 45.25 MHz 01 TMDS rate of 45.26-92.5 MHz 10 TMDS rate of 92.51-185 MHz 11 TMDS rate of 184.1-370 MHz
mpll_gmp_ cntrl[1:0]	MPLL gmp Control This bus controls the effective loop-filter resistance (= 1/gmp) to increase or decrease MPLL bandwidth and to compensate for changes in the Divider module (n_cntrl). 00 TMDS rate up to 45.25 MHz 01 TMDS rate of 45.26-92.5 MHz 10 TMDS rate of 92.51-185 MHz 11 TMDS rate of 184.1-370 MHz

35.7.23 MPLL Measure Control (HDMI_PHY_MPLLMEASCTRL)

Register name: MPLLMEASCTRL

Access type: Read/write

Address: 0x16

Value at reset: 0x0000

NOTE

With the exception of mpll_atb_sense_sel, mpll_meas_iv[9], and mpll_meas_iv[11], two or more of the previous register bits must not be set to 1 simultaneously, because doing so can lead to a hardware problem.

The table below describes the mpll_meas_iv[11:0] bit settings.

Table 35-19. mpll_meas_iv[11:0]

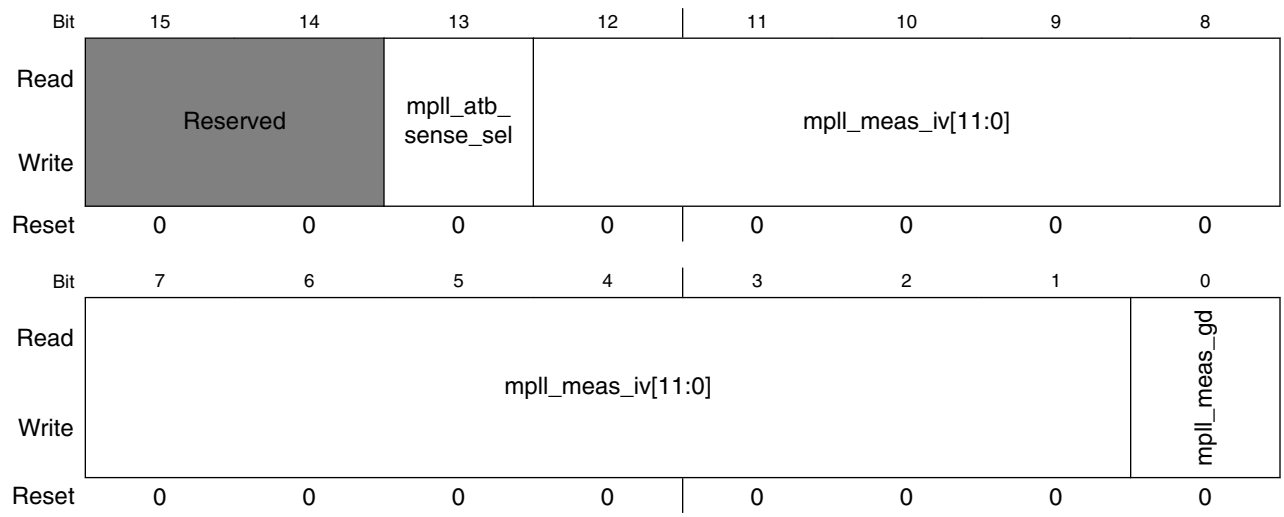
Effective Bit	mpll_meas_iv[11:0] Value	Description
mpll_meas_iv[0]	000000000001	Connects the internal positive DCC control line (vc0p) to the atb_sense line.
mpll_meas_iv[1]	000000000010	Connects the internal negative DCC control line (vc0m) to the atb_sense line.

Table continues on the next page...

Table 35-19. mpll_meas_iv[11:0] (continued)

Effective Bit	mpll_meas_iv[11:0] Value	Description
mpll_meas_iv[2]	000000000100	Connects VP supply voltage to the atb_sense line.
mpll_meas_iv[3]	000000001000	Connects vp_cp to the atb_sense line.
mpll_meas_iv[4]	000000010000	Connects the internal supply voltage of the VCO (ivco) to the atb_sense line through a low-pass filter.
mpll_meas_iv[5]	000000100000	Connects the vp_cko voltage to the atb_sense line.
mpll_meas_iv[6]	000001000000	Connects the vpsf node to the atb_sense line.
mpll_meas_iv[7]	000010000000	Connects vref to the atb_sense line.
mpll_meas_iv[8]	000100000000	Connects vcntrl to the atb_sense line.
mpll_meas_iv[9]	001000000000	Enables the phase mixer and cko_pm_p/m.
mpll_meas_iv[10]	010000000000	Measures the voltage corresponding to the phase of the divide-by-5 output (fb_clk) with respect to the phase of the mpll_ana input (ck_refclk_p).
mpll_meas_iv[11]	100000000000	Forces div_x1 low; therefore, bitclk_p/m is multiplexed to ck_ref_p/m (normally, if n_cntrl = 00, ck0_p/m is multiplexed to ck_ref_p/m and DCC loop is enabled.

Address: 0h base + 16h offset = 16h



HDMI_PHY_MPLLMEASCTRL field descriptions

Field	Description
15–14 -	This field is reserved. Reserved
13 mpll_atb_sense_ sel	MPLL Analog Test Bus Sense Select This bit enables or disables internal signals of the PLL to be connected to the analog test bus. Without setting this bit, no measurements can be made on the atb_sense line. Default (reset) value of mpll_atb_sense_sel is 0. 0 Disable the ability to measure internal DC signals on the atb_sense line in the PLL. 1 Enable the ability to measure internal DC signals on the atb_sense line in the PLL.

Table continues on the next page...

HDMI_PHY_MPLLMEASCTRL field descriptions (continued)

Field	Description
12–1 mpll_meas_iv[11:0]	MPLL Measure Internal Voltage This bus enables or disables measuring various PLL node voltages and branch currents. For information about the bit settings, see the mpll_meas_iv[11:0] table.
0 mpll_meas_gd	MPLL Measure Ground This bit connects or disconnects the ground signal to the atb_sense (analog test bus) bus. 0 Disconnect the ground signal from the atb_sense bus. 1 Connect the ground signal to the atb_sense bus.

35.7.24 MPLL and PLL Phase, Scope Clock Select, and MUX Clock Control (HDMI_PHY_MSM_CTRL)

Register name: -

Access type: Read/write

Address: 0x17

Value at reset: 0x0000

The table below describes the cko_sel[1:0] bit settings. (The cko_sel[1:0] default value is 00.)

Table 35-20. cko_sel[1:0]

cko_sel[1:0]	TMDS Clock Mode	PLL and MPLL Mode	Output TMDS Clock on TMDS Channel
00	Non-Coherent	Normal	ck_ref_mpll_p/m (Pre-PLL out)
01	Off (PLL and MPLL are both off)	Normal	Off (No output TMDS clock)
10	Test (PHY in test mode)	Normal	PCLK (Input reference clock to HDMI 3D Tx PHY)
11	Coherent	Normal	fb_clk (MPLL feedback clock)
00	Off (PLL and MPLL are both off)	Bypass	Off (No output TMDS clock)
01	Off (PLL and MPLL are both off)	Bypass	Off (No output TMDS clock)
10	Non-Coherent	Bypass	PCLK (Input reference clock to HDMI 3D Tx PHY)
11	Coherent	Bypass	fb_clk (MPLL feedback clock)

Address: 0h base + 17h offset = 17h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Reserved		mpll_ph_sel_ck	mpll_ph_sel[9:0]										cko_sel[1:0]		scope_ck_sel
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HDMI_PHY_MSM_CTRL field descriptions

Field	Description
15–14 -	This field is reserved. Reserved
13 mpll_ph_sel_ck	MPLL Phase Select Clock This bit enables or disables latching ph_sel[9:0] into a 9-bit DAC used in the phase mixer. 0 Disable latching mpll_ph_sel[9:0] into the 9-bit DAC. 1 Enable latching mpll_ph_sel[9:0] into the 9-bit DAC.
12–3 mpll_ph_sel[9:0]	MPLL Phase Select This bus is a control word for the MPLL's phase mixer and enables the phase of pll_cko_pm_p /m to be varied ± 0.5 UI of the VCO frequency, which has a range of 925-1,850 MHz.
2–1 cko_sel[1:0]	Clock Output Select This bus selects the clock to be connected to the output TMDS clock channel. **Notes:** • Normal mode: The color depth or pixel repetition is required, which means that the PLL is powered on (pll_pwr_on bit is set to 1) and the bypass_ppll bit is set to 0. • Bypass mode: The color depth and the pixel repetition is not required, which means that the PLL is powered off (pll_pwr_on bit is set to 0) and the bypass_ppll bit is set to 1. For information about the cko_sel[1:0] bit settings and corresponding PLL/MPLL modes, see the cko_sel[1:0] table.
0 scope_ck_sel	Scope Clock Select Selects the clock to connect to the scope clock signal: the differential pll_cko_p/m or the differential mpll_cko_p/m. 0 Connect the mpll_cko_pm_p/m to the clock scope (this clock has a range of 23.125-45.25 MHz) where the mpll_meas_iv[9] bit must be high. 1 Connect the pll_cko_pm_p/m to the clock scope (this clock has a range of 23.125-45.25 MHz) where the pll_meas_iv[9] bit must be high.

35.7.25 Scope, Comparator Result and Power Bad Status (HDMI_PHY_SCRPB_STATUS)

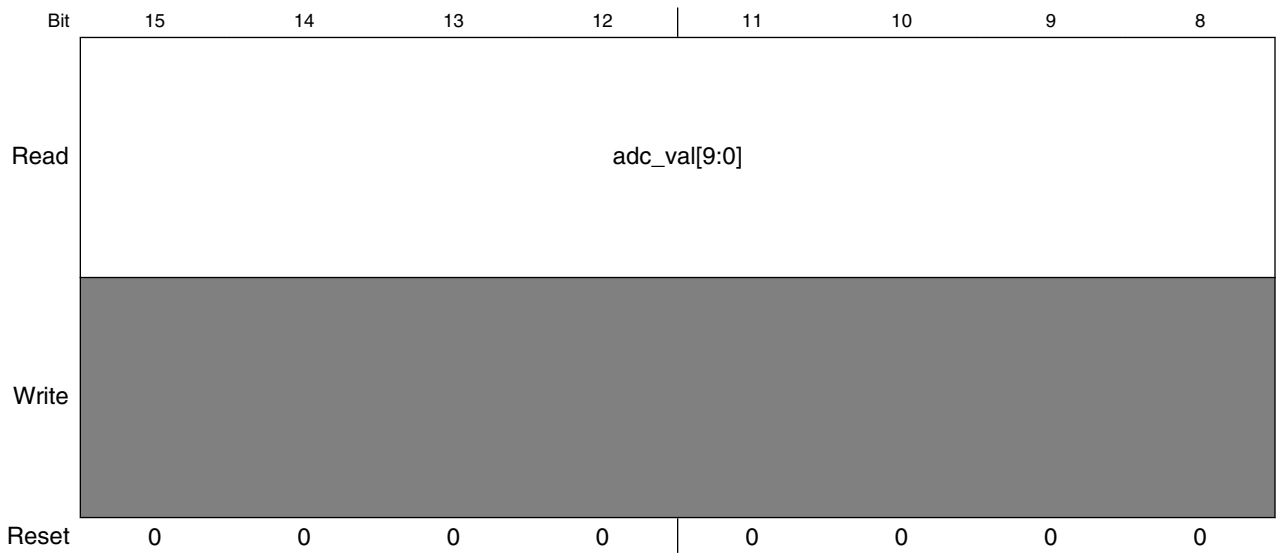
Register name: -

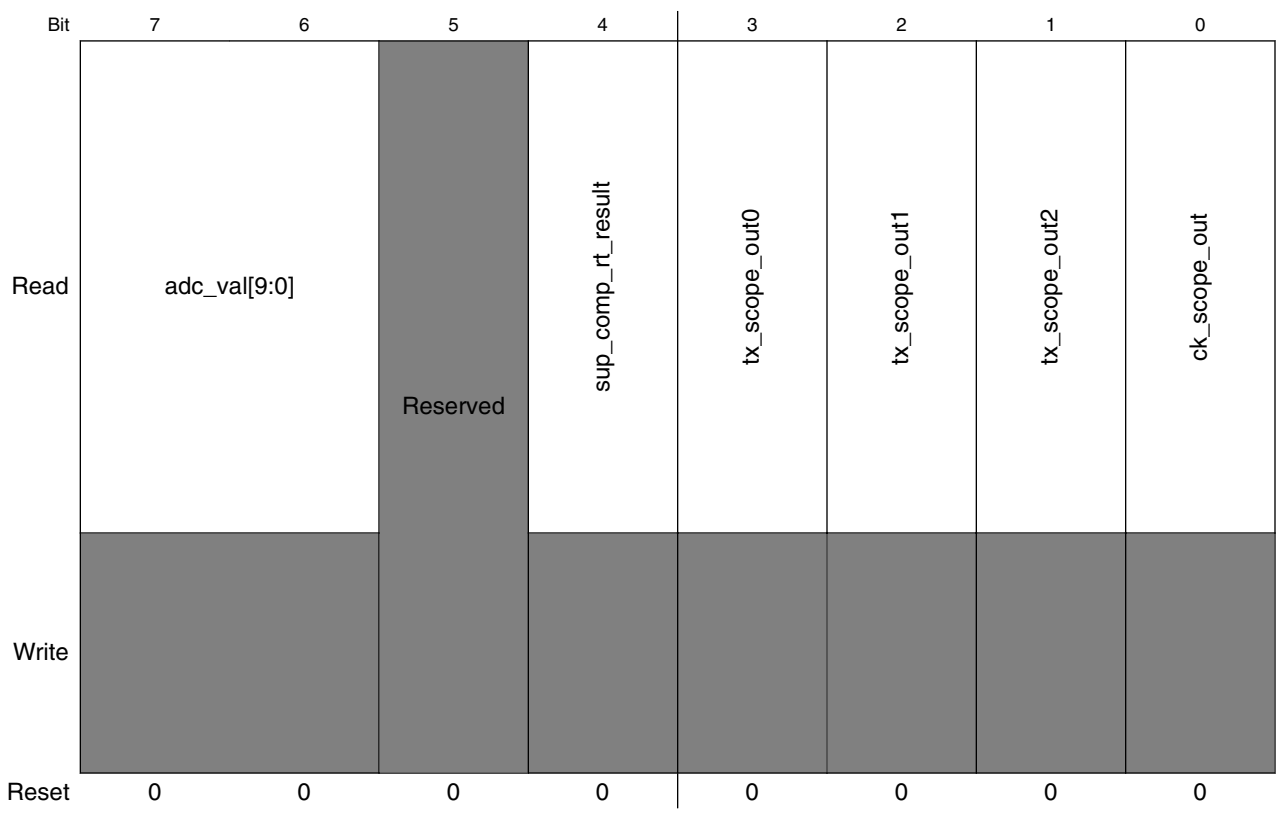
Access type: Read-only

Address: 0x18

Value at reset: N/A

Address: 0h base + 18h offset = 18h





HDMI_PHY_SCRPB_STATUS field descriptions

Field	Description
15–6 adc_val[9:0]	ADC/DAC bit word (analog signal value)
5 -	This field is reserved. Reserved
4 sup_comp_rt_result	Support Comparator Resistance Termination Result This bit represents the result of the comparison process. 0 The first input is greater than or equal to the second input. 1 The first input is less than the second input.
3 tx_scope_out0	Scoping Value for Lane 0 0 The driver output of lane 0 is not differential. 1 The driver output of lane 0 is differential.
2 tx_scope_out1	Scoping Value for Lane 1 0 The driver output of lane 1 is not differential. 1 The driver output of lane 1 is differential.
1 tx_scope_out2	Scoping Value for Lane 2 0 The driver output of lane 2 is not differential. 1 The driver output of lane 2 is differential.
0 ck_scope_out	Clock Scope Output Signal 0 The clock driver output is not differential. 1 The clock driver output is differential.

35.7.26 Transmission Termination (HDMI_PHY_TXTERM)

Register name: TXTERM

Access type: Read/write

Address: 0x19

Value at reset: 0x0007

Address: 0h base + 19h offset = 19h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	Reserved													d_tx_term[2:0]		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

HDMI_PHY_TXTERM field descriptions

Field	Description
15–3 -	This field is reserved. Reserved
d_tx_term[2:0]	Digital Transmission Termination This bus defines the transmission termination (resistance) value, which is set by the HDMI controller. The formula for the resistance value is: R = 50 / (1 - 0.125 x d_tx_term) This equation is valid only when d_tx_term equals 0-6. • 000: 50 Ω • 001: 56.14 Ω • 010: 66.67 Ω • 011: 80 Ω • 100: 100 Ω • 101: 133.33 Ω • 110: 200 Ω • 111: Open circuit

35.7.27 Power Sequence, TX Clock Alignment, Resistance Calibration, Pattern Generator Skip Bit, and TMDS Encoder Enable (HDMI_PHY_PTRPT_ENBL)

Register name: -

Access type: Read/write/override

Address: 0x1A

Value at reset: 0x0000

Address: 0h base + 1Ah offset = 1Ah

Bit	15	14	13	12	11	10	9	8
Read	Override	Reserved						pg_skip_bit2
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	pg_skip_bit1	pg_skip_bit0	ck_ref_enb	rcal_enb	tx_ck_align_enb	tx_ready	cko_word_enb	refclk_enb
Write								
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_PTRPT_ENBL field descriptions

Field	Description
15 Override	If the Override bit is set to 1, the working value is the Override bit value, not the registered value.
14–9 -	This field is reserved. Reserved
8 pg_skip_bit2	Pattern Generator Skip Bit 2 This bit enables or disables pattern generator skip bit feature for channel 2. 0 Disable the pattern generator skip bit feature for the third transmitting channel, if the Override bit is 0. 1 Enable the pattern generator skip bit feature for the third transmitting channel, if the Override bit is 0.
7 pg_skip_bit1	Pattern Generator Skip Bit 1 This bit enables or disables pattern generator skip bit feature for channel 1. 0 Disable the pattern generator skip bit feature for the second transmitting channel, if the Override bit is 0. 1 Enable the pattern generator skip bit feature for the second transmitting channel, if the Override bit is 0.
6 pg_skip_bit0	Pattern Generator Skip Bit 0 This bit enables or disables pattern generator skip bit feature for channel 0. 0 Disable the pattern generator skip bit feature for the first transmitting channel, if the Override bit is 0. 1 Enable the pattern generator skip bit feature for the first transmitting channel, if the Override bit is 0.
5 ck_ref_enb	Clock Reference Enable This bit powers up the clock alignment and the resistance calibration modules. 0 Powers down the clock alignment and the resistance calibration modules, if the Override bit is 0. 1 Powers up the clock alignment and the resistance calibration modules, if the Override bit is 0.
4 rcal_enb	Resistance Calibration Enable This bit enables or disables the resistance clock alignment process. 0 Disable the resistance calibration FSM, if the Override bit is 0. 1 Enable the resistance calibration FSM, if the Override bit is 0.
3 tx_ck_align_enb	Transmission Clock Alignment Enable This bit disables or enables the clock alignment FSM. 0 Disable transmission clock alignment FSM, if the Override bit is 0. 1 Enable transmission clock alignment FSM, if the Override bit is 0.
2 tx_ready	Transmission Ready This bit indicates whether the PHY transmit driver is ready to transmit data. 0 PHY transmit driver is not ready to transmit data, if the Override bit is 0. 1 PHY transmit driver is ready to transmit data, if the Override bit is 0.
1 cko_word_enb	Output Clock Word Enable This bit enables the output word clock. 0 Disable the output clock word, if the Override bit is 0. 1 Enable the output clock word, if the Override bit is 0

Table continues on the next page...

HDMI_PHY_PTRPT_ENBL field descriptions (continued)

Field	Description
0 refclk_enb	Reference Clock Enable This bit enables or disables the input reference clock. 0 Disable the input reference clock, if the Override bit is 0. 1 Enable the input reference clock, if the Override bit is 0.

35.7.28 Pattern Generator Mode (HDMI_PHY_PATTERNGEN)

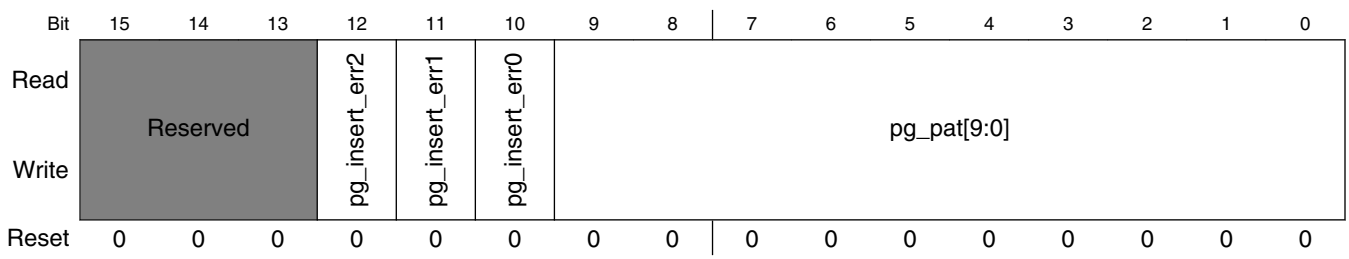
Register name: PATTERNGEN

Access type: Read/write

Address: 0x1B

Value at reset: 0x0000

Address: 0h base + 1Bh offset = 1Bh



HDMI_PHY_PATTERNGEN field descriptions

Field	Description
15–13 -	This field is reserved. Reserved
12 pg_insert_err2	Pattern Generator Insert Error Two This bit enables or disables error insertion inside the generated pattern for channel 2. 0 Do not insert error inside generated pattern for the third transmit channel. 1 Insert error inside generated pattern for the third transmit channel.
11 pg_insert_err1	Pattern Generator Insert Error One This bit enables or disables error insertion inside the generated pattern for channel 1. 0 Do not insert error inside generated pattern for the second transmit channel. 1 Insert error inside generated pattern for the second transmit channel.
10 pg_insert_err0	Pattern Generator Insert Error Zero This bit enables or disables error insertion inside the generated pattern for channel 0.

Table continues on the next page...

HDMI_PHY_PATTERNGEN field descriptions (continued)

Field	Description
	0 Do not insert error inside generated pattern for the first transmit channel. 1 Insert error inside generated pattern for the first transmit channel.
pg_pat[9:0]	Pattern Generator Generated Pattern This bus carries the generated pattern from the Pattern Generator module.

35.7.29 The Soft-Reset and DAC Enable, Clock Alignment and PG Mode (HDMI_PHY_SDCAP_MODE)

Register name: -

Access type: Read/write

Address: 0x1C

Value at reset: 0x0000

Address: 0h base + 1Ch offset = 1Ch

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved				pg_mode2[2:0]			pg_ mode1[2:0]
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	pg_mode1[2:0]		pg_mode0[2:0]			tx_ck_align_ mode	adc_enb	soft_reset
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_SDCAP_MODE field descriptions

Field	Description
15–12 -	This field is reserved. Reserved
11–9 pg_mode2[2:0]	Pattern Generator Mode 2 This bus is used to select the mode of the Pattern Generator module for channel 2. 000 Disable the Pattern Generator module. 001 Enable the Pattern Generator module and generate a sequence of patterns for the third channel using LFSR 15 equation (for example, x15 + x14 + 1). 010 Enable the Pattern Generator module and generate a sequence of patterns for the third channel using LFSR 7 equation (for example, x7 + x6 + 1). 011 Enable the Pattern Generator module and generate a sequence of patterns for the third channel with the fixed word, which is saved inside the pg_pat[9:0] field of the PATTERNGEN register. 100 Enable the Pattern Generator module and generate a sequence of patterns for the third channel with DC-balanced word (for example, 0's or 1's and their inverted values).

Table continues on the next page...

HDMI_PHY_SDCAP_MODE field descriptions (continued)

Field	Description
	101 Enable the Pattern Generator module and generate a sequence of patterns for the third channel with fixed pattern using word of 0's word, 1's word, random word, the saved pattern inside the pg_pat[9:0] field of the PATTERNGEN register, and the inverted pg_pat[9:0] values (for example, 9 bits of 0's, 9 bits of 1's, 3FF, pg_pat[9:0], ~ pg_pat[9:0]). 110 Reserved 111 Reserved
8–6 pg_mode1[2:0]	Pattern Generator Mode 1 This bus is used to select the mode of the Pattern Generator module for channel 1. 000 Disable the Pattern Generator module. 001 Enable the Pattern Generator module and generate a sequence of patterns for the second channel using LFSR 15 equation (for example, x15 + x14 + 1). 010 second the Pattern Generator module and generate a sequence of patterns for the first channel using LFSR 7 equation (for example, x7 + x6 + 1). 011 Enable the Pattern Generator module and generate a sequence of patterns for the second channel with the fixed word, which is saved inside the pg_pat[9:0] field of the PATTERNGEN register. 100 Enable the Pattern Generator module and generate a sequence of patterns for the second channel with DC-balanced word (for example, 0's or 1's and their inverted values). 101 Enable the Pattern Generator module and generate a sequence of patterns for the second channel with fixed pattern using word of 0's word, 1's word, random word, the saved pattern inside the pg_pat[9:0] field of the PATTERNGEN register, and the inverted pg_pat[9:0] values (for example, 9 bits of 0's, 9 bits of 1's, 3FF, pg_pat[9:0], ~ pg_pat[9:0]). 110 Reserved 111 Reserved
5–3 pg_mode0[2:0]	Pattern Generator Mode 0 This bus is used to select the mode of the Pattern Generator module for channel 0. 000 Disable the Pattern Generator module. 001 Enable the Pattern Generator module and generate a sequence of patterns for the first channel using LFSR 15 equation (for example, x15 + x14 + 1). 010 Enable the Pattern Generator module and generate a sequence of patterns for the first channel using LFSR 7 equation (for example, x7 + x6 + 1). 011 Enable the Pattern Generator module and generate a sequence of patterns for the first channel with the fixed word, which is saved inside the pg_pat[9:0] field of the PATTERNGEN register. 100 Enable the Pattern Generator module and generate a sequence of patterns for the first channel with DC-balanced word (for example, 0's or 1's and their inverted values). 101 Enable the Pattern Generator module and generate a sequence of patterns for the first channel with fixed pattern using word of 0's word, 1's word, random word, the saved pattern inside the pg_pat[9:0] field of the PATTERNGEN register, and the inverted pg_pat[9:0] values (for example, 9 bits of 0's, 9 bits of 1's, 3FF, pg_pat[9:0], ~ pg_pat[9:0]). 110 Reserved 111 Reserved
2 tx_ck_align_mode	Transmission Clock Alignment Mode This bit selects the Transmission Clock Alignment mode. 0 Align the three lanes based on lane 1. 1 Align each of the three lanes separately in the following order: lane 2, lane 0, then lane 1.
1 adc_enb	Analog-to-Digital Converter Enable This bit enables or disables the analog-to-digital converter.

Table continues on the next page...

HDMI_PHY_SDCAP_MODE field descriptions (continued)

Field	Description
	0 Disable the analog-to-digital converter. 1 Enable the analog-to-digital converter, which is used in the Resistance Calibration module.
0 soft_reset	Soft Reset This bit enables or disables the soft-reset feature. 0 Do not perform a soft reset. 1 Perform a soft reset by resetting all the system FSMs except the I^2C and Control Register modules.

35.7.30 Scope Mode register (HDMI_PHY_SCOPEMODE)

Register name: SCOPEMODE

Access type: Read/write

Address: 0x1D

Value at reset: 0x0000

Address: 0h base + 1Dh offset = 1Dh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	Reserved		ck_scope_enb	scope_enb2	scope_enb1	scope_enb0	scope_sample_cnt[9:0]									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HDMI_PHY_SCOPEMODE field descriptions

Field	Description
15–14 -	This field is reserved. Reserved
13 ck_scope_enb	Clock Scope Enable This bit enables or disables the tracing of 1's on the clock channel. 0 Disable the tracing of 1's on the clock. 1 Enable the tracing of 1's on the clock.
12 scope_enb2	Scope Enable 2 This bit enables or disables the tracing of 1's on channel 2. 0 Disable the tracing of 1's on the third channel. 1 Enable the tracing of 1's on the third channel.
11 scope_enb1	Scope Enable 1 This bit enables or disables the tracing of 1's on channel 1.

Table continues on the next page...

HDMI_PHY_SCOPEMODE field descriptions (continued)

Field	Description
	0 Disable the tracing of 1's on the second channel. 1 Enable the tracing of 1's on the second channel.
10 scope_enb0	Scope Enable 0 This bit enables or disables the tracing of 1's on channel 0. 0 Disable the tracing of 1's on the first channel. 1 Enable the tracing of 1's on the first channel.
scope_sample_cnt[9:0]	Scope Sample Counter Indicates the number of samples that will be counted (should be multiple of the LFSR length). This count includes only the LSB bits; if the LFSR15 was used, you must program the new bits under the 0x24 register.

35.7.31 Digital Transmission Mode (HDMI_PHY_DIGTXMODE)

Register name: DIGTXMODE

Access type: Read/write

Address: 0x1E

Value at reset: 0x0000

dtb_select[6:0]

The dtb_select[6:0] encodings for the bit pairs are as follows:

Power Sequence

- 0x00: dtb[1] = pll_pwr_on and dtb[0] = pll_rst
- 0x01: dtb[1] = pll_pwr_on and dtb[0] = pll_gear_shift
- 0x02: dtb[1] = pll_pwr_on and dtb[0] = pll_drv_ana
- 0x03: dtb[1] = pll_pwr_on and dtb[0] = cko_word_enb
- 0x04: dtb[1] = pll_pwr_on and dtb[0] = tx_ready
- 0x05: dtb[1] = mpll_pwr_on and dtb[0] = mpll_rst
- 0x06: dtb[1] = mpll_pwr_on and dtb[0] = mpll_gear_shift
- 0x07: dtb[1] = mpll_pwr_on and dtb[0] = mpll_drv_ana
- 0x08: dtb[1] = mpll_pwr_on and dtb[0] = cko_word_enb
- 0x09: dtb[1] = mpll_pwr_on and dtb[0] = tx_ready
- 0x0A: dtb[1] = mpll_pwr_on and dtb[0] = tx_ser_div_en1
- 0x0B: dtb[1] = mpll_pwr_on and dtb[0] = tx_ser_clk_en1
- 0x0C: dtb[1] = ck_ref_enb and dtb[0] = refclk_enb
- 0x0D: dtb[1] = cko_word_enb and dtb[0] = refclk_enb

Transmission Clock Alignment

- 0x0E: Reserved
- 0x0F: dtb[1] = tx_ck_align_enb and dtb[0] = tx_ser_clk_kill0
- 0x10: dtb[1] = tx_ck_align_enb and dtb[0] = tx_ser_clk_kill1
- 0x11: dtb[1] = tx_ck_align_enb and dtb[0] = tx_ser_clk_kill2
- 0x12: dtb[1] = tx_ck_align_enb and dtb[0] = tx_ck_align_done

Resistance Calibration and Analog-to-Digital Converter

- 0x13: Reserved
- 0x14: dtb[1] = ck_rescal[6] and dtb[0] = ck_rescal[5]
- 0x15: dtb[1] = ck_rescal[4] and dtb[0] = ck_rescal[3]
- 0x16: dtb[1] = ck_rescal[2] and dtb[0] = ck_rescal[1]
- 0x17: dtb[1] = ck_rescal[0] and dtb[0] = sup_comp_mode
- 0x18: dtb[1] = sup_comp_rt_result and dtb[0] = sup_comp_rt_r
- 0x19: dtb[1] = sup_dac_n[7] and dtb[0] = sup_dac_n[6]
- 0x1A: dtb[1] = sup_dac_n[5] and dtb[0] = sup_dac_n[4]
- 0x1B: dtb[1] = sup_dac_n[3] and dtb[0] = sup_dac_n[2]
- 0x1C: dtb[1] = sup_dac_n[1] and dtb[0] = sup_dac_n[0]
- 0x1D: dtb[1] = sup_dac_th_n[2] and dtb[0] = sup_dac_th_n[1]
- 0x1E: dtb[1] = sup_dac_th_n[0] and dtb[0] = rescal_rep[6]
- 0x1F: dtb[1] = rescal_rep[5] and dtb[0] = rescal_rep[4]
- 0x20: dtb[1] = rescal_rep[3] and dtb[0] = rescal_rep[2]
- 0x21: dtb[1] = rescal_rep[1] and dtb[0] = rescal_rep[0]
- 0x22: dtb[1] = rcal_enb and dtb[0] = rcal_adc_done
- 0x23: dtb[1] = sup_comp_rt_pwron and dtb[0] = rcal_adc_done
- 0x24: Reserved
- 0x25: Reserved
- 0x26: Reserved
- 0x27: Reserved

TMDS Data Pattern

- 0x28: dtb[1] = tmds_data2[9] and dtb[0] = tmds_data2[8]
- 0x29: dtb[1] = tmds_data2[8] and dtb[0] = tmds_data2[7]
- 0x2A: dtb[1] = tmds_data2[7] and dtb[0] = tmds_data2[6]
- 0x2B: dtb[1] = tmds_data2[6] and dtb[0] = tmds_data2[5]
- 0x2C: dtb[1] = tmds_data2[5] and dtb[0] = tmds_data2[4]
- 0x2D: dtb[1] = tmds_data2[4] and dtb[0] = tmds_data2[3]
- 0x2E: dtb[1] = tmds_data2[3] and dtb[0] = tmds_data2[2]
- 0x2F: dtb[1] = tmds_data2[2] and dtb[0] = tmds_data2[1]
- 0x30: dtb[1] = tmds_data2[1] and dtb[0] = tmds_data2[0]
- 0x31: dtb[1] = tmds_data2[0] and dtb[0] = tmds_data1[9]

- 0x32: dtb[1] = tmds_data1[9] and dtb[0] = tmds_data1[8]
- 0x33: dtb[1] = tmds_data1[8] and dtb[0] = tmds_data1[7]
- 0x34: dtb[1] = tmds_data1[7] and dtb[0] = tmds_data1[6]
- 0x35: dtb[1] = tmds_data1[6] and dtb[0] = tmds_data1[5]
- 0x36: dtb[1] = tmds_data1[5] and dtb[0] = tmds_data1[4]
- 0x37: dtb[1] = tmds_data1[4] and dtb[0] = tmds_data1[3]
- 0x38: dtb[1] = tmds_data1[3] and dtb[0] = tmds_data1[2]
- 0x39: dtb[1] = tmds_data1[2] and dtb[0] = tmds_data1[1]
- 0x3A: dtb[1] = tmds_data1[1] and dtb[0] = tmds_data1[0]
- 0x3B: dtb[1] = tmds_data1[0] and dtb[0] = tmds_data0[9]
- 0x3C: dtb[1] = tmds_data0[9] and dtb[0] = tmds_data0[8]
- 0x3D: dtb[1] = tmds_data0[8] and dtb[0] = tmds_data0[7]
- 0x3E: dtb[1] = tmds_data0[7] and dtb[0] = tmds_data0[6]
- 0x3F: dtb[1] = tmds_data0[6] and dtb[0] = tmds_data0[5]
- 0x40: dtb[1] = tmds_data0[5] and dtb[0] = tmds_data0[4]
- 0x41: dtb[1] = tmds_data0[4] and dtb[0] = tmds_data0[3]
- 0x42: dtb[1] = tmds_data0[3] and dtb[0] = tmds_data0[2]
- 0x43: dtb[1] = tmds_data0[2] and dtb[0] = tmds_data0[1]
- 0x44: dtb[1] = tmds_data0[1] and dtb[0] = tmds_data0[0]
- 0x45: dtb[1] = tmds_data0[0] and dtb[0] = tx_ready

Pattern Generator

- 0x46: dtb[1] = tmds_data2[0] and dtb[0] = pg_dtb2
- 0x47: dtb[1] = tmds_data1[0] and dtb[0] = pg_dtb1
- 0x48: dtb[1] = tmds_data0[0] and dtb[0] = pg_dtb0
- 0x49: dtb[1] = pg_dtb2 and dtb[0] = pg_dtb0
- 0x4A: dtb[1] = pg_dtb1 and dtb[0] = pg_dtb0
- 0x4B: Reserved
- 0x4C: Reserved
- 0x4D: Reserved
- 0x4E: Reserved
- 0x4F: Reserved

Scope

- 0x50: dtb[1] = scope_enb2 and dtb[0] = scope_done2
- 0x51: dtb[1] = scope_enb1 and dtb[0] = scope_done1
- 0x52: dtb[1] = scope_enb0 and dtb[0] = scope_done0
- 0x53: dtb[1] = ck_scope_enb and dtb[0] = ck_scope_done
- 0x54: dtb[1] = scope_enb2 and dtb[0] = tx_scope_out2
- 0x55: dtb[1] = scope_enb1 and dtb[0] = tx_scope_out1
- 0x56: dtb[1] = scope_enb0 and dtb[0] = tx_scope_out0

- 0x57: Reserved
- 0x58: Reserved
- 0x59: Reserved

I2C

- 0x5A: dtb[1] = start_cond and dtb[0] = cregs_write (Not stored in the control register.)
- 0x5B: dtb[1] = sda_pull_dn_n and dtb[0] = cregs_ack (Not stored in the control register.)
- 0x5C: dtb[1] = cregs_read and dtb[0] = cregs_rd_data[0] (Not stored in the control register.)
- 0x5D: Reserved
- 0x5E: Reserved
- 0x5F: Reserved
- 0x60: Reserved
- 0x61: Reserved
- 0x62: Reserved
- 0x63: Reserved

Rx Sense

- 0x64: dtb[1] = rx_sense of clock driver and dtb[0] = rx_sense of CH2's driver
- 0x65: dtb[1] = rx_sense of CH1's driver and dtb[0] = rx_sense of CH0's driver
- 0x66: -

Address: 0h base + 1Eh offset = 1Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Reserved			dtb_select[6:0]						invert_data2	invert_data1	invert_data0	cko_word_div_enb	skip_bit2	skip_bit1	skip_bit0
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HDMI_PHY_DIGTXMODE field descriptions

Field	Description
15–13 -	This field is reserved. Reserved
12–7 dtb_select[6:0]	Debug Test Bus Select This field determines the pair of bits placed on the dtb[1:0] bus. These selected pairs of bits come from the control register. The values that appear on dtb[1:0] are the current values actually stored in the control register (not the override values) with the exception of I^2C values, which are not stored in the control register.

Table continues on the next page...

HDMI_PHY_DIGTXMODE field descriptions (continued)

Field	Description
	For information about the bit pairs, see dtb_select[6:0] .
6 invert_data2	Inverter Data 2 This bit enables or disables the inverting feature for the transmitted pattern on channel 2. 0 Disable the inverting feature on the third channel. 1 Enable the inverting feature on the third channel.
5 invert_data1	Inverter Data 1 This bit enables or disables the inverting feature for the transmitted pattern on channel 1. 0 Disable the inverting feature on the second channel. 1 Enable the inverting feature on the second channel.
4 invert_data0	Inverter Data 0 This bit enables or disables the inverting feature for the transmitted pattern on channel 0. 0 Disable the inverting feature on the first channel. 1 Enable the inverting feature on the first channel.
3 cko_word_div_enb	Clock Output Word Divider Enable This bit enables or disables the output clock word divider. 0 Disable the output clock divider. 1 Enable the output clock divider.
2 skip_bit2	Skip Bit 2 This bit enables or disables skipping of the ninth bit of the transmitted pattern on channel 2. 0 Disable the skipping feature on the third channel. 1 Enable the skipping feature on the third channel.
1 skip_bit1	Skip Bit 1 This bit enables or disables skipping of the ninth bit of the transmitted pattern on channel 1. 0 Disable the skipping feature on the second channel. 1 Enable the skipping feature on the second channel.
0 skip_bit0	Skip Bit 0 This bit enables or disables skipping of the ninth bit of the transmitted pattern on channel 0. 0 Disable the skipping feature on the first channel. 1 Enable the skipping feature on the first channel.

35.7.32 Scope, Transmission Clock Alignment, and Resistance Calibration Set-on-Done Status (HDMI_PHY_STR_STATUS)

Register name: -

Access type: Read/write/asynchronous set-on-done

Address: 0x1F

Value at reset: 0x0000

Address: 0h base + 1Fh offset = 1Fh

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved		rcal_adc_done	tx_ck_align_done	ck_scope_done	scope_done2	scope_done1	scope_done0
Reset	0	0	0	0	0	0	0	0

HDMI_PHY_STR_STATUS field descriptions

Field	Description
15–6 -	This field is reserved. Reserved
5 rcal_adc_done	Resistance Calibration Analog-to-Digital Converter Done This bit indicates the status of completing the resistance calibration FSM. 0 The resistance calibration FSM is not complete. 1 The resistance calibration FSM is complete.
4 tx_ck_align_done	Transmission Clock Alignment Done This bit indicates the status of completing the transmission clock alignment FSM. 0 The transmission clock alignment FSM is not complete. 1 The transmission clock alignment FSM is complete.
3 ck_scope_done	Clock Scope Done This bit indicates the status of tracing of 1's on the clock channel. 0 The tracing process on the clock channel is not complete. 1 The tracing process on the clock channel is complete.
2 scope_done2	Scope Done 2 This bit indicates the status of tracing of 1's on channel 2. 0 The tracing process on channel 2 is not complete. 1 The tracing process on channel 2 is complete.
1 scope_done1	Scope Done 1 This bit indicates the status of tracing of 1's on channel 1. 0 The tracing process on channel 1 is not complete. 1 The tracing process on channel 1 is complete.
0 scope_done0	Scope Done 0 This bit indicates the status of tracing of 1's on channel 0. 0 The tracing process on channel 0 is not complete. 1 The tracing process on channel 0 is complete.

35.7.33 Scope Counter on Channel 0 (HDMI_PHY_SCOPECNT0)

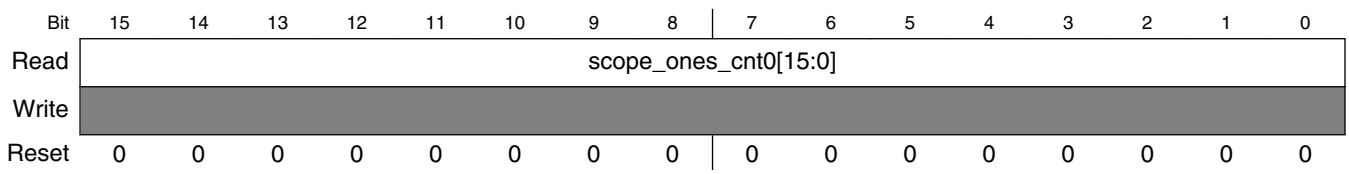
Register name: SCOPECNT0

Access type: Read-only

Address: 0x20

Value at reset: 0x0000

Address: 0h base + 20h offset = 20h



HDMI_PHY_SCOPECNT0 field descriptions

Field	Description
scope_ones_ cnt0[15:0]	Scope 1's Counter 0 This register carries the number of counted 1's on channel 0. If the LFSR15 was used to generate the scope patterns, you must read the MSB bits under 0x25 register.

35.7.34 Scope Counter on Channel 1 (HDMI_PHY_SCOPECNT1)

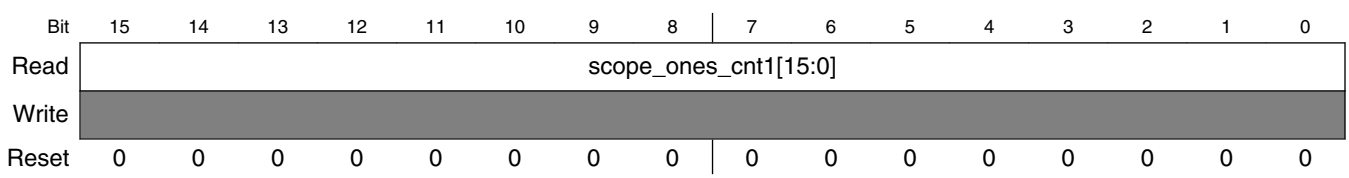
Register name: SCOPECNT1

Access type: Read-only

Address: 0x21

Value at reset: 0x0000

Address: 0h base + 21h offset = 21h



HDMI_PHY_SCOPECNT1 field descriptions

Field	Description
scope_ones_ cnt1[15:0]	Scope 1's Counter 1

HDMI_PHY_SCOPECNT1 field descriptions (continued)

Field	Description
	This register carries the number of counted 1's on channel 1. If the LFSR15 was used to generate the scope patterns, you must read the MSB bits under 0x25 register.

35.7.35 Scope Counter on Channel 2 (HDMI_PHY_SCOPECNT2)

Register name: SCOPECNT2

Access type: Read-only

Address: 0x22

Value at reset: 0x0000

Address: 0h base + 22h offset = 22h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	scope_ones_cnt2[15:0]															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HDMI_PHY_SCOPECNT2 field descriptions

Field	Description
scope_ones_ cnt2[15:0]	Scope 1's Counter 2 This register carries the number of counted 1's on channel 2. If the LFSR15 was used to generate the scope patterns, you must read the MSB bits under 0x26 register.

35.7.36 Scope Counter on Clock Channel (HDMI_PHY_SCOPECNTCLK)

Register name: SCOPECNTCK

Access type: Read-only

Address: 0x23

Value at reset: 0x0000

Address: 0h base + 23h offset = 23h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	ck_scope_ones_cnt[15:0]															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HDMI_PHY_SCOPECNTCLK field descriptions

Field	Description
ck_scope_ones_cnt[15:0]	Clock Scope 1's Counter This register carries the number of counted 1's on the clock channel. If the LFSR15 was used to generate the scope patterns, you must read the MSB bits under 0x26 register.

35.7.37 Scope Sample Count MSB, Scope Sample Repetition (HDMI_PHY_SCOPESAMPLE)

Register name: SCOPESAMPLE

Access type: Read/write

Address: 0x24

Value at reset: 0x13c0

Address: 0h base + 24h offset = 24h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Reserved			scope_sample_rep[6:0]					scope_sample_cnt[15:9]							
Write																
Reset	0	0	0	1	0	0	1	1	1	1	0	0	0	0	0	0

HDMI_PHY_SCOPESAMPLE field descriptions

Field	Description
15–13 -	This field is reserved. Reserved
12–6 scope_sample_rep[6:0]	Scope Sample Repetition Number of repetitions made by the scope FSM. The total samples captured is scope_sample_rep x scope_sample_cnt.
scope_sample_cnt[15:9]	Scope Sample Counter Indicates the number of samples that will be counted (should be multiple of the LFSR length). These samples are the MSB bits only.

35.7.38 Scope Counter MSB Channel 0 and Channel 1 (HDMI_PHY_SCOPECNTMSB01)

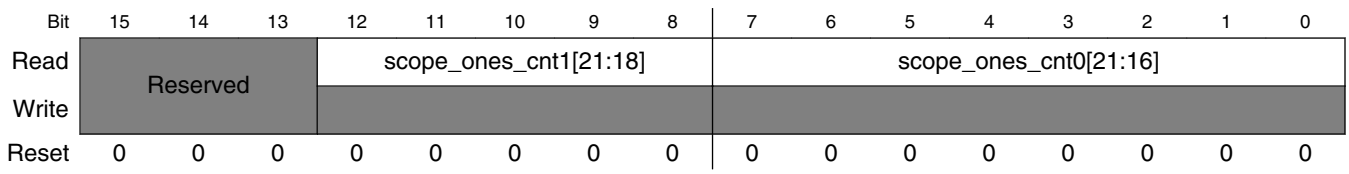
Register name: SCOPECNTMSB01

Access type: Read-only

Address: 0x25

Value at reset: 0x0000

Address: 0h base + 25h offset = 25h



HDMI_PHY_SCOPECNTMSB01 field descriptions

Field	Description
15–13 -	This field is reserved. Reserved
12–8 scope_ones_cnt1[21:18]	Scope 1's Counter 1 This register carries the number of counted 1's on channel 1. These 1's are the MSB bits only.
scope_ones_cnt0[21:16]	Scope 1's Counter 0 This register carries the number of counted 1's on channel 0. These 1's are the MSB bits only.

35.7.39 Scope Counter MSB Channel 2 and Clock Channel (HDMI_PHY_SCOPECNTMSB2CK)

Register name: SCOPECNTMSB2CK

Access type: Read-only

Address: 0x26

Value at reset: 0x0000

Address: 0h base + 26h offset = 26h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Reserved			ck_scope_ones_cnt[21:16]					scope_ones_cnt2[21:16]							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HDMI_PHY_SCOPECNTMSB2CK field descriptions

Field	Description
15–13 - Reserved	This field is reserved.
12–6 ck_scope_ones_cnt[21:16]	Clock Scope 1's Counter This register carries the number of counted 1's on the clock channel. These 1's are the MSB bits only.
scope_ones_cnt2[21:16]	Scope 1's Counter 2 This register carries the number of counted 1's on channel 2. These 1's are the MSB bits only.

35.8 Appendix A: Driver Voltage Level Configuration

This appendix describes the driver voltage level configuration.

This configuration depends on the source termination value, the driver pre-emphasis settings, and the target signal voltage level swing.

A correct configuration must be set to meet both eye diagram mask and the specified high and low signal voltage levels.

To correctly configure the driver voltage level, the following parameters and signals (represented through their symbol) must be taken into consideration:

- VPH_{RXTERM} = 3.3 V -> 3.3-V supply rail connected to HDMI PHY sink termination resistors
- RXTERM = 50 Ω -> HDMI PHY sink termination resistors

Control Register	Signal	Symbol
0x19	d_tx_temp[2:0]	RTERM
0x0E	sup_tx_lvl[4.:0]	TXLVL
0x0E	sup_ck_lvl[4.:0]	CKLVL
0x09	tx_symon	SYMON
0x09	tx_traon	TRAON
0x09	tx_trbon	TRBON

The following table defines the pre-emphasis factor (PREEMPH) to be used in the information that follows.

SYMON	TRAON	TRBON	PREEMPH
0	0	0	-
1	0	0	0.00
1	0	1	0.08
1	1	0	0.17
1	1	1	0.25

The following equations can be used to calculate-for a certain signal voltage swing (VSWING) and PHY configuration (PREEMPH, RTERM)-the signal's high and low voltage levels (VHI, VLO).

VHI = VLO + VSWING

TXLVL = CKLVL = [0.772 - (VPHRXTERM - VLO)] / 0.01405

For a certain termination value and pre-emphasis configuration (factor), users input only the VSWING value. With this data, users can obtain the respective VHI and VLO DC levels and the TXLVL and CKLVL configuration to apply to the PHY.

Lower TXLVL/CKLVL values result in higher signal amplitudes, while higher TXLVL/ CKLVL values result in lower signal amplitudes.

Values for VSWING, VHI, and VLO must be within HDMI 1.4a specification limits. The table below provides driver voltage level settings for some example scenarios.

Table 35-21. Example Driver Voltage Level Settings

RTERM	SYMON	TRAON	TRBON	VHI (V)	VLO (V)	VSWING (V)	TXLVL	TXLVL (BIN)	CKLVL	CKLVL (BIN)
100	1'b1	1'b0	1'b0	3.200	2.800	0.400	19	10011	19	10011
100	1'b1	1'b0	1'b0	3.175	2.675	0.500	10	01010	10	01010
100	1'b1	1'b0	1'b1	3.107	2.607	0.500	6	00110	6	00110
133	1'b1	1'b0	1'b0	3.225	2.825	0.400	21	10101	21	10101
133	1'b1	1'b0	1'b0	3.206	2.706	0.500	13	01101	13	01101
133	1'b1	1'b0	1'b1	3.143	2.643	0.500	8	01000	8	01000

35.9 Appendix B

This appendix describes the PLL/MPLL configurations that must be made for each video mode of operation.

The PLL/MPLL configurations are divided in two distinct sections. Power-Up Requirements describes how to configure the PLL/MPLL mode of operation, and PLL/MPLL Generic Configuration Settings describes all the settings required to configure the PLL/MPLL for each specific video mode.

35.9.1 Single or Two-PLL in Coherent or Non-Coherent Mode of Operation

Table 35-22. Single or Two-PLL in Coherent or Non-Coherent Mode of Operation

Mode of Operation	bypass_ppll Register address: 0013: Bit 11	cko_sel<1> Register address: 0017: Bit 2	cko_sel<0> Register address: 0017: Bit 1
Two-PLL Non-Coherent (default)	0	0	0
Two-PLL Coherent	0	1	1
Single-PLL Non-Coherent	1	1	0
Single-PLL Coherent	1	1	1

NOTE

Single-PLL Coherent and Non-Coherent modes can be set for only video formats with no pixel repetition and color depth equal to 8 bits.

35.9.2 PLL/MPLL Generic Configuration Settings

Each supported video format can be referred to by its input pixel clock frequency, pixel repetition, and color depth in bits.

The table beblow provides all the PLL/MPLL settings necessary to completely configure the blocks for any supported video format.

Table B-2 PLL/MPLL Generic Configuration Settings

Divider Settings

Table 35-23. PLL/MPLL Generic Configuration Settings

			Divider Settings											PLL Charge Pump Settings								MPLL Charge Pump Settings							
			0000:Bit14	0000:Bit13	0000:Bit8	0000:Bit7	0000:Bit6	0000:Bit5	0000:Bit4	0000:Bit3	0000:Bit2	0000:Bit1	0000:Bit0	0010:Bit5	0010:Bit4	0010:Bit3	0010:Bit2	0010:Bit1	0010:Bit0	0015:Bit3	0015:Bit2	0010:Bit11	0010:Bit10	0010:Bit9	0010:Bit8	0010:Bit7	0010:Bit6	0015:Bit1	0015:Bit0
PixelClock(MHz)	PixelRepetition	ColorDepthinBits	prep_div<1>	prep_div<0>	mpll_n_cntrl<1>	mpll_n_cntrl<0>	pll_n_cntrl<1>	pll_n_cntrl<0>	pixel_rep<2>	pixel_rep<1>	pixel_rep<0>	clr_dpth<1>	clr_dpth<0>	pll_prop_cntrl<2>	pll_prop_cntrl<1>	pll_prop_cntrl<0>	pll_int_cntrl<2>	pll_int_cntrl<1>	pll_int_cntrl<0>	pll_gmp_cntrl<1>	pll_gmp_cntrl<0>	mpll_prop_cntrl<2>	mpll_prop_cntrl<1>	mpll_prop_cntrl<0>	mpll_int_cntrl<2>	mpll_int_cntrl<1>	mpll_int_cntrl<0>	mpll_gmp_cntrl<1>	mpll_gmp_cntrl<0>
13.5	2	8	0	0	1	1	1	1	0	0	0	1	1	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0
13.5	2	10	0	1	1	1	1	1	1	0	0	0	1	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0
13.5	2	12	1	0	1	1	1	1	1	0	0	1	0	0	1	1	1	0	0	0	0	0	1	1	0	1	1	0	0
13.5	2	16	1	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	1
13.5	4	8	0	0	1	0	1	1	0	0	1	1	1	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	1
13.5	4	10	0	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	1
13.5	4	12	1	0	1	0	1	1	1	0	1	1	0	0	1	1	1	0	0	0	0	0	1	1	0	1	1	0	1
13.5	4	16	1	1	0	1	1	1	0	1	0	1	1	0	1	1	1	0	0	0	0	1	0	0	1	0	0	1	0
18	3	8	0	0	1	0	1	1	0	0	1	1	0	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	1
18	3	16	1	1	0	1	1	1	0	1	0	1	0	0	1	1	1	0	0	0	0	1	0	0	1	0	0	1	0
24.175	1	8	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0
24.175	1	10	0	1	1	1	1	1	0	0	0	0	1	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0

Table continues on the next page...

Table 35-23. PLL/MPLL Generic Configuration Settings (continued)

			Divider Settings											PLL Charge Pump Settings								MPLL Charge Pump Settings							
			0006: Bit 14	0006: Bit 13	0006: Bit 8	0006: Bit 7	0006: Bit 6	0006: Bit 5	0006: Bit 4	0006: Bit 3	0006: Bit 2	0006: Bit 1	0006: Bit 0	0010: Bit 5	0010: Bit 4	0010: Bit 3	0010: Bit 2	0010: Bit 1	0010: Bit 0	0015: Bit 3	0015: Bit 2	0010: Bit 11	0010: Bit 10	0010: Bit 9	0010: Bit 8	0010: Bit 7	0010: Bit 6	0015: Bit 1	0015: Bit 0
PixelClock (MHz)	PixelRepetition	ColorDepthinBits	prep_div<1>	prep_div<0>	mpll_n_cntrl<1>	mpll_n_cntrl<0>	pll_n_cntrl<1>	pll_n_cntrl<0>	pixel_rep<2>	pixel_rep<1>	pixel_rep<0>	clr_dpth<1>	clr_dpth<0>	pll_prop_cntrl<2>	pll_prop_cntrl<1>	pll_prop_cntrl<0>	pll_int_cntrl<2>	pll_int_cntrl<1>	pll_int_cntrl<0>	pll_gmp_cntrl<1>	pll_gmp_cntrl<0>	mpll_prop_cntrl<2>	mpll_prop_cntrl<1>	mpll_prop_cntrl<0>	mpll_int_cntrl<2>	mpll_int_cntrl<1>	mpll_int_cntrl<0>	mpll_gmp_cntrl<1>	mpll_gmp_cntrl<0>
24.175	1	12	1	0	1	1	1	1	0	0	0	1	0	0	1	1	1	0	0	0	0	0	1	1	0	1	1	0	0
24.175	1	16	1	1	1	0	1	0	0	0	0	1	1	0	1	1	1	0	0	0	1	1	0	0	1	0	0	0	1
27	1	8	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0
27	1	10	0	1	1	1	1	1	0	0	0	0	1	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0
27	1	12	1	0	1	1	1	1	0	0	0	1	0	0	1	1	1	0	0	0	0	0	1	1	0	1	1	0	0
27	1	16	1	1	1	0	1	0	0	0	0	1	1	0	1	1	1	0	0	0	1	1	0	0	1	0	0	0	1
27	2	8	0	0	1	0	1	0	0	0	0	1	1	0	1	1	1	0	0	0	1	1	0	0	1	0	0	0	1
27	2	10	0	1	1	0	1	1	0	0	1	0	1	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	1
27	2	12	1	0	1	0	1	1	0	0	1	1	0	0	1	1	1	0	0	0	0	0	1	1	0	1	1	0	1
27	2	16	1	1	0	1	1	0	0	0	1	1	1	0	1	1	1	0	0	0	1	1	0	0	1	0	0	1	0
27	4	8	0	0	0	1	1	0	0	0	1	1	1	0	1	1	1	0	0	0	1	1	0	0	1	0	0	1	0
27	4	10	0	1	0	1	1	1	0	1	0	0	1	0	1	1	1	0	0	0	0	1	0	0	1	0	0	1	0
27	4	12	1	0	0	1	1	1	0	1	0	1	0	0	1	1	1	0	0	0	0	0	1	1	0	1	1	1	0
27	4	16	1	1	0	0	1	0	0	1	0	1	1	0	1	1	1	0	0	0	1	1	0	0	1	0	0	1	1
36	1	8	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0
36	1	16	1	1	1	0	1	0	0	0	0	1	1	0	1	1	1	0	0	0	1	1	0	0	1	0	0	0	1
50, 35	1	8	0	0	1	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	0	0	1	0	0	0	1

Table continues on the next page...

Table 35-23. PLL/MPLL Generic Configuration Settings (continued)

			Divider Settings											PLL Charge Pump Settings								MPLL Charge Pump Settings							
			00006:Bit14	00006:Bit13	00006:Bit8	00006:Bit7	00006:Bit6	00006:Bit5	00006:Bit4	00006:Bit3	00006:Bit2	00006:Bit1	00006:Bit0	00010:Bit5	00010:Bit4	00010:Bit3	00010:Bit2	00010:Bit1	00010:Bit0	00015:Bit3	00015:Bit2	00010:Bit11	00010:Bit10	00010:Bit9	00010:Bit8	00010:Bit7	00010:Bit6	00015:Bit1	00015:Bit0
PixelClock(MHz)	PixelRepetition	ColorDepthinBits	prep_div<1>	prep_div<0>	mpll_n_cntrl<1>	mpll_n_cntrl<0>	pll_n_cntrl<1>	pll_n_cntrl<0>	pixel_rep<2>	pixel_rep<1>	pixel_rep<0>	clr_dpth<1>	clr_dpth<0>	pll_prop_cntrl<2>	pll_prop_cntrl<1>	pll_prop_cntrl<0>	pll_int_cntrl<2>	pll_int_cntrl<1>	pll_int_cntrl<0>	pll_gmp_cntrl<1>	pll_gmp_cntrl<0>	mpll_prop_cntrl<2>	mpll_prop_cntrl<1>	mpll_prop_cntrl<0>	mpll_int_cntrl<2>	mpll_int_cntrl<1>	mpll_int_cntrl<0>	mpll_gmp_cntrl<1>	mpll_gmp_cntrl<0>
50, 35	1	10	0	1	1	0	1	0	0	0	0	0	1	0	1	1	1	0	0	0	1	1	0	0	1	0	0	0	1
50, 35	1	12	1	0	1	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1	1	0	1	1	0	1
50, 35	1	16	1	1	0	1	0	1	0	0	0	1	1	0	1	1	1	0	0	1	0	1	0	0	1	0	0	1	0
50, 35	2	8	0	0	0	1	0	1	0	0	0	1	1	0	1	1	1	0	0	1	0	1	0	0	1	0	0	1	0
50, 35	2	10	0	1	0	1	1	0	0	0	1	0	1	0	1	1	1	0	0	0	1	1	0	0	1	0	0	1	0
50, 35	2	12	1	0	0	1	1	0	0	0	1	1	0	0	1	1	1	0	0	0	1	0	1	1	0	1	1	1	0
50, 35	2	16	1	1	0	0	0	1	0	0	1	1	1	0	1	1	1	0	0	1	0	1	0	0	1	0	0	1	1
54	1	8	0	0	1	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	0	0	1	0	0	0	1
54	1	10	0	1	1	0	1	0	0	0	0	0	1	0	1	1	1	0	0	0	1	1	0	0	1	0	0	0	1
54	1	12	1	0	1	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1	1	0	1	1	0	1
54	1	16	1	1	0	1	0	1	0	0	0	1	1	0	1	1	1	0	0	1	0	1	0	0	1	0	0	1	0
54	2	8	0	0	0	1	0	1	0	0	0	1	1	0	1	1	1	0	0	1	0	1	0	0	1	0	0	1	0
54	2	10	0	1	0	1	1	0	0	0	1	0	1	0	1	1	1	0	0	0	1	1	0	0	1	0	0	1	0

Table continues on the next page...

Table 35-23. PLL/MPLL Generic Configuration Settings (continued)

			Divider Settings											PLL Charge Pump Settings								MPLL Charge Pump Settings							
			0006:Bit14	0006:Bit13	0006:Bit8	0006:Bit7	0006:Bit6	0006:Bit5	0006:Bit4	0006:Bit3	0006:Bit2	0006:Bit1	0006:Bit0	0010:Bit5	0010:Bit4	0010:Bit3	0010:Bit2	0010:Bit1	0010:Bit0	0015:Bit3	0015:Bit2	0010:Bit11	0010:Bit10	0010:Bit9	0010:Bit8	0010:Bit7	0010:Bit6	0015:Bit1	0015:Bit0
PixelClock(MHz)	PixelRepetition	ColorDepthinBits	prep_div<1>	prep_div<0>	mpll_n_cntrl<1>	mpll_n_cntrl<0>	pll_n_cntrl<1>	pll_n_cntrl<0>	pixel_rep<2>	pixel_rep<1>	pixel_rep<0>	clr_dpth<1>	clr_dpth<0>	pll_prop_cntrl<2>	pll_prop_cntrl<1>	pll_prop_cntrl<0>	pll_int_cntrl<2>	pll_int_cntrl<1>	pll_int_cntrl<0>	pll_gmp_cntrl<1>	pll_gmp_cntrl<0>	mpll_prop_cntrl<2>	mpll_prop_cntrl<1>	mpll_prop_cntrl<0>	mpll_int_cntrl<2>	mpll_int_cntrl<1>	mpll_int_cntrl<0>	mpll_gmp_cntrl<1>	mpll_gmp_cntrl<0>
54	2	12	1	0	0	1	1	0	0	0	1	1	0	0	1	1	1	0	0	0	1	0	1	1	0	1	1	1	0
54	2	16	1	1	0	0	0	1	0	0	1	1	1	0	1	1	1	0	0	1	0	1	0	0	1	0	0	1	1
58.4	1	8	0	0	1	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	0	0	1	0	0	0	1
58.4	1	10	0	1	1	0	1	0	0	0	0	0	1	0	1	1	1	0	0	0	1	0	1	1	0	1	1	0	1
58.4	1	12	1	0	1	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1	1	0	1	1	0	1
58.4	1	16	1	1	0	1	0	1	0	0	0	1	1	0	1	1	1	0	0	1	0	1	0	0	1	0	0	1	0
72	1	8	0	0	1	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	1	0	1	1	0	1
72	1	10	0	1	1	0	1	0	0	0	0	0	1	0	1	1	1	0	0	0	1	0	1	1	0	1	1	0	1
72	1	12	1	0	0	1	0	1	0	0	0	1	0	0	1	1	1	0	0	1	0	1	0	0	1	0	0	1	0
72	1	16	1	1	0	1	0	1	0	0	0	1	1	0	1	1	1	0	0	1	0	0	1	1	0	1	1	1	0
74.25	1	8	0	0	1	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	1	0	1	1	0	1
74.25	1	10	0	1	0	1	0	1	0	0	0	0	1	0	1	1	1	0	0	1	0	1	0	1	1	0	1	1	0
74.25	1	12	1	0	0	1	0	1	0	0	0	1	0	0	1	1	1	0	0	1	0	1	0	0	1	0	0	1	0
74.25	1	16	1	1	0	1	0	1	0	0	0	1	1	0	1	1	1	0	0	1	0	0	1	1	0	1	1	1	0

Table continues on the next page...

Table 35-23. PLL/MPLL Generic Configuration Settings (continued)

			Divider Settings											PLL Charge Pump Settings								MPLL Charge Pump Settings							
			00006:Bit14	00006:Bit13	00006:Bit8	00006:Bit7	00006:Bit6	00006:Bit5	00006:Bit4	00006:Bit3	00006:Bit2	00006:Bit1	00006:Bit0	00010:Bit5	00010:Bit4	00010:Bit3	00010:Bit2	00010:Bit1	00010:Bit0	00015:Bit3	00015:Bit2	00010:Bit11	00010:Bit10	00010:Bit9	00010:Bit8	00010:Bit7	00010:Bit6	00015:Bit1	00015:Bit0
PixelClock(MHz)	PixelRepetition	ColorDepthinBits	prep_div<1>	prep_div<0>	mpll_n_cntrl<1>	mpll_n_cntrl<0>	pll_n_cntrl<1>	pll_n_cntrl<0>	pixel_rep<2>	pixel_rep<1>	pixel_rep<0>	clr_dpth<1>	clr_dpth<0>	pll_prop_cntrl<2>	pll_prop_cntrl<1>	pll_prop_cntrl<0>	pll_int_cntrl<2>	pll_int_cntrl<1>	pll_int_cntrl<0>	pll_gmp_cntrl<1>	pll_gmp_cntrl<0>	mpll_prop_cntrl<2>	mpll_prop_cntrl<1>	mpll_prop_cntrl<0>	mpll_int_cntrl<2>	mpll_int_cntrl<1>	mpll_int_cntrl<0>	mpll_gmp_cntrl<1>	mpll_gmp_cntrl<0>
108	1	8	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	0	1	0	0	1	0
108	1	10	0	1	0	1	0	1	0	0	0	0	1	0	1	1	1	0	0	1	0	1	0	0	1	0	0	1	0
108	1	12	1	0	0	1	0	1	0	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	0	1	1	1	0
108	1	16	1	1	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	1	1	1	0	0	1	0	0	1	1
118,8	1	8	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	0	1	0	0	1	0
118,8	1	10	0	1	0	1	0	1	0	0	0	0	1	0	1	1	1	0	0	1	0	1	0	0	1	0	0	1	0
118,8	1	12	1	0	0	1	0	1	0	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	0	1	1	1	0
118,8	1	16	1	1	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	1	1	1	0	0	1	0	0	1	1
144	1	8	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0	1	0	0	1	1	0	1	1	1	0
144	1	10	0	1	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	1	1	1	0	1	1	0	1	1	1
144	1	12	1	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	1	1	1	0	0	1	0	0	1	1
144	1	16	1	1	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	1	1	0	1	1	0	1	1	1	1

Table continues on the next page...

Table 35-23. PLL/MPLL Generic Configuration Settings (continued)

			Divider Settings											PLL Charge Pump Settings								MPLL Charge Pump Settings							
			0000:Bit14	0000:Bit13	0000:Bit8	0000:Bit7	0000:Bit6	0000:Bit5	0000:Bit4	0000:Bit3	0000:Bit2	0000:Bit1	0000:Bit0	0010:Bit5	0010:Bit4	0010:Bit3	0010:Bit2	0010:Bit1	0010:Bit0	0015:Bit3	0015:Bit2	0010:Bit11	0010:Bit10	0010:Bit9	0010:Bit8	0010:Bit7	0010:Bit6	0015:Bit1	0015:Bit0
PixelClock(MHz)	PixelRepetition	ColorDepthinBits	prep_div<1>	prep_div<0>	mpll_n_cntrl<1>	mpll_n_cntrl<0>	pll_n_cntrl<1>	pll_n_cntrl<0>	pixel_rep<2>	pixel_rep<1>	pixel_rep<0>	clr_dpth<1>	clr_dpth<0>	pll_prop_cntrl<2>	pll_prop_cntrl<1>	pll_prop_cntrl<0>	pll_int_cntrl<2>	pll_int_cntrl<1>	pll_int_cntrl<0>	pll_gmp_cntrl<1>	pll_gmp_cntrl<0>	mpll_prop_cntrl<2>	mpll_prop_cntrl<1>	mpll_prop_cntrl<0>	mpll_int_cntrl<2>	mpll_int_cntrl<1>	mpll_int_cntrl<0>	mpll_gmp_cntrl<1>	mpll_gmp_cntrl<0>
148.5	1	8	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0	1	0	0	1	1	0	1	1	1	0
148.5	1	10	0	1	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	1	1	1	0	1	1	0	1	1	1
148.5	1	12	1	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	1	1	1	0	0	1	0	0	1	1
148.5	1	16	1	1	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	1	1	0	1	1	0	1	1	1	1
216	1	8	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0	1	0	0	1	1	0	1	1	1	0
216	1	10	0	1	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	1	1	1	0	1	1	0	1	1	1
216	1	12	1	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	1	1	1	0	0	1	0	0	1	1

35.10 Appendix C: 3D Video Formats

3D video formats are created from any of the supported VICs. The 3D structure in which the frames are organized determines the way data is transferred.

The 3D video format is indicated using the VIC in the AVI InfoFrame (indicating one of the 2D formats defined in the CEA-861-D standard) in conjunction with the 3D_Structure field in the HDMI vendor-specific InfoFrame.

For Side-by-Side (Half) and Top-and-Bottom 3D structures, pixel clock frequency is equal to the original VIC pixel clock frequency.

For the L+depth+GFX+GFX-depth 3D structure, pixel clock frequency is equal to 4x the original VIC pixel clock frequency.

For other 3D structures, pixel clock frequency is equal to 2x the original VIC pixel clock frequency.

The following table summarizes the original 2D VIC pixel clock frequencies required to support the 3D structures.

Table 35-24. 2D VIC Pixel Clock Frequencies Required to Support 3D Structures

3D Structure	2D Pixel Clock Frequency
0000 (Frame Packing)	2x 2D pixel clock frequency
0001 (Field alternative)	2x 2D pixel clock frequency
0010 (Line alternative)	2x 2D pixel clock frequency
0011 (Side-by-Side (Full))	2x 2D pixel clock frequency
0100 (L + depth)	2x 2D pixel clock frequency
0101 (L + depth + GPX + GFX-depth)	4x 2D pixel clock frequency
0110 (Top-and-Bottom)	2D pixel clock frequency
1000 (Side-by-Side (Half))	2D pixel clock frequency

VIC-specific PLL/MPLL configurations apply to Side-by-Side (Half) and Top-and-Bottom 3D structures. To obtain the PLL/MPLL configuration for a specific VIC used within a 3D structure, see PLL/MPLL Generic Configuration Settings. Each configuration corresponds to a specific video format's pixel clock, pixel repetition, and color depth combination.

Example:

VIC 32: No pixel repetition, PCLK = 74.25 MHz, CD = 8,10,12,16

VIC 32 within 3D structure:

- Side-by-Side (Half) 3D structure (same pixel clock frequency): Same PLL/MPLL configuration as for VIC 32.
- Frame Packing (doubled pixel clock frequency): PLL/MPLL configuration corresponds to a row in Table 35-23 with the following: no pixel repetition, PCLK = 148.5 MHz, CD = 8,10,12,16.

For VICs where PCLK = 148.5 MHz, no color depth is supported for 3D video format, because the pixel clock frequency would be greater than 340 MHz.

Table 35-25 provides examples of primary 3D video format timings, and Table 35-26 provides examples of other 3D video format timings. Secondary 3D video format timings can be found in the HDMI specification.

Table 35-25. Primary 3D Video Format Timings

VIC	Video	Vertical Refresh Rate (Hz)	2D Pixel Clock (MHz)	3D Structure	3D Pixel Clock (MHz)
4	1,280x720p	59.94/60	74.25	0000 (Frame Packing)	148.50
4	1,280x720p	59.94/60	74.25	0110 (Top-and-Bottom)	74.25
4	1,280x720p	59.94/60	74.25	1000 (Side-by-Side (Half))	74.25
19	1,280x720p	50	74.25	0000 (Frame Packing)	148.50
19	1,280x720p	50	74.25	0110 (Top-and-Bottom)	74.25
19	1,280x720p	50	74.25	1000 (Side-by-Side (Half))	74.25
60	1,280x720p	23.97/24	58.40	0000 (Frame Packing)	118.8
62	1,280x720p	29.97/30	74.25	0000 (Frame Packing)	148.50
5	1,920x1,080i	59.94/60	74.25	0000 (Frame Packing)	148.50
5	1,920x1,080i	59.94/60	74.25	1000 (Side-by-Side (Half))	74.25
20	1,920x1,080i	50	74.25	0000 (Frame Packing)	148.50
20	1,920x1,080i	50	74.25	1000 (Side-by-Side (Half))	74.25
32	1,920x1,080p	23.98/24	74.25	0000 (Frame Packing)	148.50
32	1,920x1,080p	23.98/24	74.25	0110 (Top-and-Bottom)	74.25
32	1,920x1,080p	23.98/24	74.25	1000 (Side-by-Side (Half))	74.25
34	1,920x1,080p	29.98/30	74.25	0000 (Frame Packing)	148.50
34	1,920x1,080p	29.98/30	74.25	0110 (Top-and-Bottom)	74.25
16	1,920x1,080p	59.94/60	148.50	0110 (Top-and-Bottom)	148.50
31	1,920x1,080p	50	148.50	0110 (Top-and-Bottom)	148.50

Table 35-26. Examples of Other 3D Video Formats

VIC	Video	Vertical Refresh Rate (Hz)	2D Pixel Clock (MHz)	3D Structure	3D Pixel Clock (MHz)
5	1,920x1,080i	59.94/60	74.25	0001 (Field alternative)	148.5
20	1,920x1,080i	50	74.25	0001 (Field alternative)	148.5
16	1,920x1,080p	59.94/60	148.50	0010 (Line alternative)	297
31	1.920x1,080p	50	148.50	0010 (Line alternative)	297
16	1,920x1,080p	59.94/60	148.50	0011 (Side-by-Side (Full))	297
31	1,920x1,080p	50	148.50	0011 (Side-by-Side (Full))	297
19	1,280x720p	50	74.25	0100 (L + depth)	148.5
19	1,280x720p	50	74.25	0101 (L + depth + GFX + GFX-depth)	297

Chapter 36
I2C Controller (I2C)

36.1 Overview

This chapter describes block-level operation and programming of I2C. The chapter is intended for a block-driver software developer. To understand how the block is integrated at the SoC level, a system software developer should see discussions of the block in the appropriate SoC-level chapter(s).

References: This document assumes an understanding of the following document:

- *The I2C Bus Specification*, Version 2.1, by Philips Semiconductor

The Inter IC (I2C) provides functionality of a standard I2C slave and master. The I2C is designed to be compatible with the standard NXP I2C bus protocol.

NOTE

Four independent I2C channels are available.

I2C is a two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange, minimizing the interconnection between devices. This bus is suitable for applications requiring occasional communications over a short distance between many devices. The flexible I2C standard allows additional devices to be connected to the bus for expansion and system development. See the connection diagram in the figure below.

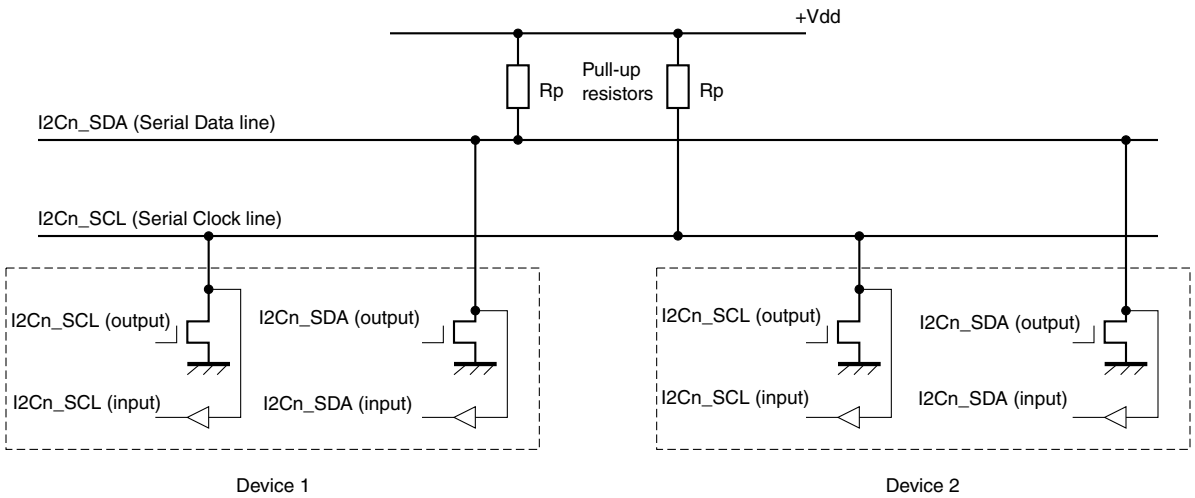


Figure 36-1. Connection of devices to I2C bus

The I2C interface speed is dependent on the I2C bus loading and timing characteristics. For pin requirement details, see *The I2C Bus Specification*. The I2C system is a true multimaster bus including arbitration and collision detection that prevents data corruption if multiple devices attempt to control the bus simultaneously. This feature supports complex applications with multiprocessor control and can be used for rapid testing and alignment of end products through external connections to an assembly-line computer. The figure below shows the block diagram of I2C.

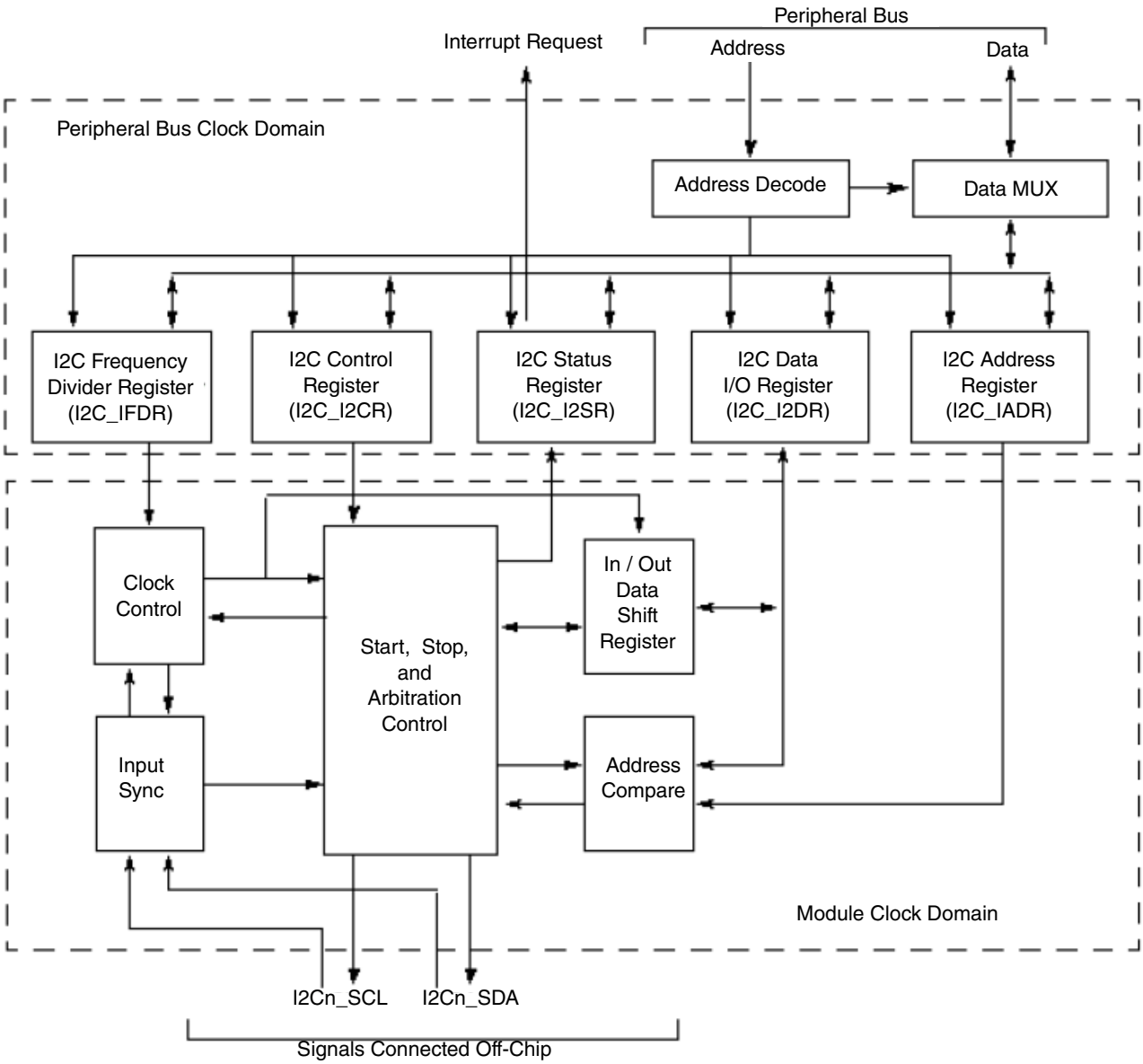


Figure 36-2. I2C block diagram

36.1.1 Features

The I2C has the following key features:

- Compatibility with I2C bus standard
- Multimaster operation
- Software programmability for one of 64 different serial clock frequencies
- Software-selectable acknowledge bit
- Interrupt-driven, byte-by-byte data transfer
- Arbitration-lost interrupt with automatic mode switching from master to slave

- Calling address identification interrupt
- Start and stop signal generation/detection
- Repeated Start signal generation
- Acknowledge bit generation/detection
- Bus-busy detection

36.1.2 Modes and operations

The I2C operates primarily in two functional modes: Standard mode and Fast mode.

- In Standard mode, I2C supports the data transfer rates up to 100 kbits/s.
- In Fast mode, data transfer rates up to 400 kbits/s can be achieved. Per block operation, there is no special configuration required for Fast or Standard mode. It is the data transfer rate that distinguishes Standard and Fast mode.

36.2 External Signals

This section discusses I2C signals that connect off-chip.

For I2C compliance, all devices connected to the I2Cn_SCL and I2Cn_SDA signals must have open-drain or open-collector outputs. The logic AND function is implemented on both lines with external pull-up resistors.

Inputs of I2Cn_SCL and I2Cn_SDA also need to be manually enabled by setting the SION bit in the IOMUX after the corresponding PADs are selected as I2C function.

The table below describes all I2C signals that connect off-chip.

Table 36-1. I2C External Signals

Signal	Description	Pad	Mode	Direction
I2C1_SCL (SCL)	Serial Clock	CSI0_DAT9	ALT4	IO
		EIM_D21	ALT6	
I2C1_SDA (SDA)	Serial Data	CSI0_DAT8	ALT4	IO
		EIM_D28	ALT1	
I2C2_SCL (SCL)	Serial Clock	EIM_EB2	ALT6	IO
		KEY_COL3	ALT4	
I2C2_SDA (SDA)	Serial Data	EIM_D16	ALT6	IO
		KEY_ROW3	ALT4	
I2C3_SCL (SCL)	Serial Clock	EIM_D17	ALT6	IO
		GPIO_3	ALT2	

Table continues on the next page...

Table 36-1. I2C External Signals (continued)

Signal	Description	Pad	Mode	Direction
		GPIO_5	ALT6	
I2C3_SDA (SDA)	Serial Data	EIM_D18	ALT6	IO
		GPIO_6	ALT2	
		GPIO_16	ALT6	
I2C4_SCL (SCL)	Serial Clock	ENET_TX_EN	ALT9	IO
		GPIO_7	ALT8	
		NANDF_WP_B	ALT9	
I2C4_SDA (SDA)	Serial Data	ENET_TXD1	ALT9	IO
		GPIO_8	ALT8	
		NANDF_CS3	ALT9	

36.3 Clocks

There are two input clocks for I2C.

The following table describes the clock sources for I2C. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 36-2. I2C Clocks

Clock name	Clock Root	Description
ipg_clk_patref	perclk_clk_root	Module clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

- Peripheral clock: This clock is used for peripheral bus register read/writes.
- Module clock: This is the functional clock of the I2C. The serial bit clock frequency is derived from the module clock. The module clock and peripheral clocks are synchronous with each other. The minimum frequency of the module clock should be 12.8 MHz for Fast mode to achieve 400-kbps operation.

36.4 Functional description

This section provides a complete functional description of the block.

36.4.1 I2C system configuration

After a reset, the I2C defaults to Slave Receive operations. Thus, when not operating as a master or responding to a slave transmit address, the I2C defaults to the Slave Receive state.

For exceptions, see Initialization sequence.

NOTE

The I2C is designed to be compatible with the Philips™ I2C bus protocol. For information on system configuration, protocol, and restrictions, see the *I2C Bus Specification*, version 2.1, by Philips Semiconductors. The I2C supports Standard and Fast modes only.

36.4.2 Arbitration procedure

If multiple devices simultaneously request the bus, the bus clock is determined by a synchronization procedure in which the low period equals the longest clock-low period among the devices, and the high period equals the shortest. A data arbitration procedure determines the relative priority of competing devices.

A device loses arbitration if it sends logic high while another sends logic low; it immediately switches to Slave Receive mode and stops driving I2Cn_SDA. In this case, the transition from master to Slave mode does not generate a Stop condition. Meanwhile, hardware sets the arbitration lost bit in the I2C Status register (I2C_I2SR[IAL] to indicate loss of arbitration).

36.4.3 Clock synchronization

Because wire-AND logic is used, a high-to-low transition on SCL affects devices connected to the bus. Devices start counting their low period when the master drives SCL low. When a device clock goes low, it holds SCL low until the Clock High state is reached. However, the low-to-high change in this device clock may not change the state of SCL if another device clock is still in its low period. Therefore, the device with the longest low period holds the synchronized clock SCL low.

Devices with shorter low periods enter a High Wait state during this time (see Figure 36-3). When all devices involved have counted off their low periods, the synchronized clock SCL is released and pulled high. There is then no difference between device clocks and the state of SCL, so all of the devices start counting their high periods. The first device to complete its high period pulls SCL low again.

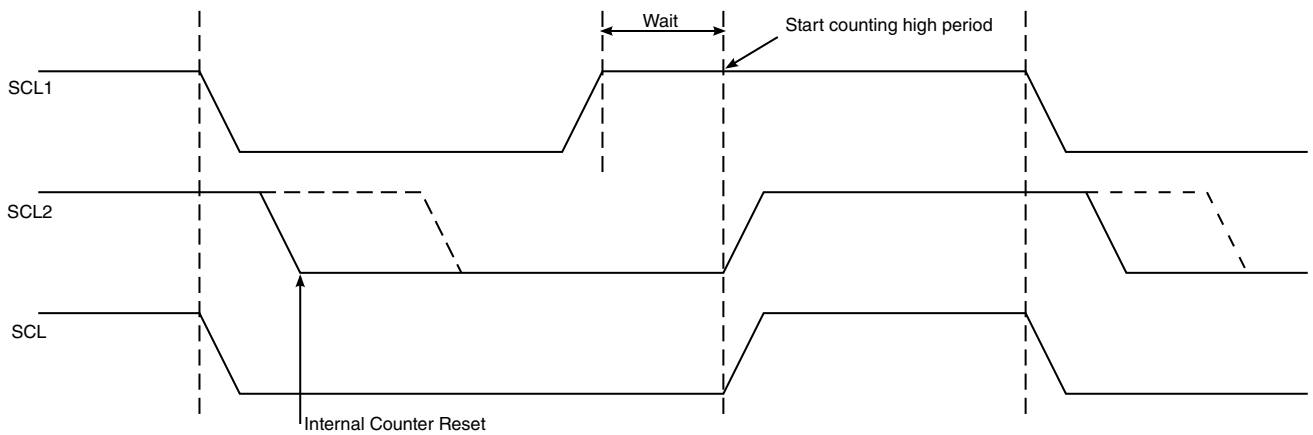


Figure 36-3. Synchronized clock SCL

36.4.4 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfers. Slave devices can hold SCL low after completing one byte transfer (9 bits). In such a case, the clock mechanism halts the bus clock and forces the master clock into a Wait state until the slave releases SCL.

36.4.5 Clock stretching

Slaves can use the clock synchronization mechanism to slow down the transfer bit rate. After the master has driven SCL low, the slave can drive SCL low for the required period and then release it. If the slave SCL low period is longer than the master SCL low period, the resulting SCL bus signal low period is stretched.

36.4.6 Peripheral bus accesses

I2C is a 16-bit block. Only half-word accesses should be performed to the block.

36.4.7 Generation of transfer error on IP bus

If an address is received on the peripheral slave bus interface but it is not implemented, an access error is generated.

36.4.8 Reset

The I2C can be reset in the following ways:

- Global reset: A hard asynchronous reset of the whole I2C
- Software reset: An internal reset for the whole I2C (except for I2C_IADR and I2C_IFDR registers) initiated by deasserting the I2C_I2CR[IEN] bit

36.4.9 Interrupts

There is only one interrupt from the block, which is enabled by setting the I2C_I2CR[IIEN] bit.

The interrupt is generated in any one of the following conditions:

- One byte transfer is completed (the interrupt is set at the falling edge of the ninth clock).
- An address is received that matches its own specific address in Slave Receive mode.
- Arbitration is lost.

36.4.10 Byte order

The block only supports the Little-Endian mode.

36.5 Initialization

NOTE

Ensure the input select pins for IOMUXC are configured correctly for I2C.

36.5.1 Initialization sequence

Before the interface can transfer serial data, registers must be initialized, as listed here.

1. Set the data sampling rate (I2C_IFDR[IC] to obtain SCL frequency from the system bus clock.
2. Update the address in the (I2C_IADR) to define its slave address (address can range from 0 to 0x7f).
3. Set the I2C enable bit (I2C_I2CR[IEN]) to enable the I2C bus interface system.
4. Modify the bits in the I2C_I2CR to select Master/Slave mode, Transmit/Receive mode, and Interrupt-Enable or not.

36.5.2 Generation of Start

After completion of the initialization procedure, serial data can be transmitted by selecting the Master Transmit mode. On a multimaster bus system, the busy bus (I2C_I2SR[IBB]) must be tested to determine whether the serial bus is free. If the bus is free (IBB = 0), the Start signal and the first byte (the slave address) can be sent. The data written to the data register comprises the address of the desired slave and the LSB indicates the transfer direction.

The free time between a Stop and the next Start condition is built into the hardware that generates the Start cycle. Depending on the relative frequencies of the system clock and the SCL period, it may be necessary to wait until the I2C is not busy after writing the calling address to the data register (I2C_I2DR), before proceeding to load data into the data register (I2C_I2DR).

36.5.3 Post-transfer software response

Sending or receiving a byte sets the data transferring bit (I2C_I2SR[ICF]), which indicates one byte of communication is finished. Upon completion, the interrupt status (I2C_I2SR[IIF]) is also set. An external interrupt is generated if the interrupt enable (I2C_I2CR[IIEN]) is set. The software must first clear the interrupt status (I2C_I2SR[IIF]) in the interrupt routine.

See the flow chart in Figure 36-5.

The data transferring bit (I2C_I2SR[ICF]) is cleared either by reading from I2C_I2DR in Receive mode or by writing to this register in Transmit mode.

The software can service the I2C I/O in the main program by monitoring the interrupt status (I2C_I2SR[IIF]) if the interrupt enable is deasserted. In this case, the interrupt status should be polled in the data transferring bit (I2C_I2SR[ICF]) because the operation is different when arbitration is lost.

When an interrupt occurs at the end of the address cycle, the master is always in Transmit mode; that is, the address is sent. If Master Receive mode is required, then I2C_I2CR[MTX] should be toggled and a dummy read of the I2C_I2DR register must be executed to trigger receive data.

During Slave-mode address cycles (I2C_I2SR[IAAS] = 1), the slave read/write bit I2C_I2SR[SRW] is read to determine the direction of the next transfer. The transmit/receive bit (I2C_I2CR[MTX]) should also be programmed accordingly. For Slave-mode data cycles (IAAS = 0), SRW is invalid. MTX should be read to determine the current transfer direction.

36.5.4 Generation of Stop

A data transfer ends when the master signals a Stop, which can occur after all data is sent.

For a master receiver to terminate a data transfer, it must inform the slave transmitter by not acknowledging the last data byte. This is done by setting the transmit acknowledge bit (I2C_I2CR[TXAK]) before reading the next-to-last byte. Before the last byte is read, a Stop signal must be generated.

36.5.5 Generation of Repeated Start

After the data transfer, if the master still requires the bus, it can signal another Start followed by another slave address without signaling a Stop.

36.5.6 Slave mode

In the slave interrupt service routine (see Figure 36-5), the block addressed as slave bit (IAAS) should be tested to check if a calling of its own address has just been received. If IAAS is set, software should set the Transmit/Receive mode select bit (I2C_I2CR[MTX]) according to the I2C_I2SR[SRW]. Writing to the I2C_I2CR clears the IAAS automatically. The only time IAAS is read as set is from the interrupt at the end of the address cycle where an address match occurred; interrupts resulting from subsequent data transfers will have IAAS cleared. A data transfer can now be initiated by writing information to I2C_I2DR for slave transmits, or read from I2C_I2DR in Slave Receive mode. A dummy read of I2C_I2DR in Slave Receive mode releases SCL, allowing the master to send data.

In the slave transmitter routine, the receive acknowledge bit (I2C_I2SR[RXAK]) must be tested before sending the next byte of data. Setting RXAK means an end-of-data signal from the master receiver, after which the software must switch it from Transmit to Receiver mode. Reading the data register (I2C_I2DR) then releases SCL so the master can generate a Stop signal.

36.5.7 Arbitration lost

If several devices try to engage the bus at the same time, one becomes master. Hardware immediately switches devices that lose arbitration to Slave Receive mode. Data output to 12Cn_SDA stops, but 12Cn_SCL is still generated until the end of the byte during which arbitration is lost. An interrupt occurs at the falling edge of the ninth clock of this transfer if the arbitration is lost (I2C_I2SR[IAL] = 1), and the Slave mode is selected (I2C_I2CR[MSTA] = 0).

See the flow chart in Figure 36-5.

If a device that is not a master tries to transmit or do a Start, hardware inhibits the transmission, clears MSTA without signaling a Stop, generates an interrupt to the Arm platform, and sets I2C_I2SR[IAL] to indicate a failed attempt to engage the bus. When considering these cases, the slave service routine should first test I2C_I2SR[IAL], and the software should clear it if it is set.

For Multimaster mode, when an I2C is enabled when the bus is busy and asserts Start, the I2C_I2SR[IAL] bit gets set only for 12Cn_SDA=0, 12Cn_SCL=0/1, 12Cn_SDA=1, and 12Cn_SCL=0; but not for 12Cn_SDA=1 and I2Cn_SCL=1, which is the equivalent of Bus Idle state.

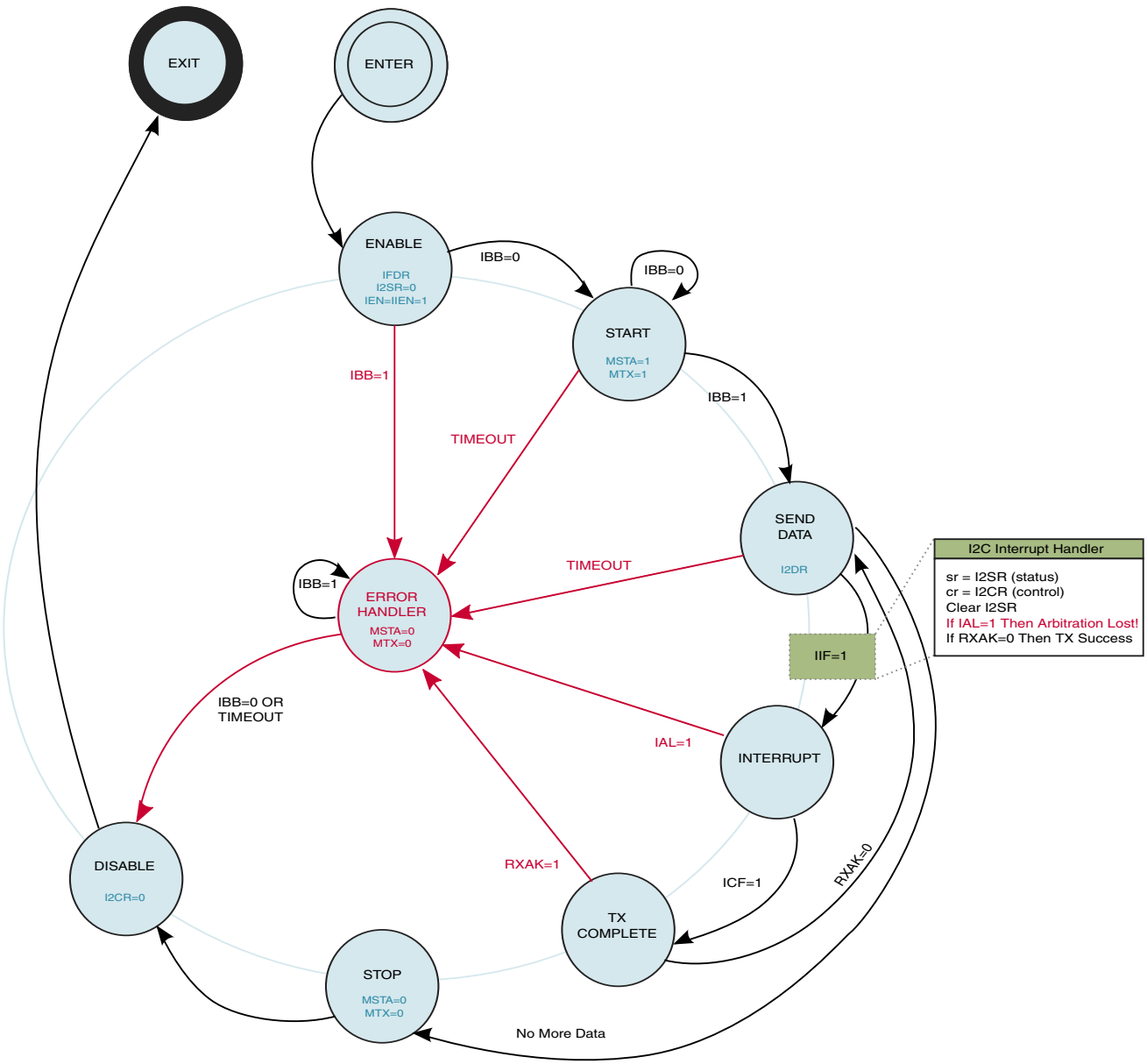


Figure 36-4. I2C Programming state diagram

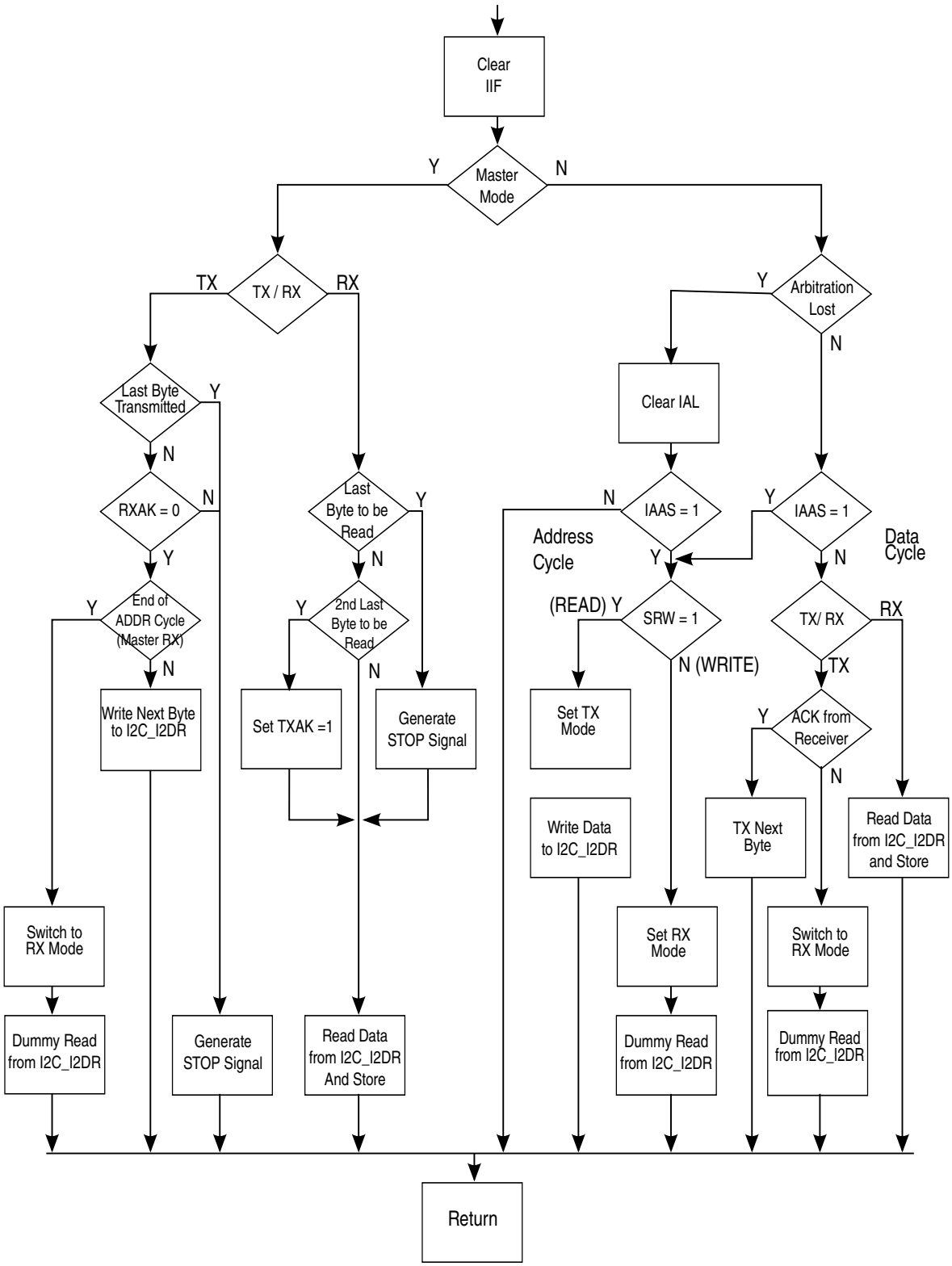


Figure 36-5. Flowchart of typical I2C interrupt routine

NOTE

For a Repeated Start only, the Stop-generation stage does not occur in Master mode. A loop repeats itself without stopping for the next start.

For Master Receive mode, I2C is programmed as Master Transmit during Address mode and after slave address transfer; the MTX bit should be cleared and a dummy read on the I2C_I2DR register should be performed so I2C can read the next receive data.

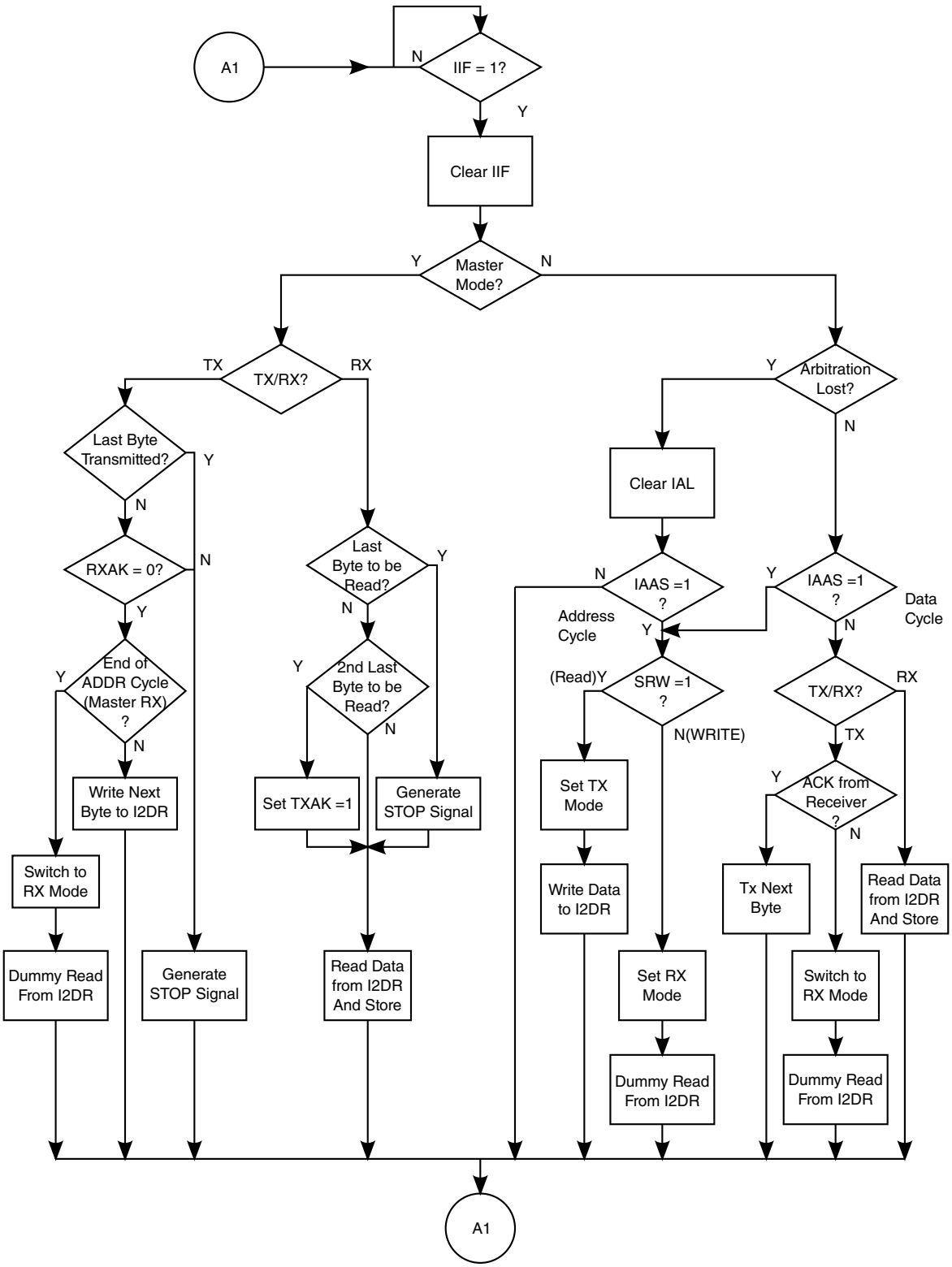


Figure 36-6. Flowchart for typical I2C polling routine

NOTE

The timeout value depends on the bus frequency at which I2C is operating. The minimum timeout for polling the IIF bit at a maximum I2C bus frequency of 400 kHz is T_{min} = 25 µs (=2.5 x 10 µs). This value can be calculated for any bus frequency. The formula is $T_{min} = 10/F_{SCL}$, where F_{SCL} is the frequency of the I2C clock (SCL).

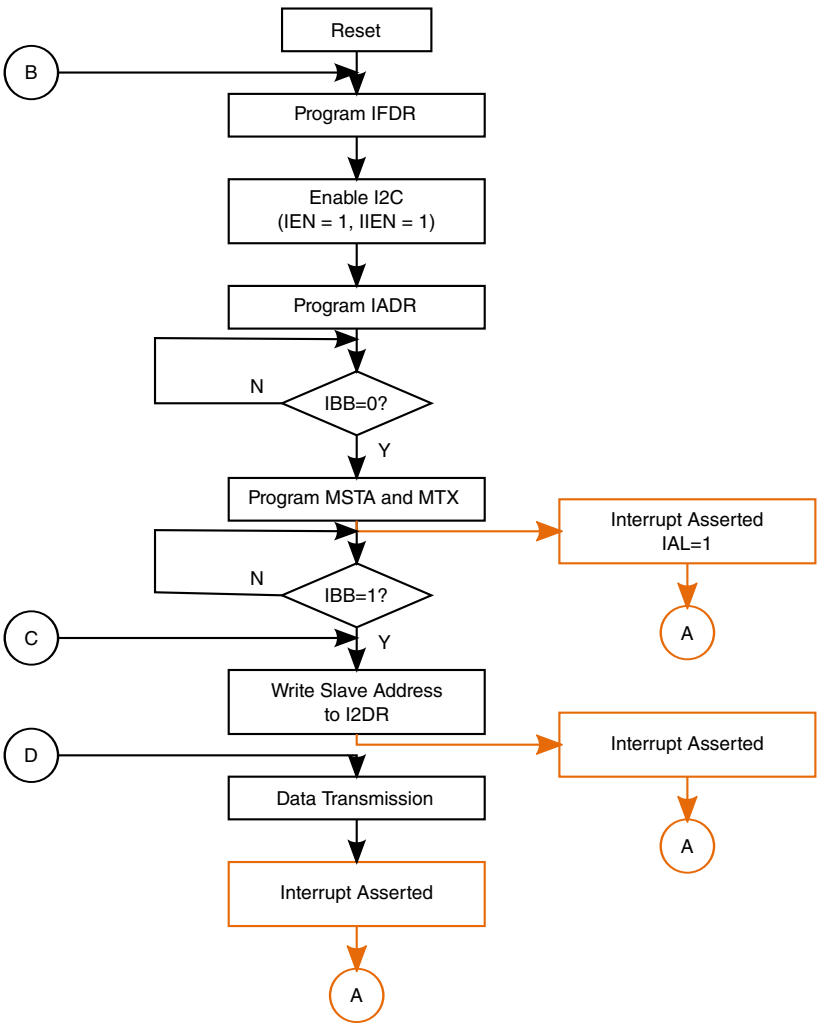


Figure 36-7. Detailed flowchart of a typical I2C Master Transmit mode, part 1

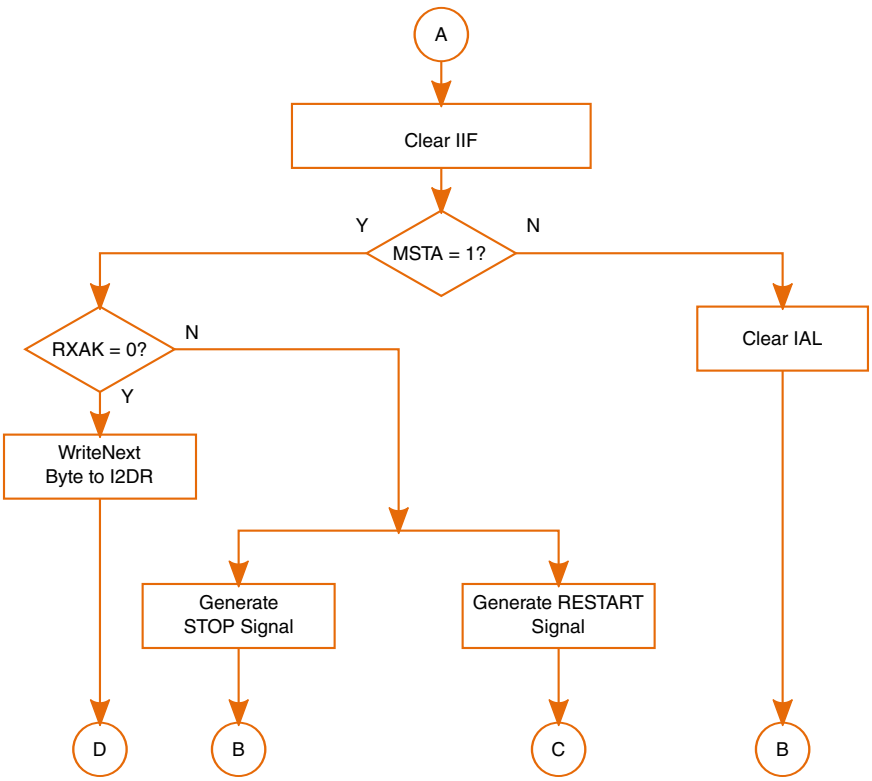


Figure 36-8. Detailed flowchart of a typical I2C Master Transmit mode, part 2

Figure 36-7 and Figure 36-8 show the Master Transmit mode operation with interrupt subroutine. If an interrupt is generated and the MSTA bit is 0, then bus arbitration is lost and IAL is set. Software can clear the IAL bit and reprogram I2C. If the MSTA bit is 1, then it is a transfer-generated interrupt. In this case, software can check the RXAK bit for a data receive acknowledgement by the slave and, accordingly, decide to do one of the following:

- Generate a STOP
- Generate a REPEATED START by writing to the I2C_I2CR register
- Perform the next data transfer by writing to the I2C_I2DR register

NOTE

The IBB bit is asserted by a Start condition on the bus, and it is deasserted by a Stop condition on the bus. Therefore, if arbitration is lost due to an unexpected Stop condition during transfer, then IBB is cleared. If arbitration is lost due to a data mismatch, then it is not cleared. Software should always clear the IEN bit and then set it if arbitration is lost.

36.6 Software restriction

Software should ensure that there is a delay of at least two module clock cycles after it sets the I2C_I2CR[RSTA] bit and before writing to the I2C_I2DR register. The maximum possible clock period of the module clock is 78 ns.

36.7 I2C Memory Map/Register Definition

The I2C contains five 16-bit registers.

NOTE

Registers at offsets 0x0002, 0x0006, 0x000A, and 0x000E are reserved for future additions.

I2C memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21A_0000	I2C Address Register (I2C1_IADR)	16	R/W	0000h	36.7.1/1993
21A_0004	I2C Frequency Divider Register (I2C1_IFDR)	16	R/W	0000h	36.7.2/1993
21A_0008	I2C Control Register (I2C1_I2CR)	16	R/W	0000h	36.7.3/1995
21A_000C	I2C Status Register (I2C1_I2SR)	16	R/W	0081h	36.7.4/1996
21A_0010	I2C Data I/O Register (I2C1_I2DR)	16	R/W	0000h	36.7.5/1998
21A_4000	I2C Address Register (I2C2_IADR)	16	R/W	0000h	36.7.1/1993
21A_4004	I2C Frequency Divider Register (I2C2_IFDR)	16	R/W	0000h	36.7.2/1993
21A_4008	I2C Control Register (I2C2_I2CR)	16	R/W	0000h	36.7.3/1995
21A_400C	I2C Status Register (I2C2_I2SR)	16	R/W	0081h	36.7.4/1996
21A_4010	I2C Data I/O Register (I2C2_I2DR)	16	R/W	0000h	36.7.5/1998
21A_8000	I2C Address Register (I2C3_IADR)	16	R/W	0000h	36.7.1/1993
21A_8004	I2C Frequency Divider Register (I2C3_IFDR)	16	R/W	0000h	36.7.2/1993
21A_8008	I2C Control Register (I2C3_I2CR)	16	R/W	0000h	36.7.3/1995
21A_800C	I2C Status Register (I2C3_I2SR)	16	R/W	0081h	36.7.4/1996
21A_8010	I2C Data I/O Register (I2C3_I2DR)	16	R/W	0000h	36.7.5/1998
21F_8000	I2C Address Register (I2C4_IADR)	16	R/W	0000h	36.7.1/1993
21F_8004	I2C Frequency Divider Register (I2C4_IFDR)	16	R/W	0000h	36.7.2/1993
21F_8008	I2C Control Register (I2C4_I2CR)	16	R/W	0000h	36.7.3/1995
21F_800C	I2C Status Register (I2C4_I2SR)	16	R/W	0081h	36.7.4/1996
21F_8010	I2C Data I/O Register (I2C4_I2DR)	16	R/W	0000h	36.7.5/1998

36.7.1 I2C Address Register (I2Cx_IADR)

Address: Base address + 0h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								ADR							0
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Cx_IADR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value 0.
7–1 ADR	Slave address. Contains the specific slave address to be used by the I2C. Slave mode is the default I2C mode for an address match on the bus. **NOTE:** The I2C_IADR holds the address to which the I2C responds when addressed as a slave. The slave address is not the address sent on the bus during the address transfer. The register is not reset by a software reset.
0 Reserved	This read-only field is reserved and always has the value 0.

36.7.2 I2C Frequency Divider Register (I2Cx_IFDR)

The I2C_IFDR provides a programmable prescaler to configure the clock for bit-rate selection. The register does not get reset by a software reset.

I2C clock is sourced from PERCLK_ROOT which is routed from IPG_CLK_ROOT. I2C clock frequency can easily obtained by using the following formula:

I2C clock Frequency = (PERCLK_ROOT frequency)/(division factor corresponding to IFDR)

By default, IPG_CLK_ROOT and PERCLK_ROOT frequencies are set to 49.5 MHz, where the root clock is sourced from PLL2's PFD2. Obtaining the frequencies can be accomplished by:

PLL2 = 528 MHz

PLL2_PFD2 = 528 MHz * 18 / 24 = 396 MHz

IPG_CLK_ROOT = (PLL2_PFD2 / ahb_podf)/ ipg_podf = (396 MHz/4)/2 = 49.5 MHz

PER_CLK_ROOT = IPG_CLK_ROOT/perclk_podf = 49.5 MHz/1 = 49.5 MHz

NOTE

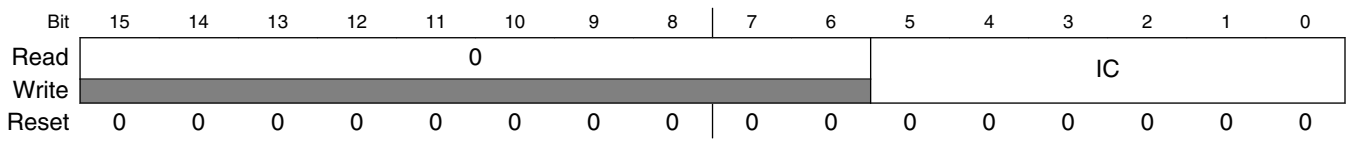
The above calculation assumes that the default CCM register settings, routing, and division factors are used. If different routing, PFD values, and/or division factors are used, the user must adjust the parameters accordingly to calculate the correct clock frequency.

The following table describes the divider and register values for the register field "IC."

Table 36-3. I2C_IFDR Register Field Values

IC	Divider		IC	Divider		IC	Divider		IC	Divider
0x00	30		0x10	288		0x20	22		0x30	160
0x01	32		0x11	320		0x21	24		0x31	192
0x02	36		0x12	384		0x22	26		0x32	224
0x03	42		0x13	480		0x23	28		0x33	256
0x04	48		0x14	576		0x24	32		0x34	320
0x05	52		0x15	640		0x25	36		0x35	384
0x06	60		0x16	768		0x26	40		0x36	448
0x07	72		0x17	960		0x27	44		0x37	512
0x08	80		0x18	1152		0x28	48		0x38	640
0x09	88		0x19	1280		0x29	56		0x39	768
0x0A	104		0x1A	1536		0x2A	64		0x3A	896
0x0B	128		0x1B	1920		0x2B	72		0x3B	1024
0x0C	144		0x1C	2304		0x2C	80		0x3C	1280
0x0D	160		0x1D	2560		0x2D	96		0x3D	1536
0x0E	192		0x1E	3072		0x2E	112		0x3E	1792
0x0F	240		0x1F	3840		0x2F	128		0x3F	2048

Address: Base address + 4h offset



I2Cx_IFDR field descriptions

Field	Description
15–6 Reserved	This read-only field is reserved and always has the value 0.
IC	I2C clock rate. Prescales the clock for bit-rate selection. Due to potentially slow I2Cn_SCL and I2Cn_SDA rise and fall times, bus signals are sampled at the prescaler frequency. The serial bit clock frequency may be lower than IPG_CLK_ROOT divided by the divider shown in the I2C Data I/O Register. **NOTE:** The IC value should not be changed during the data transfer, however, it can be changed before a Repeat Start or Start programming sequence in I2C. The I2C protocol supports bit rates of up to 400 kbps. The IC bits need to be programmed in accordance with this constraint.

36.7.3 I2C Control Register (I2Cx_I2CR)

The I2C_I2CR is used to enable the I2C and the I2C interrupt. It also contains bits that govern operation as a slave or a master.

Address: Base address + 8h offset

Bit	15	14	13	12	11	10	9	8
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	IEN	IIEN	MSTA	MTX	TXAK	0	0	
Write						RSTA		
Reset	0	0	0	0	0	0	0	0

I2Cx_I2CR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value 0.
7 IEN	I2C enable. Also controls the software reset of the entire I2C. Resetting the bit generates an internal reset to the block. If the block is enabled in the middle of a byte transfer, Slave mode ignores the current bus transfer and starts operating when the next Start condition is detected. Master mode is not aware that the bus is busy, so initiating a start cycle may corrupt the current bus cycle, ultimately causing either the current master or the I2C to lose arbitration. Subsequently, bus operation returns to normal. 0 The block is disabled, but registers can still be accessed. 1 The I2C is enabled. This bit must be set before any other I2C_I2CR bits have an effect.
6 IIEN	I2C interrupt enable. **NOTE:** If data is written during the Start condition, that is, just after setting the I2C_I2CR[MSTA] and I2C_I2CR[MTX] bits, then the ICF bit is cleared at the falling edge of SCLK after Start. If data is written after the Start condition and falling edge of SCLK, then the ICF bit is cleared as soon as data is written. 0 I2C interrupts are disabled, but the status flag I2C_I2SR[IIF] continues to be set when an Interrupt condition occurs. 1 I2C interrupts are enabled. An I2C interrupt occurs if I2C_I2SR[IIF] is also set.
5 MSTA	Master/Slave mode select bit. If the master loses arbitration, MSTA is cleared without generating a Stop signal. **NOTE:** The module clock should be on for writing to the MSTA bit. **NOTE:** The MSTA bit is cleared by software to generate a Stop condition; it can also be cleared by hardware when the I2C loses the bus arbitration.

Table continues on the next page...

I2Cx_I2CR field descriptions (continued)

Field	Description
	0 Slave mode. Changing MSTA from 1 to 0 generates a Stop and selects Slave mode. 1 Master mode. Changing MSTA from 0 to 1 signals a Start on the bus and selects Master mode.
4 MTX	Transmit/Receive mode select bit. Selects the direction of master and slave transfers. 0 Receive. When a slave is addressed, the software should set MTX according to the slave read/write bit in the I2C status register (I2C_I2SR[SRW]). 1 Transmit. In Master mode, MTX should be set according to the type of transfer required. Therefore, for address cycles, MTX is always 1.
3 TXAK	Transmit acknowledge enable. Specifies the value driven onto I2Cn_SDA during acknowledge cycles for both master and slave receivers. **NOTE:** Writing TXAK applies only when the I2C bus is a receiver. 0 An acknowledge signal is sent to the bus at the ninth clock bit after receiving one byte of data. 1 No acknowledge signal response is sent (that is, the acknowledge bit = 1).
2 RSTA	Repeat start. Always reads as 0. Attempting a repeat start without bus mastership causes loss of arbitration. 0 No repeat start 1 Generates a Repeated Start condition
Reserved	This read-only field is reserved and always has the value 0.

36.7.4 I2C Status Register (I2Cx_I2SR)

The I2C_I2SR contains bits that indicate transaction direction and status.

Address: Base address + Ch offset

Bit	15	14	13	12	11	10	9	8
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	ICF	IAAS	IBB	IAL	0	SRW	IIF	RXAK
Write								
Reset	1	0	0	0	0	0	0	1

I2Cx_I2SR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value 0.
7 ICF	Data transferring bit. While one byte of data is transferred, ICF is cleared. 0 Transfer is in progress. 1 Transfer is complete. This bit is set by the falling edge of the ninth clock of the last byte transfer.
6 IAAS	I2C addressed as a slave bit. The Arm platform is interrupted if the interrupt enable (I2C_I2CR[IIEN]) is set. The Arm platform must check the slave read/write bit (SRW) and set its Transfer/Receive mode accordingly. Writing to I2C_I2CR clears this bit. 0 Not addressed 1 Addressed as a slave. Set when its own address (I2C_IADR) matches the calling address.
5 IBB	I2C bus busy bit. Indicates the status of the bus. **NOTE:** When I2C is enabled (I2C_I2CR[IEN] = 1), it continuously polls the bus data (SDA) and clock (SCL) signals to determine a Start or Stop condition. 0 Bus is idle. If a Stop signal is detected, IBB is cleared. 1 Bus is busy. When Start is detected, IBB is set.
4 IAL	Arbitration lost. Set by hardware in the following circumstances (IAL must be cleared by software by writing a "0" to it at the start of the interrupt service routine): • I2Cn_SDA input samples low when the master drives high during an address or data-transmit cycle. • I2Cn_SDA input samples low when the master drives high during the acknowledge bit of a data-receive cycle. For the above two cases, the bit is set at the falling edge of the ninth I2Cn_SCL clock during the ACK cycle. • A Start cycle is attempted when the bus is busy. • A Repeated Start cycle is requested in Slave mode. • A Stop condition is detected when the master did not request it. **NOTE:** Software cannot set the bit. 0 No arbitration lost. 1 Arbitration is lost.
3 Reserved	This read-only field is reserved and always has the value 0.
2 SRW	Slave read/write. When the I2C is addressed as a slave, IAAS is set, and the slave read/write bit (SRW) indicates the value of the R/W command bit of the calling address sent from the master. SRW is valid only when a complete transfer has occurred, no other transfers have been initiated, and the I2C is a slave and has an address match. 0 Slave receive, master writing to slave 1 Slave transmit, master reading from slave
1 IIF	I2C interrupt. Must be cleared by the software by writing a "0" to it in the interrupt routine. **NOTE:** The software cannot set the bit. 0 No I2C interrupt pending. 1 An interrupt is pending.

Table continues on the next page...

I2Cx_I2SR field descriptions (continued)

Field	Description
	This causes a processor interrupt request (if the interrupt enable is asserted [IIEN = 1]). The interrupt is set when one of the following occurs: • One byte transfer is completed (the interrupt is set at the falling edge of the ninth clock). • An address is received that matches its own specific address in Slave Receive mode. • Arbitration is lost.
0 RXAK	Received acknowledge. This is the value received from the I2Cn_SDA input for the acknowledge bit during a bus cycle. 0 An "acknowledge" signal was received after the completion of an 8-bit data transmission on the bus. 1 A "No acknowledge" signal was detected at the ninth clock.

36.7.5 I2C Data I/O Register (I2Cx_I2DR)

In Master Receive mode, reading the data register allows a read to occur and initiates the next byte to be received. In Slave mode, the same function is available after it is addressed.

Address: Base address + 10h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								DATA							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Cx_I2DR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value 0.
DATA	Data Byte. Holds the last data byte received or the next data byte to be transferred. Software writes the next data byte to be transmitted or reads the data byte received. **NOTE:** The core-written value in I2C_I2DR cannot be read back by the core. Only data written by the I2C bus side can be read.

Chapter 37
IOMUX Controller (IOMUXC)

37.1 Overview

The IOMUX Controller (IOMUXC), together with the IOMUX, enables the IC to share one pad to several functional blocks. This sharing is done by multiplexing the pad's input and output signals.

Every module requires a specific pad setting (such as pull up or keeper), and for each pad, there are up to 8 muxing options (called ALT modes). The pad settings parameters are controlled by the IOMUXC.

The IOMUX consists only of combinatorial logic combined from several basic IOMUX cells. Each basic IOMUX cell handles only one pad signal's muxing.

Figure 37-1 illustrates the IOMUX/IOMUXC connectivity in the system.

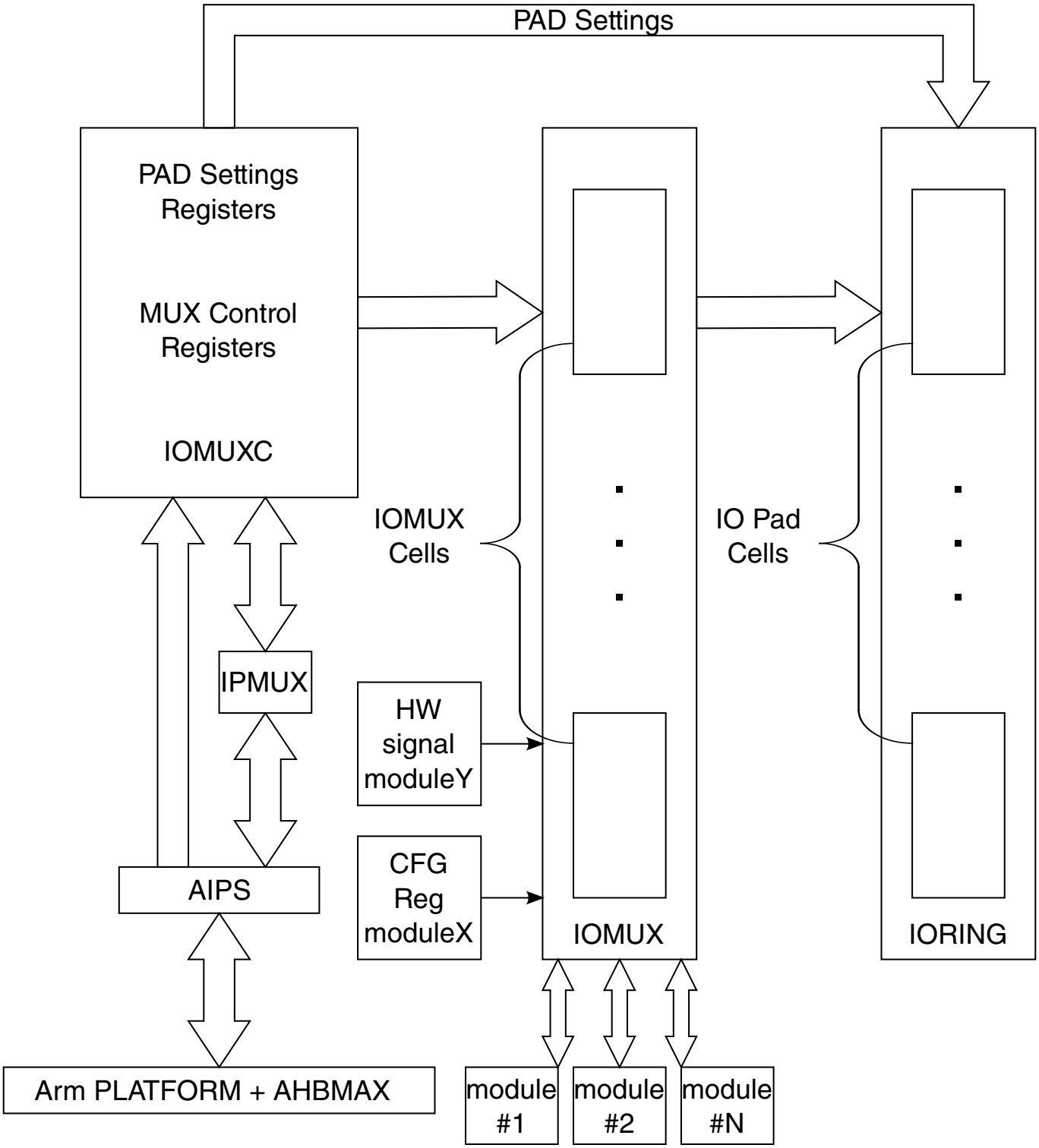


Figure 37-1. IOMUX SoC Level Block Diagram

37.1.1 Features

The IOMUXC features are:

- 32-bit software mux control registers (IOMUXC_SW_MUX_CTL_PAD_<PAD NAME> or IOMUXC_SW_MUX_CTL_GRP_<GROUP NAME>) to configure 1 of 8 alternate (ALT) MUX_MODE feilds of each pad or a predefined group of pads and to enable the forcing of an input path of the pad(s) (SION bit).
- 32-bit software pad control registers (IOMUXC_SW_PAD_CTL_PAD_<PAD_NAME> or IOMUXC_SW_PAD_CTL_GRP_<GROUP NAME>) to configure specific pad settings of each pad, or a predefined group of pads.
- 32-bit general purpose registers - several (GPR0 to GPR*n*) 32-bit registers according to SoC requirements for any usage.
- 32-bit input select control registers to control the input path to a module when more than one pad drives this module input.

Each SW MUX/PAD CTL IOMUXC register handles only one pad or one pad's group.

Only the minimum number of registers required by software are implemented by hardware. For example, if only ALT0 and ALT1 modes are used on Pad x then only one bit register will be generated as the MUX_MODE control field in the software mux control register of Pad x.

The software mux control registers may allow the forcing of pads to become input (input path enabled) regardless of the functional direction driven. This may be useful for loopback and GPIO data capture.

37.2 Clocks

The table found here describes the clock sources for IOMUXC.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 37-1. IOMUXC Clocks

Clock name	Clock Root	Description
ipt_clk_io	enfc_clk_root	IO clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

37.3 Functional description

This section provides a complete functional description of the block.

The IOMUXC consists of two sub-blocks:

- IOMUXC_REGISTERS includes all of the IOMUXC registers (see Features).
- IOMUXC_LOGIC includes all of the IOMUXC combinatorial logic (IP interface controls, address decoder, observability muxes).

The IOMUX consists of a number (about the number of pads in the SoC) of basic iomux_cell units. If only one functional mode is required for a specific pad, there is no need for IOMUX and the signals can be connected directly from the module to the I/O. The IOMUX cell is required whenever two or more functional modes are required for a specific pad or when one functional mode and the one test mode are required.

The basic iomux_cell design, which allows two levels of HW signal control (in ALT6 and ALT7 modes - ALT7 gets highest priority) is shown in Figure 37-2.

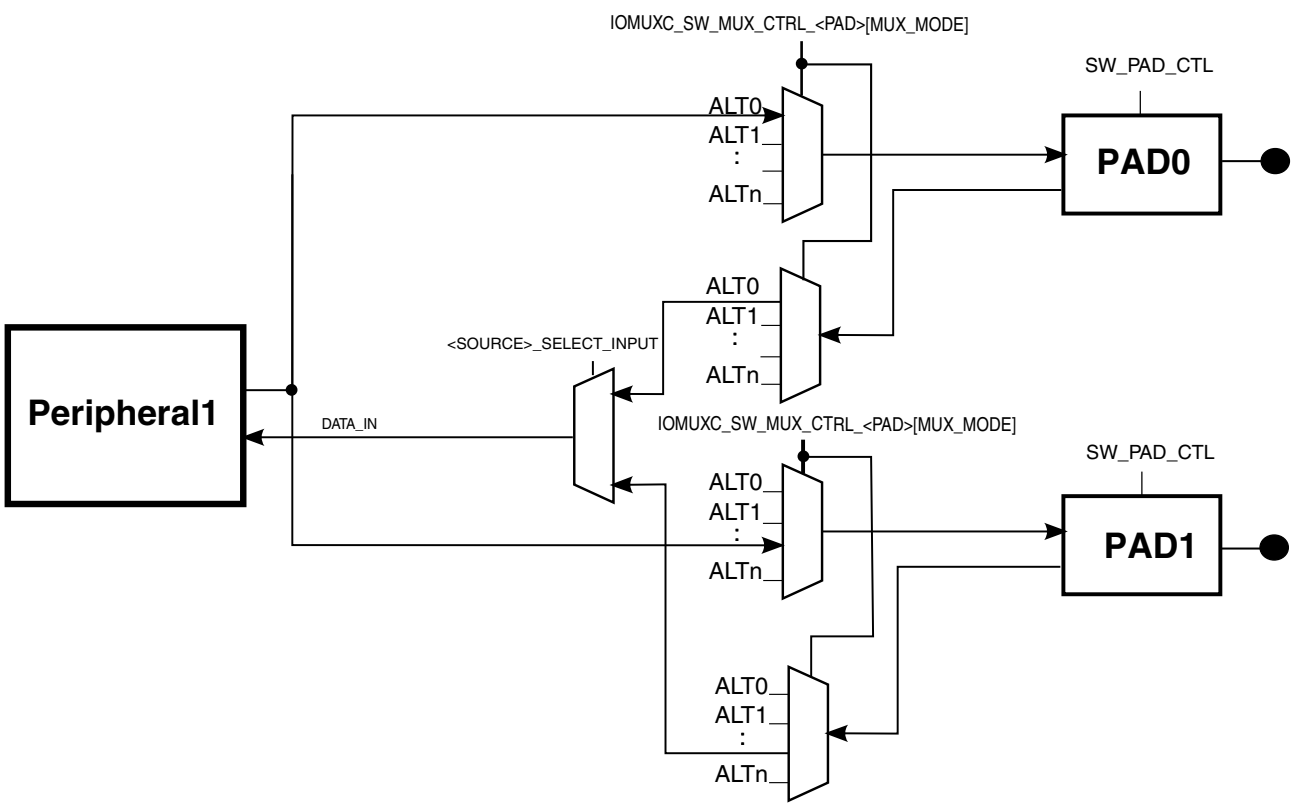


Figure 37-2. IOMUX Cell Block Diagram

37.3.1 ALT6 and ALT7 extended muxing modes

The ALT7 and ALT6 extended muxing modes allow any signal in the system (such as fuse, pad input, JTAG, or software register) to override any software configuration and to force the ALT6/ALT7 muxing mode.

It also allows an IOMUX software register to control a group of pads.

37.3.2 SW Loopback through SION bit

A limited option exists to override the default pad functionality and force the input path to be active (ipp_ibe==1'b1) regardless of the value driven by the corresponding module. This can be done by setting the SION (Software Input On) bit in the IOMUXC_SW_MUX_CTL register (when available) to "1".

Uses include:

- LoopBack - Module x drives the pad and also receives pad value as an input.
- GPIO Capture - Module x drives the pad and the value is captured by GPIO.

37.3.3 Daisy chain - multi pads driving same module input pin

In some cases, more than one pad may drive a single module input pin. Such cases require the addition of one more level of IOMUXing; all of these input signals are muxed, and a dedicated software controlled register controls the mux in order to select the required input path.

A module port involved in "daisy chain" requires two software configuration commands, one for selecting the mode for this pad (programable via the IOMUXC_SW_MUX_CTL_<PAD> registers) and one for defining it as the input path (via the daisy chain registers).

This means that a module port involved in "daisy chain" requires two software configuration commands, one for selecting the mode for this pad (programable via the IOMUXC_SW_MUX_CTL_<PAD> registers) and one for defining it as the input path (via the daisy chain registers). The daisy chain is illustrated in the figure below.

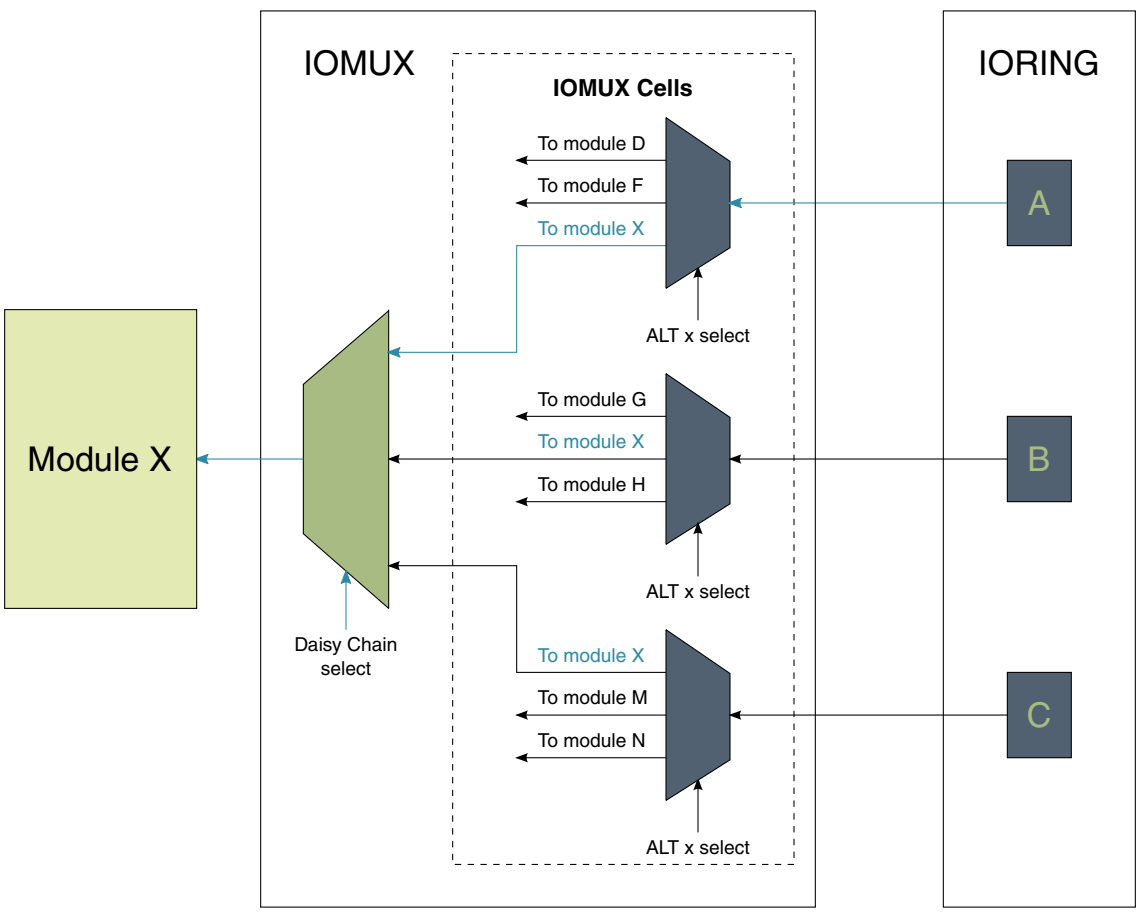


Figure 37-3. Daisy chain illustration

37.4 IOMUXC Memory Map/Register Definition

IOMUXC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0000	GPR0 (IOMUXC_GPR0)	32	R/W	0000_0000h	37.4.1/2031
20E_0004	GPR1 (IOMUXC_GPR1)	32	R/W	4840_0005h	37.4.2/2034
20E_0008	GPR2 (IOMUXC_GPR2)	32	R/W	0000_0000h	37.4.3/2037
20E_000C	GPR3 (IOMUXC_GPR3)	32	R/W	01E0_0000h	37.4.4/2039
20E_0010	GPR4 (IOMUXC_GPR4)	32	R/W	0000_0000h	37.4.5/2043
20E_0014	GPR5 (IOMUXC_GPR5)	32	R/W	0000_0000h	37.4.6/2046
20E_0018	GPR6 (IOMUXC_GPR6)	32	R/W	2222_2222h	37.4.7/2047
20E_001C	GPR7 (IOMUXC_GPR7)	32	R/W	2222_2222h	37.4.8/2048
20E_0020	GPR8 (IOMUXC_GPR8)	32	R/W	0000_0000h	37.4.9/2048
20E_0024	GPR9 (IOMUXC_GPR9)	32	R/W	0000_0000h	37.4.10/ 2050
20E_0028	GPR10 (IOMUXC_GPR10)	32	R/W	0000_3800h	37.4.11/ 2051
20E_002C	GPR11 (IOMUXC_GPR11)	32	R/W	0000_0000h	37.4.12/ 2053
20E_0030	GPR12 (IOMUXC_GPR12)	32	R/W	0F00_0000h	37.4.13/ 2053
20E_0034	GPR13 (IOMUXC_GPR13)	32	R/W	0000_0008h	37.4.14/ 2055
20E_004C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA10)	32	R/W	0000_0005h	37.4.15/ 2057
20E_0050	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA11)	32	R/W	0000_0005h	37.4.16/ 2058
20E_0054	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA12)	32	R/W	0000_0005h	37.4.17/ 2059
20E_0058	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA13)	32	R/W	0000_0005h	37.4.18/ 2060
20E_005C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA14)	32	R/W	0000_0005h	37.4.19/ 2061
20E_0060	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA15)	32	R/W	0000_0005h	37.4.20/ 2062
20E_0064	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA16)	32	R/W	0000_0005h	37.4.21/ 2063
20E_0068	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA17)	32	R/W	0000_0005h	37.4.22/ 2064
20E_006C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA18)	32	R/W	0000_0005h	37.4.23/ 2065
20E_0070	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA19)	32	R/W	0000_0005h	37.4.24/ 2066
20E_0074	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA04)	32	R/W	0000_0005h	37.4.25/ 2067

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0078	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA05)	32	R/W	0000_0005h	37.4.26/ 2068
20E_007C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA06)	32	R/W	0000_0005h	37.4.27/ 2069
20E_0080	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA07)	32	R/W	0000_0005h	37.4.28/ 2070
20E_0084	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA08)	32	R/W	0000_0005h	37.4.29/ 2071
20E_0088	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA09)	32	R/W	0000_0005h	37.4.30/ 2072
20E_008C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA_EN)	32	R/W	0000_0005h	37.4.31/ 2073
20E_0090	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_HSYNC)	32	R/W	0000_0005h	37.4.32/ 2074
20E_0094	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_PIXCLK)	32	R/W	0000_0005h	37.4.33/ 2075
20E_0098	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_VSYNC)	32	R/W	0000_0005h	37.4.34/ 2076
20E_009C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DI0_DISP_CLK)	32	R/W	0000_0005h	37.4.35/ 2077
20E_00A0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DI0_PIN15)	32	R/W	0000_0005h	37.4.36/ 2077
20E_00A4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DI0_PIN02)	32	R/W	0000_0005h	37.4.37/ 2078
20E_00A8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DI0_PIN03)	32	R/W	0000_0005h	37.4.38/ 2079
20E_00AC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DI0_PIN04)	32	R/W	0000_0005h	37.4.39/ 2079
20E_00B0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA00)	32	R/W	0000_0005h	37.4.40/ 2080
20E_00B4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA01)	32	R/W	0000_0005h	37.4.41/ 2081
20E_00B8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA10)	32	R/W	0000_0005h	37.4.42/ 2082
20E_00BC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA11)	32	R/W	0000_0005h	37.4.43/ 2082
20E_00C0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA12)	32	R/W	0000_0005h	37.4.44/ 2083
20E_00C4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA13)	32	R/W	0000_0005h	37.4.45/ 2084
20E_00C8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA14)	32	R/W	0000_0005h	37.4.46/ 2085
20E_00CC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA15)	32	R/W	0000_0005h	37.4.47/ 2086

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_00D0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA16)	32	R/W	0000_0005h	37.4.48/ 2087
20E_00D4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA17)	32	R/W	0000_0005h	37.4.49/ 2088
20E_00D8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA18)	32	R/W	0000_0005h	37.4.50/ 2089
20E_00DC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA19)	32	R/W	0000_0005h	37.4.51/ 2090
20E_00E0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA02)	32	R/W	0000_0005h	37.4.52/ 2091
20E_00E4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA20)	32	R/W	0000_0005h	37.4.53/ 2092
20E_00E8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA21)	32	R/W	0000_0005h	37.4.54/ 2093
20E_00EC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA22)	32	R/W	0000_0005h	37.4.55/ 2094
20E_00F0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA23)	32	R/W	0000_0005h	37.4.56/ 2095
20E_00F4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA03)	32	R/W	0000_0005h	37.4.57/ 2096
20E_00F8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA04)	32	R/W	0000_0005h	37.4.58/ 2097
20E_00FC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA05)	32	R/W	0000_0005h	37.4.59/ 2098
20E_0100	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA06)	32	R/W	0000_0005h	37.4.60/ 2099
20E_0104	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA07)	32	R/W	0000_0005h	37.4.61/ 2100
20E_0108	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA08)	32	R/W	0000_0005h	37.4.62/ 2101
20E_010C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA09)	32	R/W	0000_0005h	37.4.63/ 2102
20E_0110	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR16)	32	R/W	0000_0000h	37.4.64/ 2103
20E_0114	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR17)	32	R/W	0000_0000h	37.4.65/ 2104
20E_0118	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR18)	32	R/W	0000_0000h	37.4.66/ 2105
20E_011C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR19)	32	R/W	0000_0000h	37.4.67/ 2106
20E_0120	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR20)	32	R/W	0000_0000h	37.4.68/ 2107
20E_0124	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR21)	32	R/W	0000_0000h	37.4.69/ 2108

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0128	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR22)	32	R/W	0000_0000h	37.4.70/ 2109
20E_012C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR23)	32	R/W	0000_0000h	37.4.71/ 2110
20E_0130	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR24)	32	R/W	0000_0000h	37.4.72/ 2111
20E_0134	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR25)	32	R/W	0000_0000h	37.4.73/ 2112
20E_0138	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_BCLK)	32	R/W	0000_0000h	37.4.74/ 2113
20E_013C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_CS0_B)	32	R/W	0000_0000h	37.4.75/ 2113
20E_0140	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_CS1_B)	32	R/W	0000_0000h	37.4.76/ 2114
20E_0144	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA16)	32	R/W	0000_0005h	37.4.77/ 2115
20E_0148	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA17)	32	R/W	0000_0005h	37.4.78/ 2116
20E_014C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA18)	32	R/W	0000_0005h	37.4.79/ 2117
20E_0150	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA19)	32	R/W	0000_0005h	37.4.80/ 2118
20E_0154	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA20)	32	R/W	0000_0005h	37.4.81/ 2119
20E_0158	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA21)	32	R/W	0000_0005h	37.4.82/ 2120
20E_015C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA22)	32	R/W	0000_0005h	37.4.83/ 2121
20E_0160	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA23)	32	R/W	0000_0005h	37.4.84/ 2122
20E_0164	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA24)	32	R/W	0000_0005h	37.4.85/ 2123
20E_0168	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA25)	32	R/W	0000_0005h	37.4.86/ 2124
20E_016C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA26)	32	R/W	0000_0005h	37.4.87/ 2125
20E_0170	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA27)	32	R/W	0000_0005h	37.4.88/ 2126
20E_0174	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA28)	32	R/W	0000_0005h	37.4.89/ 2127
20E_0178	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA29)	32	R/W	0000_0005h	37.4.90/ 2128
20E_017C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA30)	32	R/W	0000_0005h	37.4.91/ 2129

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0180	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA31)	32	R/W	0000_0005h	37.4.92/ 2130
20E_0184	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD00)	32	R/W	0000_0000h	37.4.93/ 2131
20E_0188	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD01)	32	R/W	0000_0000h	37.4.94/ 2132
20E_018C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD10)	32	R/W	0000_0000h	37.4.95/ 2133
20E_0190	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD11)	32	R/W	0000_0000h	37.4.96/ 2134
20E_0194	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD12)	32	R/W	0000_0000h	37.4.97/ 2135
20E_0198	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD13)	32	R/W	0000_0000h	37.4.98/ 2136
20E_019C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD14)	32	R/W	0000_0000h	37.4.99/ 2137
20E_01A0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD15)	32	R/W	0000_0000h	37.4.100/ 2138
20E_01A4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD02)	32	R/W	0000_0000h	37.4.101/ 2139
20E_01A8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD03)	32	R/W	0000_0000h	37.4.102/ 2140
20E_01AC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD04)	32	R/W	0000_0000h	37.4.103/ 2141
20E_01B0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD05)	32	R/W	0000_0000h	37.4.104/ 2142
20E_01B4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD06)	32	R/W	0000_0000h	37.4.105/ 2143
20E_01B8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD07)	32	R/W	0000_0000h	37.4.106/ 2144
20E_01BC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD08)	32	R/W	0000_0000h	37.4.107/ 2145
20E_01C0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD09)	32	R/W	0000_0000h	37.4.108/ 2146
20E_01C4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_EB0_B)	32	R/W	0000_0000h	37.4.109/ 2147
20E_01C8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_EB1_B)	32	R/W	0000_0000h	37.4.110/ 2148
20E_01CC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_EB2_B)	32	R/W	0000_0005h	37.4.111/ 2149
20E_01D0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_EB3_B)	32	R/W	0000_0005h	37.4.112/ 2150
20E_01D4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_LBA_B)	32	R/W	0000_0000h	37.4.113/ 2151

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_01D8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_OE_B)	32	R/W	0000_0000h	37.4.114/ 2152
20E_01DC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_RW)	32	R/W	0000_0000h	37.4.115/ 2153
20E_01E0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_WAIT_B)	32	R/W	0000_0000h	37.4.116/ 2154
20E_01E4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_CRS_DV)	32	R/W	0000_0005h	37.4.117/ 2155
20E_01E8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_MDC)	32	R/W	0000_0005h	37.4.118/ 2156
20E_01EC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO)	32	R/W	0000_0005h	37.4.119/ 2157
20E_01F0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_REF_CLK)	32	R/W	0000_0005h	37.4.120/ 2158
20E_01F4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RX_ER)	32	R/W	0000_0005h	37.4.121/ 2159
20E_01F8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA0)	32	R/W	0000_0005h	37.4.122/ 2160
20E_01FC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA1)	32	R/W	0000_0005h	37.4.123/ 2161
20E_0200	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TX_EN)	32	R/W	0000_0005h	37.4.124/ 2162
20E_0204	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA0)	32	R/W	0000_0005h	37.4.125/ 2163
20E_0208	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA1)	32	R/W	0000_0005h	37.4.126/ 2164
20E_020C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO00)	32	R/W	0000_0005h	37.4.127/ 2165
20E_0210	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO01)	32	R/W	0000_0005h	37.4.128/ 2166
20E_0214	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO16)	32	R/W	0000_0005h	37.4.129/ 2167
20E_0218	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO17)	32	R/W	0000_0005h	37.4.130/ 2168
20E_021C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO18)	32	R/W	0000_0005h	37.4.131/ 2169
20E_0220	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO19)	32	R/W	0000_0005h	37.4.132/ 2170
20E_0224	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO02)	32	R/W	0000_0005h	37.4.133/ 2171
20E_0228	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO03)	32	R/W	0000_0005h	37.4.134/ 2172
20E_022C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO04)	32	R/W	0000_0005h	37.4.135/ 2173

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0230	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO05)	32	R/W	0000_0005h	37.4.136/ 2174
20E_0234	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO06)	32	R/W	0000_0005h	37.4.137/ 2175
20E_0238	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO07)	32	R/W	0000_0005h	37.4.138/ 2176
20E_023C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO08)	32	R/W	0000_0005h	37.4.139/ 2177
20E_0240	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO09)	32	R/W	0000_0005h	37.4.140/ 2178
20E_0244	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_COL0)	32	R/W	0000_0005h	37.4.141/ 2179
20E_0248	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_COL1)	32	R/W	0000_0005h	37.4.142/ 2180
20E_024C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_COL2)	32	R/W	0000_0005h	37.4.143/ 2181
20E_0250	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_COL3)	32	R/W	0000_0005h	37.4.144/ 2182
20E_0254	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_COL4)	32	R/W	0000_0005h	37.4.145/ 2183
20E_0258	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_ROW0)	32	R/W	0000_0005h	37.4.146/ 2184
20E_025C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_ROW1)	32	R/W	0000_0005h	37.4.147/ 2185
20E_0260	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_ROW2)	32	R/W	0000_0005h	37.4.148/ 2186
20E_0264	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_ROW3)	32	R/W	0000_0005h	37.4.149/ 2187
20E_0268	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_ROW4)	32	R/W	0000_0005h	37.4.150/ 2188
20E_026C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_ALE)	32	R/W	0000_0005h	37.4.151/ 2189
20E_0270	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CLE)	32	R/W	0000_0005h	37.4.152/ 2189
20E_0274	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CS0_B)	32	R/W	0000_0005h	37.4.153/ 2190
20E_0278	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CS1_B)	32	R/W	0000_0005h	37.4.154/ 2191
20E_027C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CS2_B)	32	R/W	0000_0005h	37.4.155/ 2191
20E_0280	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CS3_B)	32	R/W	0000_0005h	37.4.156/ 2192
20E_0284	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA00)	32	R/W	0000_0005h	37.4.157/ 2193

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0288	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA01)	32	R/W	0000_0005h	37.4.158/ 2194
20E_028C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA02)	32	R/W	0000_0005h	37.4.159/ 2195
20E_0290	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA03)	32	R/W	0000_0005h	37.4.160/ 2196
20E_0294	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA04)	32	R/W	0000_0005h	37.4.161/ 2197
20E_0298	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA05)	32	R/W	0000_0005h	37.4.162/ 2198
20E_029C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA06)	32	R/W	0000_0005h	37.4.163/ 2199
20E_02A0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA07)	32	R/W	0000_0005h	37.4.164/ 2200
20E_02A4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_READY_B)	32	R/W	0000_0005h	37.4.165/ 2201
20E_02A8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_WP_B)	32	R/W	0000_0005h	37.4.166/ 2201
20E_02AC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RD0)	32	R/W	0000_0005h	37.4.167/ 2202
20E_02B0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RD1)	32	R/W	0000_0005h	37.4.168/ 2203
20E_02B4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RD2)	32	R/W	0000_0005h	37.4.169/ 2204
20E_02B8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RD3)	32	R/W	0000_0005h	37.4.170/ 2205
20E_02BC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RX_CTL)	32	R/W	0000_0005h	37.4.171/ 2206
20E_02C0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RXC)	32	R/W	0000_0005h	37.4.172/ 2207
20E_02C4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TD0)	32	R/W	0000_0005h	37.4.173/ 2208
20E_02C8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TD1)	32	R/W	0000_0005h	37.4.174/ 2209
20E_02CC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TD2)	32	R/W	0000_0005h	37.4.175/ 2210
20E_02D0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TD3)	32	R/W	0000_0005h	37.4.176/ 2211
20E_02D4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TX_CTL)	32	R/W	0000_0005h	37.4.177/ 2212
20E_02D8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TXC)	32	R/W	0000_0005h	37.4.178/ 2213
20E_02DC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_CLK)	32	R/W	0000_0005h	37.4.179/ 2214

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_02E0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_CMD)	32	R/W	0000_0005h	37.4.180/ 2215
20E_02E4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0)	32	R/W	0000_0005h	37.4.181/ 2216
20E_02E8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1)	32	R/W	0000_0005h	37.4.182/ 2217
20E_02EC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2)	32	R/W	0000_0005h	37.4.183/ 2218
20E_02F0	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3)	32	R/W	0000_0005h	37.4.184/ 2219
20E_02F4	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CLK)	32	R/W	0000_0005h	37.4.185/ 2220
20E_02F8	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CMD)	32	R/W	0000_0005h	37.4.186/ 2221
20E_02FC	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0)	32	R/W	0000_0005h	37.4.187/ 2222
20E_0300	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1)	32	R/W	0000_0005h	37.4.188/ 2223
20E_0304	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2)	32	R/W	0000_0005h	37.4.189/ 2224
20E_0308	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3)	32	R/W	0000_0005h	37.4.190/ 2225
20E_030C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_CLK)	32	R/W	0000_0005h	37.4.191/ 2226
20E_0310	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_CMD)	32	R/W	0000_0005h	37.4.192/ 2227
20E_0314	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA0)	32	R/W	0000_0005h	37.4.193/ 2228
20E_0318	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA1)	32	R/W	0000_0005h	37.4.194/ 2229
20E_031C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA2)	32	R/W	0000_0005h	37.4.195/ 2230
20E_0320	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA3)	32	R/W	0000_0005h	37.4.196/ 2230
20E_0324	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA4)	32	R/W	0000_0005h	37.4.197/ 2231
20E_0328	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA5)	32	R/W	0000_0005h	37.4.198/ 2232
20E_032C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA6)	32	R/W	0000_0005h	37.4.199/ 2233
20E_0330	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA7)	32	R/W	0000_0005h	37.4.200/ 2234
20E_0334	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_RESET)	32	R/W	0000_0005h	37.4.201/ 2235

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0338	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_CLK)	32	R/W	0000_0005h	37.4.202/ 2236
20E_033C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_CMD)	32	R/W	0000_0005h	37.4.203/ 2237
20E_0340	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA0)	32	R/W	0000_0005h	37.4.204/ 2238
20E_0344	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA1)	32	R/W	0000_0005h	37.4.205/ 2239
20E_0348	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA2)	32	R/W	0000_0005h	37.4.206/ 2240
20E_034C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA3)	32	R/W	0000_0005h	37.4.207/ 2241
20E_0350	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA4)	32	R/W	0000_0005h	37.4.208/ 2241
20E_0354	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA5)	32	R/W	0000_0005h	37.4.209/ 2242
20E_0358	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA6)	32	R/W	0000_0005h	37.4.210/ 2243
20E_035C	Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA7)	32	R/W	0000_0005h	37.4.211/ 2244
20E_0360	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA10)	32	R/W	0001_B0B0h	37.4.212/ 2245
20E_0364	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA11)	32	R/W	0001_B0B0h	37.4.213/ 2246
20E_0368	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA12)	32	R/W	0001_B0B0h	37.4.214/ 2248
20E_036C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA13)	32	R/W	0001_B0B0h	37.4.215/ 2250
20E_0370	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA14)	32	R/W	0001_B0B0h	37.4.216/ 2252
20E_0374	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA15)	32	R/W	0001_B0B0h	37.4.217/ 2253
20E_0378	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA16)	32	R/W	0001_B0B0h	37.4.218/ 2255
20E_037C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA17)	32	R/W	0001_B0B0h	37.4.219/ 2257
20E_0380	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA18)	32	R/W	0001_B0B0h	37.4.220/ 2259
20E_0384	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA19)	32	R/W	0001_B0B0h	37.4.221/ 2260
20E_0388	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA04)	32	R/W	0001_B0B0h	37.4.222/ 2262
20E_038C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA05)	32	R/W	0001_B0B0h	37.4.223/ 2264

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0390	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA06)	32	R/W	0001_B0B0h	37.4.224/ 2266
20E_0394	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA07)	32	R/W	0001_B0B0h	37.4.225/ 2267
20E_0398	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA08)	32	R/W	0001_B0B0h	37.4.226/ 2269
20E_039C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA09)	32	R/W	0001_B0B0h	37.4.227/ 2271
20E_03A0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA_EN)	32	R/W	0001_B0B0h	37.4.228/ 2273
20E_03A4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_HSYNC)	32	R/W	0001_B0B0h	37.4.229/ 2274
20E_03A8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_PIXCLK)	32	R/W	0001_B0B0h	37.4.230/ 2276
20E_03AC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_VSYNC)	32	R/W	0001_B0B0h	37.4.231/ 2278
20E_03B0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DI0_DISP_CLK)	32	R/W	0001_B0B0h	37.4.232/ 2280
20E_03B4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN15)	32	R/W	0001_B0B0h	37.4.233/ 2281
20E_03B8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN02)	32	R/W	0001_B0B0h	37.4.234/ 2283
20E_03BC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN03)	32	R/W	0001_B0B0h	37.4.235/ 2285
20E_03C0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN04)	32	R/W	0001_B0B0h	37.4.236/ 2287
20E_03C4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA00)	32	R/W	0001_B0B0h	37.4.237/ 2288
20E_03C8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA01)	32	R/W	0001_B0B0h	37.4.238/ 2290
20E_03CC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA10)	32	R/W	0001_B0B0h	37.4.239/ 2292
20E_03D0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA11)	32	R/W	0001_B0B0h	37.4.240/ 2294
20E_03D4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA12)	32	R/W	0001_B0B0h	37.4.241/ 2295
20E_03D8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA13)	32	R/W	0001_B0B0h	37.4.242/ 2297
20E_03DC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA14)	32	R/W	0001_B0B0h	37.4.243/ 2299
20E_03E0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA15)	32	R/W	0001_B0B0h	37.4.244/ 2301
20E_03E4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA16)	32	R/W	0001_B0B0h	37.4.245/ 2302

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_03E8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA17)	32	R/W	0001_B0B0h	37.4.246/ 2304
20E_03EC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA18)	32	R/W	0001_B0B0h	37.4.247/ 2306
20E_03F0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA19)	32	R/W	0001_B0B0h	37.4.248/ 2308
20E_03F4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA02)	32	R/W	0001_B0B0h	37.4.249/ 2309
20E_03F8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA20)	32	R/W	0001_B0B0h	37.4.250/ 2311
20E_03FC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA21)	32	R/W	0001_B0B0h	37.4.251/ 2313
20E_0400	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA22)	32	R/W	0001_B0B0h	37.4.252/ 2315
20E_0404	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA23)	32	R/W	0001_B0B0h	37.4.253/ 2316
20E_0408	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA03)	32	R/W	0001_B0B0h	37.4.254/ 2318
20E_040C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA04)	32	R/W	0001_B0B0h	37.4.255/ 2320
20E_0410	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA05)	32	R/W	0001_B0B0h	37.4.256/ 2322
20E_0414	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA06)	32	R/W	0001_B0B0h	37.4.257/ 2323
20E_0418	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA07)	32	R/W	0001_B0B0h	37.4.258/ 2325
20E_041C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA08)	32	R/W	0001_B0B0h	37.4.259/ 2327
20E_0420	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA09)	32	R/W	0001_B0B0h	37.4.260/ 2329
20E_0424	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR00)	32	R/W	0000_8000h	37.4.261/ 2331
20E_0428	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR01)	32	R/W	0000_8000h	37.4.262/ 2333
20E_042C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR10)	32	R/W	0000_8000h	37.4.263/ 2335
20E_0430	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR11)	32	R/W	0000_8000h	37.4.264/ 2337
20E_0434	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR12)	32	R/W	0000_8000h	37.4.265/ 2339
20E_0438	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR13)	32	R/W	0000_8000h	37.4.266/ 2341
20E_043C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR14)	32	R/W	0000_8000h	37.4.267/ 2343

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0440	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR15)	32	R/W	0000_8000h	37.4.268/ 2345
20E_0444	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR02)	32	R/W	0000_8000h	37.4.269/ 2347
20E_0448	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR03)	32	R/W	0000_8000h	37.4.270/ 2349
20E_044C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR04)	32	R/W	0000_8000h	37.4.271/ 2351
20E_0450	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR05)	32	R/W	0000_8000h	37.4.272/ 2353
20E_0454	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR06)	32	R/W	0000_8000h	37.4.273/ 2355
20E_0458	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR07)	32	R/W	0000_8000h	37.4.274/ 2357
20E_045C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR08)	32	R/W	0000_8000h	37.4.275/ 2359
20E_0460	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR09)	32	R/W	0000_8000h	37.4.276/ 2361
20E_0464	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS_B)	32	R/W	0000_8030h	37.4.277/ 2363
20E_0468	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_CS0_B)	32	R/W	0000_8000h	37.4.278/ 2365
20E_046C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_CS1_B)	32	R/W	0000_8000h	37.4.279/ 2367
20E_0470	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0)	32	R/W	0000_8030h	37.4.280/ 2369
20E_0474	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1)	32	R/W	0000_8030h	37.4.281/ 2371
20E_0478	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2)	32	R/W	0000_8030h	37.4.282/ 2373
20E_047C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3)	32	R/W	0000_8030h	37.4.283/ 2375
20E_0480	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM4)	32	R/W	0000_8030h	37.4.284/ 2377
20E_0484	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM5)	32	R/W	0000_8030h	37.4.285/ 2379
20E_0488	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM6)	32	R/W	0000_8030h	37.4.286/ 2381
20E_048C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM7)	32	R/W	0000_8030h	37.4.287/ 2383
20E_0490	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS_B)	32	R/W	0000_8030h	37.4.288/ 2385
20E_0494	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET)	32	R/W	0008_3030h	37.4.289/ 2387

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0498	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA0)	32	R/W	0000_8000h	37.4.290/ 2389
20E_049C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA1)	32	R/W	0000_8000h	37.4.291/ 2391
20E_04A0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA2)	32	R/W	0000_B000h	37.4.292/ 2393
20E_04A4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE0)	32	R/W	0000_3000h	37.4.293/ 2395
20E_04A8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE1)	32	R/W	0000_3000h	37.4.294/ 2397
20E_04AC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK0_P)	32	R/W	0000_8030h	37.4.295/ 2399
20E_04B0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK1_P)	32	R/W	0000_8030h	37.4.296/ 2401
20E_04B4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ODT0)	32	R/W	0000_3030h	37.4.297/ 2403
20E_04B8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ODT1)	32	R/W	0000_3030h	37.4.298/ 2405
20E_04BC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0_P)	32	R/W	0000_2030h	37.4.299/ 2407
20E_04C0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS1_P)	32	R/W	0000_2030h	37.4.300/ 2409
20E_04C4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2_P)	32	R/W	0000_2030h	37.4.301/ 2411
20E_04C8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3_P)	32	R/W	0000_2030h	37.4.302/ 2413
20E_04CC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS4_P)	32	R/W	0000_2030h	37.4.303/ 2415
20E_04D0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS5_P)	32	R/W	0000_2030h	37.4.304/ 2417
20E_04D4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS6_P)	32	R/W	0000_2030h	37.4.305/ 2419
20E_04D8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS7_P)	32	R/W	0000_2030h	37.4.306/ 2421
20E_04DC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDWE_B)	32	R/W	0000_8000h	37.4.307/ 2423
20E_04E0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR16)	32	R/W	0000_B0B1h	37.4.308/ 2424
20E_04E4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR17)	32	R/W	0000_B0B1h	37.4.309/ 2426
20E_04E8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR18)	32	R/W	0000_B0B1h	37.4.310/ 2428
20E_04EC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR19)	32	R/W	0000_B0B1h	37.4.311/ 2429

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_04F0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR20)	32	R/W	0000_B0B1h	37.4.312/ 2431
20E_04F4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR21)	32	R/W	0000_B0B1h	37.4.313/ 2433
20E_04F8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR22)	32	R/W	0000_B0B1h	37.4.314/ 2435
20E_04FC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR23)	32	R/W	0000_B0B1h	37.4.315/ 2436
20E_0500	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR24)	32	R/W	0000_B0B1h	37.4.316/ 2438
20E_0504	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR25)	32	R/W	0000_B0B1h	37.4.317/ 2440
20E_0508	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_BCLK)	32	R/W	0000_B0B1h	37.4.318/ 2442
20E_050C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_CS0_B)	32	R/W	0000_B0B1h	37.4.319/ 2443
20E_0510	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_CS1_B)	32	R/W	0000_B0B1h	37.4.320/ 2445
20E_0514	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA16)	32	R/W	0001_B0B0h	37.4.321/ 2447
20E_0518	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA17)	32	R/W	0001_B0B0h	37.4.322/ 2449
20E_051C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA18)	32	R/W	0001_B0B0h	37.4.323/ 2450
20E_0520	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA19)	32	R/W	0001_B0B0h	37.4.324/ 2452
20E_0524	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA20)	32	R/W	0001_B0B0h	37.4.325/ 2454
20E_0528	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA21)	32	R/W	0001_B0B0h	37.4.326/ 2456
20E_052C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA22)	32	R/W	0001_30B0h	37.4.327/ 2457
20E_0530	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA23)	32	R/W	0001_B0B0h	37.4.328/ 2459
20E_0534	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA24)	32	R/W	0001_B0B0h	37.4.329/ 2461
20E_0538	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA25)	32	R/W	0001_B0B0h	37.4.330/ 2463
20E_053C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA26)	32	R/W	0001_B0B0h	37.4.331/ 2464
20E_0540	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA27)	32	R/W	0001_B0B0h	37.4.332/ 2466
20E_0544	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA28)	32	R/W	0001_B0B0h	37.4.333/ 2468

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0548	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA29)	32	R/W	0001_B0B0h	37.4.334/ 2470
20E_054C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA30)	32	R/W	0001_B0B0h	37.4.335/ 2471
20E_0550	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA31)	32	R/W	0001_30B0h	37.4.336/ 2473
20E_0554	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD00)	32	R/W	0000_B0B1h	37.4.337/ 2475
20E_0558	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD01)	32	R/W	0000_B0B1h	37.4.338/ 2477
20E_055C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD10)	32	R/W	0000_B0B1h	37.4.339/ 2478
20E_0560	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD11)	32	R/W	0000_B0B1h	37.4.340/ 2480
20E_0564	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD12)	32	R/W	0000_B0B1h	37.4.341/ 2482
20E_0568	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD13)	32	R/W	0000_B0B1h	37.4.342/ 2484
20E_056C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD14)	32	R/W	0000_B0B1h	37.4.343/ 2485
20E_0570	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD15)	32	R/W	0000_B0B1h	37.4.344/ 2487
20E_0574	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD02)	32	R/W	0000_B0B1h	37.4.345/ 2489
20E_0578	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD03)	32	R/W	0000_B0B1h	37.4.346/ 2491
20E_057C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD04)	32	R/W	0000_B0B1h	37.4.347/ 2492
20E_0580	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD05)	32	R/W	0000_B0B1h	37.4.348/ 2494
20E_0584	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD06)	32	R/W	0000_B0B1h	37.4.349/ 2496
20E_0588	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD07)	32	R/W	0000_B0B1h	37.4.350/ 2498
20E_058C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD08)	32	R/W	0000_B0B1h	37.4.351/ 2499
20E_0590	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD09)	32	R/W	0000_B0B1h	37.4.352/ 2501
20E_0594	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_EB0_B)	32	R/W	0000_B0B1h	37.4.353/ 2503
20E_0598	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_EB1_B)	32	R/W	0000_B0B1h	37.4.354/ 2505
20E_059C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_EB2_B)	32	R/W	0001_B0B0h	37.4.355/ 2506

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_05A0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_EB3_B)	32	R/W	0001_B0B0h	37.4.356/ 2508
20E_05A4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_LBA_B)	32	R/W	0000_B0B1h	37.4.357/ 2510
20E_05A8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_OE_B)	32	R/W	0000_B0B1h	37.4.358/ 2512
20E_05AC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_RW)	32	R/W	0000_B0B1h	37.4.359/ 2513
20E_05B0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_WAIT_B)	32	R/W	0000_B060h	37.4.360/ 2515
20E_05B4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_CRS_DV)	32	R/W	0001_B0B0h	37.4.361/ 2517
20E_05B8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_MDC)	32	R/W	0001_B0B0h	37.4.362/ 2519
20E_05BC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_MDIO)	32	R/W	0001_B0B0h	37.4.363/ 2520
20E_05C0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_REF_CLK)	32	R/W	0001_B0B0h	37.4.364/ 2522
20E_05C4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RX_ER)	32	R/W	0001_B0B0h	37.4.365/ 2524
20E_05C8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RX_DATA0)	32	R/W	0001_B0B0h	37.4.366/ 2526
20E_05CC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RX_DATA1)	32	R/W	0001_B0B0h	37.4.367/ 2527
20E_05D0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TX_EN)	32	R/W	0001_B0B0h	37.4.368/ 2529
20E_05D4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TX_DATA0)	32	R/W	0001_B0B0h	37.4.369/ 2531
20E_05D8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TX_DATA1)	32	R/W	0001_B0B0h	37.4.370/ 2533
20E_05DC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO00)	32	R/W	0001_30B0h	37.4.371/ 2534
20E_05E0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO01)	32	R/W	0001_B0B0h	37.4.372/ 2536
20E_05E4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO16)	32	R/W	0001_B0B0h	37.4.373/ 2538
20E_05E8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO17)	32	R/W	0001_B0B0h	37.4.374/ 2540
20E_05EC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO18)	32	R/W	0001_B0B0h	37.4.375/ 2541
20E_05F0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO19)	32	R/W	0001_B0B0h	37.4.376/ 2543
20E_05F4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO02)	32	R/W	0001_B0B0h	37.4.377/ 2545

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_05F8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO03)	32	R/W	0001_B0B0h	37.4.378/ 2547
20E_05FC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO04)	32	R/W	0001_B0B0h	37.4.379/ 2548
20E_0600	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO05)	32	R/W	0001_B0B0h	37.4.380/ 2550
20E_0604	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO06)	32	R/W	0001_B0B0h	37.4.381/ 2552
20E_0608	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO07)	32	R/W	0001_B0B0h	37.4.382/ 2554
20E_060C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO08)	32	R/W	0001_B0B0h	37.4.383/ 2555
20E_0610	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO09)	32	R/W	0001_B0B0h	37.4.384/ 2557
20E_0614	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD)	32	R/W	0000_B060h	37.4.385/ 2559
20E_0618	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK)	32	R/W	0000_7060h	37.4.386/ 2561
20E_061C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI)	32	R/W	0000_7060h	37.4.387/ 2562
20E_0620	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO)	32	R/W	0000_90B1h	37.4.388/ 2564
20E_0624	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS)	32	R/W	0000_7060h	37.4.389/ 2566
20E_0628	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TRSTB)	32	R/W	0000_7060h	37.4.390/ 2567
20E_062C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_COL0)	32	R/W	0001_B0B0h	37.4.391/ 2569
20E_0630	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_COL1)	32	R/W	0001_B0B0h	37.4.392/ 2571
20E_0634	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_COL2)	32	R/W	0001_B0B0h	37.4.393/ 2572
20E_0638	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_COL3)	32	R/W	0001_B0B0h	37.4.394/ 2574
20E_063C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_COL4)	32	R/W	0001_B0B0h	37.4.395/ 2576
20E_0640	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW0)	32	R/W	0001_B0B0h	37.4.396/ 2578
20E_0644	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW1)	32	R/W	0001_B0B0h	37.4.397/ 2579
20E_0648	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW2)	32	R/W	0001_B0B0h	37.4.398/ 2581
20E_064C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW3)	32	R/W	0001_B0B0h	37.4.399/ 2583

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0650	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW4)	32	R/W	0001_30B0h	37.4.400/ 2585
20E_0654	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_ALE)	32	R/W	0001_B0B0h	37.4.401/ 2586
20E_0658	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CLE)	32	R/W	0001_B0B0h	37.4.402/ 2588
20E_065C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CS0_B)	32	R/W	0001_B0B0h	37.4.403/ 2590
20E_0660	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CS1_B)	32	R/W	0001_B0B0h	37.4.404/ 2592
20E_0664	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CS2_B)	32	R/W	0001_B0B0h	37.4.405/ 2593
20E_0668	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CS3_B)	32	R/W	0001_B0B0h	37.4.406/ 2595
20E_066C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA00)	32	R/W	0001_B0B0h	37.4.407/ 2597
20E_0670	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA01)	32	R/W	0001_B0B0h	37.4.408/ 2599
20E_0674	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA02)	32	R/W	0001_B0B0h	37.4.409/ 2600
20E_0678	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA03)	32	R/W	0001_B0B0h	37.4.410/ 2602
20E_067C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA04)	32	R/W	0001_B0B0h	37.4.411/ 2604
20E_0680	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA05)	32	R/W	0001_B0B0h	37.4.412/ 2606
20E_0684	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA06)	32	R/W	0001_B0B0h	37.4.413/ 2607
20E_0688	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA07)	32	R/W	0001_B0B0h	37.4.414/ 2609
20E_068C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_READY_B)	32	R/W	0001_B0B0h	37.4.415/ 2611
20E_0690	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_WP_B)	32	R/W	0001_B0B0h	37.4.416/ 2613
20E_0694	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RD0)	32	R/W	0001_B030h	37.4.417/ 2615
20E_0698	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RD1)	32	R/W	0001_B030h	37.4.418/ 2617
20E_069C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RD2)	32	R/W	0001_B030h	37.4.419/ 2619
20E_06A0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RD3)	32	R/W	0001_B030h	37.4.420/ 2621
20E_06A4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RX_CTL)	32	R/W	0001_3030h	37.4.421/ 2623

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_06A8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RXC)	32	R/W	0001_3030h	37.4.422/ 2625
20E_06AC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TD0)	32	R/W	0001_B030h	37.4.423/ 2627
20E_06B0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TD1)	32	R/W	0001_B030h	37.4.424/ 2629
20E_06B4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TD2)	32	R/W	0001_B030h	37.4.425/ 2631
20E_06B8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TD3)	32	R/W	0001_B030h	37.4.426/ 2633
20E_06BC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TX_CTL)	32	R/W	0001_3030h	37.4.427/ 2635
20E_06C0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TXC)	32	R/W	0001_3030h	37.4.428/ 2637
20E_06C4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_CLK)	32	R/W	0001_B0B0h	37.4.429/ 2639
20E_06C8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_CMD)	32	R/W	0001_B0B0h	37.4.430/ 2641
20E_06CC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0)	32	R/W	0001_B0B0h	37.4.431/ 2642
20E_06D0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1)	32	R/W	0001_B0B0h	37.4.432/ 2644
20E_06D4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2)	32	R/W	0001_B0B0h	37.4.433/ 2646
20E_06D8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3)	32	R/W	0001_B0B0h	37.4.434/ 2648
20E_06DC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CLK)	32	R/W	0001_B0B0h	37.4.435/ 2649
20E_06E0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CMD)	32	R/W	0001_B0B0h	37.4.436/ 2651
20E_06E4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0)	32	R/W	0001_B0B0h	37.4.437/ 2653
20E_06E8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1)	32	R/W	0001_B0B0h	37.4.438/ 2655
20E_06EC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2)	32	R/W	0001_B0B0h	37.4.439/ 2656
20E_06F0	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3)	32	R/W	0001_B0B0h	37.4.440/ 2658
20E_06F4	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_CLK)	32	R/W	0001_B0B0h	37.4.441/ 2660
20E_06F8	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_CMD)	32	R/W	0001_B0B0h	37.4.442/ 2662
20E_06FC	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA0)	32	R/W	0001_B0B0h	37.4.443/ 2663

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0700	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA1)	32	R/W	0001_B0B0h	37.4.444/ 2665
20E_0704	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA2)	32	R/W	0001_B0B0h	37.4.445/ 2667
20E_0708	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA3)	32	R/W	0001_B0B0h	37.4.446/ 2669
20E_070C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA4)	32	R/W	0001_B0B0h	37.4.447/ 2670
20E_0710	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA5)	32	R/W	0001_B0B0h	37.4.448/ 2672
20E_0714	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA6)	32	R/W	0001_B0B0h	37.4.449/ 2674
20E_0718	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA7)	32	R/W	0001_B0B0h	37.4.450/ 2676
20E_071C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_RESET)	32	R/W	0001_B0B0h	37.4.451/ 2677
20E_0720	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_CLK)	32	R/W	0001_B0B0h	37.4.452/ 2679
20E_0724	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_CMD)	32	R/W	0001_B0B0h	37.4.453/ 2681
20E_0728	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA0)	32	R/W	0001_B0B0h	37.4.454/ 2683
20E_072C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA1)	32	R/W	0001_B0B0h	37.4.455/ 2684
20E_0730	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA2)	32	R/W	0001_B0B0h	37.4.456/ 2686
20E_0734	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA3)	32	R/W	0001_B0B0h	37.4.457/ 2688
20E_0738	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA4)	32	R/W	0001_B0B0h	37.4.458/ 2690
20E_073C	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA5)	32	R/W	0001_B0B0h	37.4.459/ 2691
20E_0740	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA6)	32	R/W	0001_B0B0h	37.4.460/ 2693
20E_0744	Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA7)	32	R/W	0001_B0B0h	37.4.461/ 2695
20E_0748	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B7DS)	32	R/W	0000_0030h	37.4.462/ 2697
20E_074C	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_ADDDS)	32	R/W	0000_0030h	37.4.463/ 2697
20E_0750	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL)	32	R/W	0000_0000h	37.4.464/ 2698
20E_0754	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDRPKE)	32	R/W	0000_1000h	37.4.465/ 2699

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0758	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDRPK)	32	R/W	0000_2000h	37.4.466/ 2700
20E_075C	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDRHYS)	32	R/W	0000_0000h	37.4.467/ 2701
20E_0760	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDRMODE)	32	R/W	0000_0000h	37.4.468/ 2702
20E_0764	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B0DS)	32	R/W	0000_0030h	37.4.469/ 2703
20E_0768	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII)	32	R/W	0008_0000h	37.4.470/ 2703
20E_076C	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_CTLDS)	32	R/W	0000_0030h	37.4.471/ 2704
20E_0770	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B1DS)	32	R/W	0000_0030h	37.4.472/ 2705
20E_0774	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE)	32	R/W	0008_0000h	37.4.473/ 2705
20E_0778	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B2DS)	32	R/W	0000_0030h	37.4.474/ 2706
20E_077C	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B3DS)	32	R/W	0000_0030h	37.4.475/ 2707
20E_0780	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B4DS)	32	R/W	0000_0030h	37.4.476/ 2708
20E_0784	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B5DS)	32	R/W	0000_0030h	37.4.477/ 2708
20E_0788	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_RGMII_TERM)	32	R/W	0000_0000h	37.4.478/ 2709
20E_078C	Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B6DS)	32	R/W	0000_0030h	37.4.479/ 2710
20E_0790	Select Input Register (IOMUXC_ANALOG_USB_OTG_ID_SELECT_INPUT)	32	R/W	0000_0000h	37.4.480/ 2711
20E_0794	Select Input Register (IOMUXC_ASRC_ASRCK_CLOCK_6_SELECT_INPUT)	32	R/W	0000_0000h	37.4.481/ 2711
20E_0798	Select Input Register (IOMUXC_AUD4_INPUT_DA_AMX_SELECT_INPUT)	32	R/W	0000_0000h	37.4.482/ 2712
20E_079C	Select Input Register (IOMUXC_AUD4_INPUT_DB_AMX_SELECT_INPUT)	32	R/W	0000_0000h	37.4.483/ 2713
20E_07A0	Select Input Register (IOMUXC_AUD4_INPUT_RXCLK_AMX_SELECT_INPUT)	32	R/W	0000_0000h	37.4.484/ 2714
20E_07A4	Select Input Register (IOMUXC_AUD4_INPUT_RXFS_AMX_SELECT_INPUT)	32	R/W	0000_0000h	37.4.485/ 2715
20E_07A8	Select Input Register (IOMUXC_AUD4_INPUT_TXCLK_AMX_SELECT_INPUT)	32	R/W	0000_0000h	37.4.486/ 2716
20E_07AC	Select Input Register (IOMUXC_AUD4_INPUT_TXFS_AMX_SELECT_INPUT)	32	R/W	0000_0000h	37.4.487/ 2717

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_07B0	Select Input Register (IOMUXC_AUD5_INPUT_DA_AMX_SELECT_INPUT)	32	R/W	0000_0000h	37.4.488/ 2718
20E_07B4	Select Input Register (IOMUXC_AUD5_INPUT_DB_AMX_SELECT_INPUT)	32	R/W	0000_0000h	37.4.489/ 2719
20E_07B8	Select Input Register (IOMUXC_AUD5_INPUT_RXCLK_AMX_SELECT_INPUT)	32	R/W	0000_0000h	37.4.490/ 2720
20E_07BC	Select Input Register (IOMUXC_AUD5_INPUT_RXFS_AMX_SELECT_INPUT)	32	R/W	0000_0000h	37.4.491/ 2721
20E_07C0	Select Input Register (IOMUXC_AUD5_INPUT_TXCLK_AMX_SELECT_INPUT)	32	R/W	0000_0000h	37.4.492/ 2722
20E_07C4	Select Input Register (IOMUXC_AUD5_INPUT_TXFS_AMX_SELECT_INPUT)	32	R/W	0000_0000h	37.4.493/ 2723
20E_07C8	Select Input Register (IOMUXC_FLEXCAN1_RX_SELECT_INPUT)	32	R/W	0000_0000h	37.4.494/ 2723
20E_07CC	Select Input Register (IOMUXC_FLEXCAN2_RX_SELECT_INPUT)	32	R/W	0000_0000h	37.4.495/ 2724
20E_07D4	Select Input Register (IOMUXC_CCM_PMIC_READY_SELECT_INPUT)	32	R/W	0000_0000h	37.4.496/ 2725
20E_07D8	Select Input Register (IOMUXC_ECSPI1_CSPI_CLK_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.497/ 2725
20E_07DC	Select Input Register (IOMUXC_ECSPI1_MISO_SELECT_INPUT)	32	R/W	0000_0000h	37.4.498/ 2726
20E_07E0	Select Input Register (IOMUXC_ECSPI1_MOSI_SELECT_INPUT)	32	R/W	0000_0000h	37.4.499/ 2727
20E_07E4	Select Input Register (IOMUXC_ECSPI1_SS0_SELECT_INPUT)	32	R/W	0000_0000h	37.4.500/ 2727
20E_07E8	Select Input Register (IOMUXC_ECSPI1_SS1_SELECT_INPUT)	32	R/W	0000_0000h	37.4.501/ 2728
20E_07EC	Select Input Register (IOMUXC_ECSPI1_SS2_SELECT_INPUT)	32	R/W	0000_0000h	37.4.502/ 2729
20E_07F0	Select Input Register (IOMUXC_ECSPI1_SS3_SELECT_INPUT)	32	R/W	0000_0000h	37.4.503/ 2730
20E_07F4	Select Input Register (IOMUXC_ECSPI2_CSPI_CLK_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.504/ 2730
20E_07F8	Select Input Register (IOMUXC_ECSPI2_MISO_SELECT_INPUT)	32	R/W	0000_0000h	37.4.505/ 2731
20E_07FC	Select Input Register (IOMUXC_ECSPI2_MOSI_SELECT_INPUT)	32	R/W	0000_0000h	37.4.506/ 2732
20E_0800	Select Input Register (IOMUXC_ECSPI2_SS0_SELECT_INPUT)	32	R/W	0000_0000h	37.4.507/ 2732
20E_0804	Select Input Register (IOMUXC_ECSPI2_SS1_SELECT_INPUT)	32	R/W	0000_0000h	37.4.508/ 2733
20E_0808	Select Input Register (IOMUXC_ECSPI4_SS0_SELECT_INPUT)	32	R/W	0000_0000h	37.4.509/ 2734

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_080C	Select Input Register (IOMUXC_ENET_REF_CLK_SELECT_INPUT)	32	R/W	0000_0000h	37.4.510/ 2735
20E_0810	Select Input Register (IOMUXC_ENET_MAC0_MDIO_SELECT_INPUT)	32	R/W	0000_0000h	37.4.511/ 2736
20E_0814	Select Input Register (IOMUXC_ENET_MAC0_RX_CLK_SELECT_INPUT)	32	R/W	0000_0000h	37.4.512/ 2737
20E_0818	Select Input Register (IOMUXC_ENET_MAC0_RX_DATA0_SELECT_INPUT)	32	R/W	0000_0000h	37.4.513/ 2738
20E_081C	Select Input Register (IOMUXC_ENET_MAC0_RX_DATA1_SELECT_INPUT)	32	R/W	0000_0000h	37.4.514/ 2739
20E_0820	Select Input Register (IOMUXC_ENET_MAC0_RX_DATA2_SELECT_INPUT)	32	R/W	0000_0000h	37.4.515/ 2740
20E_0824	Select Input Register (IOMUXC_ENET_MAC0_RX_DATA3_SELECT_INPUT)	32	R/W	0000_0000h	37.4.516/ 2741
20E_0828	Select Input Register (IOMUXC_ENET_MAC0_RX_EN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.517/ 2742
20E_082C	Select Input Register (IOMUXC_ESAI_RX_FS_SELECT_INPUT)	32	R/W	0000_0000h	37.4.518/ 2743
20E_0830	Select Input Register (IOMUXC_ESAI_TX_FS_SELECT_INPUT)	32	R/W	0000_0000h	37.4.519/ 2744
20E_0834	Select Input Register (IOMUXC_ESAI_RX_HF_CLK_SELECT_INPUT)	32	R/W	0000_0000h	37.4.520/ 2745
20E_0838	Select Input Register (IOMUXC_ESAI_TX_HF_CLK_SELECT_INPUT)	32	R/W	0000_0000h	37.4.521/ 2746
20E_083C	Select Input Register (IOMUXC_ESAI_RX_CLK_SELECT_INPUT)	32	R/W	0000_0000h	37.4.522/ 2747
20E_0840	Select Input Register (IOMUXC_ESAI_TX_CLK_SELECT_INPUT)	32	R/W	0000_0000h	37.4.523/ 2748
20E_0844	Select Input Register (IOMUXC_ESAI_SDO0_SELECT_INPUT)	32	R/W	0000_0000h	37.4.524/ 2749
20E_0848	Select Input Register (IOMUXC_ESAI_SDO1_SELECT_INPUT)	32	R/W	0000_0000h	37.4.525/ 2750
20E_084C	Select Input Register (IOMUXC_ESAI_SDO2_SDI3_SELECT_INPUT)	32	R/W	0000_0000h	37.4.526/ 2751
20E_0850	Select Input Register (IOMUXC_ESAI_SDO3_SDI2_SELECT_INPUT)	32	R/W	0000_0000h	37.4.527/ 2752
20E_0854	Select Input Register (IOMUXC_ESAI_SDO4_SDI1_SELECT_INPUT)	32	R/W	0000_0000h	37.4.528/ 2753
20E_0858	Select Input Register (IOMUXC_ESAI_SDO5_SDI0_SELECT_INPUT)	32	R/W	0000_0000h	37.4.529/ 2754
20E_085C	Select Input Register (IOMUXC_HDMI_ICECIN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.530/ 2755
20E_0860	Select Input Register (IOMUXC_HDMI_II2C_CLKIN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.531/ 2756

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0864	Select Input Register (IOMUXC_HDMI_II2C_DATAIN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.532/ 2757
20E_0868	Select Input Register (IOMUXC_I2C1_SCL_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.533/ 2758
20E_086C	Select Input Register (IOMUXC_I2C1_SDA_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.534/ 2759
20E_0870	Select Input Register (IOMUXC_I2C2_SCL_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.535/ 2760
20E_0874	Select Input Register (IOMUXC_I2C2_SDA_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.536/ 2761
20E_0878	Select Input Register (IOMUXC_I2C3_SCL_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.537/ 2761
20E_087C	Select Input Register (IOMUXC_I2C3_SDA_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.538/ 2762
20E_0880	Select Input Register (IOMUXC_I2C4_SCL_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.539/ 2763
20E_0884	Select Input Register (IOMUXC_I2C4_SDA_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.540/ 2763
20E_0888	Select Input Register (IOMUXC_IPU1_SENS1_DATA10_SELECT_INPUT)	32	R/W	0000_0000h	37.4.541/ 2764
20E_088C	Select Input Register (IOMUXC_IPU1_SENS1_DATA11_SELECT_INPUT)	32	R/W	0000_0000h	37.4.542/ 2765
20E_0890	Select Input Register (IOMUXC_IPU1_SENS1_DATA12_SELECT_INPUT)	32	R/W	0000_0000h	37.4.543/ 2766
20E_0894	Select Input Register (IOMUXC_IPU1_SENS1_DATA13_SELECT_INPUT)	32	R/W	0000_0000h	37.4.544/ 2767
20E_0898	Select Input Register (IOMUXC_IPU1_SENS1_DATA14_SELECT_INPUT)	32	R/W	0000_0000h	37.4.545/ 2768
20E_089C	Select Input Register (IOMUXC_IPU1_SENS1_DATA15_SELECT_INPUT)	32	R/W	0000_0000h	37.4.546/ 2769
20E_08A0	Select Input Register (IOMUXC_IPU1_SENS1_DATA16_SELECT_INPUT)	32	R/W	0000_0000h	37.4.547/ 2770
20E_08A4	Select Input Register (IOMUXC_IPU1_SENS1_DATA17_SELECT_INPUT)	32	R/W	0000_0000h	37.4.548/ 2771
20E_08A8	Select Input Register (IOMUXC_IPU1_SENS1_DATA18_SELECT_INPUT)	32	R/W	0000_0000h	37.4.549/ 2772
20E_08AC	Select Input Register (IOMUXC_IPU1_SENS1_DATA19_SELECT_INPUT)	32	R/W	0000_0000h	37.4.550/ 2773
20E_08B0	Select Input Register (IOMUXC_IPU1_SENS1_DATA_EN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.551/ 2774
20E_08B4	Select Input Register (IOMUXC_IPU1_SENS1_HSYNC_SELECT_INPUT)	32	R/W	0000_0000h	37.4.552/ 2775
20E_08B8	Select Input Register (IOMUXC_IPU1_SENS1_PIX_CLK_SELECT_INPUT)	32	R/W	0000_0000h	37.4.553/ 2776

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_08BC	Select Input Register (IOMUXC_IPU1_SENS1_VSYNC_SELECT_INPUT)	32	R/W	0000_0000h	37.4.554/ 2777
20E_08C0	Select Input Register (IOMUXC_KEY_COL5_SELECT_INPUT)	32	R/W	0000_0000h	37.4.555/ 2777
20E_08C4	Select Input Register (IOMUXC_KEY_COL6_SELECT_INPUT)	32	R/W	0000_0000h	37.4.556/ 2778
20E_08C8	Select Input Register (IOMUXC_KEY_COL7_SELECT_INPUT)	32	R/W	0000_0000h	37.4.557/ 2779
20E_08CC	Select Input Register (IOMUXC_KEY_ROW5_SELECT_INPUT)	32	R/W	0000_0000h	37.4.558/ 2779
20E_08D0	Select Input Register (IOMUXC_KEY_ROW6_SELECT_INPUT)	32	R/W	0000_0000h	37.4.559/ 2780
20E_08D4	Select Input Register (IOMUXC_KEY_ROW7_SELECT_INPUT)	32	R/W	0000_0000h	37.4.560/ 2781
20E_08DC	Select Input Register (IOMUXC_MLB_MLB_CLK_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.561/ 2781
20E_08E0	Select Input Register (IOMUXC_MLB_MLB_DATA_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.562/ 2782
20E_08E4	Select Input Register (IOMUXC_MLB_MLB_SIG_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.563/ 2783
20E_08E8	Select Input Register (IOMUXC_SDMA_EVENTS14_SELECT_INPUT)	32	R/W	0000_0000h	37.4.564/ 2784
20E_08EC	Select Input Register (IOMUXC_SDMA_EVENTS15_SELECT_INPUT)	32	R/W	0000_0000h	37.4.565/ 2785
20E_08F0	Select Input Register (IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT)	32	R/W	0000_0000h	37.4.566/ 2785
20E_08F4	Select Input Register (IOMUXC_SPDIF_TX_CLK2_SELECT_INPUT)	32	R/W	0000_0000h	37.4.567/ 2786
20E_08F8	Select Input Register (IOMUXC_UART1_UART_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	37.4.568/ 2787
20E_08FC	Select Input Register (IOMUXC_UART1_UART_RX_DATA_SELECT_INPUT)	32	R/W	0000_0000h	37.4.569/ 2787
20E_0900	Select Input Register (IOMUXC_UART2_UART_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	37.4.570/ 2788
20E_0904	Select Input Register (IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT)	32	R/W	0000_0000h	37.4.571/ 2789
20E_0908	Select Input Register (IOMUXC_UART3_UART_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	37.4.572/ 2789
20E_090C	Select Input Register (IOMUXC_UART3_UART_RX_DATA_SELECT_INPUT)	32	R/W	0000_0000h	37.4.573/ 2790
20E_0910	Select Input Register (IOMUXC_UART4_UART_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	37.4.574/ 2791
20E_0914	Select Input Register (IOMUXC_UART4_UART_RX_DATA_SELECT_INPUT)	32	R/W	0000_0000h	37.4.575/ 2791

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0918	Select Input Register (IOMUXC_UART5_UART_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	37.4.576/ 2792
20E_091C	Select Input Register (IOMUXC_UART5_UART_RX_DATA_SELECT_INPUT)	32	R/W	0000_0000h	37.4.577/ 2793
20E_0920	Select Input Register (IOMUXC_USB_OTG_OC_SELECT_INPUT)	32	R/W	0000_0000h	37.4.578/ 2794
20E_0924	Select Input Register (IOMUXC_USB_H1_OC_SELECT_INPUT)	32	R/W	0000_0000h	37.4.579/ 2795
20E_0928	Select Input Register (IOMUXC_USDHC1_CARD_CLK_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.580/ 2796
20E_092C	Select Input Register (IOMUXC_USDHC1_WP_ON_SELECT_INPUT)	32	R/W	0000_0000h	37.4.581/ 2797
20E_0930	Select Input Register (IOMUXC_USDHC2_CARD_CLK_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.582/ 2798
20E_0934	Select Input Register (IOMUXC_USDHC3_CARD_CLK_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.583/ 2799
20E_0938	Select Input Register (IOMUXC_USDHC4_CARD_CLK_IN_SELECT_INPUT)	32	R/W	0000_0000h	37.4.584/ 2800

37.4.1 GPR0 (IOMUXC_GPR0)

Address: 20E_0000h base + 0h offset = 20E_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CLOCK_8_MUX_SEL		CLOCK_0_MUX_SEL		CLOCK_B_MUX_SEL		CLOCK_3_MUX_SEL		CLOCK_A_MUX_SEL		CLOCK_2_MUX_SEL		CLOCK_9_MUX_SEL		CLOCK_1_MUX_SEL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TX_CLK2_MUX_SEL		-			PCIE_RX0_EQ			DMAREQ_MUX_SEL7	DMAREQ_MUX_SEL6	DMAREQ_MUX_SEL5	DMAREQ_MUX_SEL4	DMAREQ_MUX_SEL3	DMAREQ_MUX_SEL2	DMAREQ_MUX_SEL1	DMAREQ_MUX_SEL0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR0 field descriptions

Field	Description
31–30 CLOCK_8_MUX_SEL	Selects the source of asrck_clock_8 in ASRC according to clock muxing scheme 00 audmux.amx_output_rxclk_p7 muxed with ssi3.ssi_srck 01 audmux.amx_output_rxclk_p7 10 ssi3.ssi_srck 11 ssi3.rx_bit_clk
29–28 CLOCK_0_MUX_SEL	Selects the source of asrck_clock_0 in ASRC according to clock muxing scheme: 00 esai.ipp_ind_sckr muxed with esai.ipp_do_sckr 01 esai.ipp_ind_sckr 10 esai.ipp_do_sckr 11 Reserved
27–26 CLOCK_B_MUX_SEL	Selects the source of asrck_clock_b in ASRC according to clock muxing scheme: 00 audmux.amx_output_txclk_p7 muxed with ssi3.ssi_stck 01 audmux.amx_output_txclk_p7 10 ssi3.ssi_stck 11 ssi3.tx_bit_clk
25–24 CLOCK_3_MUX_SEL	Selects the source of asrck_clock_3 in ASRC according to clock muxing scheme: 00 audmux.amx_output_rxclk_p7 muxed with ssi3.ssi_srck 01 audmux.amx_output_rxclk_p7 10 ssi3.ssi_srck 11 ssi3.rx_bit_clk
23–22 CLOCK_A_MUX_SEL	Selects the source of asrck_clock_a in ASRC according to clock muxing scheme: 00 audmux.amx_output_txclk_p2 muxed with ssi2.ssi_stck 01 audmux.amx_output_txclk_p2 10 ssi2.ssi_stck 11 ssi2.tx_bit_clk
21–20 CLOCK_2_MUX_SEL	Selects the source of asrck_clock_2 in ASRC according to clock muxing scheme: 00 audmux.amx_output_rxclk_p2 muxed with ssi2.ssi_srck 01 audmux.amx_output_rxclk_p2 10 ssi2.ssi_srck 11 ssi2.rx_bit_clk
19–18 CLOCK_9_MUX_SEL	Selects the source of asrck_clock_9 in ASRC according to clock muxing scheme: 00 audmux.amx_output_txclk_p1 muxed with ssi1.ssi_stck 01 audmux.amx_output_txclk_p1 10 ssi1.ssi_stck 11 ssi1.tx_bit_clk
17–16 CLOCK_1_MUX_SEL	Selects the source of asrck_clock_1 in ASRC according to clock muxing scheme: 00 audmux.amx_output_rxclk_p1 muxed with ssi1.ssi_srck 01 audmux.amx_output_rxclk_p1 10 ssi1.ssi_srck 11 ssi1.rx_bit_clk

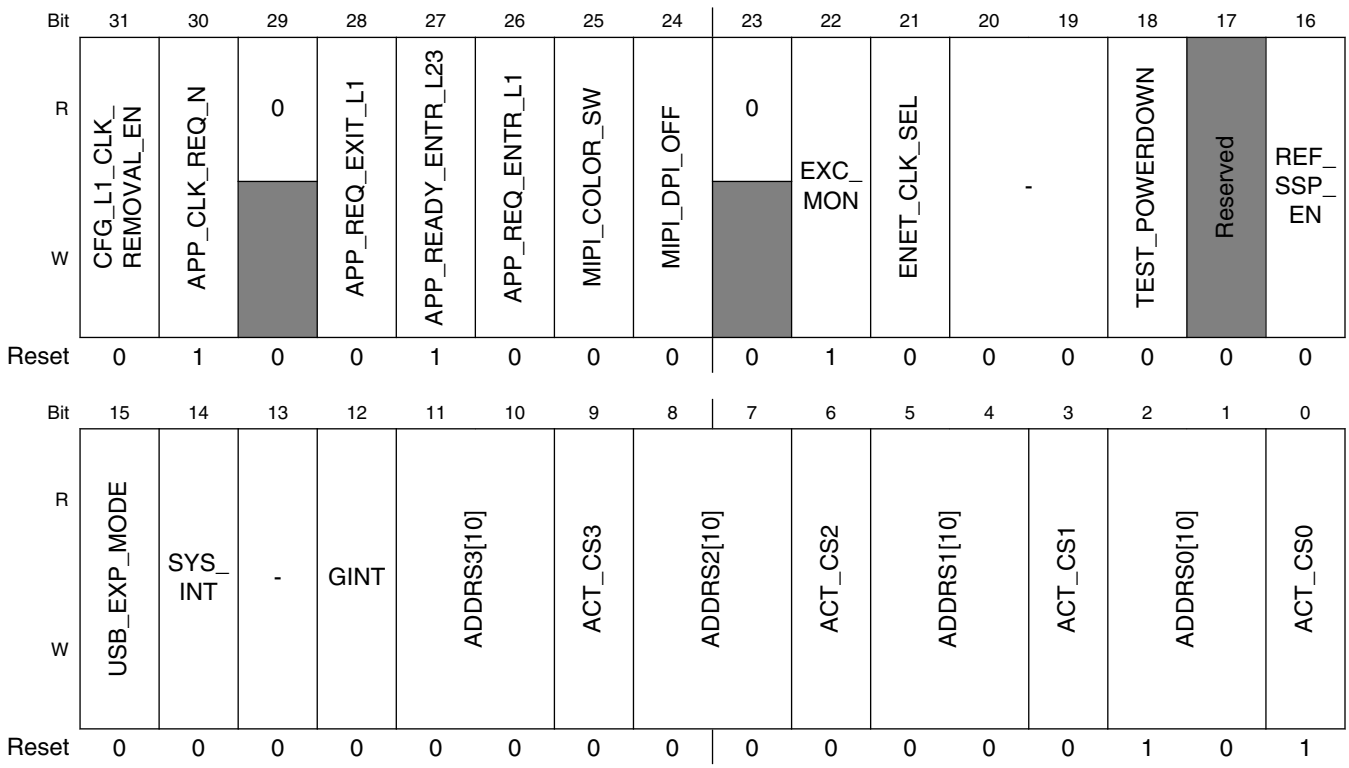
Table continues on the next page...

IOMUXC_GPR0 field descriptions (continued)

Field	Description
15–14 TX_CLK2_MUX_SEL	Selects the source of tx_clk2 in SPDIF according to ASRC clock muxing scheme: 00 same source as for asrc.asrck_clock_1 01 same source as for asrc.asrck_clock_2 10 same source as for asrc.asrck_clock_3 11 Reserved
13–11 -	
10–8 PCIE_RX0_EQ	control pcie phy's rx0_eq bits
7 DMAREQ_MUX_SEL7	Selects between two possible sources for SDMA_EVENT[14]: 0 spdif.drq0_spdif_b 1 iomux.sdma_ext_events[1] - External DMA Request via DISP0_DAT17 or GPIO_18
6 DMAREQ_MUX_SEL6	Selects between two possible sources for SDMA_EVENT[23]: 0 esai. 1 i2c3.ipi_int_b
5 DMAREQ_MUX_SEL5	Selects between two possible sources for SDMA_EVENT[9]: 0 ecspi4.ipd_req_cspi_rdma_b 1 epit2.ipi_int_epit_oc
4 DMAREQ_MUX_SEL4	Selects between two possible sources for SDMA_EVENT[10]: 0 ecspi4.ipd_req_cspi_tdma_b 1 i2c1.ipi_int_b
3 DMAREQ_MUX_SEL3	Selects between two possible sources for SDMA_EVENT[5]: 0 ecspi2.ipd_req_cspi_rdma_b 1 i2c1.ipi_int_b
2 DMAREQ_MUX_SEL2	Selects between two possible sources for SDMA_EVENT[4]: 0 ecspi1.ipd_req_cspi_tdma_b 1 i2c2.ipi_int_b
1 DMAREQ_MUX_SEL1	Selects between two possible sources for SDMA_EVENT[3]: 0 ecspi1.ipd_req_cspi_rdma_b 1 i2c3.ipi_int_b
0 DMAREQ_MUX_SEL0	Selects between two possible sources for SDMA_EVENT[2]: 0 ipu1. 1 iomux.sdma_events[0] - External DMA Request from pad DISP0_DAT16 or GPIO_17

37.4.2 GPR1 (IOMUXC_GPR1)

Address: 20E_0000h base + 4h offset = 20E_0004h



IOMUXC_GPR1 field descriptions

Field	Description
31 CFG_L1_CLK_REMOVAL_EN	PCIe_CTL (CLK LOGIC CONTROLLER GLUE) - Enable the reference clock removal in L1 state. This is a bit from application register.
30 APP_CLK_REQ_N	PCIe_CTL (CLK LOGIC CONTROLLER GLUE) - Indicates that application logic is ready to have reference clock removed.
29 Reserved	This read-only field is reserved and always has the value 0.
28 APP_REQ_EXIT_L1	PCIe_CTL - Application Request to Exit L1. Request from the application to exit ASPM state L1. 0 PCIe application request is not set 1 PCIe application request is set
27 APP_READY_ENTR_L23	PCIe_CTL - Application Ready to Enter L23. Indication from the application that it is ready to enter the L23 state. 0 PCIe application is not ready to enter L23 1 PCIe application is ready to enter L23

Table continues on the next page...

IOMUXC_GPR1 field descriptions (continued)

Field	Description
26 APP_REQ_ENTR_L1	PCIe_CTL - Application Request to Enter L1. Request from the application to enter ASPM state L1. 0 PCIe application request is not set 1 PCIe application request is set
25 MIPI_COLOR_SW	MIPI color switch control 0 MIPI color switch request is not set 1 MIPI color switch request is set
24 MIPI_DPI_OFF	MIPI DPI shutdown request 0 MIPI DPI shutdown request is not set 1 MIPI DPI shutdown request is set
23 Reserved	This read-only field is reserved and always has the value 0.
22 EXC_MON	Exclusive monitor response select of illegal command (of lal gaskets, except MMDC) 0 OKEY response 1 SLVError (default)
21 ENET_CLK_SEL	choose enet reference clk mode 0 get enet tx reference clk from pad (external OSC for both external PHY and Internal Controller) 1 get enet tx reference clk from internal clock from anatop (loopback through pad), this clock also sent out to external PHY
20–19 -	0 Gasket is selected 1 IOMUX is selected
18 TEST_POWERDOWN	PCIe_PHY - All Circuits Power-Down Control Function: Powers down all circuitry in the PHY for IDDQ testing. 0 Power down is not requested 1 Power down is requested
17 -	This field is reserved.
16 REF_SSP_EN	PCIe_PHY - Reference Clock Enable for SS function. Function: Enables the reference clock to the prescaler. The phy_ref_ssp_en signal must remain deasserted until the reference clock is running at the appropriate frequency, at which point phy_ref_ssp_en can be asserted. For lower power states, phy_ref_ssp_en can also be deasserted. 0 PCIe PHY reference clock is disabled 1 PCIe PHY reference clock is enabled
15 USB_EXP_MODE	USB Exposure mode. This is a factory test function. This field to remain disabled. 0 Exposure mode is disabled. 1 Exposure mode is enabled.
14 SYS_INT	PCIe_CTL - When SYS_INT goes from low to high, the core generates an Assert_INTx Message. When sys_int goes from high to low, the core generates a Deassert_INTx Message. 0 PCIe system interrupt request is not asserted 1 PCIe system interrupt request is asserted

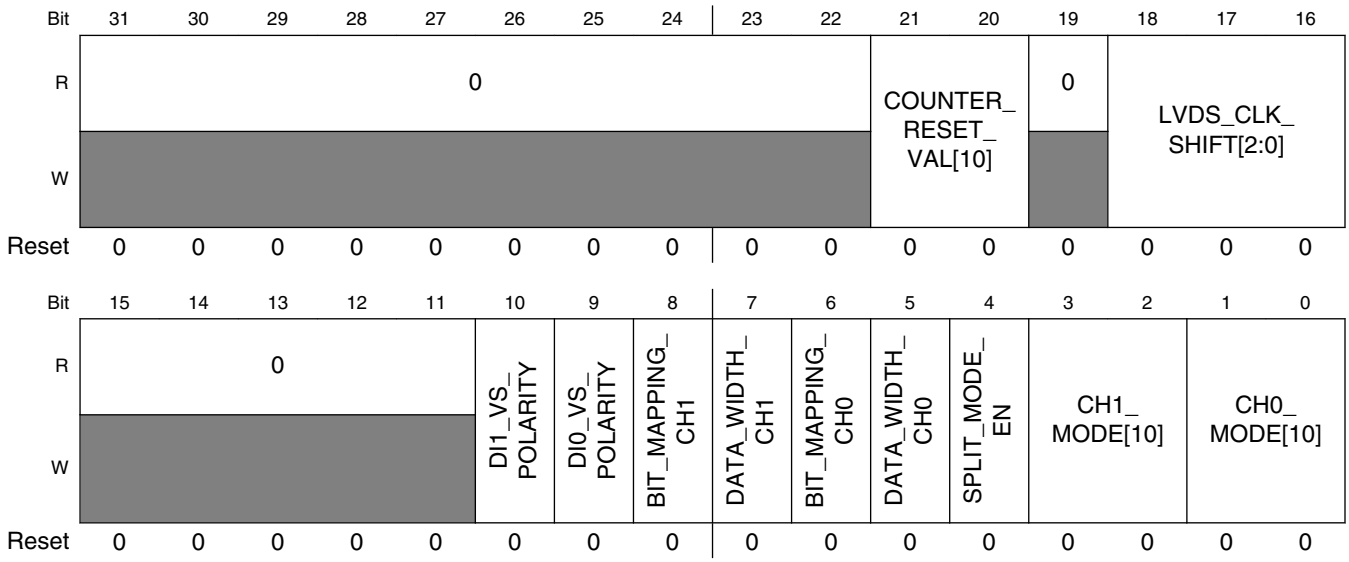
Table continues on the next page...

IOMUXC_GPR1 field descriptions (continued)

Field	Description
13 -	''usb_otg_id' pin iomux select control. (It functions as the 'daisy chain' mux control) 0 selects ENET_RX_ER 1 selects GPIO_1.
12 GINT	Global interrupt "0" bit (connected to Arm IRQ#0 and GPC) 0 Global interrupt request is not asserted 1 Global interrupt request is asserted
11–10 ADDRS3[10]	Active Chip Select and Address Space. Each of the ACT_CSx represents one of the four chip selects of the EIM. When ACT_CSx=1'b1, the corresponding chip select is active and has a valid address space according to its address space configuration determined by ADDRSx[10] bits ADDRSx[10] is setting the space for each chip select which is active. The address space of the first active chip select must be the biggest one, the following active chip select address spaces may be equal or lower. Total address space size is 128 MByte. The supported configurations are: CS0(128M), CS1 (0M), CS2 (0M), CS3(0M) [default configuration] CS0(64M), CS1(64M), CS2(0M), CS3(0M) CS0(64M), CS1(32M), CS2(32M), CS3(0M) CS0(32M), CS1(32M), CS2(32M), CS3(32M) Address Space Configuration options (ADDRSx[10]): 00 32 MByte 01 64 MByte 10 128 MByte 11 Reserved
9 ACT_CS3	See description for ADDRS3[10]
8–7 ADDRS2[10]	See description for ADDRS3[10]
6 ACT_CS2	See description for ADDRS3[10]
5–4 ADDRS1[10]	See description for ADDRS3[10]
3 ACT_CS1	See description for ADDRS3[10]
2–1 ADDRS0[10]	See description for ADDRS3[10]
0 ACT_CS0	See description for ADDRS3[10]

37.4.3 GPR2 (IOMUXC_GPR2)

Address: 20E_0000h base + 8h offset = 20E_0008h



IOMUXC_GPR2 field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value 0.
21–20 COUNTER_ RESET_VAL[10]	Reset value for the LDB counter which determines when the shift registers are loaded with data. **NOTE:** Used for debug purposes only. In normal functional operation must be '00' 00 Reset value is 5 01 Reset value is 3 10 Reset value is 4 11 Reset value is 6
19 Reserved	This read-only field is reserved and always has the value 0.
18–16 LVDS_CLK_ SHIFT[2:0]	Shifts the LVDS output clock in relation to the data. **NOTE:** Used for debug purposes only. In normal functional operation must be '000' 000 Output clock is '1100011' (normal operation) 001 Output clock is '1110001' 010 Output clock is '1111000' 011 Output clock is '1000111' 100 Output clock is '0001111' 101 Output clock is '0011111' 110 Output clock is '0111100' 111 Output clock is '1100011'
15–11 Reserved	This read-only field is reserved and always has the value 0.

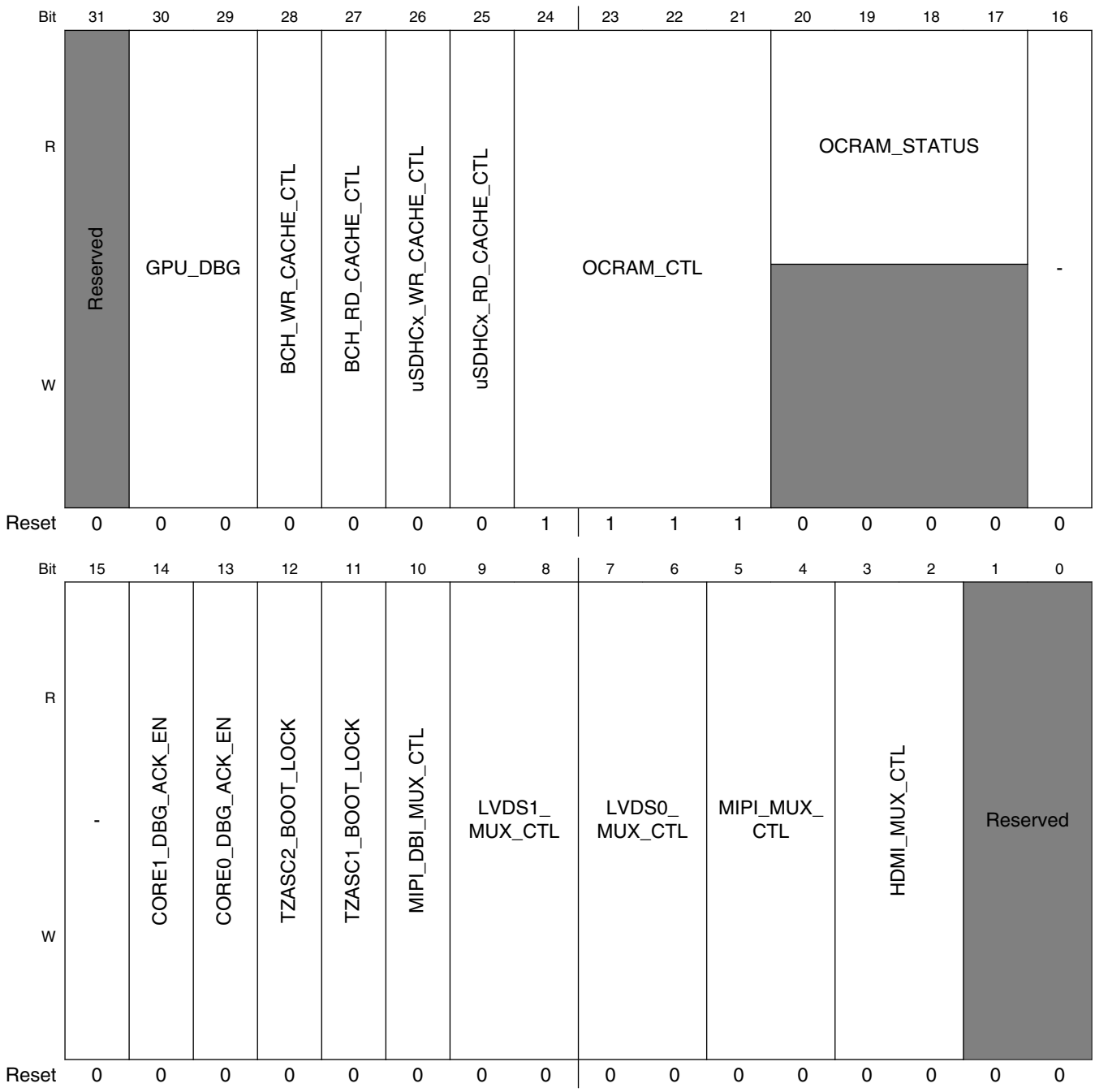
Table continues on the next page...

IOMUXC_GPR2 field descriptions (continued)

Field	Description
10 DI1_VS_ POLARITY	Vsync polarity for IPU's DI1 interface. 0 ipu_di1_vsync is active high. 1 ipu_di1_vsync is active low.
9 DI0_VS_ POLARITY	Vsync polarity for IPU's DI0 interface. 0 ipu_di0_vsync is active high. 1 ipu_di0_vsync is active low.
8 BIT_MAPPING_ CH1	Data mapping for LVDS channel 1. 0 Use SPWG standard. 1 Use JEIDA standard.
7 DATA_WIDTH_ CH1	Data width for LVDS channel 1. **NOTE:** This bit must be set when using JEIDA standard (bit_mapping_ch1 is set) 0 Data width is 18 bits wide (lvds1_tx3 is not used) 1 Data width is 24 bits wide.
6 BIT_MAPPING_ CH0	Data mapping for LVDS channel 0. 0 Use SPWG standard. 1 Use JEIDA standard.
5 DATA_WIDTH_ CH0	Data width for LVDS channel 0. **NOTE:** This bit must be set when using JEIDA standard (bit_mapping_ch0 is set) 0 Data width is 18 bits wide (lvds0_tx3 is not used) 1 Data width is 24 bits wide.
4 SPLIT_MODE_ EN	Enable split mode. 0 Split mode is disabled. 1 Split mode is enabled. In this mode both channels should be enabled and working with the same DI (ch0_mode and ch1_mode should both be either '01' or '11')
3–2 CH1_MODE[10]	LVDS channel 1 operation mode 00 Channel disabled. 01 Channel enabled, routed to DI0 10 Channel disabled. 11 Channel enabled, routed to DI1.
CH0_MODE[10]	LVDS channel 0 operation mode 00 Channel disabled. 01 Channel enabled, routed to DI0 10 Channel disabled. 11 Channel enabled, routed to DI1.

37.4.4 GPR3 (IOMUXC_GPR3)

Address: 20E_0000h base + Ch offset = 20E_000Ch



IOMUXC_GPR3 field descriptions

Field	Description
31 -	This field is reserved. Reserved

Table continues on the next page...

IOMUXC_GPR3 field descriptions (continued)

Field	Description
30–29 GPU_DBG	GPU debug busses to IOMUX 00 GPU3D 01 GPU2D 10 OpenVG 11 Reserved
28 BCH_WR_ CACHE_CTL	Control BCH block cacheable attribute of AXI write transactions [1] 0 Cacheable attribute is off for write transactions. 1 Cacheable attribute is on for write transactions.
27 BCH_RD_ CACHE_CTL	Control BCH block cacheable attribute of AXI read transactions [1] 0 Cacheable attribute is off for read transactions. 1 Cacheable attribute is on for read transactions.
26 uSDHCx_WR_ CACHE_CTL	Control uSDHCx [1-4] blocks cacheable attribute of AXI write transactions [1] 0 Cacheable attribute is off for write transactions. 1 Cacheable attribute is on for write transactions.
25 uSDHCx_RD_ CACHE_CTL	Control uSDHCx [1-4] blocks cacheable attribute of AXI read transactions [1] 0 Cacheable attribute is off for read transactions. 1 Cacheable attribute is on for read transactions.
24–21 OCRAM_CTL	OCRAM_CTL[24] write address pipeline control bit. When this feature is enabled, the write address from the AXI master would be delayed 1 cycle before it can be accepted by the on-chip RAM. This can avoid setup time issue for the write access on the memory cell at high frequency. Enable this feature would cost at most 1 more clock cycle for each AXI write transaction, i.e., at most 1 more clock cycle for each write burst with multiple beats of data. When this feature is disabled, the write address from the AXI master can be accepted by the on-chip RAM without delay, and data can be written to memory at this cycle (if no other access and write data is also ready at this cycle). 0 write address pipeline is disabled 1 write address pipeline is enabled OCRAM_CTL[23] - write data pipeline control bit When this feature is enabled, the write data from the AXI master would be delayed 1 cycle before it can be accepted by the on-chip RAM. This can avoid setup time issue for the write access on the memory cell at high frequency. Enable this feature would cost at most 1 more clock cycle for each AXI write transaction, i.e., at most 1 more clock cycle for each write burst with multiple beats of data. When this feature is disabled, the write data from the AXI master can be accepted by the on-chip RAM without delay, and data can be written to memory at this cycle (if no other access and write address is also ready at this cycle). 0 write data pipeline is disabled 1 write data pipeline is enabled OCRAM_CTL[22] read address pipeline control bit. When this feature is enabled, the read address from the AXI master would be delayed 1 cycle before it can be accepted by the on-chip RAM. This can avoid setup time issue for the read access on the memory cell at high frequency. Enable this feature would cost at most 1 more clock cycle for each AXI read transaction, i.e., at most 1 more clock cycle for each read burst with multiple beats of data. When this

Table continues on the next page...

IOMUXC_GPR3 field descriptions (continued)

Field	Description
	feature is disabled, the read address from the AXI master can be accepted by the on-chip RAM without delay, and data can become ready for master at next clock cycle (if no other access and no read data wait). 0 read address pipeline is disabled 1 read address pipeline is enabled OCRAM_CTL[21] - read data wait state control bit When theread data wait state is enabled, it will cost 2 cycles for each read access, (each beat of a read burst). This can avoid the potential timing problem caused by the relatively longer memory access time at higher frequency. When this feature is disabled, it only costs 1 clock cycle to finish a read transaction, i.e., get read data back in the next cycle of read request becomes valid on the bus. 0 read data pipeline is disabled 1 read data pipeline is enabled
20–17 OCRAM_ STATUS	This field shows the OCRAM pipeline settings status, controlled by OCRAM_CTL[24:21] bits respectively. When the control bit is changed, the corresponding status bit goes high and keeps high until this new configuration is applied the internal logic. This provides a way for software to detect that the configuration has become valid. The suggested flow for changing the configuration in software is: • set/clear the control bit • poll the status bit until it goes to 0 OCRAM_STATUS[17] shows the write address pipeline status. This bit value reflects the propagation of the respective control bit to OCRAM memory. OCRAM_STATUS[18] shows the write data pipeline status. This bit value reflects the propagation of the respective control bit to OCRAM memory. OCRAM_STATUS[19] shows the read address pipeline status. This bit value reflects the propagation of the respective control bit to OCRAM memory. OCRAM_STATUS[20] shows the read data pipeline status. This bit value reflects the propagation of the respective control bit to OCRAM memory. 0 read data pipeline configuration valid 1 read data pipeline control bit changed
16–15 -	Mask control of Core 2 debug acknowledge to global debug acknowledge 0 Core 2 debug acknowledge is part of global acknowledge. 1 Core 2 debug acknowledge is masked by this bit, and it is not part of global acknowledge.
14 CORE1_DBG_ ACK_EN	Mask control of Core 1 debug acknowledge to global debug acknowledge. 0 Core 1 debug acknowledge is part of global acknowledge. 1 Core 1 debug acknowledge is masked by this bit, and it is not part of global acknowledge.
13 CORE0_DBG_ ACK_EN	Mask control of Core 1 debug acknowledge to global debug acknowledge 0 Core 0 debug acknowledge is part of global acknowledge. 1 Core 0 debug acknowledge is masked by this bit, and it is not part of global acknowledge.
12 TZASC2_BOOT_ LOCK	TZASC-2 secure boot lock 0 secure boot lock is disabled. 1 secure boot lock is enabled

Table continues on the next page...

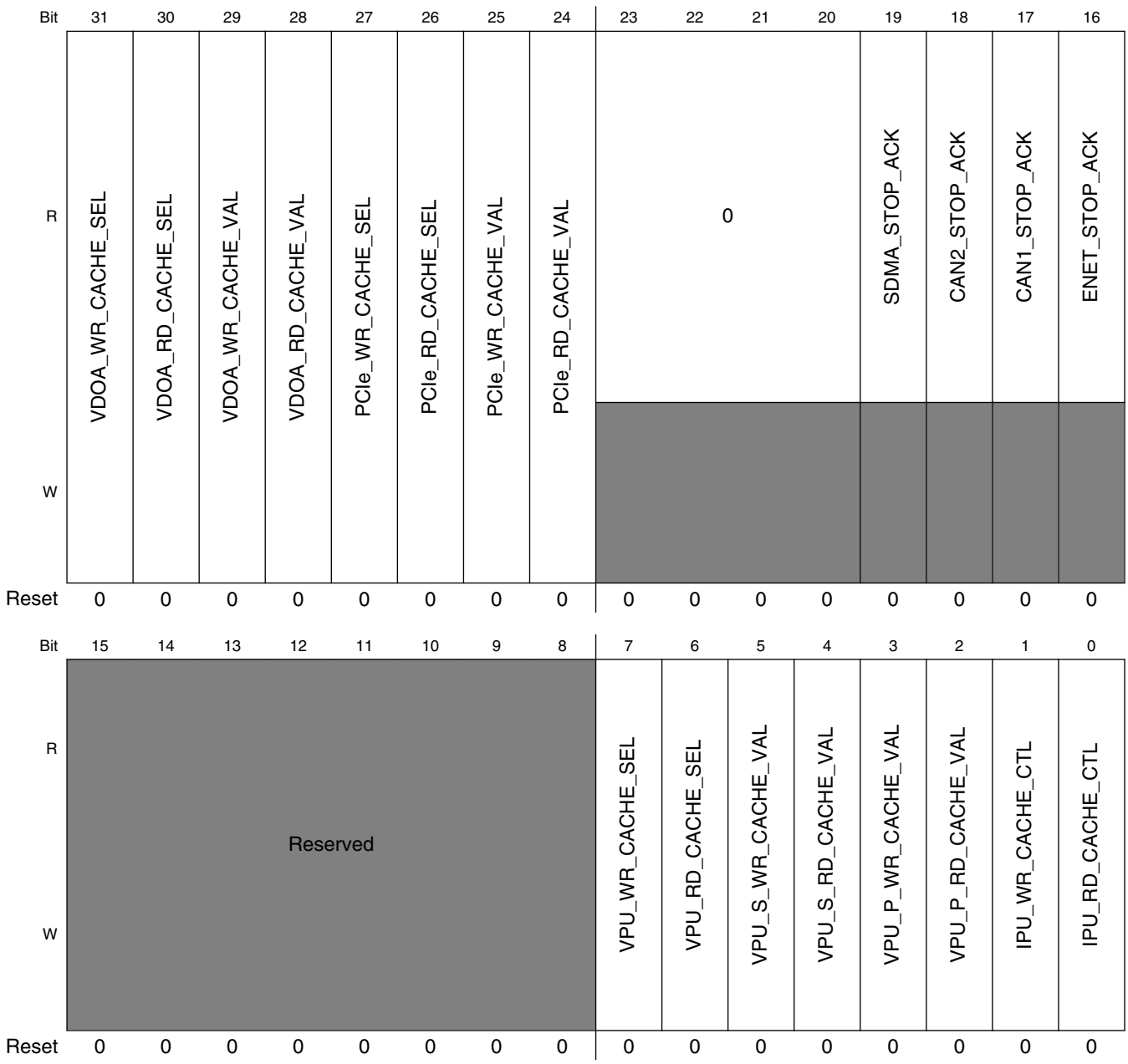
IOMUXC_GPR3 field descriptions (continued)

Field	Description
11 TZASC1_BOOT_LOCK	TZASC-1 secure boot lock 0 secure boot lock is disabled. 1 secure boot lock is enabled
10 MIPI_DBI_MUX_CTL	MIPI DBI mux control 0 IPU1 DI_1 input source is MIPI DBI 1 IPU1 DI_1 input source is itself (from IPU1 DI_1 pad/mux)
9–8 LVDS1_MUX_CTL	LVDS1 MUX control 00 LVDS1 source is IPU1 DI0 port 01 LVDS1 source is IPU1 DI1 port 10 Reserved
7–6 LVDS0_MUX_CTL	LVDS0 MUX control 00 LVDS0 source is IPU1 DI0 port 01 LVDS0 source is IPU1 DI1 port 10 Reserved
5–4 MIPI_MUX_CTL	MIPI MUX control 00 MIPI source is IPU1 DI0 port 01 MIPI source is IPU1 DI1 port 10 Reserved
3–2 HDMI_MUX_CTL	HDMI MUX control 00 HDMI source is IPU1 DI0 port 01 HDMI source is IPU1 DI1 port 10 Reserved
-	This field is reserved. Reserved

1. Set of the cache bits, enable packet optimization through the bus system to DDR controller. The only side affect - is that bus may change the nature of the accesses, which may lead to problems when accessing FIFO type address. In most typical cases - these bits should be set. For the GPU3D, GPU2D and OpenVG, such settings is possible through the IP programming model. For few peripherals, for these bits to take effect, it is required to also select set '1' to 'cache-mux' control bit.

37.4.5 GPR4 (IOMUXC_GPR4)

Address: 20E_0000h base + 10h offset = 20E_0010h



IOMUXC_GPR4 field descriptions

Field	Description
31 VDOA_WR_CACHE_SEL	This bit selects the cacheable attribute of VDOA AXI write transcations [1] 0 The write transactions cacheable attribute is driven by the VDOA core 1 The write transactions cacheable attribute is driven by VDOA_WR_CACHE_VAL.

Table continues on the next page...

IOMUXC_GPR4 field descriptions (continued)

Field	Description
30 VDOA_RD_ CACHE_SEL	This bit selects the cacheable attribute of VDOA AXI read transcations) [1] 0 The read transaction cacheable attribute is driven by the VDOA core 1 The read transaction cacheable attribute is driven by VDOA_RD_CACHE_VAL.
29 VDOA_WR_ CACHE_VAL	VDOA block cacheable attribute value of AXI write transactions The value of VDOA_WR_CACHE_VAL is affecting the transactions only if VDOA_WR_CACHE_SEL is set. [1] 0 Cacheable attribute is off for write transactions. 1 Cacheable attribute is on for write transactions.
28 VDOA_RD_ CACHE_VAL	VDOA block cacheable attribute value of AXI read transactions The value of VDOA_RD_CACHE_VAL is affecting the transactions only if VDOA_RD_CACHE_SEL is set. [1] 0 Cacheable attribute is off for read transactions. 1 Cacheable attribute is on for read transactions.
27 PCIe_WR_ CACHE_SEL	This bit selects the cacheable attribute of PCIe AXI write transcations [1] 0 The write transactions cacheable attribute is driven by the PCIe core 1 The write transactions cacheable attribute is driven by PCIe_WR_CACHE_VAL.
26 PCIe_RD_ CACHE_SEL	This bit selects the cacheable attribute of PCIe AXI read transcations) [1] 0 The read transaction cacheable attribute is driven by the PCIe core 1 The read transaction cacheable attribute is driven by PCIe_RD_CACHE_VAL.
25 PCIe_WR_ CACHE_VAL	PCIe block cacheable attribute value of AXI write transactions The value of PCIe_WR_CACHE_VAL is affecting the transactions only if PCIe_WR_CACHE_SEL is set. [1] 0 Cacheable attribute is off for write transactions. 1 Cacheable attribute is on for write transactions.
24 PCIe_RD_ CACHE_VAL	PCIe block cacheable attribute value of AXI read transactions The value of PCIe_RD_CACHE_VAL is affecting the transactions only if PCIe_RD_CACHE_SEL is set. [1] 0 Cacheable attribute is off for read transactions. 1 Cacheable attribute is on for read transactions.
23–20 Reserved	This read-only field is reserved and always has the value 0.
19 SDMA_STOP_ ACK	SDMA stop acknowledge. This is status (read only) bit. 0 SDMA stop acknowledge is not asserted. 1 SDMA stop acknowledge is asserted, SDMA is in STOP mode.
18 CAN2_STOP_ ACK	CAN-2 stop acknowledge. This is status (read only) bit. 0 CAN-2 stop acknowledge is not asserted. 1 CAN-2 stop acknowledge is asserted, CAN-2 is in STOP mode.
17 CAN1_STOP_ ACK	CAN-1 stop acknowledge. This is status (read only) bit. 0 CAN-1 stop acknowledge is not asserted. 1 CAN-1 stop acknowledge is asserted, CAN-1 is in STOP mode.

Table continues on the next page...

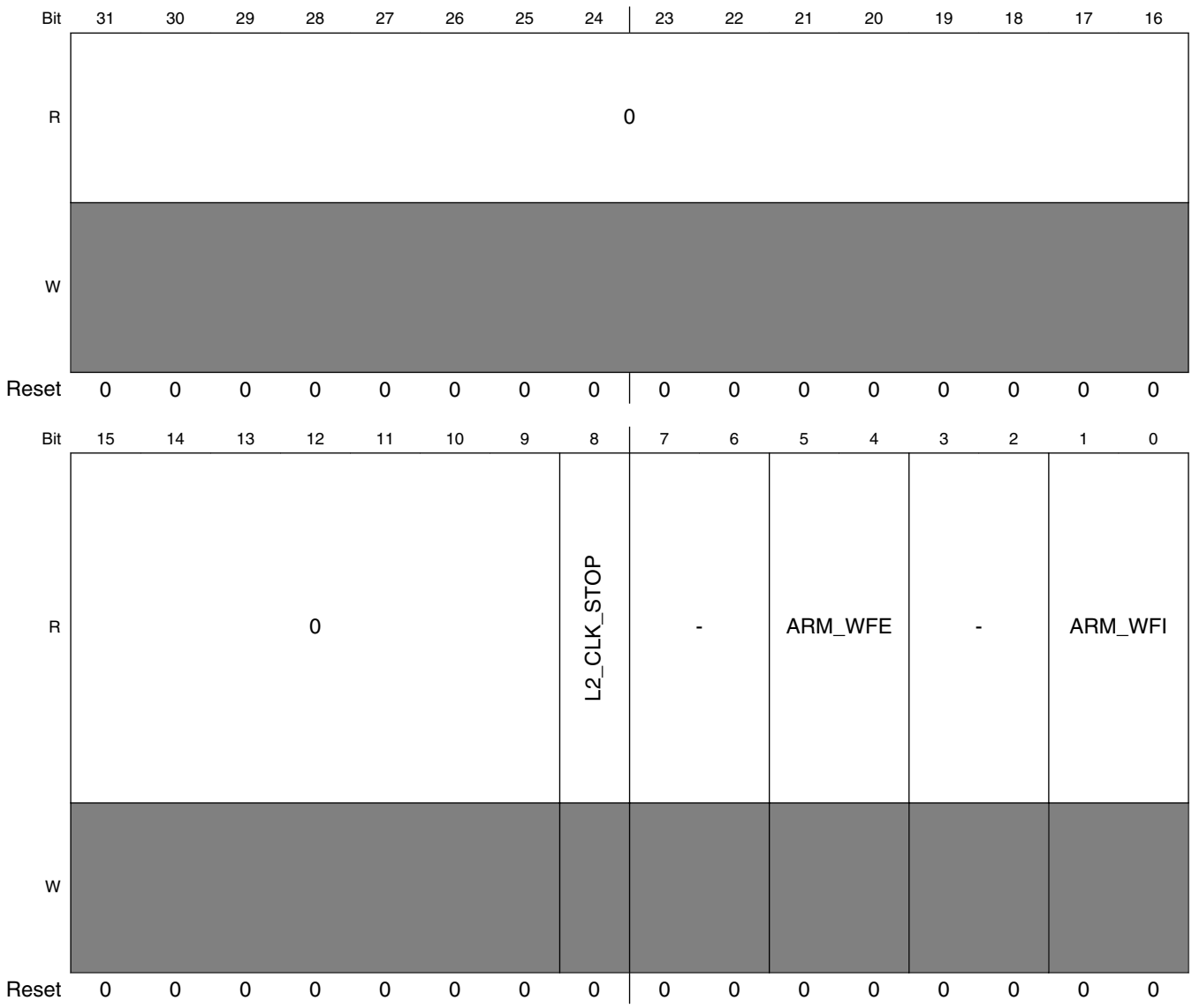
IOMUXC_GPR4 field descriptions (continued)

Field	Description
16 ENET_STOP_ACK	ENET stop acknowledge. This is status (read only) bit. 0 ENET stop acknowledge is not asserted. 1 ENET stop acknowledge is asserted, ENET is in STOP mode.
15–8 -	This field is reserved. Reserved
7 VPU_WR_CACHE_SEL	This bit selects the cacheable attribute of VPU AXI write transcations (both primary and secondary AXI buses) [1] 0 The write transactions cacheable attribute is driven by the VPU core 1 The write transactions cacheable attribute is driven by VPU_SEC_WR_CACHE_VAL for secondary bus and VPU_P_WR_CACHE_VAL for primary bus.
6 VPU_RD_CACHE_SEL	This bit selects the cacheable attribute of VPU AXI read transcations (both primary and secondary AXI buses) [1] 0 The read transaction cacheable attribute is driven by the VPU core 1 The read transaction cacheable attribute is driven by VPU_SEC_RD_CACHE_VAL for secondary bus and VPU_P_RD_CACHE_VAL for primary bus.
5 VPU_S_WR_CACHE_VAL	VPU (secondary bus) block cacheable attribute value of AXI write transactions The value of VPU_S_WR_CACHE_VAL is affecting the transactions only if VPU_WR_CACHE_SEL is set. [1]
4 VPU_S_RD_CACHE_VAL	VPU (secondary bus) block cacheable attribute value of AXI read transactions The value of VPU_S_RD_CACHE_VAL is affecting the transactions only if VPU_RD_CACHE_SEL is set. [1]
3 VPU_P_WR_CACHE_VAL	VPU (primary bus) block cacheable attribute value of AXI write transactions The value of VPU_P_WR_CACHE_VAL is affecting the transactions only if VPU_WR_CACHE_SEL is set. [1] 0 Cacheable attribute is off for write transactions. 1 Cacheable attribute is on for write transactions.
2 VPU_P_RD_CACHE_VAL	VPU (primary bus) block cacheable attribute value of AXI read transactions The value of VPU_P_RD_CACHE_VAL is affecting the transactions only if VPU_RD_CACHE_SEL is set. [1] 0 Cacheable attribute is off for read transactions. 1 Cacheable attribute is on for read transactions.
1 IPU_WR_CACHE_CTL	Control IPU-1 block cacheable attribute of AXI write transactions [1] 0 Cacheable attribute is off for write transactions. 1 Cacheable attribute is on for write transactions.
0 IPU_RD_CACHE_CTL	Control IPU-1 block cacheable attribute of AXI read transactions [1] 0 Cacheable attribute is off for read transactions. 1 Cacheable attribute is on for read transactions.

1. Set of the cache bits, enable packet optimization through the bus system to DDR controller. The only side affect - is that bus may change the nature of the accesses, which may lead to problems when accessing FIFO type address. In most typical cases - these bits should be set. For the GPU3D, GPU2D and OpenVG, such settings is possible through the IP programming model. For few peripherals, for these bits to take effect, it is required to also select set '1' to 'cache-mux' control bit.

37.4.6 GPR5 (IOMUXC_GPR5)

Address: 20E_0000h base + 14h offset = 20E_0014h



IOMUXC_GPR5 field descriptions

Field	Description
31–9 Reserved	This read-only field is reserved and always has the value 0.
8 L2_CLK_STOP	L2 cache clock stop indication (this is a status, read only bit] 0 L2 cache clock is running 1 L2 cache clock stopped
7–6 -	

Table continues on the next page...

IOMUXC_GPR5 field descriptions (continued)

Field	Description
5–4 ARM_WFE	Arm WFE event out indication on WFE state of the cores (these are status, read only bits) 0 Arm Core[GPR5-index - 4] is not in â€œWait for Eventâ€ mode 1 Arm Core[GPR5-index - 4] is in â€œWait for Eventâ€ mode
3–2 -	
ARM_WFI	Arm WFI event out indicating on WFI state of the cores (these are status, read only bits) 0 Arm Core[GPR5-index] is not in â€œWait for Interruptâ€ mode 1 Arm Core[GPR5-index] is in â€œWait for Interruptâ€ mode

37.4.7 GPR6 (IOMUXC_GPR6)

Address: 20E_0000h base + 18h offset = 20E_0018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IPU1_ID11_RD_QoS				IPU1_ID10_RD_QoS				IPU1_ID01_RD_QoS				IPU1_ID00_RD_QoS				IPU1_ID11_WR_QoS				IPU1_ID10_WR_QoS				IPU1_ID01_WR_QoS				IPU1_ID00_WR_QoS			
Reset	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0

IOMUXC_GPR6 field descriptions

Field	Description
31–28 IPU1_ID11_RD_QoS	IPU1 Read AXI ID=11 Quality of Service (QoS) priority 0000 0xxx - 3 lsbâ€™s will be passed as configured 1000 1xxx - 1111
27–24 IPU1_ID10_RD_QoS	IPU1 Read AXI ID=10 Quality of Service (QoS) priority 0000 0xxx - 3 lsbâ€™s will be passed as configured 1000 1xxx - 1111
23–20 IPU1_ID01_RD_QoS	IPU1 Read AXI ID=01 Quality of Service (QoS) priority 0000 0xxx - 3 lsbâ€™s will be passed as configured 1000 1xxx - 1111
19–16 IPU1_ID00_RD_QoS	IPU1 Read AXI ID=00 Quality of Service (QoS) priority 0000 0xxx - 3 lsbâ€™s will be passed as configured 1000 1xxx - 1111
15–12 IPU1_ID11_WR_QoS	IPU1 Write AXI ID=11 Quality of Service (QoS) priority 0000 0xxx - 3 lsbâ€™s will be passed as configured 1000 1xxx - 1111
11–8 IPU1_ID10_WR_QoS	IPU1 Write AXI ID=10 Quality of Service (QoS) priority 0000 0xxx - 3 lsbâ€™s will be passed as configured 1000 1xxx - 1111

Table continues on the next page...

IOMUXC_GPR6 field descriptions (continued)

Field	Description
7–4 IPU1_ID01_WR_QoS	IPU1 Write AXI ID=01 Quality of Service (QoS) priority 0000 0xxx - 3 lsbâ€™s will be passed as configured 1000 1xxx - 1111
IPU1_ID00_WR_QoS	IPU1 Write AXI ID=00 Quality of Service (QoS) priority 0000 0xxx -3 lsbâ€™s will be passed as configured 1000 1xxx - 1111

37.4.8 GPR7 (IOMUXC_GPR7)

Address: 20E_0000h base + 1Ch offset = 20E_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-																															
Reset	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0

IOMUXC_GPR7 field descriptions

Field	Description
-	0 TBD 1 TBD

37.4.9 GPR8 (IOMUXC_GPR8)

Address: 20E_0000h base + 20h offset = 20E_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PCS_TX_SWING_LOW							PCS_TX_SWING_FULL							PCS_TX_DEEMPH_GEN2_6DB						PCS_TX_DEEMPH_GEN2_3P5DB						PCS_TX_DEEMPH_GEN1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR8 field descriptions

Field	Description
31–25 PCS_TX_SWING_LOW	PCIe_PHY - This static value sets the launch amplitude of the transmitter when pipe0_tx_swing is set to 1'b0 (default state). 0 TBD 1 TBD
24–18 PCS_TX_SWING_FULL	PCIe_PHY - This static value sets the Tx driver de-emphasis value in the case where pipe0_tx_deemph is set to 1'b0 and the PHY is running at the Gen2 rate.

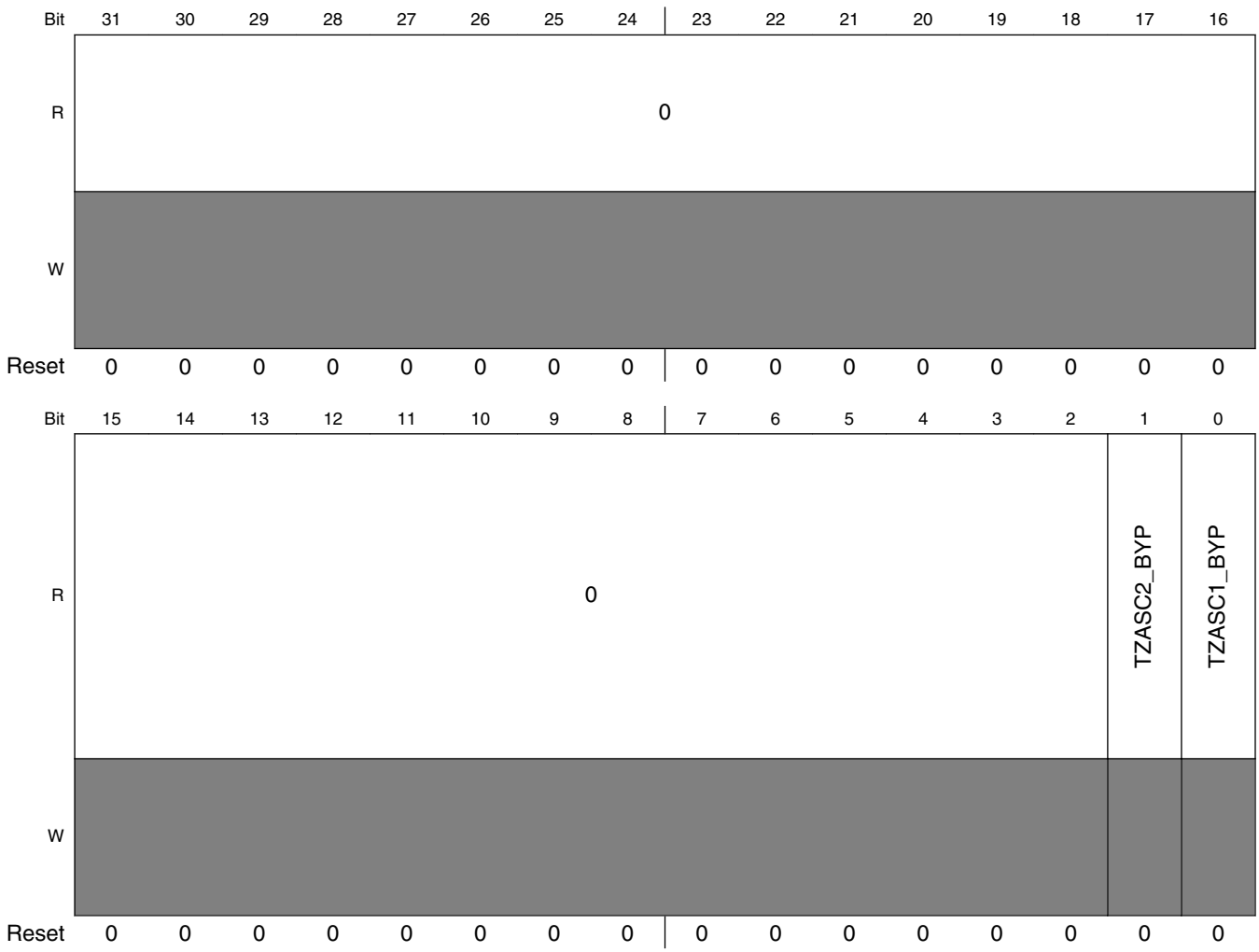
Table continues on the next page...

IOMUXC_GPR8 field descriptions (continued)

Field	Description
	0 TBD 1 TBD
17–12 PCS_TX_ DEEMPH_ GEN2_6DB	PCIe_PHY - This static value sets the Tx driver de-emphasis value in the case where pipe0_tx_deemph is set to 1'b0 and the PHY is running at the Gen2 rate. 0 TBD 1 TBD
11–6 PCS_TX_ DEEMPH_ GEN2_3P5DB	PCIe_PHY - This static value sets the Tx driver de-emphasis value in the case where pipe0_tx_deemph is set to 1'b1 (the default setting) and the PHY is running at the Gen2 rate. 0 TBD 1 TBD
PCS_TX_ DEEMPH_GEN1	PCIe_PHY - This static value sets the Tx driver de-emphasis value in the case where pipe0_tx_deemph is set to 1'b1 (the default setting) and the PHY is running at the Gen1 rate. 0 TBD 1 TBD

37.4.10 GPR9 (IOMUXC_GPR9)

Address: 20E_0000h base + 24h offset = 20E_0024h

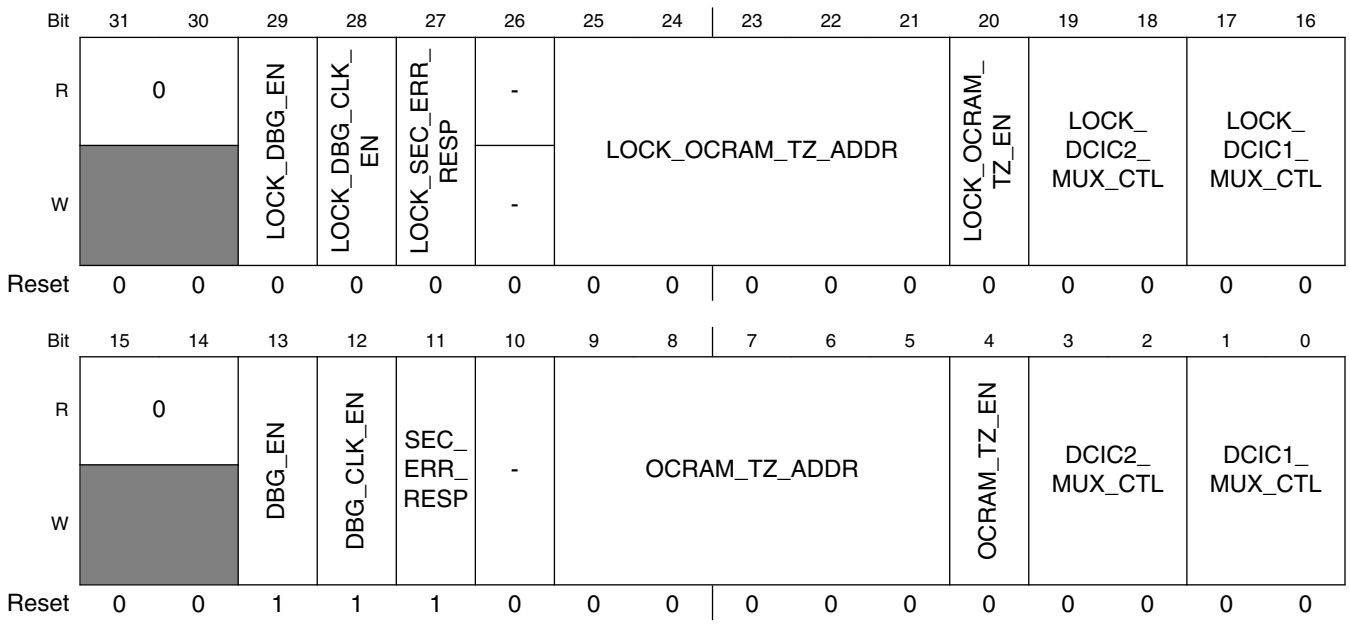


IOMUXC_GPR9 field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
1 TZASC2_BYP	TZASC-2 BYPASS MUX control 0 The TZASC-2 is bypassed and the transactions to DDR are not being checked. 1 The TZASC-2 is not bypassed and the transactions to DDR are being monitored / checked.
0 TZASC1_BYP	TZASC-1 BYPASS MUX control 0 The TZASC-1 is bypassed and the transactions to DDR are not being checked. 1 The TZASC-1 is not bypassed and the transactions to DDR are being monitored / checked.

37.4.11 GPR10 (IOMUXC_GPR10)

Address: 20E_0000h base + 28h offset = 20E_0028h



IOMUXC_GPR10 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value 0.
29 LOCK_DBG_EN	Lock DBG_EN field for changes. This is a sticky field, once set it can't be cleared (only by reset). 0 Field is not locked 1 Field is locked (read access only)
28 LOCK_DBG_CLK_EN	Lock DBG_CLK_EN field for changes. This is a sticky field, once set it can't be cleared (only by reset). 0 Field is not locked 1 Field is locked (read access only)
27 LOCK_SEC_ERR_RESP	Lock SEC_ERR_RESP field for changes. This is a sticky field, once set it can't be cleared (only by reset). 0 Field is not locked 1 Field is locked (read access only)
26 -	
25–21 LOCK_OCRAM_TZ_ADDR	Lock OCRAM_TZ_ADDR field for changes. This is a sticky field, once set it can't be cleared (only by reset). 0 Field is not locked 1 Field is locked (read access only)
20 LOCK_OCRAM_TZ_EN	Lock OCRAM_TZ_EN field for changes. This is a sticky field, once set it can't be cleared (only by reset).

Table continues on the next page...

IOMUXC_GPR10 field descriptions (continued)

Field	Description
	0 Field is not locked 1 Field is locked (read access only)
19–18 LOCK_DCIC2_ MUX_CTL	Lock DCIC2_MUX_CTL field for changes. This is a sticky field, once set it can't be cleared (only by reset). 0 Field is not locked 1 Field is locked (read access only)
17–16 LOCK_DCIC1_ MUX_CTL	Lock DCIC1_MUX_CTL field for changes. This is a sticky field, once set it can't be cleared (only by reset). 0 Field is not locked 1 Field is locked (read access only)
15–14 Reserved	This read-only field is reserved and always has the value 0.
13 DBG_EN	Arm non secure (non-invasive) debug enable 0 Debug turned off. 1 Debug enabled (default).
12 DBG_CLK_EN	Arm Debug clock enable 0 Debug turned off. 1 Debug enabled (default).
11 SEC_ERR_ RESP	Security error response enable for all security gaskets (on both AHB and AXI busses) 0 OKEY response 1 SLVError (default)
10 -	
9–5 OCRAM_TZ_ ADDR	OCRAM TrustZone (TZ) start address. This is the start address of the secure memory region within the OCRAM memory space is 4KB granularity. The start address affects the OCRAM transactions only if OCRAM_TZ_EN bit is set. The OCRAM TZ ENDADDR is not configurable and is set to the end of OCRAM memory space.
4 OCRAM_TZ_EN	OCRAM TrustZone (TZ) enable. 0 The TrustZone feature is disabled. Entire OCRAM space is available for all access types (secure/non-secure/user/supervisor). 1 The TrustZone feature is enabled. Access to address in the range specified by [ENDADDR:STARTADDR] follows the execution mode access policy described in CSU chapter.
3–2 DCIC2_MUX_ CTL	DCIC-2 MUX control 00 DCIC-2 source is IPU1 DI1 port 01 DCIC-2 source is LVDS0 port 10 DCIC-2 source is LVDS1 port 11 DCIC-2 source is MIPI DPI port
DCIC1_MUX_ CTL	DCIC-1 MUX control 00 DCIC-1 source is IPU1 DI0 port 01 DCIC-1 source is LVDS0 10 DCIC-1 source is LVDS1 11 DCIC-1 source is HDMI

37.4.12 GPR11 (IOMUXC_GPR11)

Address: 20E_0000h base + 2Ch offset = 20E_002Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR11 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 Reserved	This read-only field is reserved and always has the value 0.
15–1 Reserved	This read-only field is reserved and always has the value 0.
0 Reserved	This read-only field is reserved and always has the value 0.

37.4.13 GPR12 (IOMUXC_GPR12)

Address: 20E_0000h base + 30h offset = 20E_0030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				ARMP_IPG_CLK_EN	ARMP_AHB_CLK_EN	ARMP_ATB_CLK_EN	ARMP_APB_CLK_EN	PCIe_CTL_7			DIAG_STATUS_BUS_SELECT				APPS_PM_XMT_TURNOFF
W																
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEVICE_TYPE				APP_INIT_RST	APP_LTSSM_ENABLE	APPS_PM_XMT_PME	LOS_LEVEL					uSDHC_DBG_MUX		0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR12 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27 ARMP_IPG_ CLK_EN	Arm platform IPG clock enable 0 IPG clock is not running (gated). 1 IPG clock is running (enabled).
26 ARMP_AHB_ CLK_EN	Arm platform AHB clock enable 0 IPG clock is not running (gated). 1 IPG clock is running (enabled).
25 ARMP_ATB_ CLK_EN	Arm platform ATB clock enable 0 IPG clock is not running (gated). 1 IPG clock is running (enabled).
24 ARMP_APB_ CLK_EN	Arm platform APB clock enable 0 IPG clock is not running (gated). 1 IPG clock is running (enabled).
23–21 PCIe_CTL_7	PCIe control of diagnostic bus select TBD
20–17 DIAG_STATUS_ BUS_SELECT	PCIe_CTL - used for debug to select what part of diag_status_bus will be reflected on the 32 bits of the iomux
16 APPS_PM_XMT_ TURNOFF	PCIe_CTL - Request from the application to generate a PM_Turn_Off Message.
15–12 DEVICE_TYPE	PCIe_CTL - Device/Port Type. Indicates the specific type of this PCI Express Function (EP or RC) DEVICE_TYPE field values 0011-1111 are reserved. 0000 **PCIE_EP** — EP Mode 0010 **PCIE_RC** — RC Mode
11 APP_INIT_RST	PCIe_CTL - Request from the application to send a Hot Reset to the downstream device.
10 APP_LTSSM_ ENABLE	PCIe_CTL - Driven low by the application after reset to hold the LTSSM in the Detect state until the application is ready. When the application has finished initializing the core configuration registers, it asserts app_ltssm_enable to allow the LTSSM to continue Link establishment. 0 Application is not ready. 1 Application is not ready.
9 APPS_PM_XMT_ PME	PCIe_CTL - Wake Up. Used by application logic to wake up the PMC state machine from a D1, D2 or D3 power state. Upon wake-up, the core sends a PM_PME Message
8–4 LOS_LEVEL	PCIe_PHY - Loss-of-Signal Detector Sensitivity Level Control Function: Sets the sensitivity level for the Loss-of-Signal detector. This signal must be set to 0x9
3–2 uSDHC_DBG_ MUX	uSDHC debug mux control TBD
Reserved	This read-only field is reserved and always has the value 0.

37.4.14 GPR13 (IOMUXC_GPR13)

Address: 20E_0000h base + 34h offset = 20E_0034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	SDMA_STOP_REQ	CAN2_STOP_REQ	CAN1_STOP_REQ	ENET_STOP_REQ	-										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EPDC_WR_CACHE_SEL	EPDC_RD_CACHE_SEL	EPDC_WR_CACHE_VAL	EPDC_RD_CACHE_VAL	PXP_WR_CACHE_SEL	PXP_RD_CACHE_SEL	PXP_WR_CACHE_VAL	PXP_RD_CACHE_VAL	-		IPU_CSI1_MUX			IPU_CSI0_MUX		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

IOMUXC_GPR13 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30 SDMA_STOP_REQ	SDMA stop request 0 Do not request the SDMA module to enter STOP mode. 1 Request the SDMA module to enter STOP mode. The SDMA will acknowledge this request in IOMUXC_GPR4[SDMA_STOP_ACK].
29 CAN2_STOP_REQ	CAN2 stop request 0 Do not request the CAN2 module to enter STOP mode. 1 Request the CAN2 module to enter STOP mode. The CAN2 will acknowledge this request in IOMUXC_GPR4[CAN2_STOP_ACK].
28 CAN1_STOP_REQ	CAN1 stop request 0 Do not request the CAN1 module to enter STOP mode. 1 Request the CAN1 module to enter STOP mode. The CAN1 will acknowledge this request in IOMUXC_GPR4[CAN1_STOP_ACK].
27 ENET_STOP_REQ	ENET stop request 0 Do not request the ENET module to enter STOP mode. 1 Request the ENET module to enter STOP mode. The ENET will acknowledge this request in IOMUXC_GPR4[ENET_STOP_ACK].
26–16 -	

Table continues on the next page...

IOMUXC_GPR13 field descriptions (continued)

Field	Description
15 EPDC_WR_ CACHE_SEL	This bit selects the cacheable attribute of EPDC AXI write transcations 0 The write transactions cacheable attribute is driven by the EPDC core 1 The write transactions cacheable attribute is driven by EPDC_WR_CACHE_VAL.
14 EPDC_RD_ CACHE_SEL	This bit selects the cacheable attribute of EPDC AXI read transcations) 0 The read transaction cacheable attribute is driven by the EPDC core 1 The read transaction cacheable attribute is driven by EPDC_RD_CACHE_VAL.
13 EPDC_WR_ CACHE_VAL	EPDC block cacheable attribute value of AXI write transactions The value of EPDC_WR_CACHE_VAL is affecting the transactions only if EPDC_WR_CACHE_SEL is set. 0 Cacheable attribute is off for write transactions. 1 Cacheable attribute is on for write transactions.
12 EPDC_RD_ CACHE_VAL	EPDC block cacheable attribute value of AXI read transactions The value of EPDC_RD_CACHE_VAL is affecting the transactions only if EPDC_RD_CACHE_SEL is set. 0 Cacheable attribute is off for read transactions. 1 Cacheable attribute is on for read transactions.
11 PXP_WR_ CACHE_SEL	This bit selects the cacheable attribute of PXP AXI write transcations 0 The write transactions cacheable attribute is driven by the PXP core 1 The write transactions cacheable attribute is driven by PXP_WR_CACHE_VAL.
10 PXP_RD_ CACHE_SEL	This bit selects the cacheable attribute of PXP AXI read transcations) 0 The read transaction cacheable attribute is driven by the PXP core 1 The read transaction cacheable attribute is driven by PXP_RD_CACHE_VAL.
9 PXP_WR_ CACHE_VAL	PXP block cacheable attribute value of AXI write transactions The value of PXP_WR_CACHE_VAL is affecting the transactions only if PXP_WR_CACHE_SEL is set. 0 Cacheable attribute is off for write transactions. 1 Cacheable attribute is on for write transactions.
8 PXP_RD_ CACHE_VAL	PXP block cacheable attribute value of AXI read transactions The value of PXP_RD_CACHE_VAL is affecting the transactions only if PXP_RD_CACHE_SEL is set. 0 Cacheable attribute is off for read transactions. 1 Cacheable attribute is on for read transactions.
7–6 -	
5–3 IPU_CSI1_MUX	000 MIPI CSI0 001 MIPI CSI1 010 MIPI CSI2 011 MIPI CSI3 100 IPU CSI1
IPU_CSI0_MUX	000 MIPI_CSI0 001 MIPI_CSI1 010 MIPI_CSI2

Table continues on the next page...

IOMUXC_GPR13 field descriptions (continued)

Field	Description
	011 MIPI_CSI3 100 IPU CSI0

37.4.15 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA10)

Address: 20E_0000h base + 4Ch offset = 20E_004Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

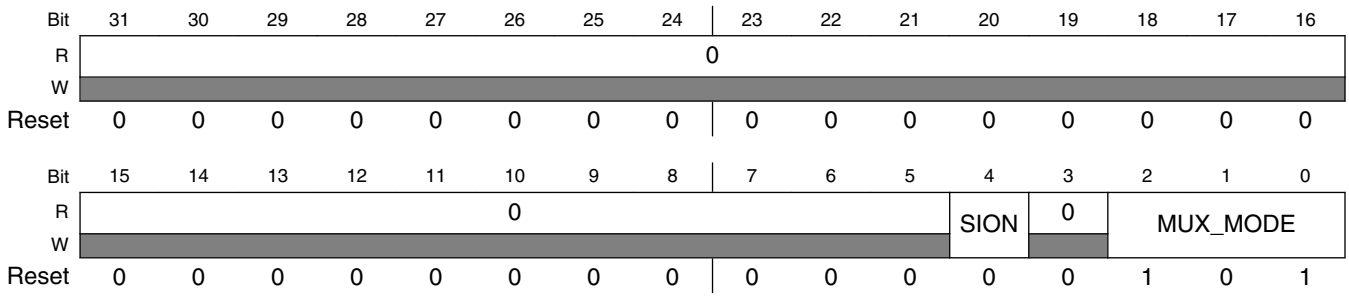
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA10 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT10. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: CSI0_DAT10. NOTE: Pad CSI0_DAT10 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA10. 001 **ALT1** — Select signal AUD3_RXC. 010 **ALT2** — Select signal ECSPI2_MISO. - Configure register IOMUXC_ECSPI2_MISO_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal UART1_TX_DATA. - Configure register IOMUXC_UART1_UART_RX_DATA_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO5_IO28. 111 **ALT7** — Select signal ARM_TRACE07.

37.4.16 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA11)

Address: 20E_0000h base + 50h offset = 20E_0050h

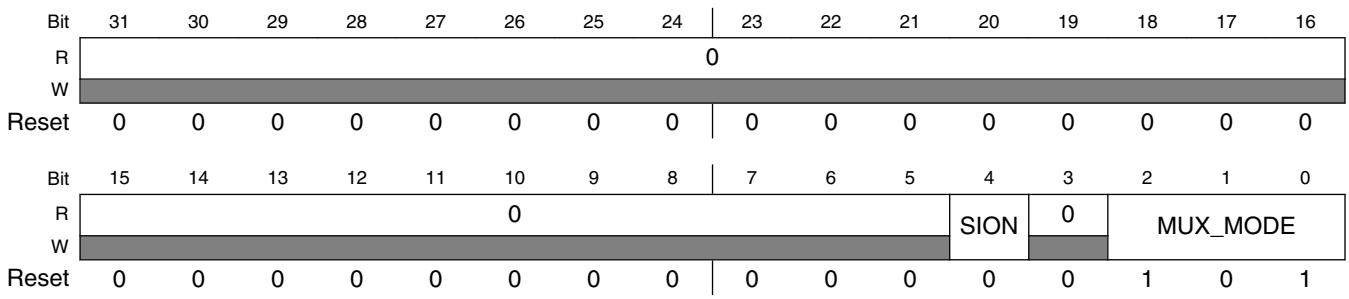


IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA11 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT11. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: CSI0_DAT11. NOTE: Pad CSI0_DAT11 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA11. 001 **ALT1** — Select signal AUD3_RXFS. 010 **ALT2** — Select signal ECSPI2_SS0. - Configure register IOMUXC_ECSPI2_SS0_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal UART1_RX_DATA. - Configure register IOMUXC_UART1_UART_RX_DATA_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO5_IO29. 111 **ALT7** — Select signal ARM_TRACE08.

37.4.17 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA12)

Address: 20E_0000h base + 54h offset = 20E_0054h



IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA12 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT12. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: CSI0_DAT12. NOTE: Pad CSI0_DAT12 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA12. 001 **ALT1** — Select signal EIM_DATA08. 011 **ALT3** — Select signal UART4_TX_DATA. - Configure register IOMUXC_UART4_UART_RX_DATA_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO5_IO30. 111 **ALT7** — Select signal ARM_TRACE09.

37.4.18 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA13)

Address: 20E_0000h base + 58h offset = 20E_0058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

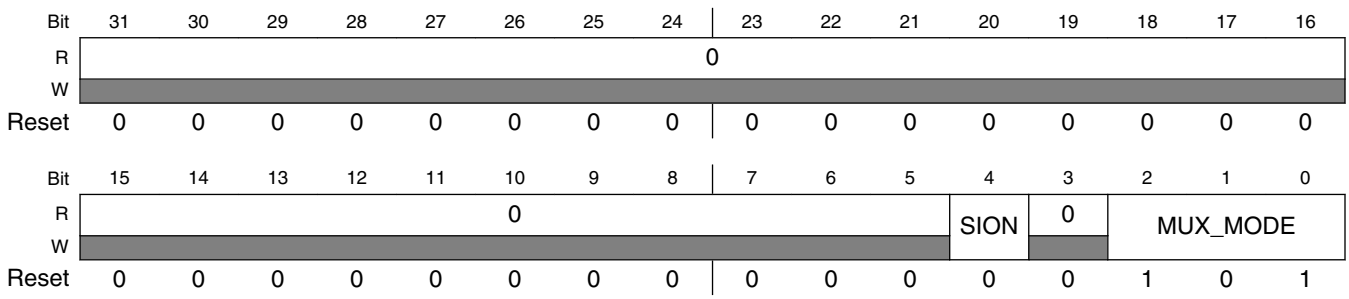
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA13 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT13. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: CSI0_DAT13. NOTE: Pad CSI0_DAT13 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA13. 001 **ALT1** — Select signal EIM_DATA09. 011 **ALT3** — Select signal UART4_RX_DATA. - Configure register IOMUXC_UART4_UART_RX_DATA_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO5_IO31. 111 **ALT7** — Select signal ARM_TRACE10.

37.4.19 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA14)

Address: 20E_0000h base + 5Ch offset = 20E_005Ch



IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA14 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT14. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: CSI0_DAT14. NOTE: Pad CSI0_DAT14 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA14. 001 **ALT1** — Select signal EIM_DATA10. 011 **ALT3** — Select signal UART5_TX_DATA. - Configure register IOMUXC_UART5_UART_RX_DATA_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO6_IO00. 111 **ALT7** — Select signal ARM_TRACE11.

37.4.20 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA15)

Address: 20E_0000h base + 60h offset = 20E_0060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

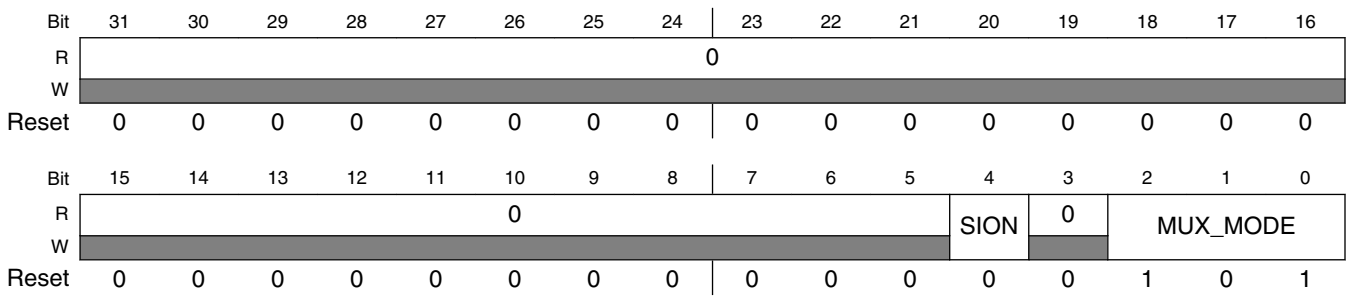
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA15 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT15. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: CSI0_DAT15. NOTE: Pad CSI0_DAT15 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA15. 001 **ALT1** — Select signal EIM_DATA11. 011 **ALT3** — Select signal UART5_RX_DATA. - Configure register IOMUXC_UART5_UART_RX_DATA_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO6_IO01. 111 **ALT7** — Select signal ARM_TRACE12.

37.4.21 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA16)

Address: 20E_0000h base + 64h offset = 20E_0064h

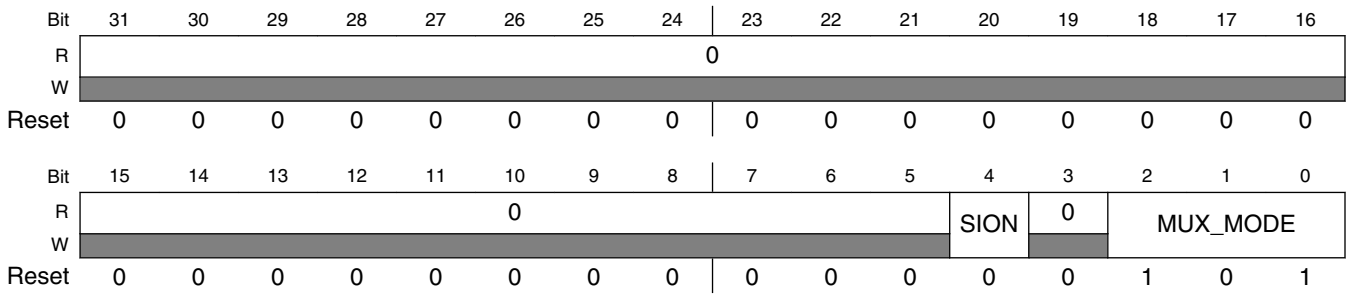


IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA16 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT16. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: CSI0_DAT16. NOTE: Pad CSI0_DAT16 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA16. 001 **ALT1** — Select signal EIM_DATA12. 011 **ALT3** — Select signal UART4_RTS_B. - Configure register IOMUXC_UART4_UART_RTS_B_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO6_IO02. 111 **ALT7** — Select signal ARM_TRACE13.

37.4.22 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA17)

Address: 20E_0000h base + 68h offset = 20E_0068h

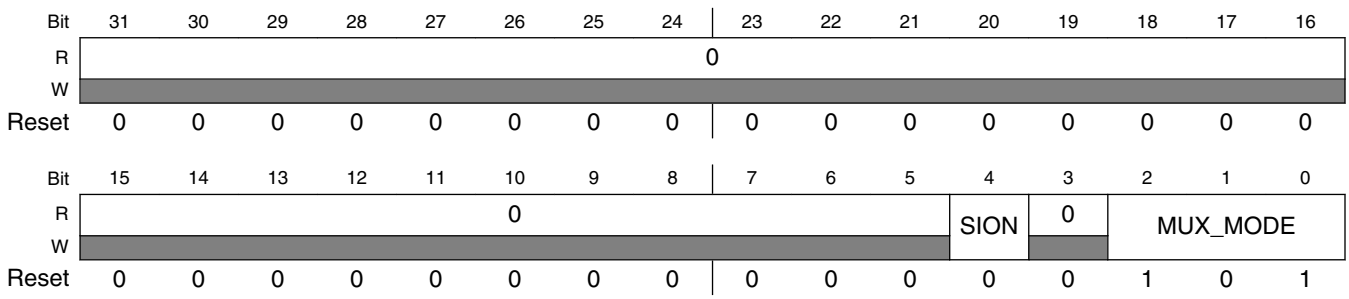


IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA17 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT17. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: CSI0_DAT17. NOTE: Pad CSI0_DAT17 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA17. 001 **ALT1** — Select signal EIM_DATA13. 011 **ALT3** — Select signal UART4_CTS_B. - Configure register IOMUXC_UART4_UART_RTS_B_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO6_IO03. 111 **ALT7** — Select signal ARM_TRACE14.

37.4.23 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA18)

Address: 20E_0000h base + 6Ch offset = 20E_006Ch

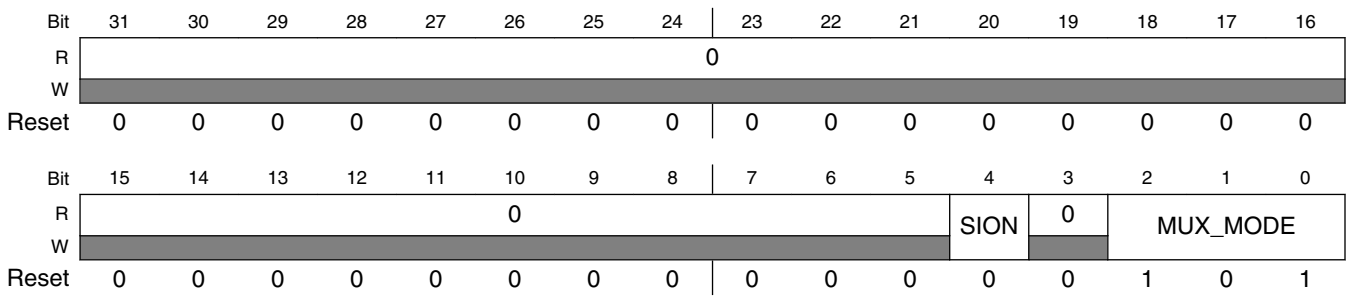


IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA18 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT18. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: CSI0_DAT18. NOTE: Pad CSI0_DAT18 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA18. 001 **ALT1** — Select signal EIM_DATA14. 011 **ALT3** — Select signal UART5_RTS_B. - Configure register IOMUXC_UART5_UART_RTS_B_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO6_IO04. 111 **ALT7** — Select signal ARM_TRACE15.

37.4.24 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA19)

Address: 20E_0000h base + 70h offset = 20E_0070h

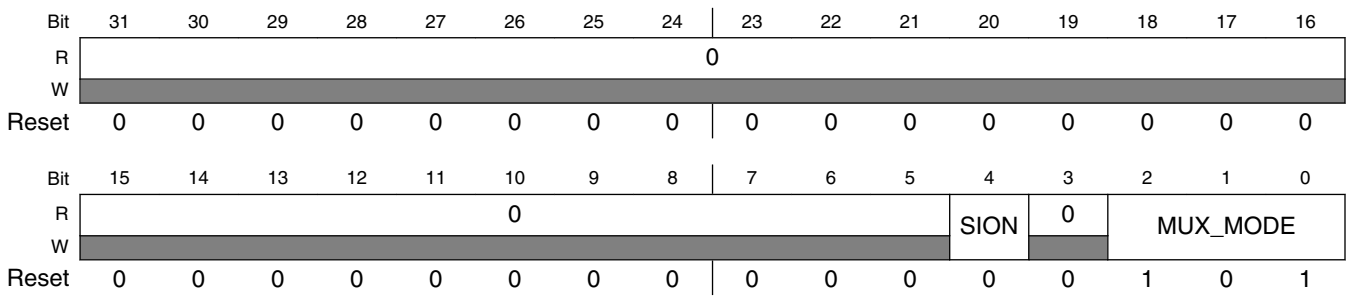


IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA19 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT19. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: CSI0_DAT19. NOTE: Pad CSI0_DAT19 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA19. 001 **ALT1** — Select signal EIM_DATA15. 011 **ALT3** — Select signal UART5_CTS_B. - Configure register IOMUXC_UART5_UART_RTS_B_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO6_IO05.

37.4.25 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA04)

Address: 20E_0000h base + 74h offset = 20E_0074h



IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA04 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT4. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: CSI0_DAT4. NOTE: Pad CSI0_DAT4 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA04. 001 **ALT1** — Select signal EIM_DATA02. 010 **ALT2** — Select signal ECSPI1_SCLK. - Configure register IOMUXC_ECSPI1_CSPI_CLK_IN_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal KEY_COL5. - Configure register IOMUXC_KEY_COL5_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal AUD3_TXC. 101 **ALT5** — Select signal GPIO5_IO22. 111 **ALT7** — Select signal ARM_TRACE01.

37.4.26 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA05)

Address: 20E_0000h base + 78h offset = 20E_0078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

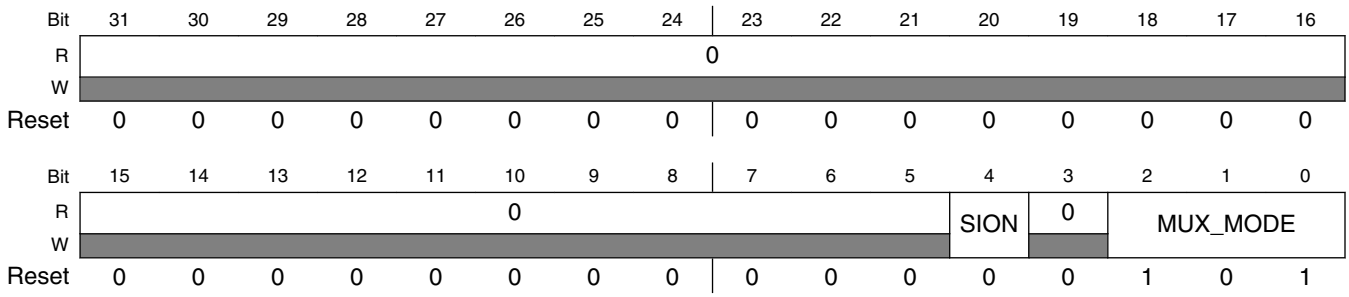
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA05 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT5. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: CSI0_DAT5. NOTE: Pad CSI0_DAT5 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA05. 001 **ALT1** — Select signal EIM_DATA03. 010 **ALT2** — Select signal ECSPI1_MOSI. - Configure register IOMUXC_ECSPI1_MOSI_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal KEY_ROW5. - Configure register IOMUXC_KEY_ROW5_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal AUD3_TXD. 101 **ALT5** — Select signal GPIO5_IO23. 111 **ALT7** — Select signal ARM_TRACE02.

37.4.27 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA06)

Address: 20E_0000h base + 7Ch offset = 20E_007Ch



IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA06 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT6. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: CSI0_DAT6. NOTE: Pad CSI0_DAT6 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA06. 001 **ALT1** — Select signal EIM_DATA04. 010 **ALT2** — Select signal ECSPI1_MISO. - Configure register IOMUXC_ECSPI1_MISO_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal KEY_COL6. - Configure register IOMUXC_KEY_COL6_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal AUD3_TXFS. 101 **ALT5** — Select signal GPIO5_IO24. 111 **ALT7** — Select signal ARM_TRACE03.

37.4.28 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA07)

Address: 20E_0000h base + 80h offset = 20E_0080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

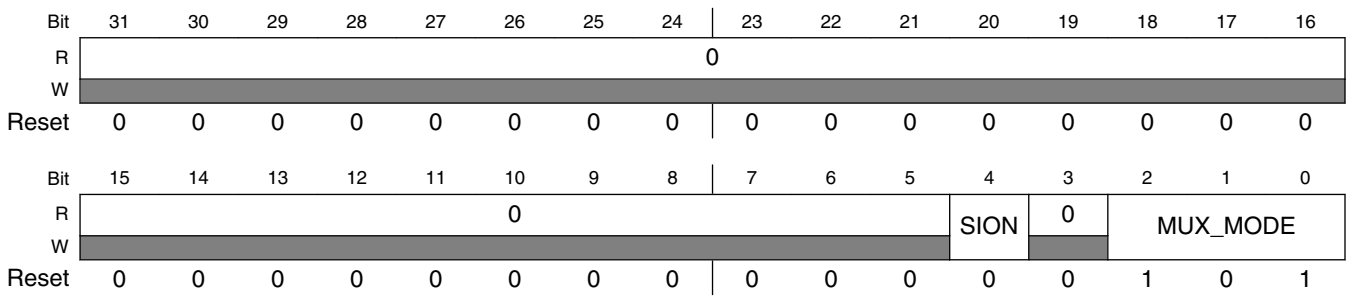
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA07 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT7. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: CSI0_DAT7. NOTE: Pad CSI0_DAT7 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA07. 001 **ALT1** — Select signal EIM_DATA05. 010 **ALT2** — Select signal ECSPI1_SS0. - Configure register IOMUXC_ECSPI1_SS0_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal KEY_ROW6. - Configure register IOMUXC_KEY_ROW6_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal AUD3_RXD. 101 **ALT5** — Select signal GPIO5_IO25. 111 **ALT7** — Select signal ARM_TRACE04.

37.4.29 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA08)

Address: 20E_0000h base + 84h offset = 20E_0084h

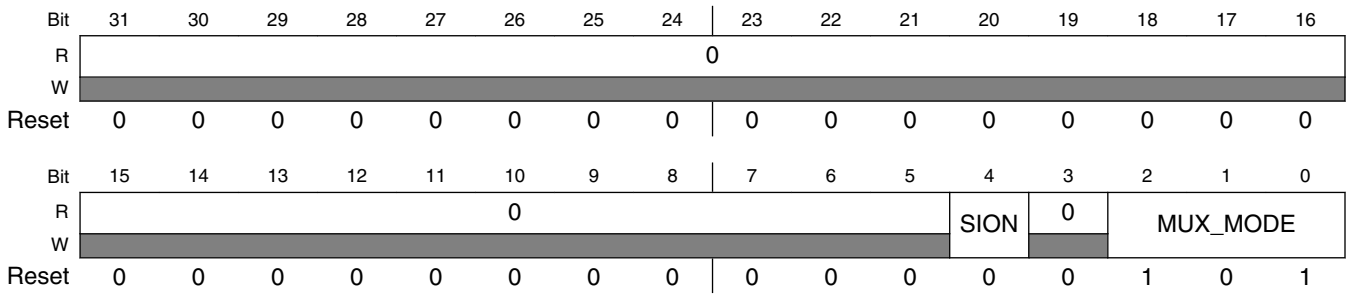


IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA08 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT8. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: CSI0_DAT8. NOTE: Pad CSI0_DAT8 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA08. 001 **ALT1** — Select signal EIM_DATA06. 010 **ALT2** — Select signal ECSPI2_SCLK. - Configure register IOMUXC_ECSPI2_CSPI_CLK_IN_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal KEY_COL7. - Configure register IOMUXC_KEY_COL7_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal I2C1_SDA. - Configure register IOMUXC_I2C1_SDA_IN_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO5_IO26. 111 **ALT7** — Select signal ARM_TRACE05.

37.4.30 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA09)

Address: 20E_0000h base + 88h offset = 20E_0088h

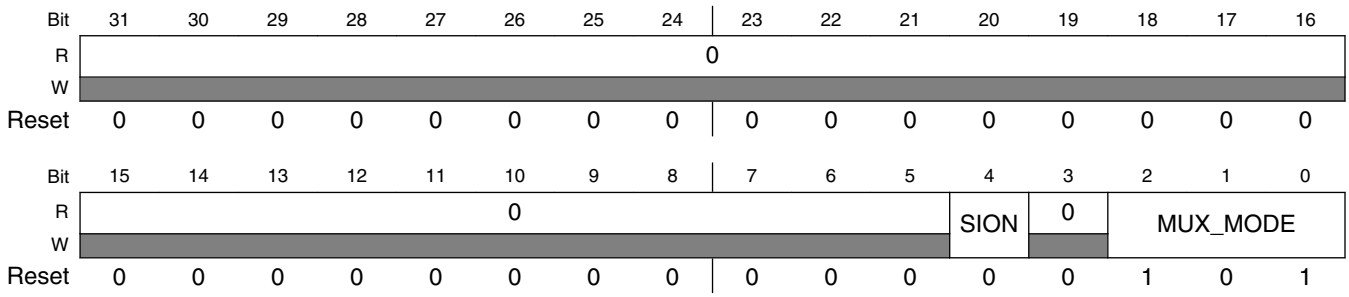


IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA09 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DAT9. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: CSI0_DAT9. NOTE: Pad CSI0_DAT9 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_CSI0_DATA09. 001 **ALT1** — Select signal EIM_DATA07. 010 **ALT2** — Select signal ECSPI2_MOSI. - Configure register IOMUXC_ECSPI2_MOSI_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal KEY_ROW7. - Configure register IOMUXC_KEY_ROW7_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal I2C1_SCL. - Configure register IOMUXC_I2C1_SCL_IN_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO5_IO27. 111 **ALT7** — Select signal ARM_TRACE06.

37.4.31 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA_EN)

Address: 20E_0000h base + 8Ch offset = 20E_008Ch



IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA_EN field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_DATA_EN. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: CSI0_DATA_EN. 000 **ALT0** — Select signal IPU1_CSI0_DATA_EN. 001 **ALT1** — Select signal EIM_DATA00. 101 **ALT5** — Select signal GPIO5_IO20. 111 **ALT7** — Select signal ARM_TRACE_CLK.

37.4.32 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_HSYNC)

Address: 20E_0000h base + 90h offset = 20E_0090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_CSI0_HSYNC field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_MCLK. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: CSI0_MCLK. 000 **ALT0** — Select signal IPU1_CSI0_HSYNC. 011 **ALT3** — Select signal CCM_CLKO1. 101 **ALT5** — Select signal GPIO5_IO19. 111 **ALT7** — Select signal ARM_TRACE_CTL.

37.4.33 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_PIXCLK)

Address: 20E_0000h base + 94h offset = 20E_0094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_CSI0_PIXCLK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_PIXCLK. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: CSI0_PIXCLK. 000 **ALT0** — Select signal IPU1_CSI0_PIXCLK. 101 **ALT5** — Select signal GPIO5_IO18. 111 **ALT7** — Select signal ARM_EVENTO.

37.4.34 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_CSI0_VSYNC)

Address: 20E_0000h base + 98h offset = 20E_0098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

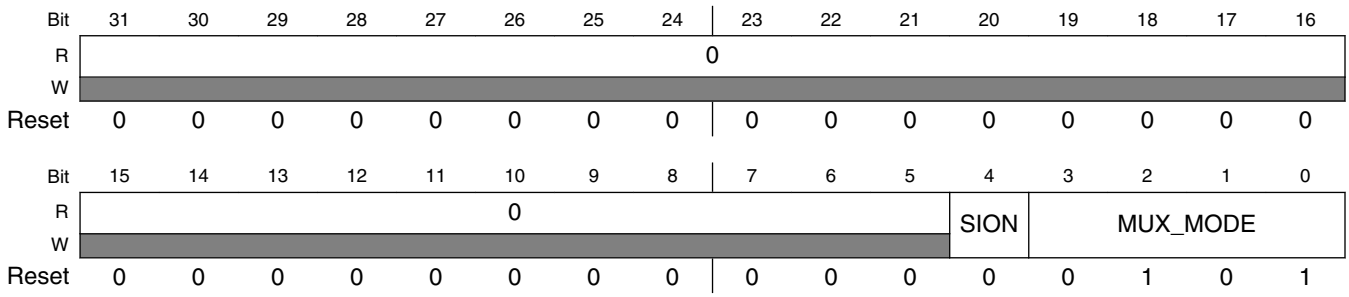
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_CSI0_VSYNC field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad CSI0_VSYNC. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: CSI0_VSYNC. 000 **ALT0** — Select signal IPU1_CSI0_VSYNC. 001 **ALT1** — Select signal EIM_DATA01. 101 **ALT5** — Select signal GPIO5_IO21. 111 **ALT7** — Select signal ARM_TRACE00.

37.4.35 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DI0_DISP_CLK)

Address: 20E_0000h base + 9Ch offset = 20E_009Ch

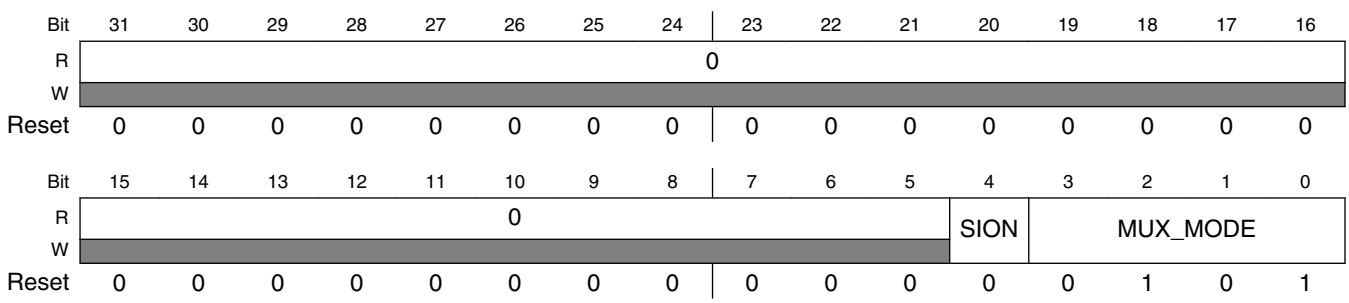


IOMUXC_SW_MUX_CTL_PAD_DI0_DISP_CLK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DI0_DISP_CLK. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: DI0_DISP_CLK. 0000 **ALT0** — Select signal IPU1_DI0_DISP_CLK. 0101 **ALT5** — Select signal GPIO4_IO16.

37.4.36 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DI0_PIN15)

Address: 20E_0000h base + A0h offset = 20E_00A0h



IOMUXC_SW_MUX_CTL_PAD_DI0_PIN15 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DI0_PIN15. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: DI0_PIN15. 0000 **ALT0** — Select signal IPU1_DI0_PIN15. 0010 **ALT2** — Select signal AUD6_TXC. 0101 **ALT5** — Select signal GPIO4_IO17.

37.4.37 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DI0_PIN02)

Address: 20E_0000h base + A4h offset = 20E_00A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DI0_PIN02 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DI0_PIN2. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: DI0_PIN2. 0000 **ALT0** — Select signal IPU1_DI0_PIN02. 0010 **ALT2** — Select signal AUD6_TXD. 0101 **ALT5** — Select signal GPIO4_IO18.

37.4.38 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DI0_PIN03)

Address: 20E_0000h base + A8h offset = 20E_00A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DI0_PIN03 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DI0_PIN3. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: DI0_PIN3. 0000 **ALT0** — Select signal IPU1_DI0_PIN03. 0010 **ALT2** — Select signal AUD6_TXFS. 0101 **ALT5** — Select signal GPIO4_IO19.

37.4.39 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DI0_PIN04)

Address: 20E_0000h base + ACh offset = 20E_00ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DI0_PIN04 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DI0_PIN4. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: DI0_PIN4. NOTE: Pad DI0_PIN4 is involved in Daisy Chain. 0000 **ALT0** — Select signal IPU1_DI0_PIN04. 0010 **ALT2** — Select signal AUD6_RXD. 0011 **ALT3** — Select signal SD1_WP. - Configure register IOMUXC_USDHC1_WP_ON_SELECT_INPUT for mode ALT3. 0101 **ALT5** — Select signal GPIO4_IO20.

37.4.40 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA00)

Address: 20E_0000h base + B0h offset = 20E_00B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA00 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.

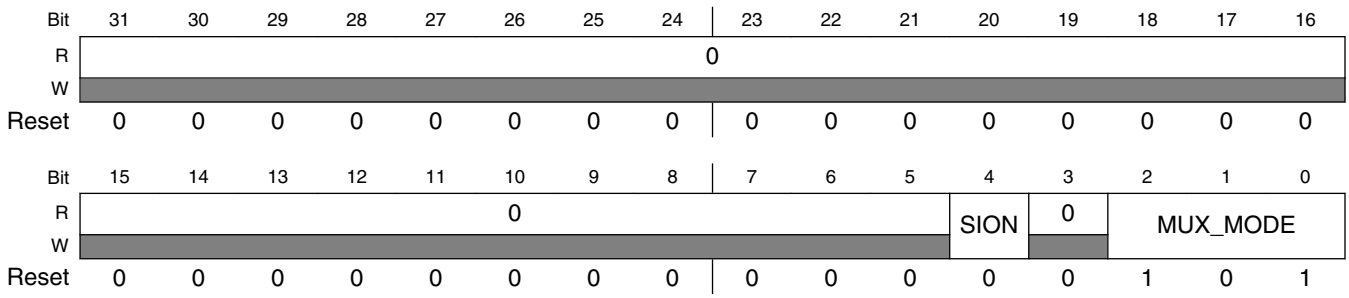
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA00 field descriptions (continued)

Field	Description
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: DISP0_DAT0. 000 **ALT0** — Select signal IPU1_DISP0_DATA00. 010 **ALT2** — Select signal ECSPI3_SCLK. 101 **ALT5** — Select signal GPIO4_IO21.

37.4.41 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA01)

Address: 20E_0000h base + B4h offset = 20E_00B4h



IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA01 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: DISP0_DAT1. 000 **ALT0** — Select signal IPU1_DISP0_DATA01. 010 **ALT2** — Select signal ECSPI3_MOSI. 101 **ALT5** — Select signal GPIO4_IO22.

37.4.42 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA10)

Address: 20E_0000h base + B8h offset = 20E_00B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA10 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT10. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: DISP0_DAT10. 000 **ALT0** — Select signal IPU1_DISP0_DATA10. 101 **ALT5** — Select signal GPIO4_IO31.

37.4.43 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA11)

Address: 20E_0000h base + BCh offset = 20E_00BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA11 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT11. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: DISP0_DAT11. 000 **ALT0** — Select signal IPU1_DISP0_DATA11. 101 **ALT5** — Select signal GPIO5_IO05.

37.4.44 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA12)

Address: 20E_0000h base + C0h offset = 20E_00C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA12 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT12. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: DISP0_DAT12.

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA12 field descriptions (continued)

Field	Description
	000 **ALT0** — Select signal IPU1_DISP0_DATA12. 101 **ALT5** — Select signal GPIO5_IO06.

37.4.45 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA13)

Address: 20E_0000h base + C4h offset = 20E_00C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

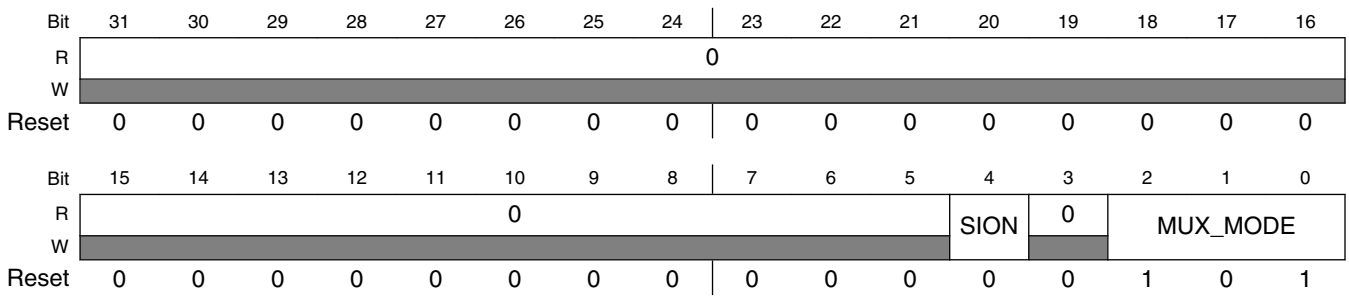
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA13 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT13. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: DISP0_DAT13. NOTE: Pad DISP0_DAT13 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_DISP0_DATA13. 011 **ALT3** — Select signal AUD5_RXFS. - Configure register IOMUXC_AUD5_INPUT_RXFS_AMX_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO5_IO07.

37.4.46 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA14)

Address: 20E_0000h base + C8h offset = 20E_00C8h



IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA14 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT14. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: DISP0_DAT14. NOTE: Pad DISP0_DAT14 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_DISP0_DATA14. 011 **ALT3** — Select signal AUD5_RXC. - Configure register IOMUXC_AUD5_INPUT_RXCLK_AMX_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO5_IO08.

37.4.47 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA15)

Address: 20E_0000h base + CCh offset = 20E_00CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

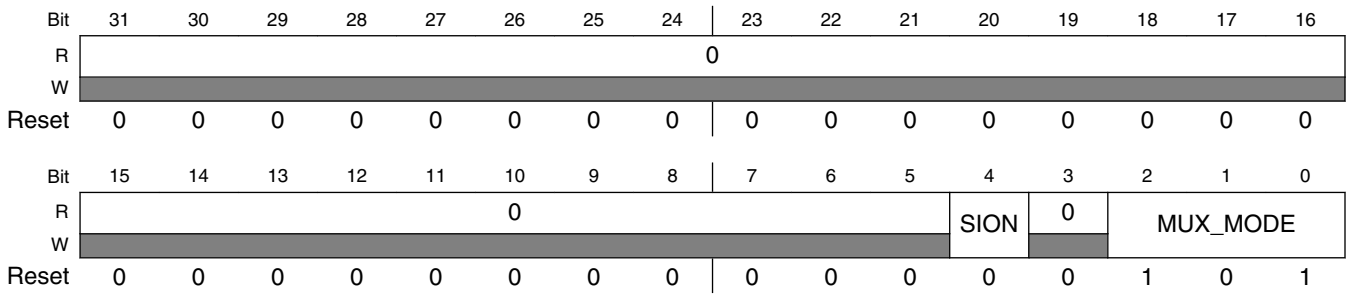
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA15 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT15. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: DISP0_DAT15. NOTE: Pad DISP0_DAT15 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_DISP0_DATA15. 010 **ALT2** — Select signal ECSPI1_SS1. - Configure register IOMUXC_ECSPI1_SS1_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal ECSPI2_SS1. - Configure register IOMUXC_ECSPI2_SS1_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO5_IO09.

37.4.48 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA16)

Address: 20E_0000h base + D0h offset = 20E_00D0h

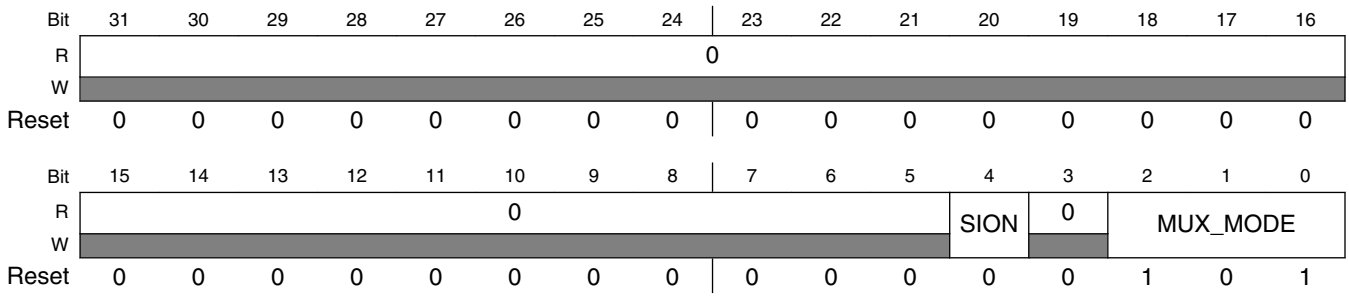


IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA16 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT16. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: DISP0_DAT16. NOTE: Pad DISP0_DAT16 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_DISP0_DATA16. 010 **ALT2** — Select signal ECSPI2_MOSI. - Configure register IOMUXC_ECSPI2_MOSI_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal AUD5_TXC. - Configure register IOMUXC_AUD5_INPUT_TXCLK_AMX_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal SDMA_EXT_EVENT0. - Configure register IOMUXC_SDMA_EVENTS14_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO5_IO10.

37.4.49 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA17)

Address: 20E_0000h base + D4h offset = 20E_00D4h

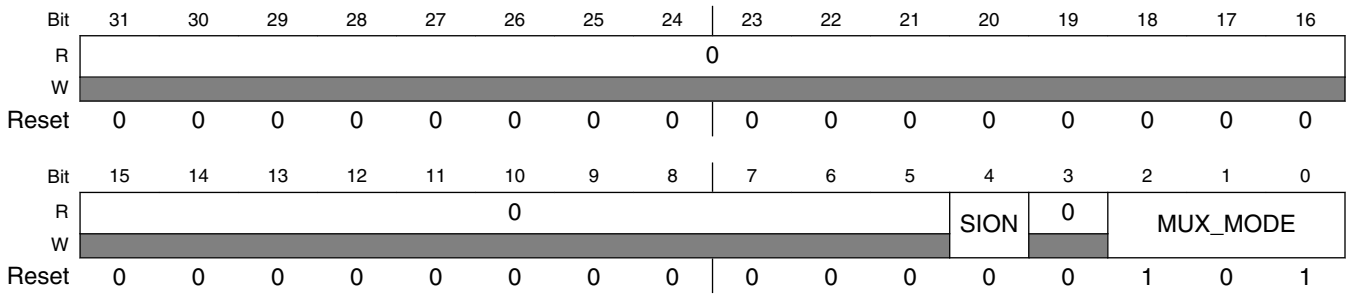


IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA17 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT17. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: DISP0_DAT17. NOTE: Pad DISP0_DAT17 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_DISP0_DATA17. 010 **ALT2** — Select signal ECSPI2_MISO. - Configure register IOMUXC_ECSPI2_MISO_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal AUD5_TXD. - Configure register IOMUXC_AUD5_INPUT_DB_AMX_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal SDMA_EXT_EVENT1. - Configure register IOMUXC_SDMA_EVENTS15_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO5_IO11.

37.4.50 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA18)

Address: 20E_0000h base + D8h offset = 20E_00D8h



IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA18 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT18. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: DISP0_DAT18. NOTE: Pad DISP0_DAT18 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_DISP0_DATA18. 010 **ALT2** — Select signal ECSPI2_SS0. - Configure register IOMUXC_ECSPI2_SS0_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal AUD5_TXFS. - Configure register IOMUXC_AUD5_INPUT_TXFS_AMX_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal AUD4_RXFS. - Configure register IOMUXC_AUD4_INPUT_RXFS_AMX_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO5_IO12. 111 **ALT7** — Select signal EIM_CS2_B.

37.4.51 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA19)

Address: 20E_0000h base + DCh offset = 20E_00DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

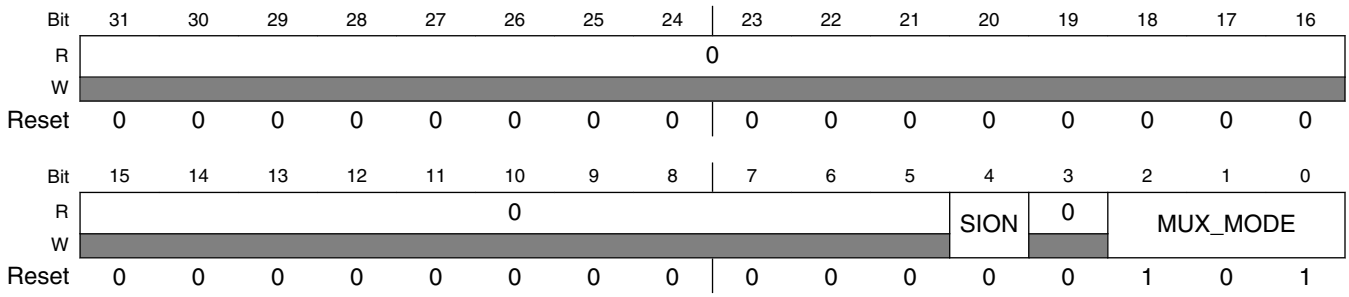
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA19 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT19. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: DISP0_DAT19. NOTE: Pad DISP0_DAT19 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_DISP0_DATA19. 010 **ALT2** — Select signal ECSPI2_SCLK. - Configure register IOMUXC_ECSPI2_CSPI_CLK_IN_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal AUD5_RXD. - Configure register IOMUXC_AUD5_INPUT_DA_AMX_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal AUD4_RXC. - Configure register IOMUXC_AUD4_INPUT_RXCLK_AMX_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO5_IO13. 111 **ALT7** — Select signal EIM_CS3_B.

37.4.52 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA02)

Address: 20E_0000h base + E0h offset = 20E_00E0h



IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA02 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT2. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: DISP0_DAT2. 000 **ALT0** — Select signal IPU1_DISP0_DATA02. 010 **ALT2** — Select signal ECSPI3_MISO. 101 **ALT5** — Select signal GPIO4_IO23.

37.4.53 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA20)

Address: 20E_0000h base + E4h offset = 20E_00E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

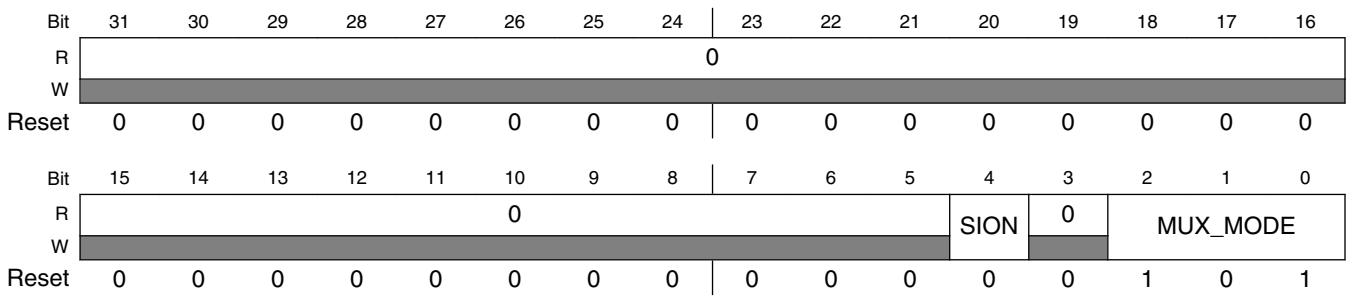
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA20 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT20. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: DISP0_DAT20. NOTE: Pad DISP0_DAT20 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_DISP0_DATA20. 010 **ALT2** — Select signal ECSPI1_SCLK. - Configure register IOMUXC_ECSPI1_CSPI_CLK_IN_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal AUD4_TXC. - Configure register IOMUXC_AUD4_INPUT_TXCLK_AMX_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO5_IO14.

37.4.54 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA21)

Address: 20E_0000h base + E8h offset = 20E_00E8h



IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA21 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT21. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: DISP0_DAT21. NOTE: Pad DISP0_DAT21 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_DISP0_DATA21. 010 **ALT2** — Select signal ECSPI1_MOSI. - Configure register IOMUXC_ECSPI1_MOSI_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal AUD4_TXD. - Configure register IOMUXC_AUD4_INPUT_DB_AMX_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO5_IO15.

37.4.55 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA22)

Address: 20E_0000h base + ECh offset = 20E_00ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

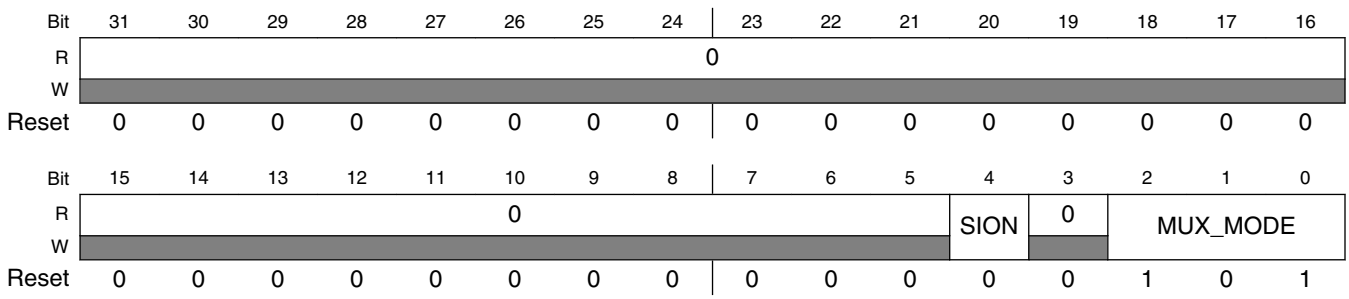
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA22 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT22. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: DISP0_DAT22. NOTE: Pad DISP0_DAT22 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_DISP0_DATA22. 010 **ALT2** — Select signal ECSPI1_MISO. - Configure register IOMUXC_ECSPI1_MISO_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal AUD4_TXFS. - Configure register IOMUXC_AUD4_INPUT_TXFS_AMX_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO5_IO16.

37.4.56 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA23)

Address: 20E_0000h base + F0h offset = 20E_00F0h



IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA23 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT23. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: DISP0_DAT23. NOTE: Pad DISP0_DAT23 is involved in Daisy Chain. 000 **ALT0** — Select signal IPU1_DISP0_DATA23. 010 **ALT2** — Select signal ECSPI1_SS0. - Configure register IOMUXC_ECSPI1_SS0_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal AUD4_RXD. - Configure register IOMUXC_AUD4_INPUT_DA_AMX_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO5_IO17.

37.4.57 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA03)

Address: 20E_0000h base + F4h offset = 20E_00F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA03 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT3. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: DISP0_DAT3. 000 **ALT0** — Select signal IPU1_DISP0_DATA03. 010 **ALT2** — Select signal ECSPI3_SS0. 101 **ALT5** — Select signal GPIO4_IO24.

37.4.58 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA04)

Address: 20E_0000h base + F8h offset = 20E_00F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA04 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT4. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: DISP0_DAT4. 000 **ALT0** — Select signal IPU1_DISP0_DATA04. 010 **ALT2** — Select signal ECSPI3_SS1. 101 **ALT5** — Select signal GPIO4_IO25.

37.4.59 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA05)

Address: 20E_0000h base + FCh offset = 20E_00FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA05 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT5. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: DISP0_DAT5. 000 **ALT0** — Select signal IPU1_DISP0_DATA05. 010 **ALT2** — Select signal ECSPI3_SS2. 011 **ALT3** — Select signal AUD6_RXFS. 101 **ALT5** — Select signal GPIO4_IO26.

37.4.60 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA06)

Address: 20E_0000h base + 100h offset = 20E_0100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA06 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT6. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: DISP0_DAT6. 000 **ALT0** — Select signal IPU1_DISP0_DATA06. 010 **ALT2** — Select signal ECSPI3_SS3. 011 **ALT3** — Select signal AUD6_RXC. 101 **ALT5** — Select signal GPIO4_IO27.

37.4.61 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA07)

Address: 20E_0000h base + 104h offset = 20E_0104h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA07 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT7. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: DISP0_DAT7. 000 **ALT0** — Select signal IPU1_DISP0_DATA07. 010 **ALT2** — Select signal ECSPI3_RDY. 101 **ALT5** — Select signal GPIO4_IO28.

37.4.62 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA08)

Address: 20E_0000h base + 108h offset = 20E_0108h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA08 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT8. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: DISP0_DAT8. 000 **ALT0** — Select signal IPU1_DISP0_DATA08. 010 **ALT2** — Select signal PWM1_OUT. 011 **ALT3** — Select signal WDOG1_B. 101 **ALT5** — Select signal GPIO4_IO29.

37.4.63 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA09)

Address: 20E_0000h base + 10Ch offset = 20E_010Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_DISP0_DATA09 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad DISP0_DAT9. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: DISP0_DAT9. 000 **ALT0** — Select signal IPU1_DISP0_DATA09. 010 **ALT2** — Select signal PWM2_OUT. 011 **ALT3** — Select signal WDOG2_B. 101 **ALT5** — Select signal GPIO4_IO30.

37.4.64 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR16)

Address: 20E_0000h base + 110h offset = 20E_0110h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR16 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_A16. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_A16. NOTE: Pad EIM_A16 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_ADDR16. 0001 **ALT1** — Select signal IPU1_DI1_DISP_CLK. 0010 **ALT2** — Select signal IPU1_CSI1_PIXCLK. - Configure register IOMUXC_IPU1_SENS1_PIX_CLK_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO2_IO22. 0111 **ALT7** — Select signal SRC_BOOT_CFG16. 1000 **ALT8** — Select signal EPDC_DATA00.

37.4.65 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR17)

Address: 20E_0000h base + 114h offset = 20E_0114h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR17 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_A17. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_A17. NOTE: Pad EIM_A17 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_ADDR17. 0001 **ALT1** — Select signal IPU1_DISP1_DATA12. 0010 **ALT2** — Select signal IPU1_CSI1_DATA12. - Configure register IOMUXC_IPU1_SENS1_DATA12_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO2_IO21. 0111 **ALT7** — Select signal SRC_BOOT_CFG17. 1000 **ALT8** — Select signal EPDC_PWR_STAT.

37.4.66 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR18)

Address: 20E_0000h base + 118h offset = 20E_0118h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

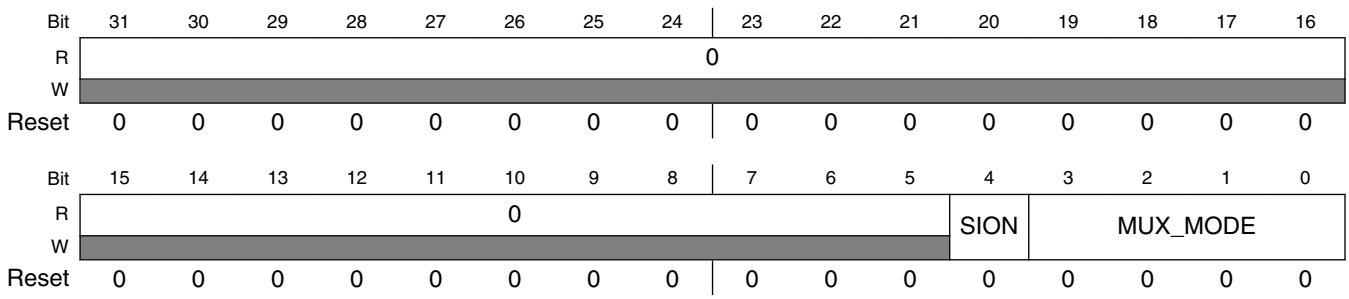
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR18 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_A18. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_A18. NOTE: Pad EIM_A18 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_ADDR18. 0001 **ALT1** — Select signal IPU1_DISP1_DATA13. 0010 **ALT2** — Select signal IPU1_CSI1_DATA13. - Configure register IOMUXC_IPU1_SENS1_DATA13_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO2_IO20. 0111 **ALT7** — Select signal SRC_BOOT_CFG18. 1000 **ALT8** — Select signal EPDC_PWR_CTRL0.

37.4.67 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR19)

Address: 20E_0000h base + 11Ch offset = 20E_011Ch

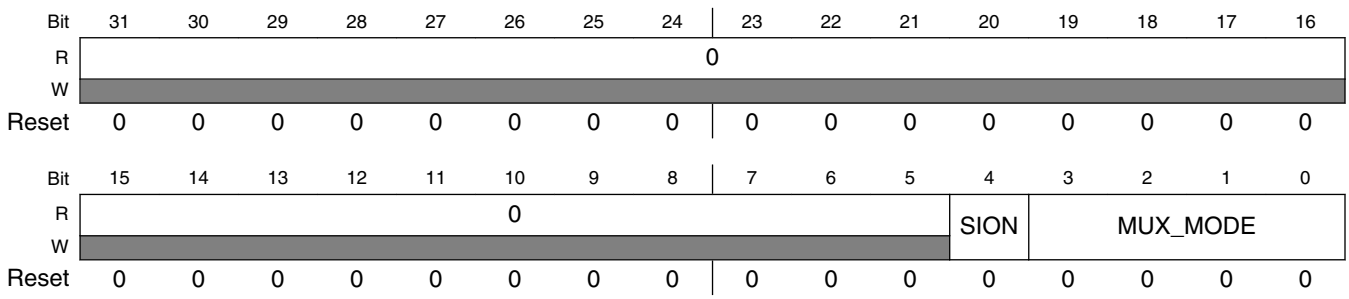


IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR19 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_A19. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_A19. NOTE: Pad EIM_A19 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_ADDR19. 0001 **ALT1** — Select signal IPU1_DISP1_DATA14. 0010 **ALT2** — Select signal IPU1_CSI1_DATA14. - Configure register IOMUXC_IPU1_SENS1_DATA14_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO2_IO19. 0111 **ALT7** — Select signal SRC_BOOT_CFG19. 1000 **ALT8** — Select signal EPDC_PWR_CTRL1.

37.4.68 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR20)

Address: 20E_0000h base + 120h offset = 20E_0120h

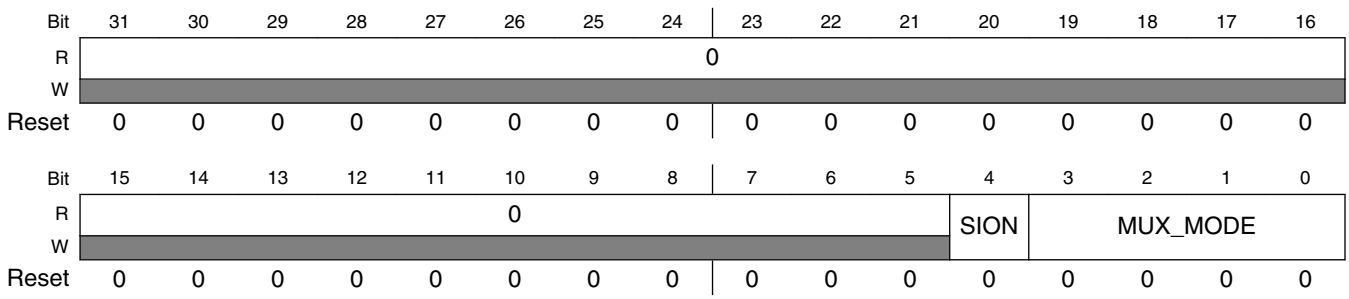


IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR20 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_A20. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_A20. NOTE: Pad EIM_A20 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_ADDR20. 0001 **ALT1** — Select signal IPU1_DISP1_DATA15. 0010 **ALT2** — Select signal IPU1_CSI1_DATA15. - Configure register IOMUXC_IPU1_SENS1_DATA15_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO2_IO18. 0111 **ALT7** — Select signal SRC_BOOT_CFG20. 1000 **ALT8** — Select signal EPDC_PWR_CTRL2.

37.4.69 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR21)

Address: 20E_0000h base + 124h offset = 20E_0124h



IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR21 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_A21. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_A21. NOTE: Pad EIM_A21 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_ADDR21. 0001 **ALT1** — Select signal IPU1_DISP1_DATA16. 0010 **ALT2** — Select signal IPU1_CSI1_DATA16. - Configure register IOMUXC_IPU1_SENS1_DATA16_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO2_IO17. 0111 **ALT7** — Select signal SRC_BOOT_CFG21. 1000 **ALT8** — Select signal EPDC_GDCLK.

37.4.70 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR22)

Address: 20E_0000h base + 128h offset = 20E_0128h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR22 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_A22. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_A22. NOTE: Pad EIM_A22 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_ADDR22. 0001 **ALT1** — Select signal IPU1_DISP1_DATA17. 0010 **ALT2** — Select signal IPU1_CSI1_DATA17. - Configure register IOMUXC_IPU1_SENS1_DATA17_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO2_IO16. 0111 **ALT7** — Select signal SRC_BOOT_CFG22. 1000 **ALT8** — Select signal EPDC_GDSP.

37.4.71 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR23)

Address: 20E_0000h base + 12Ch offset = 20E_012Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR23 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_A23. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: EIM_A23. NOTE: Pad EIM_A23 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_ADDR23. 0001 **ALT1** — Select signal IPU1_DISP1_DATA18. 0010 **ALT2** — Select signal IPU1_CSI1_DATA18. - Configure register IOMUXC_IPU1_SENS1_DATA18_SELECT_INPUT for mode ALT2. 0100 **ALT4** — Select signal IPU1_SISG3. 0101 **ALT5** — Select signal GPIO6_IO06. 0111 **ALT7** — Select signal SRC_BOOT_CFG23. 1000 **ALT8** — Select signal EPDC_GDOE.

37.4.72 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR24)

Address: 20E_0000h base + 130h offset = 20E_0130h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR24 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_A24. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: EIM_A24. NOTE: Pad EIM_A24 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_ADDR24. 0001 **ALT1** — Select signal IPU1_DISP1_DATA19. 0010 **ALT2** — Select signal IPU1_CSI1_DATA19. - Configure register IOMUXC_IPU1_SENS1_DATA19_SELECT_INPUT for mode ALT2. 0100 **ALT4** — Select signal IPU1_SISG2. 0101 **ALT5** — Select signal GPIO5_IO04. 0111 **ALT7** — Select signal SRC_BOOT_CFG24. 1000 **ALT8** — Select signal EPDC_GDRL.

37.4.73 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR25)

Address: 20E_0000h base + 134h offset = 20E_0134h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_ADDR25 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_A25. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: EIM_A25. NOTE: Pad EIM_A25 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_ADDR25. 0001 **ALT1** — Select signal ECSPI4_SS1. 0010 **ALT2** — Select signal ECSPI2_RDY. 0011 **ALT3** — Select signal IPU1_DI1_PIN12. 0100 **ALT4** — Select signal IPU1_DI0_D1_CS. 0101 **ALT5** — Select signal GPIO5_IO02. 0110 **ALT6** — Select signal HDMI_TX_CEC_LINE. - Configure register IOMUXC_HDMI_ICECIN_SELECT_INPUT for mode ALT6. 1000 **ALT8** — Select signal EPDC_DATA15. 1001 **ALT9** — Select signal EIM_ACLK_FREERUN.

37.4.74 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_BCLK)

Address: 20E_0000h base + 138h offset = 20E_0138h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_BCLK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_BCLK. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: EIM_BCLK. 0000 **ALT0** — Select signal EIM_BCLK. 0001 **ALT1** — Select signal IPU1_DI1_PIN16. 0101 **ALT5** — Select signal GPIO6_IO31. 1000 **ALT8** — Select signal EPDC_SDCE9.

37.4.75 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_CS0_B)

Address: 20E_0000h base + 13Ch offset = 20E_013Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_CS0_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_CS0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_CS0. NOTE: Pad EIM_CS0 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_CS0_B. 0001 **ALT1** — Select signal IPU1_DI1_PIN05. 0010 **ALT2** — Select signal ECSPI2_SCLK. - Configure register IOMUXC_ECSPI2_CSPI_CLK_IN_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO2_IO23. 1000 **ALT8** — Select signal EPDC_DATA06.

37.4.76 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_CS1_B)

Address: 20E_0000h base + 140h offset = 20E_0140h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_CS1_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_CS1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field.

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_EIM_CS1_B field descriptions (continued)

Field	Description
	Select 1 of 5 iomux modes to be used for pad: EIM_CS1. NOTE: Pad EIM_CS1 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_CS1_B. 0001 **ALT1** — Select signal IPU1_DI1_PIN06. 0010 **ALT2** — Select signal ECSPI2_MOSI. - Configure register IOMUXC_ECSPI2_MOSI_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO2_IO24. 1000 **ALT8** — Select signal EPDC_DATA08.

37.4.77 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA16)

Address: 20E_0000h base + 144h offset = 20E_0144h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_EIM_DATA16 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D16. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D16. NOTE: Pad EIM_D16 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_DATA16. 0001 **ALT1** — Select signal ECSPI1_SCLK. - Configure register IOMUXC_ECSPI1_CSPI_CLK_IN_SELECT_INPUT for mode ALT1. 0010 **ALT2** — Select signal IPU1_DI0_PIN05. 0011 **ALT3** — Select signal IPU1_CSI1_DATA18.

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_EIM_DATA16 field descriptions (continued)

Field	Description
	- Configure register IOMUXC_IPU1_SENS1_DATA18_SELECT_INPUT for mode ALT3. 0100 **ALT4** — Select signal HDMI_TX_DDC_SDA. - Configure register IOMUXC_HDMI_II2C_DATAIN_SELECT_INPUT for mode ALT4. 0101 **ALT5** — Select signal GPIO3_IO16. 0110 **ALT6** — Select signal I2C2_SDA. - Configure register IOMUXC_I2C2_SDA_IN_SELECT_INPUT for mode ALT6. 1000 **ALT8** — Select signal EPDC_DATA10.

37.4.78 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA17)

Address: 20E_0000h base + 148h offset = 20E_0148h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_EIM_DATA17 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D17. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D17. NOTE: Pad EIM_D17 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_DATA17. 0001 **ALT1** — Select signal ECSPI1_MISO. - Configure register IOMUXC_ECSPI1_MISO_SELECT_INPUT for mode ALT1. 0010 **ALT2** — Select signal IPU1_DI0_PIN06. 0011 **ALT3** — Select signal IPU1_CSI1_PIXCLK. - Configure register IOMUXC_IPU1_SENS1_PIX_CLK_SELECT_INPUT for mode ALT3.

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_EIM_DATA17 field descriptions (continued)

Field	Description
	0100 **ALT4** — Select signal DCIC1_OUT. 0101 **ALT5** — Select signal GPIO3_IO17. 0110 **ALT6** — Select signal I2C3_SCL. - Configure register IOMUXC_I2C3_SCL_IN_SELECT_INPUT for mode ALT6. 1000 **ALT8** — Select signal EPDC_VCOM0.

37.4.79 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA18)

Address: 20E_0000h base + 14Ch offset = 20E_014Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_EIM_DATA18 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D18. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D18. NOTE: Pad EIM_D18 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_DATA18. 0001 **ALT1** — Select signal ECSPI1_MOSI. - Configure register IOMUXC_ECSPI1_MOSI_SELECT_INPUT for mode ALT1. 0010 **ALT2** — Select signal IPU1_DI0_PIN07. 0011 **ALT3** — Select signal IPU1_CSI1_DATA17. - Configure register IOMUXC_IPU1_SENS1_DATA17_SELECT_INPUT for mode ALT3. 0100 **ALT4** — Select signal IPU1_DI1_D0_CS. 0101 **ALT5** — Select signal GPIO3_IO18.

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_EIM_DATA18 field descriptions (continued)

Field	Description
	0110 **ALT6** — Select signal I2C3_SDA. - Configure register IOMUXC_I2C3_SDA_IN_SELECT_INPUT for mode ALT6. 1000 **ALT8** — Select signal EPDC_VCOM1.

37.4.80 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA19)

Address: 20E_0000h base + 150h offset = 20E_0150h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

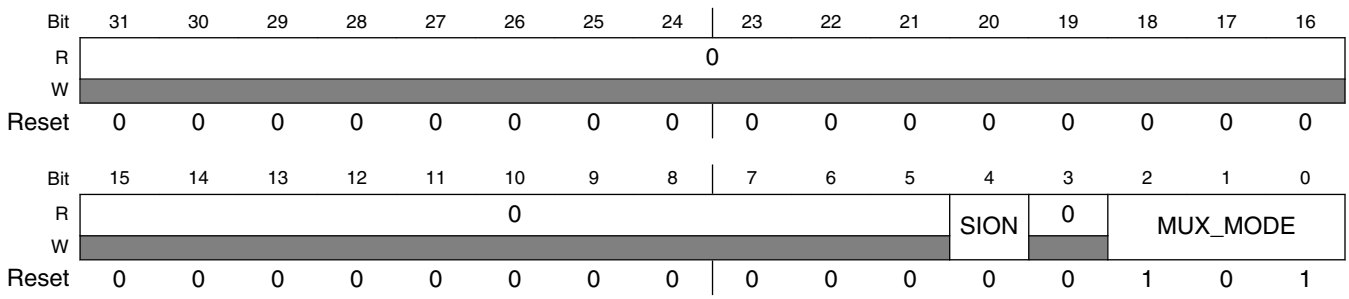
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_EIM_DATA19 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D19. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D19. NOTE: Pad EIM_D19 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_DATA19. 0001 **ALT1** — Select signal ECSPI1_SS1. - Configure register IOMUXC_ECSPI1_SS1_SELECT_INPUT for mode ALT1. 0010 **ALT2** — Select signal IPU1_DI0_PIN08. 0011 **ALT3** — Select signal IPU1_CSI1_DATA16. - Configure register IOMUXC_IPU1_SENS1_DATA16_SELECT_INPUT for mode ALT3. 0100 **ALT4** — Select signal UART1_CTS_B. - Configure register IOMUXC_UART1_UART_RTS_B_SELECT_INPUT for mode ALT4. 0101 **ALT5** — Select signal GPIO3_IO19. 0110 **ALT6** — Select signal EPIT1_OUT. 1000 **ALT8** — Select signal EPDC_DATA12.

37.4.81 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA20)

Address: 20E_0000h base + 154h offset = 20E_0154h

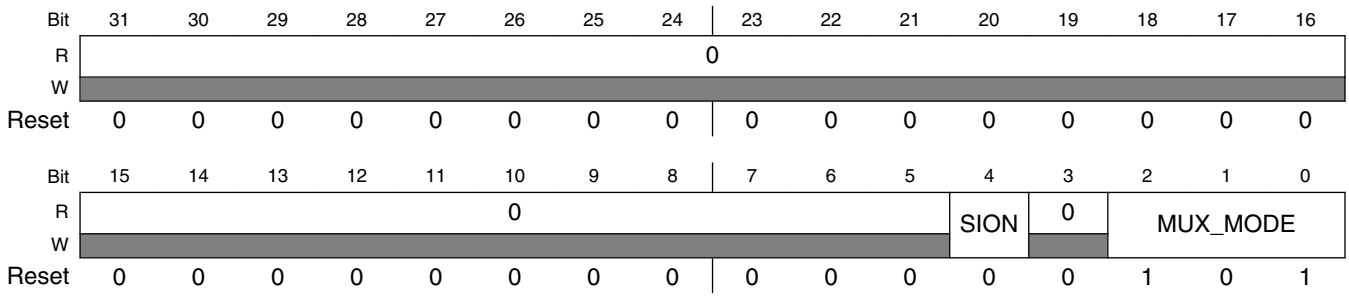


IOMUXC_SW_MUX_CTL_PAD_EIM_DATA20 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D20. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: EIM_D20. NOTE: Pad EIM_D20 is involved in Daisy Chain. 000 **ALT0** — Select signal EIM_DATA20. 001 **ALT1** — Select signal ECSPI4_SS0. - Configure register IOMUXC_ECSPI4_SS0_SELECT_INPUT for mode ALT1. 010 **ALT2** — Select signal IPU1_DI0_PIN16. 011 **ALT3** — Select signal IPU1_CSI1_DATA15. - Configure register IOMUXC_IPU1_SENS1_DATA15_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal UART1_RTS_B. - Configure register IOMUXC_UART1_UART_RTS_B_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO3_IO20. 110 **ALT6** — Select signal EPIT2_OUT.

37.4.82 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA21)

Address: 20E_0000h base + 158h offset = 20E_0158h

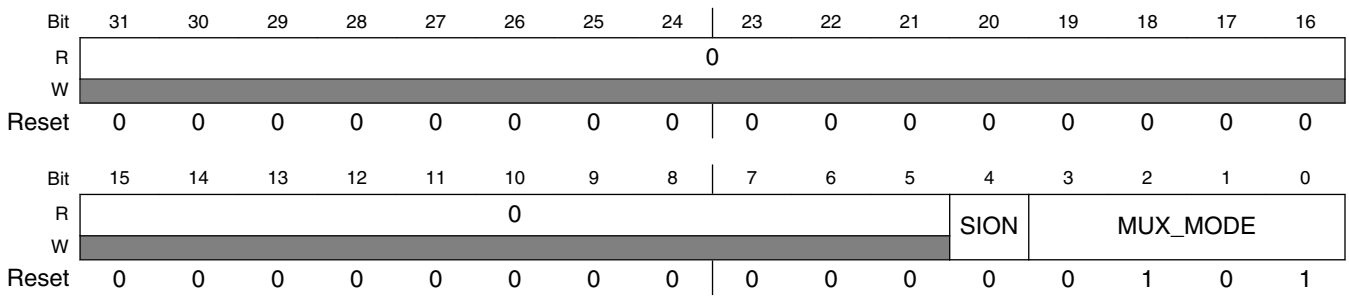


IOMUXC_SW_MUX_CTL_PAD_EIM_DATA21 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D21. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D21. NOTE: Pad EIM_D21 is involved in Daisy Chain. 000 **ALT0** — Select signal EIM_DATA21. 001 **ALT1** — Select signal ECSPI4_SCLK. 010 **ALT2** — Select signal IPU1_DI0_PIN17. 011 **ALT3** — Select signal IPU1_CSI1_DATA11. - Configure register IOMUXC_IPU1_SENS1_DATA11_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal USB_OTG_OC. - Configure register IOMUXC_USB_OTG_OC_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO3_IO21. 110 **ALT6** — Select signal I2C1_SCL. - Configure register IOMUXC_I2C1_SCL_IN_SELECT_INPUT for mode ALT6. 111 **ALT7** — Select signal SPDIF_IN. - Configure register IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT for mode ALT7.

37.4.83 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA22)

Address: 20E_0000h base + 15Ch offset = 20E_015Ch



IOMUXC_SW_MUX_CTL_PAD_EIM_DATA22 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D22. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D22. NOTE: Pad EIM_D22 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_DATA22. 0001 **ALT1** — Select signal ECSPI4_MISO. 0010 **ALT2** — Select signal IPU1_DI0_PIN01. 0011 **ALT3** — Select signal IPU1_CSI1_DATA10. - Configure register IOMUXC_IPU1_SENS1_DATA10_SELECT_INPUT for mode ALT3. 0100 **ALT4** — Select signal USB_OTG_PWR. 0101 **ALT5** — Select signal GPIO3_IO22. 0110 **ALT6** — Select signal SPDIF_OUT. 1000 **ALT8** — Select signal EPDC_SDCE6.

37.4.84 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA23)

Address: 20E_0000h base + 160h offset = 20E_0160h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_EIM_DATA23 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D23. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: EIM_D23. NOTE: Pad EIM_D23 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_DATA23. 0001 **ALT1** — Select signal IPU1_DI0_D0_CS. 0010 **ALT2** — Select signal UART3_CTS_B. - Configure register IOMUXC_UART3_UART_RTS_B_SELECT_INPUT for mode ALT2. 0011 **ALT3** — Select signal UART1_DCD_B. 0100 **ALT4** — Select signal IPU1_CSI1_DATA_EN. - Configure register IOMUXC_IPU1_SENS1_DATA_EN_SELECT_INPUT for mode ALT4. 0101 **ALT5** — Select signal GPIO3_IO23. 0110 **ALT6** — Select signal IPU1_DI1_PIN02. 0111 **ALT7** — Select signal IPU1_DI1_PIN14. 1000 **ALT8** — Select signal EPDC_DATA11.

37.4.85 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA24)

Address: 20E_0000h base + 164h offset = 20E_0164h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_EIM_DATA24 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D24. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: EIM_D24. NOTE: Pad EIM_D24 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_DATA24. 0001 **ALT1** — Select signal ECSPI4_SS2. 0010 **ALT2** — Select signal UART3_TX_DATA. - Configure register IOMUXC_UART3_UART_RX_DATA_SELECT_INPUT for mode ALT2. 0011 **ALT3** — Select signal ECSPI1_SS2. - Configure register IOMUXC_ECSPI1_SS2_SELECT_INPUT for mode ALT3. 0100 **ALT4** — Select signal ECSPI2_SS2. 0101 **ALT5** — Select signal GPIO3_IO24. 0110 **ALT6** — Select signal AUD5_RXFS. - Configure register IOMUXC_AUD5_INPUT_RXFS_AMX_SELECT_INPUT for mode ALT6. 0111 **ALT7** — Select signal UART1_DTR_B. 1000 **ALT8** — Select signal EPDC_SDCE7.

37.4.86 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA25)

Address: 20E_0000h base + 168h offset = 20E_0168h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

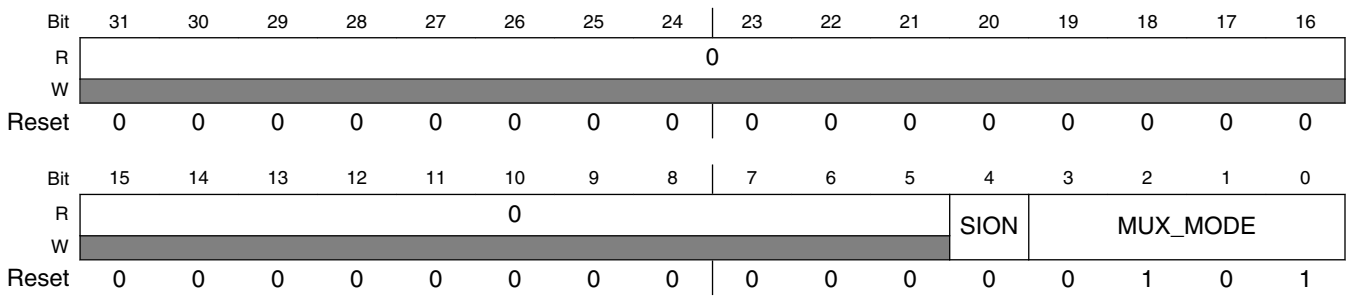
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_EIM_DATA25 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D25. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: EIM_D25. NOTE: Pad EIM_D25 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_DATA25. 0001 **ALT1** — Select signal ECSPI4_SS3. 0010 **ALT2** — Select signal UART3_RX_DATA. - Configure register IOMUXC_UART3_UART_RX_DATA_SELECT_INPUT for mode ALT2. 0011 **ALT3** — Select signal ECSPI1_SS3. - Configure register IOMUXC_ECSPI1_SS3_SELECT_INPUT for mode ALT3. 0100 **ALT4** — Select signal ECSPI2_SS3. 0101 **ALT5** — Select signal GPIO3_IO25. 0110 **ALT6** — Select signal AUD5_RXC. - Configure register IOMUXC_AUD5_INPUT_RXCLK_AMX_SELECT_INPUT for mode ALT6. 0111 **ALT7** — Select signal UART1_DSR_B. 1000 **ALT8** — Select signal EPDC_SDCE8.

37.4.87 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA26)

Address: 20E_0000h base + 16Ch offset = 20E_016Ch



IOMUXC_SW_MUX_CTL_PAD_EIM_DATA26 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D26. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: EIM_D26. NOTE: Pad EIM_D26 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_DATA26. 0001 **ALT1** — Select signal IPU1_DI1_PIN11. 0010 **ALT2** — Select signal IPU1_CSI0_DATA01. 0011 **ALT3** — Select signal IPU1_CSI1_DATA14. - Configure register IOMUXC_IPU1_SENS1_DATA14_SELECT_INPUT for mode ALT3. 0100 **ALT4** — Select signal UART2_TX_DATA. - Configure register IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT for mode ALT4. 0101 **ALT5** — Select signal GPIO3_IO26. 0110 **ALT6** — Select signal IPU1_SISG2. 0111 **ALT7** — Select signal IPU1_DISP1_DATA22. 1000 **ALT8** — Select signal EPDC_SDOED.

37.4.88 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA27)

Address: 20E_0000h base + 170h offset = 20E_0170h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_EIM_DATA27 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D27. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: EIM_D27. NOTE: Pad EIM_D27 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_DATA27. 0001 **ALT1** — Select signal IPU1_DI1_PIN13. 0010 **ALT2** — Select signal IPU1_CSI0_DATA00. 0011 **ALT3** — Select signal IPU1_CSI1_DATA13. - Configure register IOMUXC_IPU1_SENS1_DATA13_SELECT_INPUT for mode ALT3. 0100 **ALT4** — Select signal UART2_RX_DATA. - Configure register IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT for mode ALT4. 0101 **ALT5** — Select signal GPIO3_IO27. 0110 **ALT6** — Select signal IPU1_SISG3. 0111 **ALT7** — Select signal IPU1_DISP1_DATA23. 1000 **ALT8** — Select signal EPDC_SDOE.

37.4.89 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA28)

Address: 20E_0000h base + 174h offset = 20E_0174h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_EIM_DATA28 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D28. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: EIM_D28. NOTE: Pad EIM_D28 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_DATA28. 0001 **ALT1** — Select signal I2C1_SDA. - Configure register IOMUXC_I2C1_SDA_IN_SELECT_INPUT for mode ALT1. 0010 **ALT2** — Select signal ECSPI4_MOSI. 0011 **ALT3** — Select signal IPU1_CSI1_DATA12. - Configure register IOMUXC_IPU1_SENS1_DATA12_SELECT_INPUT for mode ALT3. 0100 **ALT4** — Select signal UART2_CTS_B. - Configure register IOMUXC_UART2_UART_RTS_B_SELECT_INPUT for mode ALT4. 0101 **ALT5** — Select signal GPIO3_IO28. 0110 **ALT6** — Select signal IPU1_EXT_TRIG. 0111 **ALT7** — Select signal IPU1_DI0_PIN13. 1000 **ALT8** — Select signal EPDC_PWR_CTRL3.

37.4.90 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA29)

Address: 20E_0000h base + 178h offset = 20E_0178h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_EIM_DATA29 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D29. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D29. NOTE: Pad EIM_D29 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_DATA29. 0001 **ALT1** — Select signal IPU1_DI1_PIN15. 0010 **ALT2** — Select signal ECSPI4_SS0. - Configure register IOMUXC_ECSPI4_SS0_SELECT_INPUT for mode ALT2. 0100 **ALT4** — Select signal UART2_RTS_B. - Configure register IOMUXC_UART2_UART_RTS_B_SELECT_INPUT for mode ALT4. 0101 **ALT5** — Select signal GPIO3_IO29. 0110 **ALT6** — Select signal IPU1_CSI1_VSYNC. - Configure register IOMUXC_IPU1_SENS1_VSYNC_SELECT_INPUT for mode ALT6. 0111 **ALT7** — Select signal IPU1_DI0_PIN14. 1000 **ALT8** — Select signal EPDC_PWR_WAKE.

37.4.91 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA30)

Address: 20E_0000h base + 17Ch offset = 20E_017Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_EIM_DATA30 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D30. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D30. NOTE: Pad EIM_D30 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_DATA30. 0001 **ALT1** — Select signal IPU1_DISP1_DATA21. 0010 **ALT2** — Select signal IPU1_DI0_PIN11. 0011 **ALT3** — Select signal IPU1_CSI0_DATA03. 0100 **ALT4** — Select signal UART3_CTS_B. - Configure register IOMUXC_UART3_UART_RTS_B_SELECT_INPUT for mode ALT4. 0101 **ALT5** — Select signal GPIO3_IO30. 0110 **ALT6** — Select signal USB_H1_OC. - Configure register IOMUXC_USB_H1_OC_SELECT_INPUT for mode ALT6. 1000 **ALT8** — Select signal EPDC_SDOEZ.

37.4.92 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_DATA31)

Address: 20E_0000h base + 180h offset = 20E_0180h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_EIM_DATA31 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_D31. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: EIM_D31. NOTE: Pad EIM_D31 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_DATA31. 0001 **ALT1** — Select signal IPU1_DISP1_DATA20. 0010 **ALT2** — Select signal IPU1_DI0_PIN12. 0011 **ALT3** — Select signal IPU1_CSI0_DATA02. 0100 **ALT4** — Select signal UART3_RTS_B. - Configure register IOMUXC_UART3_UART_RTS_B_SELECT_INPUT for mode ALT4. 0101 **ALT5** — Select signal GPIO3_IO31. 0110 **ALT6** — Select signal USB_H1_PWR. 1000 **ALT8** — Select signal EPDC_SDCLK_P. 1001 **ALT9** — Select signal EIM_ACLK_FREERUN.

37.4.93 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD00)

Address: 20E_0000h base + 184h offset = 20E_0184h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_AD00 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_DA0. 0000 **ALT0** — Select signal EIM_AD00. 0001 **ALT1** — Select signal IPU1_DISP1_DATA09. 0010 **ALT2** — Select signal IPU1_CSI1_DATA09. 0101 **ALT5** — Select signal GPIO3_IO00. 0111 **ALT7** — Select signal SRC_BOOT_CFG00. 1000 **ALT8** — Select signal EPDC_SDCLK_N.

37.4.94 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD01)

Address: 20E_0000h base + 188h offset = 20E_0188h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

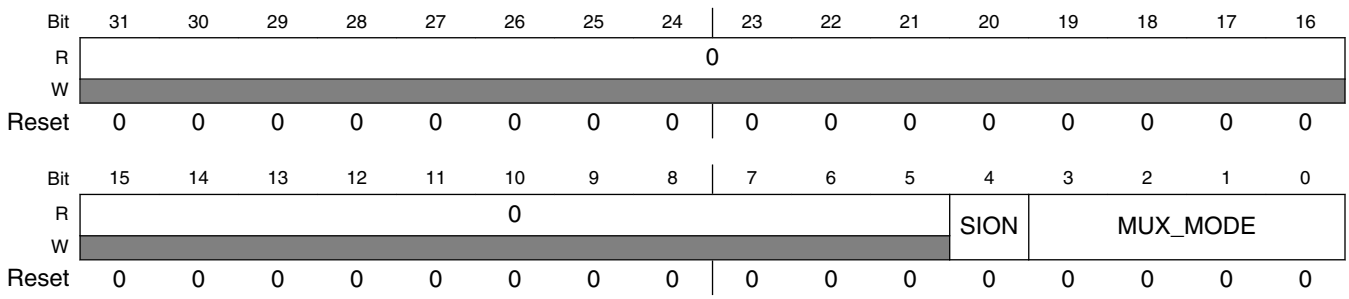
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_AD01 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_DA1. 0000 **ALT0** — Select signal EIM_AD01. 0001 **ALT1** — Select signal IPU1_DISP1_DATA08. 0010 **ALT2** — Select signal IPU1_CSI1_DATA08. 0101 **ALT5** — Select signal GPIO3_IO01. 0111 **ALT7** — Select signal SRC_BOOT_CFG01. 1000 **ALT8** — Select signal EPDC_SDLE.

37.4.95 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD10)

Address: 20E_0000h base + 18Ch offset = 20E_018Ch

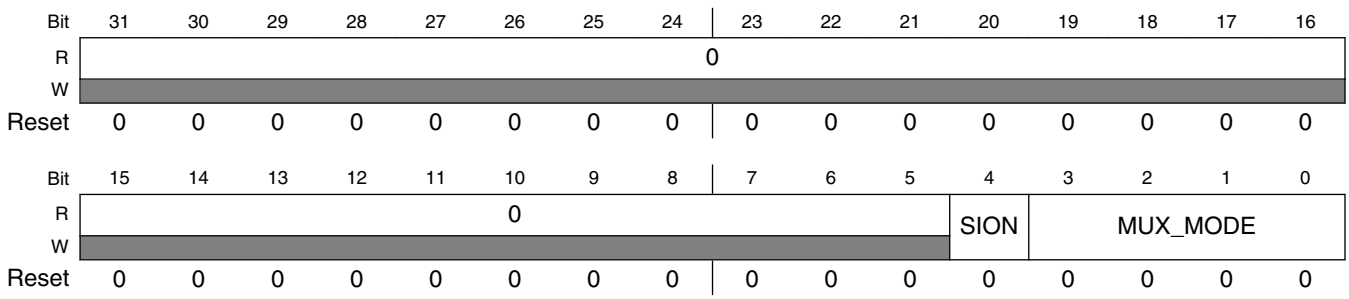


IOMUXC_SW_MUX_CTL_PAD_EIM_AD10 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA10. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_DA10. NOTE: Pad EIM_DA10 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_AD10. 0001 **ALT1** — Select signal IPU1_DI1_PIN15. 0010 **ALT2** — Select signal IPU1_CSI1_DATA_EN. - Configure register IOMUXC_IPU1_SENS1_DATA_EN_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO3_IO10. 0111 **ALT7** — Select signal SRC_BOOT_CFG10. 1000 **ALT8** — Select signal EPDC_DATA01.

37.4.96 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD11)

Address: 20E_0000h base + 190h offset = 20E_0190h

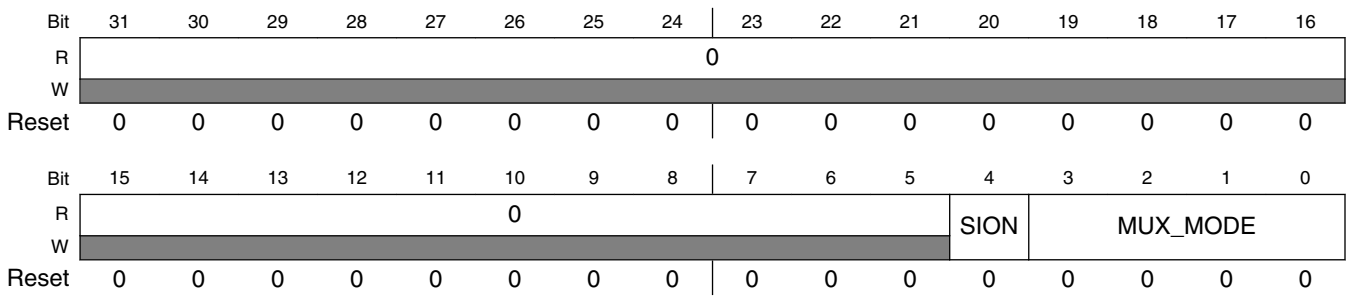


IOMUXC_SW_MUX_CTL_PAD_EIM_AD11 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA11. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_DA11. NOTE: Pad EIM_DA11 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_AD11. 0001 **ALT1** — Select signal IPU1_DI1_PIN02. 0010 **ALT2** — Select signal IPU1_CSI1_HSYNC. - Configure register IOMUXC_IPU1_SENS1_HSYNC_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO3_IO11. 0111 **ALT7** — Select signal SRC_BOOT_CFG11. 1000 **ALT8** — Select signal EPDC_DATA03.

37.4.97 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD12)

Address: 20E_0000h base + 194h offset = 20E_0194h



IOMUXC_SW_MUX_CTL_PAD_EIM_AD12 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA12. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_DA12. NOTE: Pad EIM_DA12 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_AD12. 0001 **ALT1** — Select signal IPU1_DI1_PIN03. 0010 **ALT2** — Select signal IPU1_CSI1_VSYNC. - Configure register IOMUXC_IPU1_SENS1_VSYNC_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO3_IO12. 0111 **ALT7** — Select signal SRC_BOOT_CFG12. 1000 **ALT8** — Select signal EPDC_DATA02.

37.4.98 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD13)

Address: 20E_0000h base + 198h offset = 20E_0198h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

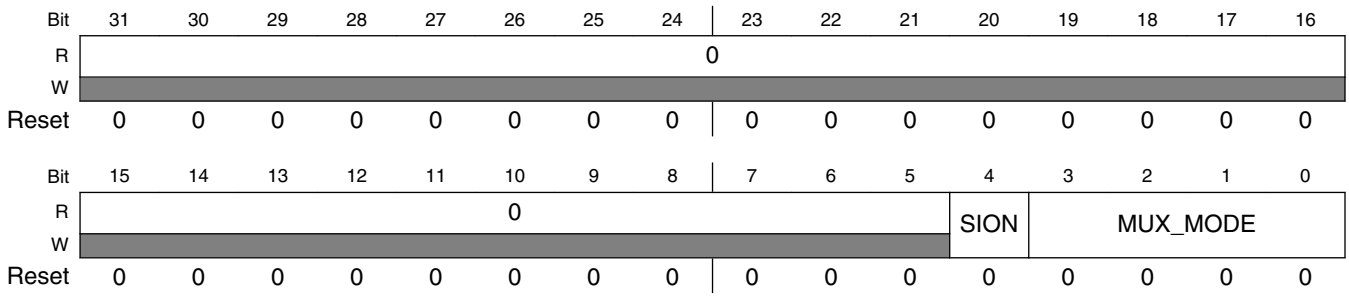
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_AD13 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA13. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_DA13. 0000 **ALT0** — Select signal EIM_AD13. 0001 **ALT1** — Select signal IPU1_DI1_D0_CS. 0101 **ALT5** — Select signal GPIO3_IO13. 0111 **ALT7** — Select signal SRC_BOOT_CFG13. 1000 **ALT8** — Select signal EPDC_DATA13.

37.4.99 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD14)

Address: 20E_0000h base + 19Ch offset = 20E_019Ch



IOMUXC_SW_MUX_CTL_PAD_EIM_AD14 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA14. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_DA14. 0000 **ALT0** — Select signal EIM_AD14. 0001 **ALT1** — Select signal IPU1_DI1_D1_CS. 0101 **ALT5** — Select signal GPIO3_IO14. 0111 **ALT7** — Select signal SRC_BOOT_CFG14. 1000 **ALT8** — Select signal EPDC_DATA14.

37.4.100 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD15)

Address: 20E_0000h base + 1A0h offset = 20E_01A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_AD15 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA15. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_DA15. 0000 **ALT0** — Select signal EIM_AD15. 0001 **ALT1** — Select signal IPU1_DI1_PIN01. 0010 **ALT2** — Select signal IPU1_DI1_PIN04. 0101 **ALT5** — Select signal GPIO3_IO15. 0111 **ALT7** — Select signal SRC_BOOT_CFG15. 1000 **ALT8** — Select signal EPDC_DATA09.

37.4.101 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD02)

Address: 20E_0000h base + 1A4h offset = 20E_01A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_AD02 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA2. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_DA2. 0000 **ALT0** — Select signal EIM_AD02. 0001 **ALT1** — Select signal IPU1_DISP1_DATA07. 0010 **ALT2** — Select signal IPU1_CSI1_DATA07. 0101 **ALT5** — Select signal GPIO3_IO02. 0111 **ALT7** — Select signal SRC_BOOT_CFG02. 1000 **ALT8** — Select signal EPDC_BDR0.

37.4.102 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD03)

Address: 20E_0000h base + 1A8h offset = 20E_01A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_AD03 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA3. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_DA3. 0000 **ALT0** — Select signal EIM_AD03. 0001 **ALT1** — Select signal IPU1_DISP1_DATA06. 0010 **ALT2** — Select signal IPU1_CSI1_DATA06. 0101 **ALT5** — Select signal GPIO3_IO03. 0111 **ALT7** — Select signal SRC_BOOT_CFG03. 1000 **ALT8** — Select signal EPDC_BDR1.

37.4.103 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD04)

Address: 20E_0000h base + 1ACh offset = 20E_01ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_AD04 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA4. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_DA4. 0000 **ALT0** — Select signal EIM_AD04. 0001 **ALT1** — Select signal IPU1_DISP1_DATA05. 0010 **ALT2** — Select signal IPU1_CSI1_DATA05. 0101 **ALT5** — Select signal GPIO3_IO04. 0111 **ALT7** — Select signal SRC_BOOT_CFG04. 1000 **ALT8** — Select signal EPDC_SDCE0.

37.4.104 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD05)

Address: 20E_0000h base + 1B0h offset = 20E_01B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_AD05 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA5. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_DA5. 0000 **ALT0** — Select signal EIM_AD05. 0001 **ALT1** — Select signal IPU1_DISP1_DATA04. 0010 **ALT2** — Select signal IPU1_CSI1_DATA04. 0101 **ALT5** — Select signal GPIO3_IO05. 0111 **ALT7** — Select signal SRC_BOOT_CFG05. 1000 **ALT8** — Select signal EPDC_SDCE1.

37.4.105 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD06)

Address: 20E_0000h base + 1B4h offset = 20E_01B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_AD06 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA6. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_DA6. 0000 **ALT0** — Select signal EIM_AD06. 0001 **ALT1** — Select signal IPU1_DISP1_DATA03. 0010 **ALT2** — Select signal IPU1_CSI1_DATA03. 0101 **ALT5** — Select signal GPIO3_IO06. 0111 **ALT7** — Select signal SRC_BOOT_CFG06. 1000 **ALT8** — Select signal EPDC_SDCE2.

37.4.106 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD07)

Address: 20E_0000h base + 1B8h offset = 20E_01B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_AD07 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA7. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_DA7. 0000 **ALT0** — Select signal EIM_AD07. 0001 **ALT1** — Select signal IPU1_DISP1_DATA02. 0010 **ALT2** — Select signal IPU1_CSI1_DATA02. 0101 **ALT5** — Select signal GPIO3_IO07. 0111 **ALT7** — Select signal SRC_BOOT_CFG07. 1000 **ALT8** — Select signal EPDC_SDCE3.

37.4.107 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD08)

Address: 20E_0000h base + 1BCh offset = 20E_01BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_AD08 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA8. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_DA8. 0000 **ALT0** — Select signal EIM_AD08. 0001 **ALT1** — Select signal IPU1_DISP1_DATA01. 0010 **ALT2** — Select signal IPU1_CSI1_DATA01. 0101 **ALT5** — Select signal GPIO3_IO08. 0111 **ALT7** — Select signal SRC_BOOT_CFG08. 1000 **ALT8** — Select signal EPDC_SDCE4.

37.4.108 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_AD09)

Address: 20E_0000h base + 1C0h offset = 20E_01C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

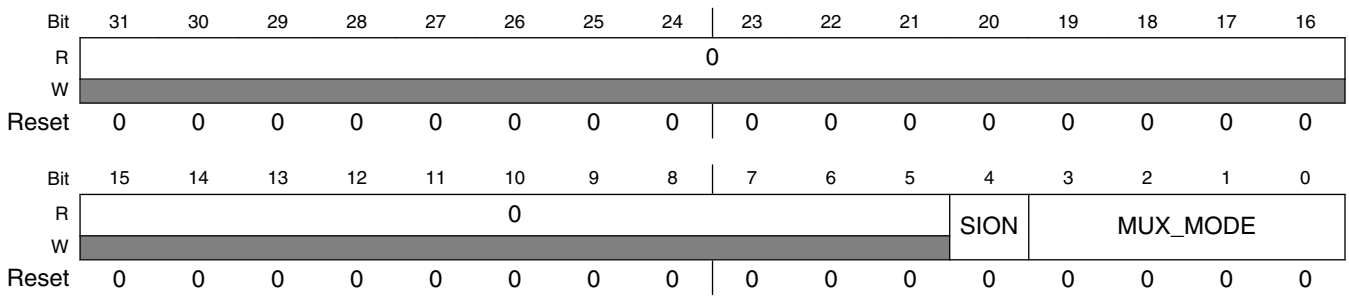
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_AD09 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_DA9. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_DA9. 0000 **ALT0** — Select signal EIM_AD09. 0001 **ALT1** — Select signal IPU1_DISP1_DATA00. 0010 **ALT2** — Select signal IPU1_CSI1_DATA00. 0101 **ALT5** — Select signal GPIO3_IO09. 0111 **ALT7** — Select signal SRC_BOOT_CFG09. 1000 **ALT8** — Select signal EPDC_SDCE5.

37.4.109 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_EB0_B)

Address: 20E_0000h base + 1C4h offset = 20E_01C4h



IOMUXC_SW_MUX_CTL_PAD_EIM_EB0_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_EB0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: EIM_EB0. NOTE: Pad EIM_EB0 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_EB0_B. 0001 **ALT1** — Select signal IPU1_DISP1_DATA11. 0010 **ALT2** — Select signal IPU1_CSI1_DATA11. - Configure register IOMUXC_IPU1_SENS1_DATA11_SELECT_INPUT for mode ALT2. 0100 **ALT4** — Select signal CCM_PMIC_READY. - Configure register IOMUXC_CCM_PMIC_READY_SELECT_INPUT for mode ALT4. 0101 **ALT5** — Select signal GPIO2_IO28. 0111 **ALT7** — Select signal SRC_BOOT_CFG27. 1000 **ALT8** — Select signal EPDC_PWR_COM.

37.4.110 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_EB1_B)

Address: 20E_0000h base + 1C8h offset = 20E_01C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

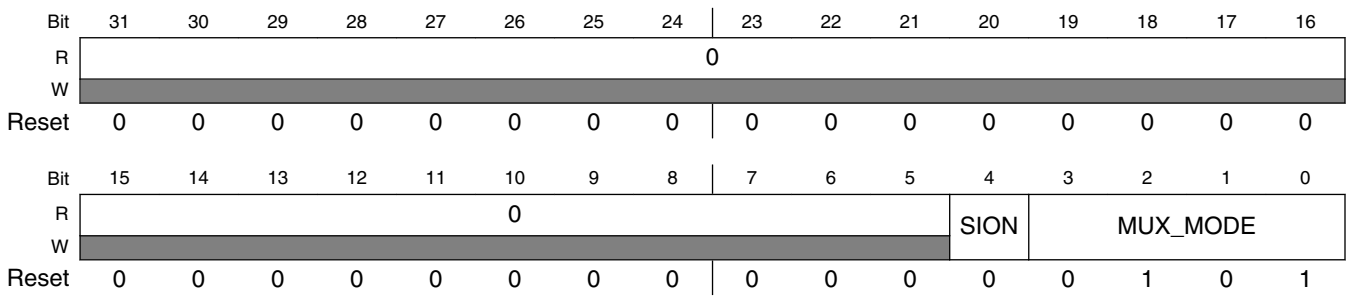
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_EB1_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_EB1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_EB1. NOTE: Pad EIM_EB1 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_EB1_B. 0001 **ALT1** — Select signal IPU1_DISP1_DATA10. 0010 **ALT2** — Select signal IPU1_CSI1_DATA10. - Configure register IOMUXC_IPU1_SENS1_DATA10_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO2_IO29. 0111 **ALT7** — Select signal SRC_BOOT_CFG28. 1000 **ALT8** — Select signal EPDC_SDSHR.

37.4.111 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_EB2_B)

Address: 20E_0000h base + 1CCh offset = 20E_01CCh

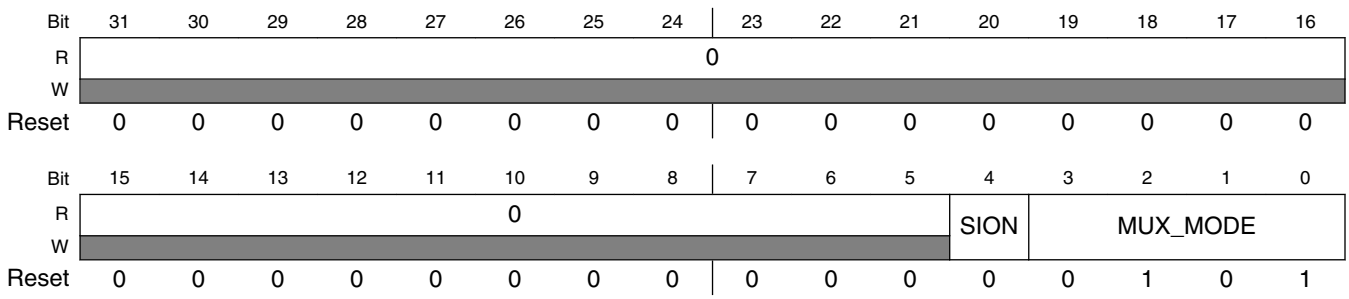


IOMUXC_SW_MUX_CTL_PAD_EIM_EB2_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_EB2. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_EB2. NOTE: Pad EIM_EB2 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_EB2_B. 0001 **ALT1** — Select signal ECSPI1_SS0. - Configure register IOMUXC_ECSPI1_SS0_SELECT_INPUT for mode ALT1. 0011 **ALT3** — Select signal IPU1_CSI1_DATA19. - Configure register IOMUXC_IPU1_SENS1_DATA19_SELECT_INPUT for mode ALT3. 0100 **ALT4** — Select signal HDMI_TX_DDC_SCL. - Configure register IOMUXC_HDMI_II2C_CLKIN_SELECT_INPUT for mode ALT4. 0101 **ALT5** — Select signal GPIO2_IO30. 0110 **ALT6** — Select signal I2C2_SCL. - Configure register IOMUXC_I2C2_SCL_IN_SELECT_INPUT for mode ALT6. 0111 **ALT7** — Select signal SRC_BOOT_CFG30. 1000 **ALT8** — Select signal EPDC_DATA05.

37.4.112 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_EB3_B)

Address: 20E_0000h base + 1D0h offset = 20E_01D0h

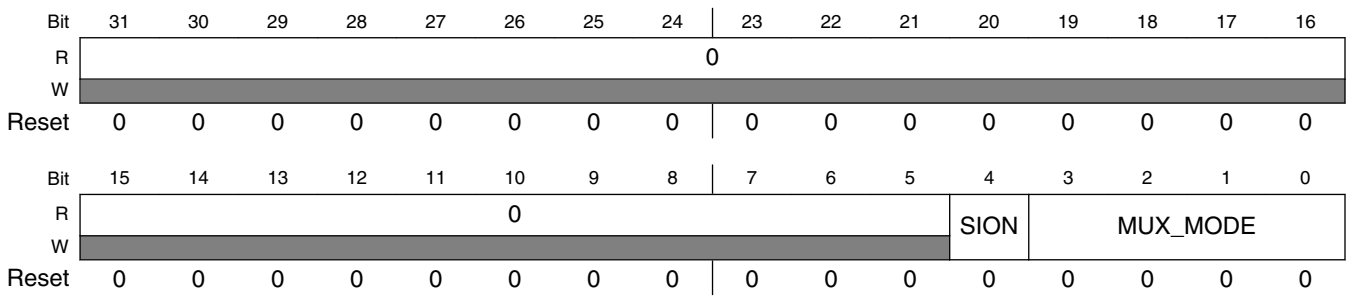


IOMUXC_SW_MUX_CTL_PAD_EIM_EB3_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_EB3. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: EIM_EB3. NOTE: Pad EIM_EB3 is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_EB3_B. 0001 **ALT1** — Select signal ECSPI4_RDY. 0010 **ALT2** — Select signal UART3_RTS_B. - Configure register IOMUXC_UART3_UART_RTS_B_SELECT_INPUT for mode ALT2. 0011 **ALT3** — Select signal UART1_RI_B. 0100 **ALT4** — Select signal IPU1_CSI1_HSYNC. - Configure register IOMUXC_IPU1_SENS1_HSYNC_SELECT_INPUT for mode ALT4. 0101 **ALT5** — Select signal GPIO2_IO31. 0110 **ALT6** — Select signal IPU1_DI1_PIN03. 0111 **ALT7** — Select signal SRC_BOOT_CFG31. 1000 **ALT8** — Select signal EPDC_SDCE0. 1001 **ALT9** — Select signal EIM_ACLK_FREERUN.

37.4.113 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_LBA_B)

Address: 20E_0000h base + 1D4h offset = 20E_01D4h



IOMUXC_SW_MUX_CTL_PAD_EIM_LBA_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_LBA. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_LBA. NOTE: Pad EIM_LBA is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_LBA_B. 0001 **ALT1** — Select signal IPU1_DI1_PIN17. 0010 **ALT2** — Select signal ECSPI2_SS1. - Configure register IOMUXC_ECSPI2_SS1_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO2_IO27. 0111 **ALT7** — Select signal SRC_BOOT_CFG26. 1000 **ALT8** — Select signal EPDC_DATA04.

37.4.114 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_OE_B)

Address: 20E_0000h base + 1D8h offset = 20E_01D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

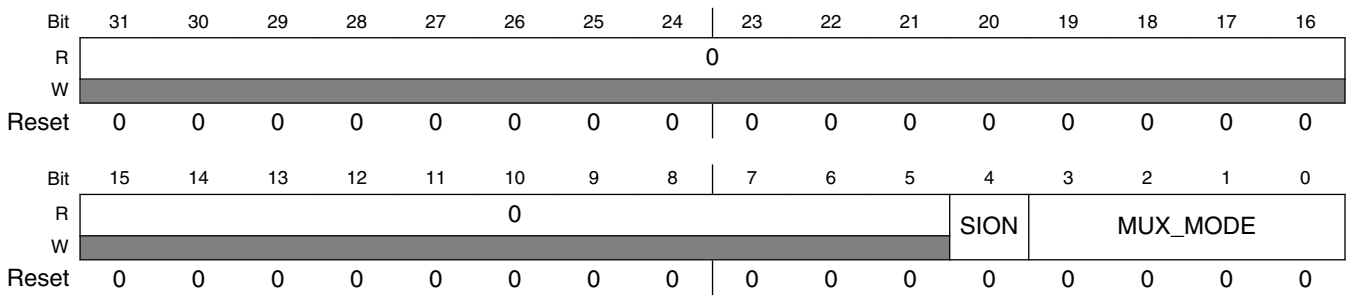
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_OE_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_OE. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_OE. NOTE: Pad EIM_OE is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_OE_B. 0001 **ALT1** — Select signal IPU1_DI1_PIN07. 0010 **ALT2** — Select signal ECSPI2_MISO. - Configure register IOMUXC_ECSPI2_MISO_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO2_IO25. 1000 **ALT8** — Select signal EPDC_PWR_IRQ.

37.4.115 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_RW)

Address: 20E_0000h base + 1DCh offset = 20E_01DCh



IOMUXC_SW_MUX_CTL_PAD_EIM_RW field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_RW. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_RW. NOTE: Pad EIM_RW is involved in Daisy Chain. 0000 **ALT0** — Select signal EIM_RW. 0001 **ALT1** — Select signal IPU1_DI1_PIN08. 0010 **ALT2** — Select signal ECSPI2_SS0. - Configure register IOMUXC_ECSPI2_SS0_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO2_IO26. 0111 **ALT7** — Select signal SRC_BOOT_CFG29. 1000 **ALT8** — Select signal EPDC_DATA07.

37.4.116 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_EIM_WAIT_B)

Address: 20E_0000h base + 1E0h offset = 20E_01E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

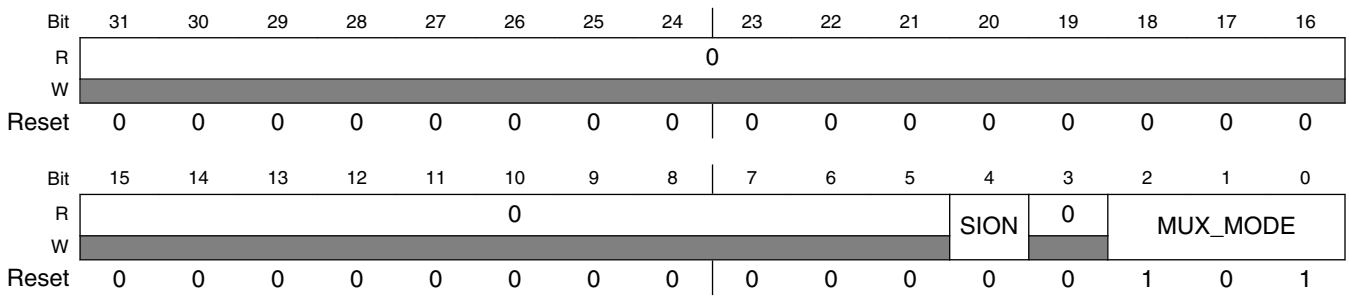
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_EIM_WAIT_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad EIM_WAIT. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: EIM_WAIT. 000 **ALT0** — Select signal EIM_WAIT_B. 001 **ALT1** — Select signal EIM_DTACK_B. 101 **ALT5** — Select signal GPIO5_IO00. 111 **ALT7** — Select signal SRC_BOOT_CFG25.

37.4.117 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_CRS_DV)

Address: 20E_0000h base + 1E4h offset = 20E_01E4h



IOMUXC_SW_MUX_CTL_PAD_ENET_CRS_DV field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_CRS_DV. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: ENET_CRS_DV. NOTE: Pad ENET_CRS_DV is involved in Daisy Chain. 001 **ALT1** — Select signal ENET_RX_EN. - Configure register IOMUXC_ENET_MAC0_RX_EN_SELECT_INPUT for mode ALT1. 010 **ALT2** — Select signal ESAI_TX_CLK. - Configure register IOMUXC_ESAI_TX_CLK_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal SPDIF_EXT_CLK. - Configure register IOMUXC_SPDIF_TX_CLK2_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO1_IO25.

37.4.118 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_MDC)

Address: 20E_0000h base + 1E8h offset = 20E_01E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

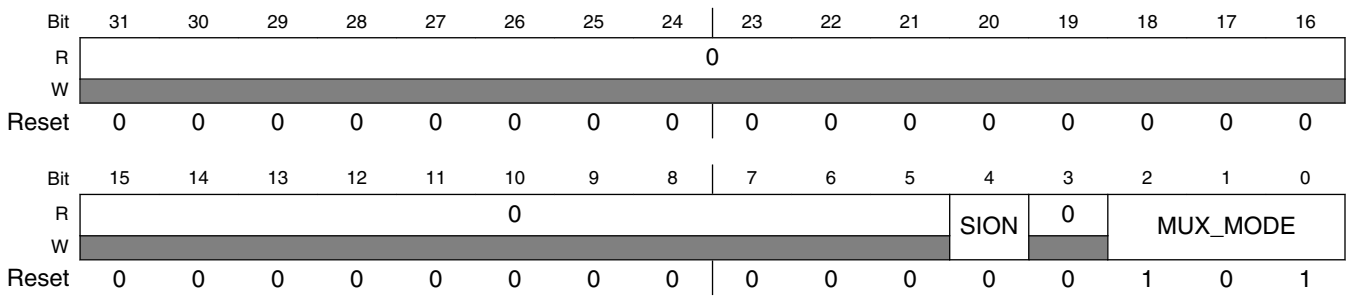
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_ENET_MDC field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_MDC. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET_MDC. NOTE: Pad ENET_MDC is involved in Daisy Chain. 000 **ALT0** — Select signal MLB_DATA. - Configure register IOMUXC_MLB_MLB_DATA_IN_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal ENET_MDC. 010 **ALT2** — Select signal ESAI_TX5_RX0. - Configure register IOMUXC_ESAI_SDO5_SDI0_SELECT_INPUT for mode ALT2. 100 **ALT4** — Select signal ENET_1588_EVENT1_IN. 101 **ALT5** — Select signal GPIO1_IO31.

37.4.119 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO)

Address: 20E_0000h base + 1ECh offset = 20E_01ECh



IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_MDIO. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET_MDIO. NOTE: Pad ENET_MDIO is involved in Daisy Chain. 001 **ALT1** — Select signal ENET_MDIO. - Configure register IOMUXC_ENET_MAC0_MDIO_SELECT_INPUT for mode ALT1. 010 **ALT2** — Select signal ESAI_RX_CLK. - Configure register IOMUXC_ESAI_RX_CLK_SELECT_INPUT for mode ALT2. 100 **ALT4** — Select signal ENET_1588_EVENT1_OUT. 101 **ALT5** — Select signal GPIO1_IO22. 110 **ALT6** — Select signal SPDIF_LOCK.

37.4.120 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_REF_CLK)

Address: 20E_0000h base + 1F0h offset = 20E_01F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_ENET_REF_CLK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_REF_CLK. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: ENET_REF_CLK. NOTE: Pad ENET_REF_CLK is involved in Daisy Chain. 001 **ALT1** — Select signal ENET_TX_CLK. In MII mode, provides a timing reference for TX_EN, TX_DATA[3:0] and TX_ER. In RGMII mode, provides 125 MHz reference clock input. 010 **ALT2** — Select signal ESAI_RX_FS. - Configure register IOMUXC_ESAI_RX_FS_SELECT_INPUT for mode ALT2. 101 **ALT5** — Select signal GPIO1_IO23. 110 **ALT6** — Select signal SPDIF_SR_CLK.

37.4.121 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RX_ER)

Address: 20E_0000h base + 1F4h offset = 20E_01F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_ENET_RX_ER field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_RX_ER. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET_RX_ER. NOTE: Pad ENET_RX_ER is involved in Daisy Chain. 000 **ALT0** — Select signal USB_OTG_ID. - Configure register IOMUXC_ANALOG_USB_OTG_ID_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal ENET_RX_ER. 010 **ALT2** — Select signal ESAI_RX_HF_CLK. - Configure register IOMUXC_ESAI_RX_HF_CLK_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal SPDIF_IN. - Configure register IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal ENET_1588_EVENT2_OUT. 101 **ALT5** — Select signal GPIO1_IO24.

37.4.122 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA0)

Address: 20E_0000h base + 1F8h offset = 20E_01F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_RXD0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: ENET_RXD0. NOTE: Pad ENET_RXD0 is involved in Daisy Chain. 001 **ALT1** — Select signal ENET_RX_DATA0. - Configure register IOMUXC_ENET_MAC0_RX_DATA0_SELECT_INPUT for mode ALT1. 010 **ALT2** — Select signal ESAI_TX_HF_CLK. - Configure register IOMUXC_ESAI_TX_HF_CLK_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal SPDIF_OUT. 101 **ALT5** — Select signal GPIO1_IO27.

37.4.123 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA1)

Address: 20E_0000h base + 1FCh offset = 20E_01FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

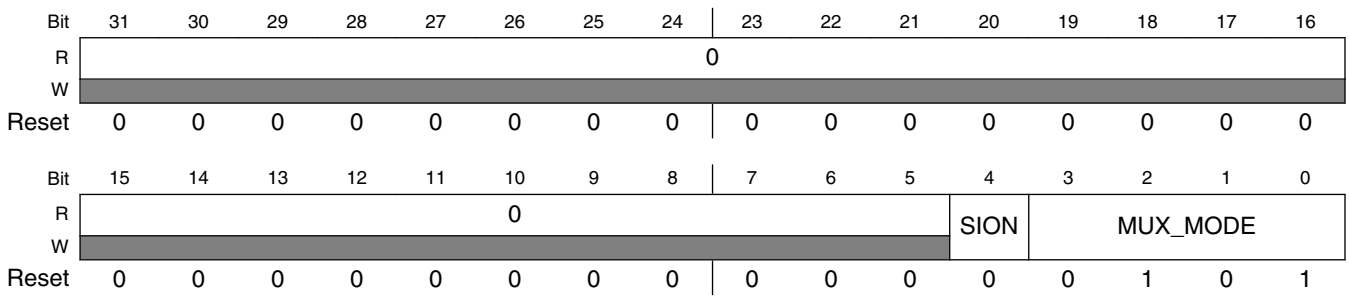
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_ENET_RX_DATA1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_RXD1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET_RXD1. NOTE: Pad ENET_RXD1 is involved in Daisy Chain. 000 **ALT0** — Select signal MLB_SIG. - Configure register IOMUXC_MLB_MLB_SIG_IN_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal ENET_RX_DATA1. - Configure register IOMUXC_ENET_MAC0_RX_DATA1_SELECT_INPUT for mode ALT1. 010 **ALT2** — Select signal ESAI_TX_FS. - Configure register IOMUXC_ESAI_TX_FS_SELECT_INPUT for mode ALT2. 100 **ALT4** — Select signal ENET_1588_EVENT3_OUT. 101 **ALT5** — Select signal GPIO1_IO26.

37.4.124 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TX_EN)

Address: 20E_0000h base + 200h offset = 20E_0200h

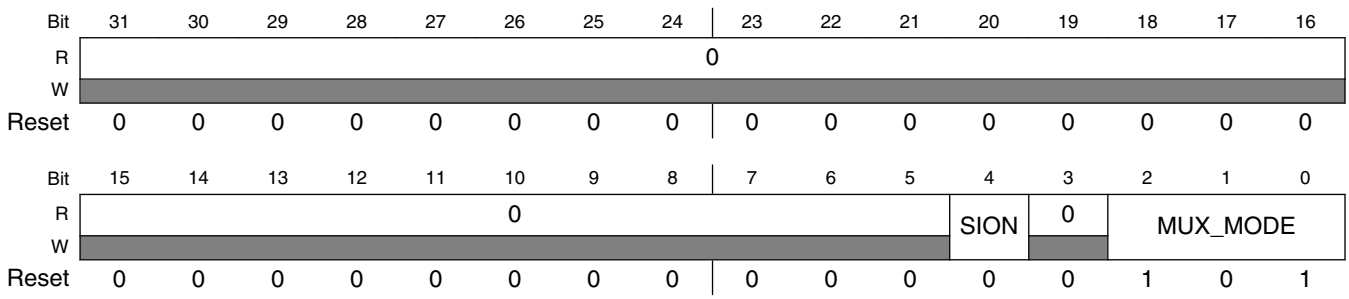


IOMUXC_SW_MUX_CTL_PAD_ENET_TX_EN field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_TX_EN. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: ENET_TX_EN. NOTE: Pad ENET_TX_EN is involved in Daisy Chain. 0001 **ALT1** — Select signal ENET_TX_EN. 0010 **ALT2** — Select signal ESAI_TX3_RX2. - Configure register IOMUXC_ESAI_SDO3_SDI2_SELECT_INPUT for mode ALT2. 0101 **ALT5** — Select signal GPIO1_IO28. 1001 **ALT9** — Select signal I2C4_SCL. - Configure register IOMUXC_I2C4_SCL_IN_SELECT_INPUT for mode ALT9.

37.4.125 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA0)

Address: 20E_0000h base + 204h offset = 20E_0204h



IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_TXD0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: ENET_TXD0. NOTE: Pad ENET_TXD0 is involved in Daisy Chain. 001 **ALT1** — Select signal ENET_TX_DATA0. 010 **ALT2** — Select signal ESAI_TX4_RX1. - Configure register IOMUXC_ESAI_SDO4_SDI1_SELECT_INPUT for mode ALT2. 101 **ALT5** — Select signal GPIO1_IO30.

37.4.126 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA1)

Address: 20E_0000h base + 208h offset = 20E_0208h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

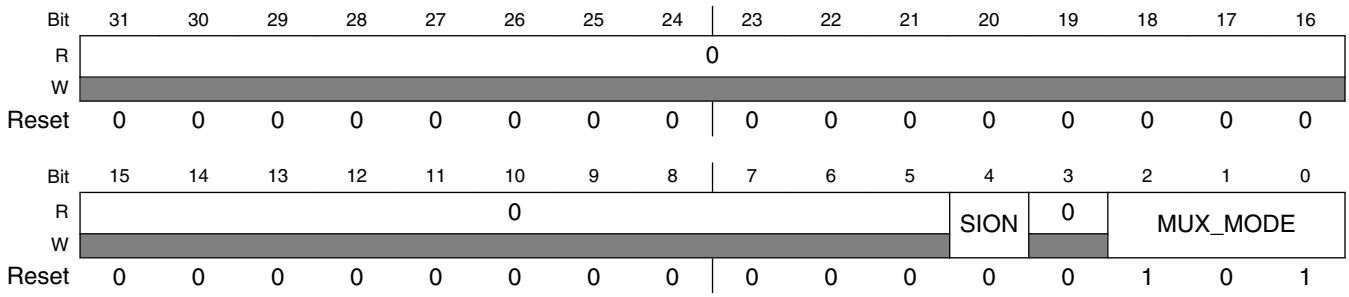
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_ENET_TX_DATA1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_TXD1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET_TXD1. NOTE: Pad ENET_TXD1 is involved in Daisy Chain. 0000 **ALT0** — Select signal MLB_CLK. - Configure register IOMUXC_MLB_MLB_CLK_IN_SELECT_INPUT for mode ALT0. 0001 **ALT1** — Select signal ENET_TX_DATA1. 0010 **ALT2** — Select signal ESAI_TX2_RX3. - Configure register IOMUXC_ESAI_SDO2_SDI3_SELECT_INPUT for mode ALT2. 0100 **ALT4** — Select signal ENET_1588_EVENT0_IN. 0101 **ALT5** — Select signal GPIO1_IO29. 1001 **ALT9** — Select signal I2C4_SDA. - Configure register IOMUXC_I2C4_SDA_IN_SELECT_INPUT for mode ALT9.

37.4.127 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO00)

Address: 20E_0000h base + 20Ch offset = 20E_020Ch



IOMUXC_SW_MUX_CTL_PAD_GPIO00 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO_0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_0. NOTE: Pad GPIO_0 is involved in Daisy Chain. 000 **ALT0** — Select signal CCM_CLKO1. 010 **ALT2** — Select signal KEY_COL5. - Configure register IOMUXC_KEY_COL5_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal ASRC_EXT_CLK. - Configure register IOMUXC_ASRC_ASRCK_CLOCK_6_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal EPIT1_OUT. 101 **ALT5** — Select signal GPIO1_IO00. 110 **ALT6** — Select signal USB_H1_PWR. 111 **ALT7** — Select signal SNVS_VIO_5.

37.4.128 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO01)

Address: 20E_0000h base + 210h offset = 20E_0210h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

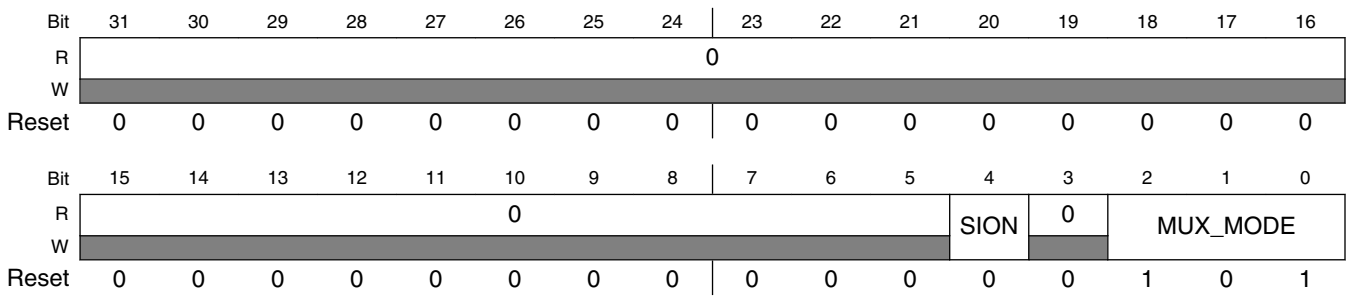
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_GPIO01 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO_1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_1. NOTE: Pad GPIO_1 is involved in Daisy Chain. 000 **ALT0** — Select signal ESAI_RX_CLK. - Configure register IOMUXC_ESAI_RX_CLK_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal WDOG2_B. 010 **ALT2** — Select signal KEY_ROW5. - Configure register IOMUXC_KEY_ROW5_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal USB_OTG_ID. - Configure register IOMUXC_ANALOG_USB_OTG_ID_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal PWM2_OUT. 101 **ALT5** — Select signal GPIO1_IO01. 110 **ALT6** — Select signal SD1_CD_B.

37.4.129 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO16)

Address: 20E_0000h base + 214h offset = 20E_0214h



IOMUXC_SW_MUX_CTL_PAD_GPIO16 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO_16. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_16. NOTE: Pad GPIO_16 is involved in Daisy Chain. 000 **ALT0** — Select signal ESAI_TX3_RX2. - Configure register IOMUXC_ESAI_SDO3_SDI2_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal ENET_1588_EVENT2_IN. 010 **ALT2** — Select signal ENET_REF_CLK. - Configure register IOMUXC_ENET_REF_CLK_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal SD1_LCTL. 100 **ALT4** — Select signal SPDIF_IN. - Configure register IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO7_IO11. 110 **ALT6** — Select signal I2C3_SDA. - Configure register IOMUXC_I2C3_SDA_IN_SELECT_INPUT for mode ALT6. 111 **ALT7** — Select signal JTAG_DE_B.

37.4.130 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO17)

Address: 20E_0000h base + 218h offset = 20E_0218h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

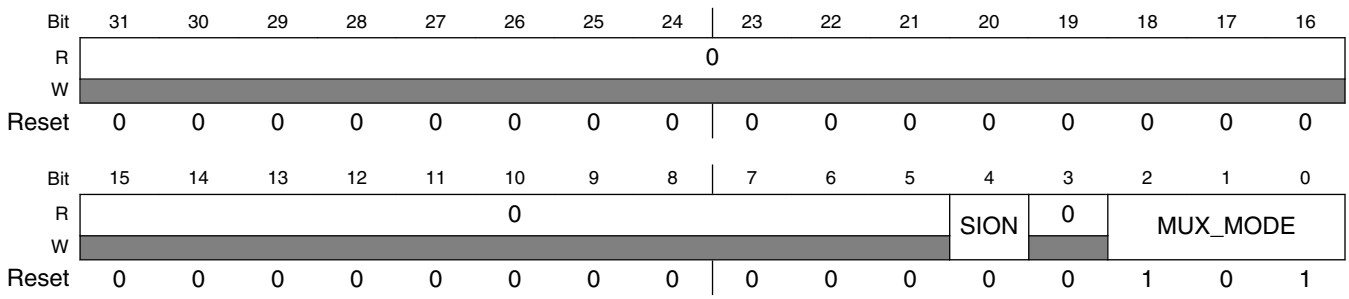
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_GPIO17 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO_17. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_17. NOTE: Pad GPIO_17 is involved in Daisy Chain. 000 **ALT0** — Select signal ESAI_TX0. - Configure register IOMUXC_ESAI_SDO0_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal ENET_1588_EVENT3_IN. 010 **ALT2** — Select signal CCM_PMIC_READY. - Configure register IOMUXC_CCM_PMIC_READY_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal SDMA_EXT_EVENT0. - Configure register IOMUXC_SDMA_EVENTS14_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal SPDIF_OUT. 101 **ALT5** — Select signal GPIO7_IO12.

37.4.131 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO18)

Address: 20E_0000h base + 21Ch offset = 20E_021Ch



IOMUXC_SW_MUX_CTL_PAD_GPIO18 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO_18. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_18. NOTE: Pad GPIO_18 is involved in Daisy Chain. 000 **ALT0** — Select signal ESAI_TX1. - Configure register IOMUXC_ESAI_SDO1_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal ENET_RX_CLK. - Configure register IOMUXC_ENET_MAC0_RX_CLK_SELECT_INPUT for mode ALT1. 010 **ALT2** — Select signal SD3_VSELECT. 011 **ALT3** — Select signal SDMA_EXT_EVENT1. - Configure register IOMUXC_SDMA_EVENTS15_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal ASRC_EXT_CLK. - Configure register IOMUXC_ASRC_ASRCK_CLOCK_6_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO7_IO13. 110 **ALT6** — Select signal SNVS_VIO_5_CTL.

37.4.132 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO19)

Address: 20E_0000h base + 220h offset = 20E_0220h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

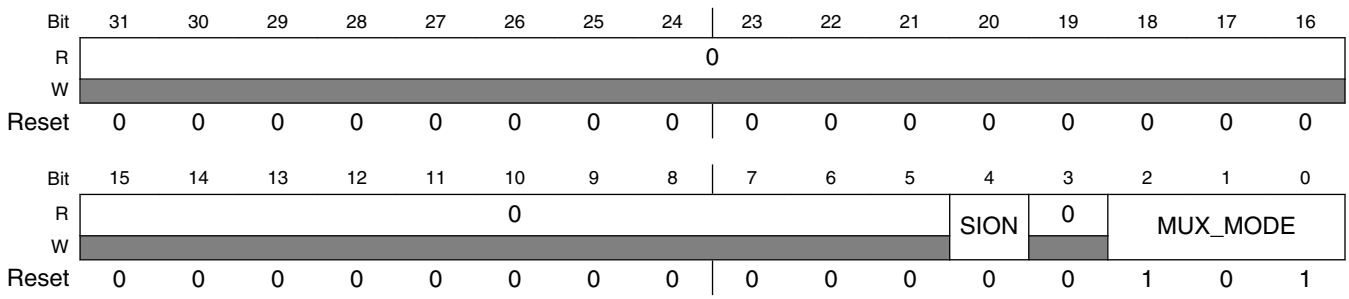
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_GPIO19 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO_19. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_19. NOTE: Pad GPIO_19 is involved in Daisy Chain. 000 **ALT0** — Select signal KEY_COL5. - Configure register IOMUXC_KEY_COL5_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal ENET_1588_EVENT0_OUT. 010 **ALT2** — Select signal SPDIF_OUT. 011 **ALT3** — Select signal CCM_CLKO1. 100 **ALT4** — Select signal ECSPI1_RDY. 101 **ALT5** — Select signal GPIO4_IO05. 110 **ALT6** — Select signal ENET_TX_ER.

37.4.133 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO02)

Address: 20E_0000h base + 224h offset = 20E_0224h



IOMUXC_SW_MUX_CTL_PAD_GPIO02 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO_2. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO_2. NOTE: Pad GPIO_2 is involved in Daisy Chain. 000 **ALT0** — Select signal ESAI_TX_FS. - Configure register IOMUXC_ESAI_TX_FS_SELECT_INPUT for mode ALT0. 010 **ALT2** — Select signal KEY_ROW6. - Configure register IOMUXC_KEY_ROW6_SELECT_INPUT for mode ALT2. 101 **ALT5** — Select signal GPIO1_IO02. 110 **ALT6** — Select signal SD2_WP. 111 **ALT7** — Select signal MLB_DATA. - Configure register IOMUXC_MLB_MLB_DATA_IN_SELECT_INPUT for mode ALT7.

37.4.134 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO03)

Address: 20E_0000h base + 228h offset = 20E_0228h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

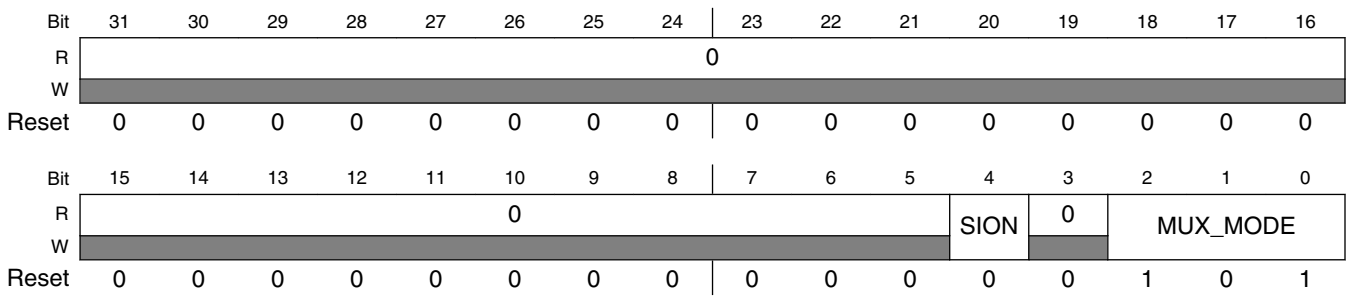
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_GPIO03 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO_3. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_3. NOTE: Pad GPIO_3 is involved in Daisy Chain. 000 **ALT0** — Select signal ESAI_RX_HF_CLK. - Configure register IOMUXC_ESAI_RX_HF_CLK_SELECT_INPUT for mode ALT0. 010 **ALT2** — Select signal I2C3_SCL. - Configure register IOMUXC_I2C3_SCL_IN_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal XTALOSC_REF_CLK_24M. 100 **ALT4** — Select signal CCM_CLKO2. 101 **ALT5** — Select signal GPIO1_IO03. 110 **ALT6** — Select signal USB_H1_OC. - Configure register IOMUXC_USB_H1_OC_SELECT_INPUT for mode ALT6. 111 **ALT7** — Select signal MLB_CLK. - Configure register IOMUXC_MLB_MLB_CLK_IN_SELECT_INPUT for mode ALT7.

37.4.135 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO04)

Address: 20E_0000h base + 22Ch offset = 20E_022Ch

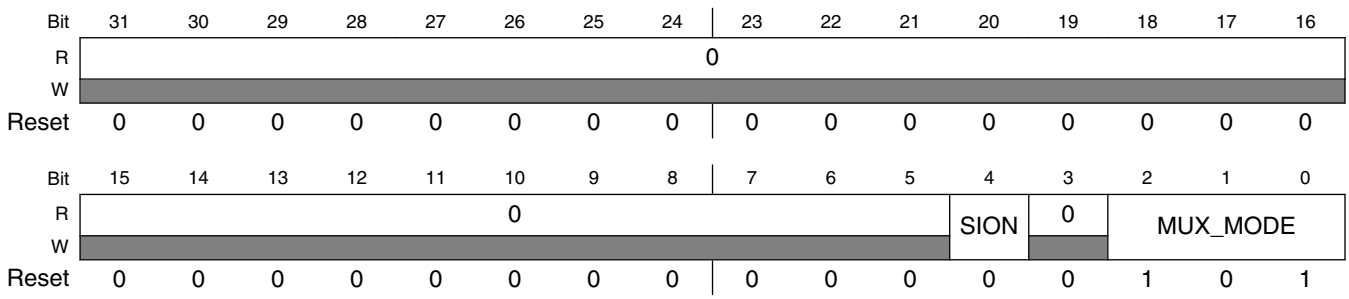


IOMUXC_SW_MUX_CTL_PAD_GPIO04 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO_4. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: GPIO_4. NOTE: Pad GPIO_4 is involved in Daisy Chain. 000 **ALT0** — Select signal ESAI_TX_HF_CLK. - Configure register IOMUXC_ESAI_TX_HF_CLK_SELECT_INPUT for mode ALT0. 010 **ALT2** — Select signal KEY_COL7. - Configure register IOMUXC_KEY_COL7_SELECT_INPUT for mode ALT2. 101 **ALT5** — Select signal GPIO1_IO04. 110 **ALT6** — Select signal SD2_CD_B.

37.4.136 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO05)

Address: 20E_0000h base + 230h offset = 20E_0230h

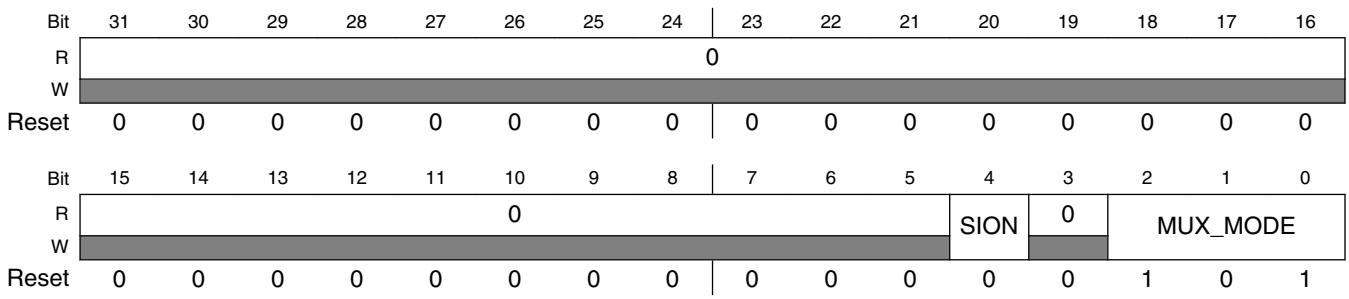


IOMUXC_SW_MUX_CTL_PAD_GPIO05 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO_5. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_5. NOTE: Pad GPIO_5 is involved in Daisy Chain. 000 **ALT0** — Select signal ESAI_TX2_RX3. - Configure register IOMUXC_ESAI_SDO2_SDI3_SELECT_INPUT for mode ALT0. 010 **ALT2** — Select signal KEY_ROW7. - Configure register IOMUXC_KEY_ROW7_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal CCM_CLKO1. 101 **ALT5** — Select signal GPIO1_IO05. 110 **ALT6** — Select signal I2C3_SCL. - Configure register IOMUXC_I2C3_SCL_IN_SELECT_INPUT for mode ALT6. 111 **ALT7** — Select signal ARM_EVENTI.

37.4.137 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO06)

Address: 20E_0000h base + 234h offset = 20E_0234h



IOMUXC_SW_MUX_CTL_PAD_GPIO06 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO_6. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO_6. NOTE: Pad GPIO_6 is involved in Daisy Chain. 000 **ALT0** — Select signal ESAI_TX_CLK. - Configure register IOMUXC_ESAI_TX_CLK_SELECT_INPUT for mode ALT0. 010 **ALT2** — Select signal I2C3_SDA. - Configure register IOMUXC_I2C3_SDA_IN_SELECT_INPUT for mode ALT2. 101 **ALT5** — Select signal GPIO1_IO06. 110 **ALT6** — Select signal SD2_LCTL. 111 **ALT7** — Select signal MLB_SIG. - Configure register IOMUXC_MLB_MLB_SIG_IN_SELECT_INPUT for mode ALT7.

37.4.138 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO07)

Address: 20E_0000h base + 238h offset = 20E_0238h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

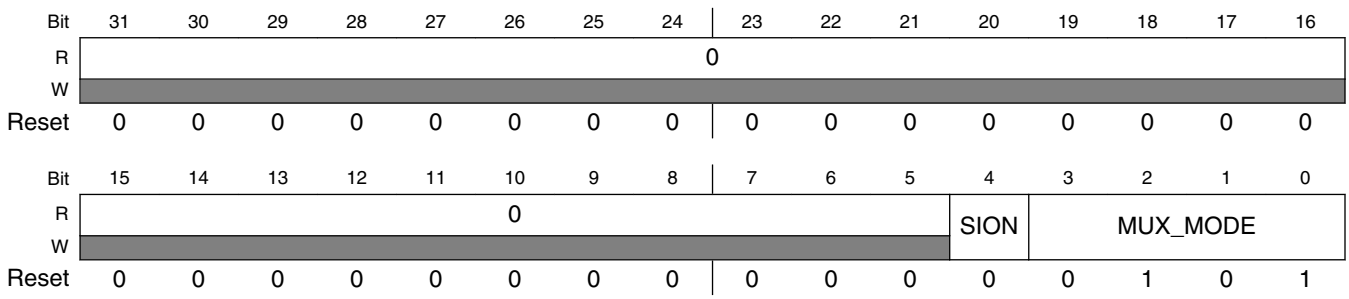
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_GPIO07 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO_7. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_7. NOTE: Pad GPIO_7 is involved in Daisy Chain. 0000 **ALT0** — Select signal ESAI_TX4_RX1. - Configure register IOMUXC_ESAI_SDO4_SDI1_SELECT_INPUT for mode ALT0. 0010 **ALT2** — Select signal EPIT1_OUT. 0011 **ALT3** — Select signal FLEXCAN1_TX. 0100 **ALT4** — Select signal UART2_TX_DATA. - Configure register IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT for mode ALT4. 0101 **ALT5** — Select signal GPIO1_IO07. 0110 **ALT6** — Select signal SPDIF_LOCK. 0111 **ALT7** — Select signal USB_OTG_HOST_MODE. 1000 **ALT8** — Select signal I2C4_SCL. - Configure register IOMUXC_I2C4_SCL_IN_SELECT_INPUT for mode ALT8.

37.4.139 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO08)

Address: 20E_0000h base + 23Ch offset = 20E_023Ch



IOMUXC_SW_MUX_CTL_PAD_GPIO08 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO_8. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: GPIO_8. NOTE: Pad GPIO_8 is involved in Daisy Chain. 0000 **ALT0** — Select signal ESAI_TX5_RX0. - Configure register IOMUXC_ESAI_SDO5_SDI0_SELECT_INPUT for mode ALT0. 0001 **ALT1** — Select signal XTALOSC_REF_CLK_32K. 0010 **ALT2** — Select signal EPIT2_OUT. 0011 **ALT3** — Select signal FLEXCAN1_RX. - Configure register IOMUXC_FLEXCAN1_RX_SELECT_INPUT for mode ALT3. 0100 **ALT4** — Select signal UART2_RX_DATA. - Configure register IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT for mode ALT4. 0101 **ALT5** — Select signal GPIO1_IO08. 0110 **ALT6** — Select signal SPDIF_SR_CLK. 0111 **ALT7** — Select signal USB_OTG_PWR_CTL_WAKE. 1000 **ALT8** — Select signal I2C4_SDA. - Configure register IOMUXC_I2C4_SDA_IN_SELECT_INPUT for mode ALT8.

37.4.140 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_GPIO09)

Address: 20E_0000h base + 240h offset = 20E_0240h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

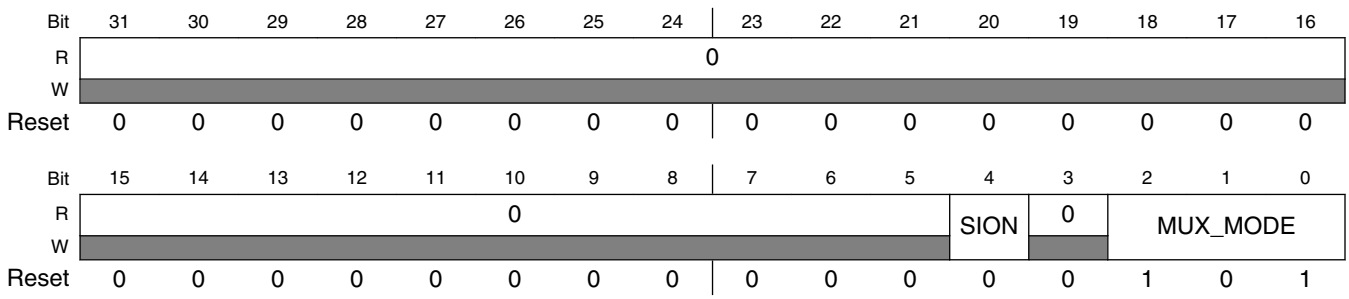
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_GPIO09 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO_9. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_9. NOTE: Pad GPIO_9 is involved in Daisy Chain. 000 **ALT0** — Select signal ESAI_RX_FS. - Configure register IOMUXC_ESAI_RX_FS_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal WDOG1_B. 010 **ALT2** — Select signal KEY_COL6. - Configure register IOMUXC_KEY_COL6_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal CCM_REF_EN_B. 100 **ALT4** — Select signal PWM1_OUT. 101 **ALT5** — Select signal GPIO1_IO09. 110 **ALT6** — Select signal SD1_WP. - Configure register IOMUXC_USDHC1_WP_ON_SELECT_INPUT for mode ALT6.

37.4.141 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_COL0)

Address: 20E_0000h base + 244h offset = 20E_0244h

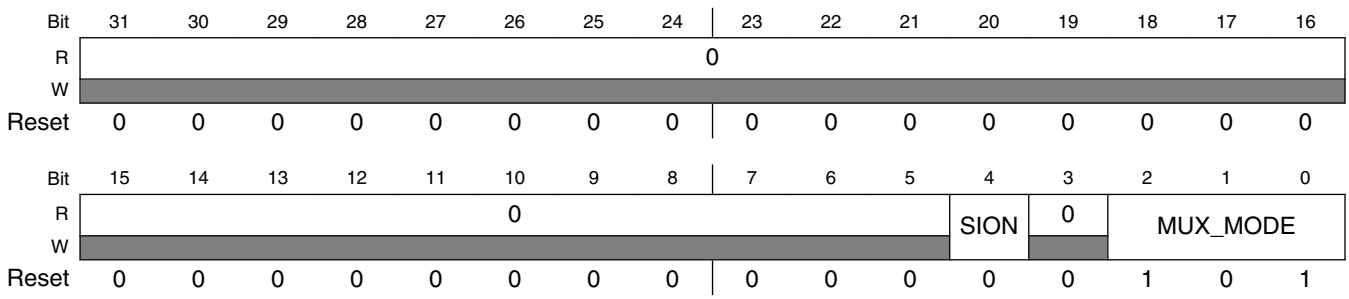


IOMUXC_SW_MUX_CTL_PAD_KEY_COL0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad KEY_COL0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: KEY_COL0. NOTE: Pad KEY_COL0 is involved in Daisy Chain. 000 **ALT0** — Select signal ECSPI1_SCLK. - Configure register IOMUXC_ECSPI1_CSPI_CLK_IN_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal ENET_RX_DATA3. - Configure register IOMUXC_ENET_MAC0_RX_DATA3_SELECT_INPUT for mode ALT1. 010 **ALT2** — Select signal AUD5_TXC. - Configure register IOMUXC_AUD5_INPUT_TXCLK_AMX_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal KEY_COL0. 100 **ALT4** — Select signal UART4_TX_DATA. - Configure register IOMUXC_UART4_UART_RX_DATA_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO4_IO06. 110 **ALT6** — Select signal DCIC1_OUT.

37.4.142 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_COL1)

Address: 20E_0000h base + 248h offset = 20E_0248h

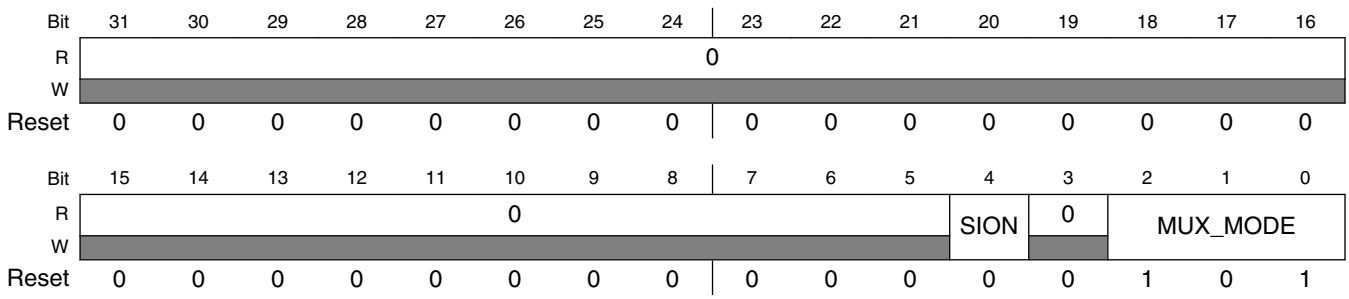


IOMUXC_SW_MUX_CTL_PAD_KEY_COL1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad KEY_COL1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: KEY_COL1. NOTE: Pad KEY_COL1 is involved in Daisy Chain. 000 **ALT0** — Select signal ECSPI1_MISO. - Configure register IOMUXC_ECSPI1_MISO_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal ENET_MDIO. - Configure register IOMUXC_ENET_MAC0_MDIO_SELECT_INPUT for mode ALT1. 010 **ALT2** — Select signal AUD5_TXFS. - Configure register IOMUXC_AUD5_INPUT_TXFS_AMX_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal KEY_COL1. 100 **ALT4** — Select signal UART5_TX_DATA. - Configure register IOMUXC_UART5_UART_RX_DATA_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO4_IO08. 110 **ALT6** — Select signal SD1_VSELECT.

37.4.143 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_COL2)

Address: 20E_0000h base + 24Ch offset = 20E_024Ch



IOMUXC_SW_MUX_CTL_PAD_KEY_COL2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad KEY_COL2. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: KEY_COL2. NOTE: Pad KEY_COL2 is involved in Daisy Chain. 000 **ALT0** — Select signal ECSPI1_SS1. - Configure register IOMUXC_ECSPI1_SS1_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal ENET_RX_DATA2. - Configure register IOMUXC_ENET_MAC0_RX_DATA2_SELECT_INPUT for mode ALT1. 010 **ALT2** — Select signal FLEXCAN1_TX. 011 **ALT3** — Select signal KEY_COL2. 100 **ALT4** — Select signal ENET_MDC. 101 **ALT5** — Select signal GPIO4_IO10. 110 **ALT6** — Select signal USB_H1_PWR_CTL_WAKE.

37.4.144 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_COL3)

Address: 20E_0000h base + 250h offset = 20E_0250h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

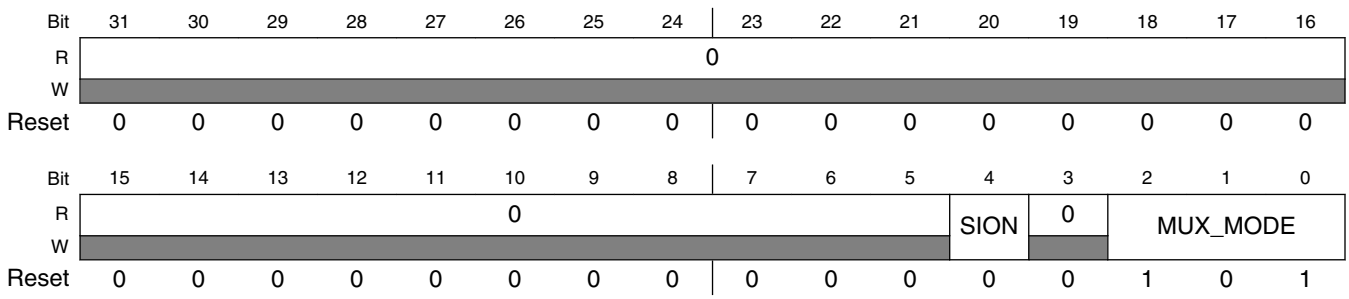
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_KEY_COL3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad KEY_COL3. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: KEY_COL3. NOTE: Pad KEY_COL3 is involved in Daisy Chain. 000 **ALT0** — Select signal ECSPI1_SS3. - Configure register IOMUXC_ECSPI1_SS3_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal ENET_CRS. 010 **ALT2** — Select signal HDMI_TX_DDC_SCL. - Configure register IOMUXC_HDMI_II2C_CLKIN_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal KEY_COL3. 100 **ALT4** — Select signal I2C2_SCL. - Configure register IOMUXC_I2C2_SCL_IN_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO4_IO12. 110 **ALT6** — Select signal SPDIF_IN. - Configure register IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT for mode ALT6.

37.4.145 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_COL4)

Address: 20E_0000h base + 254h offset = 20E_0254h

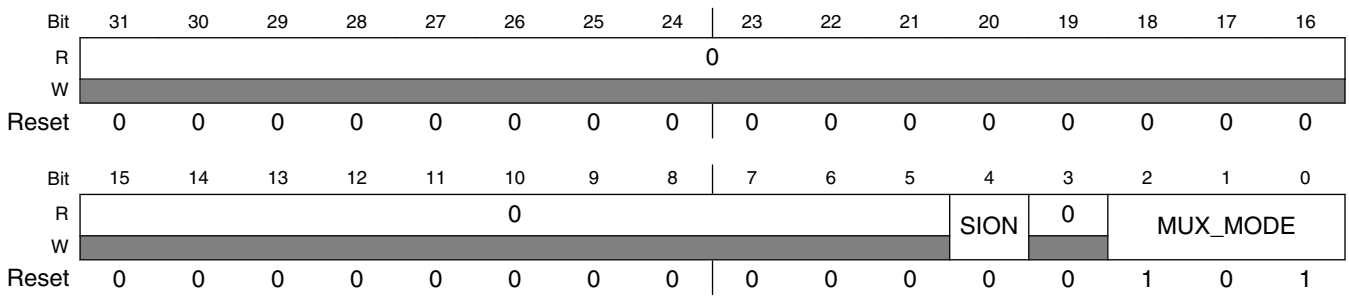


IOMUXC_SW_MUX_CTL_PAD_KEY_COL4 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad KEY_COL4. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: KEY_COL4. NOTE: Pad KEY_COL4 is involved in Daisy Chain. 000 **ALT0** — Select signal FLEXCAN2_TX. 001 **ALT1** — Select signal IPU1_SISG4. 010 **ALT2** — Select signal USB_OTG_OC. - Configure register IOMUXC_USB_OTG_OC_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal KEY_COL4. 100 **ALT4** — Select signal UART5_RTS_B. - Configure register IOMUXC_UART5_UART_RTS_B_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO4_IO14.

37.4.146 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_ROW0)

Address: 20E_0000h base + 258h offset = 20E_0258h

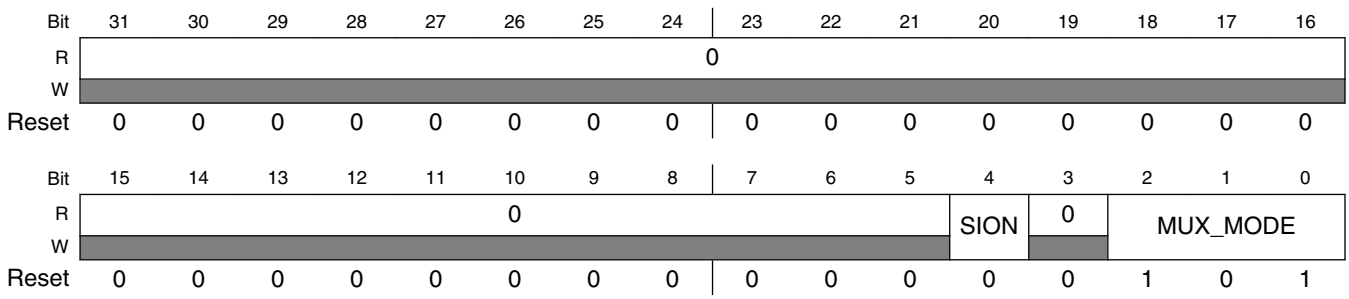


IOMUXC_SW_MUX_CTL_PAD_KEY_ROW0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad KEY_ROW0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: KEY_ROW0. NOTE: Pad KEY_ROW0 is involved in Daisy Chain. 000 **ALT0** — Select signal ECSPI1_MOSI. - Configure register IOMUXC_ECSPI1_MOSI_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal ENET_TX_DATA3. 010 **ALT2** — Select signal AUD5_TXD. - Configure register IOMUXC_AUD5_INPUT_DB_AMX_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal KEY_ROW0. 100 **ALT4** — Select signal UART4_RX_DATA. - Configure register IOMUXC_UART4_UART_RX_DATA_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO4_IO07. 110 **ALT6** — Select signal DCIC2_OUT.

37.4.147 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_ROW1)

Address: 20E_0000h base + 25Ch offset = 20E_025Ch

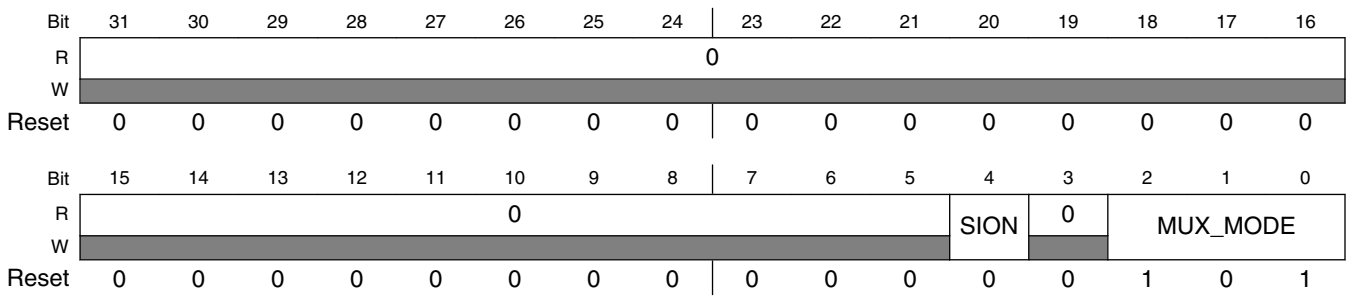


IOMUXC_SW_MUX_CTL_PAD_KEY_ROW1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad KEY_ROW1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: KEY_ROW1. NOTE: Pad KEY_ROW1 is involved in Daisy Chain. 000 **ALT0** — Select signal ECSPI1_SS0. - Configure register IOMUXC_ECSPI1_SS0_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal ENET_COL. 010 **ALT2** — Select signal AUD5_RXD. - Configure register IOMUXC_AUD5_INPUT_DA_AMX_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal KEY_ROW1. 100 **ALT4** — Select signal UART5_RX_DATA. - Configure register IOMUXC_UART5_UART_RX_DATA_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO4_IO09. 110 **ALT6** — Select signal SD2_VSELECT.

37.4.148 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_ROW2)

Address: 20E_0000h base + 260h offset = 20E_0260h

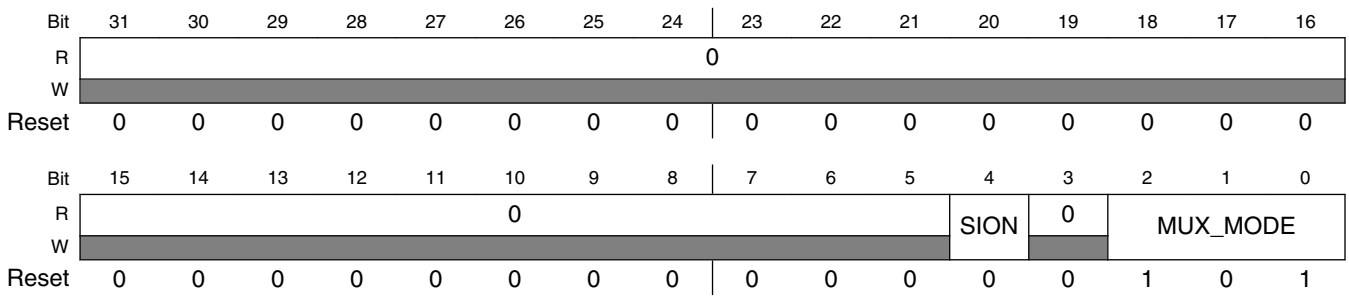


IOMUXC_SW_MUX_CTL_PAD_KEY_ROW2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad KEY_ROW2. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: KEY_ROW2. NOTE: Pad KEY_ROW2 is involved in Daisy Chain. 000 **ALT0** — Select signal ECSPI1_SS2. - Configure register IOMUXC_ECSPI1_SS2_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal ENET_TX_DATA2. 010 **ALT2** — Select signal FLEXCAN1_RX. - Configure register IOMUXC_FLEXCAN1_RX_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal KEY_ROW2. 100 **ALT4** — Select signal SD2_VSELECT. 101 **ALT5** — Select signal GPIO4_IO11. 110 **ALT6** — Select signal HDMI_TX_CEC_LINE. - Configure register IOMUXC_HDMI_ICECIN_SELECT_INPUT for mode ALT6.

37.4.149 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_ROW3)

Address: 20E_0000h base + 264h offset = 20E_0264h



IOMUXC_SW_MUX_CTL_PAD_KEY_ROW3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad KEY_ROW3. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: KEY_ROW3. NOTE: Pad KEY_ROW3 is involved in Daisy Chain. 000 **ALT0** — Select signal XTALOSC_OSC32K_32K_OUT. 001 **ALT1** — Select signal ASRC_EXT_CLK. - Configure register IOMUXC_ASRC_ASRCK_CLOCK_6_SELECT_INPUT for mode ALT1. 010 **ALT2** — Select signal HDMI_TX_DDC_SDA. - Configure register IOMUXC_HDMI_II2C_DATAIN_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal KEY_ROW3. 100 **ALT4** — Select signal I2C2_SDA. - Configure register IOMUXC_I2C2_SDA_IN_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO4_IO13. 110 **ALT6** — Select signal SD1_VSELECT.

37.4.150 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_KEY_ROW4)

Address: 20E_0000h base + 268h offset = 20E_0268h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_KEY_ROW4 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad KEY_ROW4. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: KEY_ROW4. NOTE: Pad KEY_ROW4 is involved in Daisy Chain. 000 **ALT0** — Select signal FLEXCAN2_RX. - Configure register IOMUXC_FLEXCAN2_RX_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal IPU1_SISG5. 010 **ALT2** — Select signal USB_OTG_PWR. 011 **ALT3** — Select signal KEY_ROW4. 100 **ALT4** — Select signal UART5_CTS_B. - Configure register IOMUXC_UART5_UART_RTS_B_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO4_IO15.

37.4.151 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_ALE)

Address: 20E_0000h base + 26Ch offset = 20E_026Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_NAND_ALE field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_ALE. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: NANDF_ALE. 0000 **ALT0** — Select signal NAND_ALE. 0001 **ALT1** — Select signal SD4_RESET. 0101 **ALT5** — Select signal GPIO6_IO08.

37.4.152 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CLE)

Address: 20E_0000h base + 270h offset = 20E_0270h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_NAND_CLE field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_CLE. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: NANDF_CLE. 0000 **ALT0** — Select signal NAND_CLE. 0101 **ALT5** — Select signal GPIO6_IO07.

37.4.153 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CS0_B)

Address: 20E_0000h base + 274h offset = 20E_0274h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_NAND_CS0_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_CS0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: NANDF_CS0. 0000 **ALT0** — Select signal NAND_CE0_B. 0101 **ALT5** — Select signal GPIO6_IO11.

37.4.154 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CS1_B)

Address: 20E_0000h base + 278h offset = 20E_0278h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_NAND_CS1_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_CS1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: NANDF_CS1. 0000 **ALT0** — Select signal NAND_CE1_B. 0001 **ALT1** — Select signal SD4_VSELECT. 0010 **ALT2** — Select signal SD3_VSELECT. 0101 **ALT5** — Select signal GPIO6_IO14.

37.4.155 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CS2_B)

Address: 20E_0000h base + 27Ch offset = 20E_027Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_NAND_CS2_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_CS2. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: NANDF_CS2. NOTE: Pad NANDF_CS2 is involved in Daisy Chain. 0000 **ALT0** — Select signal NAND_CE2_B. 0001 **ALT1** — Select signal IPU1_SISG0. 0010 **ALT2** — Select signal ESAI_TX0. - Configure register IOMUXC_ESAI_SDO0_SELECT_INPUT for mode ALT2. 0011 **ALT3** — Select signal EIM_CRE. 0100 **ALT4** — Select signal CCM_CLKO2. 0101 **ALT5** — Select signal GPIO6_IO15.

37.4.156 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CS3_B)

Address: 20E_0000h base + 280h offset = 20E_0280h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	MUX_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_NAND_CS3_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_CS3. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_NAND_CS3_B field descriptions (continued)

Field	Description
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: NANDF_CS3. NOTE: Pad NANDF_CS3 is involved in Daisy Chain. 0000 **ALT0** — Select signal NAND_CE3_B. 0001 **ALT1** — Select signal IPU1_SISG1. 0010 **ALT2** — Select signal ESAI_TX1. - Configure register IOMUXC_ESAI_SDO1_SELECT_INPUT for mode ALT2. 0011 **ALT3** — Select signal EIM_ADDR26. 0101 **ALT5** — Select signal GPIO6_IO16. 1001 **ALT9** — Select signal I2C4_SDA. - Configure register IOMUXC_I2C4_SDA_IN_SELECT_INPUT for mode ALT9.

37.4.157 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA00)

Address: 20E_0000h base + 284h offset = 20E_0284h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_NAND_DATA00 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_D0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: NANDF_D0. 000 **ALT0** — Select signal NAND_DATA00.

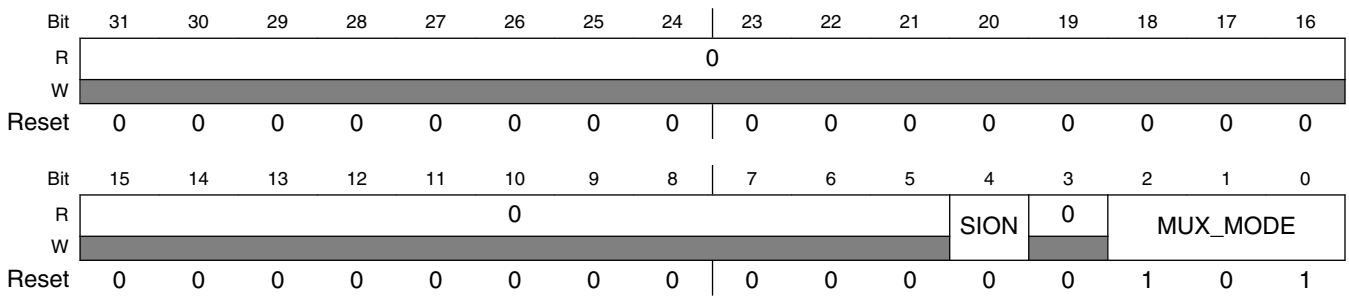
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_NAND_DATA00 field descriptions (continued)

Field	Description
	001 **ALT1** — Select signal SD1_DATA4. 101 **ALT5** — Select signal GPIO2_IO00.

37.4.158 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA01)

Address: 20E_0000h base + 288h offset = 20E_0288h



IOMUXC_SW_MUX_CTL_PAD_NAND_DATA01 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_D1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: NANDF_D1. 000 **ALT0** — Select signal NAND_DATA01. 001 **ALT1** — Select signal SD1_DATA5. 101 **ALT5** — Select signal GPIO2_IO01.

37.4.159 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA02)

Address: 20E_0000h base + 28Ch offset = 20E_028Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_NAND_DATA02 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_D2. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: NANDF_D2. 000 **ALT0** — Select signal NAND_DATA02. 001 **ALT1** — Select signal SD1_DATA6. 101 **ALT5** — Select signal GPIO2_IO02.

37.4.160 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA03)

Address: 20E_0000h base + 290h offset = 20E_0290h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_NAND_DATA03 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_D3. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: NANDF_D3. 000 **ALT0** — Select signal NAND_DATA03. 001 **ALT1** — Select signal SD1_DATA7. 101 **ALT5** — Select signal GPIO2_IO03.

37.4.161 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA04)

Address: 20E_0000h base + 294h offset = 20E_0294h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_NAND_DATA04 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_D4. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: NANDF_D4. 000 **ALT0** — Select signal NAND_DATA04. 001 **ALT1** — Select signal SD2_DATA4. 101 **ALT5** — Select signal GPIO2_IO04.

37.4.162 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA05)

Address: 20E_0000h base + 298h offset = 20E_0298h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

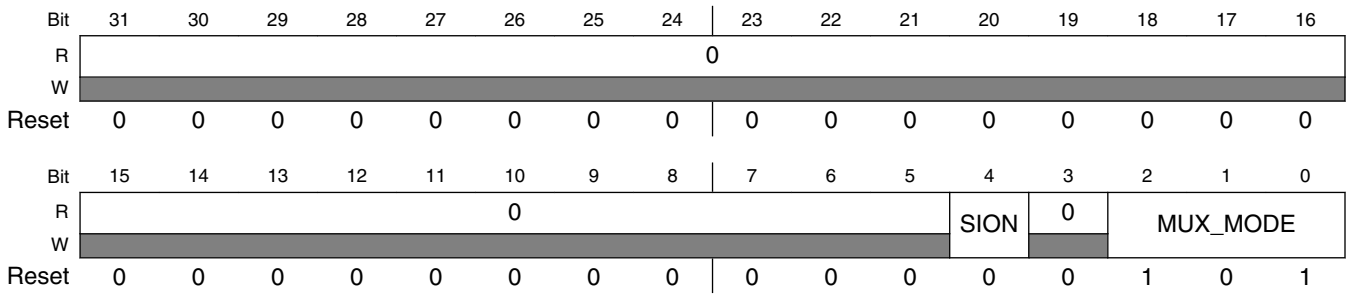
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_NAND_DATA05 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_D5. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: NANDF_D5. 000 **ALT0** — Select signal NAND_DATA05. 001 **ALT1** — Select signal SD2_DATA5. 101 **ALT5** — Select signal GPIO2_IO05.

37.4.163 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA06)

Address: 20E_0000h base + 29Ch offset = 20E_029Ch



IOMUXC_SW_MUX_CTL_PAD_NAND_DATA06 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_D6. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: NANDF_D6. 000 **ALT0** — Select signal NAND_DATA06. 001 **ALT1** — Select signal SD2_DATA6. 101 **ALT5** — Select signal GPIO2_IO06.

37.4.164 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA07)

Address: 20E_0000h base + 2A0h offset = 20E_02A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

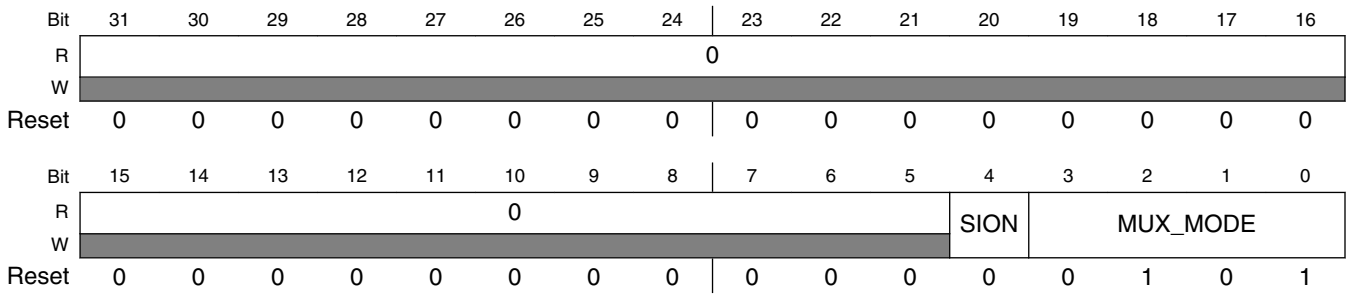
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_NAND_DATA07 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_D7. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: NANDF_D7. 000 **ALT0** — Select signal NAND_DATA07. 001 **ALT1** — Select signal SD2_DATA7. 101 **ALT5** — Select signal GPIO2_IO07.

37.4.165 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_READY_B)

Address: 20E_0000h base + 2A4h offset = 20E_02A4h

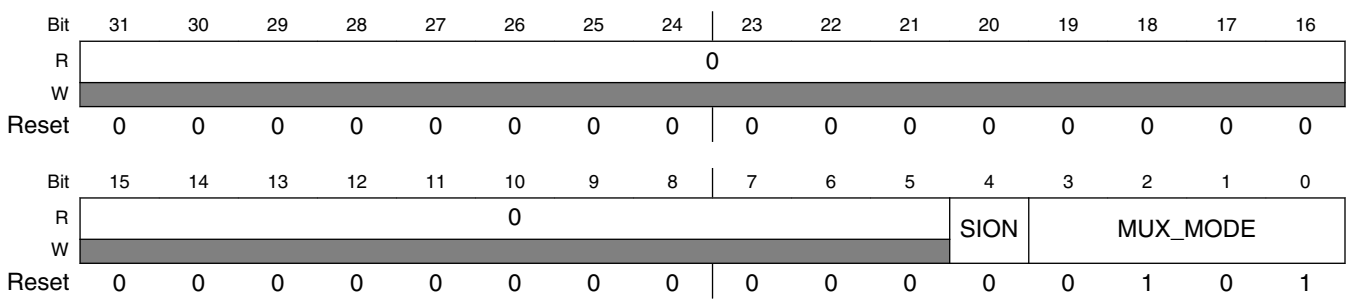


IOMUXC_SW_MUX_CTL_PAD_NAND_READY_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_RB0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: NANDF_RB0. 0000 **ALT0** — Select signal NAND_READY_B. 0101 **ALT5** — Select signal GPIO6_IO10.

37.4.166 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_NAND_WP_B)

Address: 20E_0000h base + 2A8h offset = 20E_02A8h



IOMUXC_SW_MUX_CTL_PAD_NAND_WP_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NANDF_WP_B. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: NANDF_WP_B. NOTE: Pad NANDF_WP_B is involved in Daisy Chain. 0000 **ALT0** — Select signal NAND_WP_B. 0101 **ALT5** — Select signal GPIO6_IO09. 1001 **ALT9** — Select signal I2C4_SCL. - Configure register IOMUXC_I2C4_SCL_IN_SELECT_INPUT for mode ALT9.

37.4.167 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RD0)

Address: 20E_0000h base + 2ACh offset = 20E_02ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_RGMII_RD0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad RGMII_RD0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field.

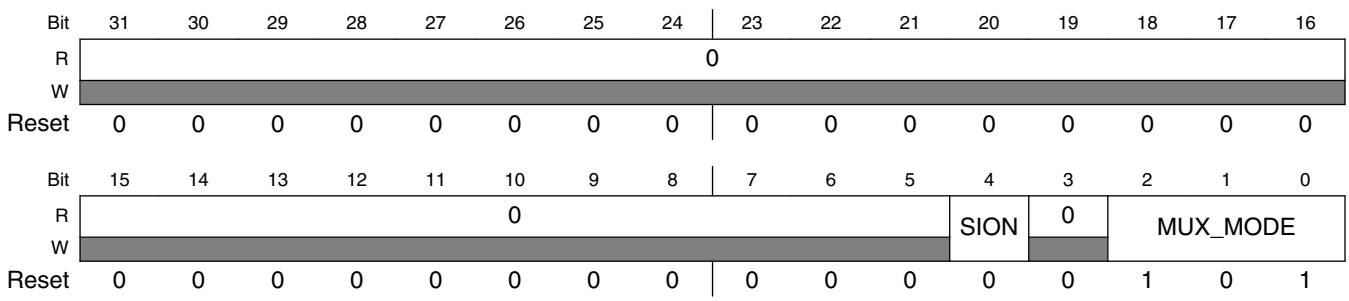
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_RGMII_RD0 field descriptions (continued)

Field	Description
	Select 1 of 3 iomux modes to be used for pad: RGMII_RD0. NOTE: Pad RGMII_RD0 is involved in Daisy Chain. 000 **ALT0** — Select signal HSI_RX_READY. 001 **ALT1** — Select signal RGMII_RD0. - Configure register IOMUXC_ENET_MAC0_RX_DATA0_SELECT_INPUT for mode ALT1. 101 **ALT5** — Select signal GPIO6_IO25.

37.4.168 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RD1)

Address: 20E_0000h base + 2B0h offset = 20E_02B0h



IOMUXC_SW_MUX_CTL_PAD_RGMII_RD1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad RGMII_RD1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: RGMII_RD1. NOTE: Pad RGMII_RD1 is involved in Daisy Chain. 000 **ALT0** — Select signal HSI_TX_FLAG. 001 **ALT1** — Select signal RGMII_RD1. - Configure register IOMUXC_ENET_MAC0_RX_DATA1_SELECT_INPUT for mode ALT1. 101 **ALT5** — Select signal GPIO6_IO27.

37.4.169 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RD2)

Address: 20E_0000h base + 2B4h offset = 20E_02B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

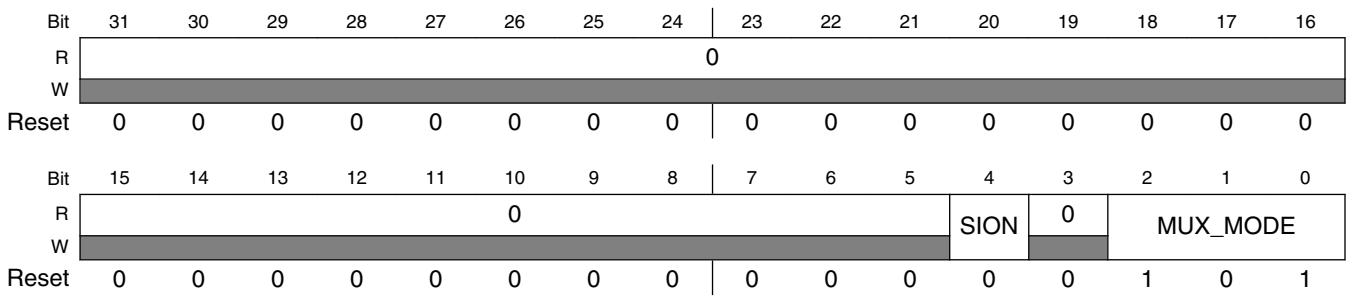
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_RGMII_RD2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad RGMII_RD2. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: RGMII_RD2. NOTE: Pad RGMII_RD2 is involved in Daisy Chain. 000 **ALT0** — Select signal HSI_TX_DATA. 001 **ALT1** — Select signal RGMII_RD2. - Configure register IOMUXC_ENET_MAC0_RX_DATA2_SELECT_INPUT for mode ALT1. 101 **ALT5** — Select signal GPIO6_IO28.

37.4.170 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RD3)

Address: 20E_0000h base + 2B8h offset = 20E_02B8h



IOMUXC_SW_MUX_CTL_PAD_RGMII_RD3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad RGMII_RD3. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: RGMII_RD3. NOTE: Pad RGMII_RD3 is involved in Daisy Chain. 000 **ALT0** — Select signal HSI_TX_WAKE. 001 **ALT1** — Select signal RGMII_RD3. - Configure register IOMUXC_ENET_MAC0_RX_DATA3_SELECT_INPUT for mode ALT1. 101 **ALT5** — Select signal GPIO6_IO29.

37.4.171 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RX_CTL)

Address: 20E_0000h base + 2BCh offset = 20E_02BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

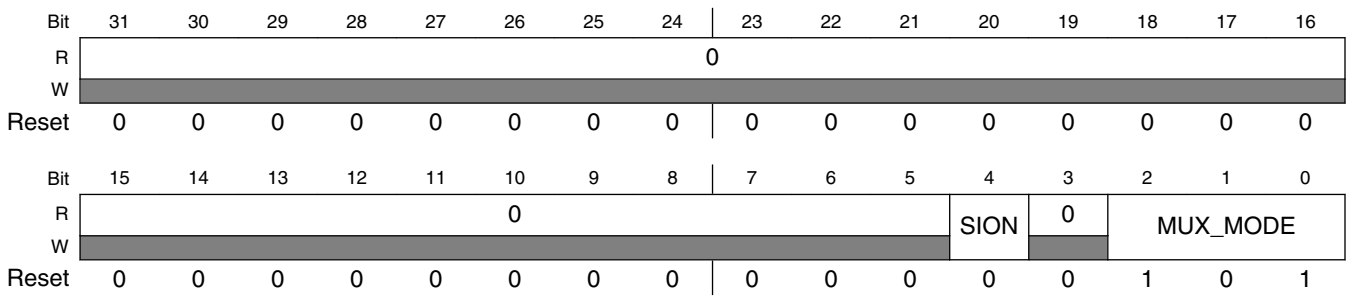
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_RGMII_RX_CTL field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad RGMII_RX_CTL. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: RGMII_RX_CTL. NOTE: Pad RGMII_RX_CTL is involved in Daisy Chain. 000 **ALT0** — Select signal USB_H3_DATA. 001 **ALT1** — Select signal RGMII_RX_CTL. - Configure register IOMUXC_ENET_MAC0_RX_EN_SELECT_INPUT for mode ALT1. 101 **ALT5** — Select signal GPIO6_IO24.

37.4.172 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_RXC)

Address: 20E_0000h base + 2C0h offset = 20E_02C0h



IOMUXC_SW_MUX_CTL_PAD_RGMII_RXC field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad RGMII_RXC. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: RGMII_RXC. NOTE: Pad RGMII_RXC is involved in Daisy Chain. 000 **ALT0** — Select signal USB_H3_STROBE. 001 **ALT1** — Select signal RGMII_RXC. - Configure register IOMUXC_ENET_MAC0_RX_CLK_SELECT_INPUT for mode ALT1. 101 **ALT5** — Select signal GPIO6_IO30.

37.4.173 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TD0)

Address: 20E_0000h base + 2C4h offset = 20E_02C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

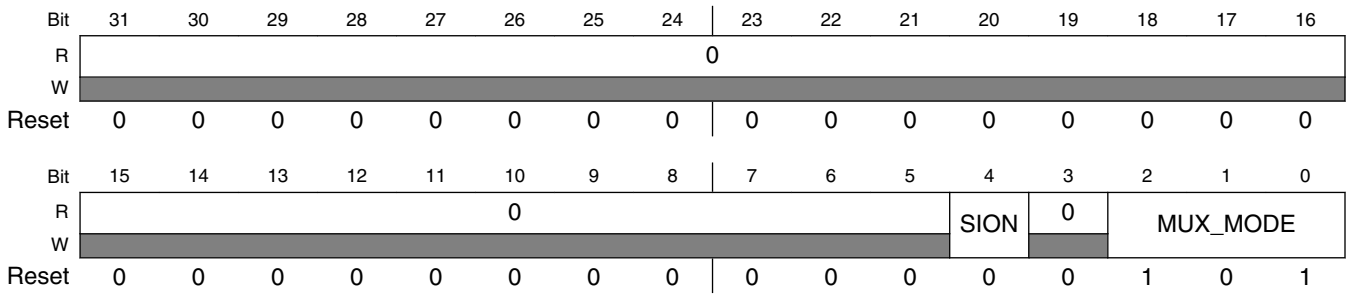
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_RGMII_TD0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad RGMII_TD0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: RGMII_TD0. 000 **ALT0** — Select signal HSI_TX_READY. 001 **ALT1** — Select signal RGMII_TD0. 101 **ALT5** — Select signal GPIO6_IO20.

37.4.174 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TD1)

Address: 20E_0000h base + 2C8h offset = 20E_02C8h



IOMUXC_SW_MUX_CTL_PAD_RGMII_TD1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad RGMII_TD1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: RGMII_TD1. 000 **ALT0** — Select signal HSI_RX_FLAG. 001 **ALT1** — Select signal RGMII_TD1. 101 **ALT5** — Select signal GPIO6_IO21.

37.4.175 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TD2)

Address: 20E_0000h base + 2CCh offset = 20E_02CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_RGMII_TD2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad RGMII_TD2. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: RGMII_TD2. 000 **ALT0** — Select signal HSI_RX_DATA. 001 **ALT1** — Select signal RGMII_TD2. 101 **ALT5** — Select signal GPIO6_IO22.

37.4.176 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TD3)

Address: 20E_0000h base + 2D0h offset = 20E_02D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_RGMII_TD3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad RGMII_TD3. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: RGMII_TD3. 000 **ALT0** — Select signal HSI_RX_WAKE. 001 **ALT1** — Select signal RGMII_TD3. 101 **ALT5** — Select signal GPIO6_IO23.

37.4.177 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TX_CTL)

Address: 20E_0000h base + 2D4h offset = 20E_02D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

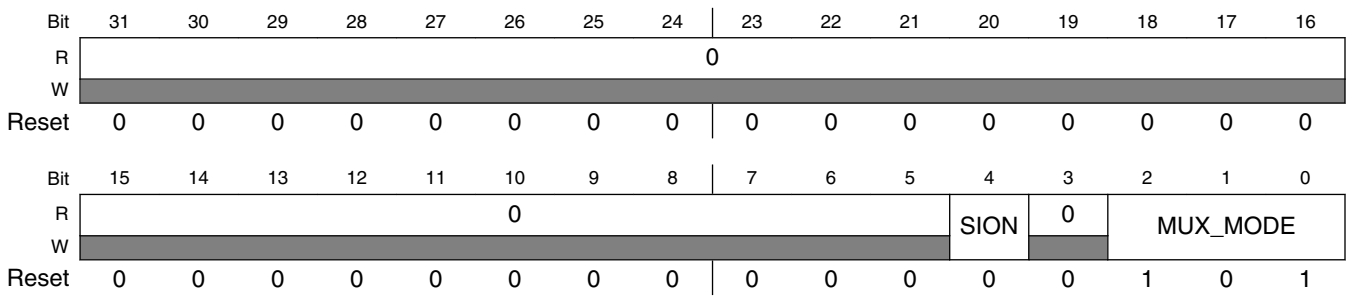
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_RGMII_TX_CTL field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad RGMII_TX_CTL. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: RGMII_TX_CTL. NOTE: Pad RGMII_TX_CTL is involved in Daisy Chain. 000 **ALT0** — Select signal USB_H2_STROBE. 001 **ALT1** — Select signal RGMII_TX_CTL. 101 **ALT5** — Select signal GPIO6_IO26. 111 **ALT7** — Select signal ENET_REF_CLK. - Configure register IOMUXC_ENET_REF_CLK_SELECT_INPUT for mode ALT7.

37.4.178 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_RGMII_TXC)

Address: 20E_0000h base + 2D8h offset = 20E_02D8h



IOMUXC_SW_MUX_CTL_PAD_RGMII_TXC field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad RGMII_TXC. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: RGMII_TXC. NOTE: Pad RGMII_TXC is involved in Daisy Chain. 000 **ALT0** — Select signal USB_H2_DATA. 001 **ALT1** — Select signal RGMII_TXC. 010 **ALT2** — Select signal SPDIF_EXT_CLK. - Configure register IOMUXC_SPDIF_TX_CLK2_SELECT_INPUT for mode ALT2. 101 **ALT5** — Select signal GPIO6_IO19. 111 **ALT7** — Select signal XTALOSC_REF_CLK_24M.

37.4.179 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_CLK)

Address: 20E_0000h base + 2DCh offset = 20E_02DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD1_CLK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_CLK. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD1_CLK. NOTE: Pad SD1_CLK is involved in Daisy Chain. 000 **ALT0** — Select signal SD1_CLK. - Configure register IOMUXC_USDHC1_CARD_CLK_IN_SELECT_INPUT for mode ALT0. 011 **ALT3** — Select signal GPT_CLKIN. 101 **ALT5** — Select signal GPIO1_IO20.

37.4.180 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_CMD)

Address: 20E_0000h base + 2E0h offset = 20E_02E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD1_CMD field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_CMD. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD1_CMD. 000 **ALT0** — Select signal SD1_CMD. 010 **ALT2** — Select signal PWM4_OUT. 011 **ALT3** — Select signal GPT_COMPARE1. 101 **ALT5** — Select signal GPIO1_IO18.

37.4.181 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0)

Address: 20E_0000h base + 2E4h offset = 20E_02E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_DAT0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD1_DAT0. 000 **ALT0** — Select signal SD1_DATA0. 011 **ALT3** — Select signal GPT_CAPTURE1. 101 **ALT5** — Select signal GPIO1_IO16.

37.4.182 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1)

Address: 20E_0000h base + 2E8h offset = 20E_02E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_DAT1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD1_DAT1. 000 **ALT0** — Select signal SD1_DATA1. 010 **ALT2** — Select signal PWM3_OUT. 011 **ALT3** — Select signal GPT_CAPTURE2. 101 **ALT5** — Select signal GPIO1_IO17.

37.4.183 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2)

Address: 20E_0000h base + 2ECh offset = 20E_02ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_DAT2. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD1_DAT2. 000 **ALT0** — Select signal SD1_DATA2. 010 **ALT2** — Select signal GPT_COMPARE2. 011 **ALT3** — Select signal PWM2_OUT. 100 **ALT4** — Select signal WDOG1_B. 101 **ALT5** — Select signal GPIO1_IO19. 110 **ALT6** — Select signal WDOG1_RESET_B_DEB.

37.4.184 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3)

Address: 20E_0000h base + 2F0h offset = 20E_02F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_DAT3. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD1_DAT3. 000 **ALT0** — Select signal SD1_DATA3. 010 **ALT2** — Select signal GPT_COMPARE3. 011 **ALT3** — Select signal PWM1_OUT. 100 **ALT4** — Select signal WDOG2_B. 101 **ALT5** — Select signal GPIO1_IO21. 110 **ALT6** — Select signal WDOG2_RESET_B_DEB.

37.4.185 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CLK)

Address: 20E_0000h base + 2F4h offset = 20E_02F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

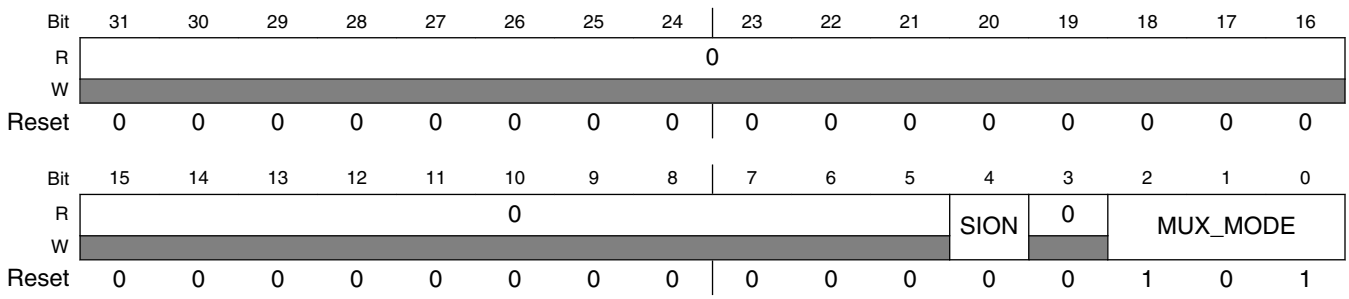
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD2_CLK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD2_CLK. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD2_CLK. NOTE: Pad SD2_CLK is involved in Daisy Chain. 000 **ALT0** — Select signal SD2_CLK. - Configure register IOMUXC_USDHC2_CARD_CLK_IN_SELECT_INPUT for mode ALT0. 010 **ALT2** — Select signal KEY_COL5. - Configure register IOMUXC_KEY_COL5_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal AUD4_RXFS. - Configure register IOMUXC_AUD4_INPUT_RXFS_AMX_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO1_IO10.

37.4.186 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CMD)

Address: 20E_0000h base + 2F8h offset = 20E_02F8h



IOMUXC_SW_MUX_CTL_PAD_SD2_CMD field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD2_CMD. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD2_CMD. NOTE: Pad SD2_CMD is involved in Daisy Chain. 000 **ALT0** — Select signal SD2_CMD. 010 **ALT2** — Select signal KEY_ROW5. - Configure register IOMUXC_KEY_ROW5_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal AUD4_RXC. - Configure register IOMUXC_AUD4_INPUT_RXCLK_AMX_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO1_IO11.

37.4.187 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0)

Address: 20E_0000h base + 2FCh offset = 20E_02FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

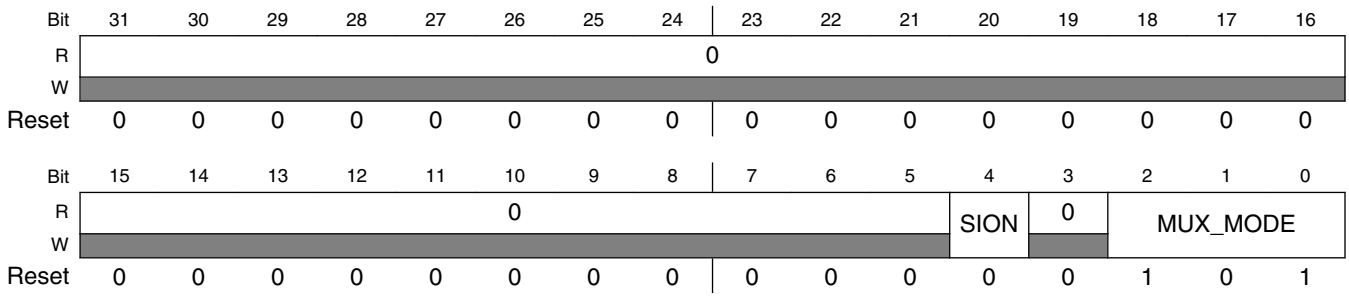
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD2_DAT0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SD2_DAT0. NOTE: Pad SD2_DAT0 is involved in Daisy Chain. 000 **ALT0** — Select signal SD2_DATA0. 011 **ALT3** — Select signal AUD4_RXD. - Configure register IOMUXC_AUD4_INPUT_DA_AMX_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal KEY_ROW7. - Configure register IOMUXC_KEY_ROW7_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO1_IO15. 110 **ALT6** — Select signal DCIC2_OUT.

37.4.188 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1)

Address: 20E_0000h base + 300h offset = 20E_0300h



IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD2_DAT1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SD2_DAT1. NOTE: Pad SD2_DAT1 is involved in Daisy Chain. 000 **ALT0** — Select signal SD2_DATA1. 010 **ALT2** — Select signal EIM_CS2_B. 011 **ALT3** — Select signal AUD4_TXFS. - Configure register IOMUXC_AUD4_INPUT_TXFS_AMX_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal KEY_COL7. - Configure register IOMUXC_KEY_COL7_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO1_IO14.

37.4.189 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2)

Address: 20E_0000h base + 304h offset = 20E_0304h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

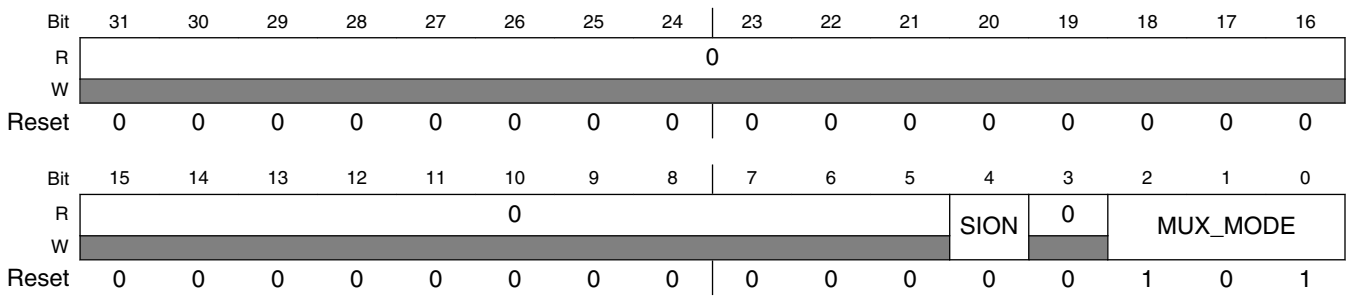
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD2_DAT2. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SD2_DAT2. NOTE: Pad SD2_DAT2 is involved in Daisy Chain. 000 **ALT0** — Select signal SD2_DATA2. 010 **ALT2** — Select signal EIM_CS3_B. 011 **ALT3** — Select signal AUD4_TXD. - Configure register IOMUXC_AUD4_INPUT_DB_AMX_SELECT_INPUT for mode ALT3. 100 **ALT4** — Select signal KEY_ROW6. - Configure register IOMUXC_KEY_ROW6_SELECT_INPUT for mode ALT4. 101 **ALT5** — Select signal GPIO1_IO13.

37.4.190 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3)

Address: 20E_0000h base + 308h offset = 20E_0308h

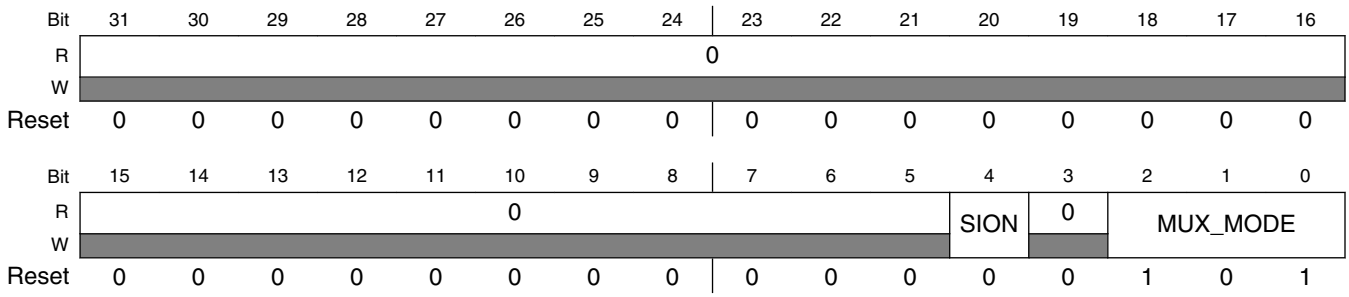


IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD2_DAT3. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD2_DAT3. NOTE: Pad SD2_DAT3 is involved in Daisy Chain. 000 **ALT0** — Select signal SD2_DATA3. 010 **ALT2** — Select signal KEY_COL6. - Configure register IOMUXC_KEY_COL6_SELECT_INPUT for mode ALT2. 011 **ALT3** — Select signal AUD4_TXC. - Configure register IOMUXC_AUD4_INPUT_TXCLK_AMX_SELECT_INPUT for mode ALT3. 101 **ALT5** — Select signal GPIO1_IO12.

37.4.191 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_CLK)

Address: 20E_0000h base + 30Ch offset = 20E_030Ch

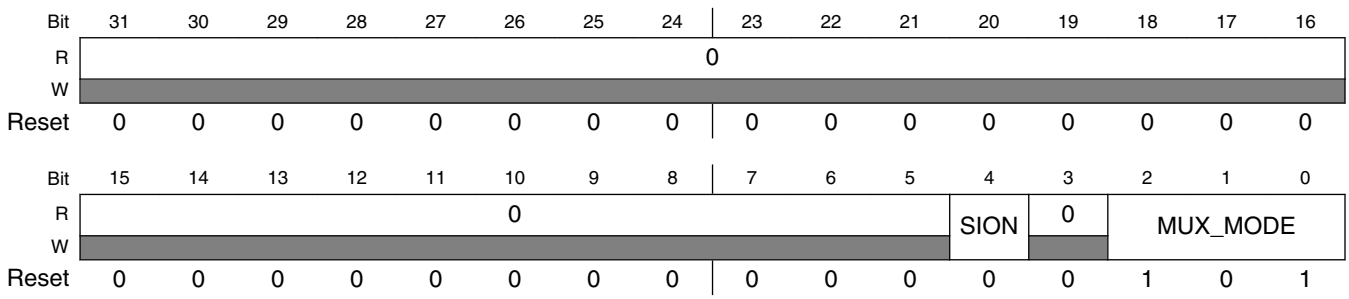


IOMUXC_SW_MUX_CTL_PAD_SD3_CLK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD3_CLK. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD3_CLK. NOTE: Pad SD3_CLK is involved in Daisy Chain. 000 **ALT0** — Select signal SD3_CLK. - Configure register IOMUXC_USDHC3_CARD_CLK_IN_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal UART2_RTS_B. - Configure register IOMUXC_UART2_UART_RTS_B_SELECT_INPUT for mode ALT1. 010 **ALT2** — Select signal FLEXCAN1_RX. - Configure register IOMUXC_FLEXCAN1_RX_SELECT_INPUT for mode ALT2. 101 **ALT5** — Select signal GPIO7_IO03.

37.4.192 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_CMD)

Address: 20E_0000h base + 310h offset = 20E_0310h



IOMUXC_SW_MUX_CTL_PAD_SD3_CMD field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD3_CMD. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD3_CMD. NOTE: Pad SD3_CMD is involved in Daisy Chain. 000 **ALT0** — Select signal SD3_CMD. 001 **ALT1** — Select signal UART2_CTS_B. - Configure register IOMUXC_UART2_UART_RTS_B_SELECT_INPUT for mode ALT1. 010 **ALT2** — Select signal FLEXCAN1_TX. 101 **ALT5** — Select signal GPIO7_IO02.

37.4.193 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA0)

Address: 20E_0000h base + 314h offset = 20E_0314h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

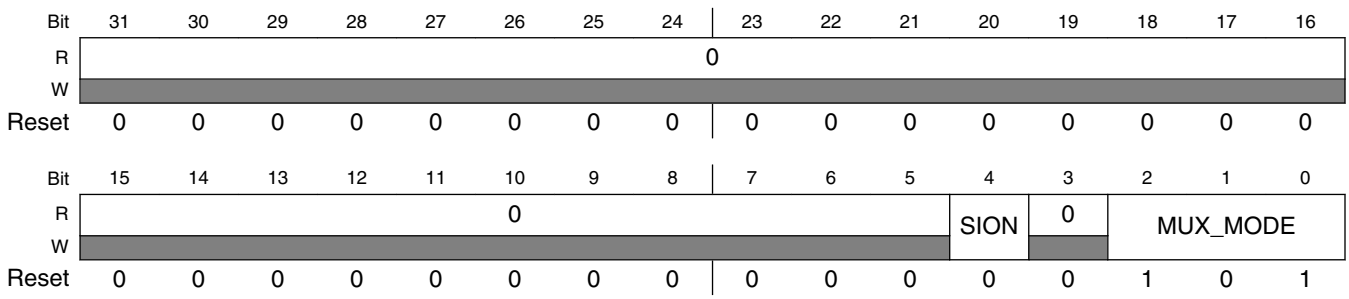
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD3_DATA0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD3_DAT0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD3_DAT0. NOTE: Pad SD3_DAT0 is involved in Daisy Chain. 000 **ALT0** — Select signal SD3_DATA0. 001 **ALT1** — Select signal UART1_CTS_B. - Configure register IOMUXC_UART1_UART_RTS_B_SELECT_INPUT for mode ALT1. 010 **ALT2** — Select signal FLEXCAN2_TX. 101 **ALT5** — Select signal GPIO7_IO04.

37.4.194 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA1)

Address: 20E_0000h base + 318h offset = 20E_0318h



IOMUXC_SW_MUX_CTL_PAD_SD3_DATA1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD3_DAT1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD3_DAT1. NOTE: Pad SD3_DAT1 is involved in Daisy Chain. 000 **ALT0** — Select signal SD3_DATA1. 001 **ALT1** — Select signal UART1_RTS_B. - Configure register IOMUXC_UART1_UART_RTS_B_SELECT_INPUT for mode ALT1. 010 **ALT2** — Select signal FLEXCAN2_RX. - Configure register IOMUXC_FLEXCAN2_RX_SELECT_INPUT for mode ALT2. 101 **ALT5** — Select signal GPIO7_IO05.

37.4.195 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA2)

Address: 20E_0000h base + 31Ch offset = 20E_031Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD3_DATA2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD3_DAT2. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: SD3_DAT2. 000 **ALT0** — Select signal SD3_DATA2. 101 **ALT5** — Select signal GPIO7_IO06.

37.4.196 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA3)

Address: 20E_0000h base + 320h offset = 20E_0320h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD3_DATA3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD3_DAT3. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD3_DAT3. NOTE: Pad SD3_DAT3 is involved in Daisy Chain. 000 **ALT0** — Select signal SD3_DATA3. 001 **ALT1** — Select signal UART3_CTS_B. - Configure register IOMUXC_UART3_UART_RTS_B_SELECT_INPUT for mode ALT1. 101 **ALT5** — Select signal GPIO7_IO07.

37.4.197 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA4)

Address: 20E_0000h base + 324h offset = 20E_0324h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD3_DATA4 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD3_DAT4. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SD3_DATA4 field descriptions (continued)

Field	Description
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD3_DAT4. NOTE: Pad SD3_DAT4 is involved in Daisy Chain. 000 **ALT0** — Select signal SD3_DATA4. 001 **ALT1** — Select signal UART2_RX_DATA. - Configure register IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT for mode ALT1. 101 **ALT5** — Select signal GPIO7_IO01.

37.4.198 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA5)

Address: 20E_0000h base + 328h offset = 20E_0328h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD3_DATA5 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD3_DAT5. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD3_DAT5. NOTE: Pad SD3_DAT5 is involved in Daisy Chain. 000 **ALT0** — Select signal SD3_DATA5. 001 **ALT1** — Select signal UART2_TX_DATA.

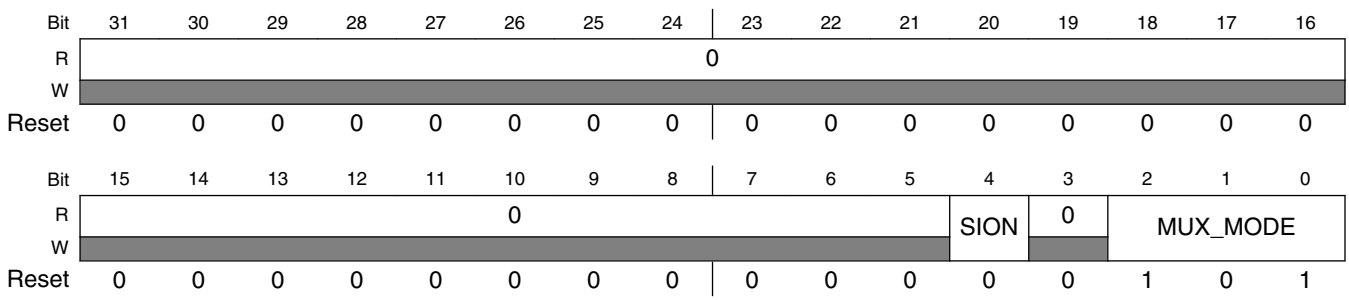
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SD3_DATA5 field descriptions (continued)

Field	Description
	- Configure register IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT for mode ALT1. 101 **ALT5** — Select signal GPIO7_IO00.

37.4.199 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA6)

Address: 20E_0000h base + 32Ch offset = 20E_032Ch



IOMUXC_SW_MUX_CTL_PAD_SD3_DATA6 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD3_DAT6. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD3_DAT6. NOTE: Pad SD3_DAT6 is involved in Daisy Chain. 000 **ALT0** — Select signal SD3_DATA6. 001 **ALT1** — Select signal UART1_RX_DATA. - Configure register IOMUXC_UART1_UART_RX_DATA_SELECT_INPUT for mode ALT1. 101 **ALT5** — Select signal GPIO6_IO18.

37.4.200 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_DATA7)

Address: 20E_0000h base + 330h offset = 20E_0330h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

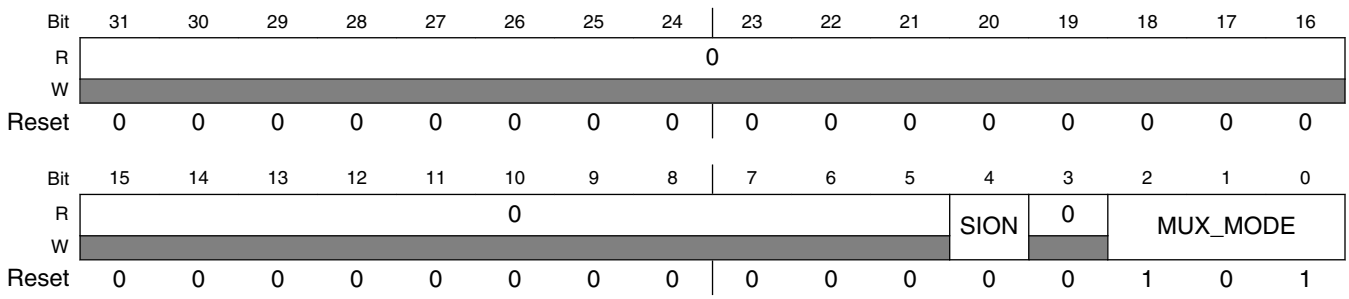
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD3_DATA7 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD3_DAT7. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD3_DAT7. NOTE: Pad SD3_DAT7 is involved in Daisy Chain. 000 **ALT0** — Select signal SD3_DATA7. 001 **ALT1** — Select signal UART1_TX_DATA. - Configure register IOMUXC_UART1_UART_RX_DATA_SELECT_INPUT for mode ALT1. 101 **ALT5** — Select signal GPIO6_IO17.

37.4.201 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD3_RESET)

Address: 20E_0000h base + 334h offset = 20E_0334h



IOMUXC_SW_MUX_CTL_PAD_SD3_RESET field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD3_RST. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD3_RST. NOTE: Pad SD3_RST is involved in Daisy Chain. 000 **ALT0** — Select signal SD3_RESET. 001 **ALT1** — Select signal UART3_RTS_B. - Configure register IOMUXC_UART3_UART_RTS_B_SELECT_INPUT for mode ALT1. 101 **ALT5** — Select signal GPIO7_IO08.

37.4.202 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_CLK)

Address: 20E_0000h base + 338h offset = 20E_0338h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

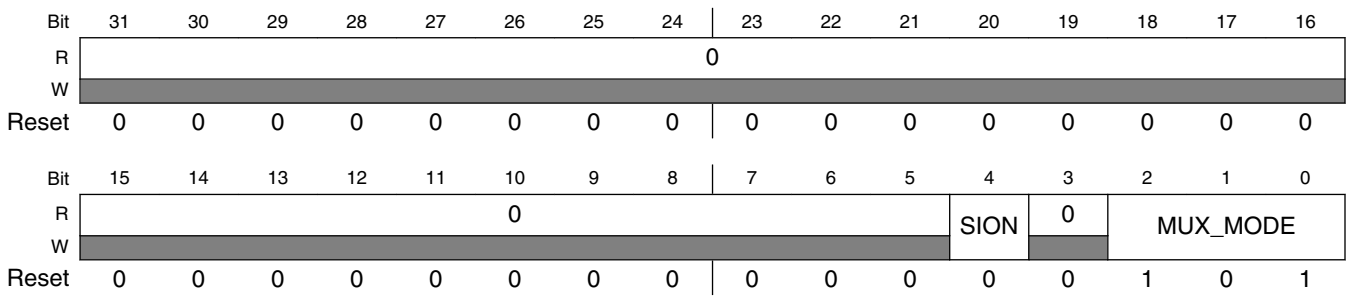
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD4_CLK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD4_CLK. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD4_CLK. NOTE: Pad SD4_CLK is involved in Daisy Chain. 000 **ALT0** — Select signal SD4_CLK. - Configure register IOMUXC_USDHC4_CARD_CLK_IN_SELECT_INPUT for mode ALT0. 001 **ALT1** — Select signal NAND_WE_B. 010 **ALT2** — Select signal UART3_RX_DATA. - Configure register IOMUXC_UART3_UART_RX_DATA_SELECT_INPUT for mode ALT2. 101 **ALT5** — Select signal GPIO7_IO10.

37.4.203 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_CMD)

Address: 20E_0000h base + 33Ch offset = 20E_033Ch



IOMUXC_SW_MUX_CTL_PAD_SD4_CMD field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD4_CMD. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SD4_CMD. NOTE: Pad SD4_CMD is involved in Daisy Chain. 000 **ALT0** — Select signal SD4_CMD. 001 **ALT1** — Select signal NAND_RE_B. 010 **ALT2** — Select signal UART3_TX_DATA. - Configure register IOMUXC_UART3_UART_RX_DATA_SELECT_INPUT for mode ALT2. 101 **ALT5** — Select signal GPIO7_IO09.

37.4.204 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA0)

Address: 20E_0000h base + 340h offset = 20E_0340h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

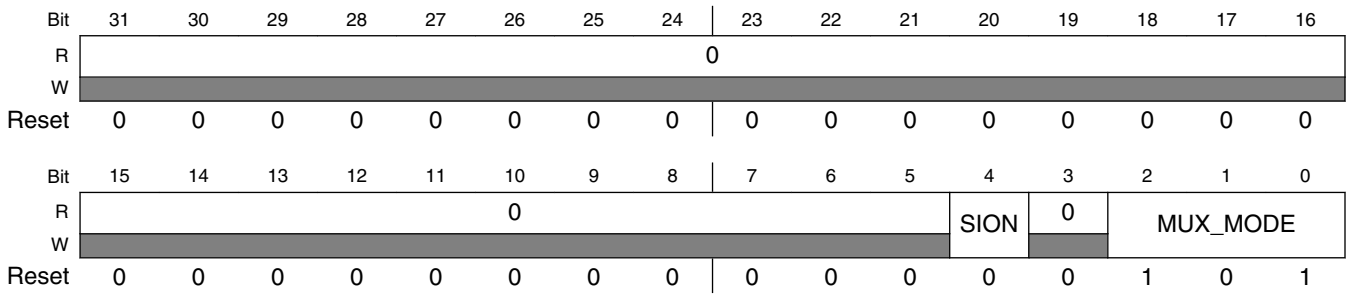
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD4_DATA0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD4_DAT0. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD4_DAT0. 001 **ALT1** — Select signal SD4_DATA0. 010 **ALT2** — Select signal NAND_DQS. 101 **ALT5** — Select signal GPIO2_IO08.

37.4.205 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA1)

Address: 20E_0000h base + 344h offset = 20E_0344h



IOMUXC_SW_MUX_CTL_PAD_SD4_DATA1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD4_DAT1. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD4_DAT1. 001 **ALT1** — Select signal SD4_DATA1. 010 **ALT2** — Select signal PWM3_OUT. 101 **ALT5** — Select signal GPIO2_IO09.

37.4.206 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA2)

Address: 20E_0000h base + 348h offset = 20E_0348h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD4_DATA2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD4_DAT2. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD4_DAT2. 001 **ALT1** — Select signal SD4_DATA2. 010 **ALT2** — Select signal PWM4_OUT. 101 **ALT5** — Select signal GPIO2_IO10.

37.4.207 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA3)

Address: 20E_0000h base + 34Ch offset = 20E_034Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD4_DATA3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD4_DAT3. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: SD4_DAT3. 001 **ALT1** — Select signal SD4_DATA3. 101 **ALT5** — Select signal GPIO2_IO11.

37.4.208 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA4)

Address: 20E_0000h base + 350h offset = 20E_0350h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD4_DATA4 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD4_DAT4. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD4_DAT4. NOTE: Pad SD4_DAT4 is involved in Daisy Chain. 001 **ALT1** — Select signal SD4_DATA4. 010 **ALT2** — Select signal UART2_RX_DATA. - Configure register IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT for mode ALT2. 101 **ALT5** — Select signal GPIO2_IO12.

37.4.209 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA5)

Address: 20E_0000h base + 354h offset = 20E_0354h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD4_DATA5 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD4_DAT5. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SD4_DATA5 field descriptions (continued)

Field	Description
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD4_DAT5. NOTE: Pad SD4_DAT5 is involved in Daisy Chain. 001 **ALT1** — Select signal SD4_DATA5. 010 **ALT2** — Select signal UART2_RTS_B. - Configure register IOMUXC_UART2_UART_RTS_B_SELECT_INPUT for mode ALT2. 101 **ALT5** — Select signal GPIO2_IO13.

37.4.210 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA6)

Address: 20E_0000h base + 358h offset = 20E_0358h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD4_DATA6 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD4_DAT6. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD4_DAT6. NOTE: Pad SD4_DAT6 is involved in Daisy Chain. 001 **ALT1** — Select signal SD4_DATA6. 010 **ALT2** — Select signal UART2_CTS_B.

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SD4_DATA6 field descriptions (continued)

Field	Description
	- Configure register IOMUXC_UART2_UART_RTS_B_SELECT_INPUT for mode ALT2. 101 **ALT5** — Select signal GPIO2_IO14.

37.4.211 Pad Mux Register (IOMUXC_SW_MUX_CTL_PAD_SD4_DATA7)

Address: 20E_0000h base + 35Ch offset = 20E_035Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SION	0	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD4_DATA7 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 SION	Software Input On Field. Force the selected mux mode input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD4_DAT7. 0 **DISABLED** — Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value 0.
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD4_DAT7. NOTE: Pad SD4_DAT7 is involved in Daisy Chain. 001 **ALT1** — Select signal SD4_DATA7. 010 **ALT2** — Select signal UART2_TX_DATA. - Configure register IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT for mode ALT2. 101 **ALT5** — Select signal GPIO2_IO15.

37.4.212 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA10)

Address: 20E_0000h base + 360h offset = 20E_0360h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA10 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT10. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT10. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT10. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT10. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA10 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT10. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.213 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA11)

Address: 20E_0000h base + 364h offset = 20E_0364h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA11 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT11. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT11. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT11. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT11. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT11. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA11 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.214 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA12)

Address: 20E_0000h base + 368h offset = 20E_0368h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA12 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT12. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT12. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA12 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT12. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT12. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT12. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA12 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.215 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA13)

Address: 20E_0000h base + 36Ch offset = 20E_036Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA13 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT13. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT13. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT13. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT13.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA13 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT13. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.216 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA14)

Address: 20E_0000h base + 370h offset = 20E_0370h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA14 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT14. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT14. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT14. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT14. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA14 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT14. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.217 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA15)

Address: 20E_0000h base + 374h offset = 20E_0374h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA15 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT15. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT15. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT15. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT15. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT15. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA15 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.218 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA16)

Address: 20E_0000h base + 378h offset = 20E_0378h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA16 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT16. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT16. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA16 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT16. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT16. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT16. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA16 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.219 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA17)

Address: 20E_0000h base + 37Ch offset = 20E_037Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA17 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT17. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT17. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT17. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT17.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA17 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT17. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.220 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA18)

Address: 20E_0000h base + 380h offset = 20E_0380h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA18 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT18. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT18. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT18. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT18. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA18 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT18. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.221 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA19)

Address: 20E_0000h base + 384h offset = 20E_0384h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA19 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT19. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT19. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT19. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT19. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT19. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA19 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.222 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA04)

Address: 20E_0000h base + 388h offset = 20E_0388h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA04 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT4. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT4. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA04 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT4. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT4. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT4. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA04 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.223 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA05)

Address: 20E_0000h base + 38Ch offset = 20E_038Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA05 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT5. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT5. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT5. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT5.

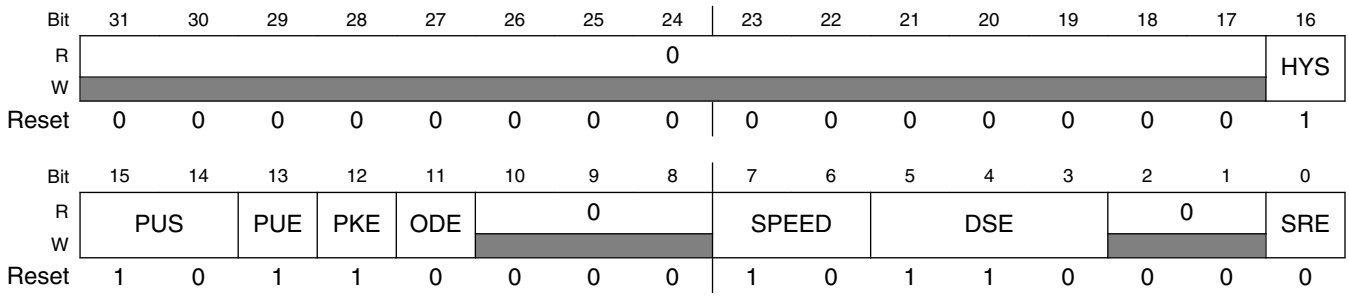
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA05 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT5. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.224 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA06)

Address: 20E_0000h base + 390h offset = 20E_0390h



IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA06 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT6. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT6. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT6. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT6. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA06 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT6. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.225 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA07)

Address: 20E_0000h base + 394h offset = 20E_0394h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA07 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT7. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT7. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT7. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT7. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT7. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA07 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.226 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA08)

Address: 20E_0000h base + 398h offset = 20E_0398h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA08 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT8. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT8. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA08 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT8. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT8. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT8. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA08 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.227 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA09)

Address: 20E_0000h base + 39Ch offset = 20E_039Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA09 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DAT9. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DAT9. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DAT9. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DAT9.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA09 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DAT9. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.228 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA_EN)

Address: 20E_0000h base + 3A0h offset = 20E_03A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA_EN field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_DATA_EN. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_DATA_EN. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_DATA_EN. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_DATA_EN. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA_EN field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_DATA_EN. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.229 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_HSYNC)

Address: 20E_0000h base + 3A4h offset = 20E_03A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_HSYNC field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_MCLK. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_MCLK. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_MCLK. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_MCLK. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_MCLK. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_HSYNC field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.230 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_PIXCLK)

Address: 20E_0000h base + 3A8h offset = 20E_03A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_PIXCLK field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_PIXCLK. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_PIXCLK. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_PIXCLK field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_PIXCLK. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_PIXCLK. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_PIXCLK. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_PIXCLK field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.231 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_CSI0_VSYNC)

Address: 20E_0000h base + 3ACh offset = 20E_03ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_CSI0_VSYNC field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: CSI0_VSYNC. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: CSI0_VSYNC. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: CSI0_VSYNC. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: CSI0_VSYNC.

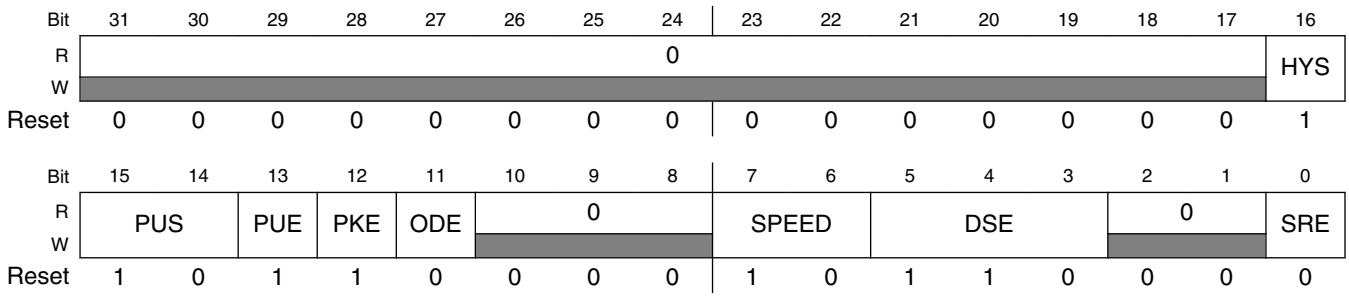
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_VSYNC field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: CSI0_VSYNC. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.232 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DI0_DISP_CLK)

Address: 20E_0000h base + 3B0h offset = 20E_03B0h



IOMUXC_SW_PAD_CTL_PAD_DI0_DISP_CLK field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DI0_DISP_CLK. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DI0_DISP_CLK. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DI0_DISP_CLK. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DI0_DISP_CLK. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DI0_DISP_CLK field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DI0_DISP_CLK. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.233 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN15)

Address: 20E_0000h base + 3B4h offset = 20E_03B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN15 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DI0_PIN15. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DI0_PIN15. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DI0_PIN15. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DI0_PIN15. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DI0_PIN15. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN15 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.234 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN02)

Address: 20E_0000h base + 3B8h offset = 20E_03B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN02 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DI0_PIN2. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DI0_PIN2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN02 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DI0_PIN2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DI0_PIN2. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DI0_PIN2. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

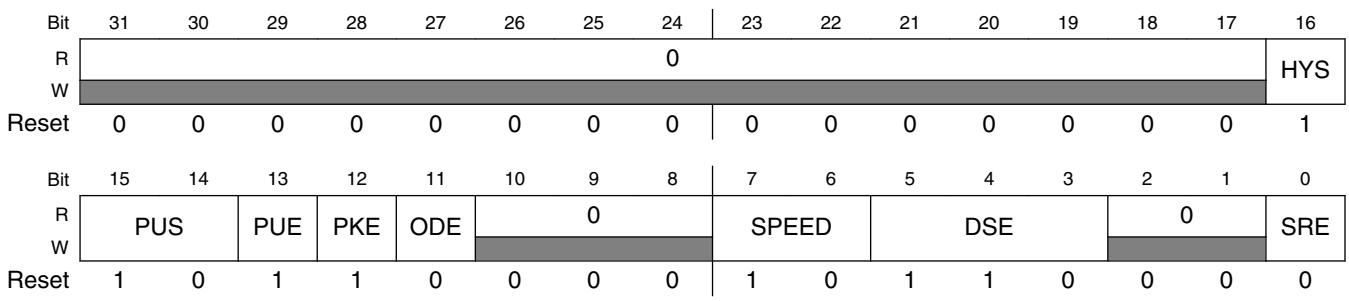
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN02 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.235 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN03)

Address: 20E_0000h base + 3BCh offset = 20E_03BCh



IOMUXC_SW_PAD_CTL_PAD_DI0_PIN03 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DI0_PIN3. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DI0_PIN3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DI0_PIN3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DI0_PIN3.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN03 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DI0_PIN3. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.236 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN04)

Address: 20E_0000h base + 3C0h offset = 20E_03C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN04 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DI0_PIN4. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DI0_PIN4. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DI0_PIN4. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DI0_PIN4. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN04 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DI0_PIN4. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.237 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA00)

Address: 20E_0000h base + 3C4h offset = 20E_03C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA00 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT0. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA00 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.238 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA01)

Address: 20E_0000h base + 3C8h offset = 20E_03C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA01 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA01 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT1. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA01 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.239 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA10)

Address: 20E_0000h base + 3CCh offset = 20E_03CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA10 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT10. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT10. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT10. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT10.

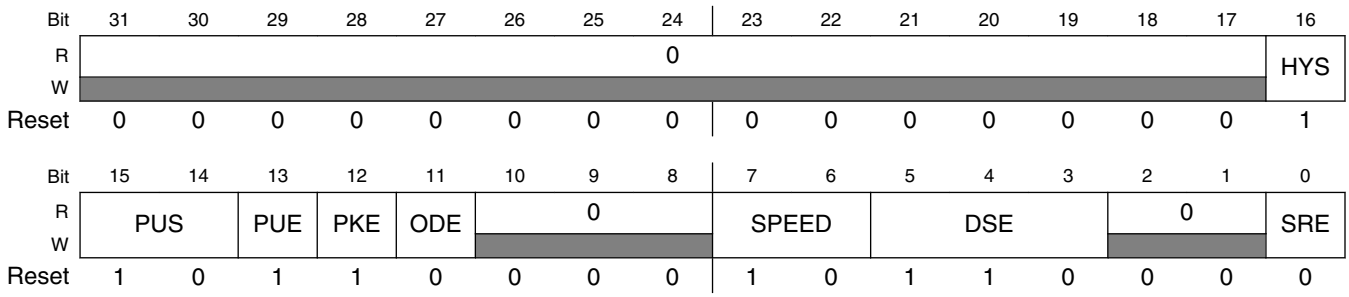
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA10 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT10. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.240 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA11)

Address: 20E_0000h base + 3D0h offset = 20E_03D0h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA11 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT11. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT11. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT11. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT11. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA11 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT11. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.241 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA12)

Address: 20E_0000h base + 3D4h offset = 20E_03D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA12 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT12. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT12. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT12. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT12. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT12. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA12 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.242 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA13)

Address: 20E_0000h base + 3D8h offset = 20E_03D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA13 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT13. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT13. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA13 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT13. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT13. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT13. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

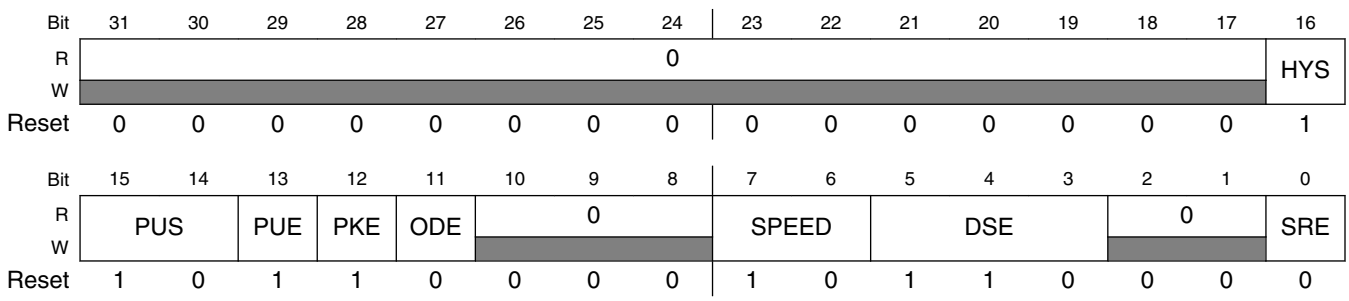
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA13 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.243 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA14)

Address: 20E_0000h base + 3DCh offset = 20E_03DCh



IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA14 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT14. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT14. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT14. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT14.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA14 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT14. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.244 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA15)

Address: 20E_0000h base + 3E0h offset = 20E_03E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA15 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT15. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT15. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT15. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT15. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA15 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT15. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.245 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA16)

Address: 20E_0000h base + 3E4h offset = 20E_03E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA16 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT16. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT16. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT16. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT16. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT16. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA16 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.246 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA17)

Address: 20E_0000h base + 3E8h offset = 20E_03E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA17 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT17. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT17. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA17 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT17. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT17. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT17. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA17 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.247 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA18)

Address: 20E_0000h base + 3ECh offset = 20E_03ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA18 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT18. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT18. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT18. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT18.

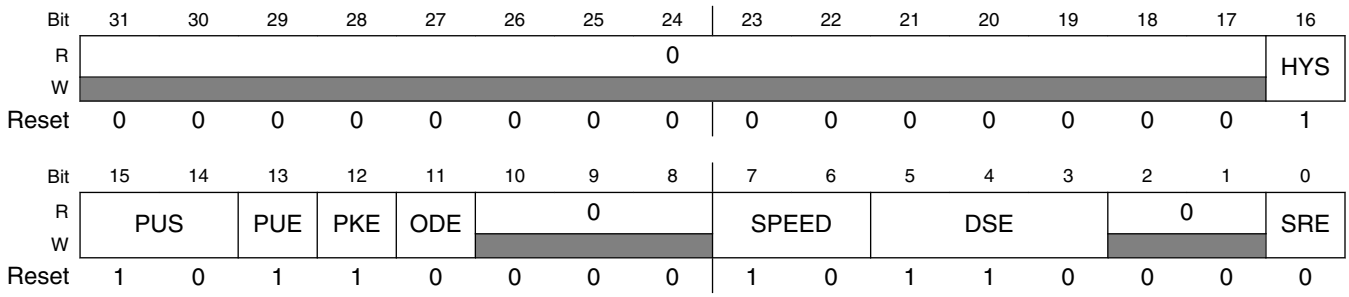
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA18 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT18. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.248 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA19)

Address: 20E_0000h base + 3F0h offset = 20E_03F0h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA19 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT19. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT19. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT19. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT19. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA19 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT19. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.249 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA02)

Address: 20E_0000h base + 3F4h offset = 20E_03F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA02 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT2. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT2. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT2. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA02 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.250 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA20)

Address: 20E_0000h base + 3F8h offset = 20E_03F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA20 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT20. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT20. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA20 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT20. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT20. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT20. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

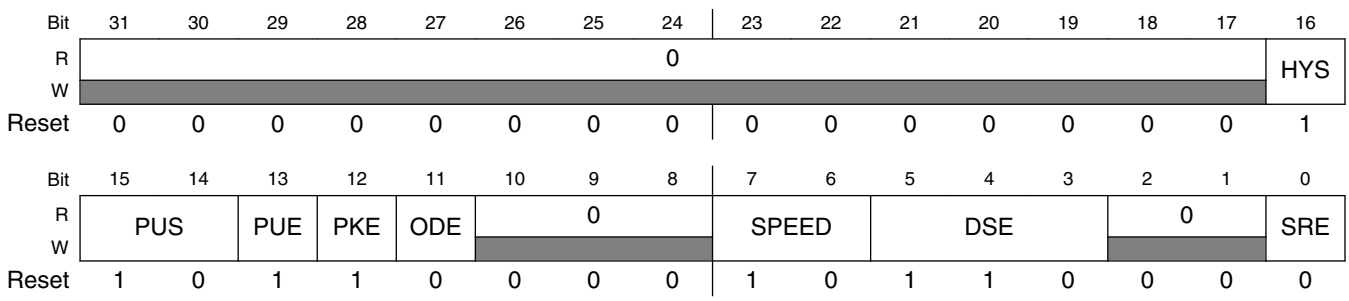
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA20 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.251 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA21)

Address: 20E_0000h base + 3FCh offset = 20E_03FCh



IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA21 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT21. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT21. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT21. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT21.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA21 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT21. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.252 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA22)

Address: 20E_0000h base + 400h offset = 20E_0400h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA22 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT22. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT22. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT22. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT22. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA22 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT22. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.253 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA23)

Address: 20E_0000h base + 404h offset = 20E_0404h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA23 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT23. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT23. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT23. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT23. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT23. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA23 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.254 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA03)

Address: 20E_0000h base + 408h offset = 20E_0408h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA03 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT3. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA03 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT3. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT3. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA03 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.255 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA04)

Address: 20E_0000h base + 40Ch offset = 20E_040Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA04 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT4. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT4. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT4. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT4.

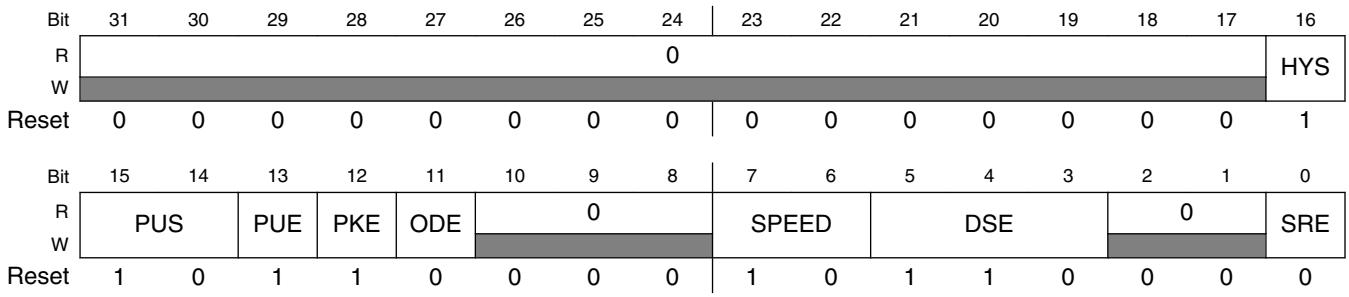
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA04 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT4. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.256 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA05)

Address: 20E_0000h base + 410h offset = 20E_0410h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA05 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT5. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT5. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT5. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT5. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA05 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT5. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.257 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA06)

Address: 20E_0000h base + 414h offset = 20E_0414h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA06 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT6. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT6. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT6. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT6. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT6. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA06 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.258 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA07)

Address: 20E_0000h base + 418h offset = 20E_0418h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA07 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT7. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT7. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA07 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT7. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT7. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT7. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

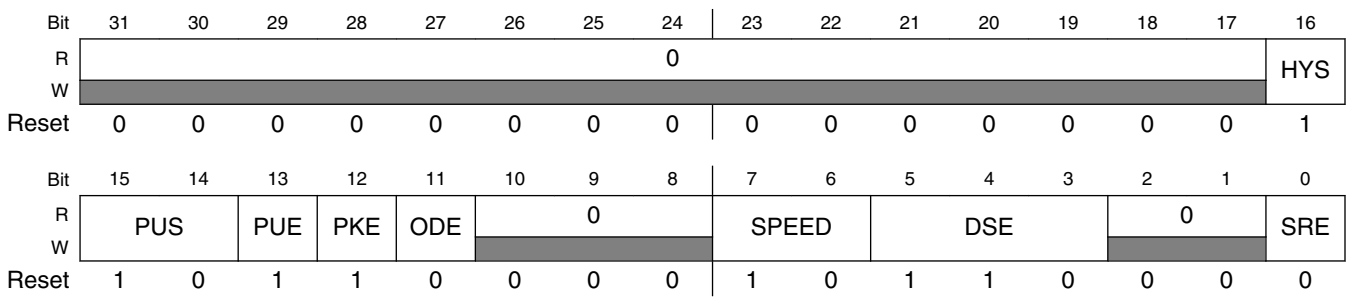
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA07 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.259 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA08)

Address: 20E_0000h base + 41Ch offset = 20E_041Ch



IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA08 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT8. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT8. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT8. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT8.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA08 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT8. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.260 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA09)

Address: 20E_0000h base + 420h offset = 20E_0420h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA09 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DISP0_DAT9. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DISP0_DAT9. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DISP0_DAT9. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DISP0_DAT9. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

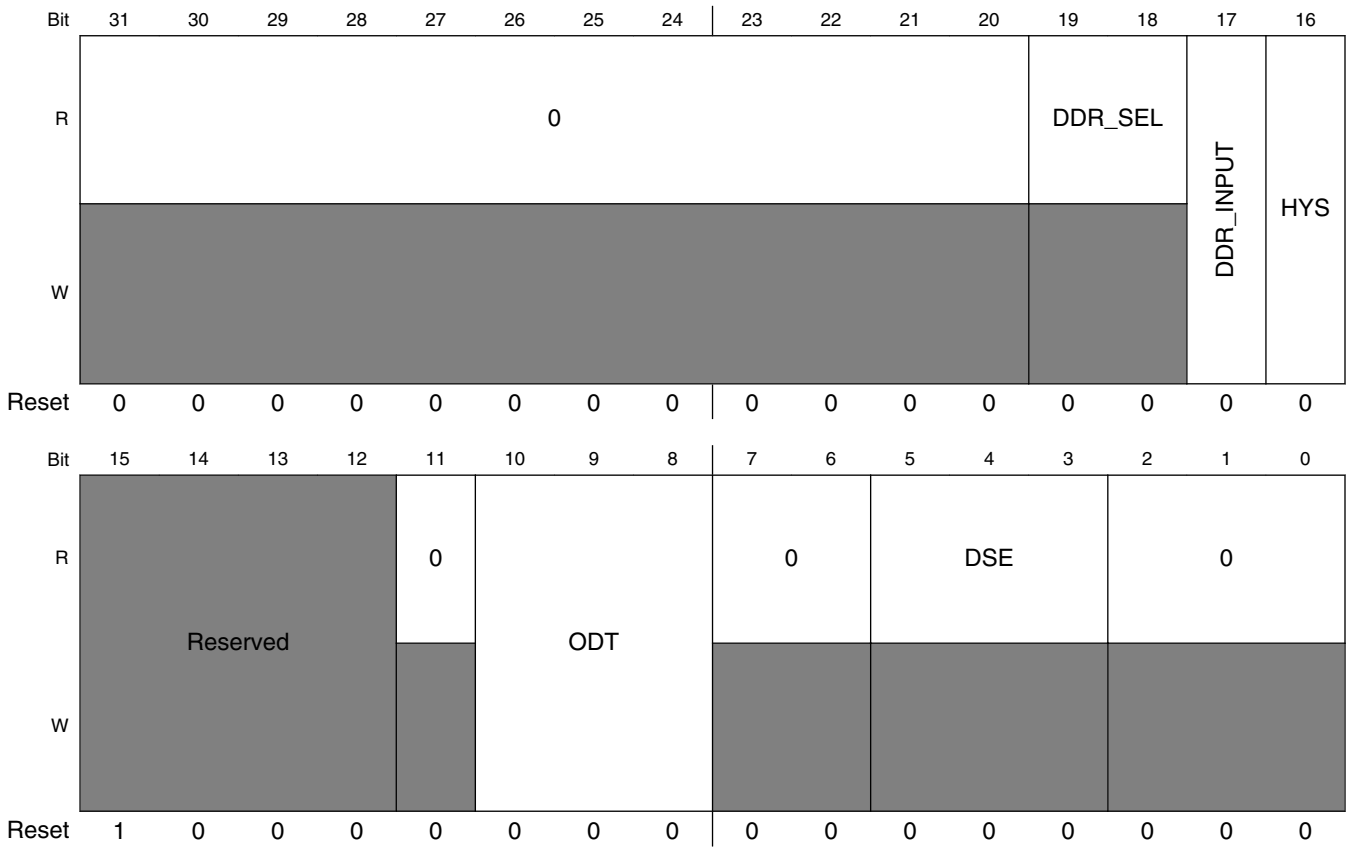
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DATA09 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: DISP0_DAT9. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.261 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR00)

Address: 20E_0000h base + 424h offset = 20E_0424h



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR00 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A0. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A0.

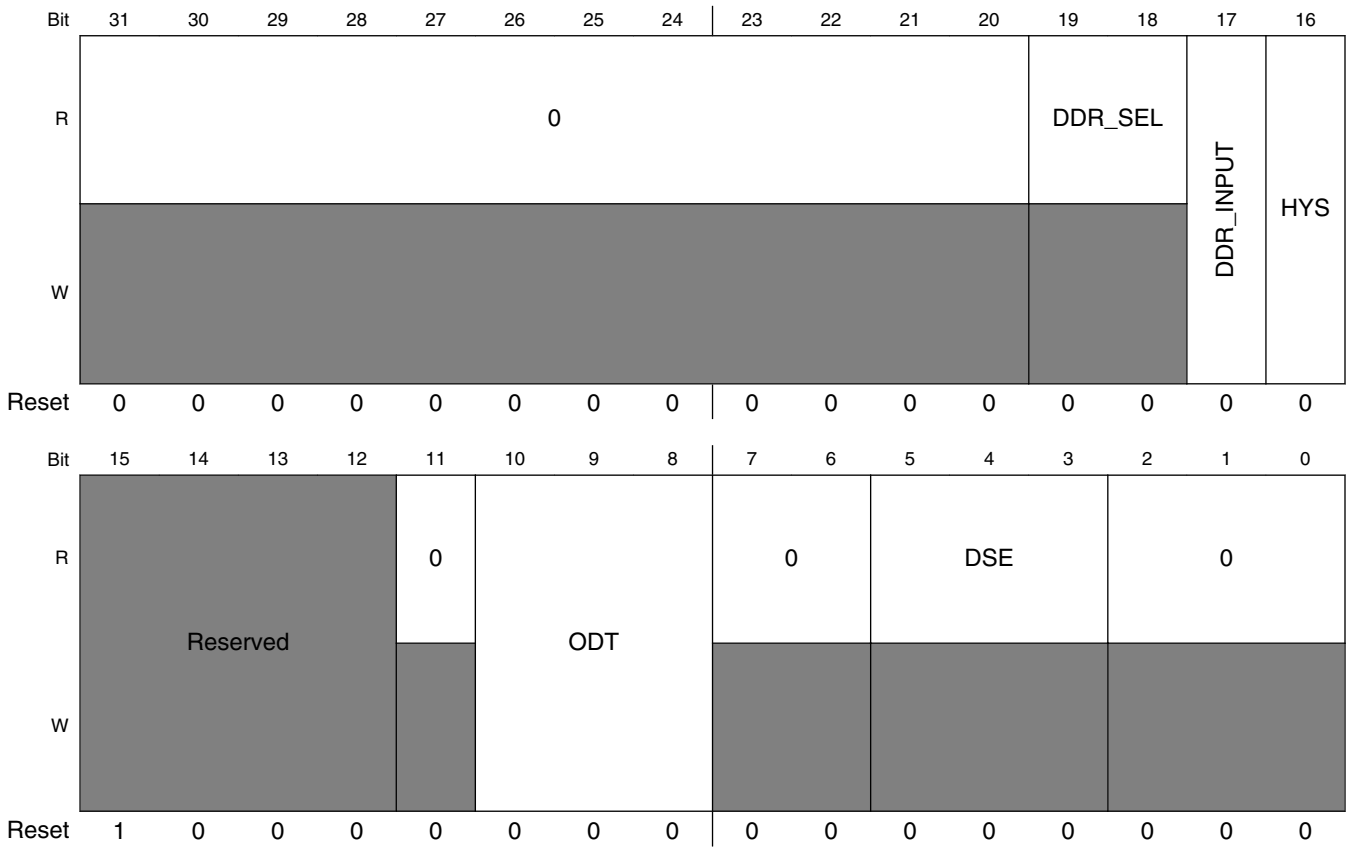
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR00 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A0. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.262 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR01)

Address: 20E_0000h base + 428h offset = 20E_0428h



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR01 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A1. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A1.

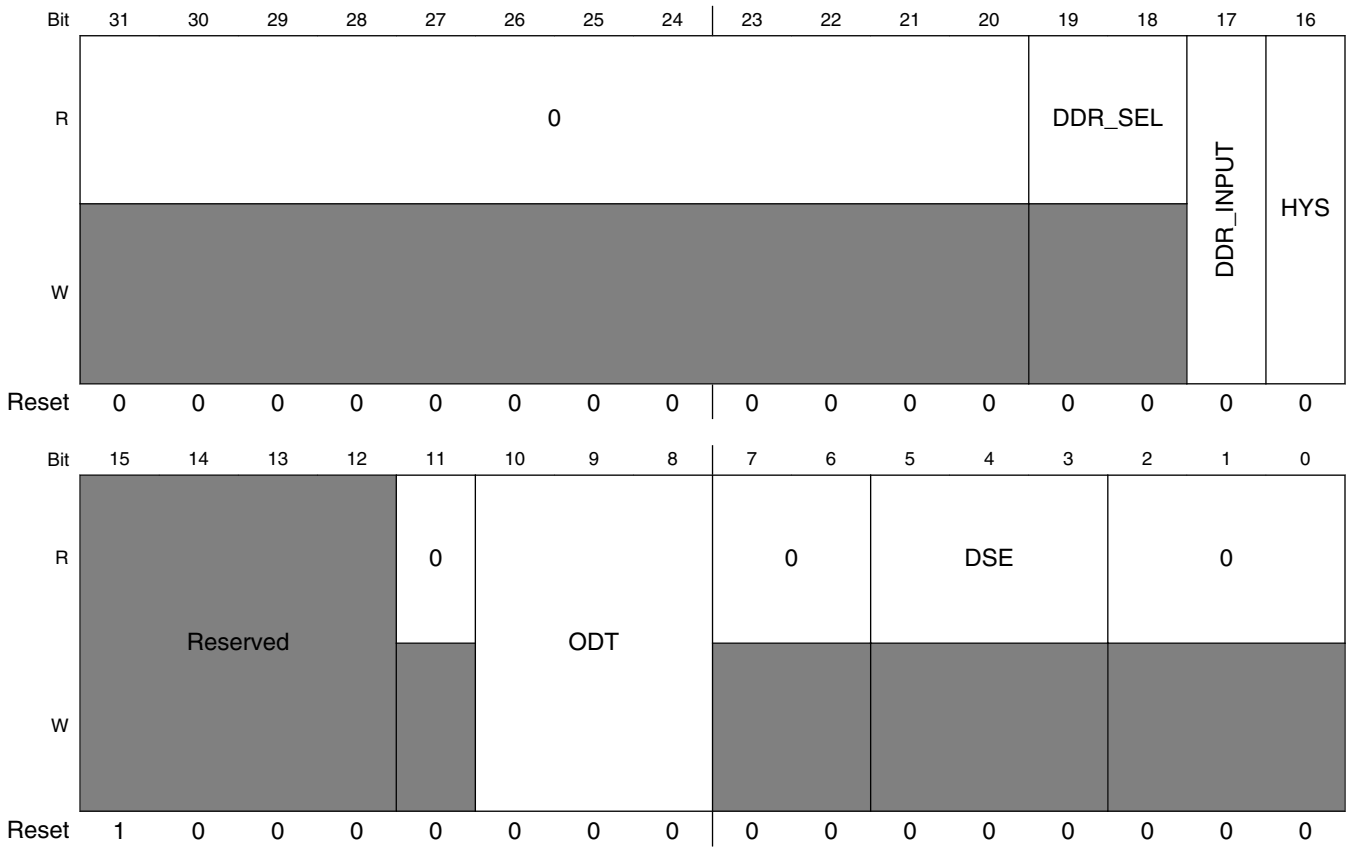
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR01 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A1. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.263 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR10)

Address: 20E_0000h base + 42Ch offset = 20E_042Ch



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR10 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A10. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A10.

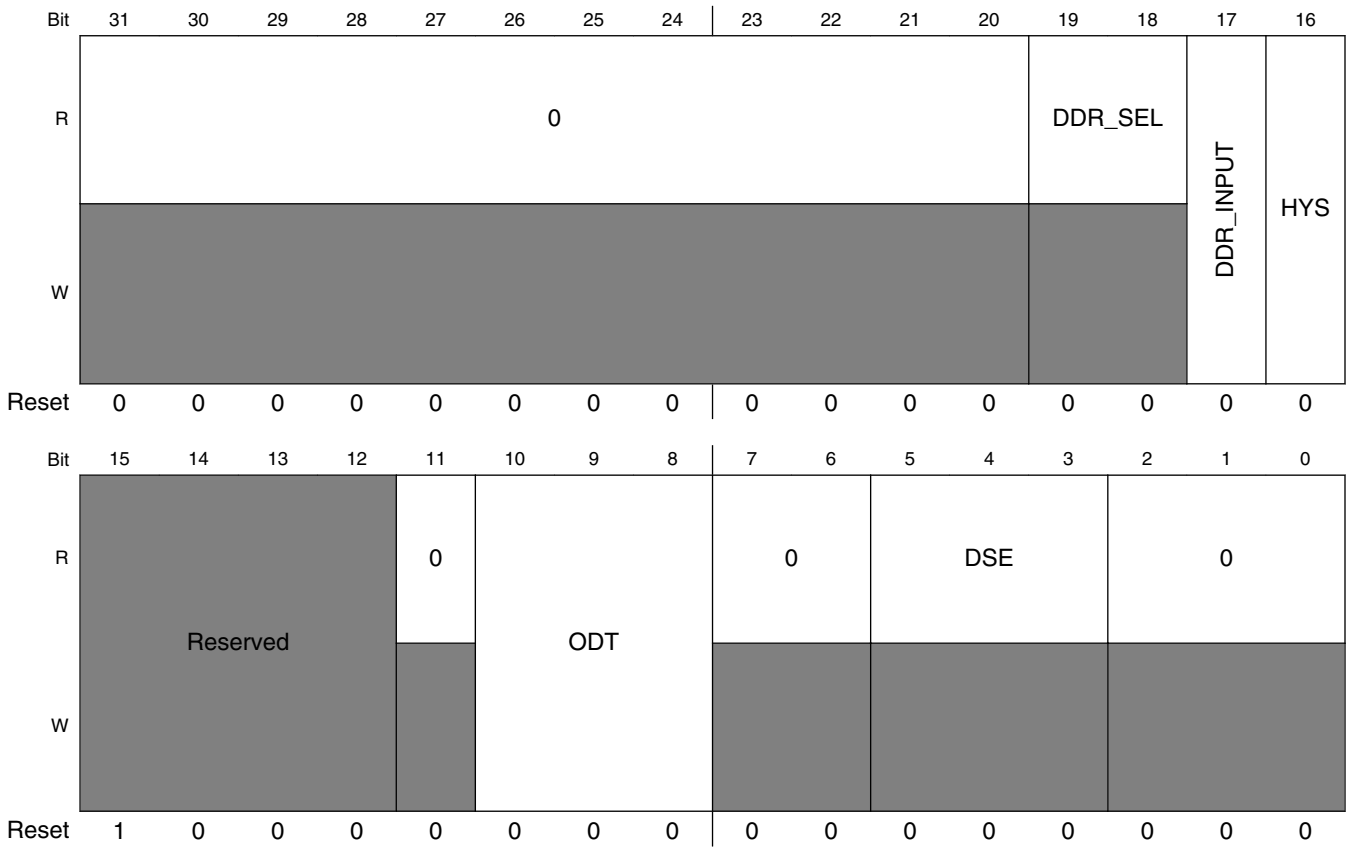
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR10 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A10. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.264 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR11)

Address: 20E_0000h base + 430h offset = 20E_0430h



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR11 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A11. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A11.

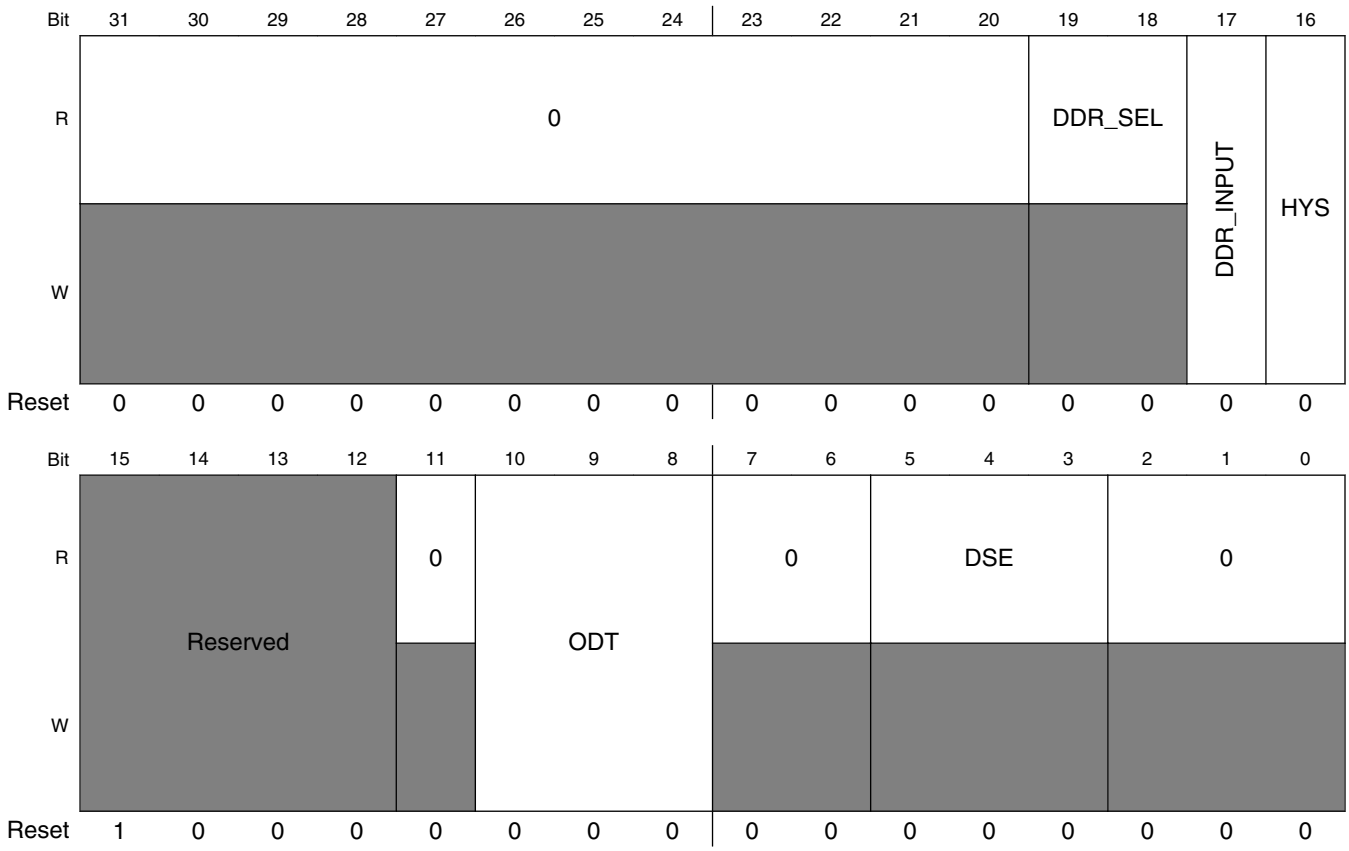
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR11 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A11. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.265 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR12)

Address: 20E_0000h base + 434h offset = 20E_0434h



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR12 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A12. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A12.

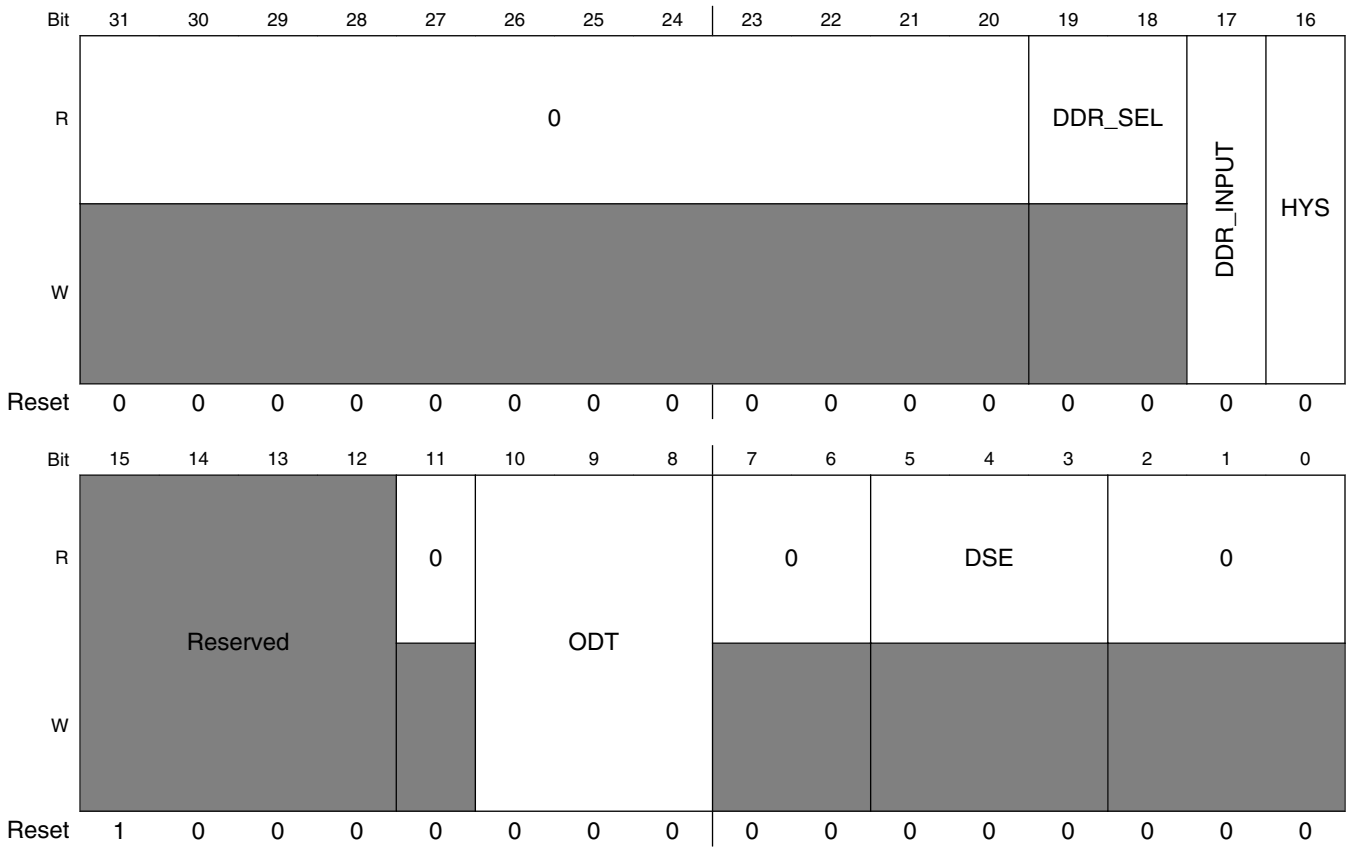
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR12 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A12. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.266 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR13)

Address: 20E_0000h base + 438h offset = 20E_0438h



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR13 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A13. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A13.

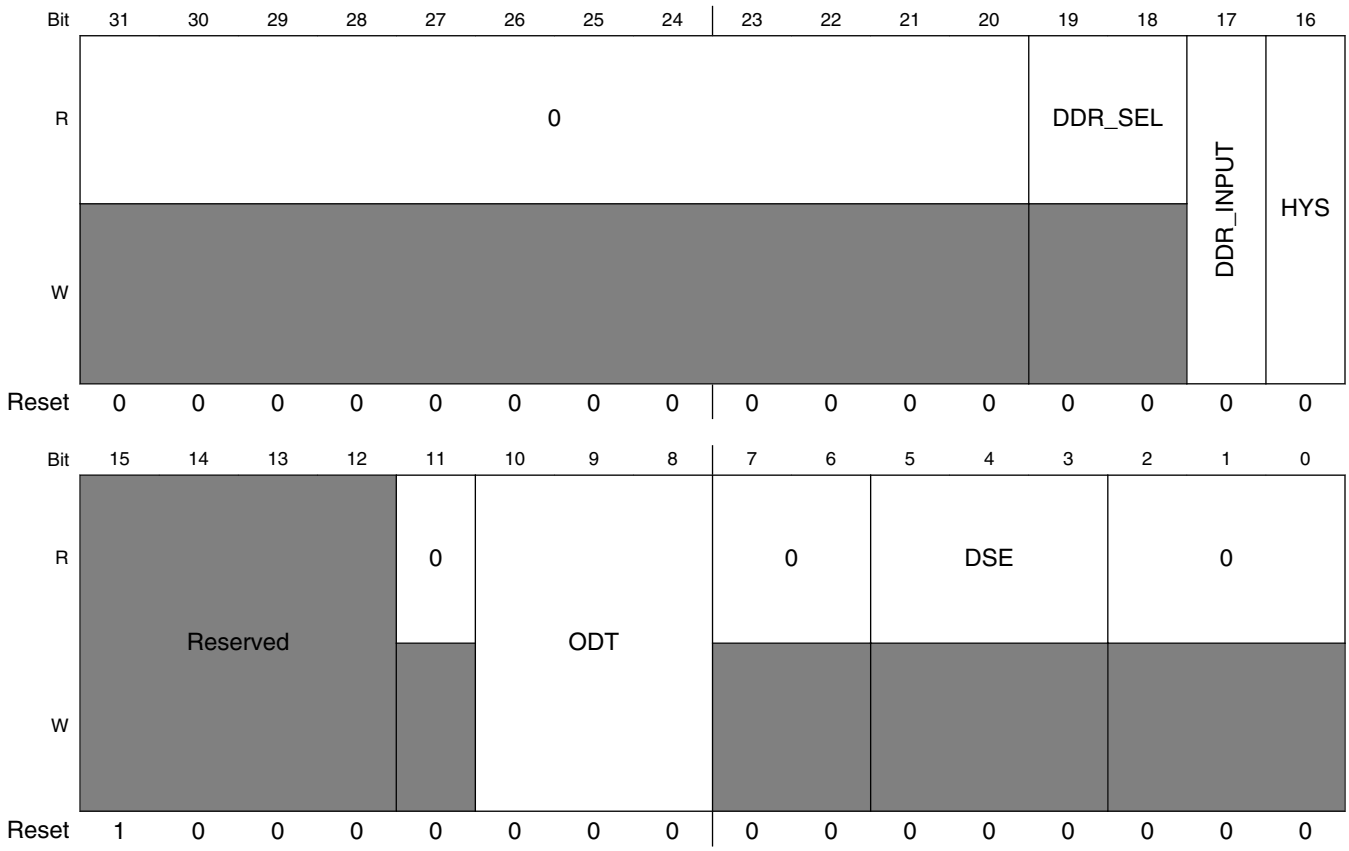
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR13 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A13. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.267 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR14)

Address: 20E_0000h base + 43Ch offset = 20E_043Ch



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR14 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A14. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A14.

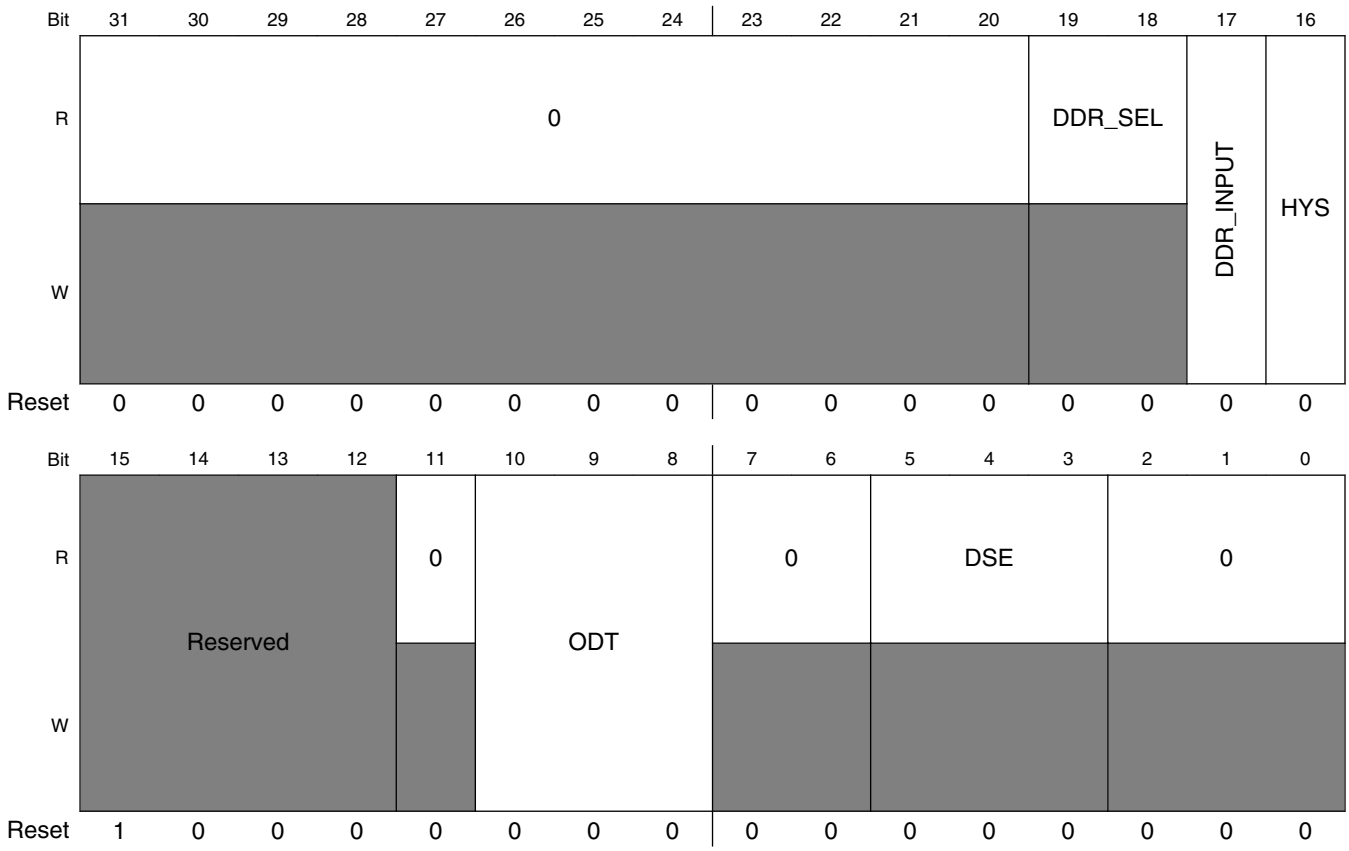
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR14 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A14. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.268 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR15)

Address: 20E_0000h base + 440h offset = 20E_0440h



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR15 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A15. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A15.

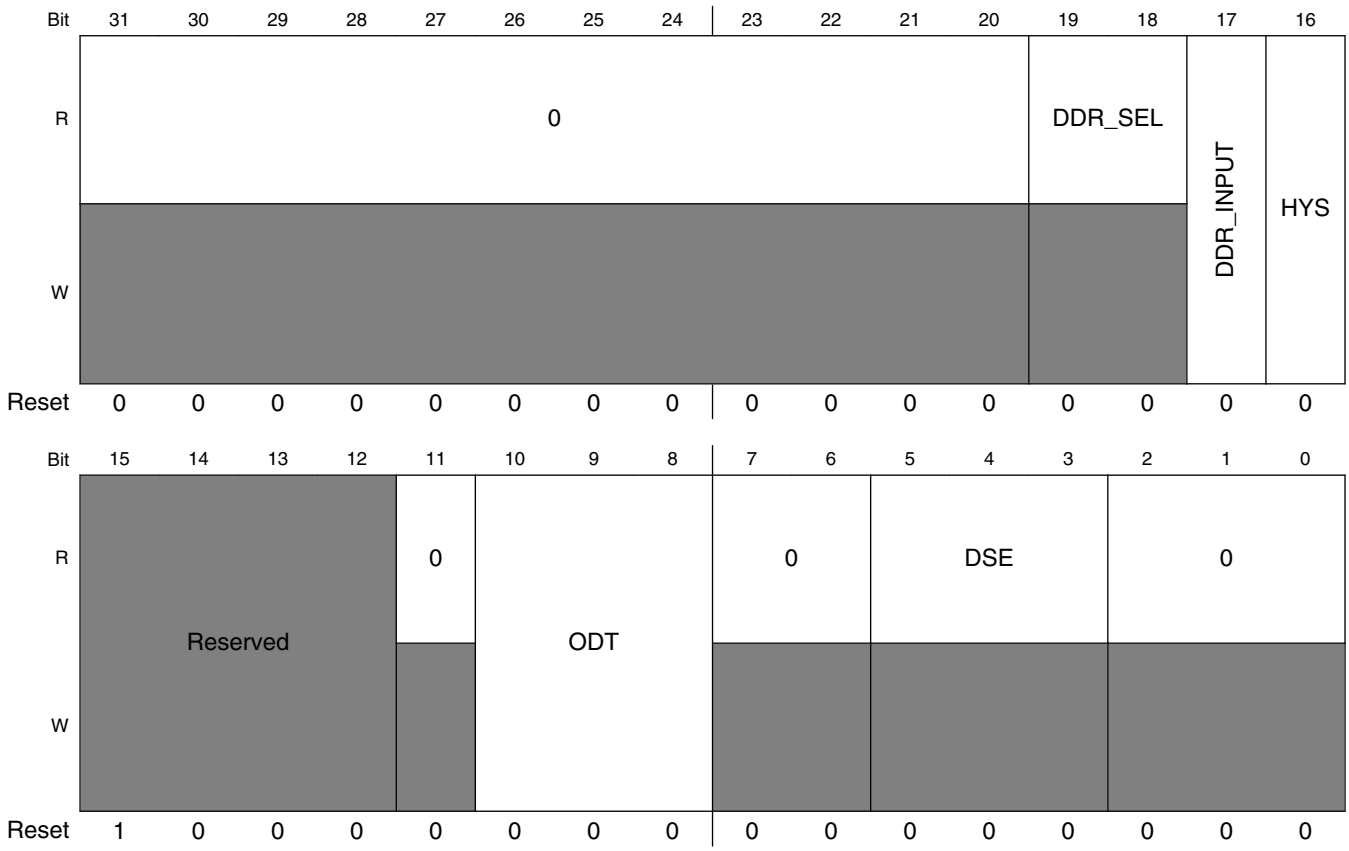
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR15 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A15. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.269 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR02)

Address: 20E_0000h base + 444h offset = 20E_0444h



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR02 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A2. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A2.

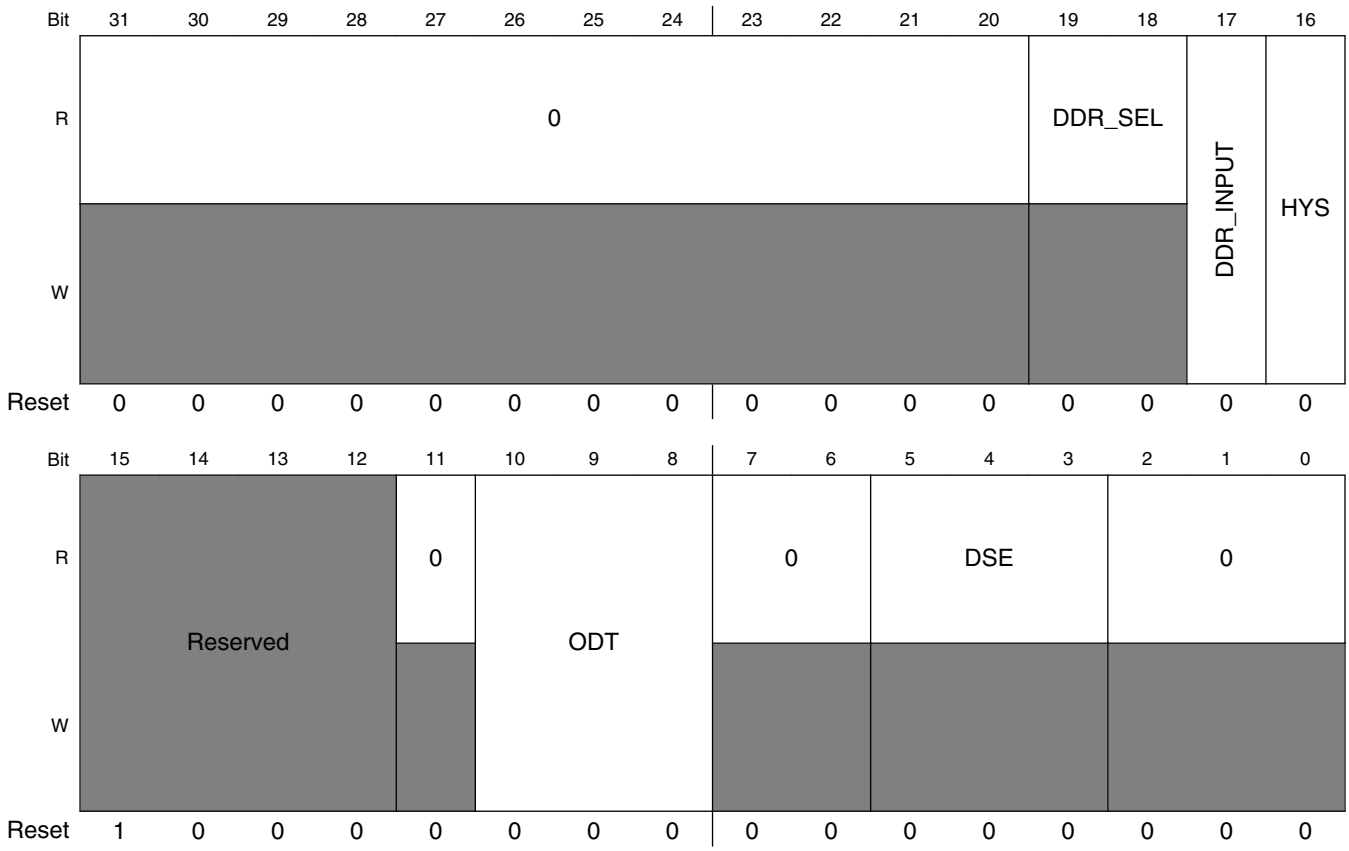
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR02 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A2. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.270 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR03)

Address: 20E_0000h base + 448h offset = 20E_0448h



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR03 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A3. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A3.

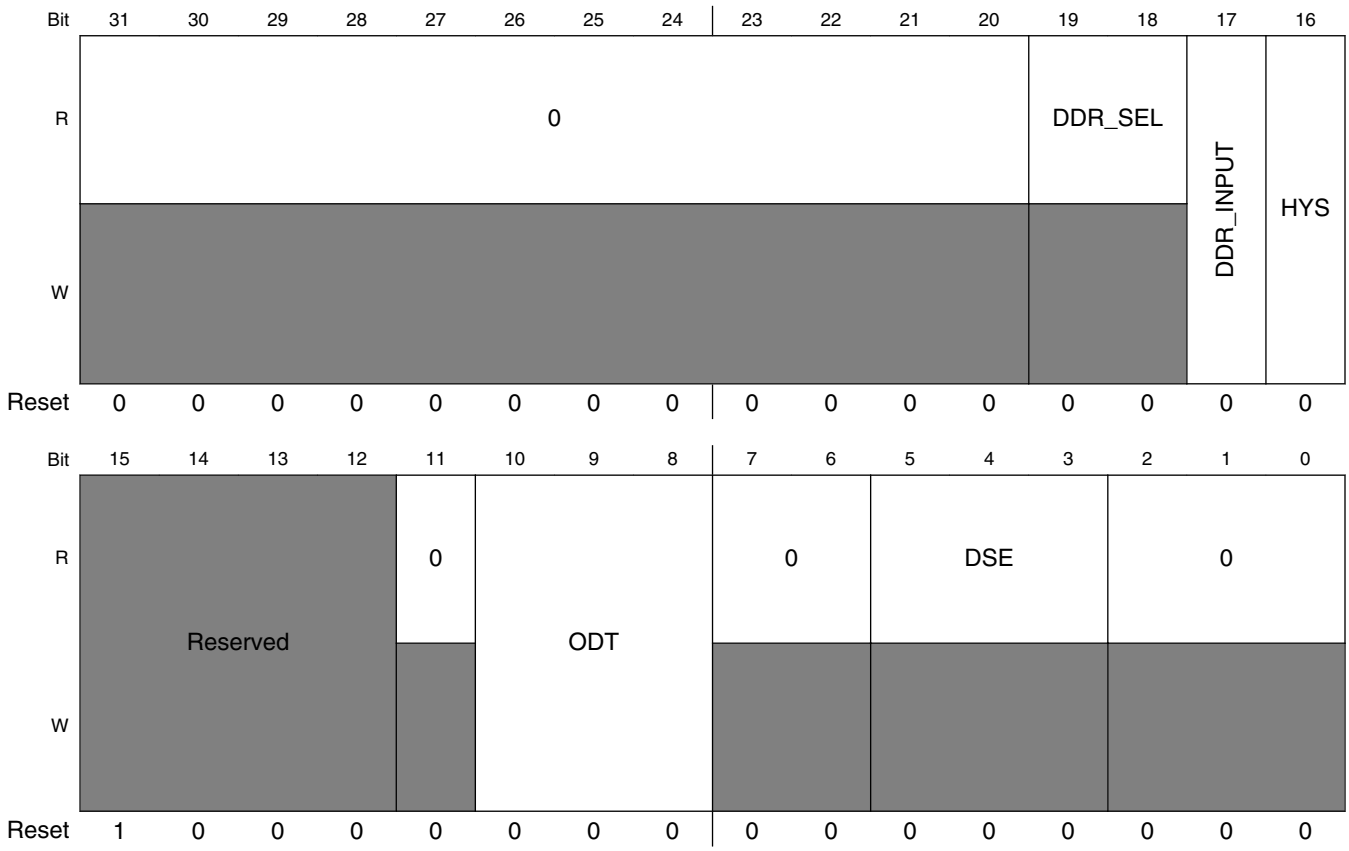
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR03 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A3. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.271 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR04)

Address: 20E_0000h base + 44Ch offset = 20E_044Ch



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR04 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A4. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A4.

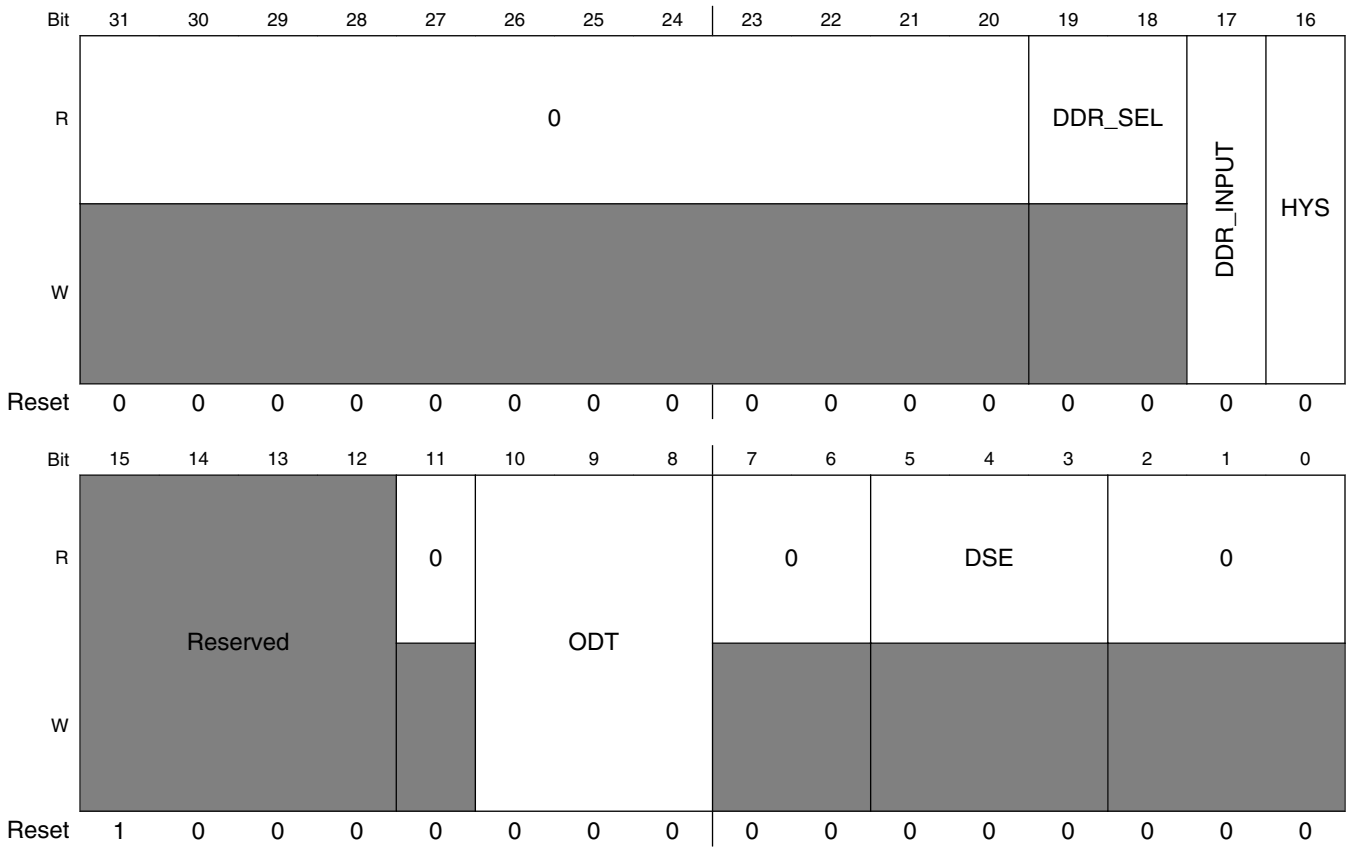
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR04 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A4. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.272 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR05)

Address: 20E_0000h base + 450h offset = 20E_0450h



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR05 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A5. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A5.

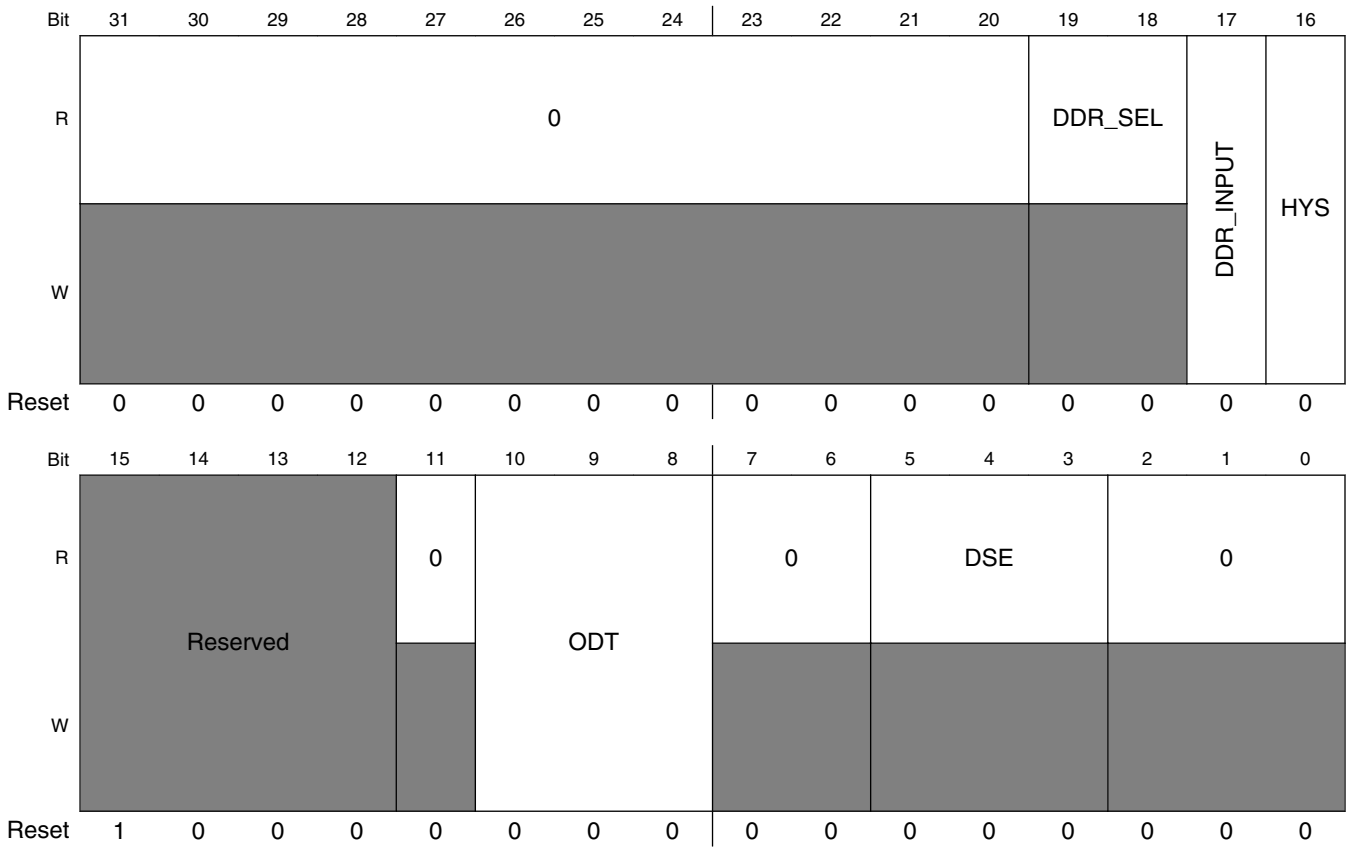
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR05 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A5. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.273 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR06)

Address: 20E_0000h base + 454h offset = 20E_0454h



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR06 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A6. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A6.

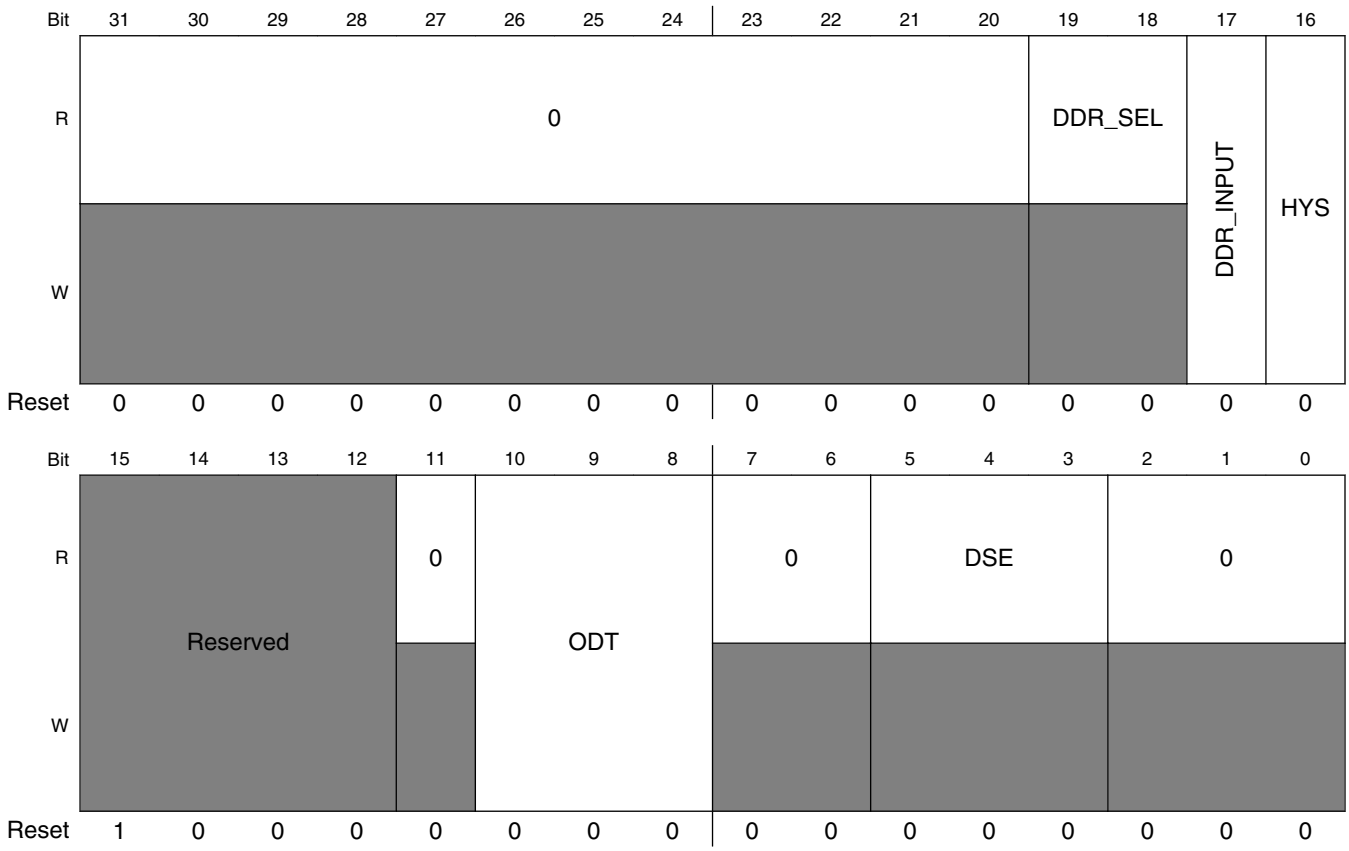
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR06 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A6. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.274 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR07)

Address: 20E_0000h base + 458h offset = 20E_0458h



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR07 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A7. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A7.

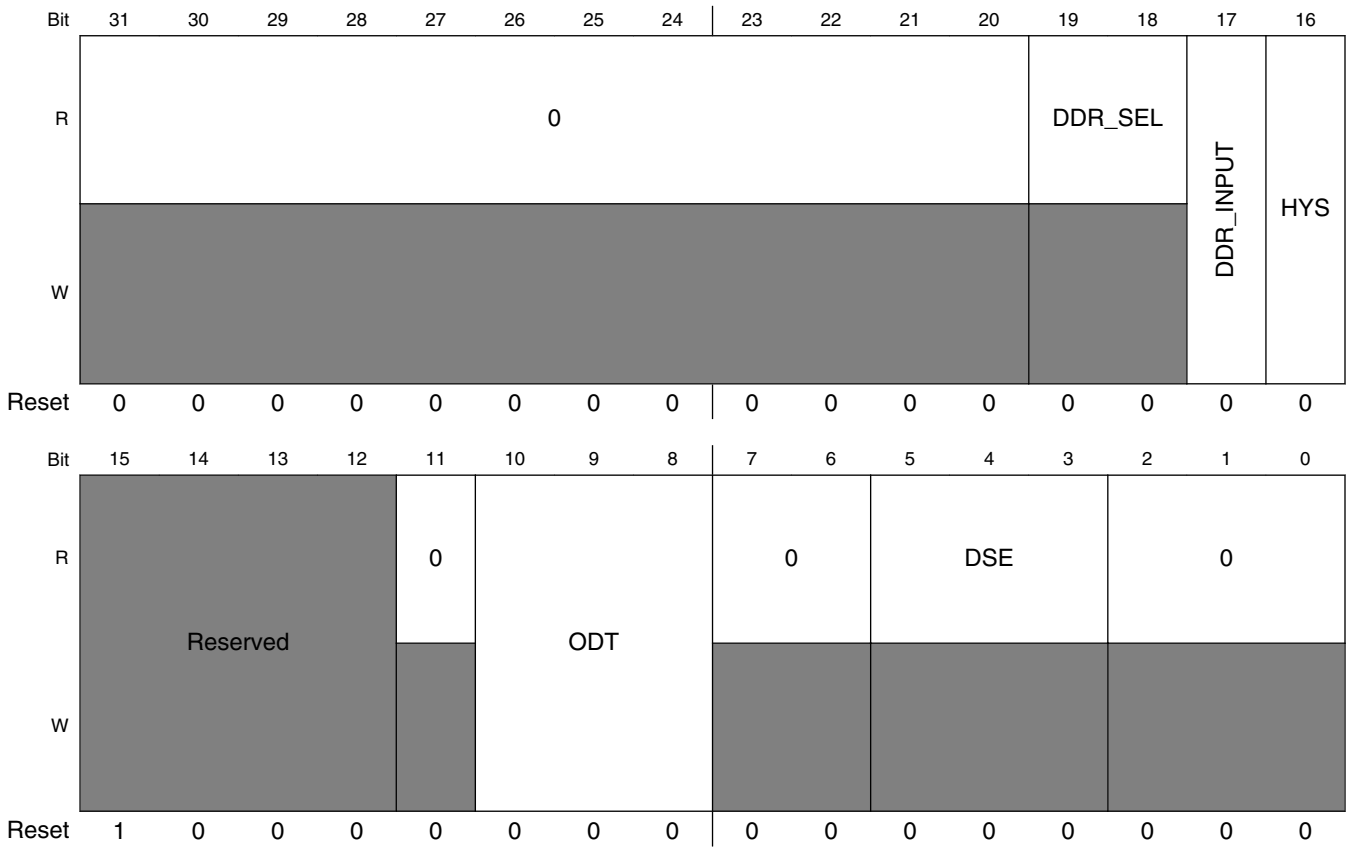
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR07 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A7. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.275 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR08)

Address: 20E_0000h base + 45Ch offset = 20E_045Ch



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR08 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A8. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A8.

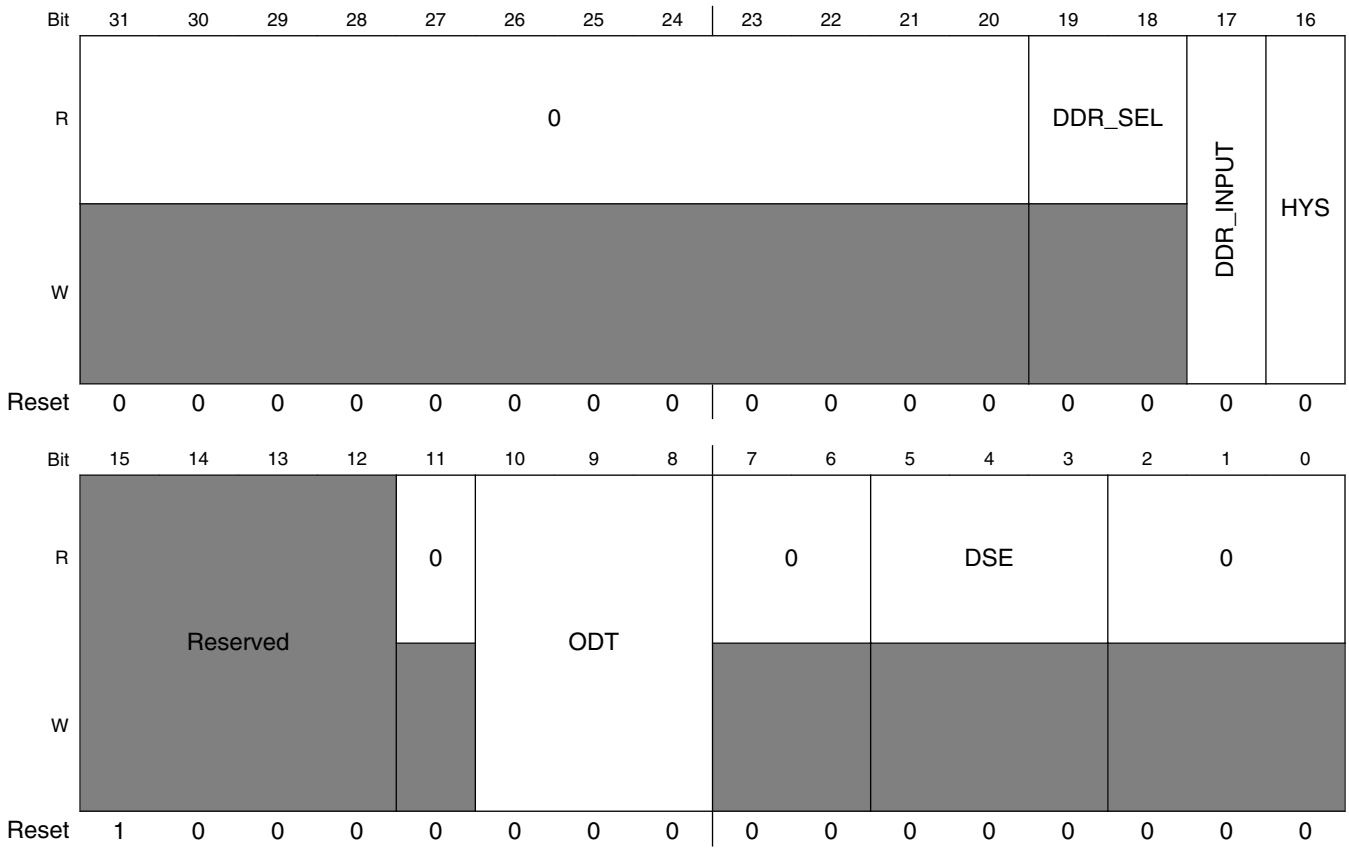
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR08 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A8. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.276 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR09)

Address: 20E_0000h base + 460h offset = 20E_0460h



IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR09 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_A9. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_A9.

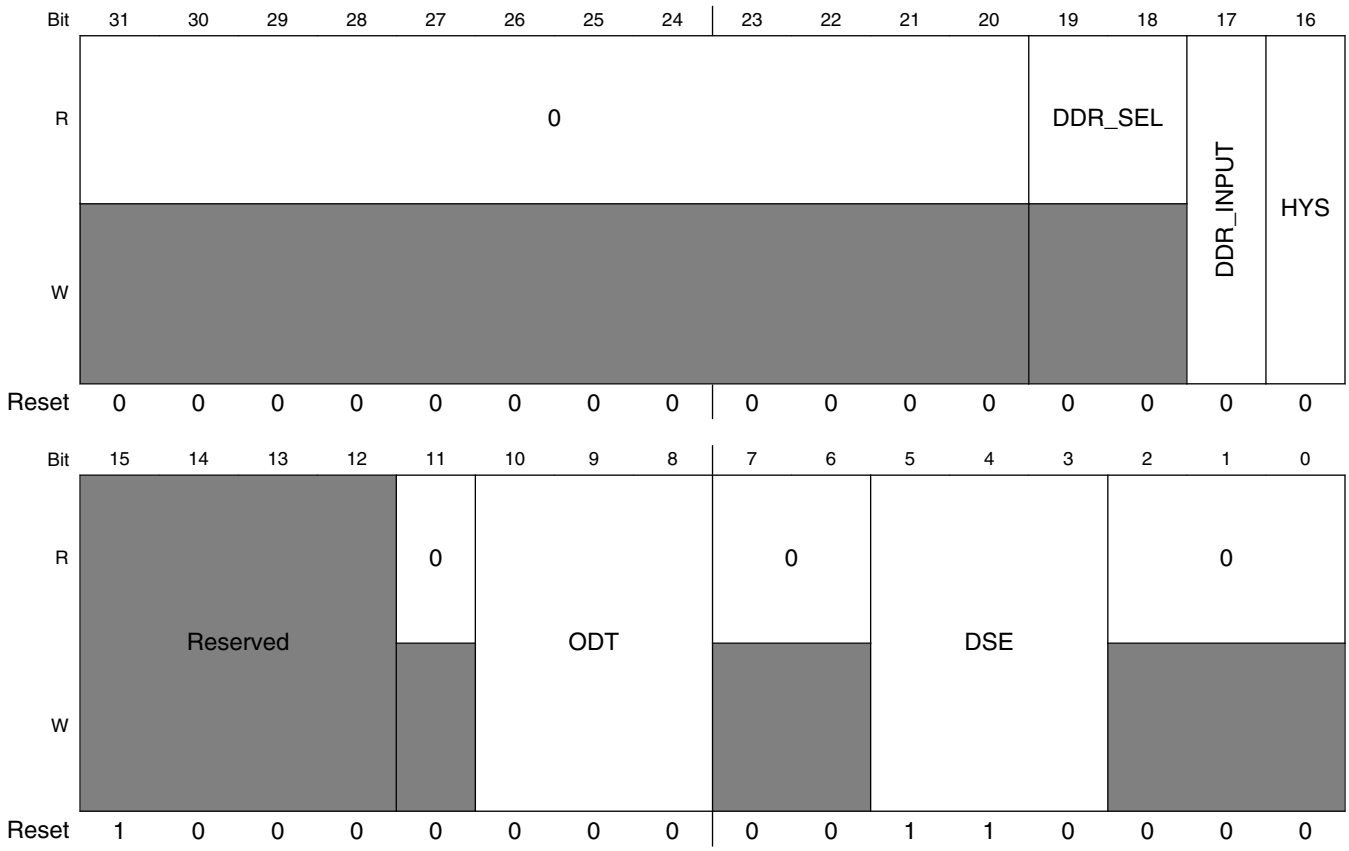
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR09 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_A9. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.277 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS_B)

Address: 20E_0000h base + 464h offset = 20E_0464h



IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS_B field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_CAS. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_CAS.

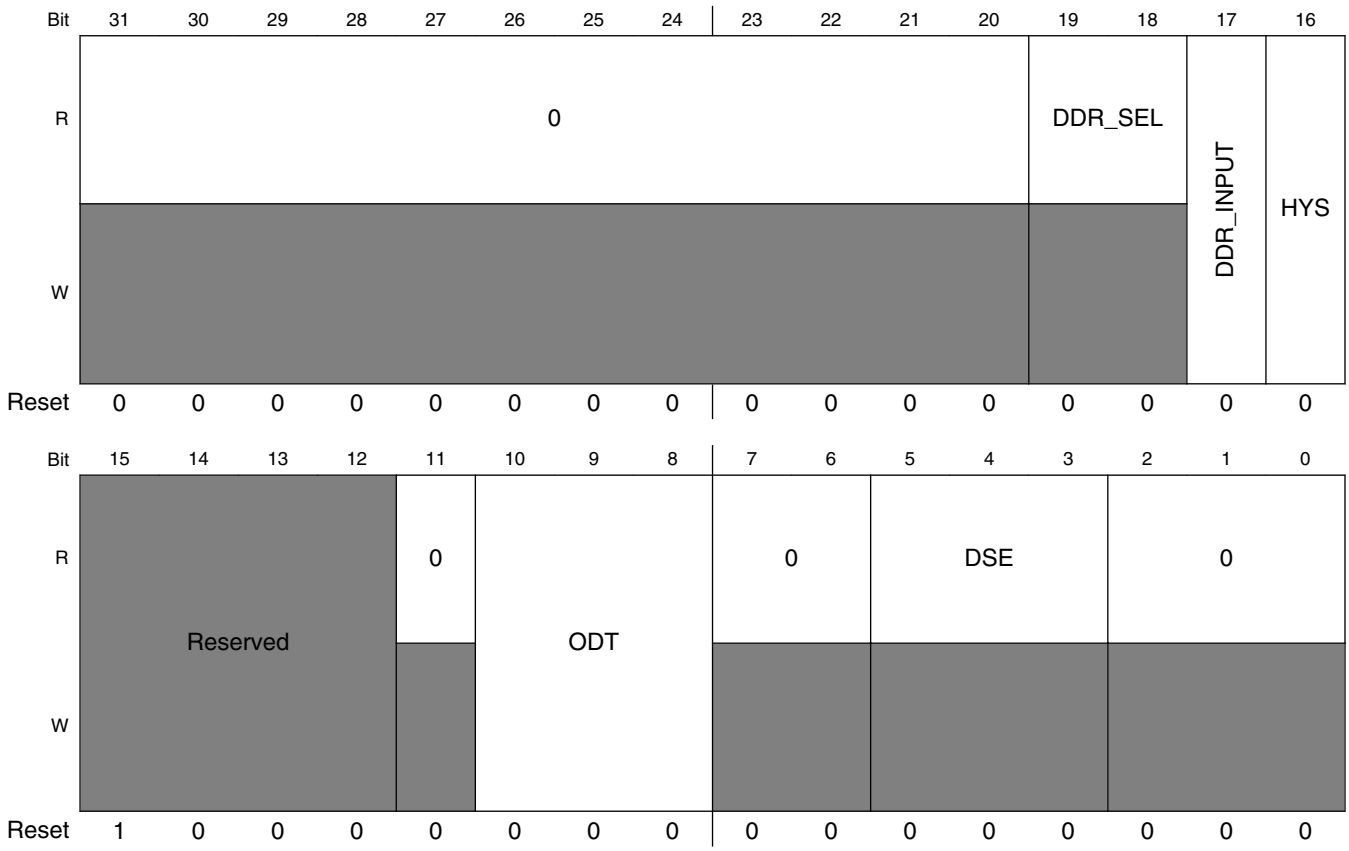
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS_B field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_CAS. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_CAS. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.278 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_CS0_B)

Address: 20E_0000h base + 468h offset = 20E_0468h



IOMUXC_SW_PAD_CTL_PAD_DRAM_CS0_B field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_CS0. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_CS0.

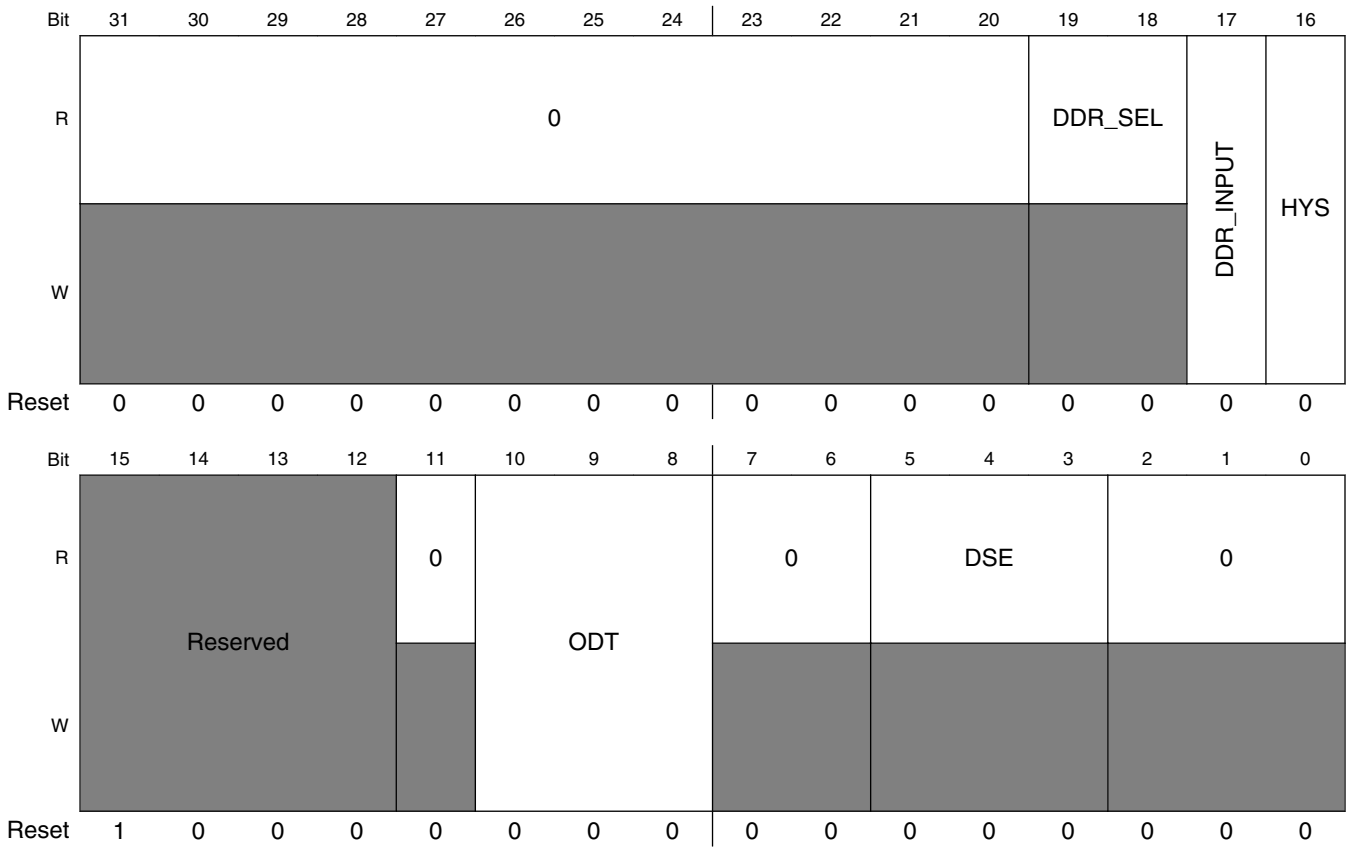
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_CS0_B field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_CS0. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_CTLDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.279 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_CS1_B)

Address: 20E_0000h base + 46Ch offset = 20E_046Ch



IOMUXC_SW_PAD_CTL_PAD_DRAM_CS1_B field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_CS1. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_CS1.

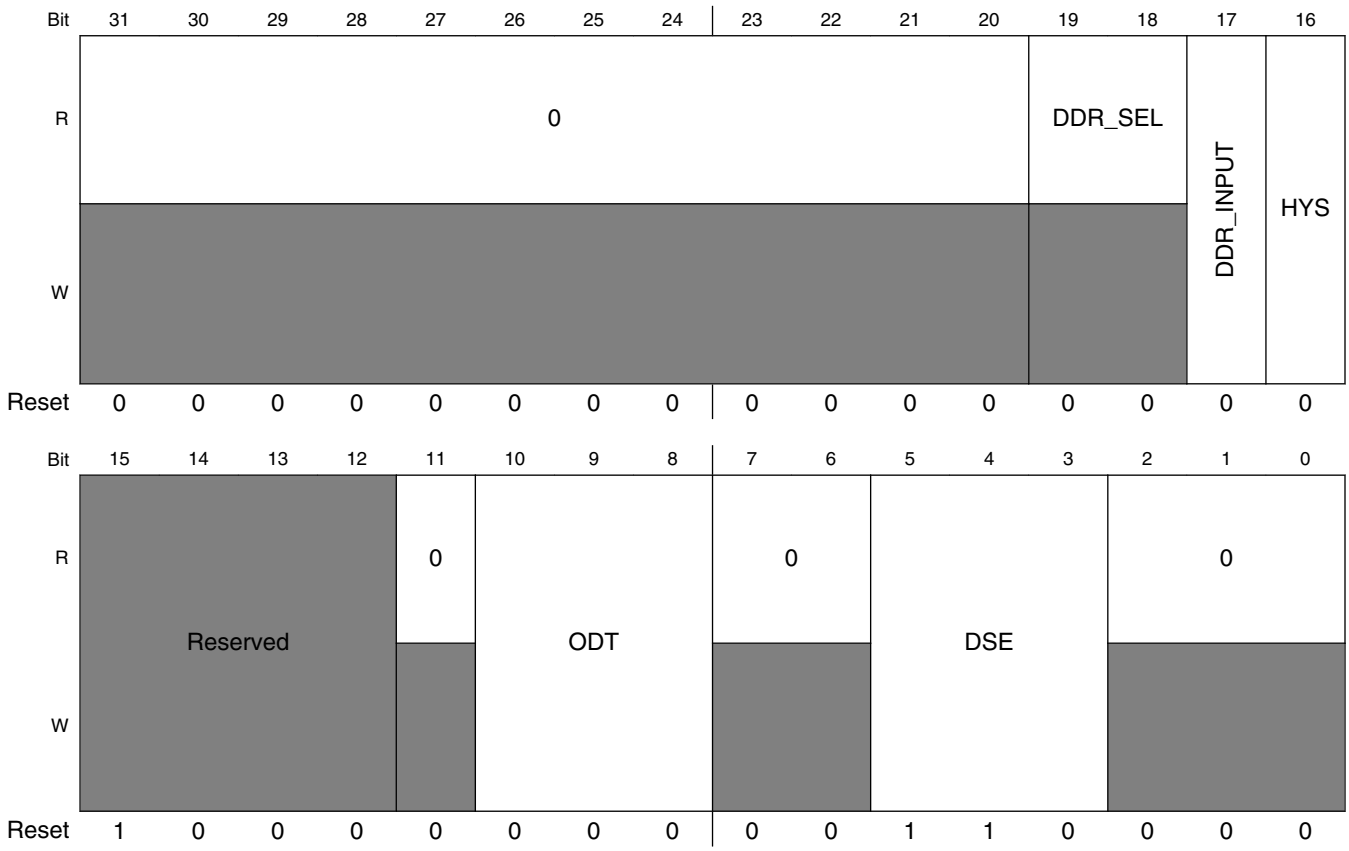
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_CS1_B field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_CS1. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_CTLDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.280 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0)

Address: 20E_0000h base + 470h offset = 20E_0470h



IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_DQM0. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_DQM0.

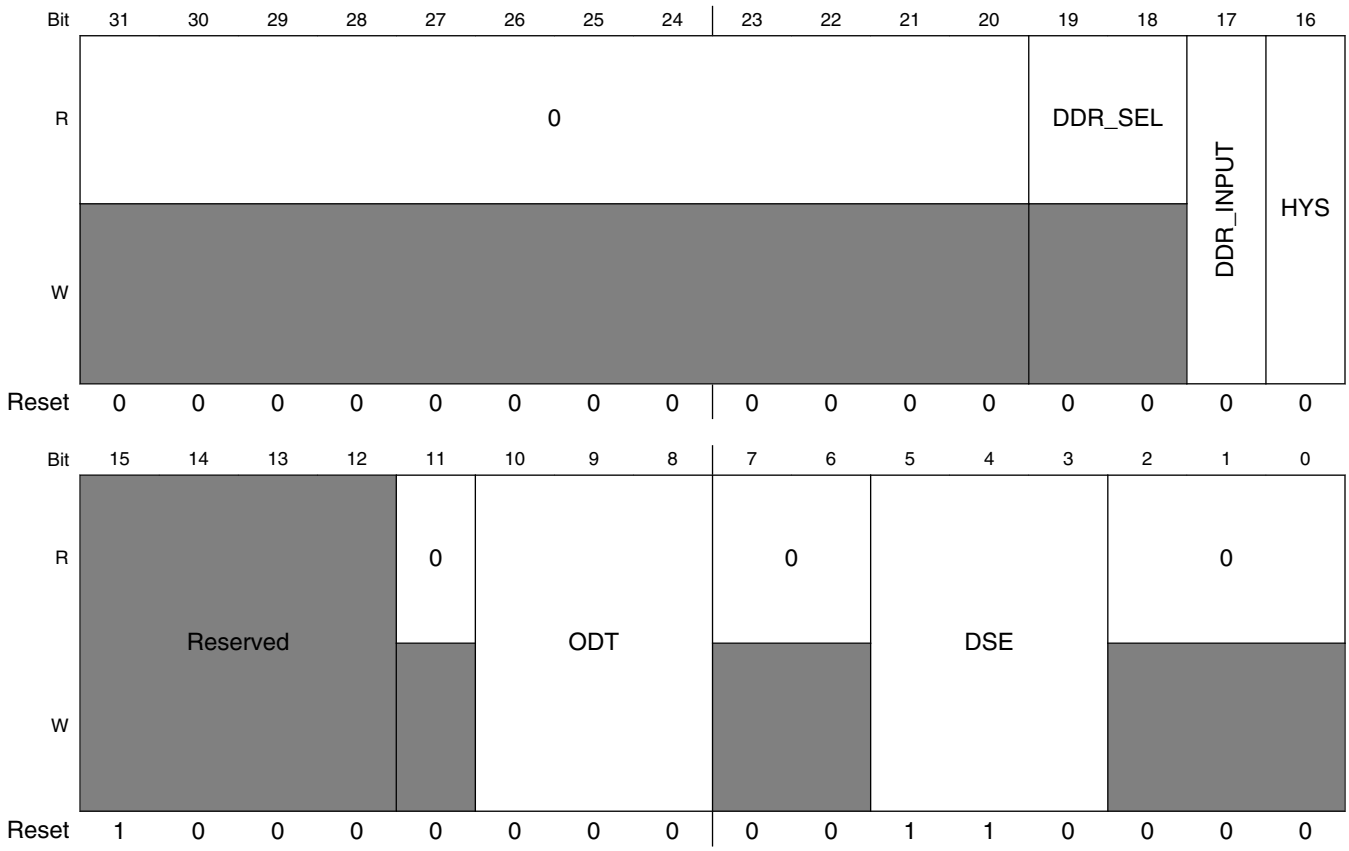
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_DQM0. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_DQM0. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.281 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1)

Address: 20E_0000h base + 474h offset = 20E_0474h



IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_DQM1. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_DQM1.

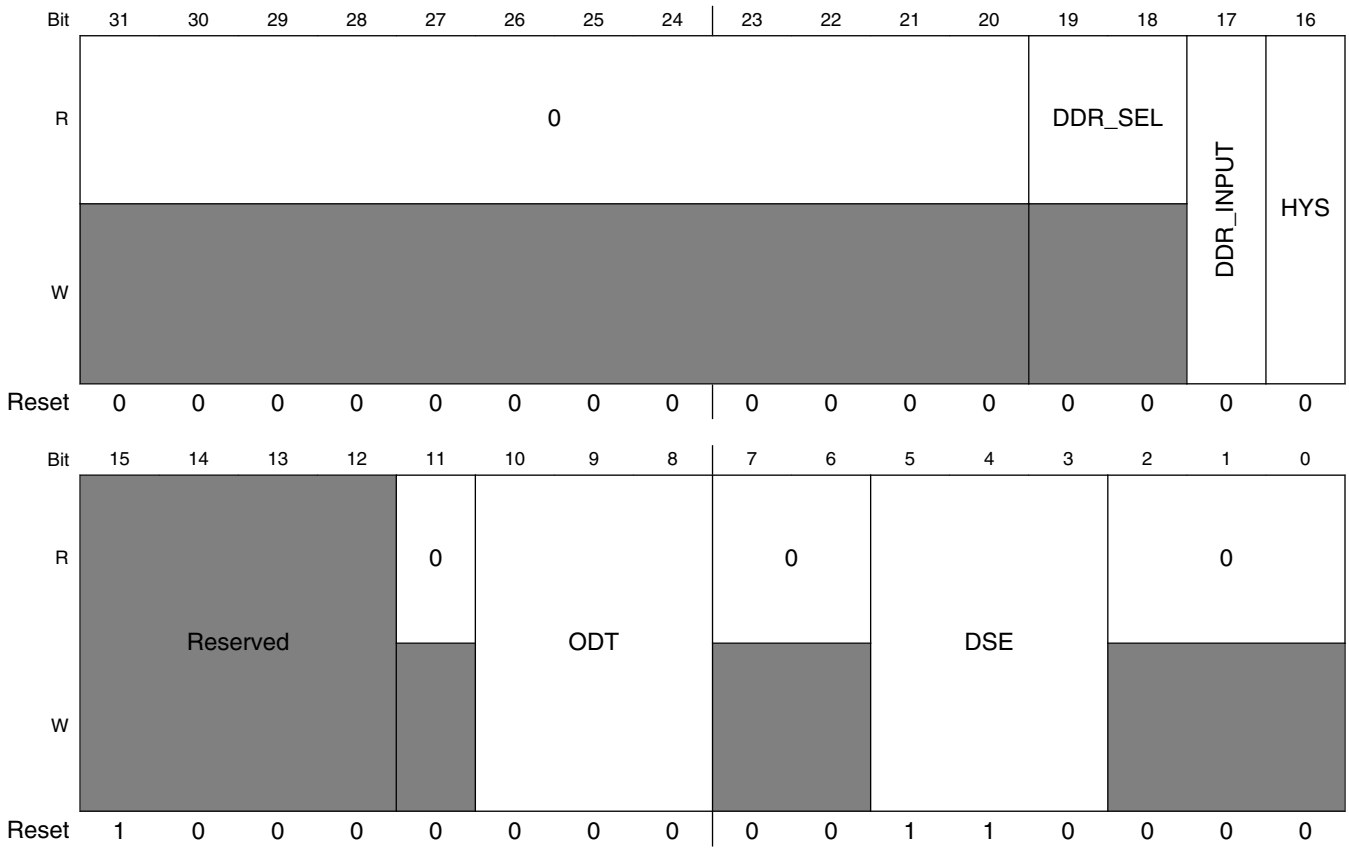
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_DQM1. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_DQM1. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.282 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2)

Address: 20E_0000h base + 478h offset = 20E_0478h



IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_DQM2. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_DQM2.

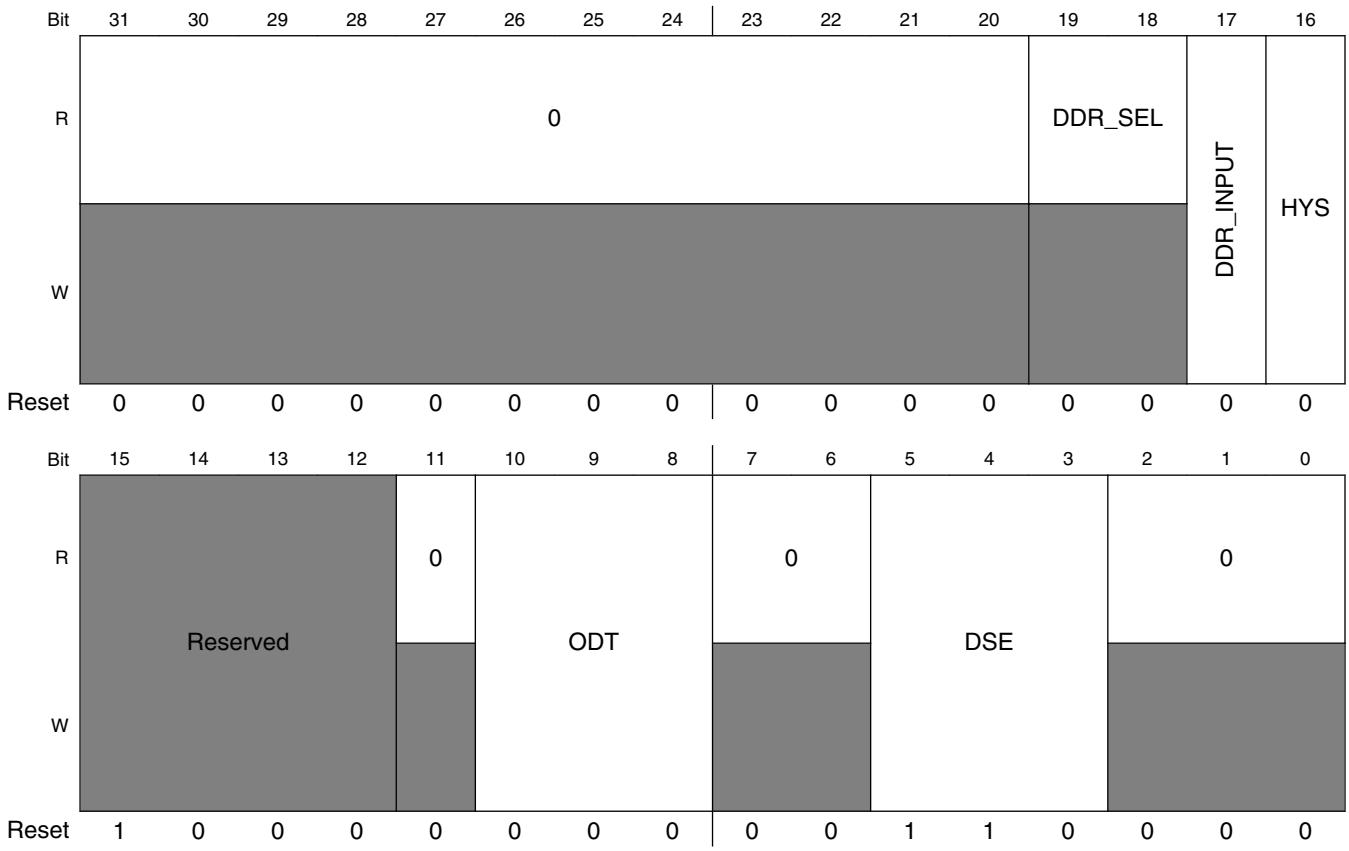
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_DQM2. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_DQM2. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.283 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3)

Address: 20E_0000h base + 47Ch offset = 20E_047Ch



IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_DQM3. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_DQM3.

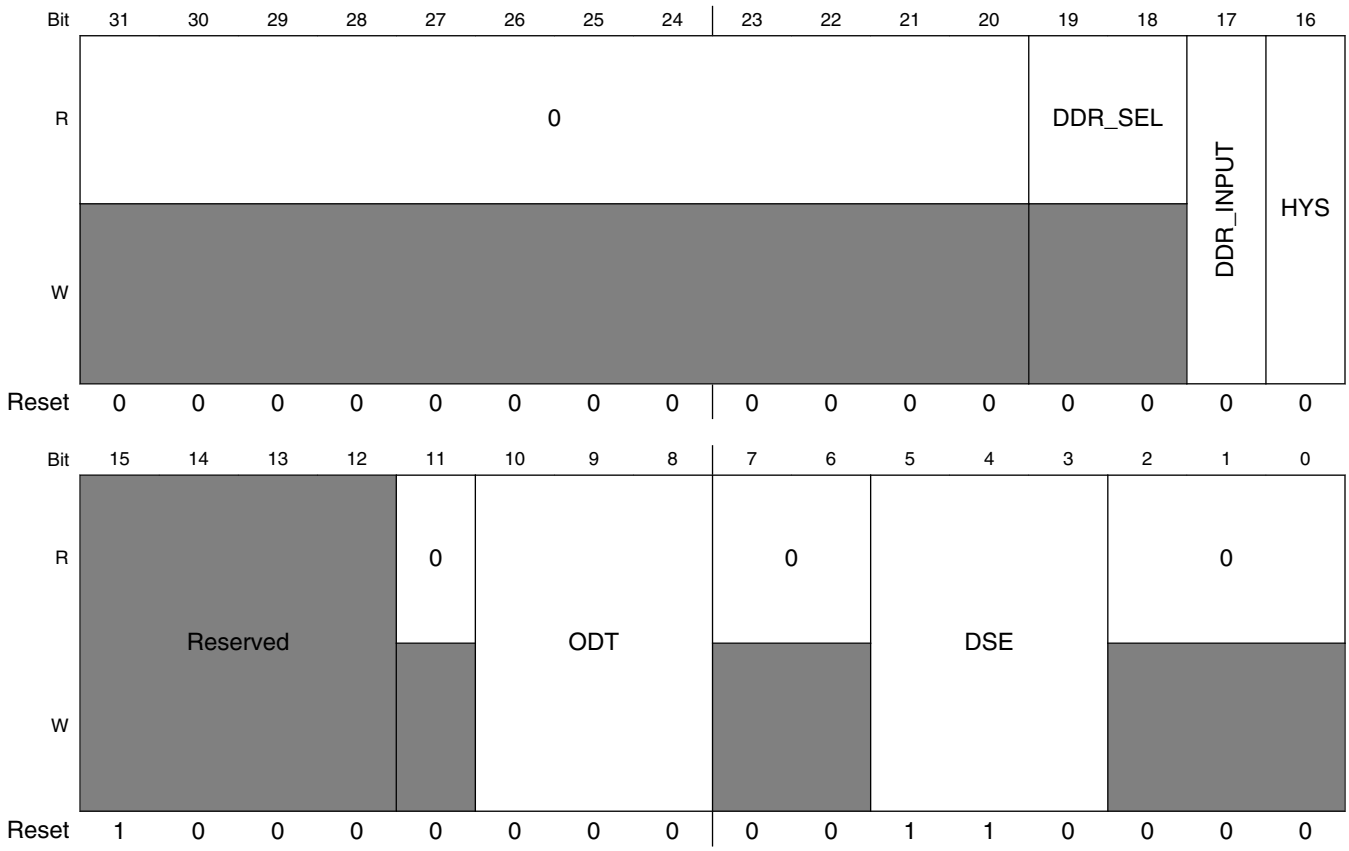
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_DQM3. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_DQM3. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.284 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM4)

Address: 20E_0000h base + 480h offset = 20E_0480h



IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM4 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_DQM4. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_DQM4.

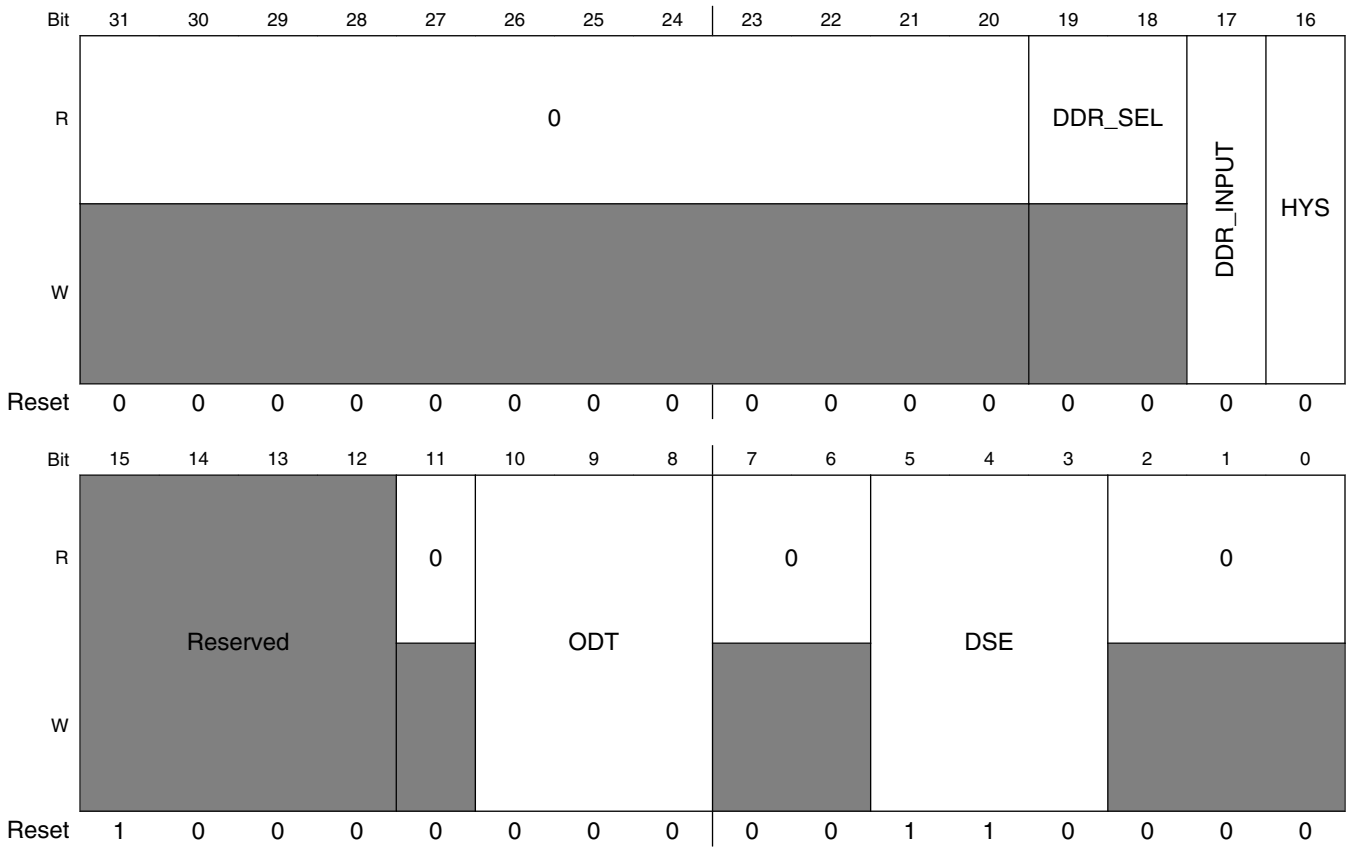
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM4 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_DQM4. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_DQM4. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.285 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM5)

Address: 20E_0000h base + 484h offset = 20E_0484h



IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM5 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_DQM5. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_DQM5.

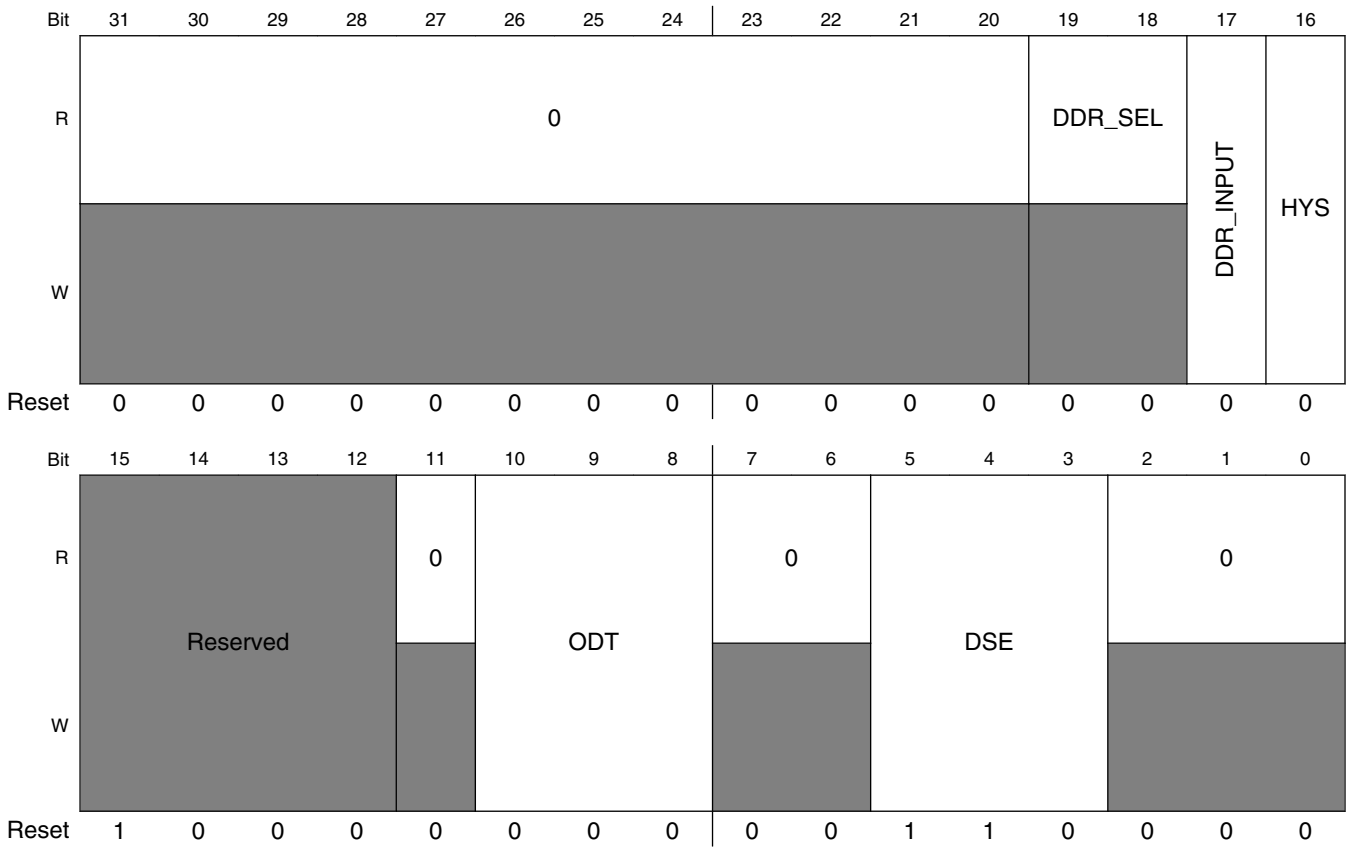
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM5 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_DQM5. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_DQM5. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.286 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM6)

Address: 20E_0000h base + 488h offset = 20E_0488h



IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM6 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_DQM6. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_DQM6.

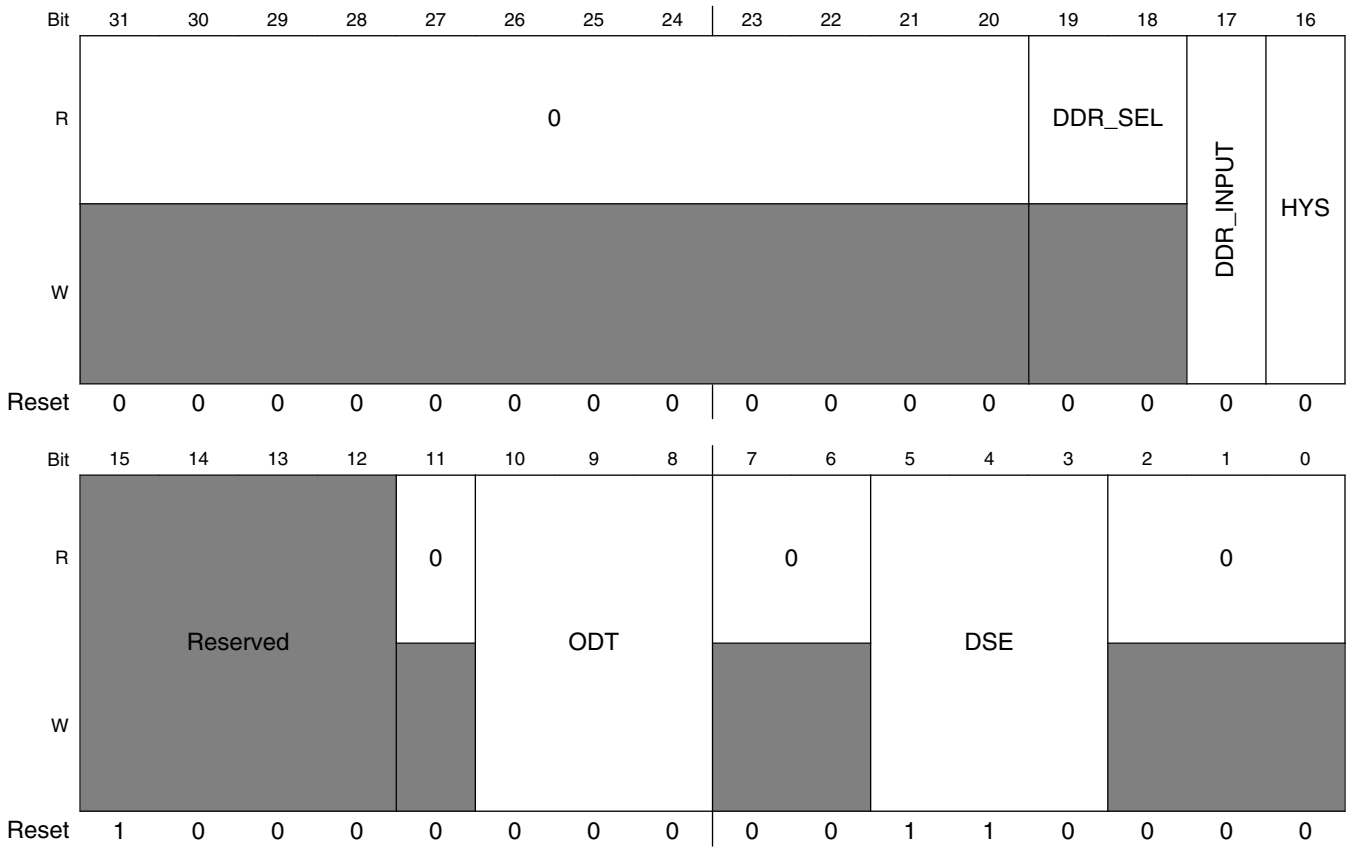
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM6 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_DQM6. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_DQM6. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.287 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM7)

Address: 20E_0000h base + 48Ch offset = 20E_048Ch



IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM7 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_DQM7. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_DQM7.

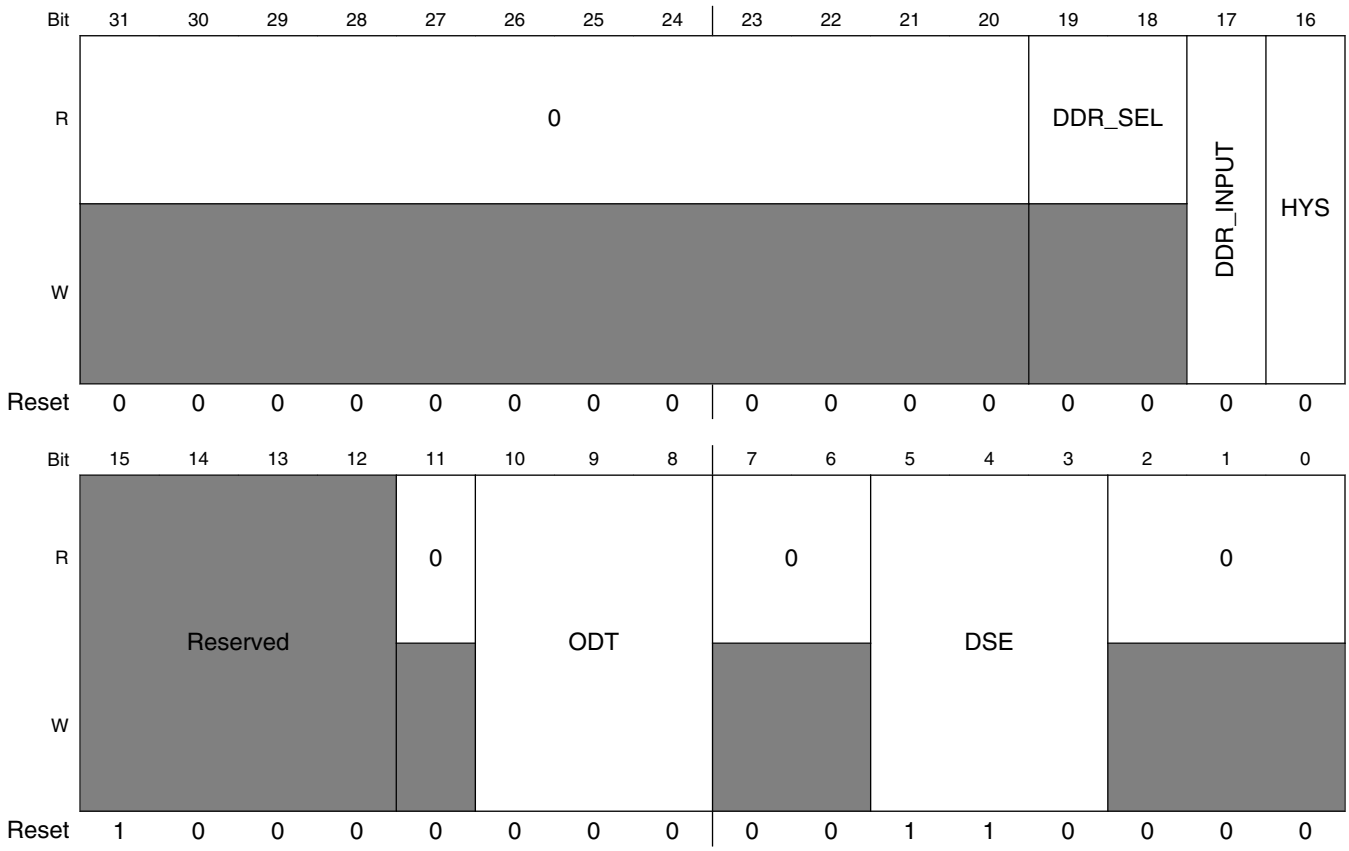
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM7 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_DQM7. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_DQM7. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.288 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS_B)

Address: 20E_0000h base + 490h offset = 20E_0490h



IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS_B field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_RAS. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_RAS.

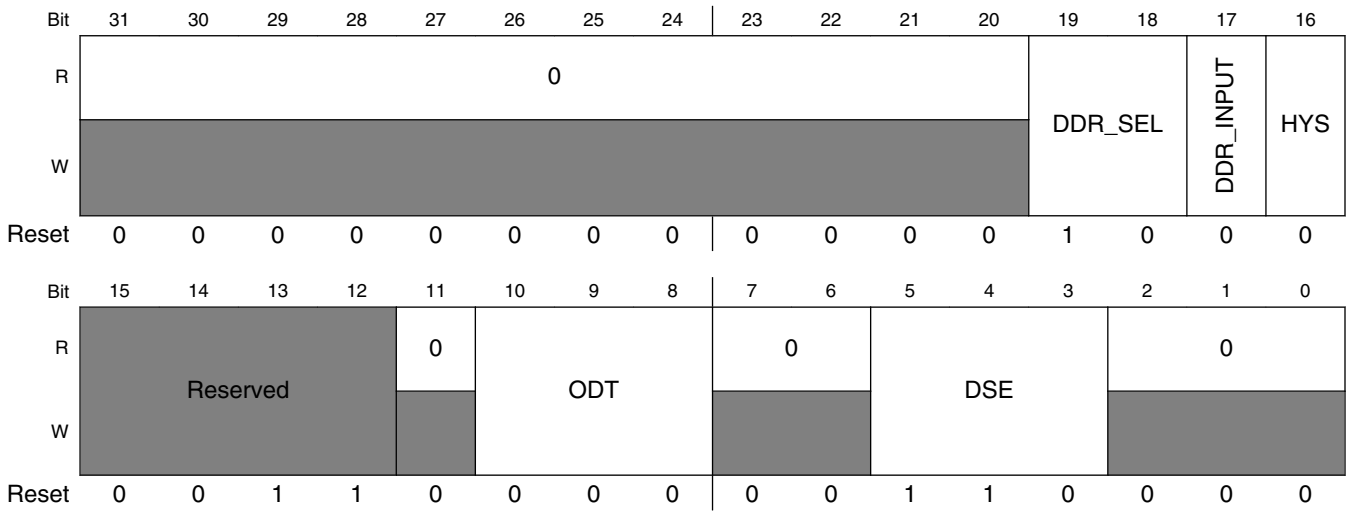
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS_B field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_RAS. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_RAS. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.289 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET)

Address: 20E_0000h base + 494h offset = 20E_0494h



IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field Select one of next values for pad: DRAM_RESET. 00 **DDR3_LPDDR2** — DDR3 and LPDDR2 mode. 01 **RESERVED1** — Reserved 10 **RESERVED2** — Reserved 11 **RESERVED3** — Reserved
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_RESET. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_RESET. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.

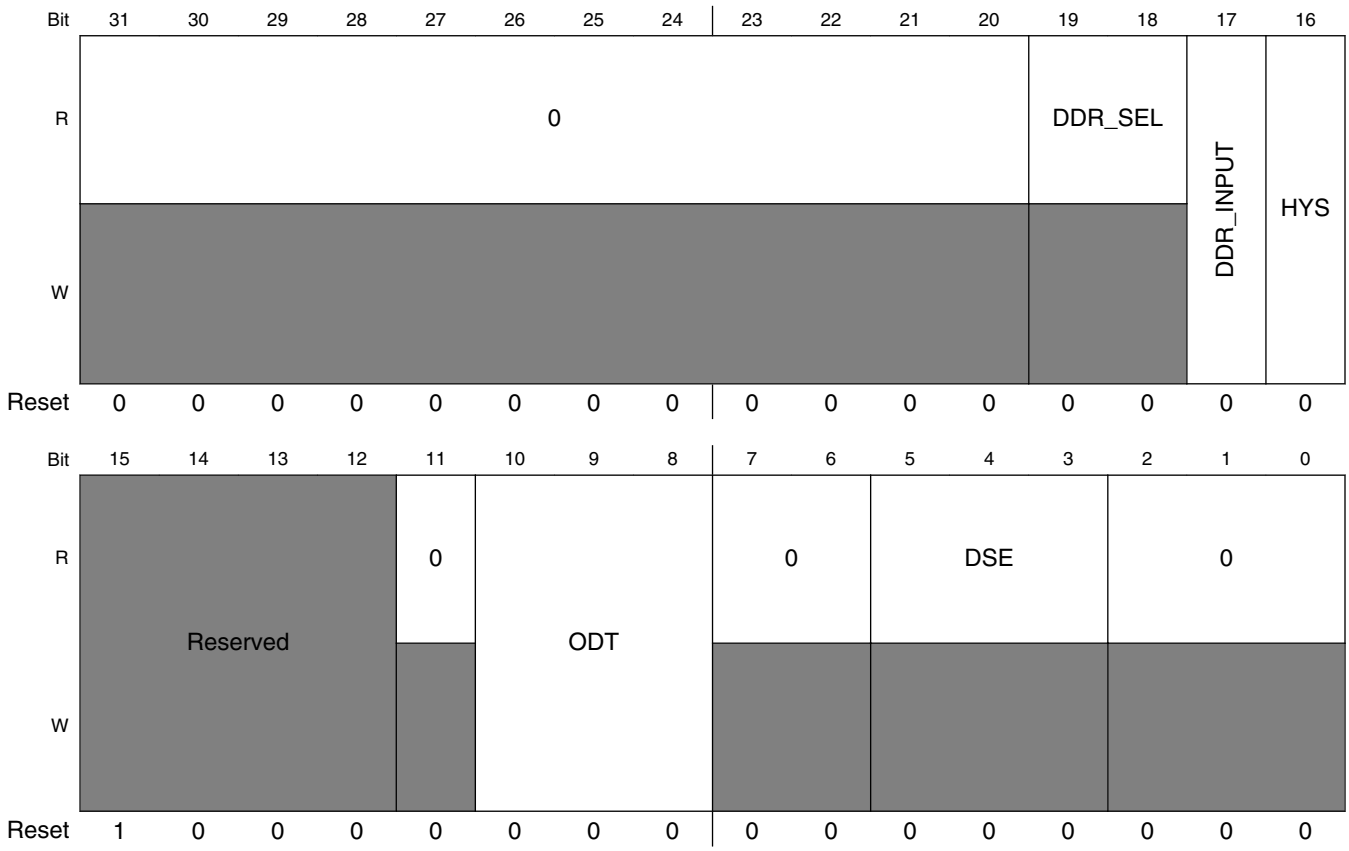
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET field descriptions (continued)

Field	Description
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_RESET. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_RESET. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.290 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA0)

Address: 20E_0000h base + 498h offset = 20E_0498h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA0 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_SDBA0. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_SDBA0.

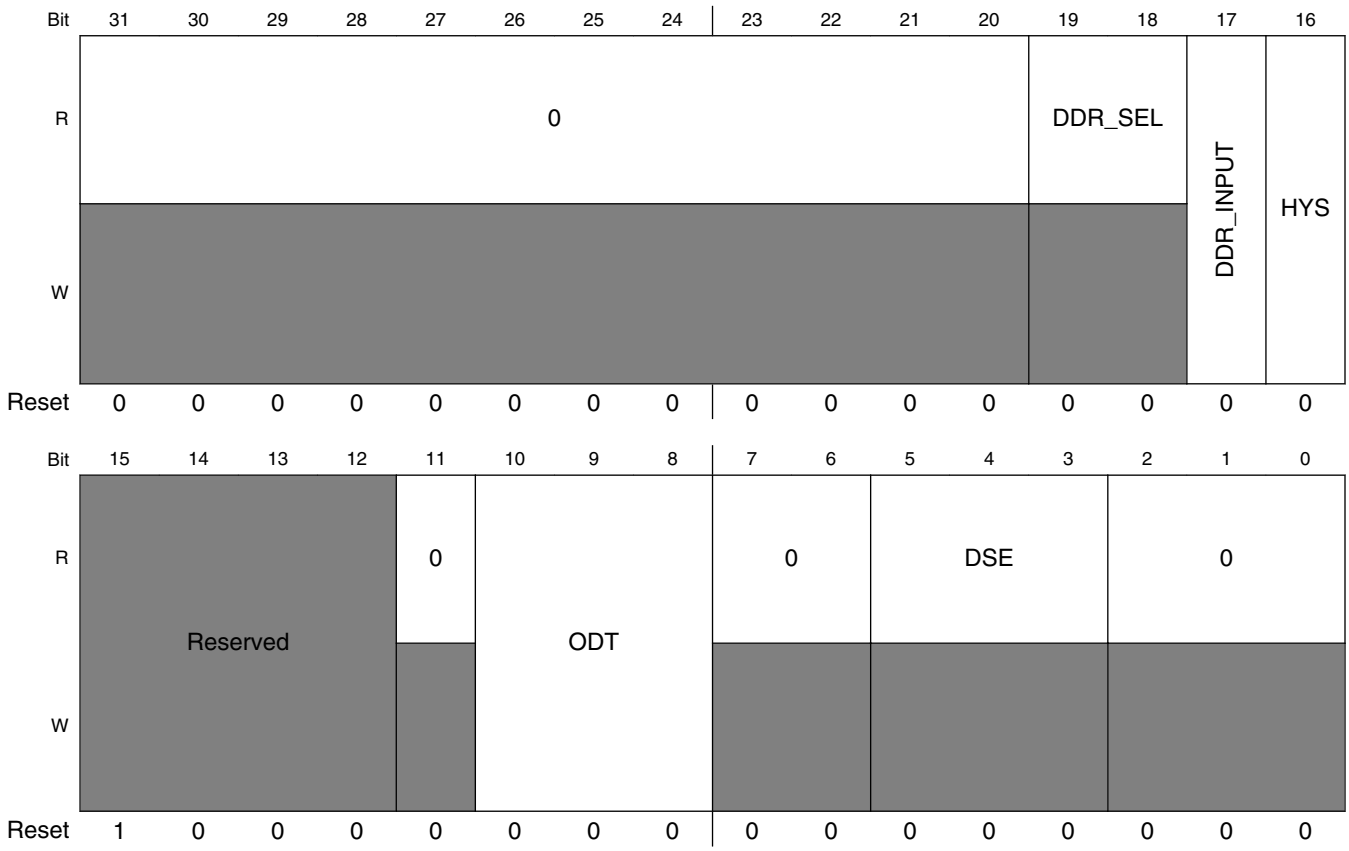
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA0 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_SDBA0. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.291 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA1)

Address: 20E_0000h base + 49Ch offset = 20E_049Ch



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA1 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_SDBA1. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_SDBA1.

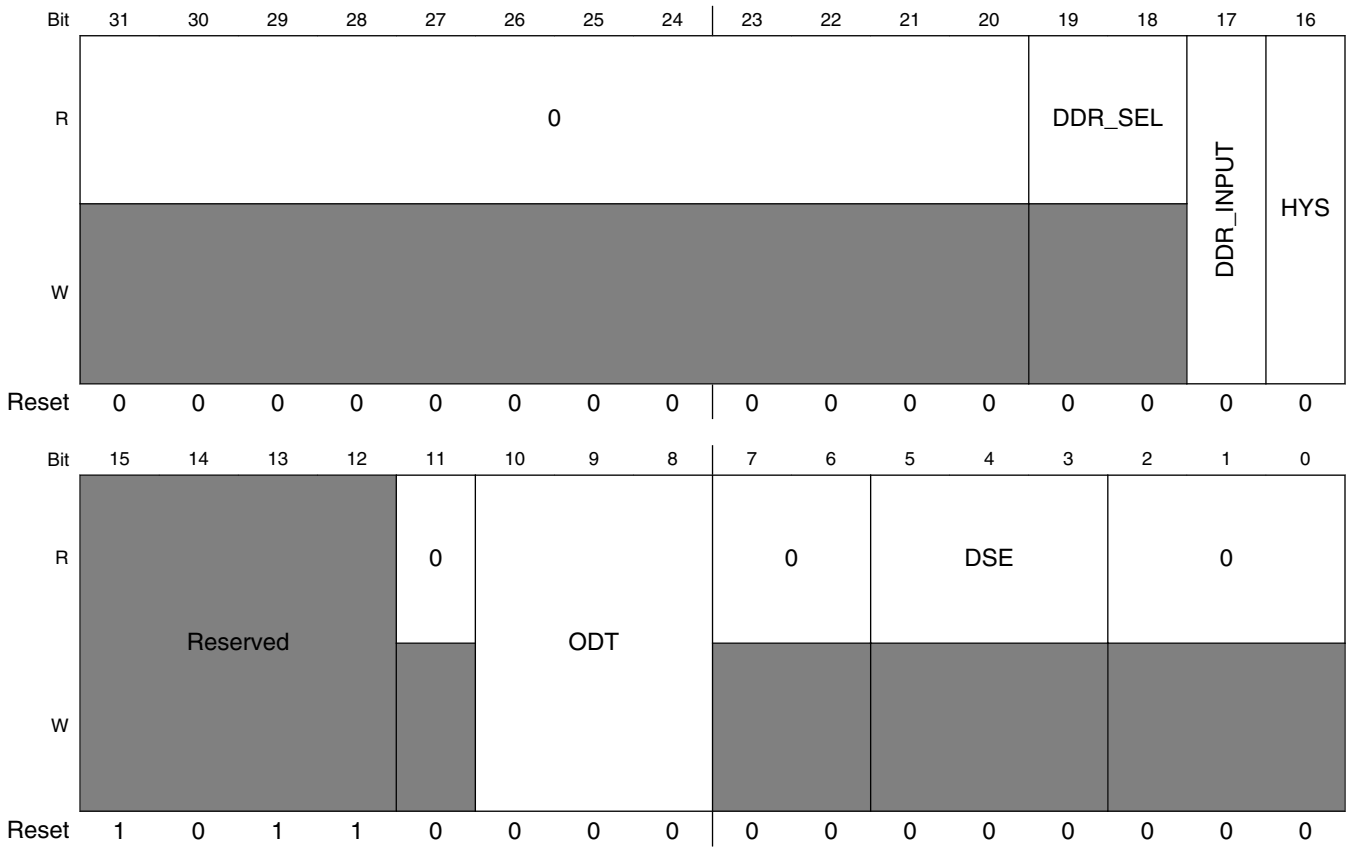
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA1 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_SDBA1. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_ADDDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.292 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA2)

Address: 20E_0000h base + 4A0h offset = 20E_04A0h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA2 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_SDBA2. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_SDBA2.

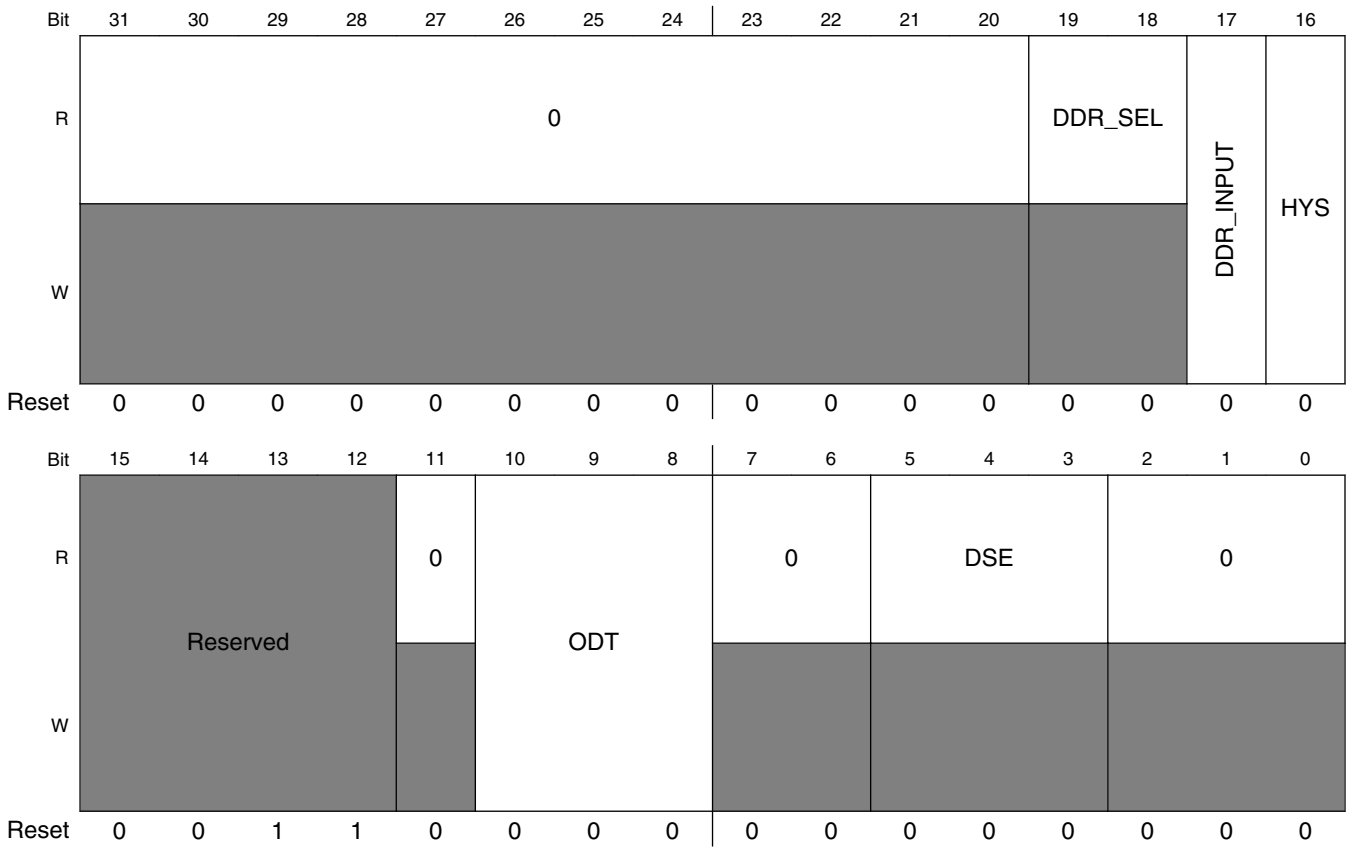
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA2 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_SDBA2. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_CTLDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.293 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE0)

Address: 20E_0000h base + 4A4h offset = 20E_04A4h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE0 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_SDCKE0. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_SDCKE0.

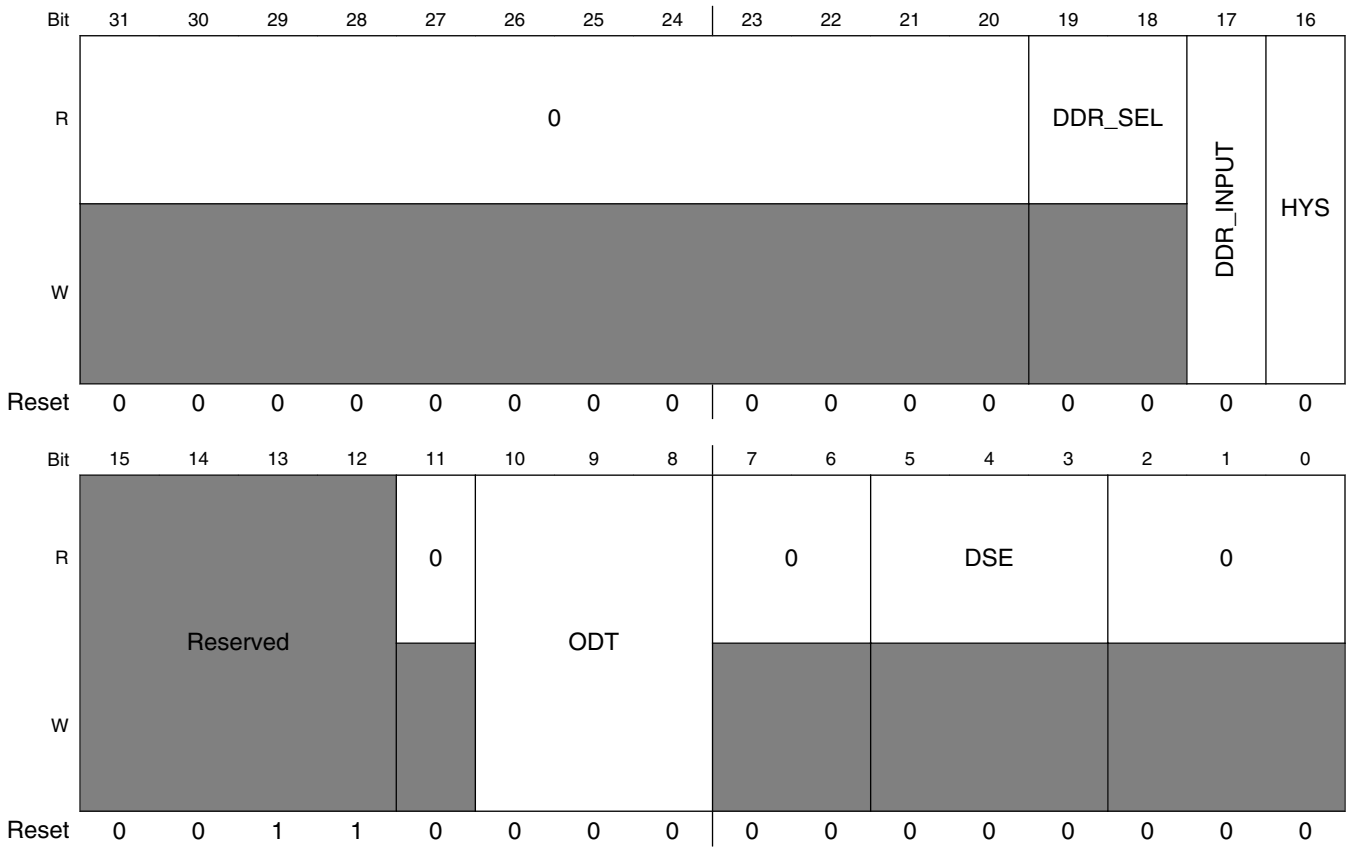
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE0 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_SDCKE0. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_CTLDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.294 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE1)

Address: 20E_0000h base + 4A8h offset = 20E_04A8h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE1 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_SDCKE1. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_SDCKE1.

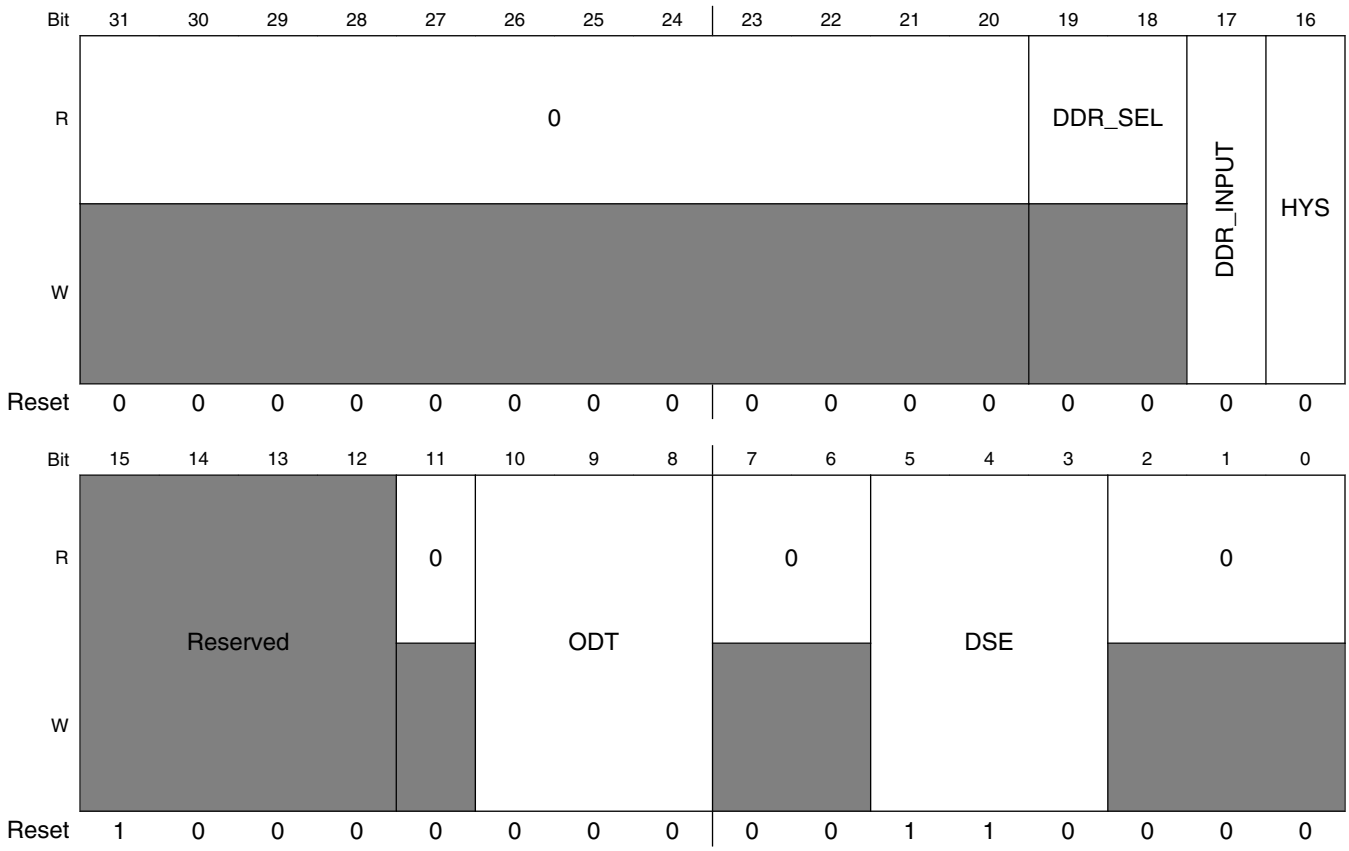
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE1 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_SDCKE1. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_CTLDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.295 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK0_P)

Address: 20E_0000h base + 4ACh offset = 20E_04ACh



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK0_P field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_SDCLK_0. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_SDCLK_0.

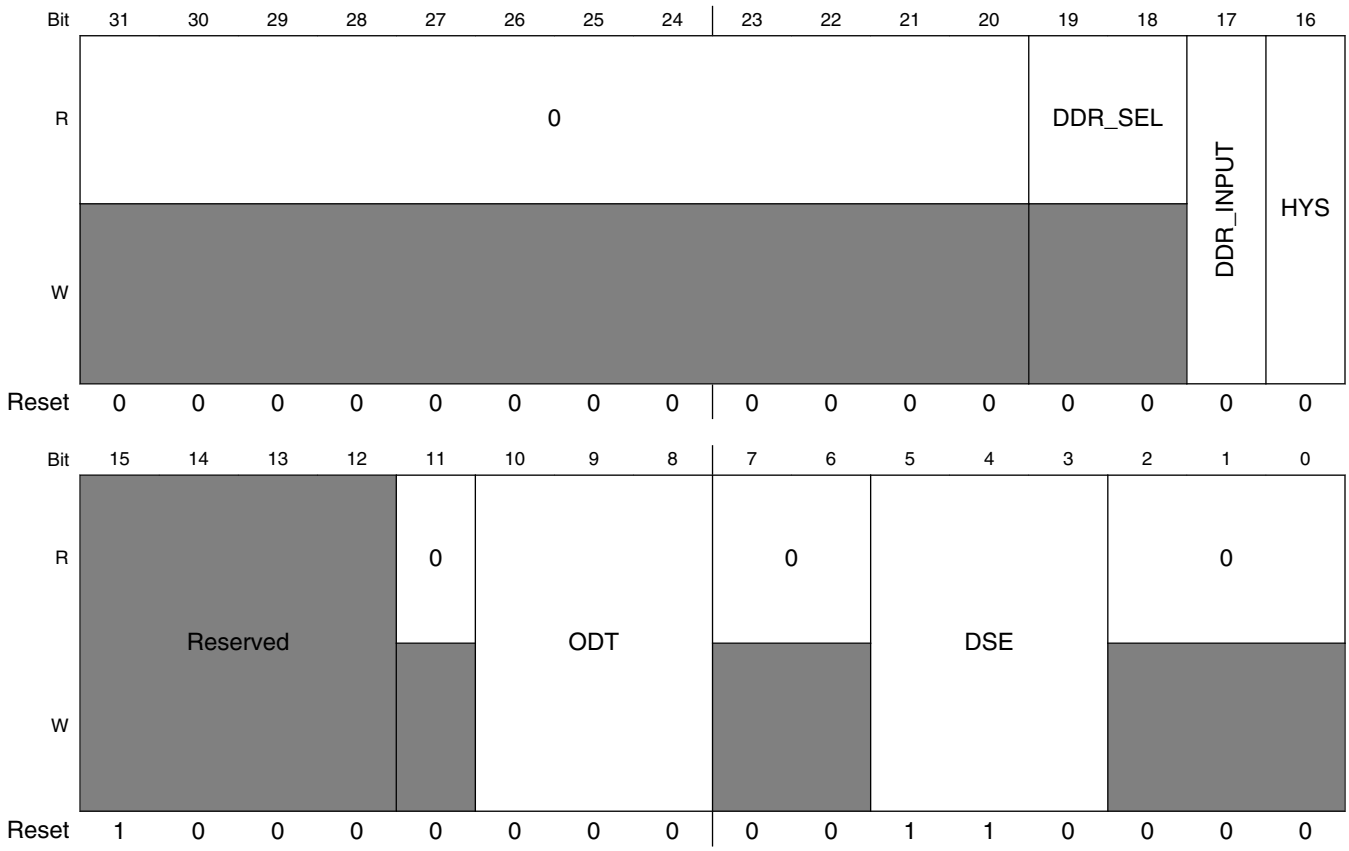
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK0_P field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_SDCLK_0. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_SDCLK_0. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.296 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK1_P)

Address: 20E_0000h base + 4B0h offset = 20E_04B0h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK1_P field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_SDCLK_1. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_SDCLK_1.

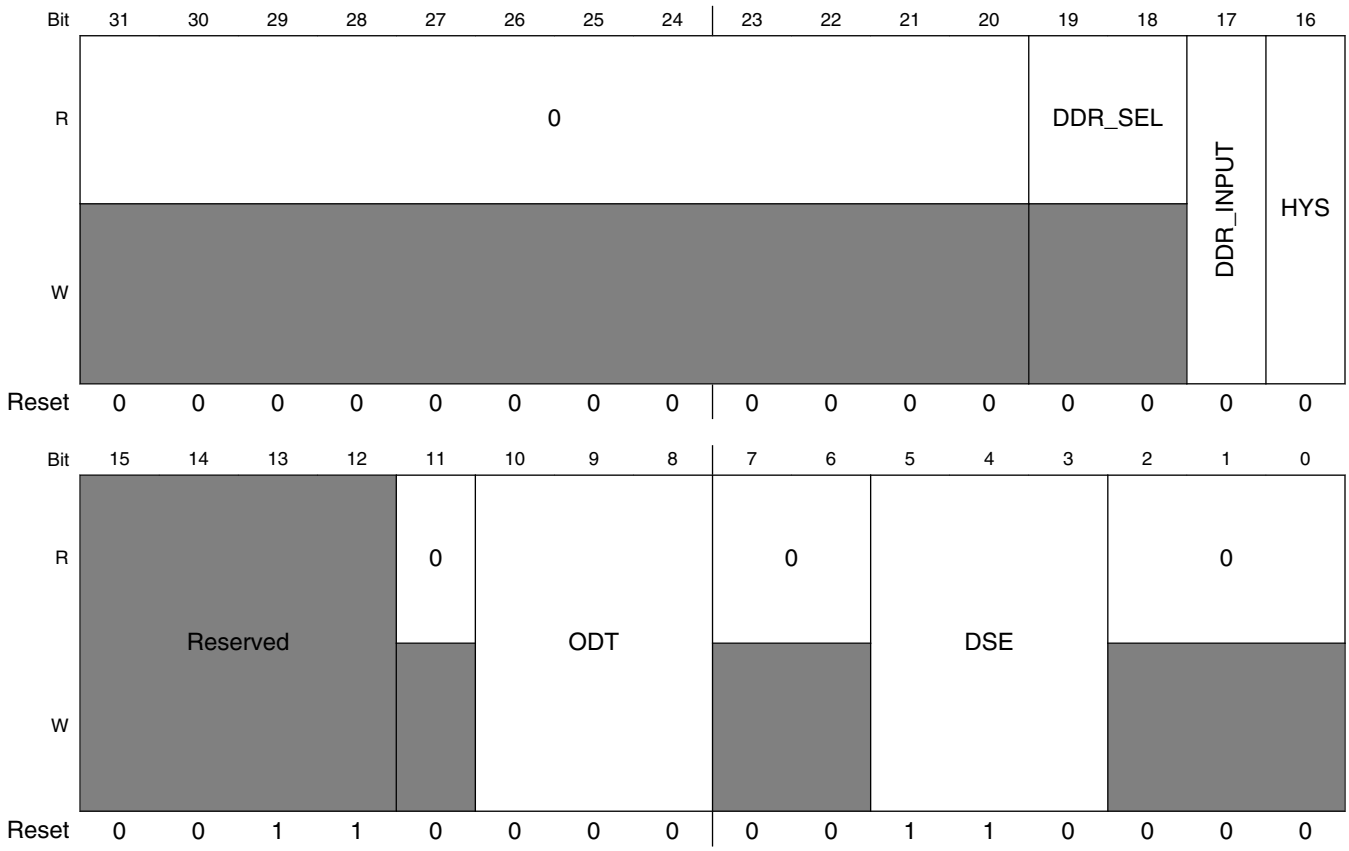
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK1_P field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_SDCLK_1. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_SDCLK_1. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.297 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ODT0)

Address: 20E_0000h base + 4B4h offset = 20E_04B4h



IOMUXC_SW_PAD_CTL_PAD_DRAM_ODT0 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_SDODT0. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_SDODT0.

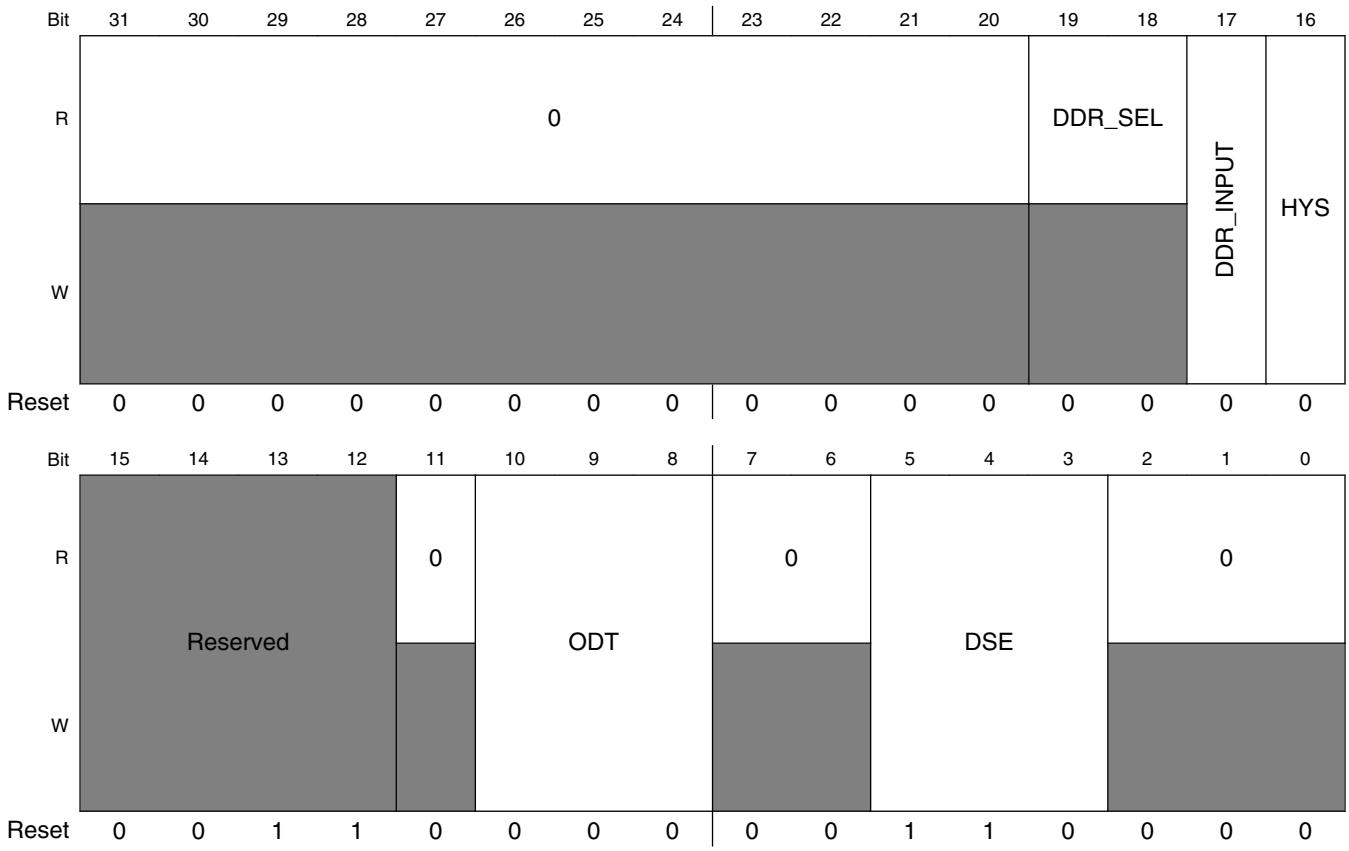
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ODT0 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_SDODT0. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_SDODT0. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.298 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_ODT1)

Address: 20E_0000h base + 4B8h offset = 20E_04B8h



IOMUXC_SW_PAD_CTL_PAD_DRAM_ODT1 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_SDODT1. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_SDODT1.

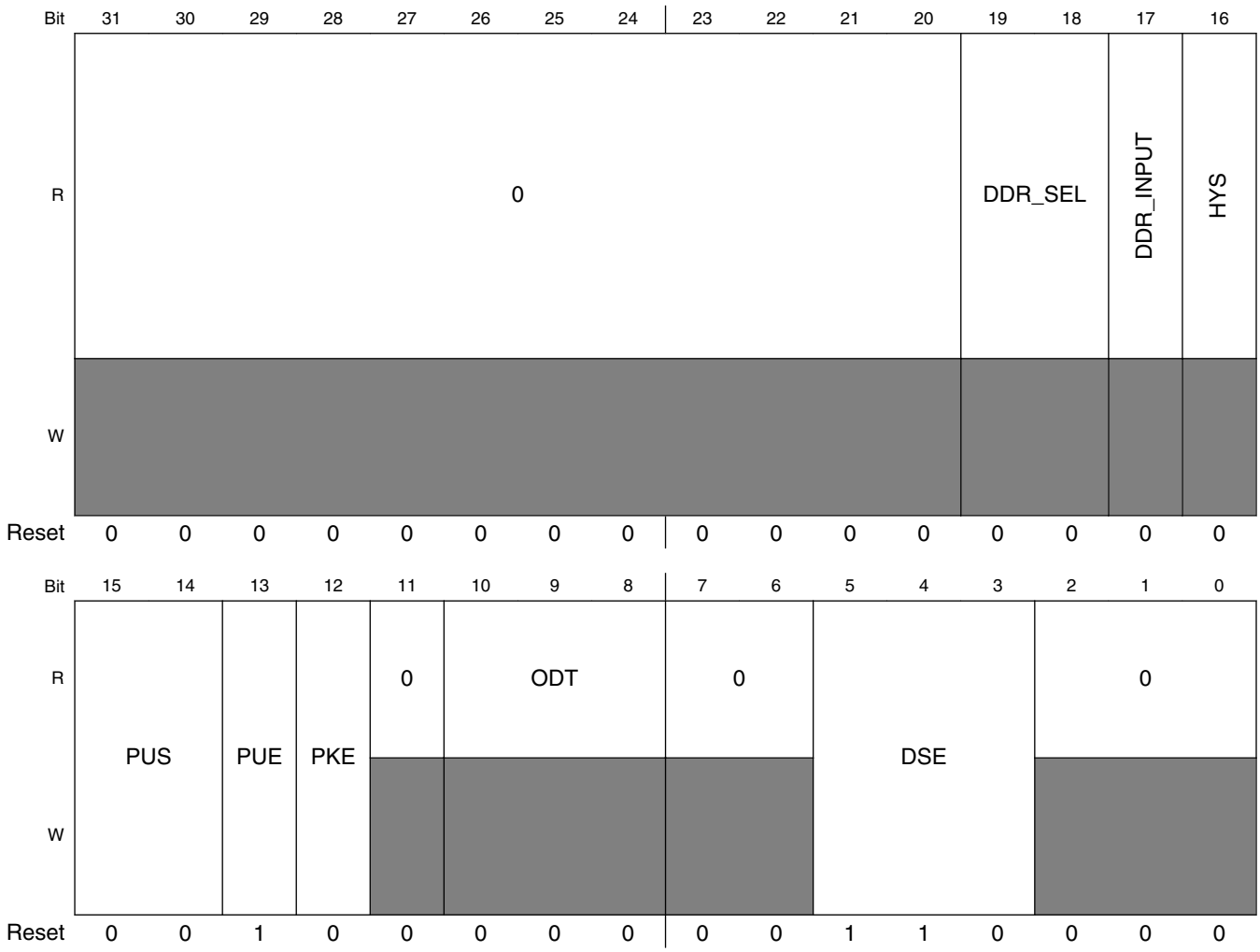
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_ODT1 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_SDODT1. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_SDODT1. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.299 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0_P)

Address: 20E_0000h base + 4BCh offset = 20E_04BCh



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0_P field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL

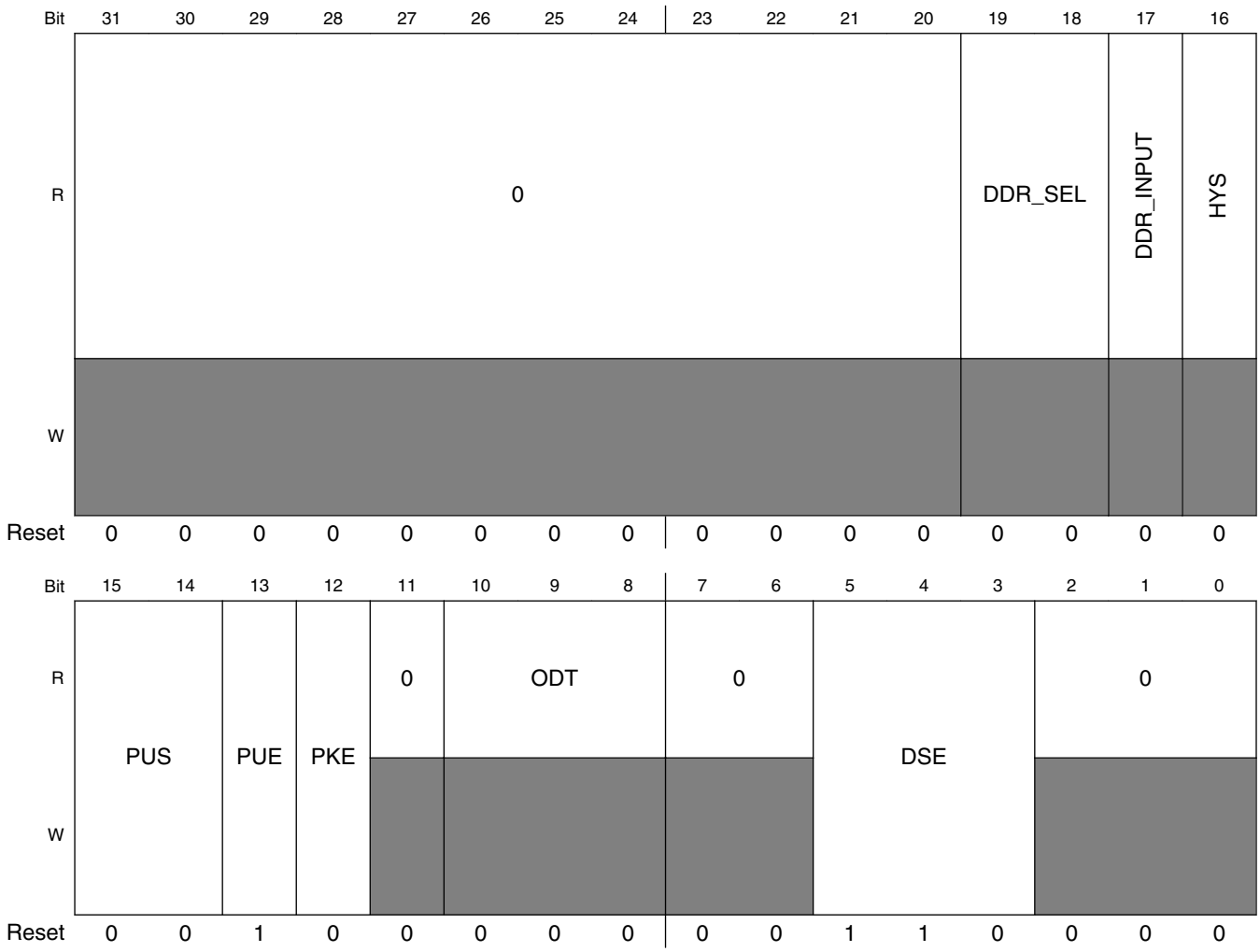
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0_P field descriptions (continued)

Field	Description
	Note: The value of this field does not reflect the value of the Group Control Register.
16 HYS	Hysteresis Enable Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRHYS Note: The value of this field does not reflect the value of the Group Control Register.
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DRAM_SDQS0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DRAM_SDQS0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DRAM_SDQS0. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Read Only Field The value of this field is fixed and cannot be changed. 000 **DISABLED** — Disabled
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_SDQS0. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.300 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS1_P)

Address: 20E_0000h base + 4C0h offset = 20E_04C0h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS1_P field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL

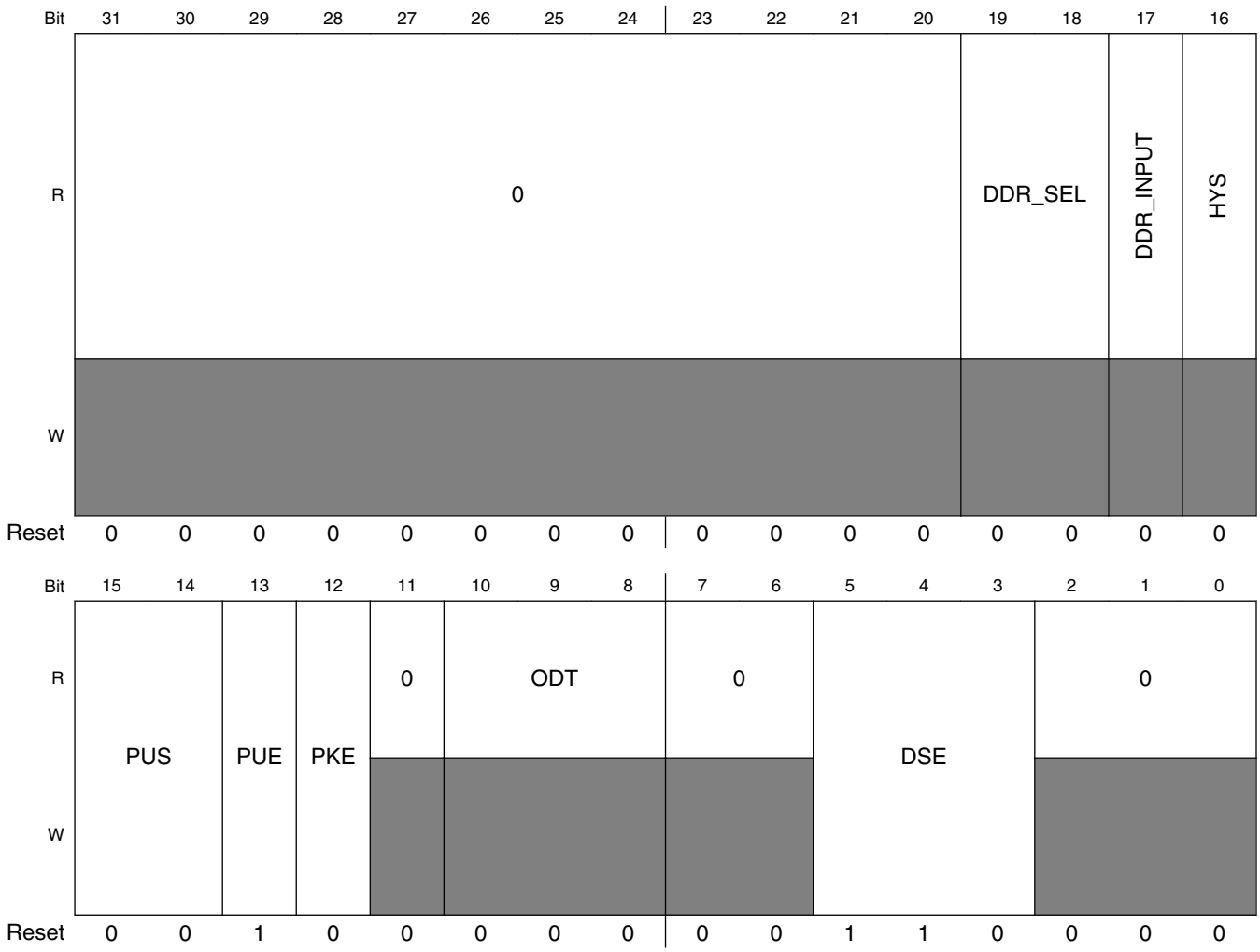
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS1_P field descriptions (continued)

Field	Description
	Note: The value of this field does not reflect the value of the Group Control Register.
16 HYS	Hysteresis Enable Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRHYS Note: The value of this field does not reflect the value of the Group Control Register.
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DRAM_SDQS1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DRAM_SDQS1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DRAM_SDQS1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Read Only Field The value of this field is fixed and cannot be changed. 000 **DISABLED** — Disabled
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_SDQS1. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.301 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2_P)

Address: 20E_0000h base + 4C4h offset = 20E_04C4h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2_P field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL

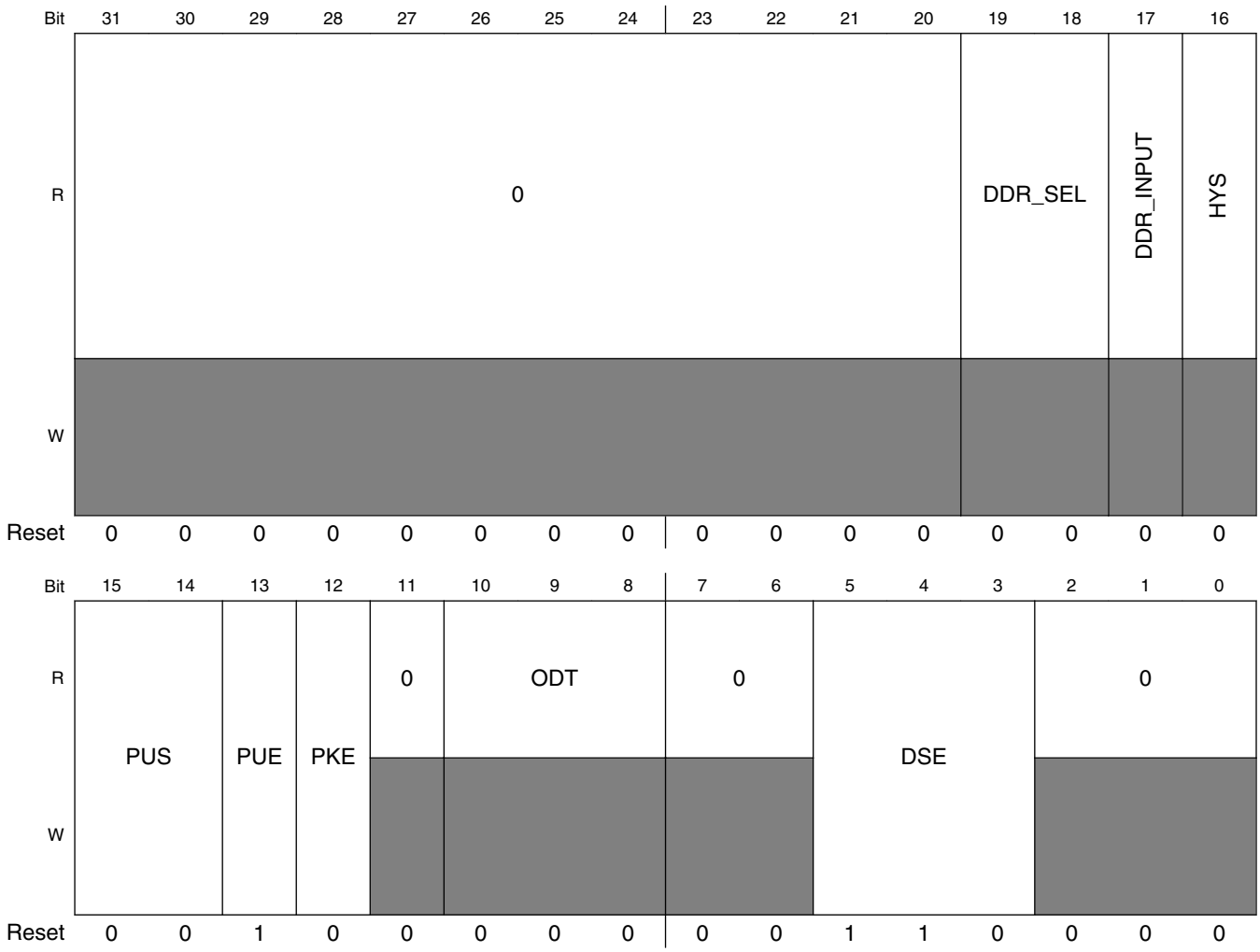
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2_P field descriptions (continued)

Field	Description
	Note: The value of this field does not reflect the value of the Group Control Register.
16 HYS	Hysteresis Enable Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRHYS Note: The value of this field does not reflect the value of the Group Control Register.
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DRAM_SDQS2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DRAM_SDQS2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DRAM_SDQS2. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Read Only Field The value of this field is fixed and cannot be changed. 000 **DISABLED** — Disabled
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_SDQS2. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.302 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3_P)

Address: 20E_0000h base + 4C8h offset = 20E_04C8h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3_P field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL

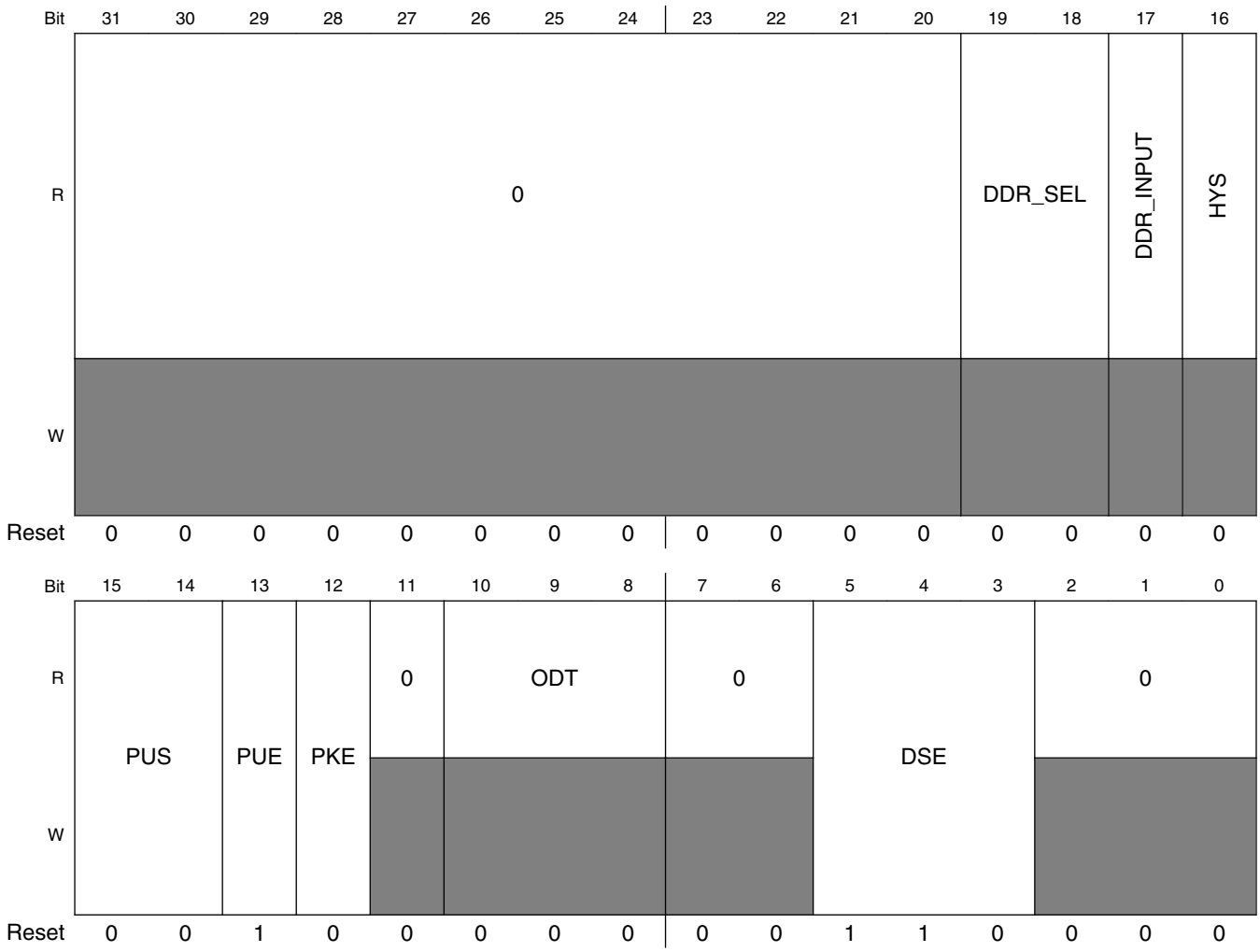
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3_P field descriptions (continued)

Field	Description
	Note: The value of this field does not reflect the value of the Group Control Register.
16 HYS	Hysteresis Enable Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRHYS Note: The value of this field does not reflect the value of the Group Control Register.
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DRAM_SDQS3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DRAM_SDQS3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DRAM_SDQS3. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Read Only Field The value of this field is fixed and cannot be changed. 000 **DISABLED** — Disabled
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_SDQS3. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.303 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS4_P)

Address: 20E_0000h base + 4CCh offset = 20E_04CCh



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS4_P field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL

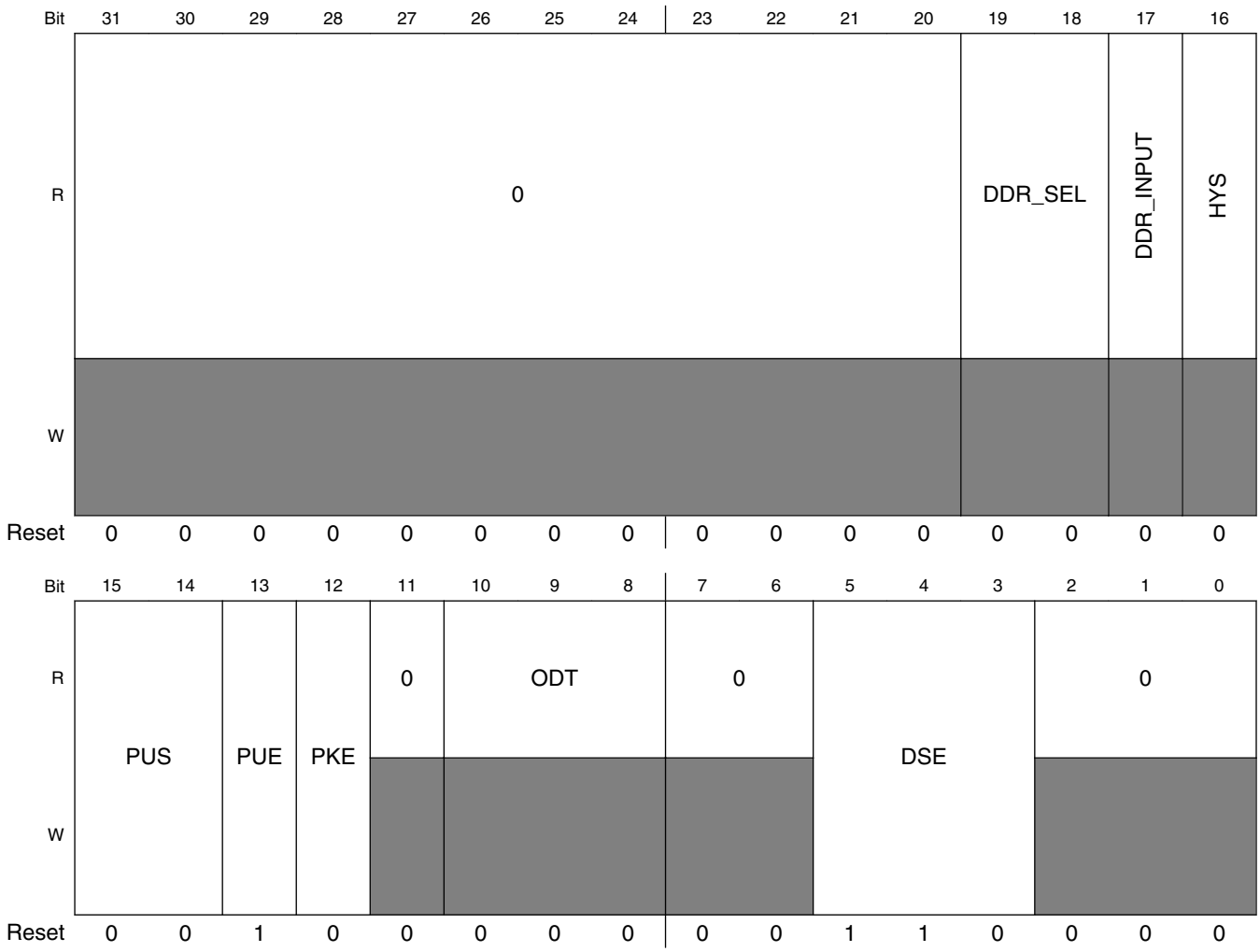
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS4_P field descriptions (continued)

Field	Description
	Note: The value of this field does not reflect the value of the Group Control Register.
16 HYS	Hysteresis Enable Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRHYS Note: The value of this field does not reflect the value of the Group Control Register.
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DRAM_SDQS4. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DRAM_SDQS4. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DRAM_SDQS4. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Read Only Field The value of this field is fixed and cannot be changed. 000 **DISABLED** — Disabled
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_SDQS4. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.304 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS5_P)

Address: 20E_0000h base + 4D0h offset = 20E_04D0h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS5_P field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL

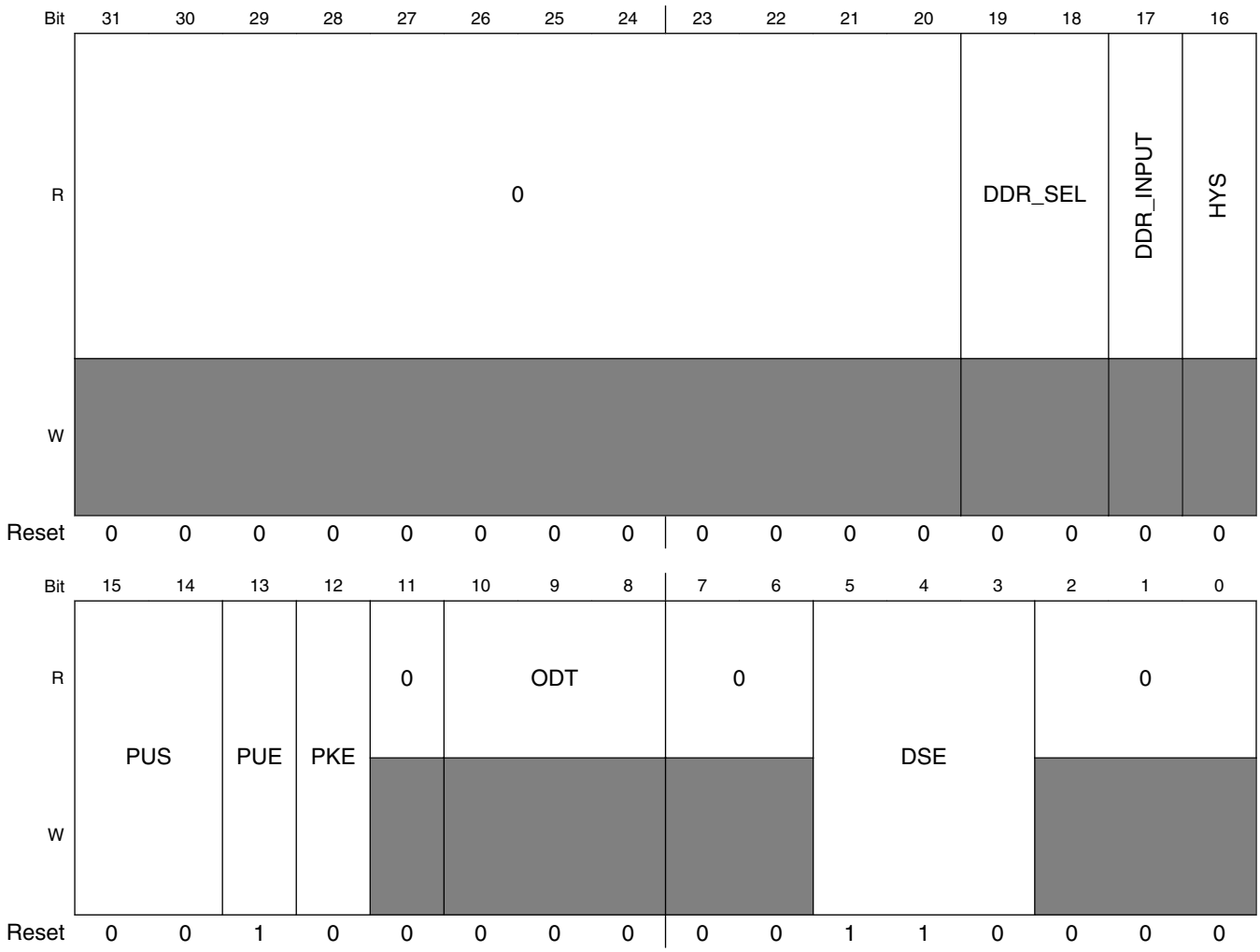
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS5_P field descriptions (continued)

Field	Description
	Note: The value of this field does not reflect the value of the Group Control Register.
16 HYS	Hysteresis Enable Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRHYS Note: The value of this field does not reflect the value of the Group Control Register.
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DRAM_SDQS5. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DRAM_SDQS5. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DRAM_SDQS5. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Read Only Field The value of this field is fixed and cannot be changed. 000 **DISABLED** — Disabled
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_SDQS5. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.305 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS6_P)

Address: 20E_0000h base + 4D4h offset = 20E_04D4h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS6_P field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL

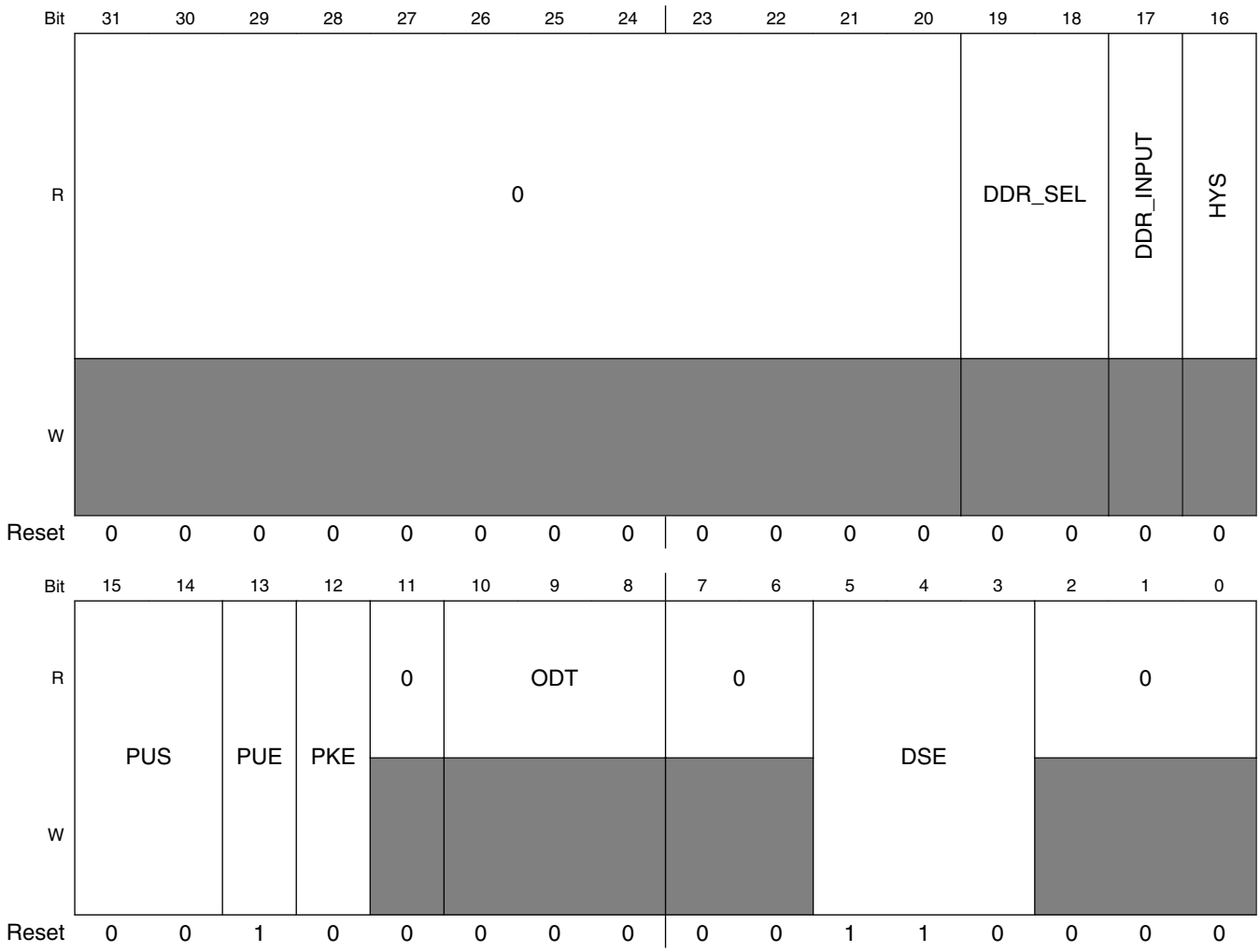
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS6_P field descriptions (continued)

Field	Description
	Note: The value of this field does not reflect the value of the Group Control Register.
16 HYS	Hysteresis Enable Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRHYS Note: The value of this field does not reflect the value of the Group Control Register.
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DRAM_SDQS6. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DRAM_SDQS6. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DRAM_SDQS6. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Read Only Field The value of this field is fixed and cannot be changed. 000 **DISABLED** — Disabled
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_SDQS6. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.306 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS7_P)

Address: 20E_0000h base + 4D8h offset = 20E_04D8h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS7_P field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL

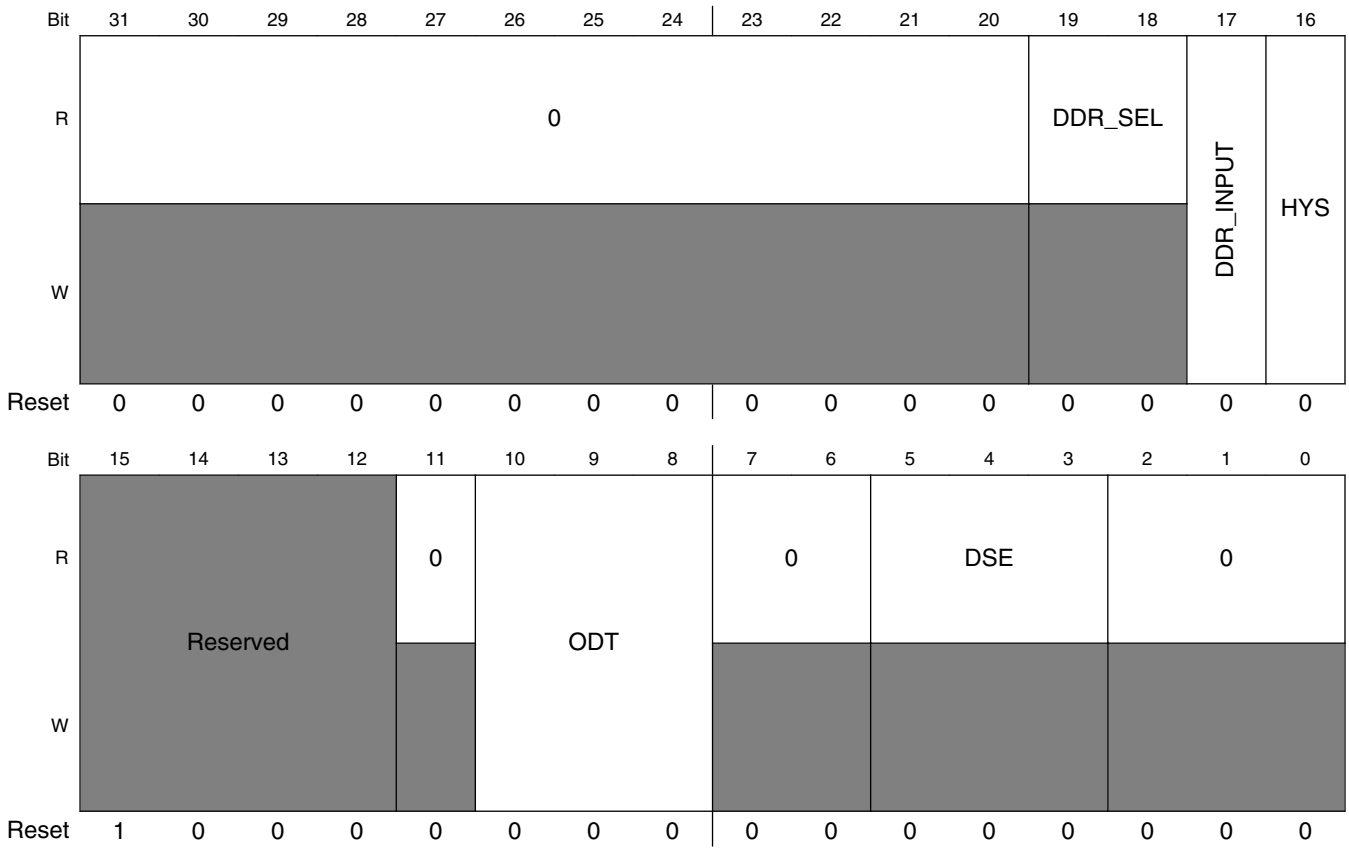
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS7_P field descriptions (continued)

Field	Description
	Note: The value of this field does not reflect the value of the Group Control Register.
16 HYS	Hysteresis Enable Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDRHYS Note: The value of this field does not reflect the value of the Group Control Register.
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: DRAM_SDQS7. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: DRAM_SDQS7. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: DRAM_SDQS7. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Read Only Field The value of this field is fixed and cannot be changed. 000 **DISABLED** — Disabled
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: DRAM_SDQS7. 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.307 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDWE_B)

Address: 20E_0000h base + 4DCh offset = 20E_04DCh



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDWE_B field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: DRAM_SDWE. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: DRAM_SDWE.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDWE_B field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–12 -	This field is reserved. Reserved
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for pad: DRAM_SDWE. 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_CTLDS Note: The value of this field does not reflect the value of the Group Control Register.
Reserved	This read-only field is reserved and always has the value 0.

37.4.308 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR16)

Address: 20E_0000h base + 4E0h offset = 20E_04E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR16 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR16 field descriptions (continued)

Field	Description
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_A16. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_A16. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_A16. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_A16. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_A16. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR16 field descriptions (continued)

Field	Description
	100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.309 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR17)

Address: 20E_0000h base + 4E4h offset = 20E_04E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR17 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_A17. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_A17. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up

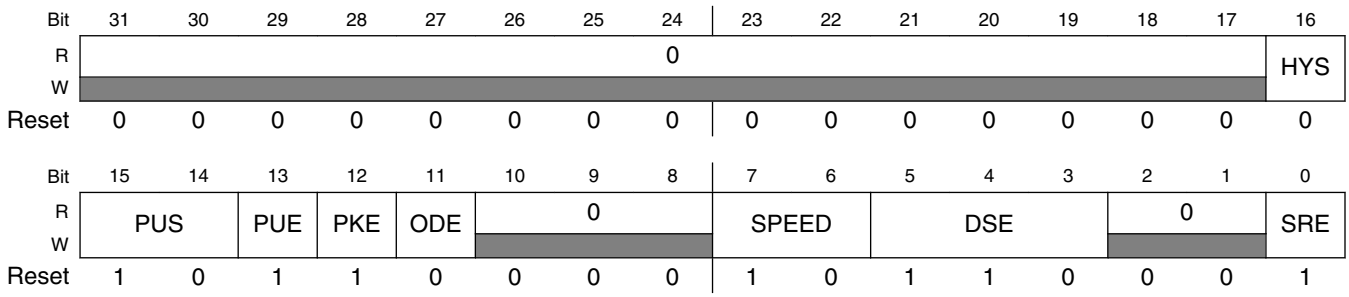
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR17 field descriptions (continued)

Field	Description
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_A17. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_A17. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_A17. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.310 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR18)

Address: 20E_0000h base + 4E8h offset = 20E_04E8h



IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR18 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_A18. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_A18. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_A18. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_A18. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR18 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_A18. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.311 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR19)

Address: 20E_0000h base + 4ECh offset = 20E_04ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR19 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_A19. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_A19. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_A19. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_A19. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_A19. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR19 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.312 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR20)

Address: 20E_0000h base + 4F0h offset = 20E_04F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR20 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_A20. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_A20. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR20 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_A20. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_A20. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_A20. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR20 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.313 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR21)

Address: 20E_0000h base + 4F4h offset = 20E_04F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR21 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_A21. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_A21. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_A21. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_A21.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR21 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_A21. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.314 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR22)

Address: 20E_0000h base + 4F8h offset = 20E_04F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR22 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_A22. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_A22. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_A22. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_A22. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR22 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_A22. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.315 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR23)

Address: 20E_0000h base + 4FCh offset = 20E_04FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR23 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_A23. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_A23. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_A23. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_A23. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_A23. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR23 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.316 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR24)

Address: 20E_0000h base + 500h offset = 20E_0500h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR24 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_A24. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_A24. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR24 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_A24. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_A24. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_A24. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR24 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.317 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR25)

Address: 20E_0000h base + 504h offset = 20E_0504h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR25 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_A25. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_A25. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_A25. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_A25.

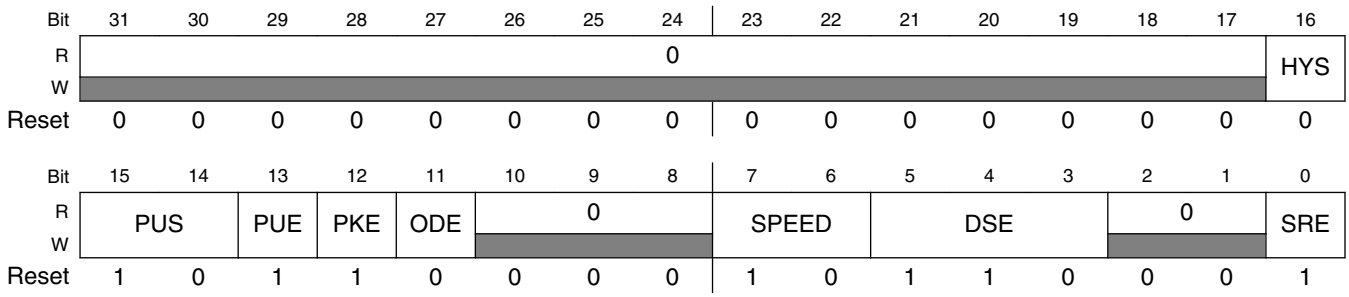
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_ADDR25 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_A25. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.318 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_BCLK)

Address: 20E_0000h base + 508h offset = 20E_0508h



IOMUXC_SW_PAD_CTL_PAD_EIM_BCLK field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_BCLK. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_BCLK. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_BCLK. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_BCLK. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_BCLK field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_BCLK. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.319 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_CS0_B)

Address: 20E_0000h base + 50Ch offset = 20E_050Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_CS0_B field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_CS0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_CS0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_CS0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_CS0. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_CS0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_CS0_B field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.320 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_CS1_B)

Address: 20E_0000h base + 510h offset = 20E_0510h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_CS1_B field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_CS1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_CS1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_CS1_B field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_CS1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_CS1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_CS1. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

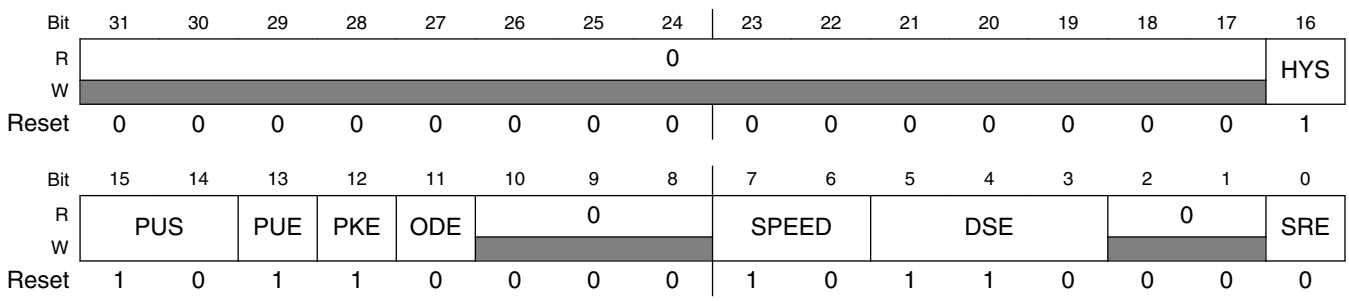
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_CS1_B field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.321 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA16)

Address: 20E_0000h base + 514h offset = 20E_0514h



IOMUXC_SW_PAD_CTL_PAD_EIM_DATA16 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D16. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D16. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D16. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D16.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA16 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D16. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.322 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA17)

Address: 20E_0000h base + 518h offset = 20E_0518h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA17 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D17. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D17. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D17. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D17. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA17 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D17. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.323 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA18)

Address: 20E_0000h base + 51Ch offset = 20E_051Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA18 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D18. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D18. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D18. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D18. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D18. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA18 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.324 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA19)

Address: 20E_0000h base + 520h offset = 20E_0520h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA19 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D19. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D19. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA19 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D19. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D19. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D19. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA19 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.325 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA20)

Address: 20E_0000h base + 524h offset = 20E_0524h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA20 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D20. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D20. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D20. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D20.

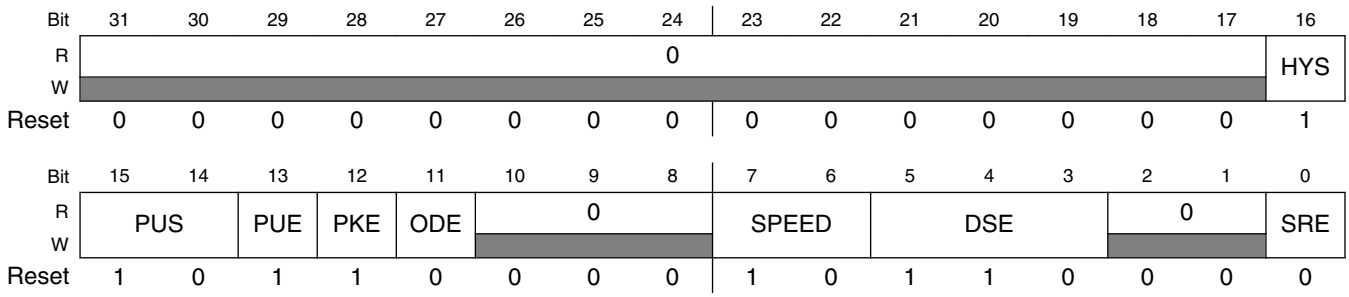
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA20 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D20. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.326 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA21)

Address: 20E_0000h base + 528h offset = 20E_0528h



IOMUXC_SW_PAD_CTL_PAD_EIM_DATA21 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D21. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D21. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D21. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D21. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA21 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D21. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.327 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA22)

Address: 20E_0000h base + 52Ch offset = 20E_052Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	0	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA22 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D22. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D22. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D22. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D22. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D22. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA22 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.328 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA23)

Address: 20E_0000h base + 530h offset = 20E_0530h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA23 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D23. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D23. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA23 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D23. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D23. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D23. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA23 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.329 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA24)

Address: 20E_0000h base + 534h offset = 20E_0534h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA24 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D24. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D24. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D24. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D24.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA24 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D24. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.330 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA25)

Address: 20E_0000h base + 538h offset = 20E_0538h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA25 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D25. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D25. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D25. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D25. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA25 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D25. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.331 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA26)

Address: 20E_0000h base + 53Ch offset = 20E_053Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA26 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D26. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D26. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D26. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D26. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D26. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA26 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.332 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA27)

Address: 20E_0000h base + 540h offset = 20E_0540h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA27 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D27. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D27. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA27 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D27. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D27. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D27. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA27 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.333 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA28)

Address: 20E_0000h base + 544h offset = 20E_0544h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA28 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D28. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D28. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D28. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D28.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA28 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D28. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.334 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA29)

Address: 20E_0000h base + 548h offset = 20E_0548h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA29 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D29. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D29. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D29. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D29. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA29 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D29. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.335 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA30)

Address: 20E_0000h base + 54Ch offset = 20E_054Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA30 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D30. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D30. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D30. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D30. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D30. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA30 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.336 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_DATA31)

Address: 20E_0000h base + 550h offset = 20E_0550h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	0	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA31 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_D31. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_D31. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA31 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_D31. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_D31. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_D31. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DATA31 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.337 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD00)

Address: 20E_0000h base + 554h offset = 20E_0554h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_AD00 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA0.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD00 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.338 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD01)

Address: 20E_0000h base + 558h offset = 20E_0558h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_AD01 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD01 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA1. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.339 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD10)

Address: 20E_0000h base + 55Ch offset = 20E_055Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_AD10 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA10. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA10. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA10. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA10. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA10. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD10 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.340 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD11)

Address: 20E_0000h base + 560h offset = 20E_0560h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_AD11 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA11. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA11. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD11 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA11. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA11. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA11. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD11 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.341 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD12)

Address: 20E_0000h base + 564h offset = 20E_0564h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_AD12 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA12. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA12. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA12. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA12.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD12 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA12. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.342 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD13)

Address: 20E_0000h base + 568h offset = 20E_0568h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_AD13 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA13. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA13. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA13. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA13. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD13 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA13. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.343 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD14)

Address: 20E_0000h base + 56Ch offset = 20E_056Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_AD14 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA14. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA14. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA14. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA14. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA14. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD14 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.344 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD15)

Address: 20E_0000h base + 570h offset = 20E_0570h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_AD15 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA15. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA15. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD15 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA15. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA15. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA15. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD15 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.345 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD02)

Address: 20E_0000h base + 574h offset = 20E_0574h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_AD02 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA2. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA2.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD02 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA2. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.346 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD03)

Address: 20E_0000h base + 578h offset = 20E_0578h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_AD03 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA3. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA3. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD03 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA3. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.347 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD04)

Address: 20E_0000h base + 57Ch offset = 20E_057Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_AD04 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA4. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA4. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA4. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA4. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA4. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD04 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.348 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD05)

Address: 20E_0000h base + 580h offset = 20E_0580h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_AD05 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA5. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA5. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD05 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA5. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA5. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA5. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

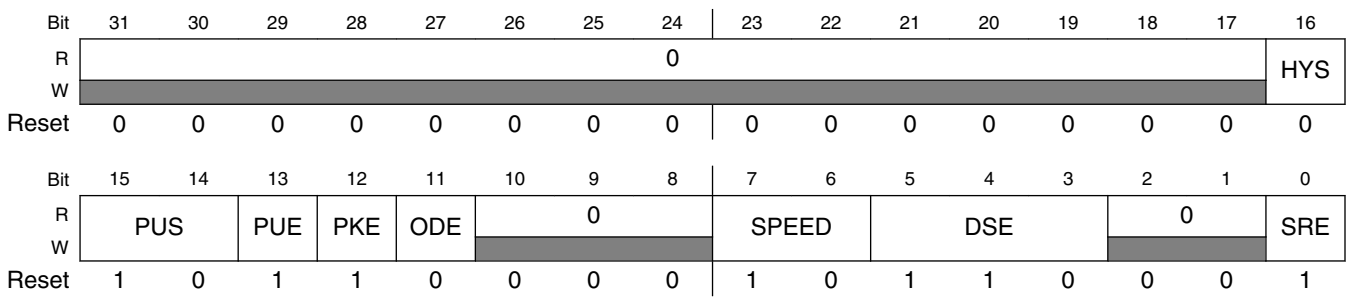
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD05 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.349 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD06)

Address: 20E_0000h base + 584h offset = 20E_0584h



IOMUXC_SW_PAD_CTL_PAD_EIM_AD06 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA6. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA6. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA6. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA6.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD06 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA6. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.350 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD07)

Address: 20E_0000h base + 588h offset = 20E_0588h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_AD07 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA7. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA7. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA7. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA7. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD07 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA7. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.351 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD08)

Address: 20E_0000h base + 58Ch offset = 20E_058Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_AD08 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA8. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA8. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA8. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA8. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA8. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD08 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.352 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_AD09)

Address: 20E_0000h base + 590h offset = 20E_0590h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_AD09 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_DA9. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_DA9. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD09 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_DA9. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_DA9. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_DA9. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_AD09 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.353 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_EB0_B)

Address: 20E_0000h base + 594h offset = 20E_0594h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_EB0_B field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_EB0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_EB0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_EB0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_EB0.

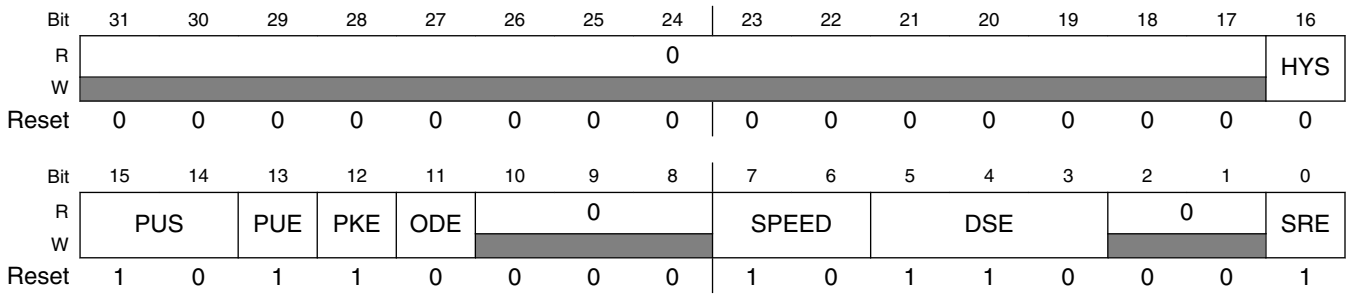
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_EB0_B field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_EB0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.354 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_EB1_B)

Address: 20E_0000h base + 598h offset = 20E_0598h



IOMUXC_SW_PAD_CTL_PAD_EIM_EB1_B field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_EB1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_EB1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_EB1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_EB1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_EB1_B field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_EB1. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.355 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_EB2_B)

Address: 20E_0000h base + 59Ch offset = 20E_059Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_EB2_B field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_EB2. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_EB2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_EB2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_EB2. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_EB2. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_EB2_B field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.356 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_EB3_B)

Address: 20E_0000h base + 5A0h offset = 20E_05A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_EB3_B field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_EB3. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_EB3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_EB3_B field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_EB3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_EB3. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_EB3. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_EB3_B field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.357 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_LBA_B)

Address: 20E_0000h base + 5A4h offset = 20E_05A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_LBA_B field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_LBA. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_LBA. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_LBA. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_LBA.

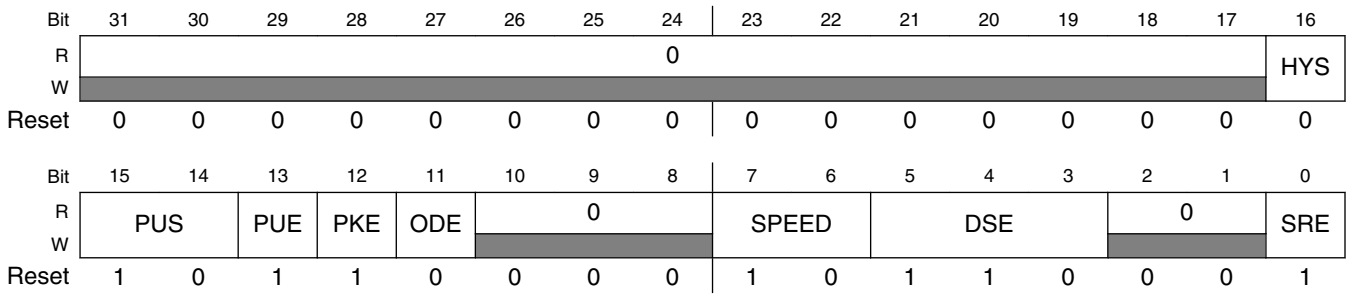
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_LBA_B field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_LBA. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.358 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_OE_B)

Address: 20E_0000h base + 5A8h offset = 20E_05A8h



IOMUXC_SW_PAD_CTL_PAD_EIM_OE_B field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_OE. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_OE. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_OE. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_OE. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_OE_B field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_OE. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.359 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_RW)

Address: 20E_0000h base + 5ACh offset = 20E_05ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_EIM_RW field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_RW. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_RW. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_RW. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_RW. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_RW. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_RW field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.360 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_EIM_WAIT_B)

Address: 20E_0000h base + 5B0h offset = 20E_05B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_WAIT_B field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: EIM_WAIT. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: EIM_WAIT. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_WAIT_B field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: EIM_WAIT. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: EIM_WAIT. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: EIM_WAIT. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

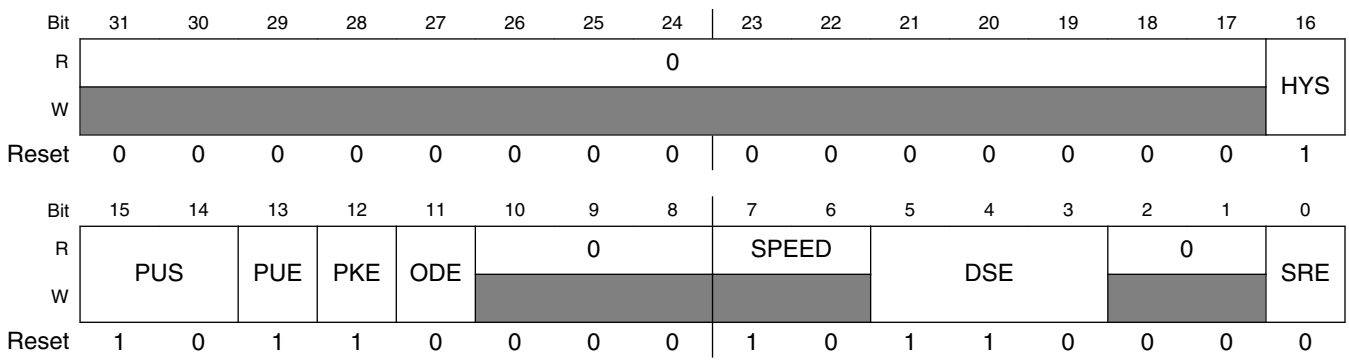
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_WAIT_B field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.361 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_CRS_DV)

Address: 20E_0000h base + 5B4h offset = 20E_05B4h



IOMUXC_SW_PAD_CTL_PAD_ENET_CRS_DV field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: ENET_CRS_DV. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: ENET_CRS_DV. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: ENET_CRS_DV. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: ENET_CRS_DV.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_CRS_DV field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. Read Only Field The value of this field is fixed and cannot be changed. 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: ENET_CRS_DV. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.362 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_MDC)

Address: 20E_0000h base + 5B8h offset = 20E_05B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_ENET_MDC field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: ENET_MDC. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: ENET_MDC. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: ENET_MDC. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: ENET_MDC. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_MDC field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: ENET_MDC. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.363 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_MDIO)

Address: 20E_0000h base + 5BCh offset = 20E_05BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_ENET_MDIO field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: ENET_MDIO. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: ENET_MDIO. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: ENET_MDIO. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: ENET_MDIO. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: ENET_MDIO. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_MDIO field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.364 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_REF_CLK)

Address: 20E_0000h base + 5C0h offset = 20E_05C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_ENET_REF_CLK field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: ENET_REF_CLK. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: ENET_REF_CLK. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_REF_CLK field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: ENET_REF_CLK. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: ENET_REF_CLK. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: ENET_REF_CLK. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_REF_CLK field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.365 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RX_ER)

Address: 20E_0000h base + 5C4h offset = 20E_05C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_ENET_RX_ER field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: ENET_RX_ER. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: ENET_RX_ER. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: ENET_RX_ER. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: ENET_RX_ER.

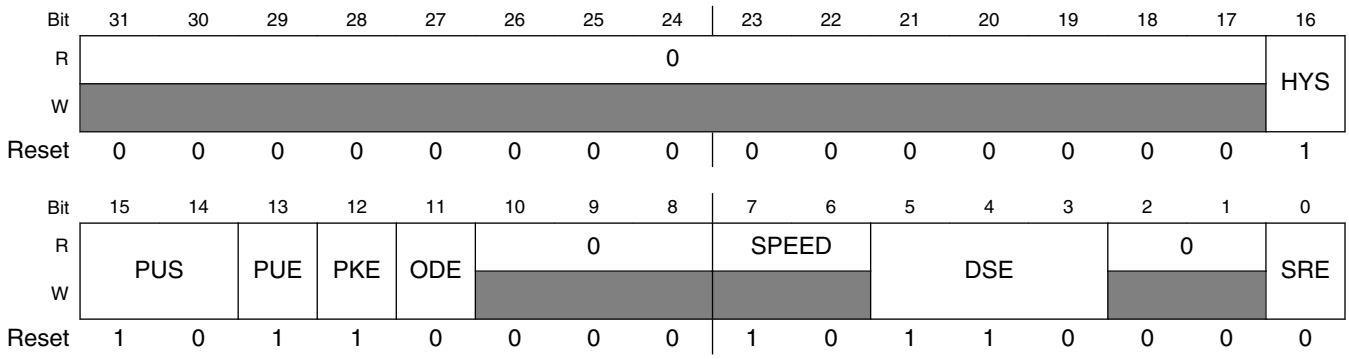
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_RX_ER field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: ENET_RX_ER. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.366 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RX_DATA0)

Address: 20E_0000h base + 5C8h offset = 20E_05C8h



IOMUXC_SW_PAD_CTL_PAD_ENET_RX_DATA0 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: ENET_RXD0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: ENET_RXD0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: ENET_RXD0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: ENET_RXD0. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_RX_DATA0 field descriptions (continued)

Field	Description
	0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. Read Only Field The value of this field is fixed and cannot be changed. 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: ENET_RXD0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.367 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RX_DATA1)

Address: 20E_0000h base + 5CCh offset = 20E_05CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_ENET_RX_DATA1 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: ENET_RXD1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: ENET_RXD1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: ENET_RXD1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: ENET_RXD1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: ENET_RXD1. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_RX_DATA1 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.368 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TX_EN)

Address: 20E_0000h base + 5D0h offset = 20E_05D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_ENET_TX_EN field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: ENET_TX_EN. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: ENET_TX_EN. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_TX_EN field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: ENET_TX_EN. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: ENET_TX_EN. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: ENET_TX_EN. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_TX_EN field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.369 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TX_DATA0)

Address: 20E_0000h base + 5D4h offset = 20E_05D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_ENET_TX_DATA0 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: ENET_TXD0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: ENET_TXD0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: ENET_TXD0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: ENET_TXD0.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_TX_DATA0 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: ENET_TXD0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.370 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TX_DATA1)

Address: 20E_0000h base + 5D8h offset = 20E_05D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_ENET_TX_DATA1 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: ENET_TXD1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: ENET_TXD1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: ENET_TXD1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: ENET_TXD1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_TX_DATA1 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: ENET_TXD1. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.371 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO00)

Address: 20E_0000h base + 5DCh offset = 20E_05DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	0	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_GPIO00 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: GPIO_0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: GPIO_0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: GPIO_0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: GPIO_0. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. Read Only Field The value of this field is fixed and cannot be changed. 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: GPIO_0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO00 field descriptions (continued)

Field	Description
	011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.372 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO01)

Address: 20E_0000h base + 5E0h offset = 20E_05E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_GPIO01 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: GPIO_1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: GPIO_1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO01 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: GPIO_1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: GPIO_1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: GPIO_1. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO01 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.373 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO16)

Address: 20E_0000h base + 5E4h offset = 20E_05E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_GPIO16 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: GPIO_16. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: GPIO_16. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: GPIO_16. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: GPIO_16.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO16 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: GPIO_16. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.374 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO17)

Address: 20E_0000h base + 5E8h offset = 20E_05E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_GPIO17 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: GPIO_17. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: GPIO_17. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: GPIO_17. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: GPIO_17. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO17 field descriptions (continued)

Field	Description
	0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. Read Only Field The value of this field is fixed and cannot be changed. 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: GPIO_17. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.375 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO18)

Address: 20E_0000h base + 5ECh offset = 20E_05ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_GPIO18 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: GPIO_18. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: GPIO_18. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: GPIO_18. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: GPIO_18. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: GPIO_18. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO18 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.376 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO19)

Address: 20E_0000h base + 5F0h offset = 20E_05F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_GPIO19 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: GPIO_19. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: GPIO_19. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO19 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: GPIO_19. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: GPIO_19. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: GPIO_19. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO19 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.377 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO02)

Address: 20E_0000h base + 5F4h offset = 20E_05F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_GPIO02 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: GPIO_2. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: GPIO_2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: GPIO_2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: GPIO_2.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO02 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: GPIO_2. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.378 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO03)

Address: 20E_0000h base + 5F8h offset = 20E_05F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_GPIO03 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: GPIO_3. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: GPIO_3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: GPIO_3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: GPIO_3. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO03 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: GPIO_3. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.379 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO04)

Address: 20E_0000h base + 5FCh offset = 20E_05FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_GPIO04 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: GPIO_4. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: GPIO_4. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: GPIO_4. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: GPIO_4. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: GPIO_4. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO04 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.380 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO05)

Address: 20E_0000h base + 600h offset = 20E_0600h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_GPIO05 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: GPIO_5. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: GPIO_5. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO05 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: GPIO_5. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: GPIO_5. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: GPIO_5. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO05 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.381 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO06)

Address: 20E_0000h base + 604h offset = 20E_0604h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_GPIO06 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: GPIO_6. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: GPIO_6. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: GPIO_6. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: GPIO_6.

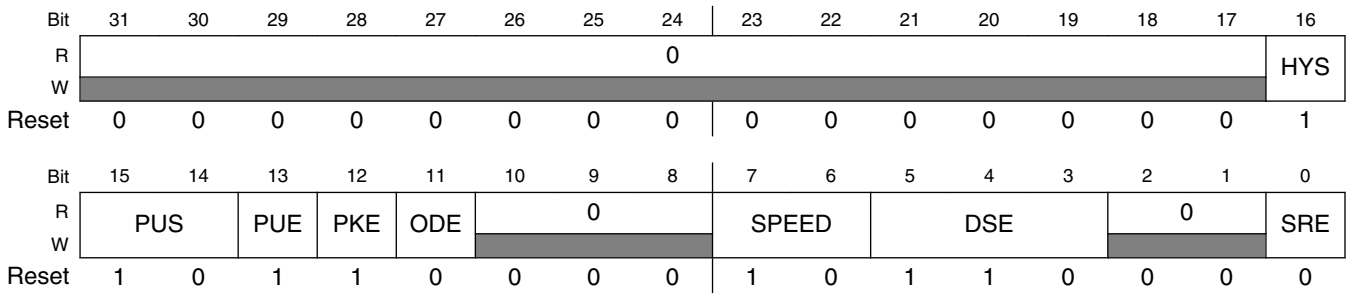
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO06 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: GPIO_6. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.382 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO07)

Address: 20E_0000h base + 608h offset = 20E_0608h



IOMUXC_SW_PAD_CTL_PAD_GPIO07 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: GPIO_7. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: GPIO_7. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: GPIO_7. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: GPIO_7. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO07 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: GPIO_7. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.383 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO08)

Address: 20E_0000h base + 60Ch offset = 20E_060Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_GPIO08 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: GPIO_8. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: GPIO_8. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: GPIO_8. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: GPIO_8. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: GPIO_8. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO08 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.384 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO09)

Address: 20E_0000h base + 610h offset = 20E_0610h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_GPIO09 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: GPIO_9. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: GPIO_9. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO09 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: GPIO_9. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: GPIO_9. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: GPIO_9. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO09 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.385 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD)

Address: 20E_0000h base + 614h offset = 20E_0614h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: JTAG_MOD. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: JTAG_MOD. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Read Only Field The value of this field is fixed and cannot be changed. 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: JTAG_MOD.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. Read Only Field The value of this field is fixed and cannot be changed. 0 **DISABLED** — Output is CMOS.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field Read Only Field The value of this field is fixed and cannot be changed. 01 **50MHZ** — Low (50 MHz)
5–3 DSE	Drive Strength Field Read Only Field The value of this field is fixed and cannot be changed. 100 **60_OHM** — 60 Ohm
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. Read Only Field The value of this field is fixed and cannot be changed. 0 **SLOW** — Slow Slew Rate

37.4.386 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK)

Address: 20E_0000h base + 618h offset = 20E_0618h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: JTAG_TCK. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: JTAG_TCK. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Read Only Field The value of this field is fixed and cannot be changed. 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: JTAG_TCK. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. Read Only Field

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK field descriptions (continued)

Field	Description
	The value of this field is fixed and cannot be changed. 0 **DISABLED** — Output is CMOS.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field Read Only Field The value of this field is fixed and cannot be changed. 01 **50MHZ** — Low (50 MHz)
5–3 DSE	Drive Strength Field Read Only Field The value of this field is fixed and cannot be changed. 100 **60_OHM** — 60 Ohm
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. Read Only Field The value of this field is fixed and cannot be changed. 0 **SLOW** — Slow Slew Rate

37.4.387 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI)

Address: 20E_0000h base + 61Ch offset = 20E_061Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: JTAG_TDI. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: JTAG_TDI. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Read Only Field The value of this field is fixed and cannot be changed. 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: JTAG_TDI. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. Read Only Field The value of this field is fixed and cannot be changed. 0 **DISABLED** — Output is CMOS.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field Read Only Field The value of this field is fixed and cannot be changed. 01 **50MHZ** — Low (50 MHz)
5–3 DSE	Drive Strength Field Read Only Field The value of this field is fixed and cannot be changed. 100 **60_OHM** — 60 Ohm
2–1 Reserved	This read-only field is reserved and always has the value 0.

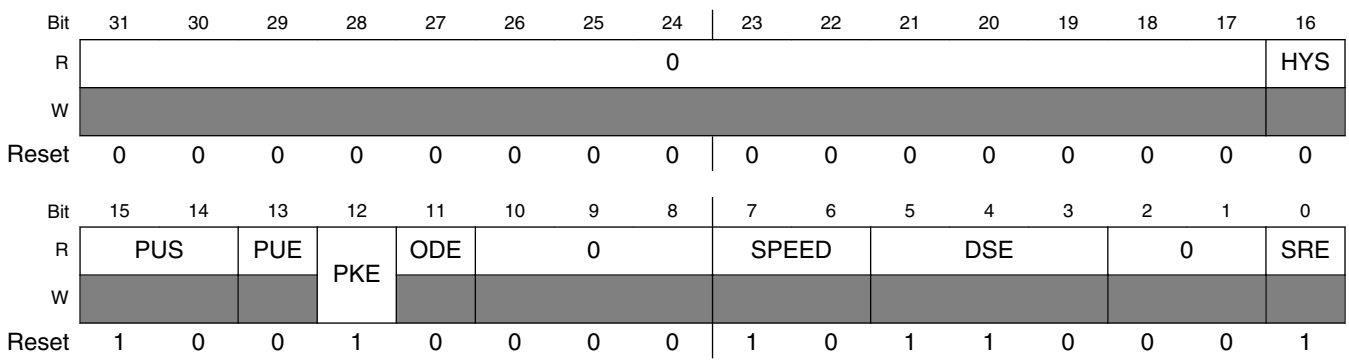
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI field descriptions (continued)

Field	Description
0 SRE	Slew Rate Field Slew rate control. Read Only Field The value of this field is fixed and cannot be changed. 0 **SLOW** — Slow Slew Rate

37.4.388 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO)

Address: 20E_0000h base + 620h offset = 20E_0620h



IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Read Only Field The value of this field is fixed and cannot be changed. 0 **DISABLED** — CMOS input
15–14 PUS	Pull Up / Down Config. Field Read Only Field The value of this field is fixed and cannot be changed. 10 **100K_OHM_PU** — 100K Ohm Pull Up
13 PUE	Pull / Keep Select Field Read Only Field The value of this field is fixed and cannot be changed.

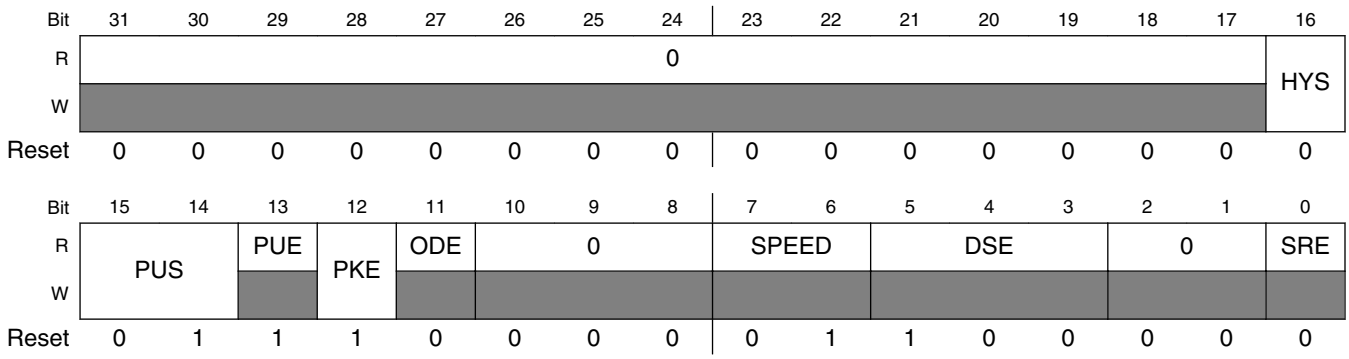
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO field descriptions (continued)

Field	Description
	0 **KEEP** — Keeper Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: JTAG_TDO. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. Read Only Field The value of this field is fixed and cannot be changed. 0 **DISABLED** — Output is CMOS.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field Read Only Field The value of this field is fixed and cannot be changed. 10 **100MHZ** — Medium (100 MHz)
5–3 DSE	Drive Strength Field Read Only Field The value of this field is fixed and cannot be changed. 110 **40_OHM** — 40 Ohm
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. Read Only Field The value of this field is fixed and cannot be changed. 1 **FAST** — Fast Slew Rate

37.4.389 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS)

Address: 20E_0000h base + 624h offset = 20E_0624h



IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: JTAG_TMS. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: JTAG_TMS. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Read Only Field The value of this field is fixed and cannot be changed. 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: JTAG_TMS. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. Read Only Field

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS field descriptions (continued)

Field	Description
	The value of this field is fixed and cannot be changed. 0 **DISABLED** — Output is CMOS.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field Read Only Field The value of this field is fixed and cannot be changed. 01 **50MHZ** — Low (50 MHz)
5–3 DSE	Drive Strength Field Read Only Field The value of this field is fixed and cannot be changed. 100 **60_OHM** — 60 Ohm
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. Read Only Field The value of this field is fixed and cannot be changed. 0 **SLOW** — Slow Slew Rate

37.4.390 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TRSTB)

Address: 20E_0000h base + 628h offset = 20E_0628h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_JTAG_TRSTB field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: JTAG_TRSTB. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: JTAG_TRSTB. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Read Only Field The value of this field is fixed and cannot be changed. 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: JTAG_TRSTB. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. Read Only Field The value of this field is fixed and cannot be changed. 0 **DISABLED** — Output is CMOS.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field Read Only Field The value of this field is fixed and cannot be changed. 01 **50MHZ** — Low (50 MHz)
5–3 DSE	Drive Strength Field Read Only Field The value of this field is fixed and cannot be changed. 100 **60_OHM** — 60 Ohm
2–1 Reserved	This read-only field is reserved and always has the value 0.

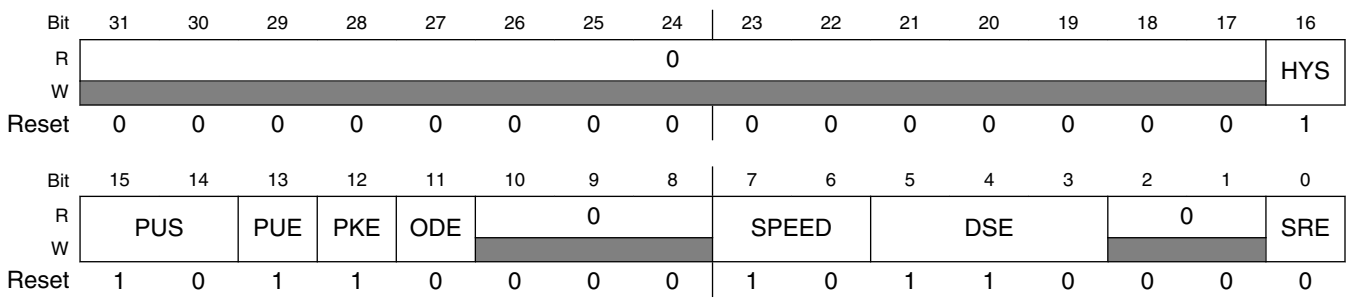
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_JTAG_TRSTB field descriptions (continued)

Field	Description
0 SRE	Slew Rate Field Slew rate control. Read Only Field The value of this field is fixed and cannot be changed. 0 **SLOW** — Slow Slew Rate

37.4.391 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_COL0)

Address: 20E_0000h base + 62Ch offset = 20E_062Ch



IOMUXC_SW_PAD_CTL_PAD_KEY_COL0 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: KEY_COL0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: KEY_COL0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: KEY_COL0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_COL0 field descriptions (continued)

Field	Description
12 PKE	Pull / Keep Enable Field Select one of next values for pad: KEY_COL0. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: KEY_COL0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.392 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_COL1)

Address: 20E_0000h base + 630h offset = 20E_0630h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_KEY_COL1 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: KEY_COL1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: KEY_COL1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: KEY_COL1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: KEY_COL1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_COL1 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: KEY_COL1. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.393 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_COL2)

Address: 20E_0000h base + 634h offset = 20E_0634h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_KEY_COL2 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: KEY_COL2. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: KEY_COL2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: KEY_COL2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: KEY_COL2. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: KEY_COL2. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_COL2 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.394 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_COL3)

Address: 20E_0000h base + 638h offset = 20E_0638h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_KEY_COL3 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: KEY_COL3. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: KEY_COL3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_COL3 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: KEY_COL3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: KEY_COL3. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: KEY_COL3. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

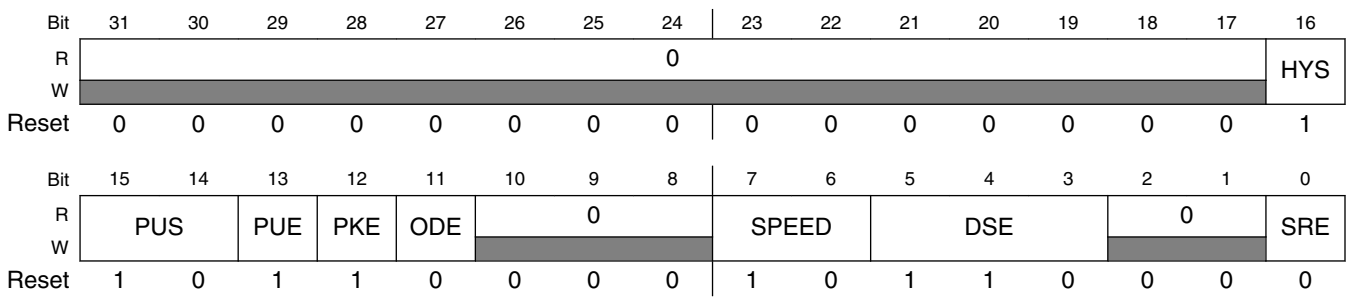
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_COL3 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.395 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_COL4)

Address: 20E_0000h base + 63Ch offset = 20E_063Ch



IOMUXC_SW_PAD_CTL_PAD_KEY_COL4 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: KEY_COL4. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: KEY_COL4. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: KEY_COL4. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: KEY_COL4.

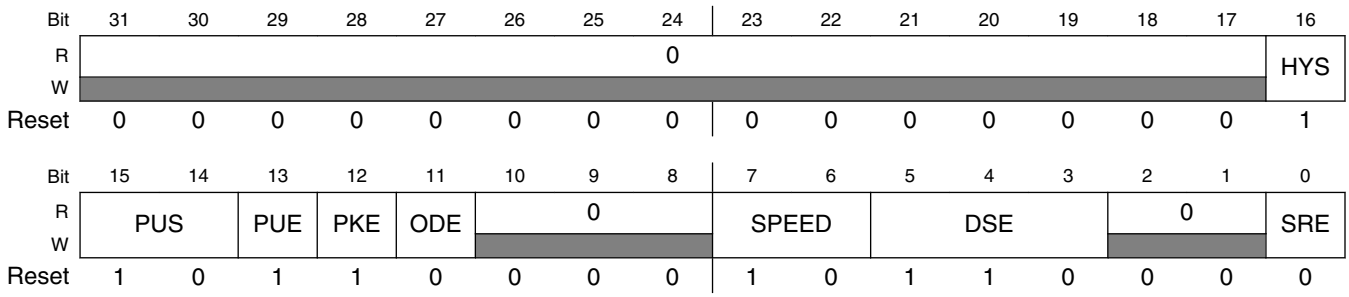
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_COL4 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: KEY_COL4. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.396 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW0)

Address: 20E_0000h base + 640h offset = 20E_0640h



IOMUXC_SW_PAD_CTL_PAD_KEY_ROW0 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: KEY_ROW0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: KEY_ROW0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: KEY_ROW0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: KEY_ROW0. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW0 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: KEY_ROW0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.397 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW1)

Address: 20E_0000h base + 644h offset = 20E_0644h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW1 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: KEY_ROW1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: KEY_ROW1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: KEY_ROW1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: KEY_ROW1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: KEY_ROW1. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW1 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.398 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW2)

Address: 20E_0000h base + 648h offset = 20E_0648h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW2 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: KEY_ROW2. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: KEY_ROW2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW2 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: KEY_ROW2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: KEY_ROW2. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: KEY_ROW2. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

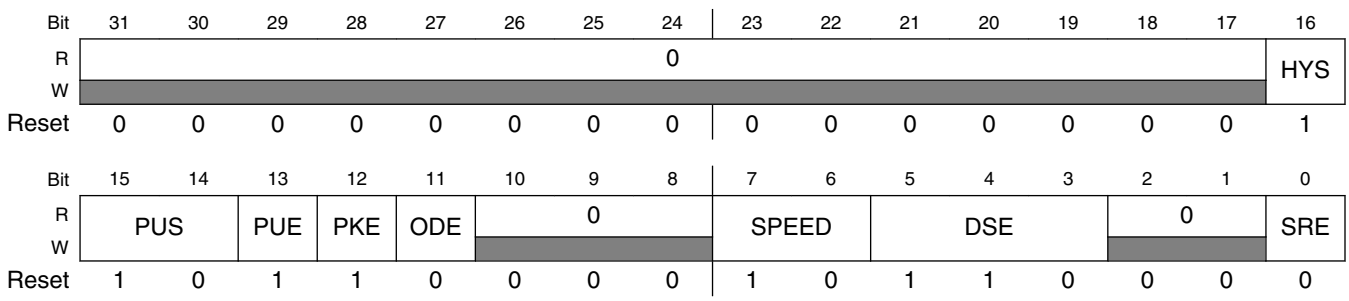
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW2 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.399 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW3)

Address: 20E_0000h base + 64Ch offset = 20E_064Ch



IOMUXC_SW_PAD_CTL_PAD_KEY_ROW3 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: KEY_ROW3. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: KEY_ROW3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: KEY_ROW3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: KEY_ROW3.

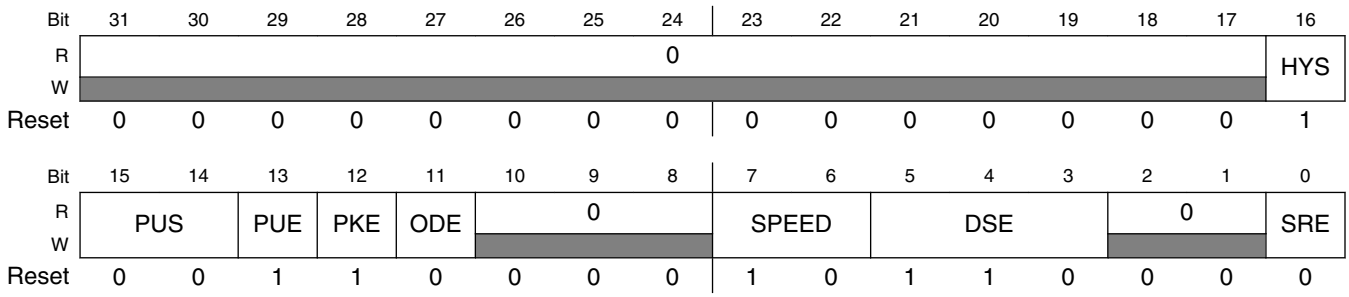
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW3 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: KEY_ROW3. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.400 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW4)

Address: 20E_0000h base + 650h offset = 20E_0650h



IOMUXC_SW_PAD_CTL_PAD_KEY_ROW4 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: KEY_ROW4. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: KEY_ROW4. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: KEY_ROW4. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: KEY_ROW4. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW4 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: KEY_ROW4. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.401 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_ALE)

Address: 20E_0000h base + 654h offset = 20E_0654h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_NAND_ALE field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_ALE. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_ALE. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_ALE. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_ALE. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_ALE. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_ALE field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.402 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CLE)

Address: 20E_0000h base + 658h offset = 20E_0658h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_NAND_CLE field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_CLE. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_CLE. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_CLE field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_CLE. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_CLE. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_CLE. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_CLE field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.403 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CS0_B)

Address: 20E_0000h base + 65Ch offset = 20E_065Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_NAND_CS0_B field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_CS0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_CS0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_CS0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_CS0.

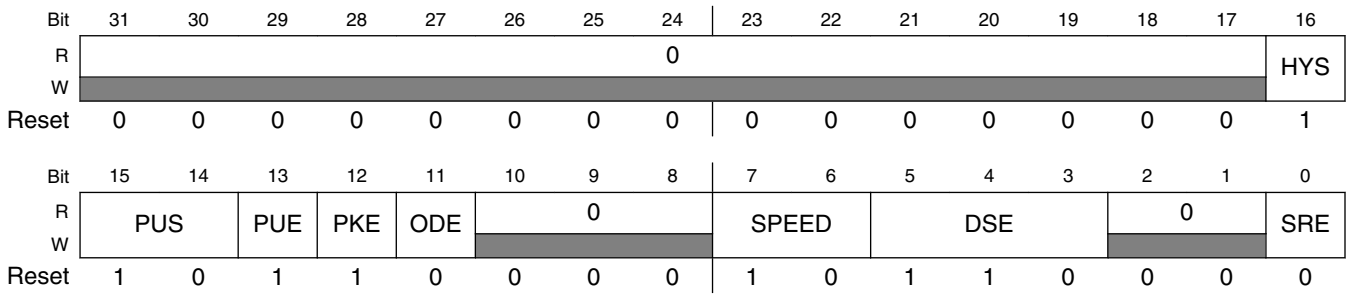
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_CS0_B field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_CS0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.404 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CS1_B)

Address: 20E_0000h base + 660h offset = 20E_0660h



IOMUXC_SW_PAD_CTL_PAD_NAND_CS1_B field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_CS1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_CS1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_CS1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_CS1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_CS1_B field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_CS1. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.405 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CS2_B)

Address: 20E_0000h base + 664h offset = 20E_0664h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_NAND_CS2_B field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_CS2. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_CS2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_CS2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_CS2. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_CS2. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_CS2_B field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.406 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CS3_B)

Address: 20E_0000h base + 668h offset = 20E_0668h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_NAND_CS3_B field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_CS3. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_CS3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_CS3_B field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_CS3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_CS3. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_CS3. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_CS3_B field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.407 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA00)

Address: 20E_0000h base + 66Ch offset = 20E_066Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA00 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_D0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_D0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_D0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_D0.

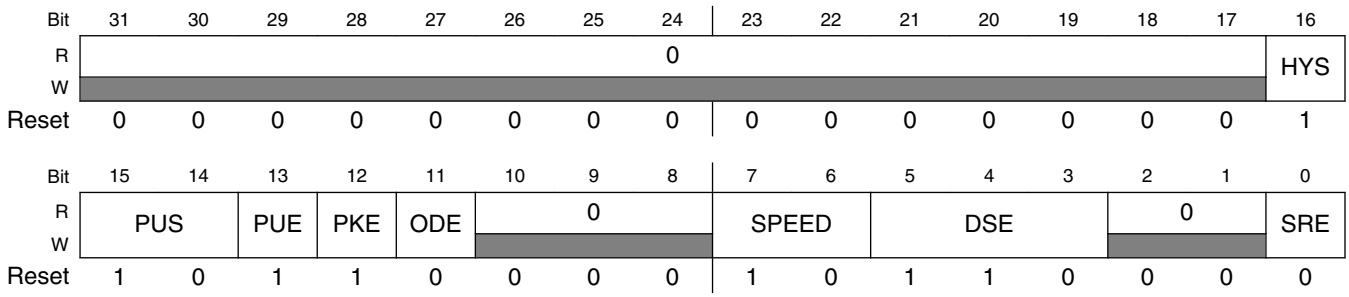
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA00 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_D0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.408 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA01)

Address: 20E_0000h base + 670h offset = 20E_0670h



IOMUXC_SW_PAD_CTL_PAD_NAND_DATA01 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_D1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_D1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_D1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_D1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA01 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_D1. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.409 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA02)

Address: 20E_0000h base + 674h offset = 20E_0674h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA02 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_D2. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_D2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_D2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_D2. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_D2. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA02 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.410 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA03)

Address: 20E_0000h base + 678h offset = 20E_0678h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA03 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_D3. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_D3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA03 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_D3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_D3. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_D3. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA03 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.411 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA04)

Address: 20E_0000h base + 67Ch offset = 20E_067Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA04 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_D4. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_D4. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_D4. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_D4.

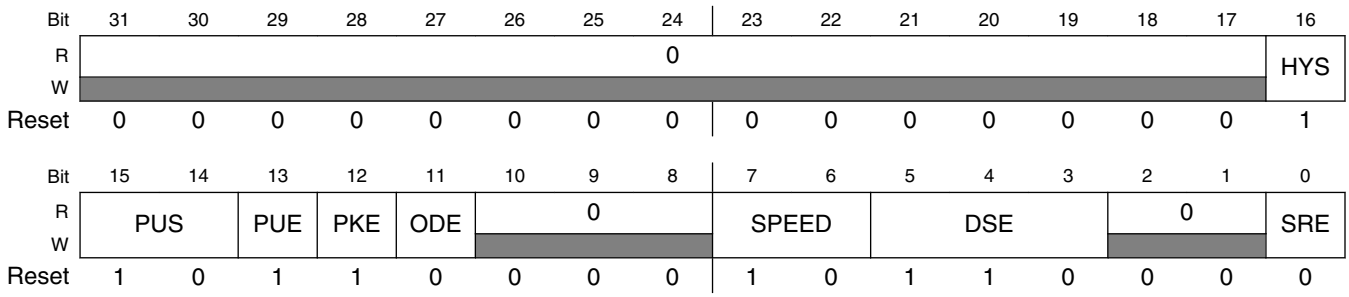
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA04 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_D4. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.412 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA05)

Address: 20E_0000h base + 680h offset = 20E_0680h



IOMUXC_SW_PAD_CTL_PAD_NAND_DATA05 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_D5. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_D5. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_D5. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_D5. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA05 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_D5. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.413 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA06)

Address: 20E_0000h base + 684h offset = 20E_0684h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA06 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_D6. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_D6. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_D6. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_D6. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_D6. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA06 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.414 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA07)

Address: 20E_0000h base + 688h offset = 20E_0688h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA07 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_D7. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_D7. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA07 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_D7. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_D7. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_D7. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA07 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.415 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_READY_B)

Address: 20E_0000h base + 68Ch offset = 20E_068Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_NAND_READY_B field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_RB0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_RB0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_RB0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_RB0.

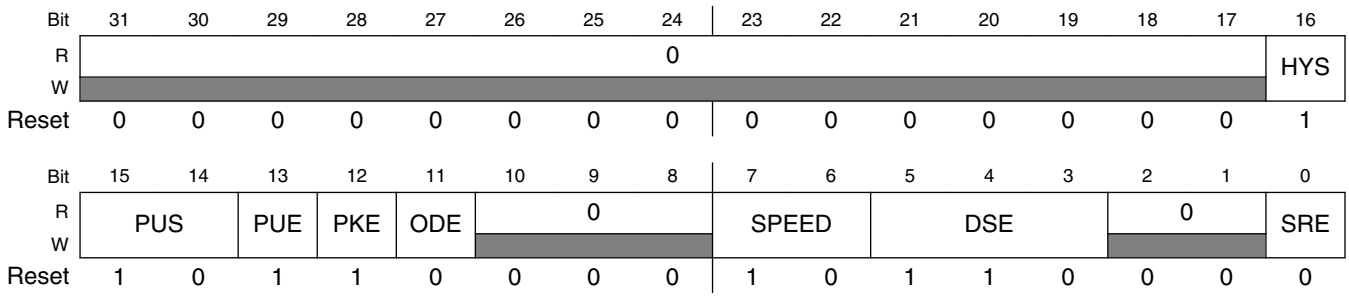
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_READY_B field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_RB0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.416 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_WP_B)

Address: 20E_0000h base + 690h offset = 20E_0690h



IOMUXC_SW_PAD_CTL_PAD_NAND_WP_B field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: NANDF_WP_B. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: NANDF_WP_B. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: NANDF_WP_B. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: NANDF_WP_B. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

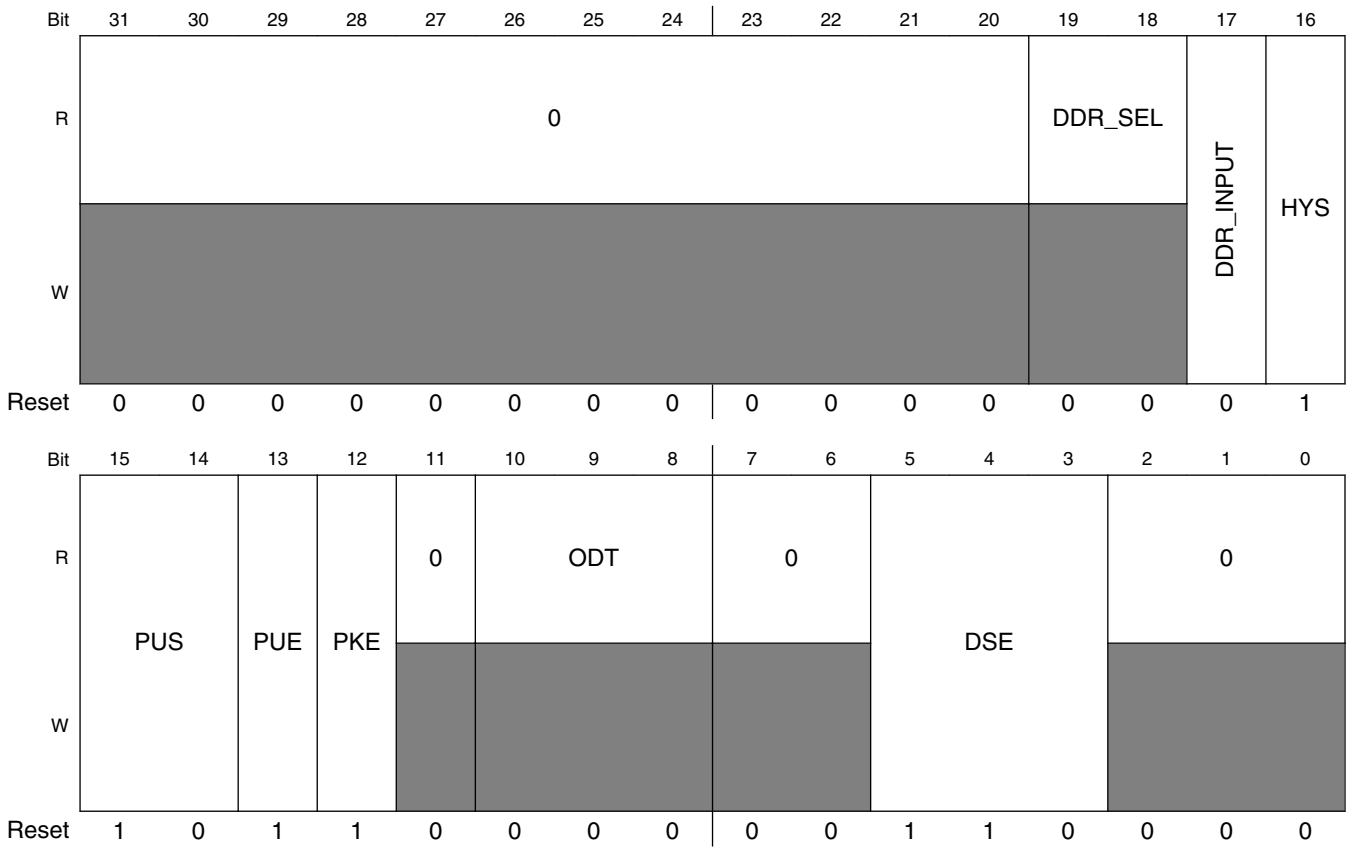
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_WP_B field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: NANDF_WP_B. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.417 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RD0)

Address: 20E_0000h base + 694h offset = 20E_0694h



IOMUXC_SW_PAD_CTL_PAD_RGMII_RD0 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: RGMII_RD0. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: RGMII_RD0.

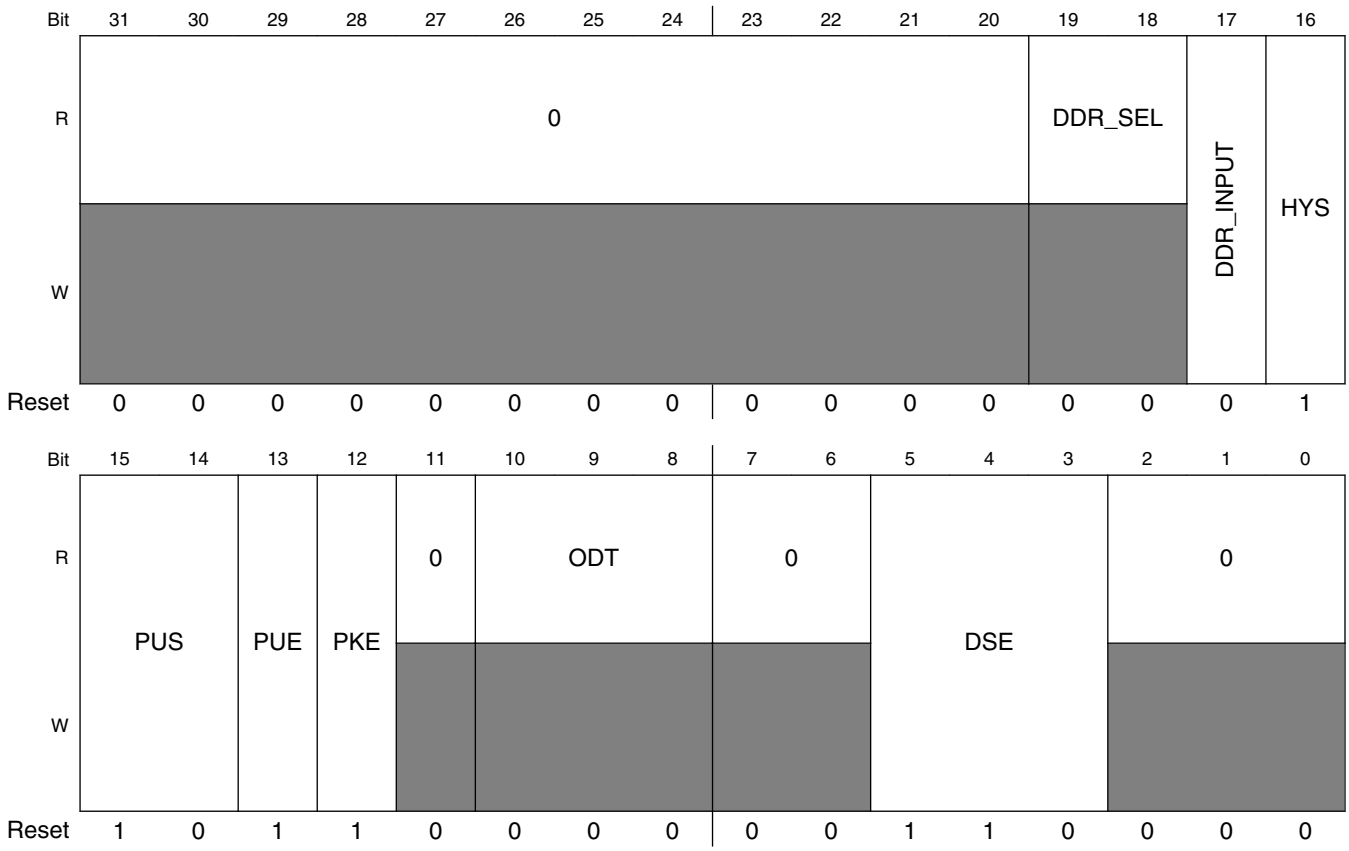
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RGMII_RD0 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: RGMII_RD0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: RGMII_RD0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: RGMII_RD0. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_RGMII_TERM Note: The value of this field does not reflect the value of the Group Control Register.
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: RGMII_RD0. 000 **HIZ** — HI-Z 001 **287_OHM** — 287/174 Ohm @ 2.5V, 247/201 Ohm pd/pu @ 1.8V 010 **121_OHM** — 121/85 Ohm @ 2.5V, 113/96 Ohm pd/pu @ 1.8V 011 **76_OHM** — 76/56 Ohm @ 2.5V, 73/63 Ohm pd/pu @ 1.8V 100 **57_OHM** — 57/43 Ohm @ 2.5V, 55/48 Ohm pd/pu @ 1.8V 101 **45_OHM** — 45/34 Ohm @ 2.5V, 43/38 Ohm pd/pu @ 1.8V 110 **37_OHM** — 37/28 Ohm @ 2.5V, 36/32 Ohm pd/pu @ 1.8V 111 **31_OHM** — 31/24 Ohm @ 2.5V, 30/27 Ohm pd/pu @ 1.8V
Reserved	This read-only field is reserved and always has the value 0.

37.4.418 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RD1)

Address: 20E_0000h base + 698h offset = 20E_0698h



IOMUXC_SW_PAD_CTL_PAD_RGMII_RD1 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: RGMII_RD1. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: RGMII_RD1.

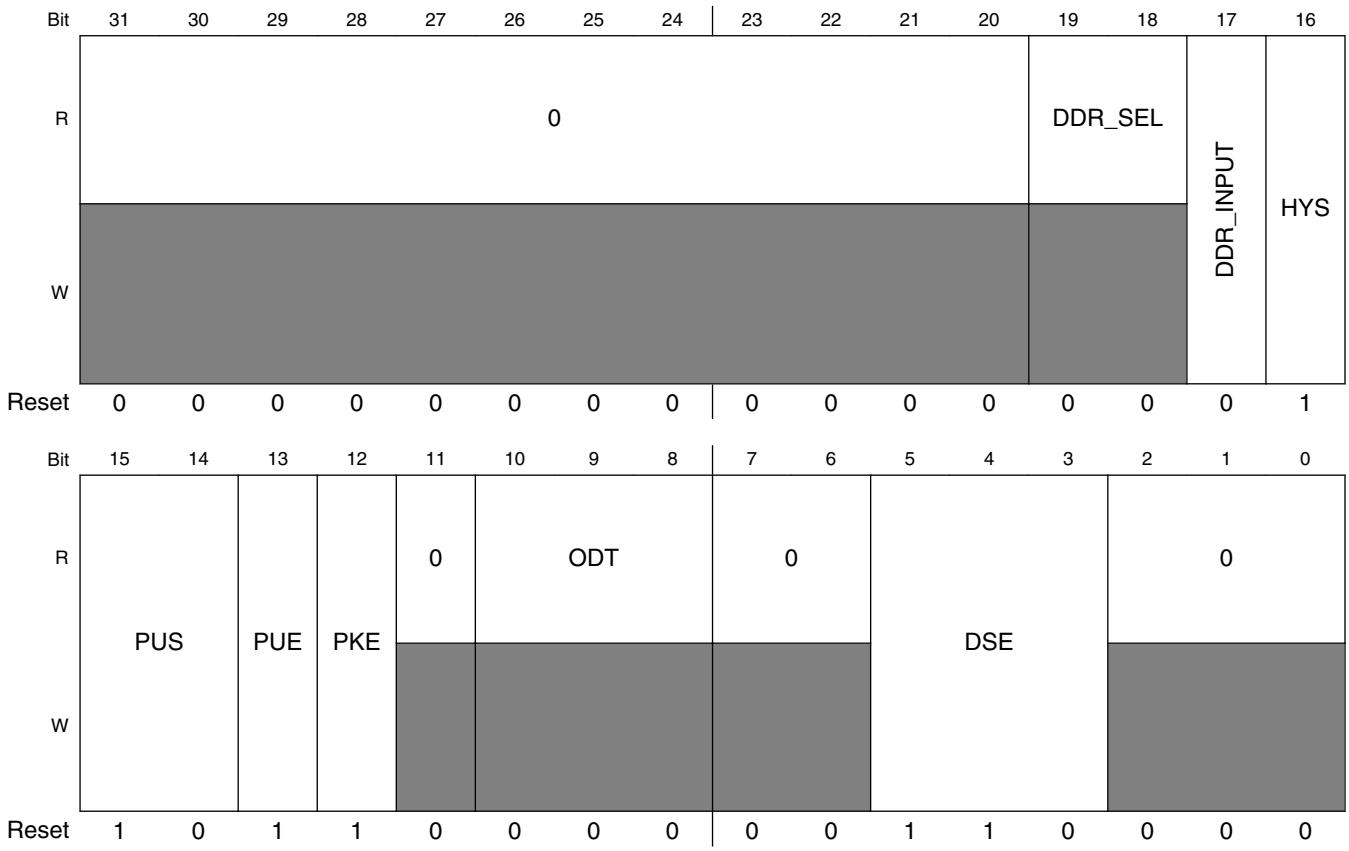
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RGMII_RD1 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: RGMII_RD1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: RGMII_RD1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: RGMII_RD1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_RGMII_TERM Note: The value of this field does not reflect the value of the Group Control Register.
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: RGMII_RD1. 000 **HIZ** — HI-Z 001 **287_OHM** — 287/174 Ohm @ 2.5V, 247/201 Ohm pd/pu @ 1.8V 010 **121_OHM** — 121/85 Ohm @ 2.5V, 113/96 Ohm pd/pu @ 1.8V 011 **76_OHM** — 76/56 Ohm @ 2.5V, 73/63 Ohm pd/pu @ 1.8V 100 **57_OHM** — 57/43 Ohm @ 2.5V, 55/48 Ohm pd/pu @ 1.8V 101 **45_OHM** — 45/34 Ohm @ 2.5V, 43/38 Ohm pd/pu @ 1.8V 110 **37_OHM** — 37/28 Ohm @ 2.5V, 36/32 Ohm pd/pu @ 1.8V 111 **31_OHM** — 31/24 Ohm @ 2.5V, 30/27 Ohm pd/pu @ 1.8V
Reserved	This read-only field is reserved and always has the value 0.

37.4.419 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RD2)

Address: 20E_0000h base + 69Ch offset = 20E_069Ch



IOMUXC_SW_PAD_CTL_PAD_RGMII_RD2 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: RGMII_RD2. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: RGMII_RD2.

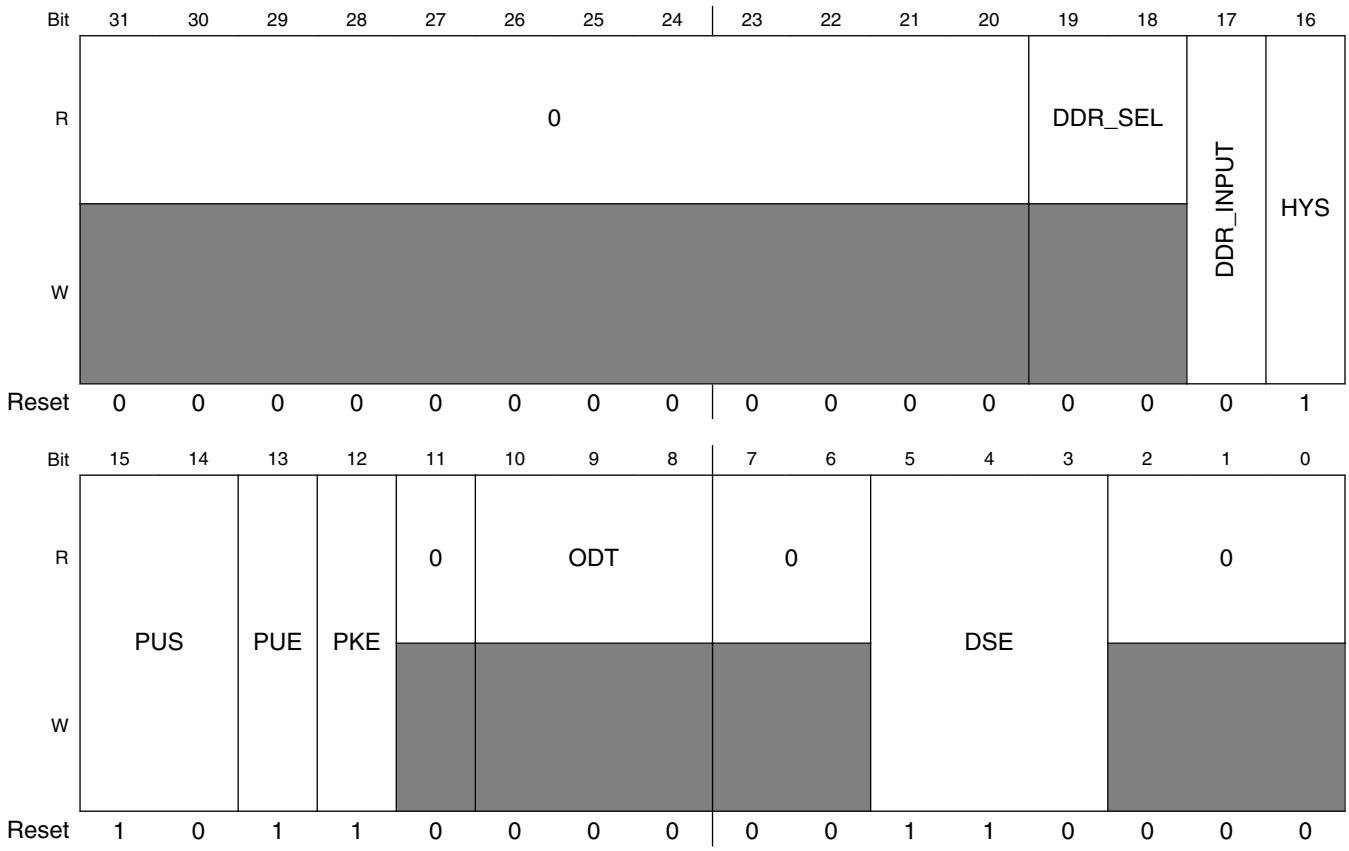
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RGMII_RD2 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: RGMII_RD2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: RGMII_RD2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: RGMII_RD2. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_RGMII_TERM Note: The value of this field does not reflect the value of the Group Control Register.
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: RGMII_RD2. 000 **HIZ** — HI-Z 001 **287_OHM** — 287/174 Ohm @ 2.5V, 247/201 Ohm pd/pu @ 1.8V 010 **121_OHM** — 121/85 Ohm @ 2.5V, 113/96 Ohm pd/pu @ 1.8V 011 **76_OHM** — 76/56 Ohm @ 2.5V, 73/63 Ohm pd/pu @ 1.8V 100 **57_OHM** — 57/43 Ohm @ 2.5V, 55/48 Ohm pd/pu @ 1.8V 101 **45_OHM** — 45/34 Ohm @ 2.5V, 43/38 Ohm pd/pu @ 1.8V 110 **37_OHM** — 37/28 Ohm @ 2.5V, 36/32 Ohm pd/pu @ 1.8V 111 **31_OHM** — 31/24 Ohm @ 2.5V, 30/27 Ohm pd/pu @ 1.8V
Reserved	This read-only field is reserved and always has the value 0.

37.4.420 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RD3)

Address: 20E_0000h base + 6A0h offset = 20E_06A0h



IOMUXC_SW_PAD_CTL_PAD_RGMII_RD3 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: RGMII_RD3. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: RGMII_RD3.

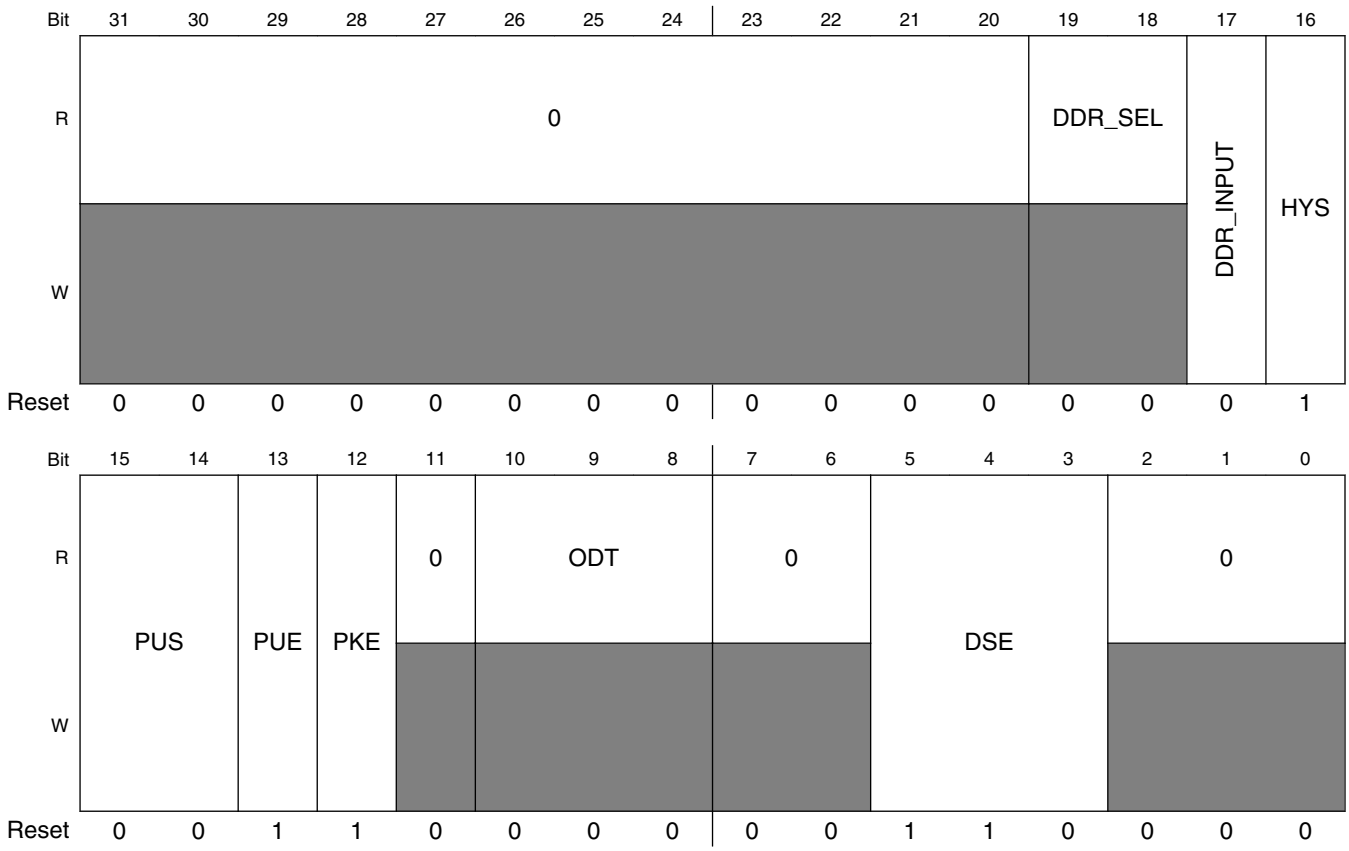
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RGMII_RD3 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: RGMII_RD3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: RGMII_RD3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: RGMII_RD3. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_RGMII_TERM Note: The value of this field does not reflect the value of the Group Control Register.
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: RGMII_RD3. 000 **HIZ** — HI-Z 001 **287_OHM** — 287/174 Ohm @ 2.5V, 247/201 Ohm pd/pu @ 1.8V 010 **121_OHM** — 121/85 Ohm @ 2.5V, 113/96 Ohm pd/pu @ 1.8V 011 **76_OHM** — 76/56 Ohm @ 2.5V, 73/63 Ohm pd/pu @ 1.8V 100 **57_OHM** — 57/43 Ohm @ 2.5V, 55/48 Ohm pd/pu @ 1.8V 101 **45_OHM** — 45/34 Ohm @ 2.5V, 43/38 Ohm pd/pu @ 1.8V 110 **37_OHM** — 37/28 Ohm @ 2.5V, 36/32 Ohm pd/pu @ 1.8V 111 **31_OHM** — 31/24 Ohm @ 2.5V, 30/27 Ohm pd/pu @ 1.8V
Reserved	This read-only field is reserved and always has the value 0.

37.4.421 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RX_CTL)

Address: 20E_0000h base + 6A4h offset = 20E_06A4h



IOMUXC_SW_PAD_CTL_PAD_RGMII_RX_CTL field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: RGMII_RX_CTL. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: RGMII_RX_CTL.

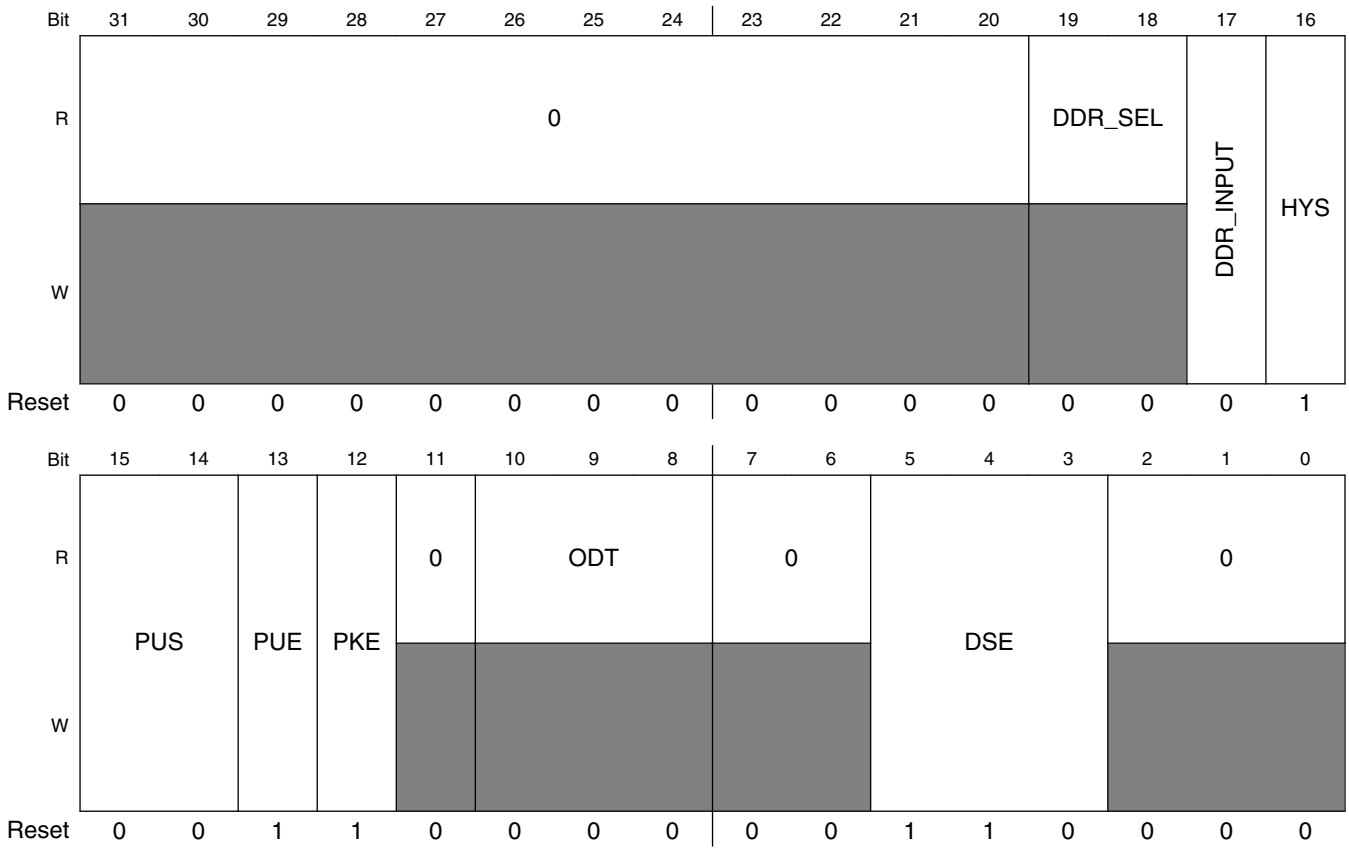
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RGMII_RX_CTL field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: RGMII_RX_CTL. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: RGMII_RX_CTL. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: RGMII_RX_CTL. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_RGMII_TERM Note: The value of this field does not reflect the value of the Group Control Register.
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: RGMII_RX_CTL. 000 **HIZ** — HI-Z 001 **287_OHM** — 287/174 Ohm @ 2.5V, 247/201 Ohm pd/pu @ 1.8V 010 **121_OHM** — 121/85 Ohm @ 2.5V, 113/96 Ohm pd/pu @ 1.8V 011 **76_OHM** — 76/56 Ohm @ 2.5V, 73/63 Ohm pd/pu @ 1.8V 100 **57_OHM** — 57/43 Ohm @ 2.5V, 55/48 Ohm pd/pu @ 1.8V 101 **45_OHM** — 45/34 Ohm @ 2.5V, 43/38 Ohm pd/pu @ 1.8V 110 **37_OHM** — 37/28 Ohm @ 2.5V, 36/32 Ohm pd/pu @ 1.8V 111 **31_OHM** — 31/24 Ohm @ 2.5V, 30/27 Ohm pd/pu @ 1.8V
Reserved	This read-only field is reserved and always has the value 0.

37.4.422 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_RXC)

Address: 20E_0000h base + 6A8h offset = 20E_06A8h



IOMUXC_SW_PAD_CTL_PAD_RGMII_RXC field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: RGMII_RXC. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: RGMII_RXC.

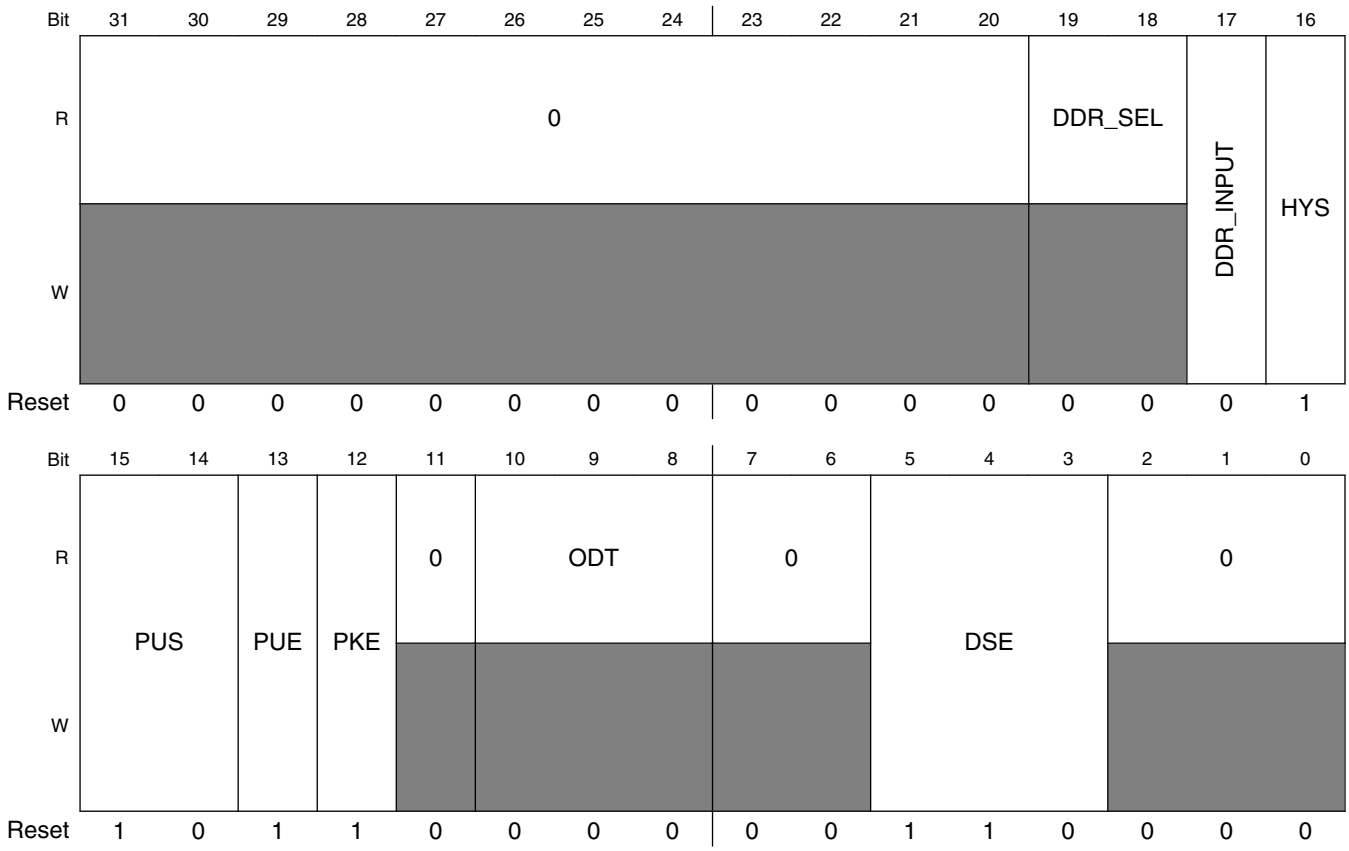
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RGMII_RXC field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: RGMII_RXC. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: RGMII_RXC. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: RGMII_RXC. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_RGMII_TERM Note: The value of this field does not reflect the value of the Group Control Register.
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: RGMII_RXC. 000 **HIZ** — HI-Z 001 **287_OHM** — 287/174 Ohm @ 2.5V, 247/201 Ohm pd/pu @ 1.8V 010 **121_OHM** — 121/85 Ohm @ 2.5V, 113/96 Ohm pd/pu @ 1.8V 011 **76_OHM** — 76/56 Ohm @ 2.5V, 73/63 Ohm pd/pu @ 1.8V 100 **57_OHM** — 57/43 Ohm @ 2.5V, 55/48 Ohm pd/pu @ 1.8V 101 **45_OHM** — 45/34 Ohm @ 2.5V, 43/38 Ohm pd/pu @ 1.8V 110 **37_OHM** — 37/28 Ohm @ 2.5V, 36/32 Ohm pd/pu @ 1.8V 111 **31_OHM** — 31/24 Ohm @ 2.5V, 30/27 Ohm pd/pu @ 1.8V
Reserved	This read-only field is reserved and always has the value 0.

37.4.423 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TD0)

Address: 20E_0000h base + 6ACh offset = 20E_06ACh



IOMUXC_SW_PAD_CTL_PAD_RGMII_TD0 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: RGMII_TD0. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: RGMII_TD0.

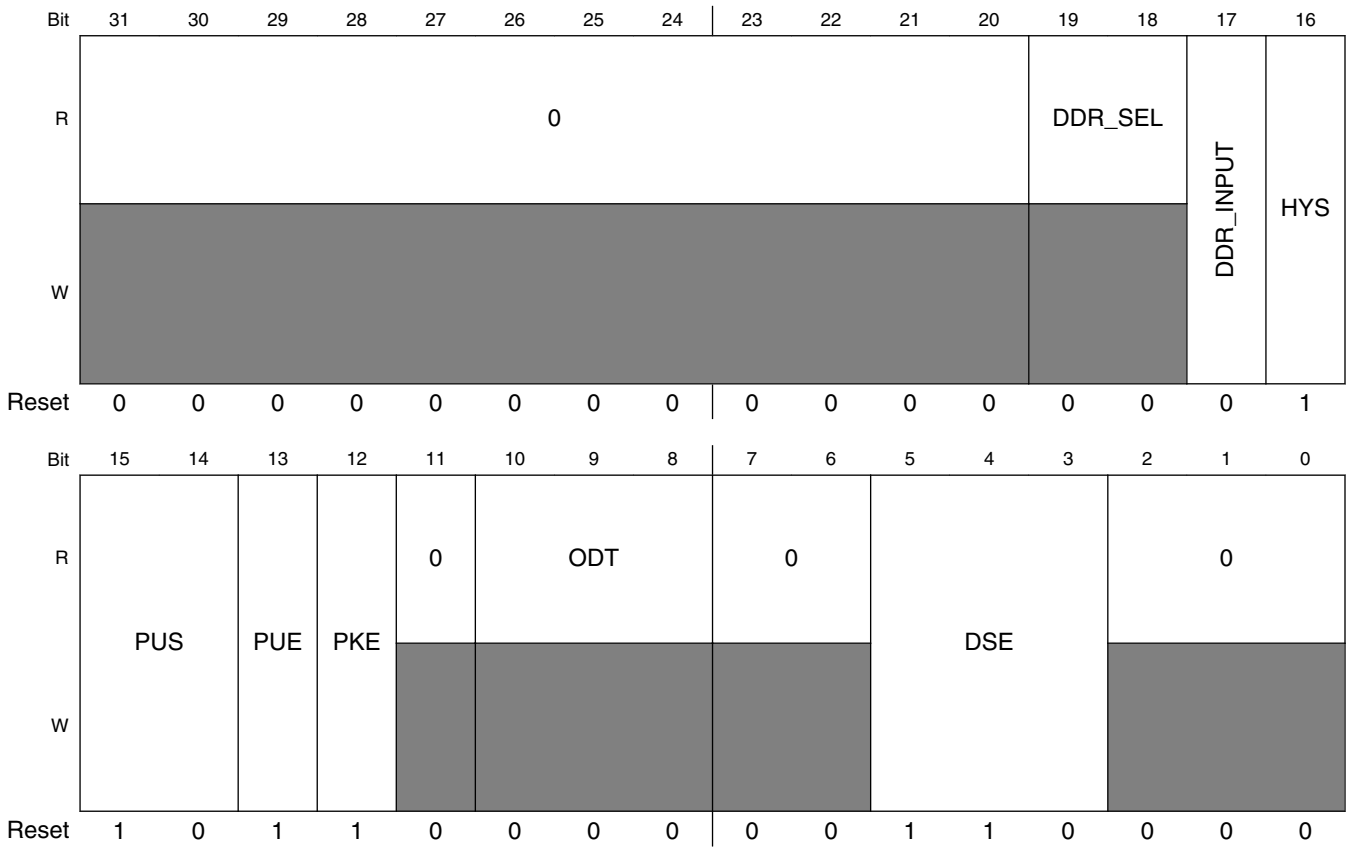
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RGMII_TD0 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: RGMII_TD0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: RGMII_TD0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: RGMII_TD0. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Read Only Field The value of this field is fixed and cannot be changed. 000 **DISABLED** — Disabled
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: RGMII_TD0. 000 **HIZ** — HI-Z 001 **287_OHM** — 287/174 Ohm @ 2.5V, 247/201 Ohm pd/pu @ 1.8V 010 **121_OHM** — 121/85 Ohm @ 2.5V, 113/96 Ohm pd/pu @ 1.8V 011 **76_OHM** — 76/56 Ohm @ 2.5V, 73/63 Ohm pd/pu @ 1.8V 100 **57_OHM** — 57/43 Ohm @ 2.5V, 55/48 Ohm pd/pu @ 1.8V 101 **45_OHM** — 45/34 Ohm @ 2.5V, 43/38 Ohm pd/pu @ 1.8V 110 **37_OHM** — 37/28 Ohm @ 2.5V, 36/32 Ohm pd/pu @ 1.8V 111 **31_OHM** — 31/24 Ohm @ 2.5V, 30/27 Ohm pd/pu @ 1.8V
Reserved	This read-only field is reserved and always has the value 0.

37.4.424 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TD1)

Address: 20E_0000h base + 6B0h offset = 20E_06B0h



IOMUXC_SW_PAD_CTL_PAD_RGMII_TD1 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: RGMII_TD1. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: RGMII_TD1.

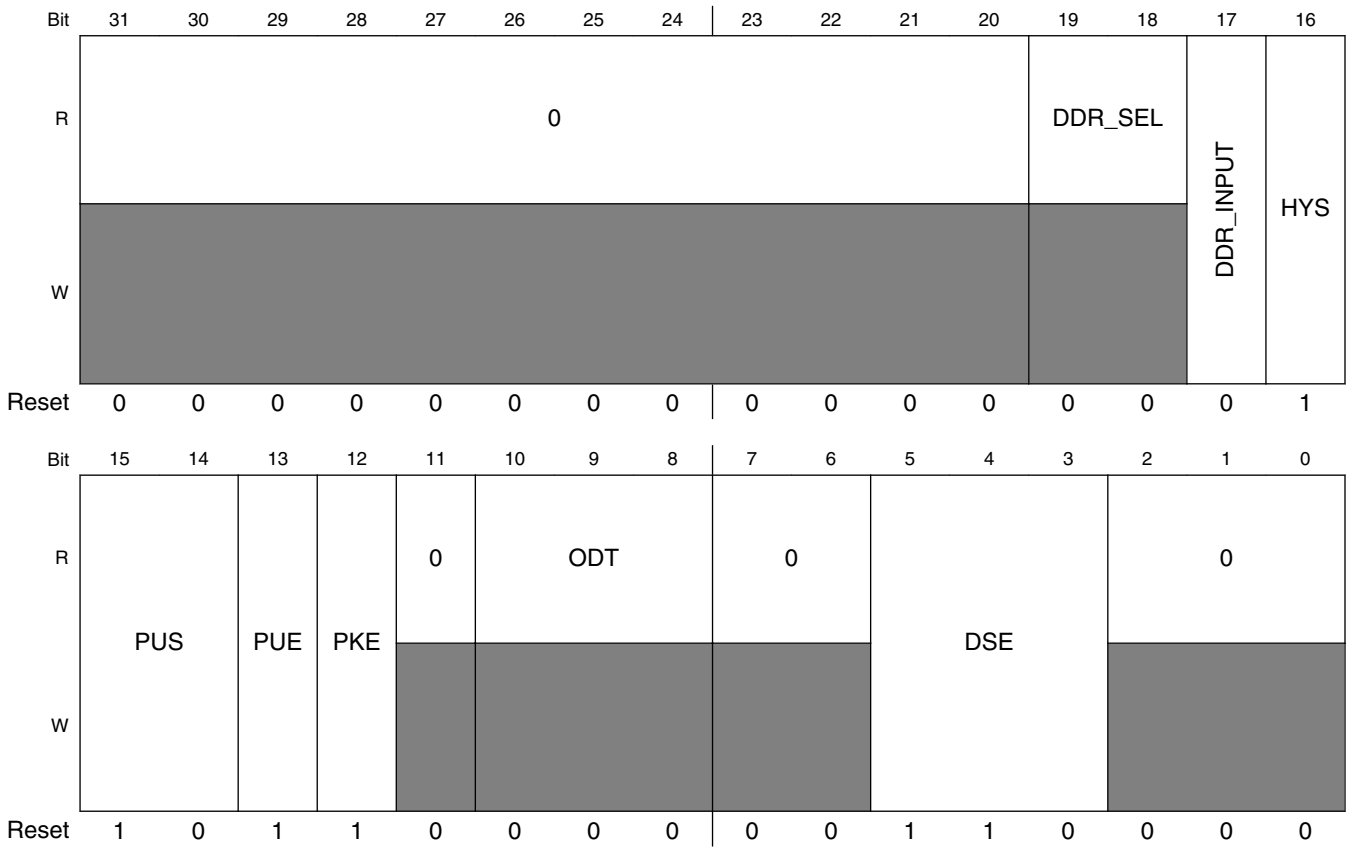
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RGMII_TD1 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: RGMII_TD1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: RGMII_TD1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: RGMII_TD1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Read Only Field The value of this field is fixed and cannot be changed. 000 **DISABLED** — Disabled
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: RGMII_TD1. 000 **HIZ** — HI-Z 001 **287_OHM** — 287/174 Ohm @ 2.5V, 247/201 Ohm pd/pu @ 1.8V 010 **121_OHM** — 121/85 Ohm @ 2.5V, 113/96 Ohm pd/pu @ 1.8V 011 **76_OHM** — 76/56 Ohm @ 2.5V, 73/63 Ohm pd/pu @ 1.8V 100 **57_OHM** — 57/43 Ohm @ 2.5V, 55/48 Ohm pd/pu @ 1.8V 101 **45_OHM** — 45/34 Ohm @ 2.5V, 43/38 Ohm pd/pu @ 1.8V 110 **37_OHM** — 37/28 Ohm @ 2.5V, 36/32 Ohm pd/pu @ 1.8V 111 **31_OHM** — 31/24 Ohm @ 2.5V, 30/27 Ohm pd/pu @ 1.8V
Reserved	This read-only field is reserved and always has the value 0.

37.4.425 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TD2)

Address: 20E_0000h base + 6B4h offset = 20E_06B4h



IOMUXC_SW_PAD_CTL_PAD_RGMII_TD2 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: RGMII_TD2. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: RGMII_TD2.

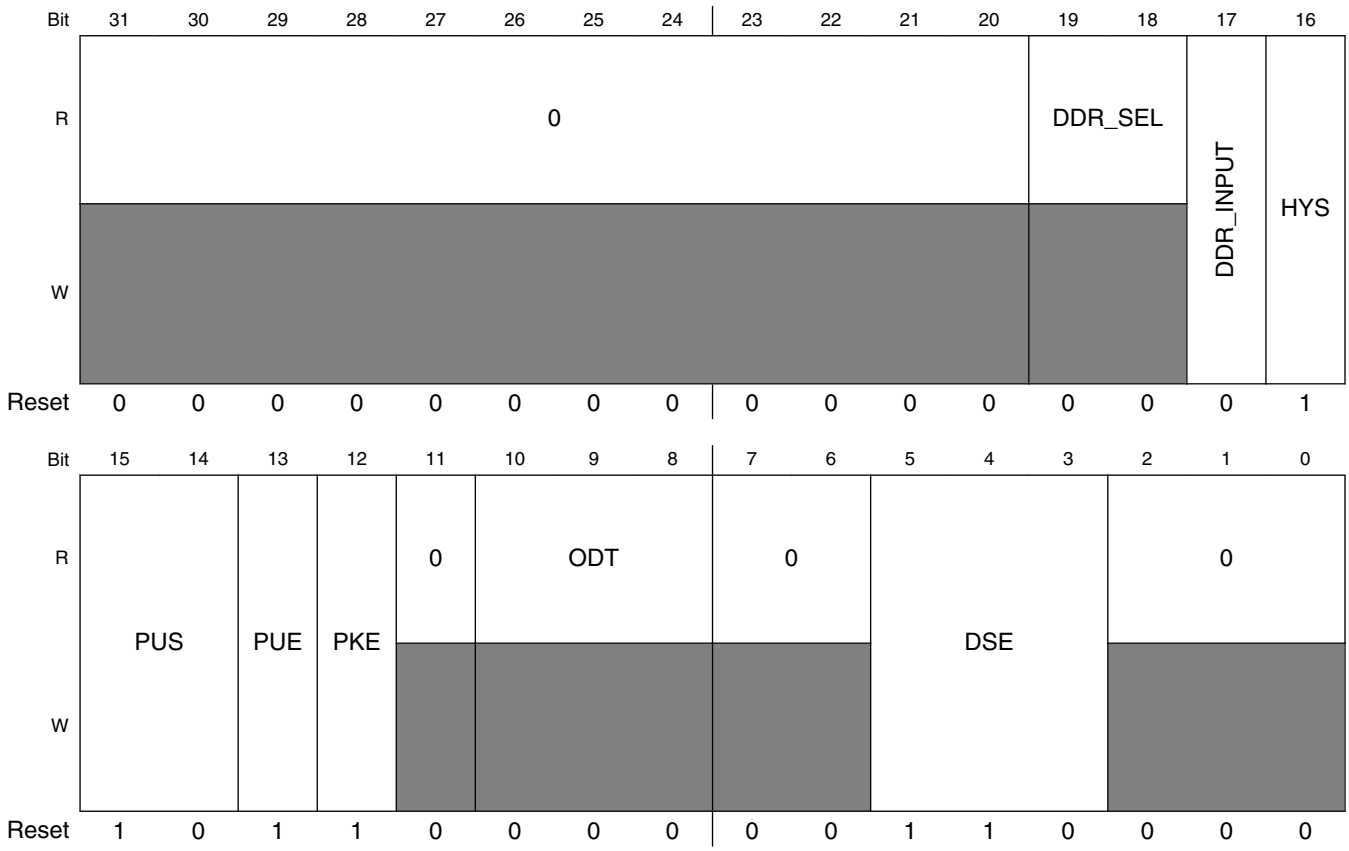
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RGMII_TD2 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: RGMII_TD2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: RGMII_TD2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: RGMII_TD2. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Read Only Field The value of this field is fixed and cannot be changed. 000 **DISABLED** — Disabled
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: RGMII_TD2. 000 **HIZ** — HI-Z 001 **287_OHM** — 287/174 Ohm @ 2.5V, 247/201 Ohm pd/pu @ 1.8V 010 **121_OHM** — 121/85 Ohm @ 2.5V, 113/96 Ohm pd/pu @ 1.8V 011 **76_OHM** — 76/56 Ohm @ 2.5V, 73/63 Ohm pd/pu @ 1.8V 100 **57_OHM** — 57/43 Ohm @ 2.5V, 55/48 Ohm pd/pu @ 1.8V 101 **45_OHM** — 45/34 Ohm @ 2.5V, 43/38 Ohm pd/pu @ 1.8V 110 **37_OHM** — 37/28 Ohm @ 2.5V, 36/32 Ohm pd/pu @ 1.8V 111 **31_OHM** — 31/24 Ohm @ 2.5V, 30/27 Ohm pd/pu @ 1.8V
Reserved	This read-only field is reserved and always has the value 0.

37.4.426 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TD3)

Address: 20E_0000h base + 6B8h offset = 20E_06B8h



IOMUXC_SW_PAD_CTL_PAD_RGMII_TD3 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for pad: RGMII_TD3. 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: RGMII_TD3.

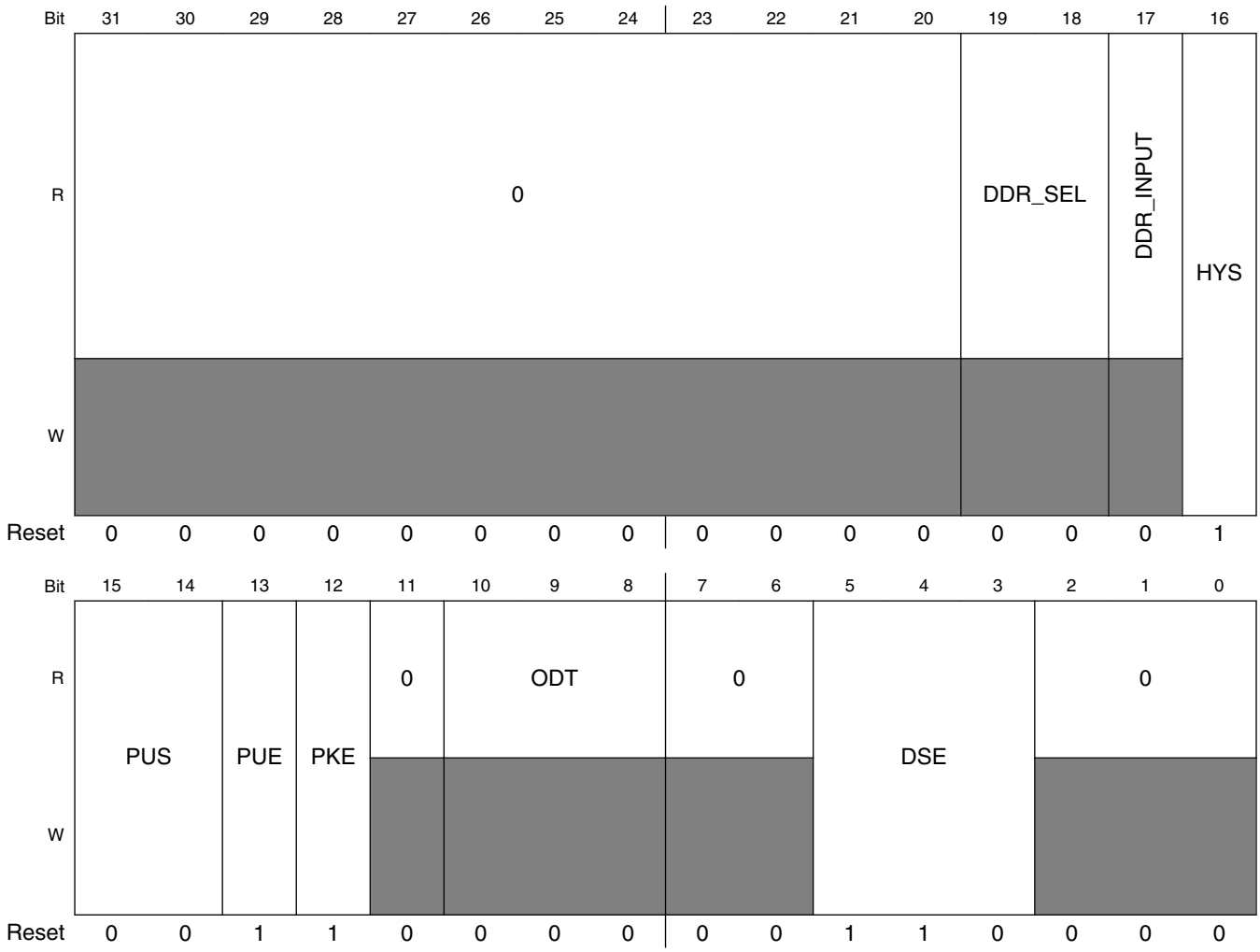
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RGMII_TD3 field descriptions (continued)

Field	Description
	0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: RGMII_TD3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: RGMII_TD3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: RGMII_TD3. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Read Only Field The value of this field is fixed and cannot be changed. 000 **DISABLED** — Disabled
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: RGMII_TD3. 000 **HIZ** — HI-Z 001 **287_OHM** — 287/174 Ohm @ 2.5V, 247/201 Ohm pd/pu @ 1.8V 010 **121_OHM** — 121/85 Ohm @ 2.5V, 113/96 Ohm pd/pu @ 1.8V 011 **76_OHM** — 76/56 Ohm @ 2.5V, 73/63 Ohm pd/pu @ 1.8V 100 **57_OHM** — 57/43 Ohm @ 2.5V, 55/48 Ohm pd/pu @ 1.8V 101 **45_OHM** — 45/34 Ohm @ 2.5V, 43/38 Ohm pd/pu @ 1.8V 110 **37_OHM** — 37/28 Ohm @ 2.5V, 36/32 Ohm pd/pu @ 1.8V 111 **31_OHM** — 31/24 Ohm @ 2.5V, 30/27 Ohm pd/pu @ 1.8V
Reserved	This read-only field is reserved and always has the value 0.

37.4.427 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TX_CTL)

Address: 20E_0000h base + 6BCh offset = 20E_06BCh



IOMUXC_SW_PAD_CTL_PAD_RGMII_TX_CTL field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RGMII_TX_CTL field descriptions (continued)

Field	Description
	The value of this field is fixed and cannot be changed. 0 **CMOS** — CMOS input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: RGMII_TX_CTL. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: RGMII_TX_CTL. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: RGMII_TX_CTL. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: RGMII_TX_CTL. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Read Only Field The value of this field is fixed and cannot be changed. 000 **DISABLED** — Disabled
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: RGMII_TX_CTL. 000 **HIZ** — HI-Z 001 **287_OHM** — 287/174 Ohm @ 2.5V, 247/201 Ohm pd/pu @ 1.8V 010 **121_OHM** — 121/85 Ohm @ 2.5V, 113/96 Ohm pd/pu @ 1.8V 011 **76_OHM** — 76/56 Ohm @ 2.5V, 73/63 Ohm pd/pu @ 1.8V 100 **57_OHM** — 57/43 Ohm @ 2.5V, 55/48 Ohm pd/pu @ 1.8V 101 **45_OHM** — 45/34 Ohm @ 2.5V, 43/38 Ohm pd/pu @ 1.8V 110 **37_OHM** — 37/28 Ohm @ 2.5V, 36/32 Ohm pd/pu @ 1.8V 111 **31_OHM** — 31/24 Ohm @ 2.5V, 30/27 Ohm pd/pu @ 1.8V

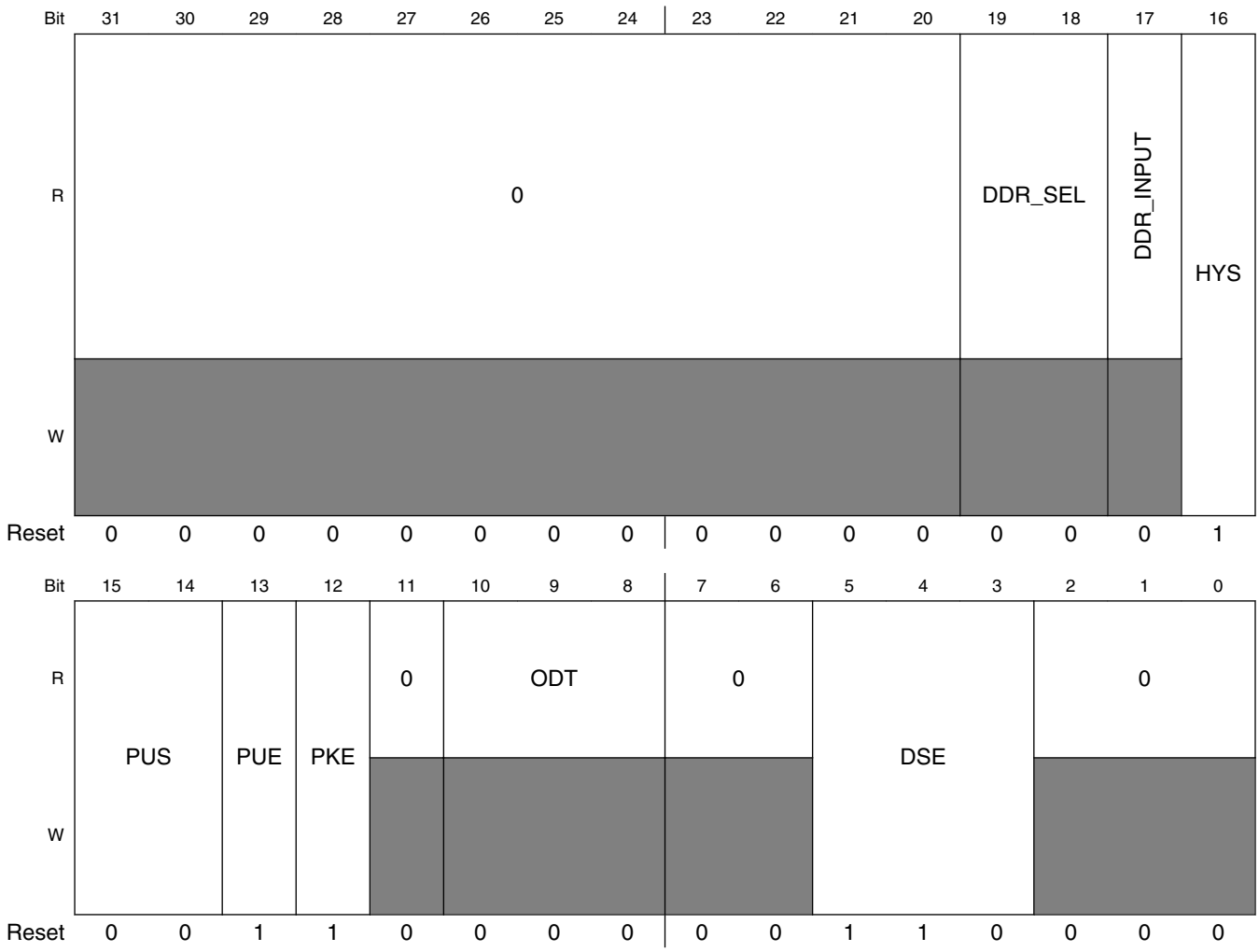
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RGMII_TX_CTL field descriptions (continued)

Field	Description
Reserved	This read-only field is reserved and always has the value 0.

37.4.428 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_RGMII_TXC)

Address: 20E_0000h base + 6C0h offset = 20E_06C0h



IOMUXC_SW_PAD_CTL_PAD_RGMII_TXC field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RGMII_TXC field descriptions (continued)

Field	Description
	This property can be configured using Group Control Register: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII Note: The value of this field does not reflect the value of the Group Control Register.
17 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field The value of this field is fixed and cannot be changed. 0 **CMOS** — CMOS input mode.
16 HYS	Hysteresis Enable Field Select one of next values for pad: RGMII_TXC. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: RGMII_TXC. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: RGMII_TXC. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: RGMII_TXC. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Read Only Field The value of this field is fixed and cannot be changed. 000 **DISABLED** — Disabled
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for pad: RGMII_TXC. 000 **HIZ** — HI-Z 001 **287_OHM** — 287/174 Ohm @ 2.5V, 247/201 Ohm pd/pu @ 1.8V

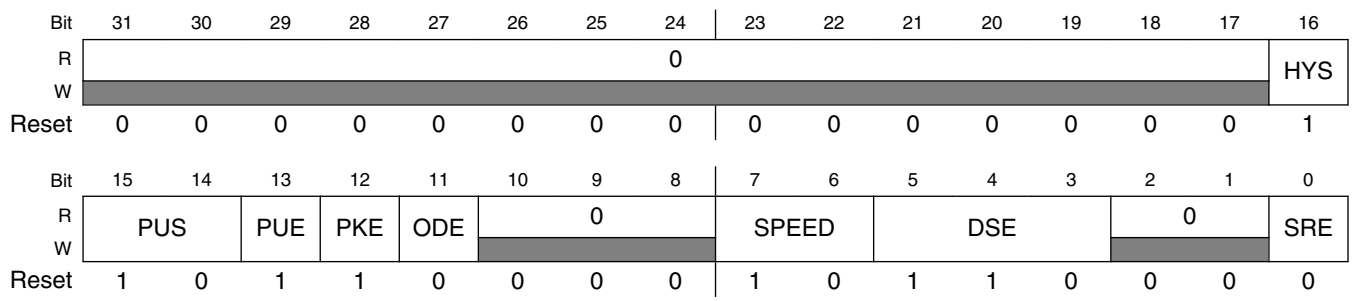
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RGMII_TXC field descriptions (continued)

Field	Description
	010 **121_OHM** — 121/85 Ohm @ 2.5V, 113/96 Ohm pd/pu @ 1.8V 011 **76_OHM** — 76/56 Ohm @ 2.5V, 73/63 Ohm pd/pu @ 1.8V 100 **57_OHM** — 57/43 Ohm @ 2.5V, 55/48 Ohm pd/pu @ 1.8V 101 **45_OHM** — 45/34 Ohm @ 2.5V, 43/38 Ohm pd/pu @ 1.8V 110 **37_OHM** — 37/28 Ohm @ 2.5V, 36/32 Ohm pd/pu @ 1.8V 111 **31_OHM** — 31/24 Ohm @ 2.5V, 30/27 Ohm pd/pu @ 1.8V
Reserved	This read-only field is reserved and always has the value 0.

37.4.429 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_CLK)

Address: 20E_0000h base + 6C4h offset = 20E_06C4h



IOMUXC_SW_PAD_CTL_PAD_SD1_CLK field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD1_CLK. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD1_CLK. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD1_CLK.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_CLK field descriptions (continued)

Field	Description
	0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD1_CLK. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD1_CLK. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.430 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_CMD)

Address: 20E_0000h base + 6C8h offset = 20E_06C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD1_CMD field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD1_CMD. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD1_CMD. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD1_CMD. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD1_CMD. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_CMD field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD1_CMD. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.431 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0)

Address: 20E_0000h base + 6CCh offset = 20E_06CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD1_DAT0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD1_DAT0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD1_DAT0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD1_DAT0. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD1_DAT0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.432 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1)

Address: 20E_0000h base + 6D0h offset = 20E_06D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD1_DAT1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD1_DAT1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD1_DAT1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD1_DAT1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD1_DAT1. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.433 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2)

Address: 20E_0000h base + 6D4h offset = 20E_06D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD1_DAT2. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD1_DAT2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD1_DAT2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD1_DAT2.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD1_DAT2. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.434 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3)

Address: 20E_0000h base + 6D8h offset = 20E_06D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD1_DAT3. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD1_DAT3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD1_DAT3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD1_DAT3. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD1_DAT3. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.435 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CLK)

Address: 20E_0000h base + 6DCh offset = 20E_06DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD2_CLK field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD2_CLK. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD2_CLK. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD2_CLK. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD2_CLK. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD2_CLK. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_CLK field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.436 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CMD)

Address: 20E_0000h base + 6E0h offset = 20E_06E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD2_CMD field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD2_CMD. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD2_CMD. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_CMD field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD2_CMD. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD2_CMD. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD2_CMD. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_CMD field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.437 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0)

Address: 20E_0000h base + 6E4h offset = 20E_06E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD2_DAT0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD2_DAT0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD2_DAT0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD2_DAT0.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD2_DAT0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.438 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1)

Address: 20E_0000h base + 6E8h offset = 20E_06E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD2_DAT1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD2_DAT1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD2_DAT1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD2_DAT1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD2_DAT1. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.439 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2)

Address: 20E_0000h base + 6ECh offset = 20E_06ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD2_DAT2. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD2_DAT2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD2_DAT2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD2_DAT2. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD2_DAT2. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.440 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3)

Address: 20E_0000h base + 6F0h offset = 20E_06F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD2_DAT3. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD2_DAT3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD2_DAT3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD2_DAT3. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD2_DAT3. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.441 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_CLK)

Address: 20E_0000h base + 6F4h offset = 20E_06F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD3_CLK field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD3_CLK. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD3_CLK. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD3_CLK. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD3_CLK.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD3_CLK field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD3_CLK. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.442 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_CMD)

Address: 20E_0000h base + 6F8h offset = 20E_06F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD3_CMD field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD3_CMD. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD3_CMD. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD3_CMD. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD3_CMD. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD3_CMD field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD3_CMD. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.443 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA0)

Address: 20E_0000h base + 6FCh offset = 20E_06FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA0 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD3_DAT0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD3_DAT0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD3_DAT0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD3_DAT0. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD3_DAT0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA0 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.444 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA1)

Address: 20E_0000h base + 700h offset = 20E_0700h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA1 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD3_DAT1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD3_DAT1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA1 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD3_DAT1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD3_DAT1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD3_DAT1. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA1 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.445 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA2)

Address: 20E_0000h base + 704h offset = 20E_0704h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA2 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD3_DAT2. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD3_DAT2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD3_DAT2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD3_DAT2.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA2 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD3_DAT2. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.446 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA3)

Address: 20E_0000h base + 708h offset = 20E_0708h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA3 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD3_DAT3. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD3_DAT3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD3_DAT3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD3_DAT3. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA3 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD3_DAT3. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.447 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA4)

Address: 20E_0000h base + 70Ch offset = 20E_070Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA4 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD3_DAT4. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD3_DAT4. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD3_DAT4. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD3_DAT4. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD3_DAT4. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA4 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.448 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA5)

Address: 20E_0000h base + 710h offset = 20E_0710h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA5 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD3_DAT5. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD3_DAT5. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA5 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD3_DAT5. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD3_DAT5. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD3_DAT5. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA5 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.449 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA6)

Address: 20E_0000h base + 714h offset = 20E_0714h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA6 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD3_DAT6. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD3_DAT6. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD3_DAT6. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD3_DAT6.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA6 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD3_DAT6. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.450 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_DATA7)

Address: 20E_0000h base + 718h offset = 20E_0718h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA7 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD3_DAT7. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD3_DAT7. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD3_DAT7. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD3_DAT7. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD3_DATA7 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD3_DAT7. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.451 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD3_RESET)

Address: 20E_0000h base + 71Ch offset = 20E_071Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD3_RESET field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD3_RST. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD3_RST. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD3_RST. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD3_RST. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD3_RST. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD3_RESET field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.452 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_CLK)

Address: 20E_0000h base + 720h offset = 20E_0720h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD4_CLK field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD4_CLK. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD4_CLK. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD4_CLK field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD4_CLK. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD4_CLK. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD4_CLK. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD4_CLK field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.453 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_CMD)

Address: 20E_0000h base + 724h offset = 20E_0724h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD4_CMD field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD4_CMD. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD4_CMD. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD4_CMD. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD4_CMD.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD4_CMD field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD4_CMD. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.454 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA0)

Address: 20E_0000h base + 728h offset = 20E_0728h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA0 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD4_DAT0. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD4_DAT0. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD4_DAT0. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD4_DAT0. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA0 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD4_DAT0. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.455 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA1)

Address: 20E_0000h base + 72Ch offset = 20E_072Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA1 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD4_DAT1. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD4_DAT1. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD4_DAT1. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD4_DAT1. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD4_DAT1. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA1 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.456 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA2)

Address: 20E_0000h base + 730h offset = 20E_0730h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA2 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD4_DAT2. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD4_DAT2. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA2 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD4_DAT2. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD4_DAT2. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD4_DAT2. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA2 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.457 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA3)

Address: 20E_0000h base + 734h offset = 20E_0734h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA3 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD4_DAT3. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD4_DAT3. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD4_DAT3. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD4_DAT3.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA3 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD4_DAT3. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.458 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA4)

Address: 20E_0000h base + 738h offset = 20E_0738h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA4 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD4_DAT4. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD4_DAT4. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD4_DAT4. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD4_DAT4. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA4 field descriptions (continued)

Field	Description
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD4_DAT4. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.459 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA5)

Address: 20E_0000h base + 73Ch offset = 20E_073Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA5 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD4_DAT5. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD4_DAT5. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD4_DAT5. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD4_DAT5. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD4_DAT5. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA5 field descriptions (continued)

Field	Description
	010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.460 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA6)

Address: 20E_0000h base + 740h offset = 20E_0740h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA6 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD4_DAT6. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD4_DAT6. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA6 field descriptions (continued)

Field	Description
	10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD4_DAT6. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD4_DAT6. 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD4_DAT6. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA6 field descriptions (continued)

Field	Description
	0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.461 Pad Control Register (IOMUXC_SW_PAD_CTL_PAD_SD4_DATA7)

Address: 20E_0000h base + 744h offset = 20E_0744h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUS		PUE	PKE	ODE	0			SPEED		DSE			0		SRE
W																
Reset	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA7 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for pad: SD4_DAT7. 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
15–14 PUS	Pull Up / Down Config. Field Select one of next values for pad: SD4_DAT7. 00 **100K_OHM_PD** — 100K Ohm Pull Down 01 **47K_OHM_PU** — 47K Ohm Pull Up 10 **100K_OHM_PU** — 100K Ohm Pull Up 11 **22K_OHM_PU** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one of next values for pad: SD4_DAT7. 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
12 PKE	Pull / Keep Enable Field Select one of next values for pad: SD4_DAT7.

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD4_DATA7 field descriptions (continued)

Field	Description
	0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Enables open drain of the pin. 0 **DISABLED** — Output is CMOS. 1 **ENABLED** — Output is Open Drain.
10–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 SPEED	Speed Field **NOTE:** The operational frequency on GPIO pads is dependent on slew rate (SRE) and supply voltage (OVDD). Please see GPIO Operating Frequency for more details. 00 **LOW** — 50 MHz (SRE=x) 01 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 10 **MEDIUM** — 100 MHz, if SRE=1: 100MHz @ 1.8V, 150MHz @ 3.3V 11 **MAXIMUM** — 100 MHz, if SRE=1: 150MHz @ 1.8V, 200MHz @ 3.3V
5–3 DSE	Drive Strength Field Select one of next values for pad: SD4_DAT7. 000 **HIZ** — HI-Z 001 **260_OHM** — 157 Ohm @ 3.3V, 260 Ohm @ 1.8V 010 **130_OHM** — 78 Ohm @ 3.3V, 130 Ohm @ 1.8V 011 **88_OHM** — 53 Ohm @ 3.3V, 88 Ohm @ 1.8V 100 **65_OHM** — 39 Ohm @ 3.3V, 65 Ohm @ 1.8V 101 **52_OHM** — 32 Ohm @ 3.3V, 52 Ohm @ 1.8V 110 **43_OHM** — 26 Ohm @ 3.3V, 43 Ohm @ 1.8V 111 **37_OHM** — 23 Ohm @ 3.3V, 37 Ohm @ 1.8V
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SRE	Slew Rate Field Slew rate control. 0 **SLOW** — Slow Slew Rate 1 **FAST** — Fast Slew Rate

37.4.462 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B7DS)

Address: 20E_0000h base + 748h offset = 20E_0748h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										DSE			0		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_GRP_B7DS field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for group: . Affected pads: DRAM_DATA56, DRAM_DATA57, DRAM_DATA58, DRAM_DATA59, DRAM_DATA60, DRAM_DATA61, DRAM_DATA62, DRAM_DATA63 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.463 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_ADDDS)

Address: 20E_0000h base + 74Ch offset = 20E_074Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										DSE			0		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_GRP_ADDDS field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.

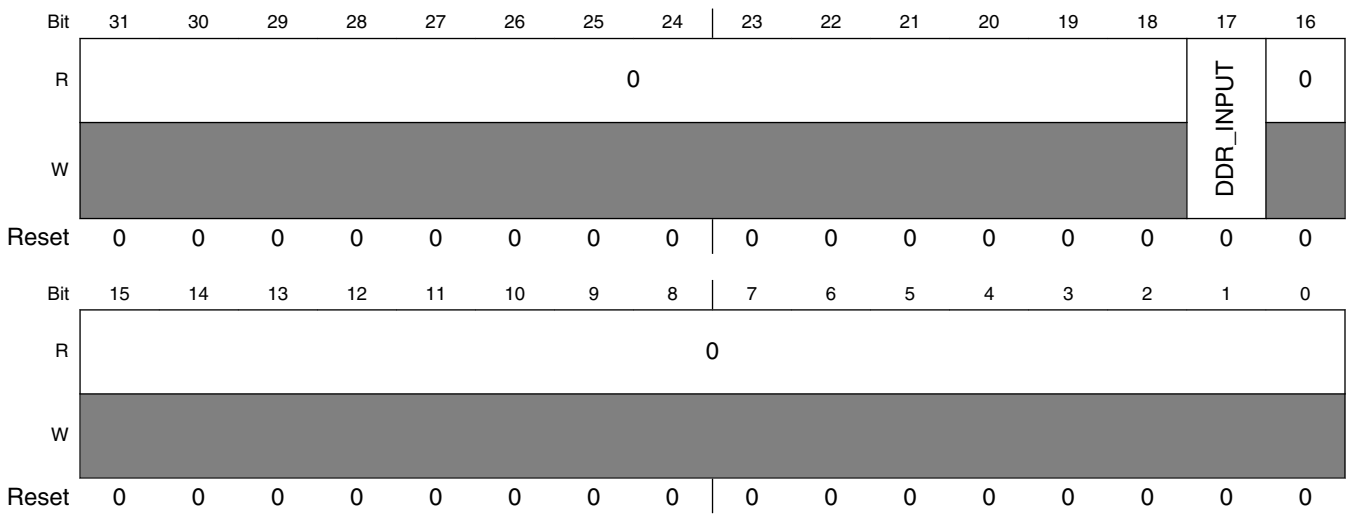
Table continues on the next page...

IOMUXC_SW_PAD_CTL_GRP_ADDDS field descriptions (continued)

Field	Description
5–3 DSE	Drive Strength Field Select one of next values for group: . Affected pads: DRAM_ADDR00, DRAM_ADDR01, DRAM_ADDR02, DRAM_ADDR03, DRAM_ADDR04, DRAM_ADDR05, DRAM_ADDR06, DRAM_ADDR07, DRAM_ADDR08, DRAM_ADDR09, DRAM_ADDR10, DRAM_ADDR11, DRAM_ADDR12, DRAM_ADDR13, DRAM_ADDR14, DRAM_ADDR15, DRAM_SDBA0, DRAM_SDBA1 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.464 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL)

Address: 20E_0000h base + 750h offset = 20E_0750h



IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value 0.
17 DDR_INPUT	DDR / CMOS Input Mode Field

Table continues on the next page...

IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL field descriptions (continued)

Field	Description
	Select one of next values for group: . Affected pads: DRAM_SDQS0_P, DRAM_SDQS1_P, DRAM_SDQS2_P, DRAM_SDQS3_P, DRAM_SDQS4_P, DRAM_SDQS5_P, DRAM_SDQS6_P, DRAM_SDQS7_P 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
Reserved	This read-only field is reserved and always has the value 0.

37.4.465 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDRPKE)

Address: 20E_0000h base + 754h offset = 20E_0754h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

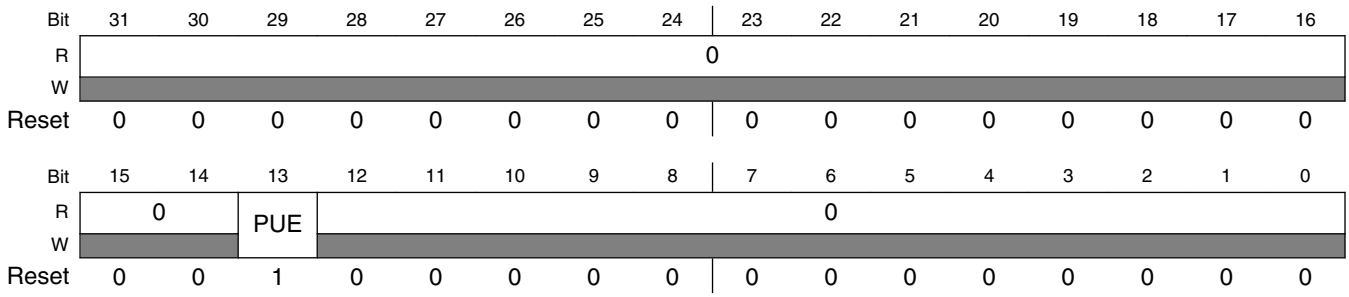
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			PKE	0											
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_GRP_DDRPKE field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value 0.
12 PKE	Pull / Keep Enable Field Select one of next values for group: . Affected pads: DRAM_DATA00, DRAM_DATA01, DRAM_DATA02, DRAM_DATA03, DRAM_DATA04, DRAM_DATA05, DRAM_DATA06, DRAM_DATA07, DRAM_DATA08, DRAM_DATA09, DRAM_DATA10, DRAM_DATA11, DRAM_DATA12, DRAM_DATA13, DRAM_DATA14, DRAM_DATA15, DRAM_DATA16, DRAM_DATA17, DRAM_DATA18, DRAM_DATA19, DRAM_DATA20, DRAM_DATA21, DRAM_DATA22, DRAM_DATA23, DRAM_DATA24, DRAM_DATA25, DRAM_DATA26, DRAM_DATA27, DRAM_DATA28, DRAM_DATA29, DRAM_DATA30, DRAM_DATA31, DRAM_DATA32, DRAM_DATA33, DRAM_DATA34, DRAM_DATA35, DRAM_DATA36, DRAM_DATA37, DRAM_DATA38, DRAM_DATA39, DRAM_DATA40, DRAM_DATA41, DRAM_DATA42, DRAM_DATA43, DRAM_DATA44, DRAM_DATA45, DRAM_DATA46, DRAM_DATA47, DRAM_DATA48, DRAM_DATA49, DRAM_DATA50, DRAM_DATA51, DRAM_DATA52, DRAM_DATA53, DRAM_DATA54, DRAM_DATA55, DRAM_DATA56, DRAM_DATA57, DRAM_DATA58, DRAM_DATA59, DRAM_DATA60, DRAM_DATA61, DRAM_DATA62, DRAM_DATA63 0 **DISABLED** — Pull/Keeper Disabled 1 **ENABLED** — Pull/Keeper Enabled
Reserved	This read-only field is reserved and always has the value 0.

37.4.466 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDRPK)

Address: 20E_0000h base + 758h offset = 20E_0758h



IOMUXC_SW_PAD_CTL_GRP_DDRPK field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
13 PUE	Pull / Keep Select Field Select one of next values for group: . Affected pads: DRAM_DATA00, DRAM_DATA01, DRAM_DATA02, DRAM_DATA03, DRAM_DATA04, DRAM_DATA05, DRAM_DATA06, DRAM_DATA07, DRAM_DATA08, DRAM_DATA09, DRAM_DATA10, DRAM_DATA11, DRAM_DATA12, DRAM_DATA13, DRAM_DATA14, DRAM_DATA15, DRAM_DATA16, DRAM_DATA17, DRAM_DATA18, DRAM_DATA19, DRAM_DATA20, DRAM_DATA21, DRAM_DATA22, DRAM_DATA23, DRAM_DATA24, DRAM_DATA25, DRAM_DATA26, DRAM_DATA27, DRAM_DATA28, DRAM_DATA29, DRAM_DATA30, DRAM_DATA31, DRAM_DATA32, DRAM_DATA33, DRAM_DATA34, DRAM_DATA35, DRAM_DATA36, DRAM_DATA37, DRAM_DATA38, DRAM_DATA39, DRAM_DATA40, DRAM_DATA41, DRAM_DATA42, DRAM_DATA43, DRAM_DATA44, DRAM_DATA45, DRAM_DATA46, DRAM_DATA47, DRAM_DATA48, DRAM_DATA49, DRAM_DATA50, DRAM_DATA51, DRAM_DATA52, DRAM_DATA53, DRAM_DATA54, DRAM_DATA55, DRAM_DATA56, DRAM_DATA57, DRAM_DATA58, DRAM_DATA59, DRAM_DATA60, DRAM_DATA61, DRAM_DATA62, DRAM_DATA63 0 **KEEP** — Keeper Enabled 1 **PULL** — Pull Enabled
Reserved	This read-only field is reserved and always has the value 0.

37.4.467 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDRHYS)

Address: 20E_0000h base + 75Ch offset = 20E_075Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															HYS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

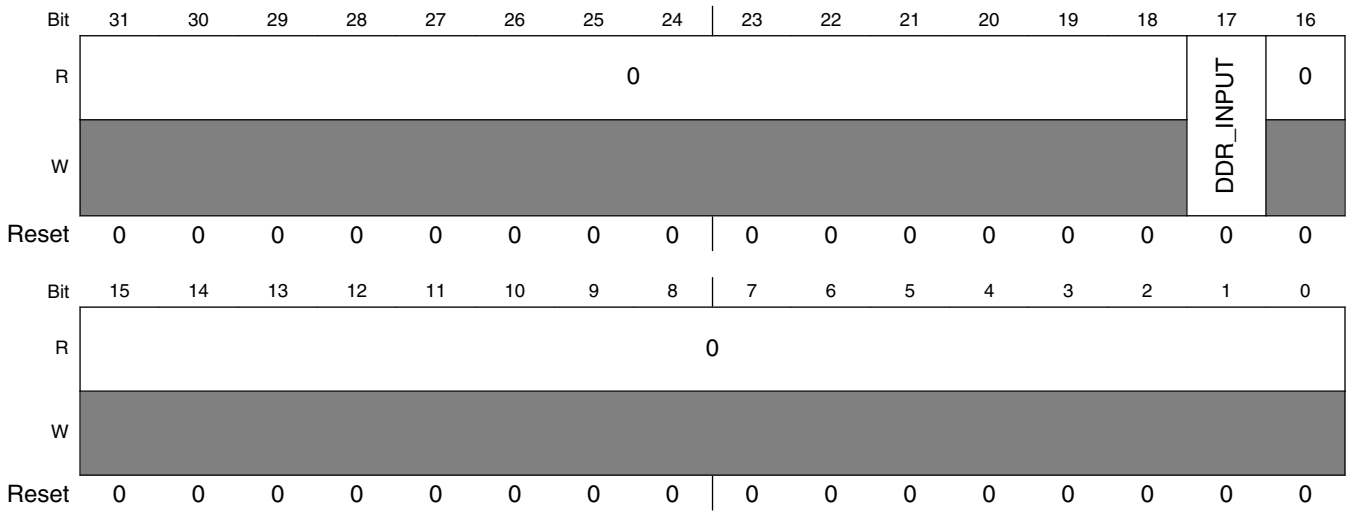
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_GRP_DDRHYS field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 HYS	Hysteresis Enable Field Select one of next values for group: . Affected pads: DRAM_DATA00, DRAM_DATA01, DRAM_DATA02, DRAM_DATA03, DRAM_DATA04, DRAM_DATA05, DRAM_DATA06, DRAM_DATA07, DRAM_DATA08, DRAM_DATA09, DRAM_DATA10, DRAM_DATA11, DRAM_DATA12, DRAM_DATA13, DRAM_DATA14, DRAM_DATA15, DRAM_DATA16, DRAM_DATA17, DRAM_DATA18, DRAM_DATA19, DRAM_DATA20, DRAM_DATA21, DRAM_DATA22, DRAM_DATA23, DRAM_DATA24, DRAM_DATA25, DRAM_DATA26, DRAM_DATA27, DRAM_DATA28, DRAM_DATA29, DRAM_DATA30, DRAM_DATA31, DRAM_DATA32, DRAM_DATA33, DRAM_DATA34, DRAM_DATA35, DRAM_DATA36, DRAM_DATA37, DRAM_DATA38, DRAM_DATA39, DRAM_DATA40, DRAM_DATA41, DRAM_DATA42, DRAM_DATA43, DRAM_DATA44, DRAM_DATA45, DRAM_DATA46, DRAM_DATA47, DRAM_DATA48, DRAM_DATA49, DRAM_DATA50, DRAM_DATA51, DRAM_DATA52, DRAM_DATA53, DRAM_DATA54, DRAM_DATA55, DRAM_DATA56, DRAM_DATA57, DRAM_DATA58, DRAM_DATA59, DRAM_DATA60, DRAM_DATA61, DRAM_DATA62, DRAM_DATA63, DRAM_SDQS0_P, DRAM_SDQS1_P, DRAM_SDQS2_P, DRAM_SDQS3_P, DRAM_SDQS4_P, DRAM_SDQS5_P, DRAM_SDQS6_P, DRAM_SDQS7_P 0 **DISABLED** — CMOS input 1 **ENABLED** — Schmitt trigger input
Reserved	This read-only field is reserved and always has the value 0.

37.4.468 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDRMODE)

Address: 20E_0000h base + 760h offset = 20E_0760h

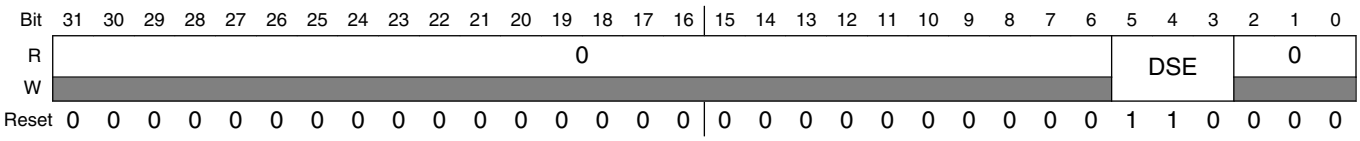


IOMUXC_SW_PAD_CTL_GRP_DDRMODE field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value 0.
17 DDR_INPUT	DDR / CMOS Input Mode Field Select one of next values for group: . Affected pads: DRAM_DATA00, DRAM_DATA01, DRAM_DATA02, DRAM_DATA03, DRAM_DATA04, DRAM_DATA05, DRAM_DATA06, DRAM_DATA07, DRAM_DATA08, DRAM_DATA09, DRAM_DATA10, DRAM_DATA11, DRAM_DATA12, DRAM_DATA13, DRAM_DATA14, DRAM_DATA15, DRAM_DATA16, DRAM_DATA17, DRAM_DATA18, DRAM_DATA19, DRAM_DATA20, DRAM_DATA21, DRAM_DATA22, DRAM_DATA23, DRAM_DATA24, DRAM_DATA25, DRAM_DATA26, DRAM_DATA27, DRAM_DATA28, DRAM_DATA29, DRAM_DATA30, DRAM_DATA31, DRAM_DATA32, DRAM_DATA33, DRAM_DATA34, DRAM_DATA35, DRAM_DATA36, DRAM_DATA37, DRAM_DATA38, DRAM_DATA39, DRAM_DATA40, DRAM_DATA41, DRAM_DATA42, DRAM_DATA43, DRAM_DATA44, DRAM_DATA45, DRAM_DATA46, DRAM_DATA47, DRAM_DATA48, DRAM_DATA49, DRAM_DATA50, DRAM_DATA51, DRAM_DATA52, DRAM_DATA53, DRAM_DATA54, DRAM_DATA55, DRAM_DATA56, DRAM_DATA57, DRAM_DATA58, DRAM_DATA59, DRAM_DATA60, DRAM_DATA61, DRAM_DATA62, DRAM_DATA63 0 **CMOS** — CMOS input mode. 1 **DIFFERENTIAL** — Differential input mode.
Reserved	This read-only field is reserved and always has the value 0.

37.4.469 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B0DS)

Address: 20E_0000h base + 764h offset = 20E_0764h

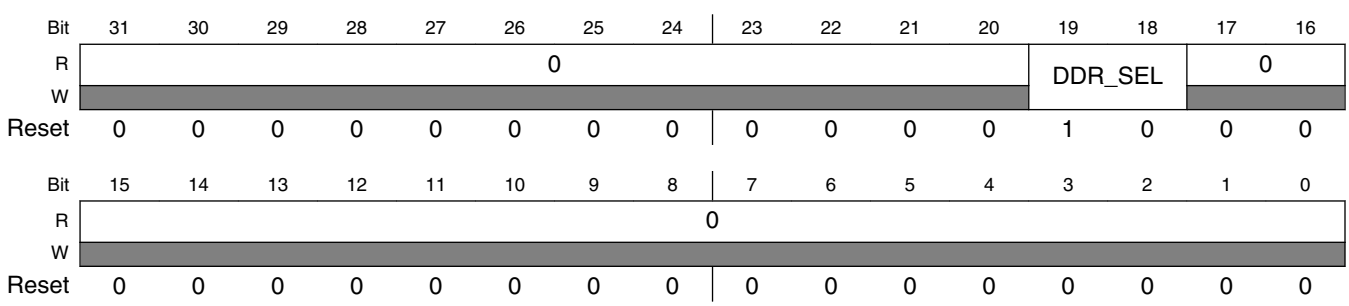


IOMUXC_SW_PAD_CTL_GRP_B0DS field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for group: . Affected pads: DRAM_DATA00, DRAM_DATA01, DRAM_DATA02, DRAM_DATA03, DRAM_DATA04, DRAM_DATA05, DRAM_DATA06, DRAM_DATA07 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.470 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII)

Address: 20E_0000h base + 768h offset = 20E_0768h

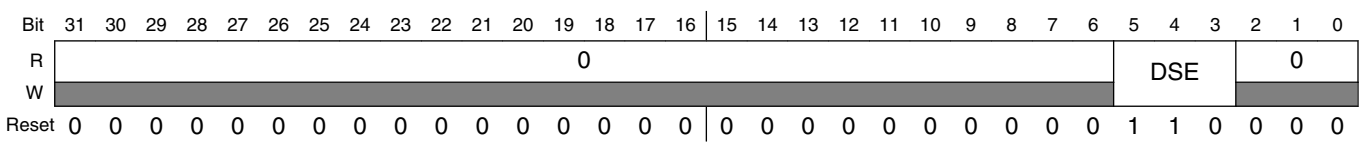


IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field Select one of next values for group: . Affected pads: RGMII_RD0, RGMII_RD1, RGMII_RD2, RGMII_RD3, RGMII_RXC, RGMII_RX_CTL, RGMII_TD0, RGMII_TD1, RGMII_TD2, RGMII_TD3, RGMII_TXC, RGMII_TX_CTL 00 **RESERVED0** — Reserved 01 **RESERVED1** — Reserved 10 **1P2V_IO** — 1.2V I/O interfaces including USB HSIC and MIPI_HSI. Provides calibrated drive strengths for signals ranging from 1.0V up to 1.3V. 11 **1P5V_IO** — 1.5V I/O interfaces including ENET. Provides calibrated drive strengths for signals ranging from 1.3V to 2.5V.
Reserved	This read-only field is reserved and always has the value 0.

37.4.471 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_CTLDS)

Address: 20E_0000h base + 76Ch offset = 20E_076Ch



IOMUXC_SW_PAD_CTL_GRP_CTLDS field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for group: . Affected pads: DRAM_CS0_B, DRAM_CS1_B, DRAM_SDBA2, DRAM_SDCKE0, DRAM_SDCKE1, DRAM_SDWE_B 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.472 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B1DS)

Address: 20E_0000h base + 770h offset = 20E_0770h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										DSE			0		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_GRP_B1DS field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for group: . Affected pads: DRAM_DATA08, DRAM_DATA09, DRAM_DATA10, DRAM_DATA11, DRAM_DATA12, DRAM_DATA13, DRAM_DATA14, DRAM_DATA15 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.473 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE)

Address: 20E_0000h base + 774h offset = 20E_0774h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												DDR_SEL		0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

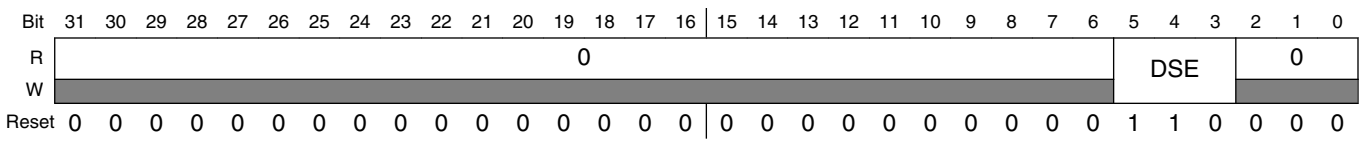
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 DDR_SEL	DDR Select Field Select one of next values for group: . Affected pads: DRAM_ADDR00, DRAM_ADDR01, DRAM_ADDR02, DRAM_ADDR03, DRAM_ADDR04, DRAM_ADDR05, DRAM_ADDR06, DRAM_ADDR07, DRAM_ADDR08, DRAM_ADDR09, DRAM_ADDR10, DRAM_ADDR11, DRAM_ADDR12, DRAM_ADDR13, DRAM_ADDR14, DRAM_ADDR15, DRAM_CAS_B, DRAM_CS0_B, DRAM_CS1_B, DRAM_DATA00, DRAM_DATA01, DRAM_DATA02, DRAM_DATA03, DRAM_DATA04, DRAM_DATA05, DRAM_DATA06, DRAM_DATA07, DRAM_DATA08, DRAM_DATA09, DRAM_DATA10, DRAM_DATA11, DRAM_DATA12, DRAM_DATA13, DRAM_DATA14, DRAM_DATA15, DRAM_DATA16, DRAM_DATA17, DRAM_DATA18, DRAM_DATA19, DRAM_DATA20, DRAM_DATA21, DRAM_DATA22, DRAM_DATA23, DRAM_DATA24, DRAM_DATA25, DRAM_DATA26, DRAM_DATA27, DRAM_DATA28, DRAM_DATA29, DRAM_DATA30, DRAM_DATA31, DRAM_DATA32, DRAM_DATA33, DRAM_DATA34, DRAM_DATA35, DRAM_DATA36, DRAM_DATA37, DRAM_DATA38, DRAM_DATA39, DRAM_DATA40, DRAM_DATA41, DRAM_DATA42, DRAM_DATA43, DRAM_DATA44, DRAM_DATA45, DRAM_DATA46, DRAM_DATA47, DRAM_DATA48, DRAM_DATA49, DRAM_DATA50, DRAM_DATA51, DRAM_DATA52, DRAM_DATA53, DRAM_DATA54, DRAM_DATA55, DRAM_DATA56, DRAM_DATA57, DRAM_DATA58, DRAM_DATA59, DRAM_DATA60, DRAM_DATA61, DRAM_DATA62, DRAM_DATA63, DRAM_DQM0, DRAM_DQM1, DRAM_DQM2, DRAM_DQM3, DRAM_DQM4, DRAM_DQM5, DRAM_DQM6, DRAM_DQM7, DRAM_ODT0, DRAM_ODT1, DRAM_RAS_B, DRAM_SDBA0, DRAM_SDBA1, DRAM_SDBA2, DRAM_SDCKE0, DRAM_SDCKE1, DRAM_SDCLK0_P, DRAM_SDCLK1_P, DRAM_SDQS0_P, DRAM_SDQS1_P, DRAM_SDQS2_P, DRAM_SDQS3_P, DRAM_SDQS4_P, DRAM_SDQS5_P, DRAM_SDQS6_P, DRAM_SDQS7_P, DRAM_SDWE_B 00 **RESERVED0** — Reserved 01 **RESERVED1** — Reserved 10 **LPDDR2** — LPDDR2 mode (240 Ohm driver unit calibration, 240, 120, 80, 60, 48, 40, 32 Ohm drive strengths at 1.2V) 11 **DDR3** — DDR3 mode (240 Ohm driver unit calibration, 240, 120, 80, 60, 48, 40, 32 Ohm drive strengths at 1.5V)
Reserved	This read-only field is reserved and always has the value 0.

37.4.474 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B2DS)

Address: 20E_0000h base + 778h offset = 20E_0778h

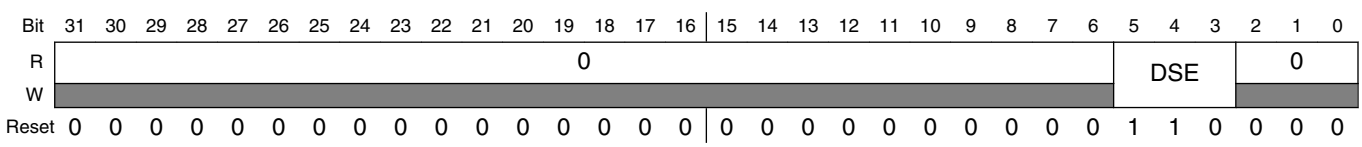


IOMUXC_SW_PAD_CTL_GRP_B2DS field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for group: . Affected pads: DRAM_DATA16, DRAM_DATA17, DRAM_DATA18, DRAM_DATA19, DRAM_DATA20, DRAM_DATA21, DRAM_DATA22, DRAM_DATA23 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.475 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B3DS)

Address: 20E_0000h base + 77Ch offset = 20E_077Ch



IOMUXC_SW_PAD_CTL_GRP_B3DS field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for group: . Affected pads: DRAM_DATA24, DRAM_DATA25, DRAM_DATA26, DRAM_DATA27, DRAM_DATA28, DRAM_DATA29, DRAM_DATA30, DRAM_DATA31 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm

Table continues on the next page...

IOMUXC_SW_PAD_CTL_GRP_B3DS field descriptions (continued)

Field	Description
	110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.476 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B4DS)

Address: 20E_0000h base + 780h offset = 20E_0780h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										DSE			0		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_GRP_B4DS field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for group: . Affected pads: DRAM_DATA32, DRAM_DATA33, DRAM_DATA34, DRAM_DATA35, DRAM_DATA36, DRAM_DATA37, DRAM_DATA38, DRAM_DATA39 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.477 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B5DS)

Address: 20E_0000h base + 784h offset = 20E_0784h

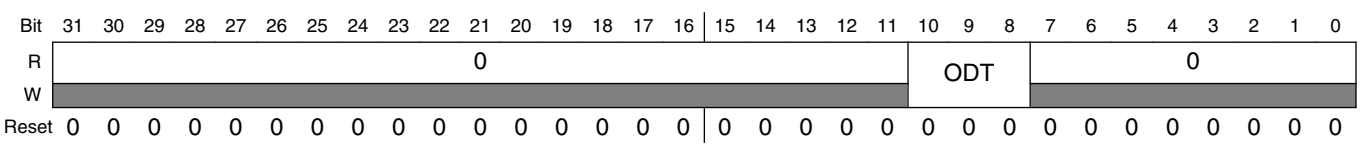
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										DSE			0		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_GRP_B5DS field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for group: . Affected pads: DRAM_DATA40, DRAM_DATA41, DRAM_DATA42, DRAM_DATA43, DRAM_DATA44, DRAM_DATA45, DRAM_DATA46, DRAM_DATA47 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.478 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_RGMII_TERM)

Address: 20E_0000h base + 788h offset = 20E_0788h



IOMUXC_SW_PAD_CTL_GRP_RGMII_TERM field descriptions

Field	Description
31–11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT	On Die Termination Field Select one of next values for group: . Affected pads: RGMII_RD0, RGMII_RD1, RGMII_RD2, RGMII_RD3, RGMII_RXC, RGMII_RX_CTL 000 **DISABLED** — Disabled 001 **120_OHM** — 120 Ohm ODT 010 **60_OHM** — 60 Ohm ODT 011 **40_OHM** — 40 Ohm ODT 100 **30_OHM** — 30 Ohm ODT 101 **24_OHM** — 24 Ohm ODT 110 **20_OHM** — 20 Ohm ODT 111 **17_OHM** — 17 Ohm ODT

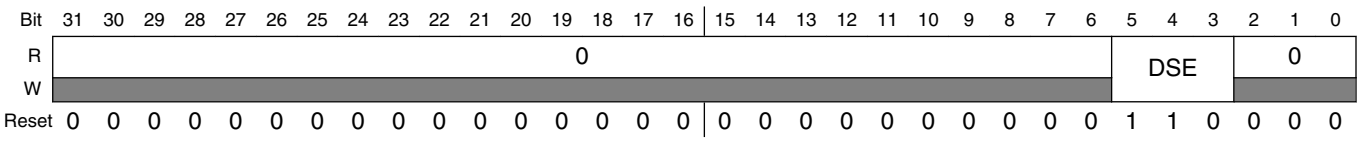
Table continues on the next page...

IOMUXC_SW_PAD_CTL_GRP_RGMII_TERM field descriptions (continued)

Field	Description
Reserved	This read-only field is reserved and always has the value 0.

37.4.479 Pad Group Control Register (IOMUXC_SW_PAD_CTL_GRP_B6DS)

Address: 20E_0000h base + 78Ch offset = 20E_078Ch

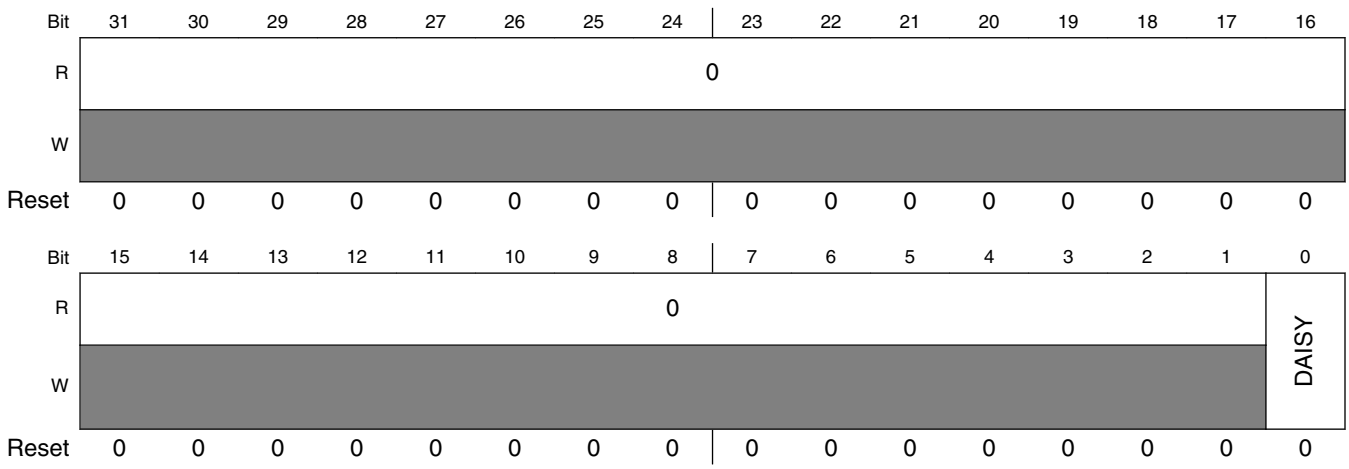


IOMUXC_SW_PAD_CTL_GRP_B6DS field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.
5–3 DSE	Drive Strength Field Select one of next values for group: . Affected pads: DRAM_DATA48, DRAM_DATA49, DRAM_DATA50, DRAM_DATA51, DRAM_DATA52, DRAM_DATA53, DRAM_DATA54, DRAM_DATA55 000 **HIZ** — HI-Z 001 **240_OHM** — 240 Ohm 010 **120_OHM** — 120 Ohm 011 **80_OHM** — 80 Ohm 100 **60_OHM** — 60 Ohm 101 **48_OHM** — 48 Ohm 110 **40_OHM** — 40 Ohm 111 **34_OHM** — 34 Ohm
Reserved	This read-only field is reserved and always has the value 0.

37.4.480 Select Input Register (IOMUXC_ANALOG_USB_OTG_ID_SELECT_INPUT)

Address: 20E_0000h base + 790h offset = 20E_0790h

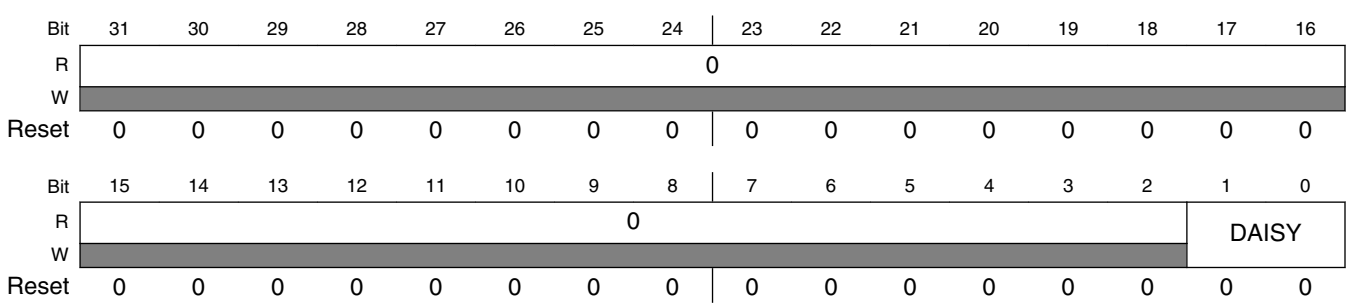


IOMUXC_ANALOG_USB_OTG_ID_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_RX_ER_ALT0** — Selecting ALT0 mode of pad ENET_RX_ER for USB_OTG_ID. 1 **GPIO01_ALT3** — Selecting ALT3 mode of pad GPIO_1 for USB_OTG_ID.

37.4.481 Select Input Register (IOMUXC_ASRC_ASRCK_CLOCK_6_SELECT_INPUT)

Address: 20E_0000h base + 794h offset = 20E_0794h

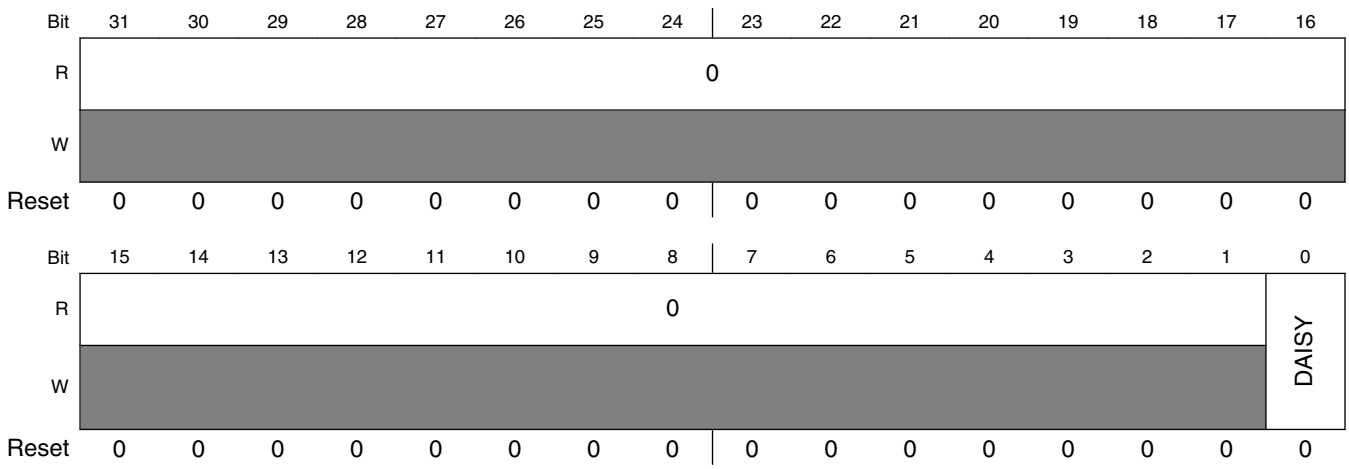


IOMUXC_ASRC_ASRCK_CLOCK_6_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **GPIO00_ALT3** — Selecting ALT3 mode of pad GPIO_0 for ASRC_EXT_CLK. 01 **GPIO18_ALT4** — Selecting ALT4 mode of pad GPIO_18 for ASRC_EXT_CLK. 10 **KEY_ROW3_ALT1** — Selecting ALT1 mode of pad KEY_ROW3 for ASRC_EXT_CLK.

37.4.482 Select Input Register (IOMUXC_AUD4_INPUT_DA_AMX_SELECT_INPUT)

Address: 20E_0000h base + 798h offset = 20E_0798h

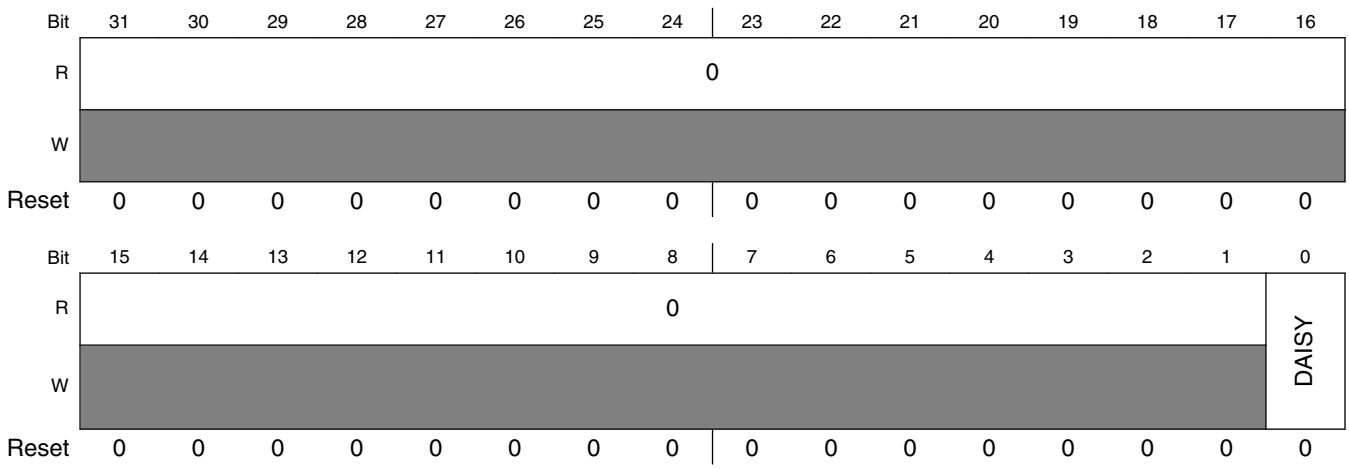


IOMUXC_AUD4_INPUT_DA_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DISP0_DATA23_ALT3** — Selecting ALT3 mode of pad DISP0_DAT23 for AUD4_RXD. 1 **SD2_DATA0_ALT3** — Selecting ALT3 mode of pad SD2_DAT0 for AUD4_RXD.

37.4.483 Select Input Register (IOMUXC_AUD4_INPUT_DB_AMX_SELECT_INPUT)

Address: 20E_0000h base + 79Ch offset = 20E_079Ch

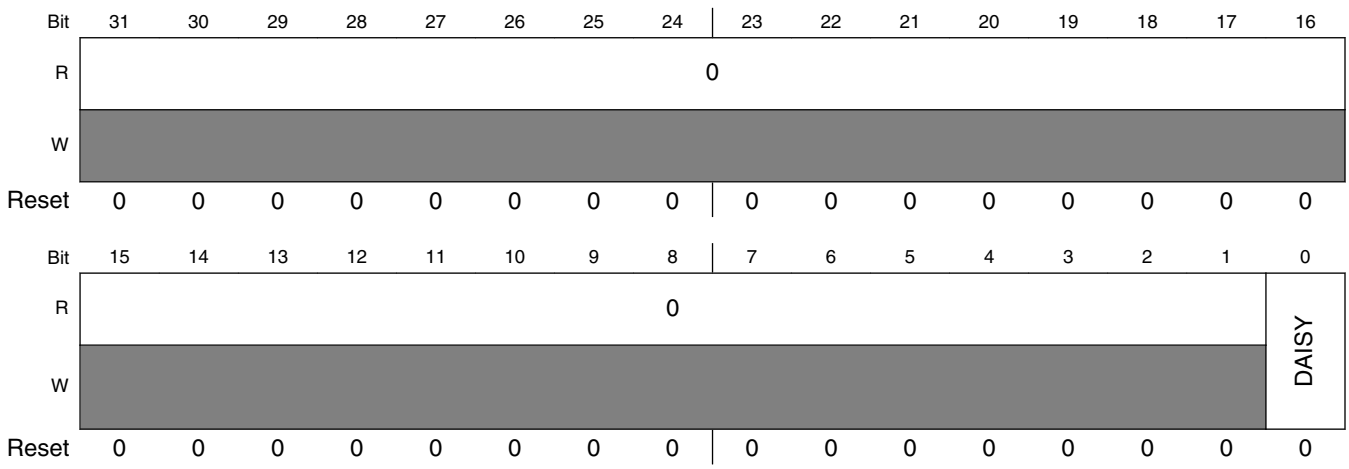


IOMUXC_AUD4_INPUT_DB_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DISP0_DATA21_ALT3** — Selecting ALT3 mode of pad DISP0_DAT21 for AUD4_TXD. 1 **SD2_DATA2_ALT3** — Selecting ALT3 mode of pad SD2_DAT2 for AUD4_TXD.

37.4.484 Select Input Register (IOMUXC_AUD4_INPUT_RXCLK_AMX_SELECT_INPUT)

Address: 20E_0000h base + 7A0h offset = 20E_07A0h

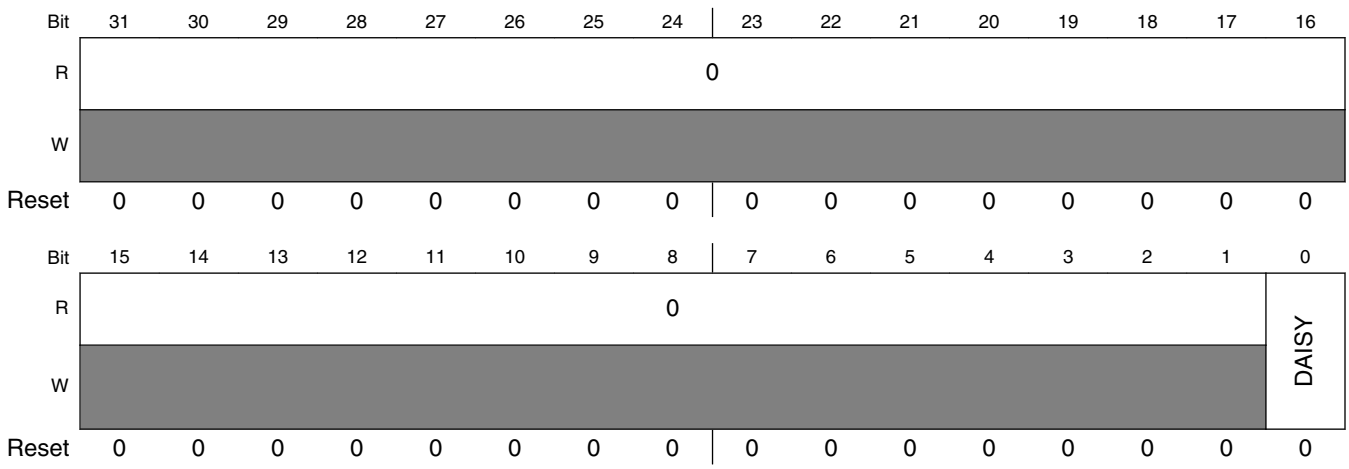


IOMUXC_AUD4_INPUT_RXCLK_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DISP0_DATA19_ALT4** — Selecting ALT4 mode of pad DISP0_DAT19 for AUD4_RXC. 1 **SD2_CMD_ALT3** — Selecting ALT3 mode of pad SD2_CMD for AUD4_RXC.

37.4.485 Select Input Register (IOMUXC_AUD4_INPUT_RXFS_AMX_SELECT_INPUT)

Address: 20E_0000h base + 7A4h offset = 20E_07A4h

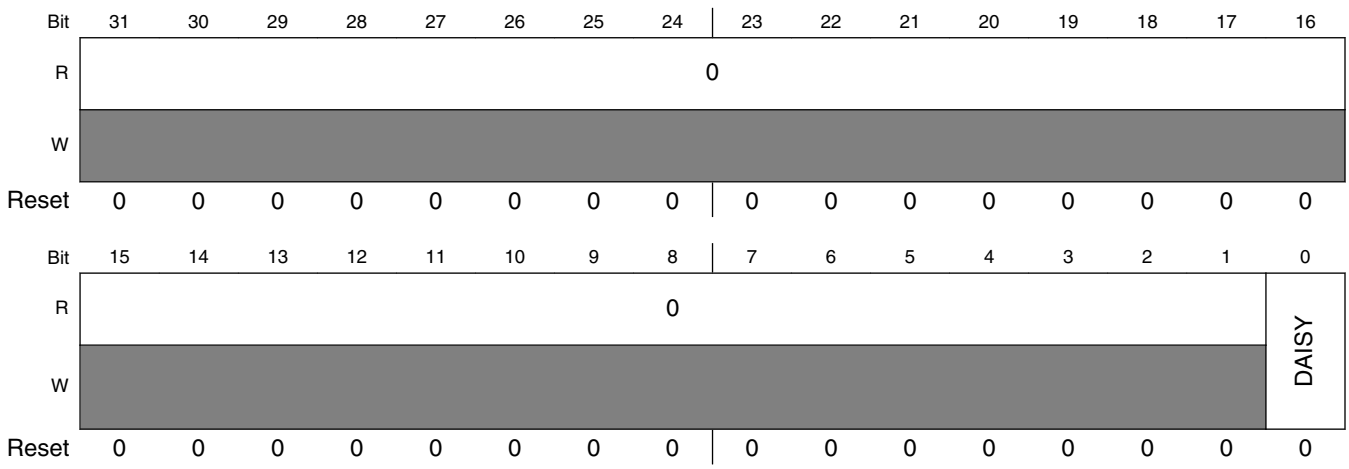


IOMUXC_AUD4_INPUT_RXFS_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DISP0_DATA18_ALT4** — Selecting ALT4 mode of pad DISP0_DAT18 for AUD4_RXFS. 1 **SD2_CLK_ALT3** — Selecting ALT3 mode of pad SD2_CLK for AUD4_RXFS.

37.4.486 Select Input Register (IOMUXC_AUD4_INPUT_TXCLK_AMX_SELECT_INPUT)

Address: 20E_0000h base + 7A8h offset = 20E_07A8h

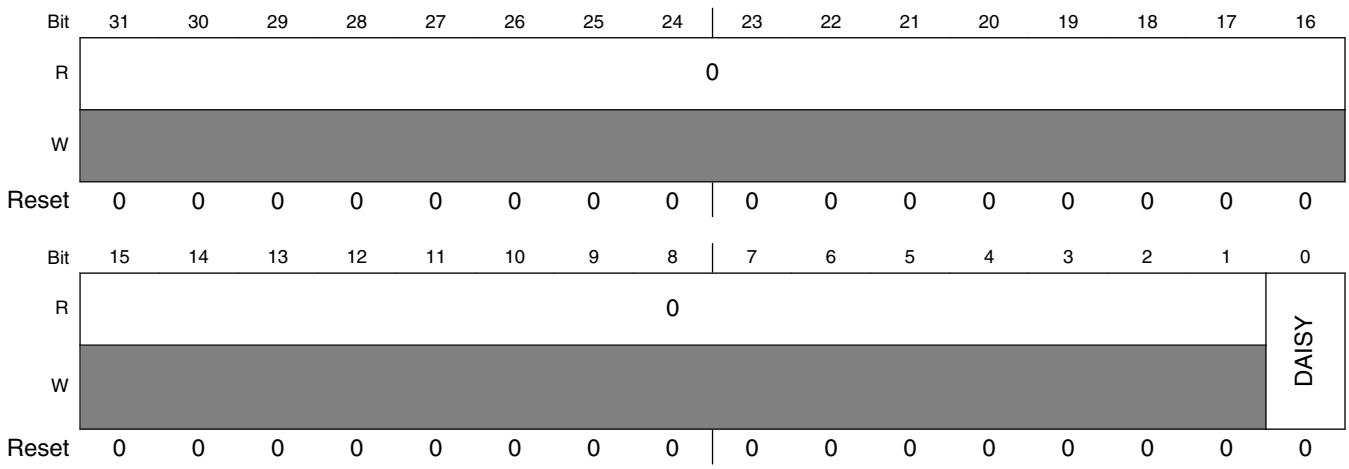


IOMUXC_AUD4_INPUT_TXCLK_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DISP0_DATA20_ALT3** — Selecting ALT3 mode of pad DISP0_DAT20 for AUD4_TXC. 1 **SD2_DATA3_ALT3** — Selecting ALT3 mode of pad SD2_DAT3 for AUD4_TXC.

37.4.487 Select Input Register (IOMUXC_AUD4_INPUT_TXFS_AMX_SELECT_INPUT)

Address: 20E_0000h base + 7ACh offset = 20E_07ACh

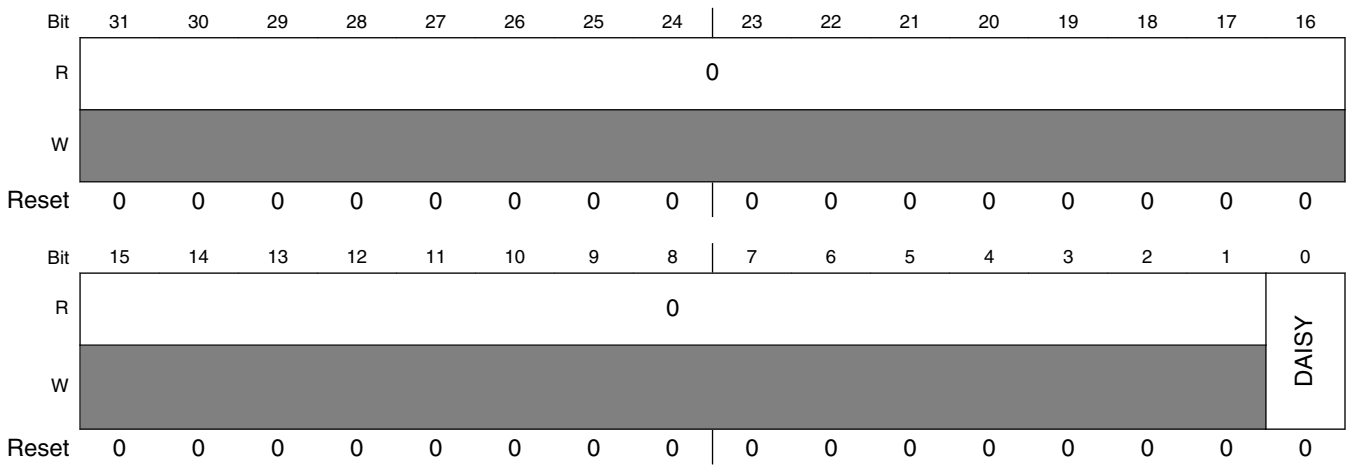


IOMUXC_AUD4_INPUT_TXFS_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DISP0_DATA22_ALT3** — Selecting ALT3 mode of pad DISP0_DAT22 for AUD4_TXFS. 1 **SD2_DATA1_ALT3** — Selecting ALT3 mode of pad SD2_DAT1 for AUD4_TXFS.

37.4.488 Select Input Register (IOMUXC_AUD5_INPUT_DA_AMX_SELECT_INPUT)

Address: 20E_0000h base + 7B0h offset = 20E_07B0h



IOMUXC_AUD5_INPUT_DA_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DISP0_DATA19_ALT3** — Selecting ALT3 mode of pad DISP0_DAT19 for AUD5_RXD. 1 **KEY_ROW1_ALT2** — Selecting ALT2 mode of pad KEY_ROW1 for AUD5_RXD.

37.4.489 Select Input Register (IOMUXC_AUD5_INPUT_DB_AMX_SELECT_INPUT)

Address: 20E_0000h base + 7B4h offset = 20E_07B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

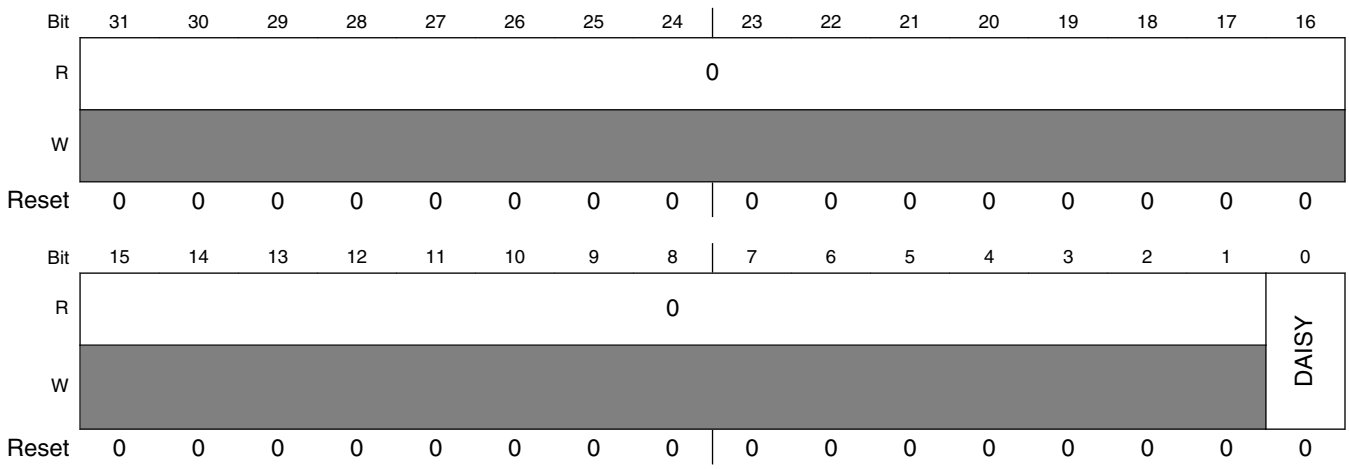
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_AUD5_INPUT_DB_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DISP0_DATA17_ALT3** — Selecting ALT3 mode of pad DISP0_DAT17 for AUD5_TXD. 1 **KEY_ROW0_ALT2** — Selecting ALT2 mode of pad KEY_ROW0 for AUD5_TXD.

37.4.490 Select Input Register (IOMUXC_AUD5_INPUT_RXCLK_AMX_SELECT_INPUT)

Address: 20E_0000h base + 7B8h offset = 20E_07B8h

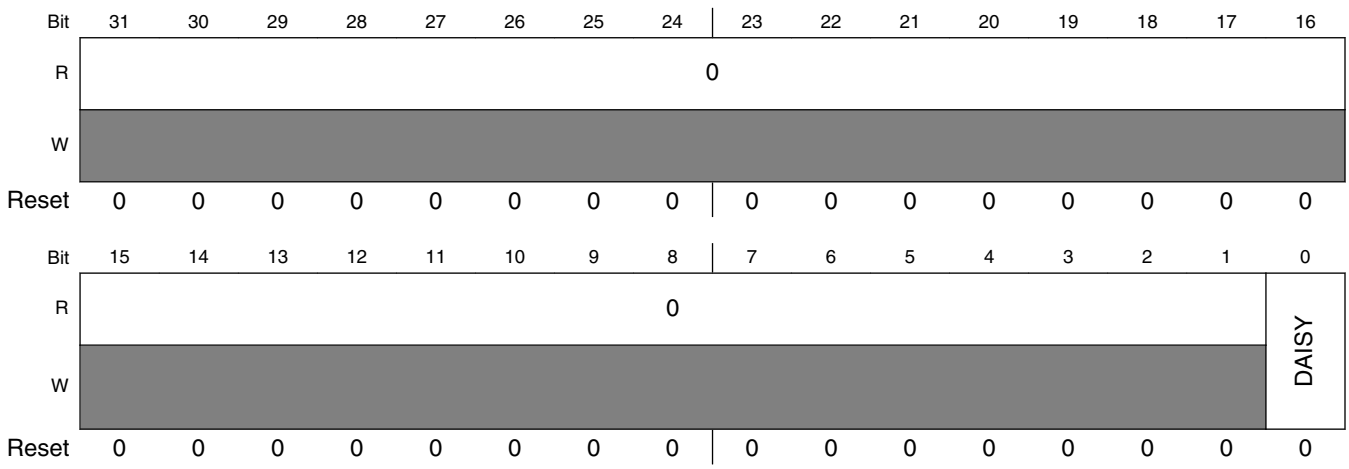


IOMUXC_AUD5_INPUT_RXCLK_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DISP0_DATA14_ALT3** — Selecting ALT3 mode of pad DISP0_DAT14 for AUD5_RXC. 1 **EIM_DATA25_ALT6** — Selecting ALT6 mode of pad EIM_D25 for AUD5_RXC.

37.4.491 Select Input Register (IOMUXC_AUD5_INPUT_RXFS_AMX_SELECT_INPUT)

Address: 20E_0000h base + 7BCh offset = 20E_07BCh

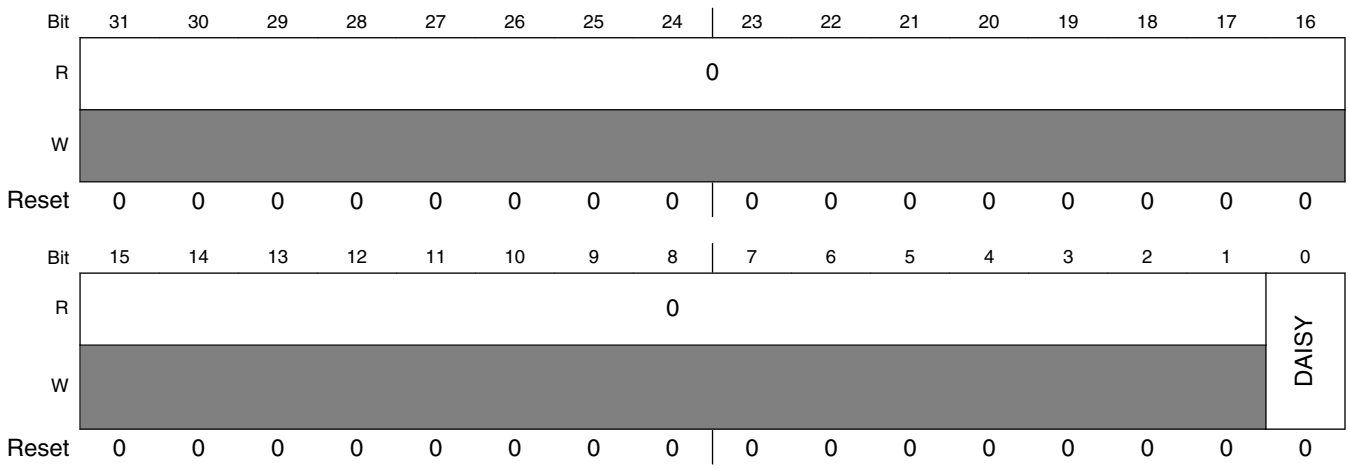


IOMUXC_AUD5_INPUT_RXFS_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DISP0_DATA13_ALT3** — Selecting ALT3 mode of pad DISP0_DAT13 for AUD5_RXFS. 1 **EIM_DATA24_ALT6** — Selecting ALT6 mode of pad EIM_D24 for AUD5_RXFS.

37.4.492 Select Input Register (IOMUXC_AUD5_INPUT_TXCLK_AMX_SELECT_INPUT)

Address: 20E_0000h base + 7C0h offset = 20E_07C0h

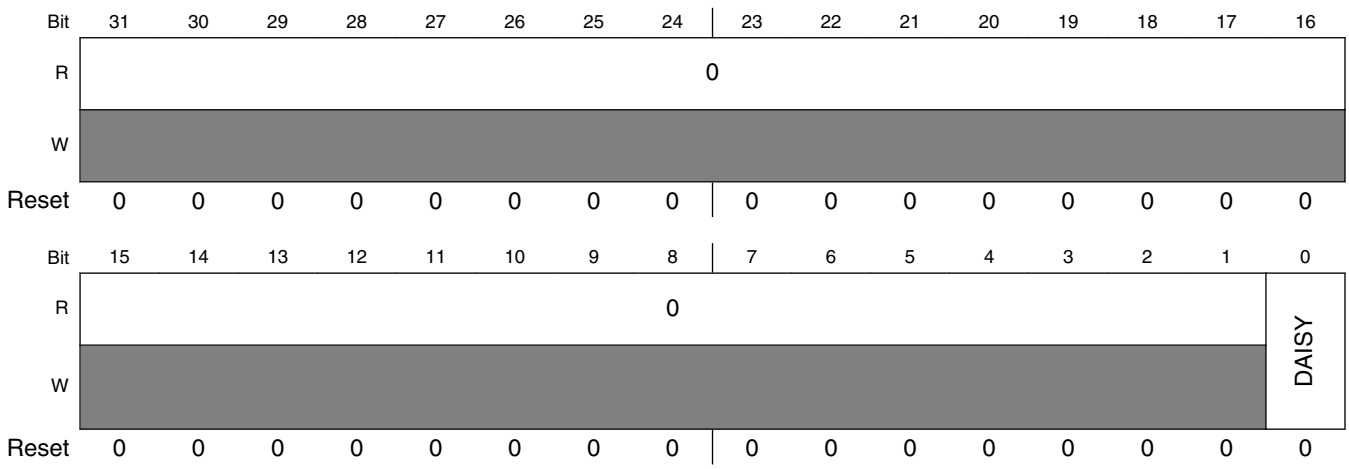


IOMUXC_AUD5_INPUT_TXCLK_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DISP0_DATA16_ALT3** — Selecting ALT3 mode of pad DISP0_DAT16 for AUD5_TXC. 1 **KEY_COL0_ALT2** — Selecting ALT2 mode of pad KEY_COL0 for AUD5_TXC.

37.4.493 Select Input Register (IOMUXC_AUD5_INPUT_TXFS_AMX_SELECT_INPUT)

Address: 20E_0000h base + 7C4h offset = 20E_07C4h

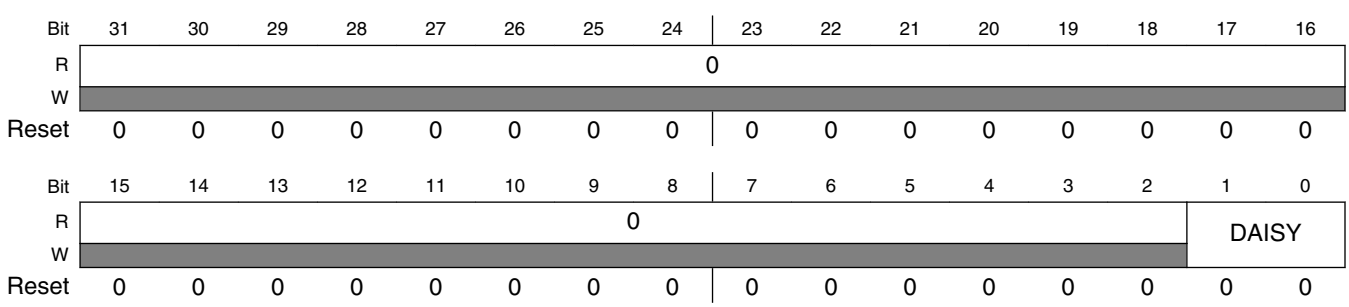


IOMUXC_AUD5_INPUT_TXFS_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DISP0_DATA18_ALT3** — Selecting ALT3 mode of pad DISP0_DAT18 for AUD5_TXFS. 1 **KEY_COL1_ALT2** — Selecting ALT2 mode of pad KEY_COL1 for AUD5_TXFS.

37.4.494 Select Input Register (IOMUXC_FLEXCAN1_RX_SELECT_INPUT)

Address: 20E_0000h base + 7C8h offset = 20E_07C8h

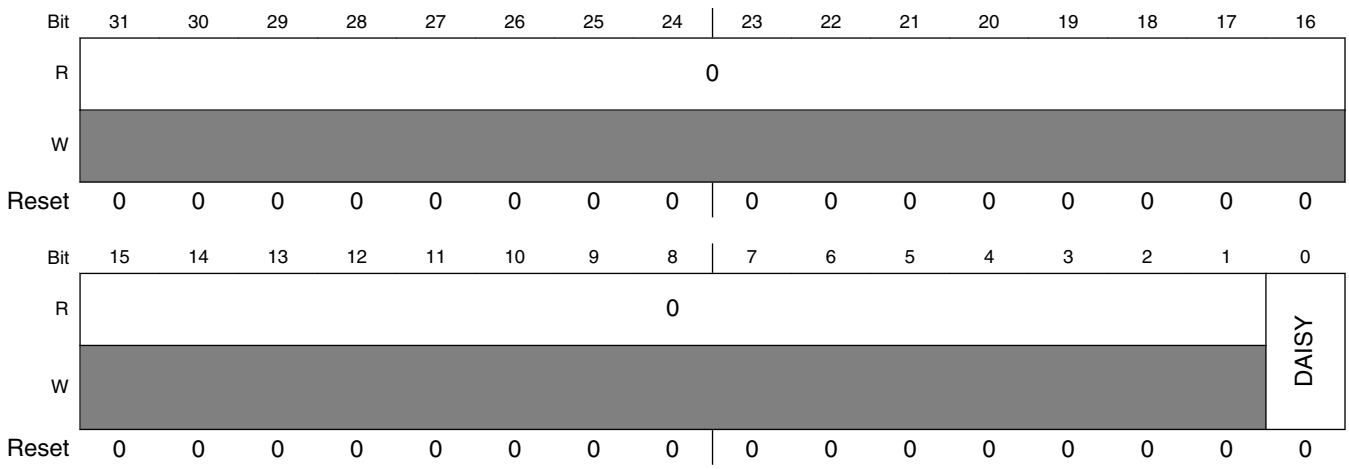


IOMUXC_FLEXCAN1_RX_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **GPIO08_ALT3** — Selecting ALT3 mode of pad GPIO_8 for FLEXCAN1_RX. 01 **KEY_ROW2_ALT2** — Selecting ALT2 mode of pad KEY_ROW2 for FLEXCAN1_RX. 10 **SD3_CLK_ALT2** — Selecting ALT2 mode of pad SD3_CLK for FLEXCAN1_RX.

37.4.495 Select Input Register (IOMUXC_FLEXCAN2_RX_SELECT_INPUT)

Address: 20E_0000h base + 7CCh offset = 20E_07CCh



IOMUXC_FLEXCAN2_RX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **KEY_ROW4_ALT0** — Selecting ALT0 mode of pad KEY_ROW4 for FLEXCAN2_RX. 1 **SD3_DATA1_ALT2** — Selecting ALT2 mode of pad SD3_DAT1 for FLEXCAN2_RX.

37.4.496 Select Input Register (IOMUXC_CCM_PMIC_READY_SELECT_INPUT)

Address: 20E_0000h base + 7D4h offset = 20E_07D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_CCM_PMIC_READY_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_EB0_B_ALT4** — Selecting ALT4 mode of pad EIM_EB0 for CCM_PMIC_READY. 1 **GPIO17_ALT2** — Selecting ALT2 mode of pad GPIO_17 for CCM_PMIC_READY.

37.4.497 Select Input Register (IOMUXC_ECSPI1_CSPI_CLK_IN_SELECT_INPUT)

Address: 20E_0000h base + 7D8h offset = 20E_07D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DAISY	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ECSPI1_CSPI_CLK_IN_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **CSI0_DATA04_ALT2** — Selecting ALT2 mode of pad CSI0_DAT4 for ECSPI1_SCLK. 01 **DISP0_DATA20_ALT2** — Selecting ALT2 mode of pad DISP0_DAT20 for ECSPI1_SCLK. 10 **EIM_DATA16_ALT1** — Selecting ALT1 mode of pad EIM_D16 for ECSPI1_SCLK. 11 **KEY_COL0_ALT0** — Selecting ALT0 mode of pad KEY_COL0 for ECSPI1_SCLK.

37.4.498 Select Input Register (IOMUXC_ECSPI1_MISO_SELECT_INPUT)

Address: 20E_0000h base + 7DCh offset = 20E_07DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DAISY	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ECSPI1_MISO_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **CSI0_DATA06_ALT2** — Selecting ALT2 mode of pad CSI0_DAT6 for ECSPI1_MISO. 01 **DISP0_DATA22_ALT2** — Selecting ALT2 mode of pad DISP0_DAT22 for ECSPI1_MISO. 10 **EIM_DATA17_ALT1** — Selecting ALT1 mode of pad EIM_D17 for ECSPI1_MISO. 11 **KEY_COL1_ALT0** — Selecting ALT0 mode of pad KEY_COL1 for ECSPI1_MISO.

37.4.499 Select Input Register (IOMUXC_ECSPI1_MOSI_SELECT_INPUT)

Address: 20E_0000h base + 7E0h offset = 20E_07E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DAISY	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ECSPI1_MOSI_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **CSI0_DATA05_ALT2** — Selecting ALT2 mode of pad CSI0_DAT5 for ECSPI1_MOSI. 01 **DISP0_DATA21_ALT2** — Selecting ALT2 mode of pad DISP0_DAT21 for ECSPI1_MOSI. 10 **EIM_DATA18_ALT1** — Selecting ALT1 mode of pad EIM_D18 for ECSPI1_MOSI. 11 **KEY_ROW0_ALT0** — Selecting ALT0 mode of pad KEY_ROW0 for ECSPI1_MOSI.

37.4.500 Select Input Register (IOMUXC_ECSPI1_SS0_SELECT_INPUT)

Address: 20E_0000h base + 7E4h offset = 20E_07E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DAISY	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ECSPI1_SS0_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **CSI0_DATA07_ALT2** — Selecting ALT2 mode of pad CSI0_DAT7 for ECSPI1_SS0. 01 **DISP0_DATA23_ALT2** — Selecting ALT2 mode of pad DISP0_DAT23 for ECSPI1_SS0. 10 **EIM_EB2_B_ALT1** — Selecting ALT1 mode of pad EIM_EB2 for ECSPI1_SS0. 11 **KEY_ROW1_ALT0** — Selecting ALT0 mode of pad KEY_ROW1 for ECSPI1_SS0.

37.4.501 Select Input Register (IOMUXC_ECSPI1_SS1_SELECT_INPUT)

Address: 20E_0000h base + 7E8h offset = 20E_07E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

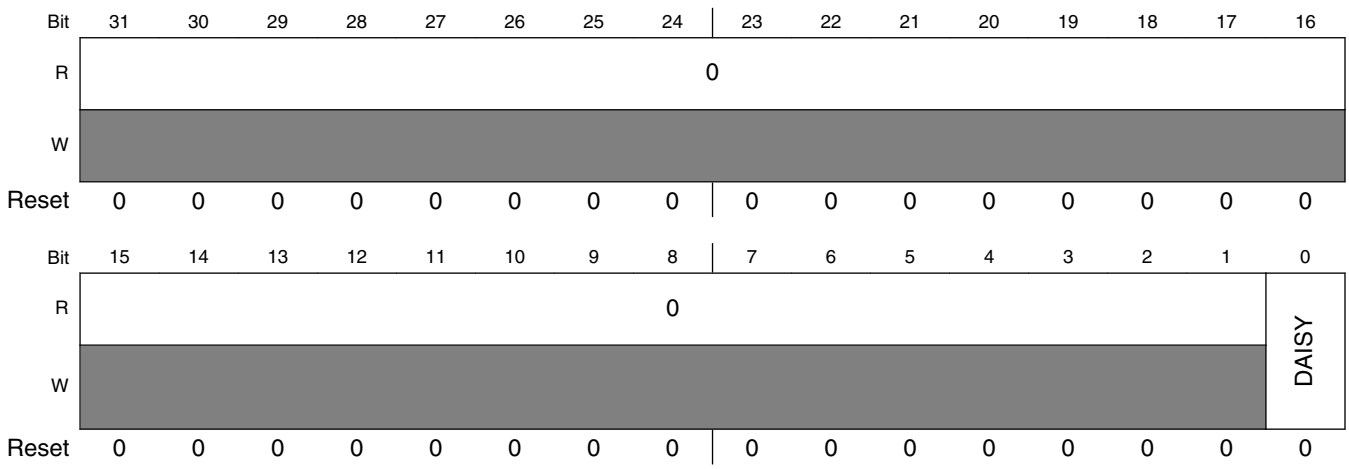
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DAISY	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ECSPI1_SS1_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **DISP0_DATA15_ALT2** — Selecting ALT2 mode of pad DISP0_DAT15 for ECSPI1_SS1. 01 **EIM_DATA19_ALT1** — Selecting ALT1 mode of pad EIM_D19 for ECSPI1_SS1. 10 **KEY_COL2_ALT0** — Selecting ALT0 mode of pad KEY_COL2 for ECSPI1_SS1.

37.4.502 Select Input Register (IOMUXC_ECSPI1_SS2_SELECT_INPUT)

Address: 20E_0000h base + 7ECh offset = 20E_07ECh

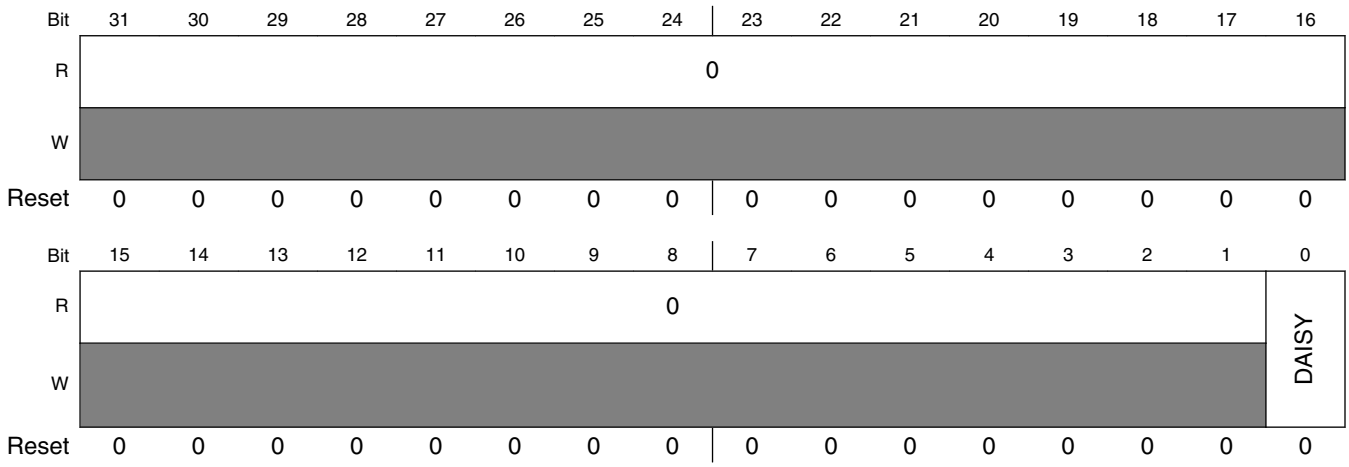


IOMUXC_ECSPI1_SS2_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_DATA24_ALT3** — Selecting ALT3 mode of pad EIM_D24 for ECSPI1_SS2. 1 **KEY_ROW2_ALT0** — Selecting ALT0 mode of pad KEY_ROW2 for ECSPI1_SS2.

37.4.503 Select Input Register (IOMUXC_ECSPI1_SS3_SELECT_INPUT)

Address: 20E_0000h base + 7F0h offset = 20E_07F0h

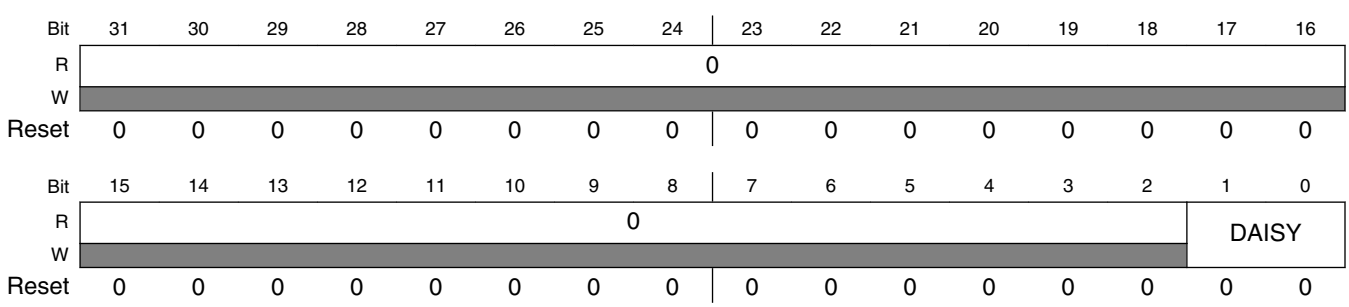


IOMUXC_ECSPI1_SS3_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_DATA25_ALT3** — Selecting ALT3 mode of pad EIM_D25 for ECSPI1_SS3. 1 **KEY_COL3_ALT0** — Selecting ALT0 mode of pad KEY_COL3 for ECSPI1_SS3.

37.4.504 Select Input Register (IOMUXC_ECSPI2_CSPI_CLK_IN_SELECT_INPUT)

Address: 20E_0000h base + 7F4h offset = 20E_07F4h



IOMUXC_ECSPI2_CSPI_CLK_IN_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **CSI0_DATA08_ALT2** — Selecting ALT2 mode of pad CSI0_DAT8 for ECSPI2_SCLK. 01 **DISP0_DATA19_ALT2** — Selecting ALT2 mode of pad DISP0_DAT19 for ECSPI2_SCLK. 10 **EIM_CS0_B_ALT2** — Selecting ALT2 mode of pad EIM_CS0 for ECSPI2_SCLK.

37.4.505 Select Input Register (IOMUXC_ECSPI2_MISO_SELECT_INPUT)

Address: 20E_0000h base + 7F8h offset = 20E_07F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

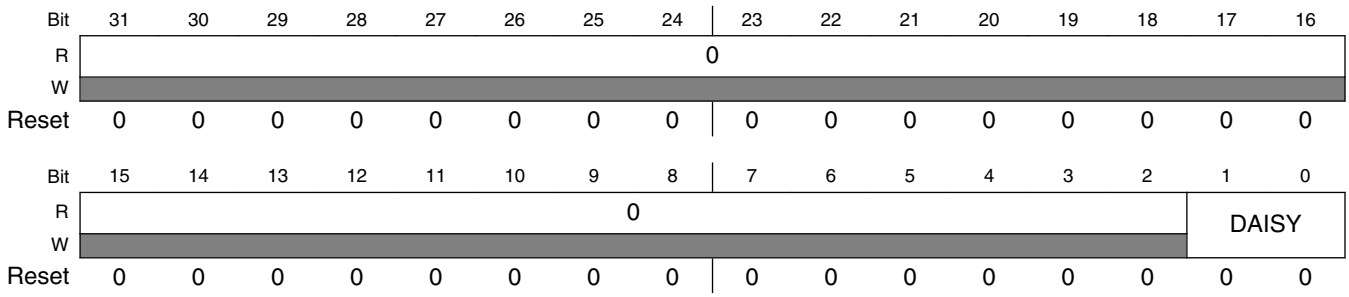
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DAISY	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ECSPI2_MISO_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **CSI0_DATA10_ALT2** — Selecting ALT2 mode of pad CSI0_DAT10 for ECSPI2_MISO. 01 **DISP0_DATA17_ALT2** — Selecting ALT2 mode of pad DISP0_DAT17 for ECSPI2_MISO. 10 **EIM_OE_B_ALT2** — Selecting ALT2 mode of pad EIM_OE for ECSPI2_MISO.

37.4.506 Select Input Register (IOMUXC_ECSPI2_MOSI_SELECT_INPUT)

Address: 20E_0000h base + 7FCh offset = 20E_07FCh

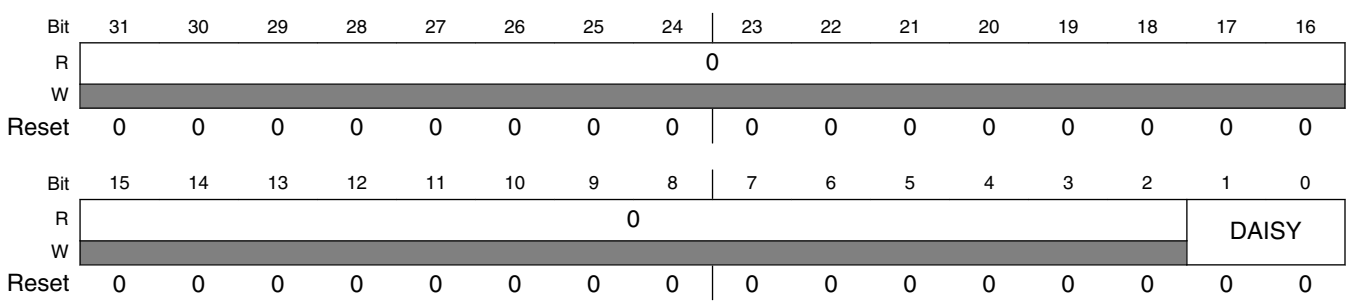


IOMUXC_ECSPI2_MOSI_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **CSI0_DATA09_ALT2** — Selecting ALT2 mode of pad CSI0_DAT9 for ECSPI2_MOSI. 01 **DISP0_DATA16_ALT2** — Selecting ALT2 mode of pad DISP0_DAT16 for ECSPI2_MOSI. 10 **EIM_CS1_B_ALT2** — Selecting ALT2 mode of pad EIM_CS1 for ECSPI2_MOSI.

37.4.507 Select Input Register (IOMUXC_ECSPI2_SS0_SELECT_INPUT)

Address: 20E_0000h base + 800h offset = 20E_0800h



IOMUXC_ECSPI2_SS0_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.

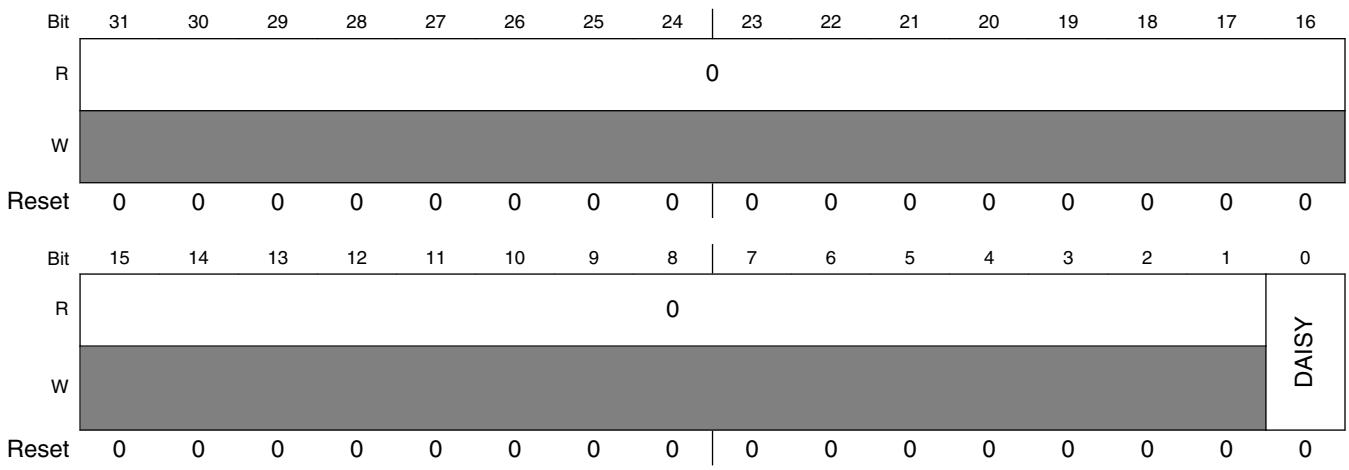
Table continues on the next page...

IOMUXC_ECSPI2_SS0_SELECT_INPUT field descriptions (continued)

Field	Description
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **CSI0_DATA11_ALT2** — Selecting ALT2 mode of pad CSI0_DAT11 for ECSPI2_SS0. 01 **DISP0_DATA18_ALT2** — Selecting ALT2 mode of pad DISP0_DAT18 for ECSPI2_SS0. 10 **EIM_RW_ALT2** — Selecting ALT2 mode of pad EIM_RW for ECSPI2_SS0.

37.4.508 Select Input Register (IOMUXC_ECSPI2_SS1_SELECT_INPUT)

Address: 20E_0000h base + 804h offset = 20E_0804h

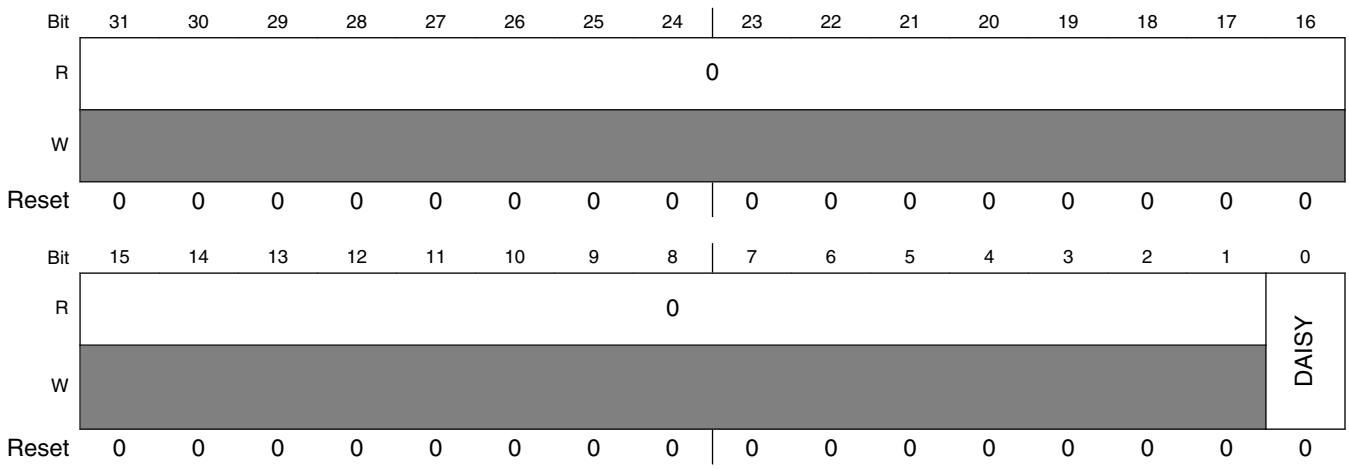


IOMUXC_ECSPI2_SS1_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DISP0_DATA15_ALT3** — Selecting ALT3 mode of pad DISP0_DAT15 for ECSPI2_SS1. 1 **EIM_LBA_B_ALT2** — Selecting ALT2 mode of pad EIM_LBA for ECSPI2_SS1.

37.4.509 Select Input Register (IOMUXC_ECSPI4_SS0_SELECT_INPUT)

Address: 20E_0000h base + 808h offset = 20E_0808h

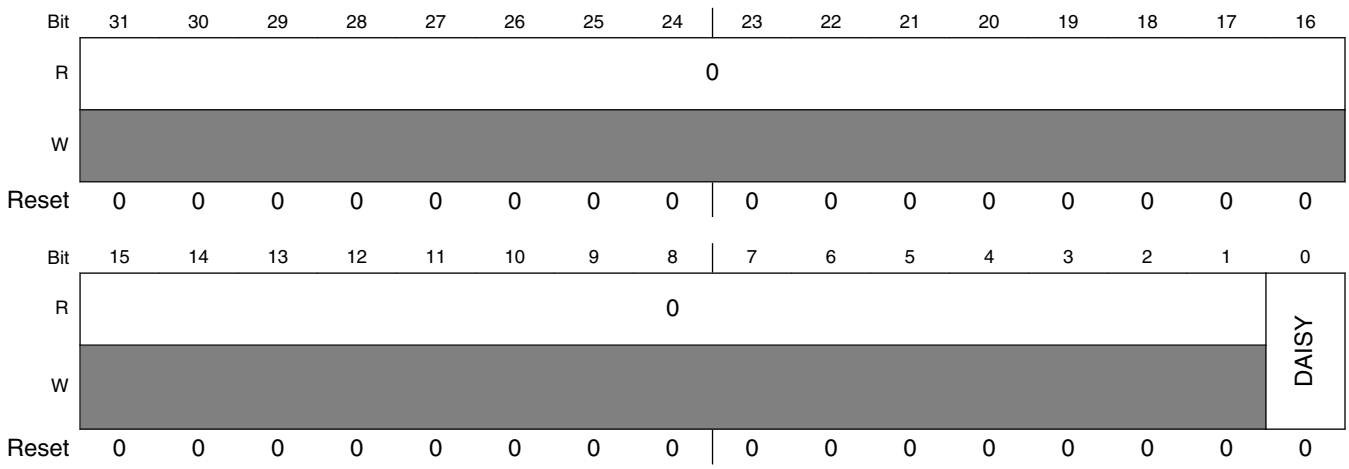


IOMUXC_ECSPI4_SS0_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_DATA20_ALT1** — Selecting ALT1 mode of pad EIM_D20 for ECSPI4_SS0. 1 **EIM_DATA29_ALT2** — Selecting ALT2 mode of pad EIM_D29 for ECSPI4_SS0.

37.4.510 Select Input Register (IOMUXC_ENET_REF_CLK_SELECT_INPUT)

Address: 20E_0000h base + 80Ch offset = 20E_080Ch

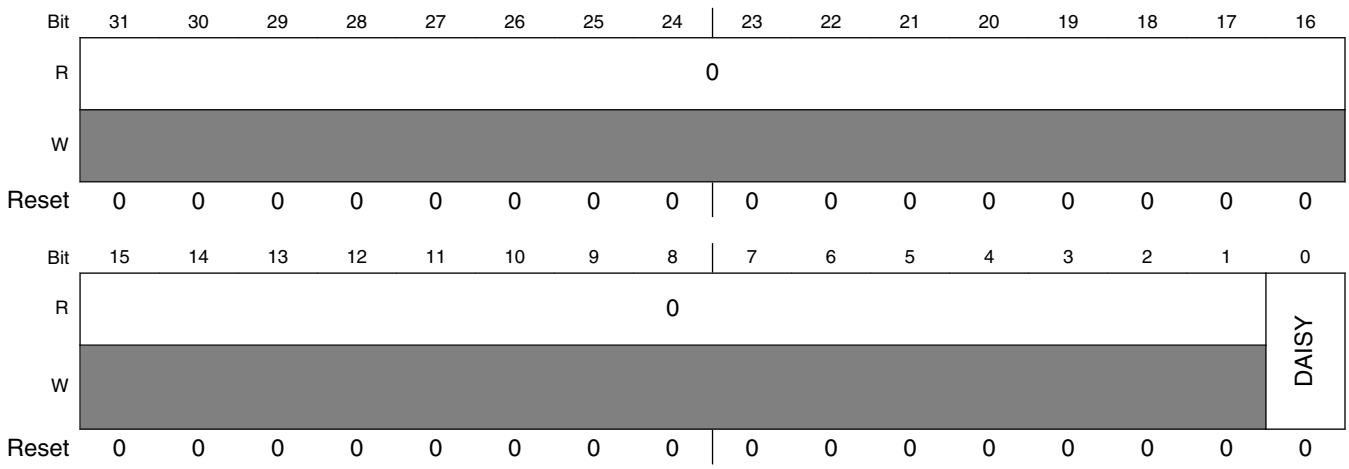


IOMUXC_ENET_REF_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **GPIO16_ALT2** — Selecting ALT2 mode of pad GPIO_16 for ENET_REF_CLK. 1 **RGMII_TX_CTL_ALT7** — Selecting ALT7 mode of pad RGMII_TX_CTL for ENET_REF_CLK.

37.4.511 Select Input Register (IOMUXC_ENET_MAC0_MDIO_SELECT_INPUT)

Address: 20E_0000h base + 810h offset = 20E_0810h

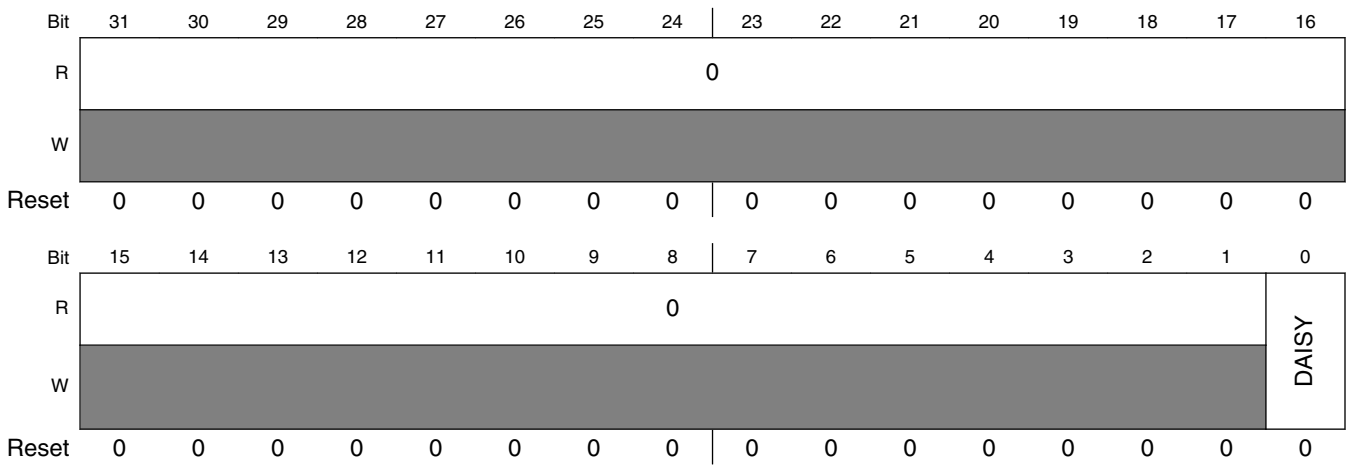


IOMUXC_ENET_MAC0_MDIO_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_MDIO_ALT1** — Selecting ALT1 mode of pad ENET_MDIO for ENET_MDIO. 1 **KEY_COL1_ALT1** — Selecting ALT1 mode of pad KEY_COL1 for ENET_MDIO.

37.4.512 Select Input Register (IOMUXC_ENET_MAC0_RX_CLK_SELECT_INPUT)

Address: 20E_0000h base + 814h offset = 20E_0814h

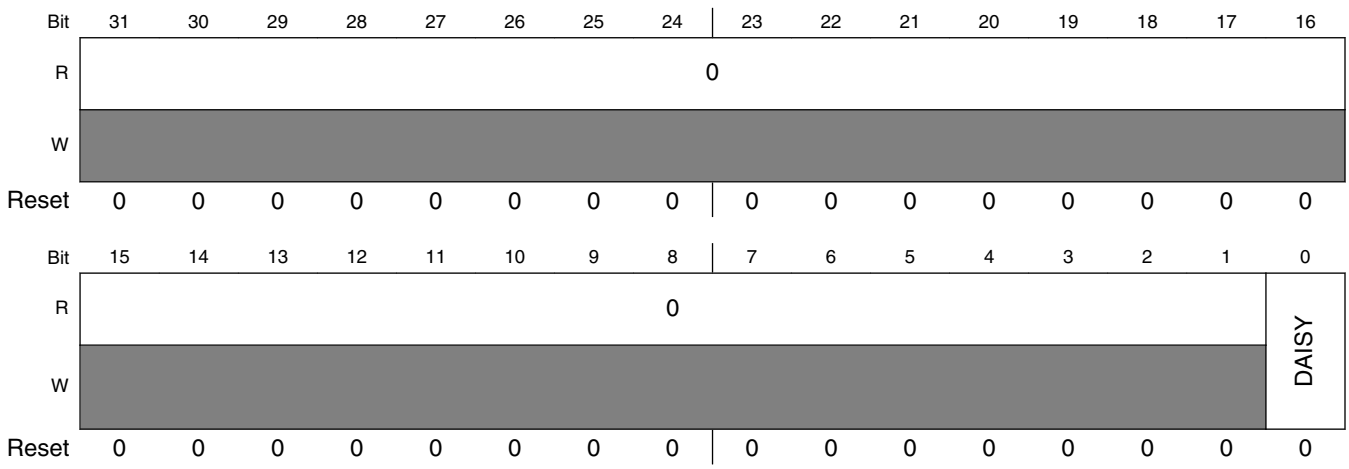


IOMUXC_ENET_MAC0_RX_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **GPIO18_ALT1** — Selecting ALT1 mode of pad GPIO_18 for ENET_RX_CLK. 1 **RGMII_RXC_ALT1** — Selecting ALT1 mode of pad RGMII_RXC for RGMII_RXC.

37.4.513 Select Input Register (IOMUXC_ENET_MAC0_RX_DATA0_SELECT_INPUT)

Address: 20E_0000h base + 818h offset = 20E_0818h

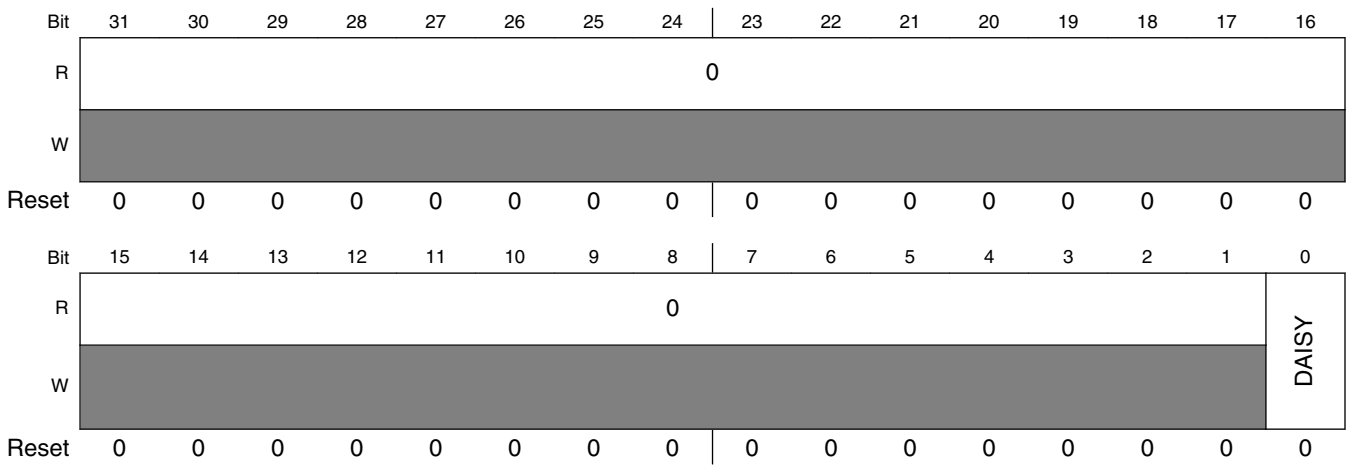


IOMUXC_ENET_MAC0_RX_DATA0_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_RX_DATA0_ALT1** — Selecting ALT1 mode of pad ENET_RXD0 for ENET_RX_DATA0. 1 **RGMII_RD0_ALT1** — Selecting ALT1 mode of pad RGMII_RD0 for RGMII_RD0.

37.4.514 Select Input Register (IOMUXC_ENET_MAC0_RX_DATA1_SELECT_INPUT)

Address: 20E_0000h base + 81Ch offset = 20E_081Ch

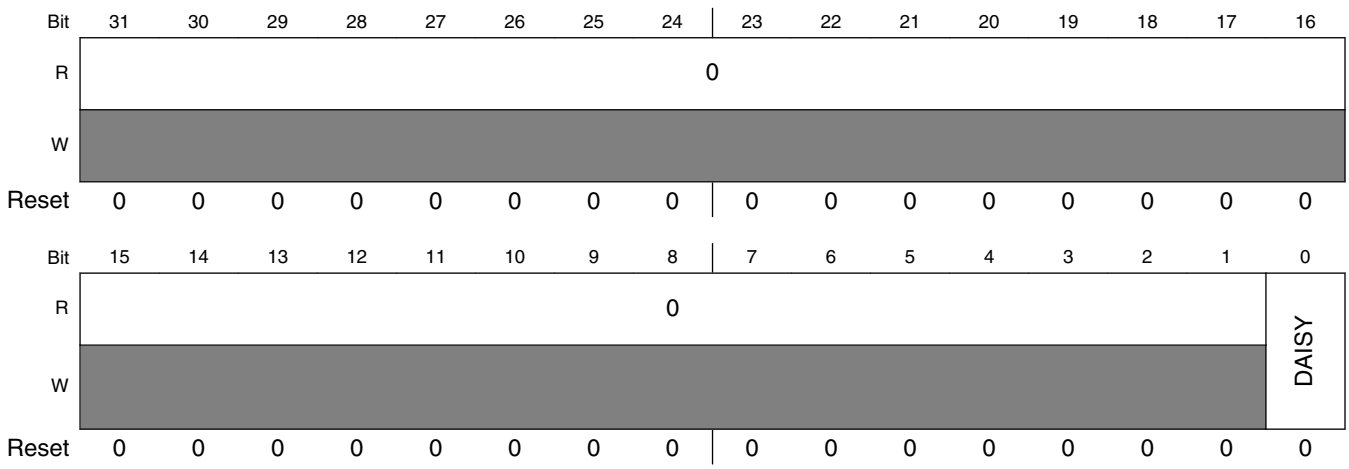


IOMUXC_ENET_MAC0_RX_DATA1_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_RX_DATA1_ALT1** — Selecting ALT1 mode of pad ENET_RXD1 for ENET_RX_DATA1. 1 **RGMII_RD1_ALT1** — Selecting ALT1 mode of pad RGMII_RD1 for RGMII_RD1.

37.4.515 Select Input Register (IOMUXC_ENET_MAC0_RX_DATA2_SELECT_INPUT)

Address: 20E_0000h base + 820h offset = 20E_0820h



IOMUXC_ENET_MAC0_RX_DATA2_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **KEY_COL2_ALT1** — Selecting ALT1 mode of pad KEY_COL2 for ENET_RX_DATA2. 1 **RGMII_RD2_ALT1** — Selecting ALT1 mode of pad RGMII_RD2 for RGMII_RD2.

37.4.516 Select Input Register (IOMUXC_ENET_MAC0_RX_DATA3_SELECT_INPUT)

Address: 20E_0000h base + 824h offset = 20E_0824h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

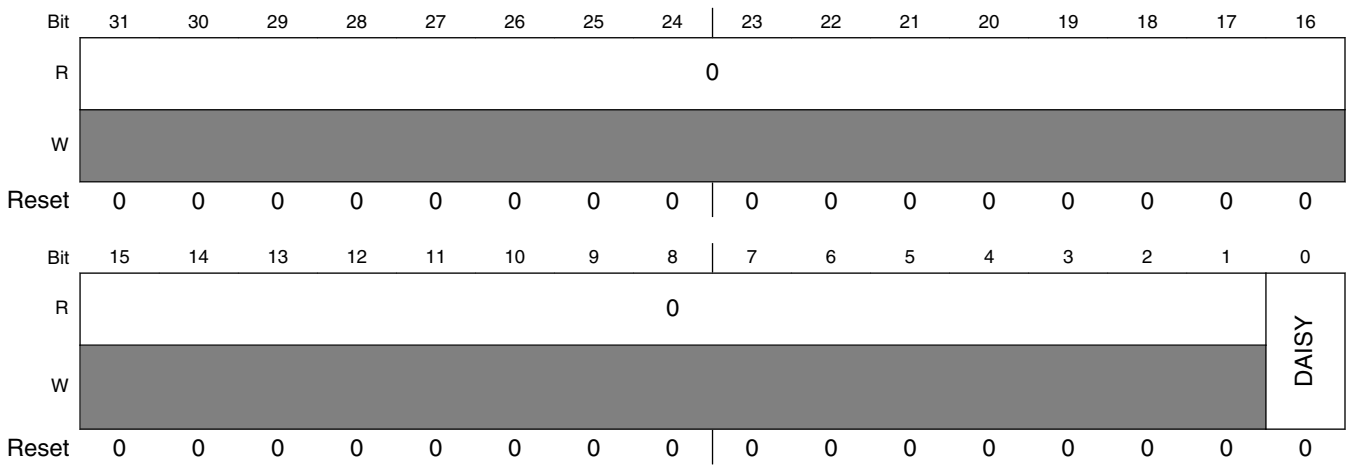
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ENET_MAC0_RX_DATA3_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **KEY_COL0_ALT1** — Selecting ALT1 mode of pad KEY_COL0 for ENET_RX_DATA3. 1 **RGMII_RD3_ALT1** — Selecting ALT1 mode of pad RGMII_RD3 for RGMII_RD3.

37.4.517 Select Input Register (IOMUXC_ENET_MAC0_RX_EN_SELECT_INPUT)

Address: 20E_0000h base + 828h offset = 20E_0828h



IOMUXC_ENET_MAC0_RX_EN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_CRS_DV_ALT1** — Selecting ALT1 mode of pad ENET_CRS_DV for ENET_RX_EN. 1 **RGMII_RX_CTL_ALT1** — Selecting ALT1 mode of pad RGMII_RX_CTL for RGMII_RX_CTL.

37.4.518 Select Input Register (IOMUXC_ESAI_RX_FS_SELECT_INPUT)

Address: 20E_0000h base + 82Ch offset = 20E_082Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

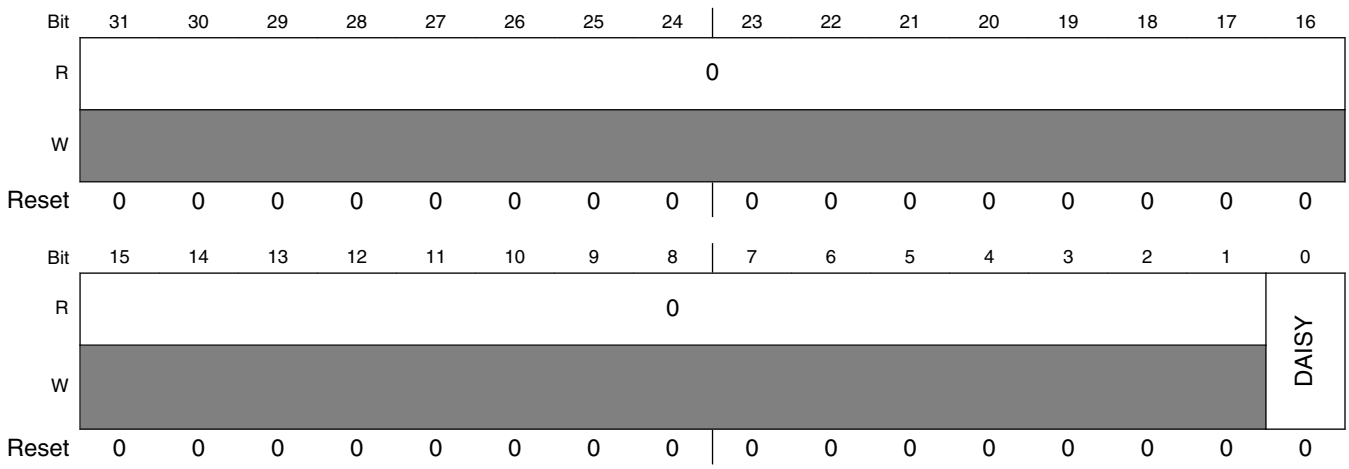
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ESAI_RX_FS_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_REF_CLK_ALT2** — Selecting ALT2 mode of pad ENET_REF_CLK for ESAI_RX_FS. 1 **GPIO09_ALT0** — Selecting ALT0 mode of pad GPIO_9 for ESAI_RX_FS.

37.4.519 Select Input Register (IOMUXC_ESAI_TX_FS_SELECT_INPUT)

Address: 20E_0000h base + 830h offset = 20E_0830h



IOMUXC_ESAI_TX_FS_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_RX_DATA1_ALT2** — Selecting ALT2 mode of pad ENET_RXD1 for ESAI_TX_FS. 1 **GPIO02_ALT0** — Selecting ALT0 mode of pad GPIO_2 for ESAI_TX_FS.

37.4.520 Select Input Register (IOMUXC_ESAI_RX_HF_CLK_SELECT_INPUT)

Address: 20E_0000h base + 834h offset = 20E_0834h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

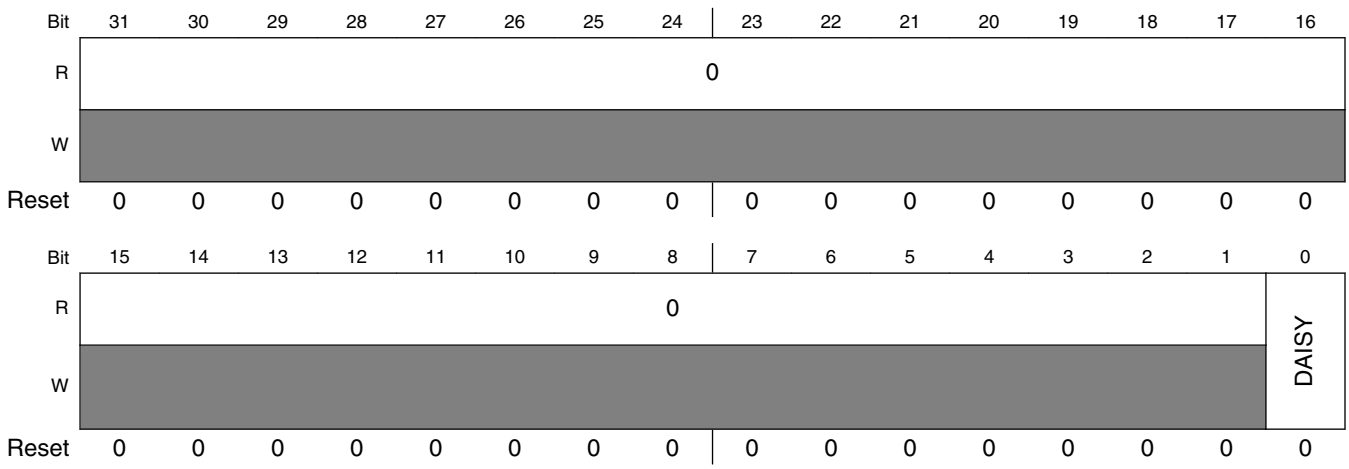
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ESAI_RX_HF_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_RX_ER_ALT2** — Selecting ALT2 mode of pad ENET_RX_ER for ESAI_RX_HF_CLK. 1 **GPIO03_ALT0** — Selecting ALT0 mode of pad GPIO_3 for ESAI_RX_HF_CLK.

37.4.521 Select Input Register (IOMUXC_ESAI_TX_HF_CLK_SELECT_INPUT)

Address: 20E_0000h base + 838h offset = 20E_0838h

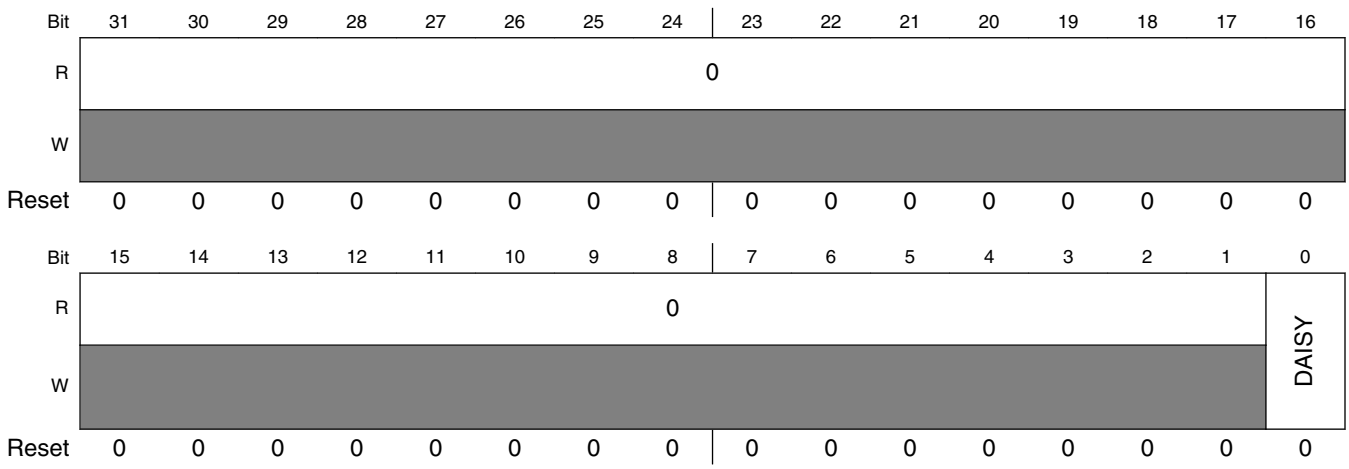


IOMUXC_ESAI_TX_HF_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_RX_DATA0_ALT2** — Selecting ALT2 mode of pad ENET_RXD0 for ESAI_TX_HF_CLK. 1 **GPIO04_ALT0** — Selecting ALT0 mode of pad GPIO_4 for ESAI_TX_HF_CLK.

37.4.522 Select Input Register (IOMUXC_ESAI_RX_CLK_SELECT_INPUT)

Address: 20E_0000h base + 83Ch offset = 20E_083Ch

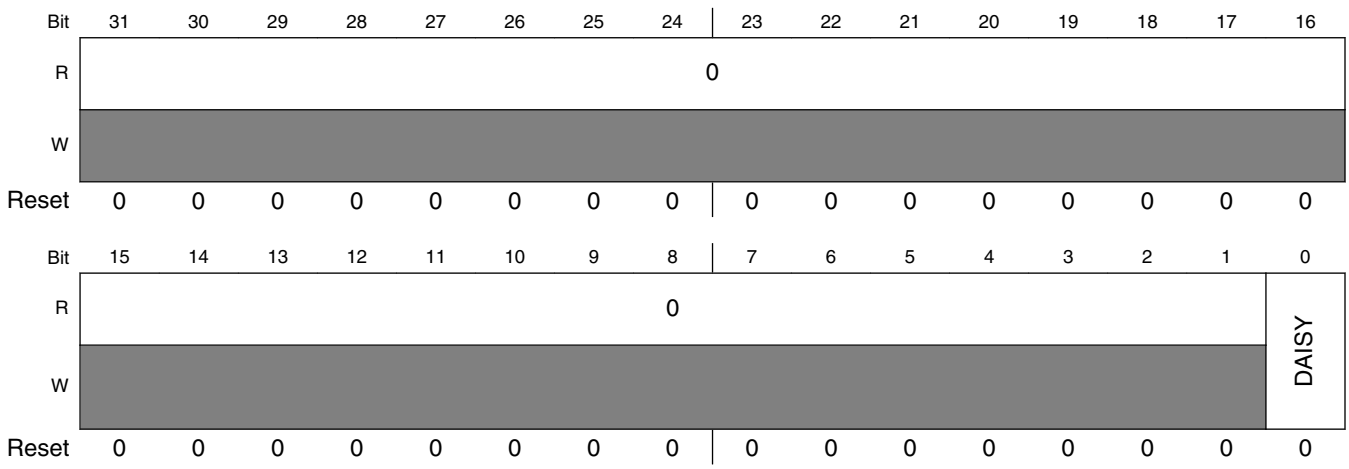


IOMUXC_ESAI_RX_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_MDIO_ALT2** — Selecting ALT2 mode of pad ENET_MDIO for ESAI_RX_CLK. 1 **GPIO01_ALT0** — Selecting ALT0 mode of pad GPIO_1 for ESAI_RX_CLK.

37.4.523 Select Input Register (IOMUXC_ESAI_TX_CLK_SELECT_INPUT)

Address: 20E_0000h base + 840h offset = 20E_0840h

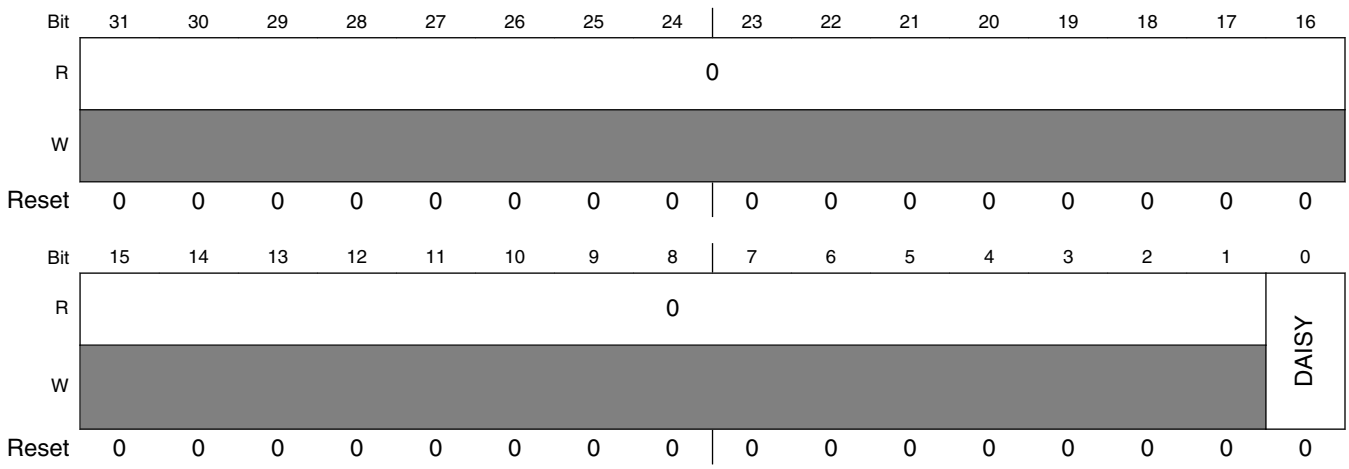


IOMUXC_ESAI_TX_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_CRS_DV_ALT2** — Selecting ALT2 mode of pad ENET_CRS_DV for ESAI_TX_CLK. 1 **GPIO06_ALT0** — Selecting ALT0 mode of pad GPIO_6 for ESAI_TX_CLK.

37.4.524 Select Input Register (IOMUXC_ESAI_SDO0_SELECT_INPUT)

Address: 20E_0000h base + 844h offset = 20E_0844h

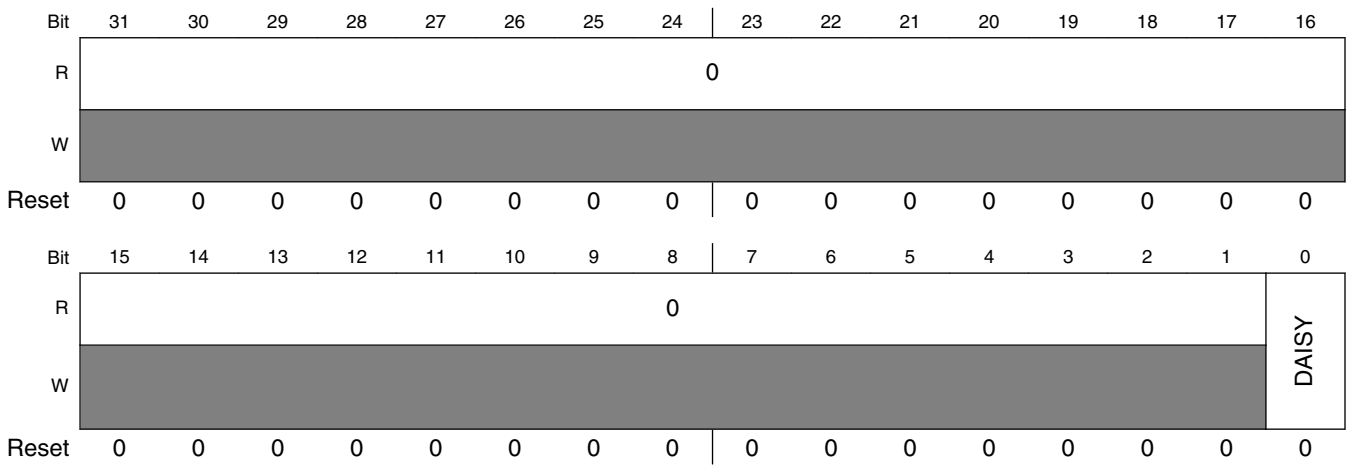


IOMUXC_ESAI_SDO0_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **GPIO17_ALT0** — Selecting ALT0 mode of pad GPIO_17 for ESAI_TX0. 1 **NAND_CS2_B_ALT2** — Selecting ALT2 mode of pad NANDF_CS2 for ESAI_TX0.

37.4.525 Select Input Register (IOMUXC_ESAI_SDO1_SELECT_INPUT)

Address: 20E_0000h base + 848h offset = 20E_0848h

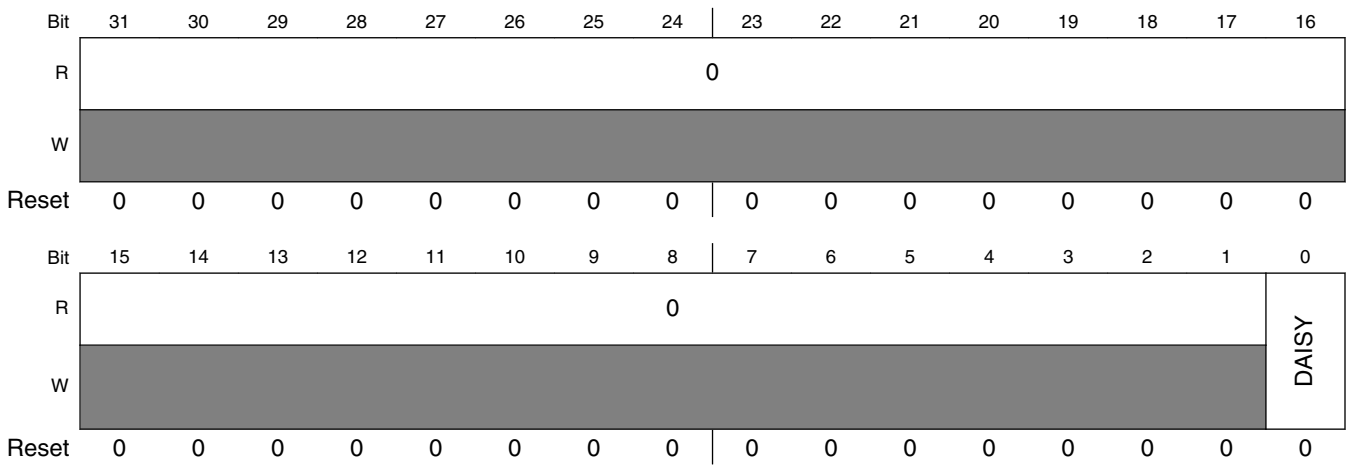


IOMUXC_ESAI_SDO1_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **GPIO18_ALT0** — Selecting ALT0 mode of pad GPIO_18 for ESAI_TX1. 1 **NAND_CS3_B_ALT2** — Selecting ALT2 mode of pad NANDF_CS3 for ESAI_TX1.

37.4.526 Select Input Register (IOMUXC_ESAI_SDO2_SDI3_SELECT_INPUT)

Address: 20E_0000h base + 84Ch offset = 20E_084Ch

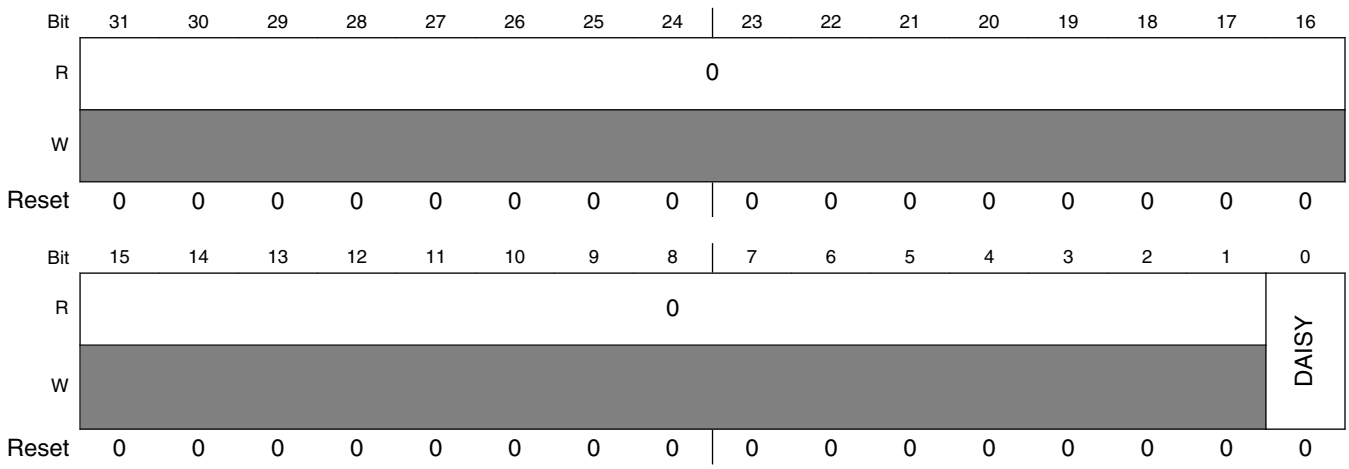


IOMUXC_ESAI_SDO2_SDI3_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_TX_DATA1_ALT2** — Selecting ALT2 mode of pad ENET_TXD1 for ESAI_TX2_RX3. 1 **GPIO05_ALT0** — Selecting ALT0 mode of pad GPIO_5 for ESAI_TX2_RX3.

37.4.527 Select Input Register (IOMUXC_ESAI_SDO3_SDI2_SELECT_INPUT)

Address: 20E_0000h base + 850h offset = 20E_0850h



IOMUXC_ESAI_SDO3_SDI2_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_TX_EN_ALT2** — Selecting ALT2 mode of pad ENET_TX_EN for ESAI_TX3_RX2. 1 **GPIO16_ALT0** — Selecting ALT0 mode of pad GPIO_16 for ESAI_TX3_RX2.

37.4.528 Select Input Register (IOMUXC_ESAI_SDO4_SDI1_SELECT_INPUT)

Address: 20E_0000h base + 854h offset = 20E_0854h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

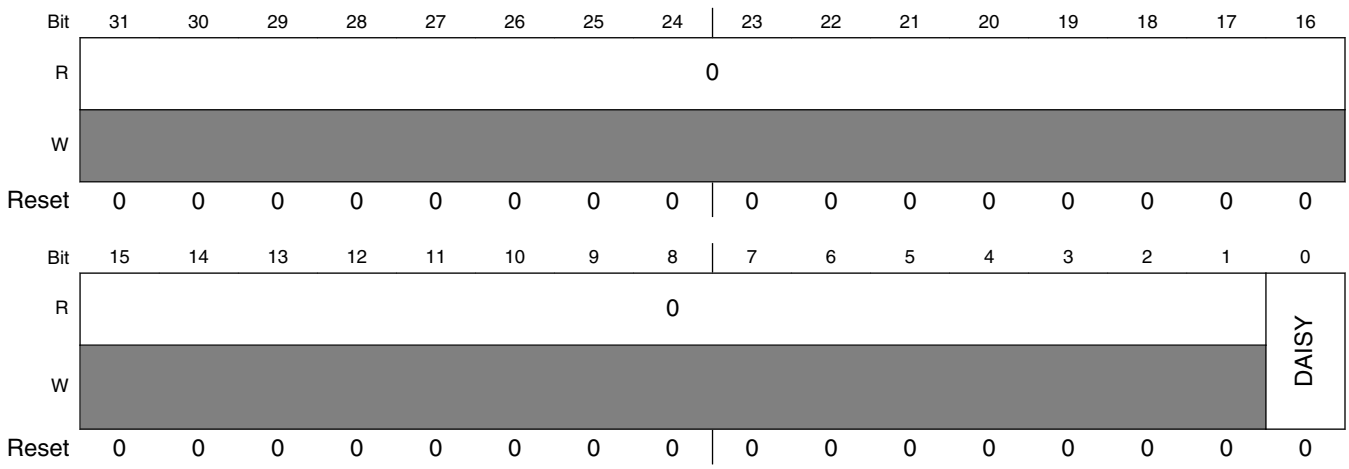
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ESAI_SDO4_SDI1_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_TX_DATA0_ALT2** — Selecting ALT2 mode of pad ENET_TXD0 for ESAI_TX4_RX1. 1 **GPIO07_ALT0** — Selecting ALT0 mode of pad GPIO_7 for ESAI_TX4_RX1.

37.4.529 Select Input Register (IOMUXC_ESAI_SDO5_SDI0_SELECT_INPUT)

Address: 20E_0000h base + 858h offset = 20E_0858h

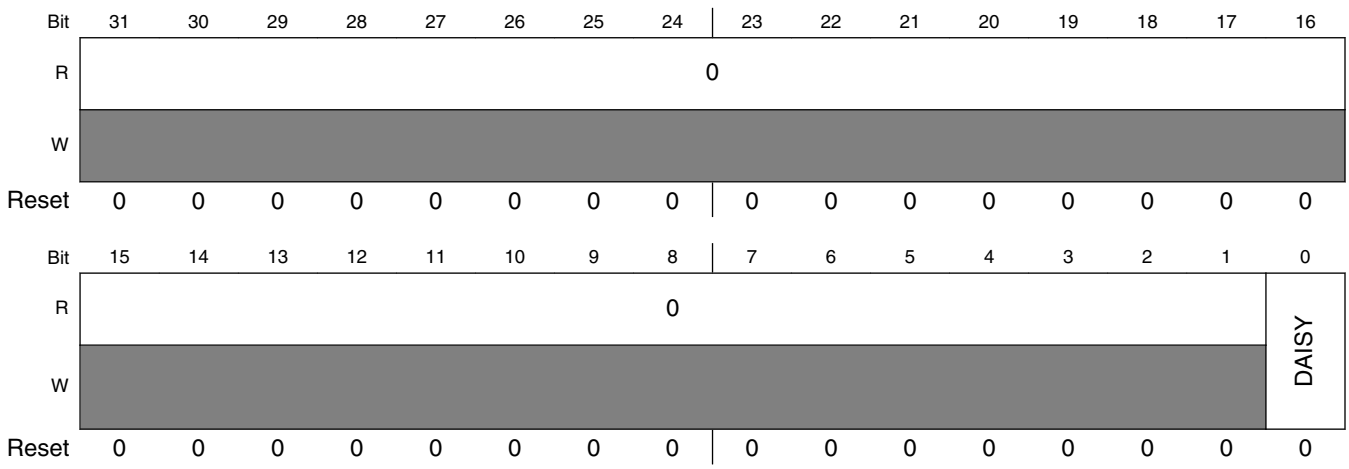


IOMUXC_ESAI_SDO5_SDI0_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_MDC_ALT2** — Selecting ALT2 mode of pad ENET_MDC for ESAI_TX5_RX0. 1 **GPIO08_ALT0** — Selecting ALT0 mode of pad GPIO_8 for ESAI_TX5_RX0.

37.4.530 Select Input Register (IOMUXC_HDMI_ICECIN_SELECT_INPUT)

Address: 20E_0000h base + 85Ch offset = 20E_085Ch

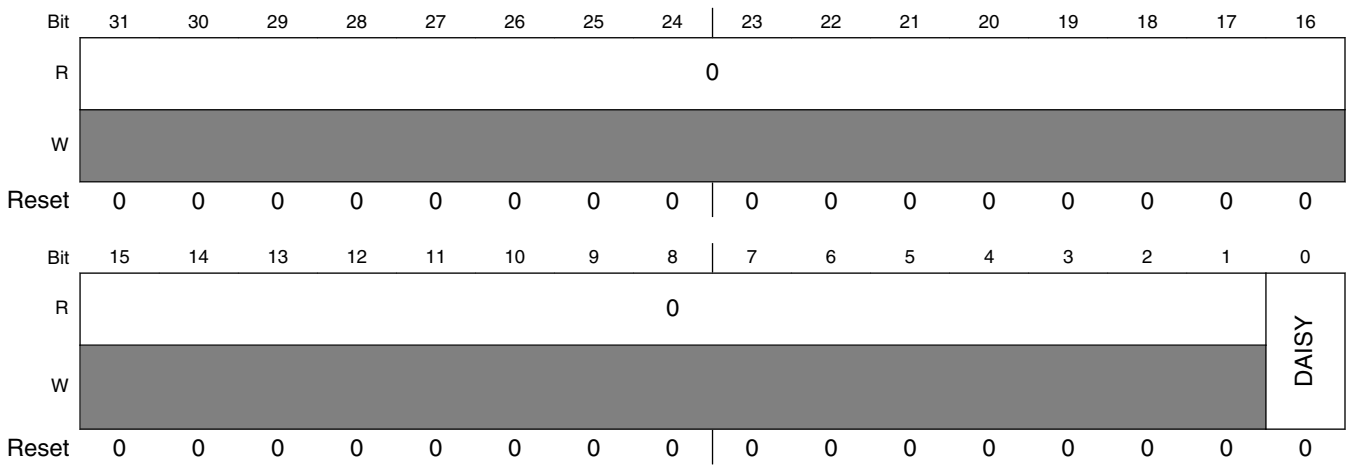


IOMUXC_HDMI_ICECIN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_ADDR25_ALT6** — Selecting ALT6 mode of pad EIM_A25 for HDMI_TX_CEC_LINE. 1 **KEY_ROW2_ALT6** — Selecting ALT6 mode of pad KEY_ROW2 for HDMI_TX_CEC_LINE.

37.4.531 Select Input Register (IOMUXC_HDMI_II2C_CLKIN_SELECT_INPUT)

Address: 20E_0000h base + 860h offset = 20E_0860h

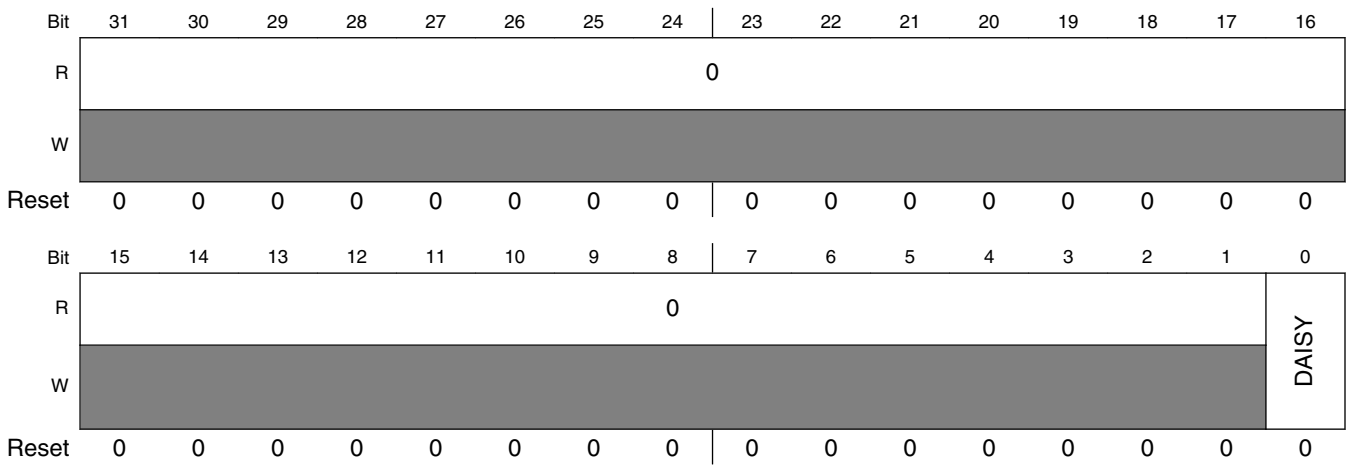


IOMUXC_HDMI_II2C_CLKIN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_EB2_B_ALT4** — Selecting ALT4 mode of pad EIM_EB2 for HDMI_TX_DDC_SCL. 1 **KEY_COL3_ALT2** — Selecting ALT2 mode of pad KEY_COL3 for HDMI_TX_DDC_SCL.

37.4.532 Select Input Register (IOMUXC_HDMI_II2C_DATAIN_SELECT_INPUT)

Address: 20E_0000h base + 864h offset = 20E_0864h

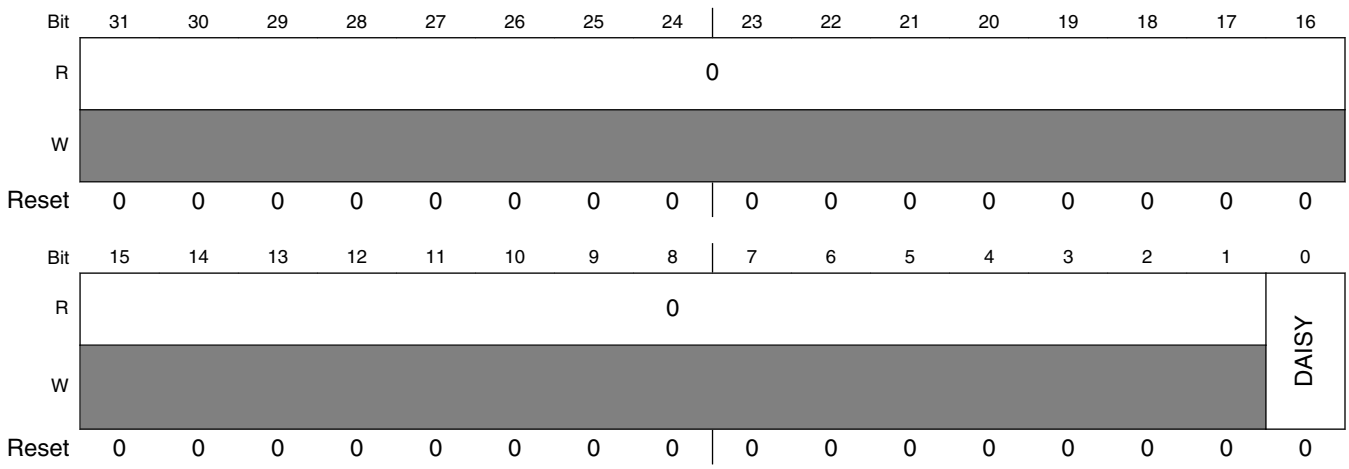


IOMUXC_HDMI_II2C_DATAIN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_DATA16_ALT4** — Selecting ALT4 mode of pad EIM_D16 for HDMI_TX_DDC_SDA. 1 **KEY_ROW3_ALT2** — Selecting ALT2 mode of pad KEY_ROW3 for HDMI_TX_DDC_SDA.

37.4.533 Select Input Register (IOMUXC_I2C1_SCL_IN_SELECT_INPUT)

Address: 20E_0000h base + 868h offset = 20E_0868h

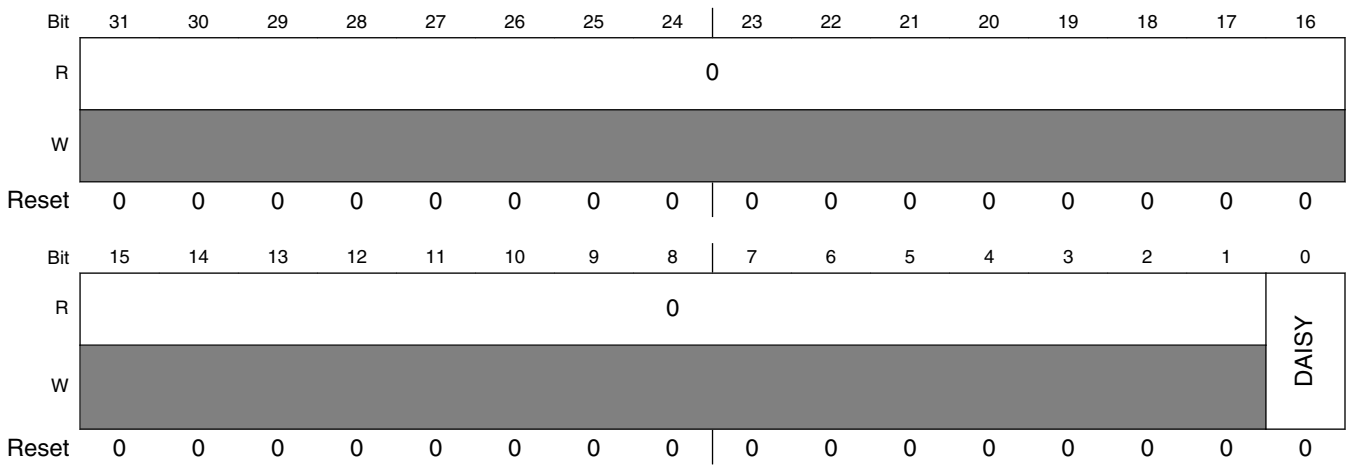


IOMUXC_I2C1_SCL_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **CSI0_DATA09_ALT4** — Selecting ALT4 mode of pad CSI0_DAT9 for I2C1_SCL. 1 **EIM_DATA21_ALT6** — Selecting ALT6 mode of pad EIM_D21 for I2C1_SCL.

37.4.534 Select Input Register (IOMUXC_I2C1_SDA_IN_SELECT_INPUT)

Address: 20E_0000h base + 86Ch offset = 20E_086Ch

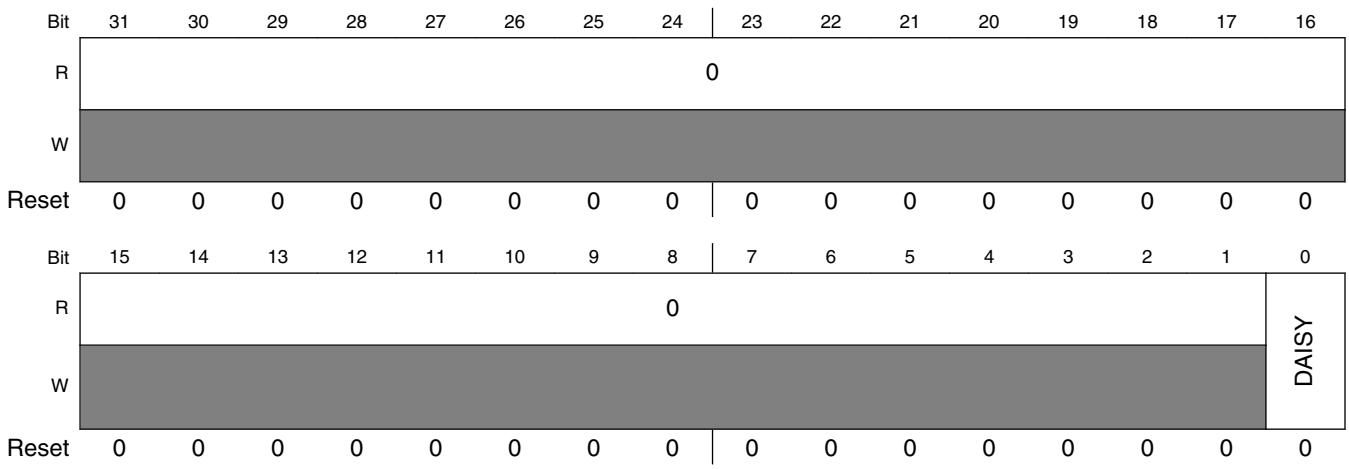


IOMUXC_I2C1_SDA_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **CSI0_DATA08_ALT4** — Selecting ALT4 mode of pad CSI0_DAT8 for I2C1_SDA. 1 **EIM_DATA28_ALT1** — Selecting ALT1 mode of pad EIM_D28 for I2C1_SDA.

37.4.535 Select Input Register (IOMUXC_I2C2_SCL_IN_SELECT_INPUT)

Address: 20E_0000h base + 870h offset = 20E_0870h

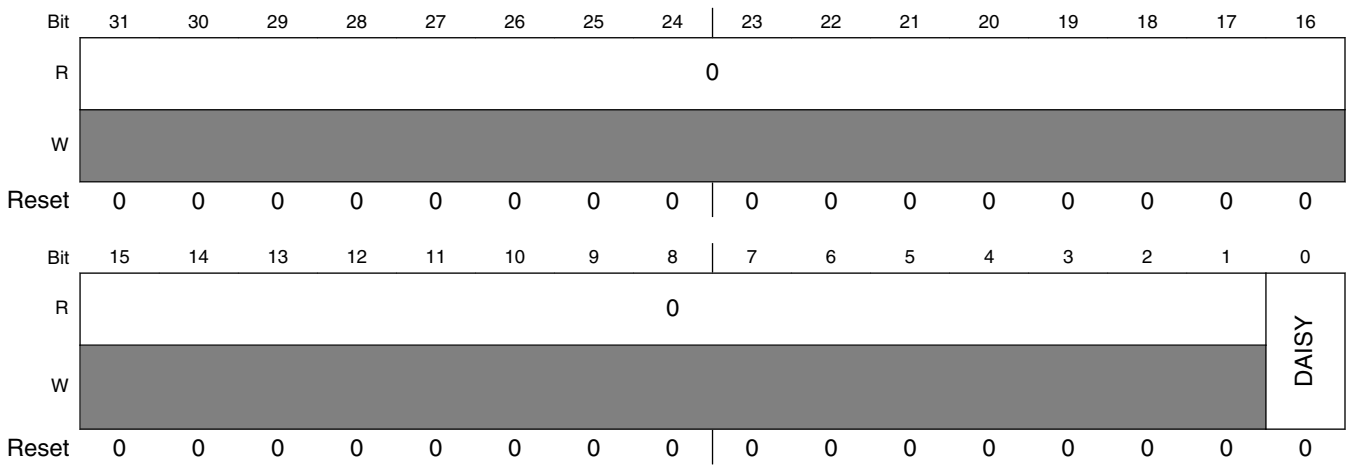


IOMUXC_I2C2_SCL_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_EB2_B_ALT6** — Selecting ALT6 mode of pad EIM_EB2 for I2C2_SCL. 1 **KEY_COL3_ALT4** — Selecting ALT4 mode of pad KEY_COL3 for I2C2_SCL.

37.4.536 Select Input Register (IOMUXC_I2C2_SDA_IN_SELECT_INPUT)

Address: 20E_0000h base + 874h offset = 20E_0874h

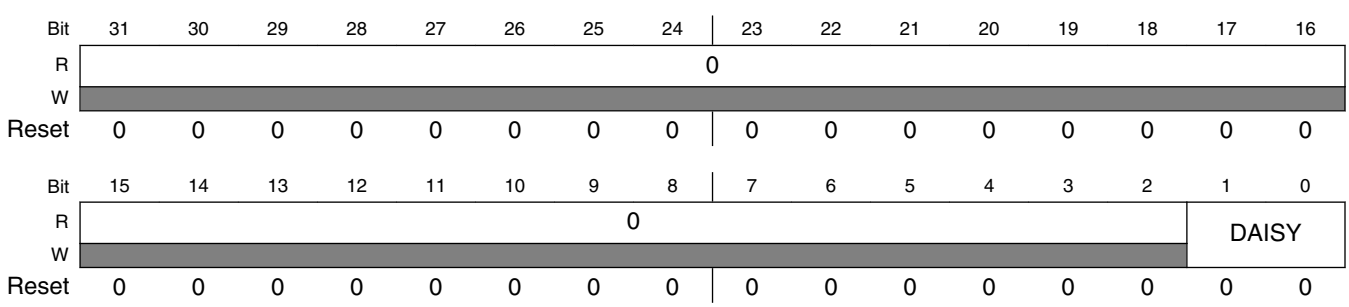


IOMUXC_I2C2_SDA_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_DATA16_ALT6** — Selecting ALT6 mode of pad EIM_D16 for I2C2_SDA. 1 **KEY_ROW3_ALT4** — Selecting ALT4 mode of pad KEY_ROW3 for I2C2_SDA.

37.4.537 Select Input Register (IOMUXC_I2C3_SCL_IN_SELECT_INPUT)

Address: 20E_0000h base + 878h offset = 20E_0878h



IOMUXC_I2C3_SCL_IN_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **EIM_DATA17_ALT6** — Selecting ALT6 mode of pad EIM_D17 for I2C3_SCL. 01 **GPIO03_ALT2** — Selecting ALT2 mode of pad GPIO_3 for I2C3_SCL. 10 **GPIO05_ALT6** — Selecting ALT6 mode of pad GPIO_5 for I2C3_SCL.

37.4.538 Select Input Register (IOMUXC_I2C3_SDA_IN_SELECT_INPUT)

Address: 20E_0000h base + 87Ch offset = 20E_087Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

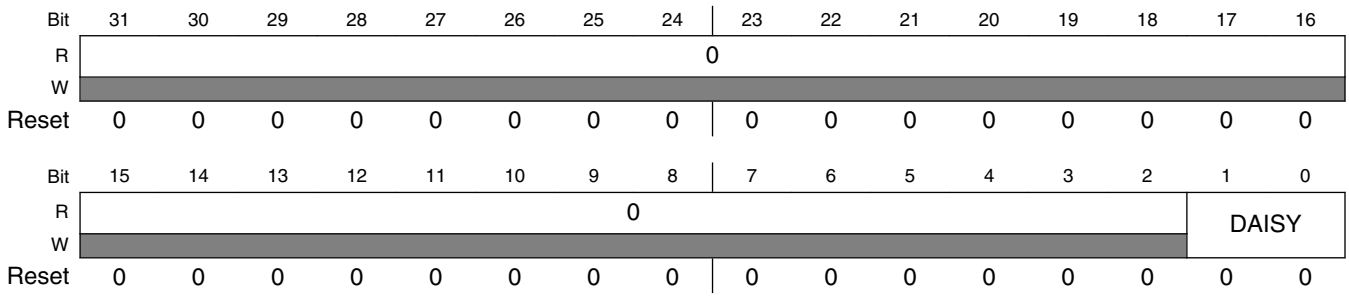
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DAISY	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_I2C3_SDA_IN_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **EIM_DATA18_ALT6** — Selecting ALT6 mode of pad EIM_D18 for I2C3_SDA. 01 **GPIO16_ALT6** — Selecting ALT6 mode of pad GPIO_16 for I2C3_SDA. 10 **GPIO06_ALT2** — Selecting ALT2 mode of pad GPIO_6 for I2C3_SDA.

37.4.539 Select Input Register (IOMUXC_I2C4_SCL_IN_SELECT_INPUT)

Address: 20E_0000h base + 880h offset = 20E_0880h

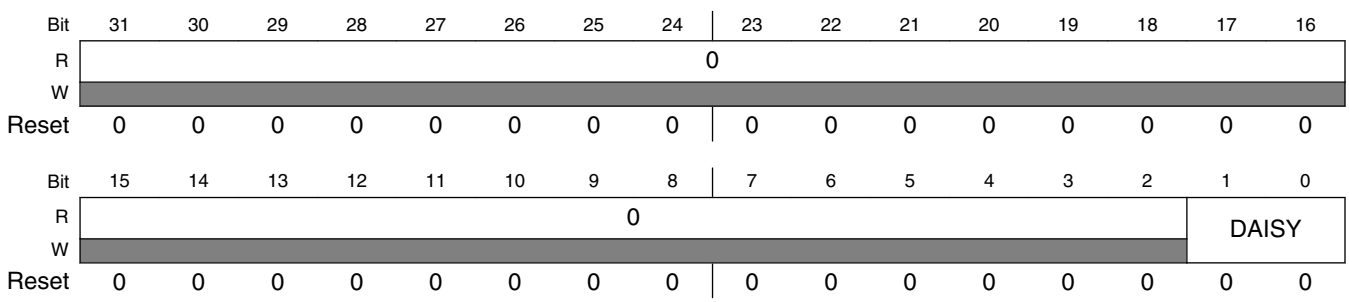


IOMUXC_I2C4_SCL_IN_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **ENET_TX_EN_ALT9** — Selecting ALT9 mode of pad ENET_TX_EN for I2C4_SCL. 01 **GPIO07_ALT8** — Selecting ALT8 mode of pad GPIO_7 for I2C4_SCL. 10 **NAND_WP_B_ALT9** — Selecting ALT9 mode of pad NANDF_WP_B for I2C4_SCL.

37.4.540 Select Input Register (IOMUXC_I2C4_SDA_IN_SELECT_INPUT)

Address: 20E_0000h base + 884h offset = 20E_0884h



IOMUXC_I2C4_SDA_IN_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IOMUXC_I2C4_SDA_IN_SELECT_INPUT field descriptions (continued)

Field	Description
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **ENET_TX_DATA1_ALT9** — Selecting ALT9 mode of pad ENET_TXD1 for I2C4_SDA. 01 **GPIO08_ALT8** — Selecting ALT8 mode of pad GPIO_8 for I2C4_SDA. 10 **NAND_CS3_B_ALT9** — Selecting ALT9 mode of pad NANDF_CS3 for I2C4_SDA.

37.4.541 Select Input Register (IOMUXC_IPU1_SENS1_DATA10_SELECT_INPUT)

Address: 20E_0000h base + 888h offset = 20E_0888h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_IPU1_SENS1_DATA10_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_DATA22_ALT3** — Selecting ALT3 mode of pad EIM_D22 for IPU1_CSI1_DATA10. 1 **EIM_EB1_B_ALT2** — Selecting ALT2 mode of pad EIM_EB1 for IPU1_CSI1_DATA10.

37.4.542 Select Input Register (IOMUXC_IPU1_SENS1_DATA11_SELECT_INPUT)

Address: 20E_0000h base + 88Ch offset = 20E_088Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

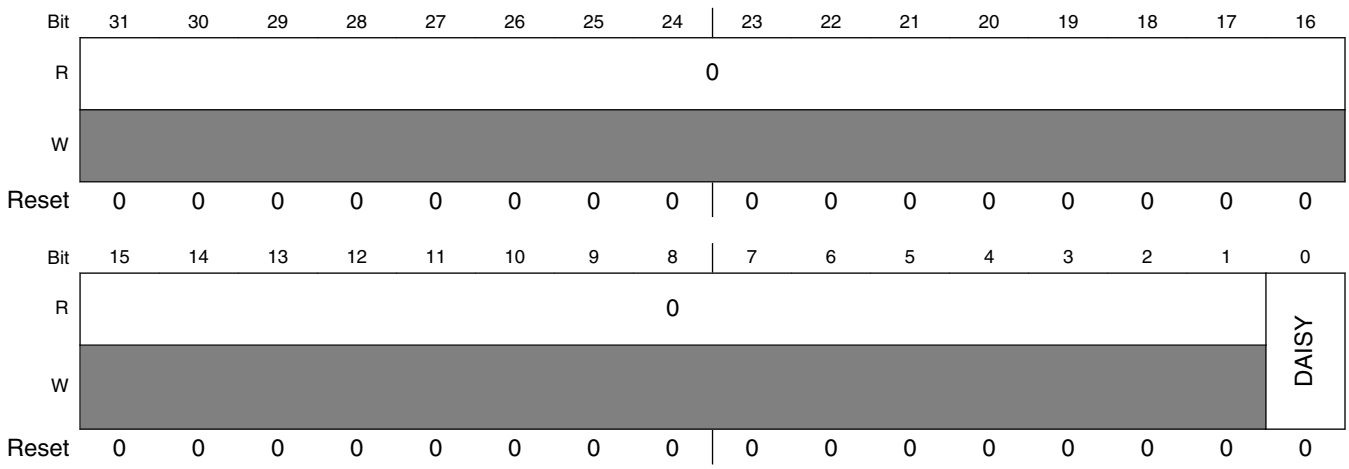
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_IPU1_SENS1_DATA11_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_DATA21_ALT3** — Selecting ALT3 mode of pad EIM_D21 for IPU1_CSI1_DATA11. 1 **EIM_EB0_B_ALT2** — Selecting ALT2 mode of pad EIM_EB0 for IPU1_CSI1_DATA11.

37.4.543 Select Input Register (IOMUXC_IPU1_SENS1_DATA12_SELECT_INPUT)

Address: 20E_0000h base + 890h offset = 20E_0890h



IOMUXC_IPU1_SENS1_DATA12_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_ADDR17_ALT2** — Selecting ALT2 mode of pad EIM_A17 for IPU1_CSI1_DATA12. 1 **EIM_DATA28_ALT3** — Selecting ALT3 mode of pad EIM_D28 for IPU1_CSI1_DATA12.

37.4.544 Select Input Register (IOMUXC_IPU1_SENS1_DATA13_SELECT_INPUT)

Address: 20E_0000h base + 894h offset = 20E_0894h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

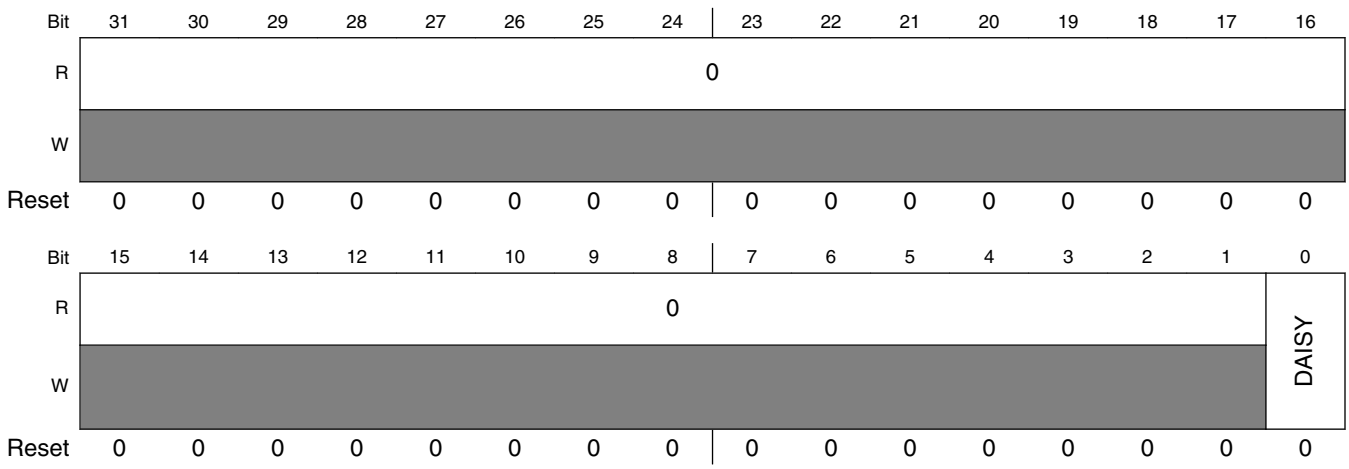
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_IPU1_SENS1_DATA13_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_ADDR18_ALT2** — Selecting ALT2 mode of pad EIM_A18 for IPU1_CSI1_DATA13. 1 **EIM_DATA27_ALT3** — Selecting ALT3 mode of pad EIM_D27 for IPU1_CSI1_DATA13.

37.4.545 Select Input Register (IOMUXC_IPU1_SENS1_DATA14_SELECT_INPUT)

Address: 20E_0000h base + 898h offset = 20E_0898h



IOMUXC_IPU1_SENS1_DATA14_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_ADDR19_ALT2** — Selecting ALT2 mode of pad EIM_A19 for IPU1_CSI1_DATA14. 1 **EIM_DATA26_ALT3** — Selecting ALT3 mode of pad EIM_D26 for IPU1_CSI1_DATA14.

37.4.546 Select Input Register (IOMUXC_IPU1_SENS1_DATA15_SELECT_INPUT)

Address: 20E_0000h base + 89Ch offset = 20E_089Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

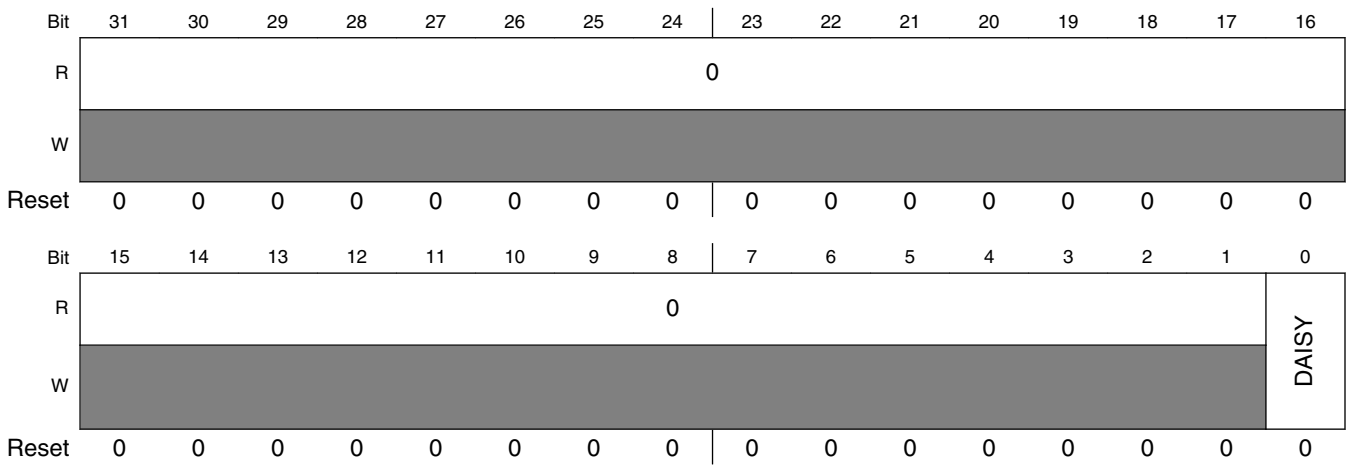
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_IPU1_SENS1_DATA15_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_ADDR20_ALT2** — Selecting ALT2 mode of pad EIM_A20 for IPU1_CSI1_DATA15. 1 **EIM_DATA20_ALT3** — Selecting ALT3 mode of pad EIM_D20 for IPU1_CSI1_DATA15.

37.4.547 Select Input Register (IOMUXC_IPU1_SENS1_DATA16_SELECT_INPUT)

Address: 20E_0000h base + 8A0h offset = 20E_08A0h



IOMUXC_IPU1_SENS1_DATA16_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_ADDR21_ALT2** — Selecting ALT2 mode of pad EIM_A21 for IPU1_CSI1_DATA16. 1 **EIM_DATA19_ALT3** — Selecting ALT3 mode of pad EIM_D19 for IPU1_CSI1_DATA16.

37.4.548 Select Input Register (IOMUXC_IPU1_SENS1_DATA17_SELECT_INPUT)

Address: 20E_0000h base + 8A4h offset = 20E_08A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

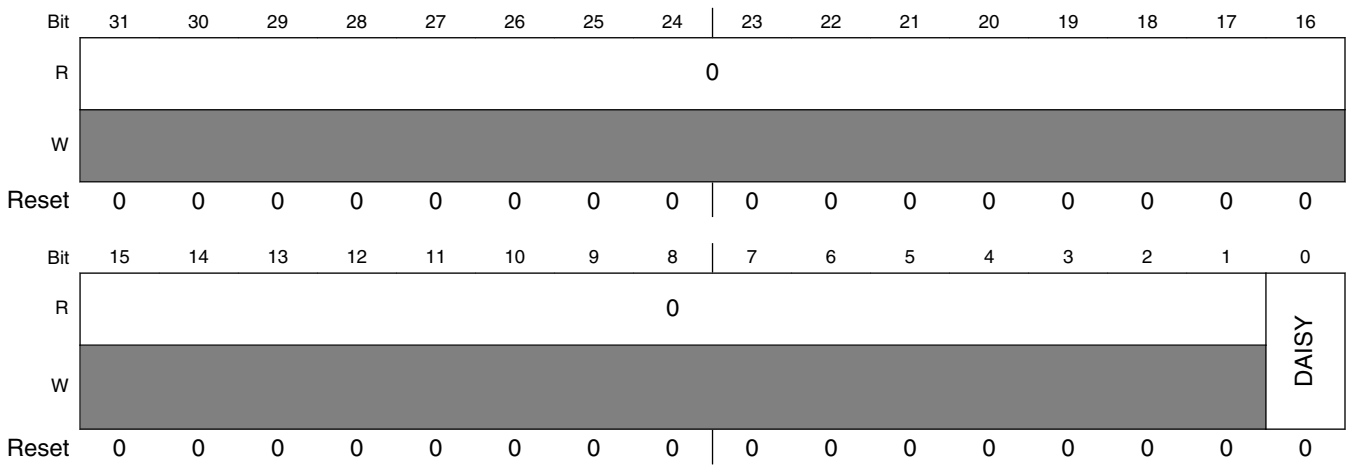
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_IPU1_SENS1_DATA17_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_ADDR22_ALT2** — Selecting ALT2 mode of pad EIM_A22 for IPU1_CSI1_DATA17. 1 **EIM_DATA18_ALT3** — Selecting ALT3 mode of pad EIM_D18 for IPU1_CSI1_DATA17.

37.4.549 Select Input Register (IOMUXC_IPU1_SENS1_DATA18_SELECT_INPUT)

Address: 20E_0000h base + 8A8h offset = 20E_08A8h



IOMUXC_IPU1_SENS1_DATA18_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_ADDR23_ALT2** — Selecting ALT2 mode of pad EIM_A23 for IPU1_CSI1_DATA18. 1 **EIM_DATA16_ALT3** — Selecting ALT3 mode of pad EIM_D16 for IPU1_CSI1_DATA18.

37.4.550 Select Input Register (IOMUXC_IPU1_SENS1_DATA19_SELECT_INPUT)

Address: 20E_0000h base + 8ACh offset = 20E_08ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_IPU1_SENS1_DATA19_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_ADDR24_ALT2** — Selecting ALT2 mode of pad EIM_A24 for IPU1_CSI1_DATA19. 1 **EIM_EB2_B_ALT3** — Selecting ALT3 mode of pad EIM_EB2 for IPU1_CSI1_DATA19.

37.4.551 Select Input Register (IOMUXC_IPU1_SENS1_DATA_EN_SELECT_INPUT)

Address: 20E_0000h base + 8B0h offset = 20E_08B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_IPU1_SENS1_DATA_EN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_DATA23_ALT4** — Selecting ALT4 mode of pad EIM_D23 for IPU1_CSI1_DATA_EN. 1 **EIM_AD10_ALT2** — Selecting ALT2 mode of pad EIM_DA10 for IPU1_CSI1_DATA_EN.

37.4.552 Select Input Register (IOMUXC_IPU1_SENS1_HSYNC_SELECT_INPUT)

Address: 20E_0000h base + 8B4h offset = 20E_08B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

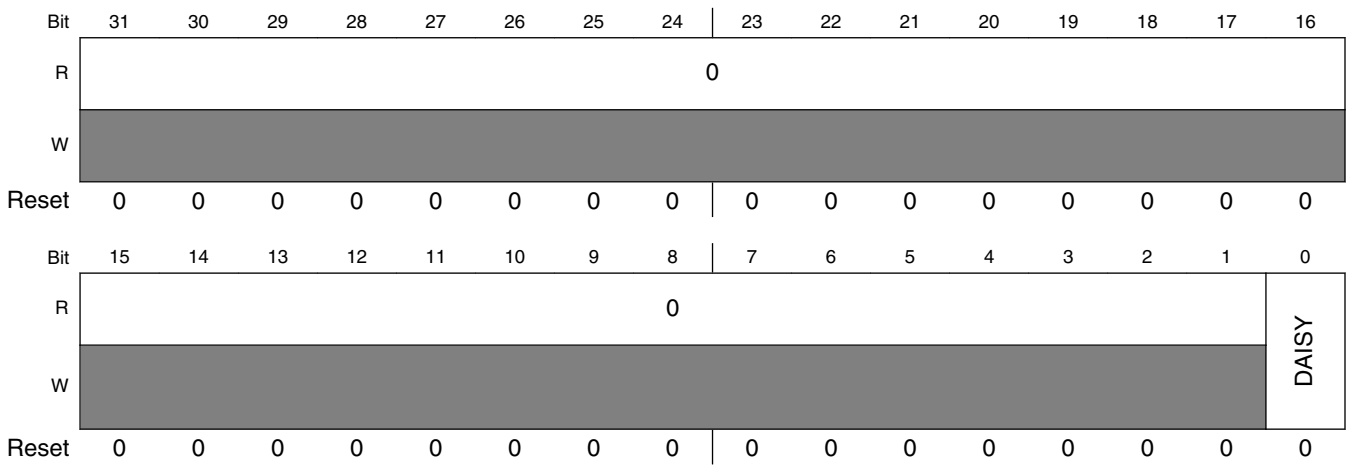
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_IPU1_SENS1_HSYNC_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_AD11_ALT2** — Selecting ALT2 mode of pad EIM_DA11 for IPU1_CSI1_HSYNC. 1 **EIM_EB3_B_ALT4** — Selecting ALT4 mode of pad EIM_EB3 for IPU1_CSI1_HSYNC.

37.4.553 Select Input Register (IOMUXC_IPU1_SENS1_PIX_CLK_SELECT_INPUT)

Address: 20E_0000h base + 8B8h offset = 20E_08B8h

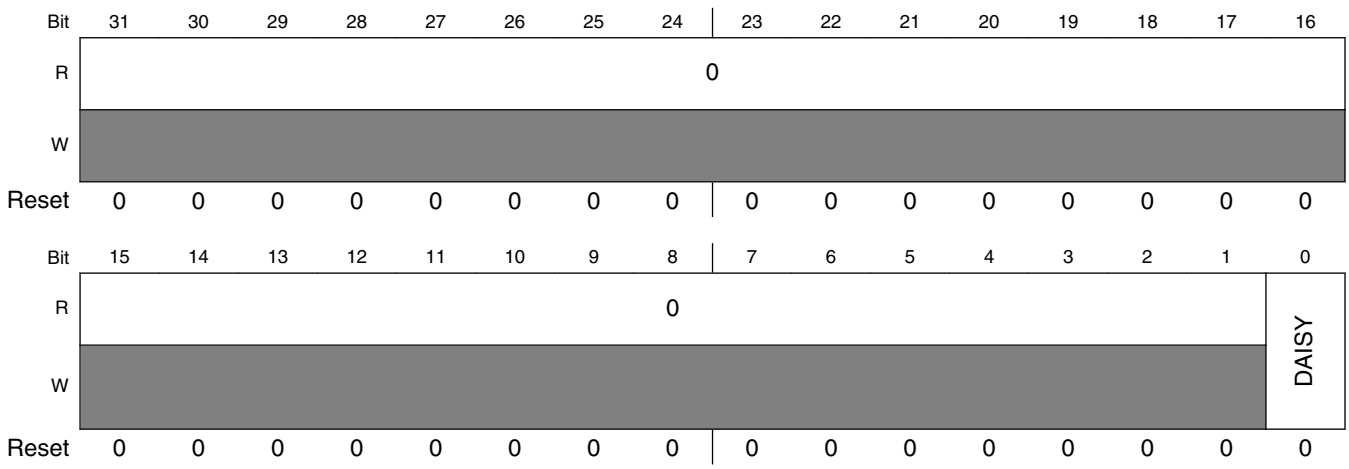


IOMUXC_IPU1_SENS1_PIX_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_ADDR16_ALT2** — Selecting ALT2 mode of pad EIM_A16 for IPU1_CSI1_PIXCLK. 1 **EIM_DATA17_ALT3** — Selecting ALT3 mode of pad EIM_D17 for IPU1_CSI1_PIXCLK.

37.4.554 Select Input Register (IOMUXC_IPU1_SENS1_VSYNC_SELECT_INPUT)

Address: 20E_0000h base + 8BCh offset = 20E_08BCh

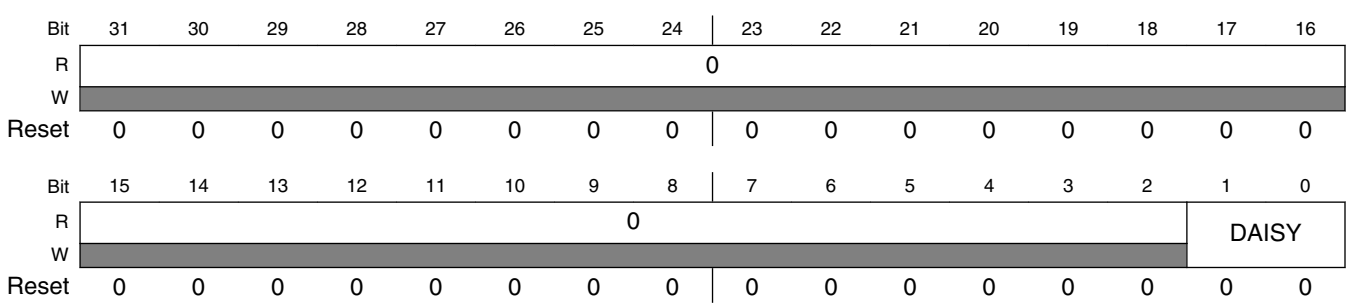


IOMUXC_IPU1_SENS1_VSYNC_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_DATA29_ALT6** — Selecting ALT6 mode of pad EIM_D29 for IPU1_CSI1_VSYNC. 1 **EIM_AD12_ALT2** — Selecting ALT2 mode of pad EIM_DA12 for IPU1_CSI1_VSYNC.

37.4.555 Select Input Register (IOMUXC_KEY_COL5_SELECT_INPUT)

Address: 20E_0000h base + 8C0h offset = 20E_08C0h



IOMUXC_KEY_COL5_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **CSI0_DATA04_ALT3** — Selecting ALT3 mode of pad CSI0_DAT4 for KEY_COL5. 01 **GPIO00_ALT2** — Selecting ALT2 mode of pad GPIO_0 for KEY_COL5. 10 **GPIO19_ALT0** — Selecting ALT0 mode of pad GPIO_19 for KEY_COL5. 11 **SD2_CLK_ALT2** — Selecting ALT2 mode of pad SD2_CLK for KEY_COL5.

37.4.556 Select Input Register (IOMUXC_KEY_COL6_SELECT_INPUT)

Address: 20E_0000h base + 8C4h offset = 20E_08C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DAISY	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_KEY_COL6_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **CSI0_DATA06_ALT3** — Selecting ALT3 mode of pad CSI0_DAT6 for KEY_COL6. 01 **GPIO09_ALT2** — Selecting ALT2 mode of pad GPIO_9 for KEY_COL6. 10 **SD2_DATA3_ALT2** — Selecting ALT2 mode of pad SD2_DAT3 for KEY_COL6.

37.4.557 Select Input Register (IOMUXC_KEY_COL7_SELECT_INPUT)

Address: 20E_0000h base + 8C8h offset = 20E_08C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DAISY	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_KEY_COL7_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **CSI0_DATA08_ALT3** — Selecting ALT3 mode of pad CSI0_DAT8 for KEY_COL7. 01 **GPIO04_ALT2** — Selecting ALT2 mode of pad GPIO_4 for KEY_COL7. 10 **SD2_DATA1_ALT4** — Selecting ALT4 mode of pad SD2_DAT1 for KEY_COL7.

37.4.558 Select Input Register (IOMUXC_KEY_ROW5_SELECT_INPUT)

Address: 20E_0000h base + 8CCh offset = 20E_08CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DAISY	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_KEY_ROW5_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IOMUXC_KEY_ROW5_SELECT_INPUT field descriptions (continued)

Field	Description
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **CSI0_DATA05_ALT3** — Selecting ALT3 mode of pad CSI0_DAT5 for KEY_ROW5. 01 **GPIO01_ALT2** — Selecting ALT2 mode of pad GPIO_1 for KEY_ROW5. 10 **SD2_CMD_ALT2** — Selecting ALT2 mode of pad SD2_CMD for KEY_ROW5.

37.4.559 Select Input Register (IOMUXC_KEY_ROW6_SELECT_INPUT)

Address: 20E_0000h base + 8D0h offset = 20E_08D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

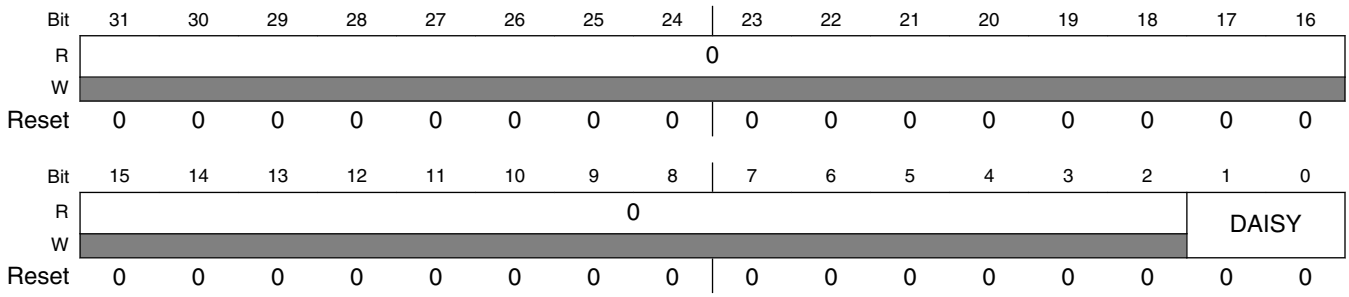
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DAISY	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_KEY_ROW6_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **CSI0_DATA07_ALT3** — Selecting ALT3 mode of pad CSI0_DAT7 for KEY_ROW6. 01 **GPIO02_ALT2** — Selecting ALT2 mode of pad GPIO_2 for KEY_ROW6. 10 **SD2_DATA2_ALT4** — Selecting ALT4 mode of pad SD2_DAT2 for KEY_ROW6.

37.4.560 Select Input Register (IOMUXC_KEY_ROW7_SELECT_INPUT)

Address: 20E_0000h base + 8D4h offset = 20E_08D4h

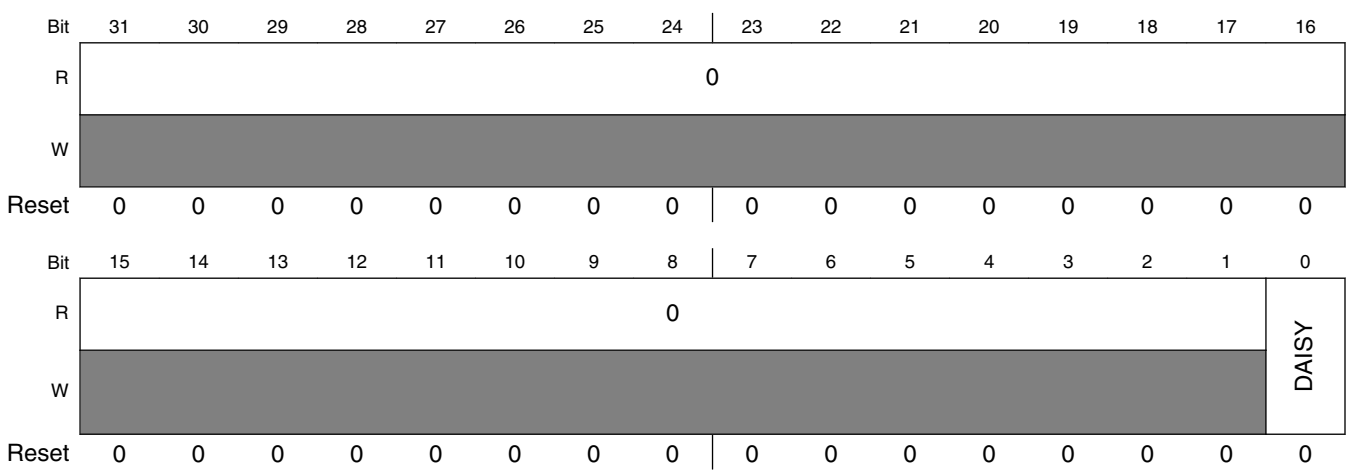


IOMUXC_KEY_ROW7_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **CSI0_DATA09_ALT3** — Selecting ALT3 mode of pad CSI0_DAT9 for KEY_ROW7. 01 **GPIO05_ALT2** — Selecting ALT2 mode of pad GPIO_5 for KEY_ROW7. 10 **SD2_DATA0_ALT4** — Selecting ALT4 mode of pad SD2_DAT0 for KEY_ROW7.

37.4.561 Select Input Register (IOMUXC_MLB_MLB_CLK_IN_SELECT_INPUT)

Address: 20E_0000h base + 8DCh offset = 20E_08DCh

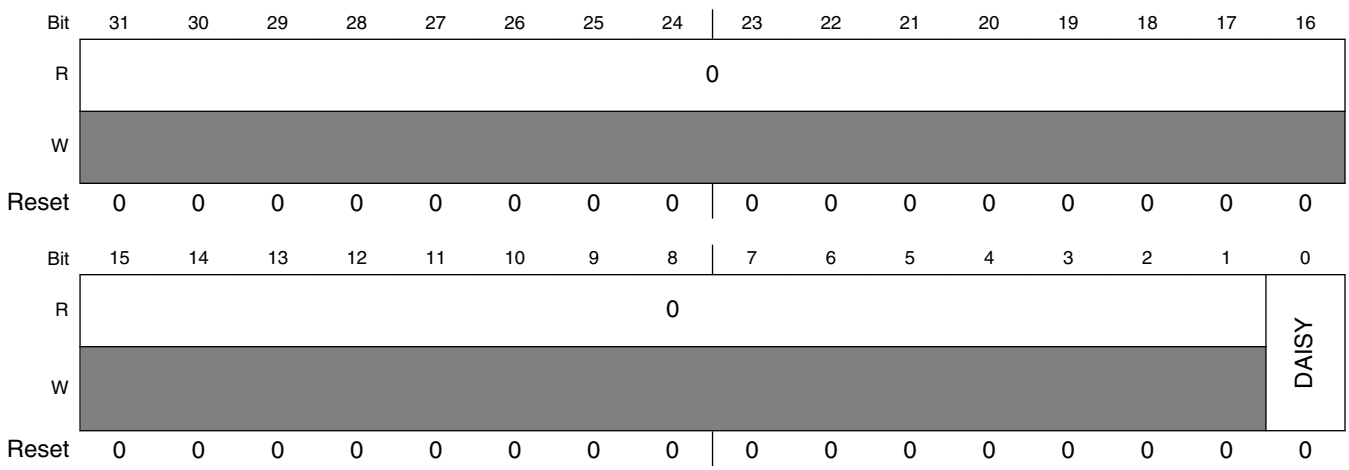


IOMUXC_MLB_MLB_CLK_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_TX_DATA1_ALT0** — Selecting ALT0 mode of pad ENET_TXD1 for MLB_CLK. 1 **GPIO03_ALT7** — Selecting ALT7 mode of pad GPIO_3 for MLB_CLK.

37.4.562 Select Input Register (IOMUXC_MLB_MLB_DATA_IN_SELECT_INPUT)

Address: 20E_0000h base + 8E0h offset = 20E_08E0h



IOMUXC_MLB_MLB_DATA_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_MDC_ALT0** — Selecting ALT0 mode of pad ENET_MDC for MLB_DATA. 1 **GPIO02_ALT7** — Selecting ALT7 mode of pad GPIO_2 for MLB_DATA.

37.4.563 Select Input Register (IOMUXC_MLB_MLB_SIG_IN_SELECT_INPUT)

Address: 20E_0000h base + 8E4h offset = 20E_08E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

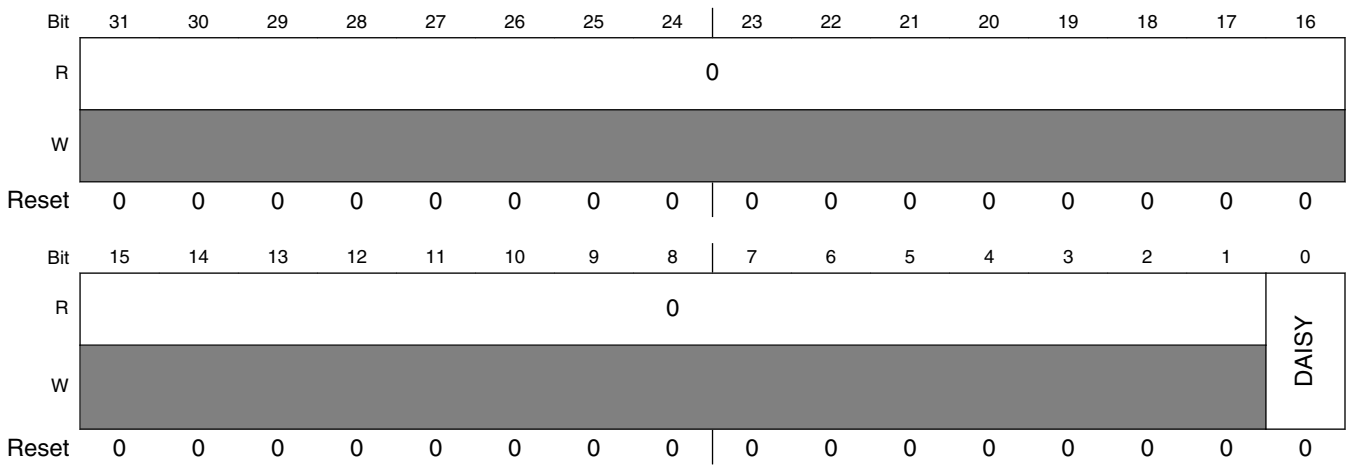
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_MLB_MLB_SIG_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_RX_DATA1_ALT0** — Selecting ALT0 mode of pad ENET_RXD1 for MLB_SIG. 1 **GPIO06_ALT7** — Selecting ALT7 mode of pad GPIO_6 for MLB_SIG.

37.4.564 Select Input Register (IOMUXC_SDMA_EVENTS14_SELECT_INPUT)

Address: 20E_0000h base + 8E8h offset = 20E_08E8h

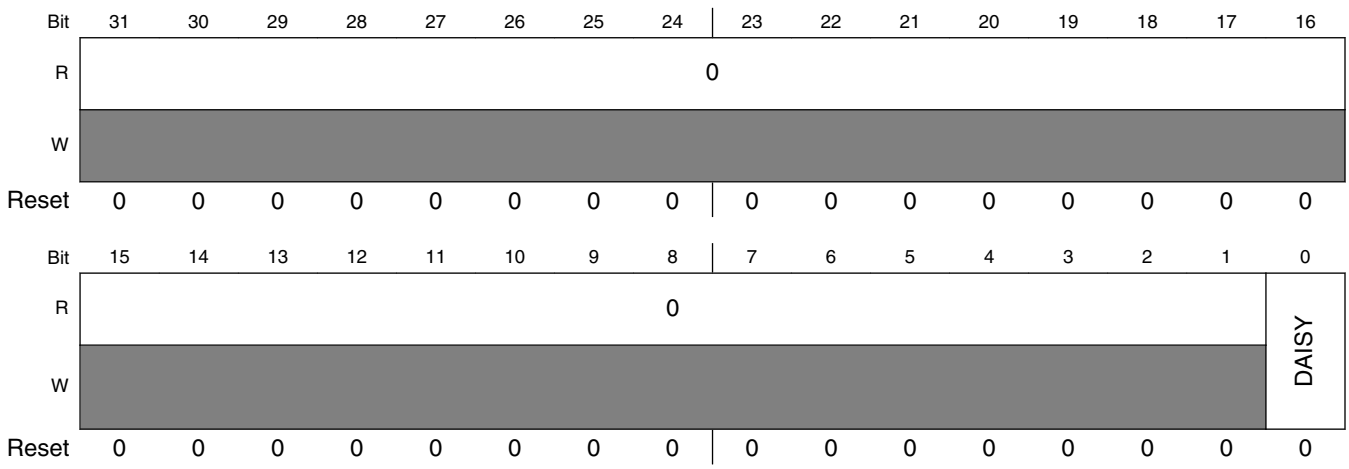


IOMUXC_SDMA_EVENTS14_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DISP0_DATA16_ALT4** — Selecting ALT4 mode of pad DISP0_DAT16 for SDMA_EXT_EVENT0. 1 **GPIO17_ALT3** — Selecting ALT3 mode of pad GPIO_17 for SDMA_EXT_EVENT0.

37.4.565 Select Input Register (IOMUXC_SDMA_EVENTS15_SELECT_INPUT)

Address: 20E_0000h base + 8ECh offset = 20E_08ECh

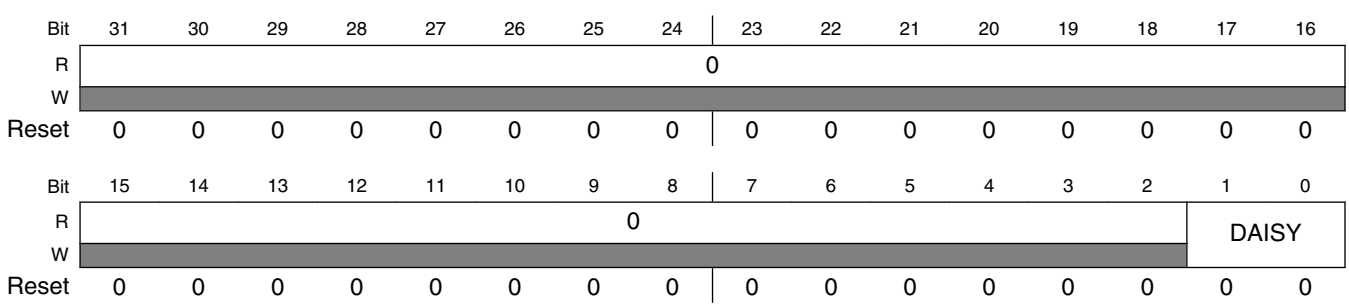


IOMUXC_SDMA_EVENTS15_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DISP0_DATA17_ALT4** — Selecting ALT4 mode of pad DISP0_DAT17 for SDMA_EXT_EVENT1. 1 **GPIO18_ALT3** — Selecting ALT3 mode of pad GPIO_18 for SDMA_EXT_EVENT1.

37.4.566 Select Input Register (IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT)

Address: 20E_0000h base + 8F0h offset = 20E_08F0h

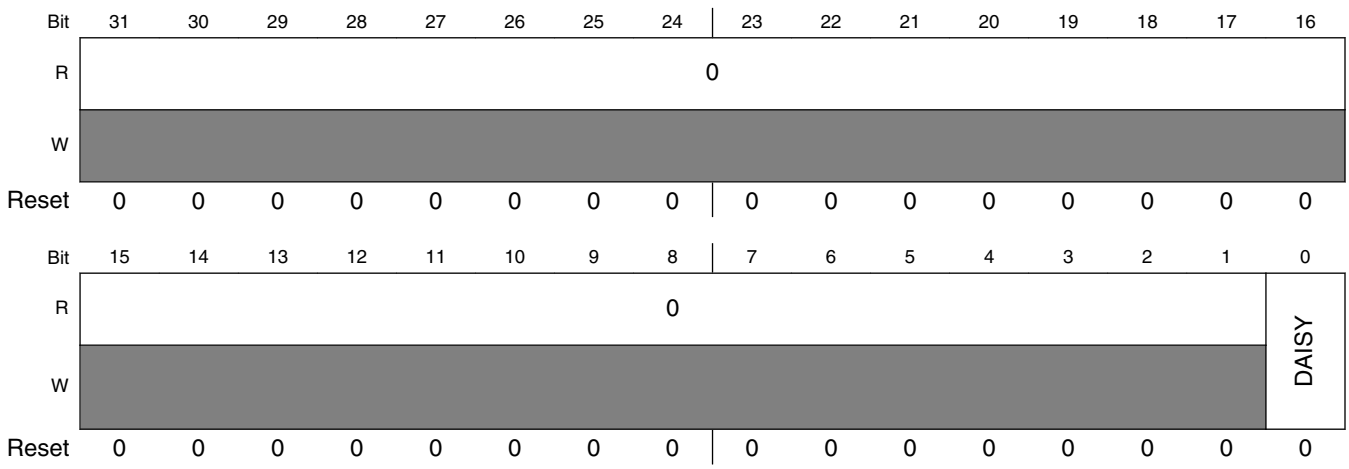


IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 00 **EIM_DATA21_ALT7** — Selecting ALT7 mode of pad EIM_D21 for SPDIF_IN. 01 **ENET_RX_ER_ALT3** — Selecting ALT3 mode of pad ENET_RX_ER for SPDIF_IN. 10 **GPIO16_ALT4** — Selecting ALT4 mode of pad GPIO_16 for SPDIF_IN. 11 **KEY_COL3_ALT6** — Selecting ALT6 mode of pad KEY_COL3 for SPDIF_IN.

37.4.567 Select Input Register (IOMUXC_SPDIF_TX_CLK2_SELECT_INPUT)

Address: 20E_0000h base + 8F4h offset = 20E_08F4h



IOMUXC_SPDIF_TX_CLK2_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **ENET_CRS_DV_ALT3** — Selecting ALT3 mode of pad ENET_CRS_DV for SPDIF_EXT_CLK. 1 **RGMII_TXC_ALT2** — Selecting ALT2 mode of pad RGMII_TXC for SPDIF_EXT_CLK.

37.4.568 Select Input Register (IOMUXC_UART1_UART_RTS_B_SELECT_INPUT)

Address: 20E_0000h base + 8F8h offset = 20E_08F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DAISY	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_UART1_UART_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. Route RTS_B signal to pad when UART is in DCE mode. Route CTS_B signal to pad when UART is in DTE mode. 00 **EIM_DATA19_ALT4** — Selecting ALT4 mode of pad EIM_D19 for UART1_CTS_B. 01 **EIM_DATA20_ALT4** — Selecting ALT4 mode of pad EIM_D20 for UART1_RTS_B. 10 **SD3_DATA0_ALT1** — Selecting ALT1 mode of pad SD3_DAT0 for UART1_CTS_B. 11 **SD3_DATA1_ALT1** — Selecting ALT1 mode of pad SD3_DAT1 for UART1_RTS_B.

37.4.569 Select Input Register (IOMUXC_UART1_UART_RX_DATA_SELECT_INPUT)

Address: 20E_0000h base + 8FCh offset = 20E_08FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DAISY	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_UART1_UART_RX_DATA_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. Route RX_DATA signal to pad when UART is in DCE mode. Route TX_DATA signal to pad when UART is in DTE mode. 00 **CSI0_DATA10_ALT3** — Selecting ALT3 mode of pad CSI0_DAT10 for UART1_TX_DATA. 01 **CSI0_DATA11_ALT3** — Selecting ALT3 mode of pad CSI0_DAT11 for UART1_RX_DATA. 10 **SD3_DATA6_ALT1** — Selecting ALT1 mode of pad SD3_DAT6 for UART1_RX_DATA. 11 **SD3_DATA7_ALT1** — Selecting ALT1 mode of pad SD3_DAT7 for UART1_TX_DATA.

37.4.570 Select Input Register (IOMUXC_UART2_UART_RTS_B_SELECT_INPUT)

Address: 20E_0000h base + 900h offset = 20E_0900h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																													DAISY		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_UART2_UART_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. Route RTS_B signal to pad when UART is in DCE mode. Route CTS_B signal to pad when UART is in DTE mode. 000 **EIM_DATA28_ALT4** — Selecting ALT4 mode of pad EIM_D28 for UART2_CTS_B. 001 **EIM_DATA29_ALT4** — Selecting ALT4 mode of pad EIM_D29 for UART2_RTS_B. 010 **SD3_CLK_ALT1** — Selecting ALT1 mode of pad SD3_CLK for UART2_RTS_B. 011 **SD3_CMD_ALT1** — Selecting ALT1 mode of pad SD3_CMD for UART2_CTS_B. 100 **SD4_DATA5_ALT2** — Selecting ALT2 mode of pad SD4_DAT5 for UART2_RTS_B. 101 **SD4_DATA6_ALT2** — Selecting ALT2 mode of pad SD4_DAT6 for UART2_CTS_B.

37.4.571 Select Input Register (IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT)

Address: 20E_0000h base + 904h offset = 20E_0904h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																													DAISY		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_UART2_UART_RX_DATA_SELECT_INPUT field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. Route RX_DATA signal to pad when UART is in DCE mode. Route TX_DATA signal to pad when UART is in DTE mode. 000 **EIM_DATA26_ALT4** — Selecting ALT4 mode of pad EIM_D26 for UART2_TX_DATA. 001 **EIM_DATA27_ALT4** — Selecting ALT4 mode of pad EIM_D27 for UART2_RX_DATA. 010 **GPIO07_ALT4** — Selecting ALT4 mode of pad GPIO_7 for UART2_TX_DATA. 011 **GPIO08_ALT4** — Selecting ALT4 mode of pad GPIO_8 for UART2_RX_DATA. 100 **SD3_DATA4_ALT1** — Selecting ALT1 mode of pad SD3_DAT4 for UART2_RX_DATA. 101 **SD3_DATA5_ALT1** — Selecting ALT1 mode of pad SD3_DAT5 for UART2_TX_DATA. 110 **SD4_DATA4_ALT2** — Selecting ALT2 mode of pad SD4_DAT4 for UART2_RX_DATA. 111 **SD4_DATA7_ALT2** — Selecting ALT2 mode of pad SD4_DAT7 for UART2_TX_DATA.

37.4.572 Select Input Register (IOMUXC_UART3_UART_RTS_B_SELECT_INPUT)

Address: 20E_0000h base + 908h offset = 20E_0908h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																													DAISY		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_UART3_UART_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field

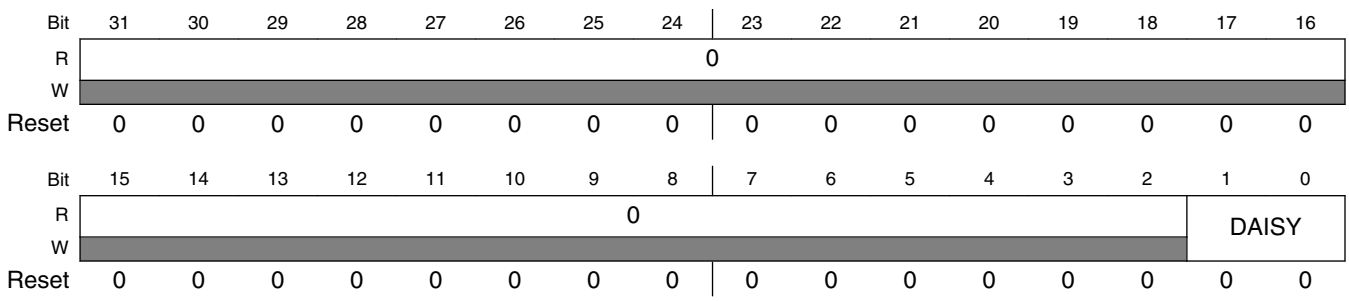
Table continues on the next page...

IOMUXC_UART3_UART_RTS_B_SELECT_INPUT field descriptions (continued)

Field	Description
	Selecting Pads Involved in Daisy Chain. Route RTS_B signal to pad when UART is in DCE mode. Route CTS_B signal to pad when UART is in DTE mode. 000 **EIM_DATA23_ALT2** — Selecting ALT2 mode of pad EIM_D23 for UART3_CTS_B. 001 **EIM_DATA30_ALT4** — Selecting ALT4 mode of pad EIM_D30 for UART3_CTS_B. 010 **EIM_DATA31_ALT4** — Selecting ALT4 mode of pad EIM_D31 for UART3_RTS_B. 011 **EIM_EB3_B_ALT2** — Selecting ALT2 mode of pad EIM_EB3 for UART3_RTS_B. 100 **SD3_DATA3_ALT1** — Selecting ALT1 mode of pad SD3_DAT3 for UART3_CTS_B. 101 **SD3_RESET_ALT1** — Selecting ALT1 mode of pad SD3_RST for UART3_RTS_B.

37.4.573 Select Input Register (IOMUXC_UART3_UART_RX_DATA_SELECT_INPUT)

Address: 20E_0000h base + 90Ch offset = 20E_090Ch

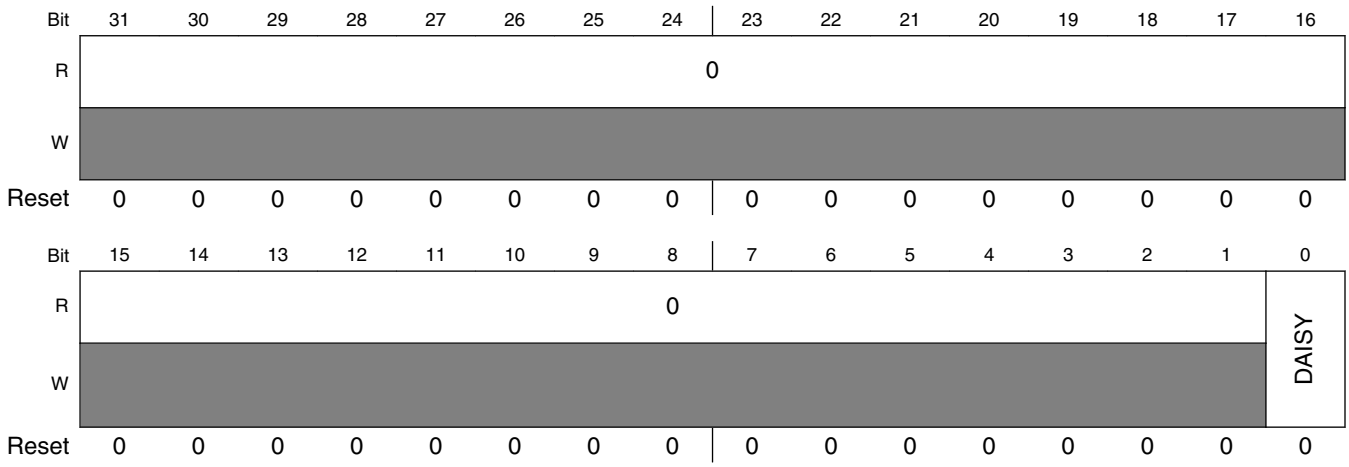


IOMUXC_UART3_UART_RX_DATA_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. Route RX_DATA signal to pad when UART is in DCE mode. Route TX_DATA signal to pad when UART is in DTE mode. 00 **EIM_DATA24_ALT2** — Selecting ALT2 mode of pad EIM_D24 for UART3_TX_DATA. 01 **EIM_DATA25_ALT2** — Selecting ALT2 mode of pad EIM_D25 for UART3_RX_DATA. 10 **SD4_CLK_ALT2** — Selecting ALT2 mode of pad SD4_CLK for UART3_RX_DATA. 11 **SD4_CMD_ALT2** — Selecting ALT2 mode of pad SD4_CMD for UART3_TX_DATA.

37.4.574 Select Input Register (IOMUXC_UART4_UART_RTS_B_SELECT_INPUT)

Address: 20E_0000h base + 910h offset = 20E_0910h

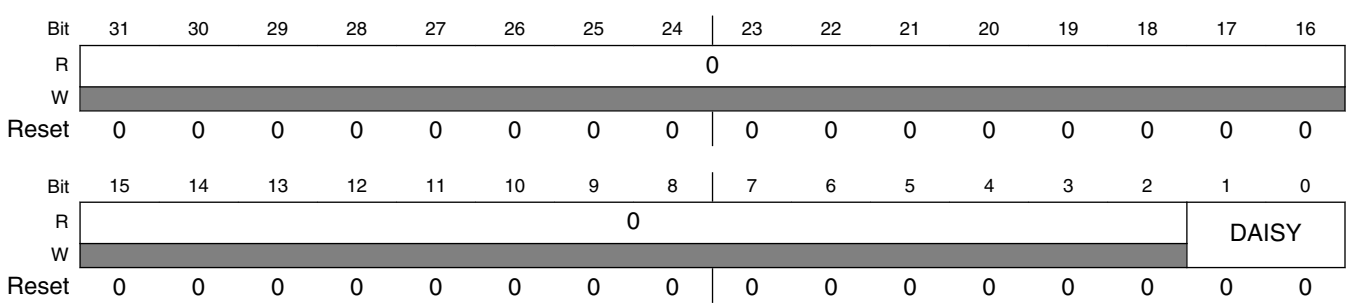


IOMUXC_UART4_UART_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. Route RTS_B signal to pad when UART is in DCE mode. Route CTS_B signal to pad when UART is in DTE mode. 0 **CSI0_DATA16_ALT3** — Selecting ALT3 mode of pad CSI0_DAT16 for UART4_RTS_B. 1 **CSI0_DATA17_ALT3** — Selecting ALT3 mode of pad CSI0_DAT17 for UART4_CTS_B.

37.4.575 Select Input Register (IOMUXC_UART4_UART_RX_DATA_SELECT_INPUT)

Address: 20E_0000h base + 914h offset = 20E_0914h



IOMUXC_UART4_UART_RX_DATA_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. Route RX_DATA signal to pad when UART is in DCE mode. Route TX_DATA signal to pad when UART is in DTE mode. 00 **CSI0_DATA12_ALT3** — Selecting ALT3 mode of pad CSI0_DAT12 for UART4_TX_DATA. 01 **CSI0_DATA13_ALT3** — Selecting ALT3 mode of pad CSI0_DAT13 for UART4_RX_DATA. 10 **KEY_COL0_ALT4** — Selecting ALT4 mode of pad KEY_COL0 for UART4_TX_DATA. 11 **KEY_ROW0_ALT4** — Selecting ALT4 mode of pad KEY_ROW0 for UART4_RX_DATA.

37.4.576 Select Input Register (IOMUXC_UART5_UART_RTS_B_SELECT_INPUT)

Address: 20E_0000h base + 918h offset = 20E_0918h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

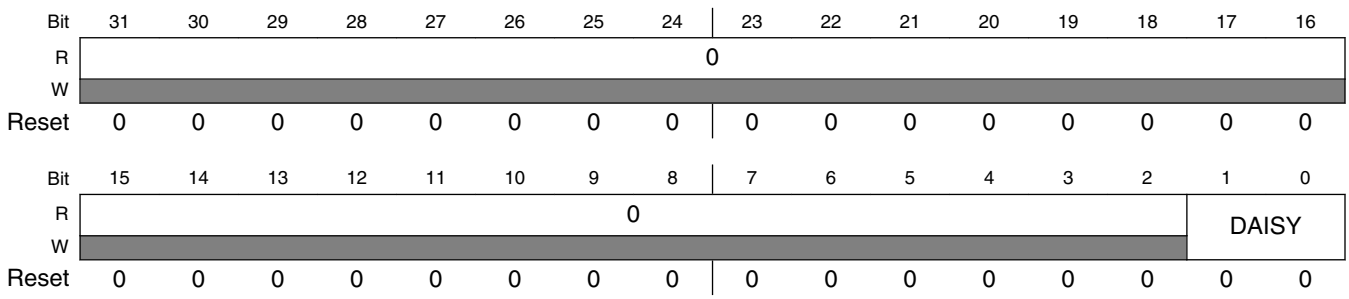
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DAISY	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_UART5_UART_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. Route RTS_B signal to pad when UART is in DCE mode. Route CTS_B signal to pad when UART is in DTE mode. 00 **CSI0_DATA18_ALT3** — Selecting ALT3 mode of pad CSI0_DAT18 for UART5_RTS_B. 01 **CSI0_DATA19_ALT3** — Selecting ALT3 mode of pad CSI0_DAT19 for UART5_CTS_B. 10 **KEY_COL4_ALT4** — Selecting ALT4 mode of pad KEY_COL4 for UART5_RTS_B. 11 **KEY_ROW4_ALT4** — Selecting ALT4 mode of pad KEY_ROW4 for UART5_CTS_B.

37.4.577 Select Input Register (IOMUXC_UART5_UART_RX_DATA_SELECT_INPUT)

Address: 20E_0000h base + 91Ch offset = 20E_091Ch

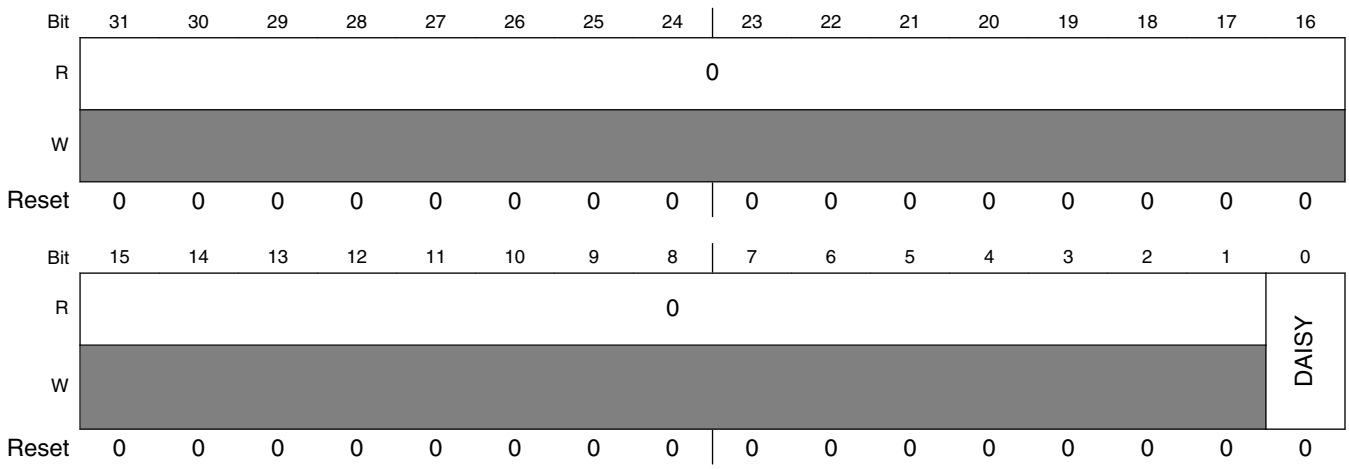


IOMUXC_UART5_UART_RX_DATA_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. Route RX_DATA signal to pad when UART is in DCE mode. Route TX_DATA signal to pad when UART is in DTE mode. 00 **CSI0_DATA14_ALT3** — Selecting ALT3 mode of pad CSI0_DAT14 for UART5_TX_DATA. 01 **CSI0_DATA15_ALT3** — Selecting ALT3 mode of pad CSI0_DAT15 for UART5_RX_DATA. 10 **KEY_COL1_ALT4** — Selecting ALT4 mode of pad KEY_COL1 for UART5_TX_DATA. 11 **KEY_ROW1_ALT4** — Selecting ALT4 mode of pad KEY_ROW1 for UART5_RX_DATA.

37.4.578 Select Input Register (IOMUXC_USB_OTG_OC_SELECT_INPUT)

Address: 20E_0000h base + 920h offset = 20E_0920h

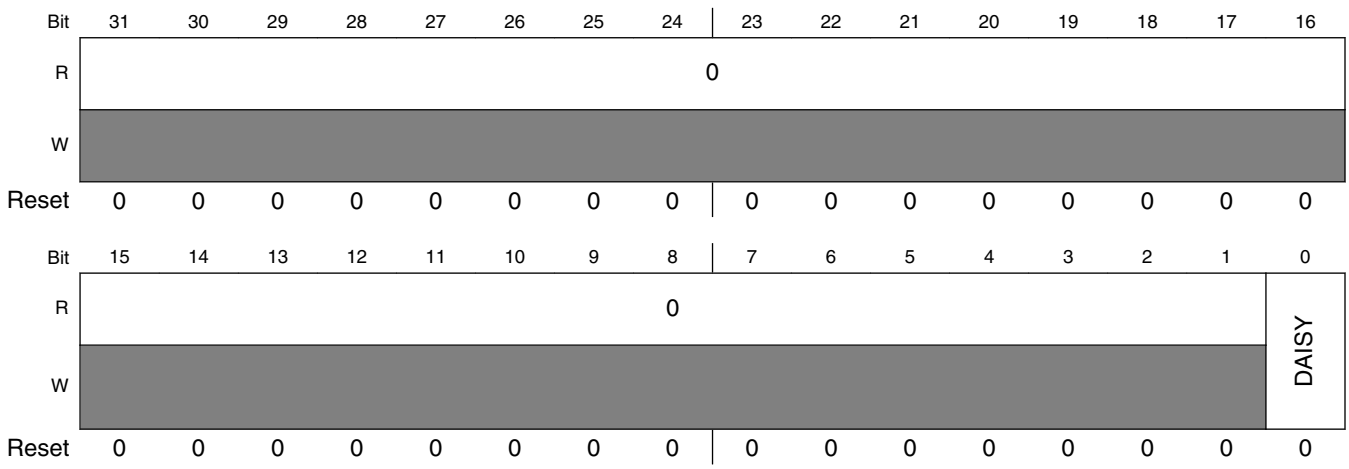


IOMUXC_USB_OTG_OC_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_DATA21_ALT4** — Selecting ALT4 mode of pad EIM_D21 for USB_OTG_OC. 1 **KEY_COL4_ALT2** — Selecting ALT2 mode of pad KEY_COL4 for USB_OTG_OC.

37.4.579 Select Input Register (IOMUXC_USB_H1_OC_SELECT_INPUT)

Address: 20E_0000h base + 924h offset = 20E_0924h

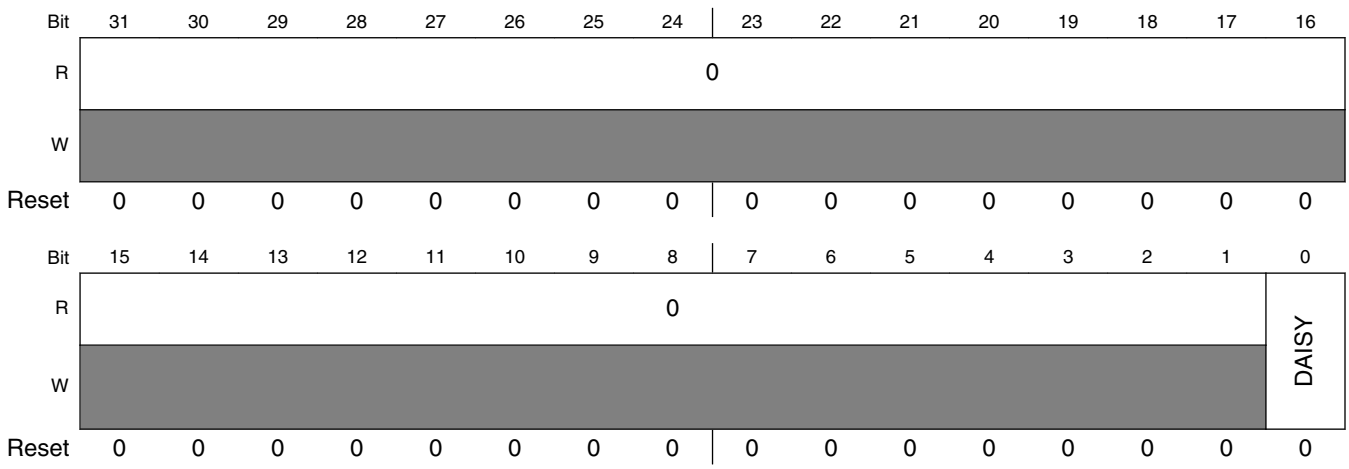


IOMUXC_USB_H1_OC_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **EIM_DATA30_ALT6** — Selecting ALT6 mode of pad EIM_D30 for USB_H1_OC. 1 **GPIO03_ALT6** — Selecting ALT6 mode of pad GPIO_3 for USB_H1_OC.

37.4.580 Select Input Register (IOMUXC_USDHC1_CARD_CLK_IN_SELECT_INPUT)

Address: 20E_0000h base + 928h offset = 20E_0928h



IOMUXC_USDHC1_CARD_CLK_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **RESERVED0** — This field value is reserved. 1 **SD1_CLK_ALT0** — Selecting ALT0 mode of pad SD1_CLK for SD1_CLK.

37.4.581 Select Input Register (IOMUXC_USDHC1_WP_ON_SELECT_INPUT)

Address: 20E_0000h base + 92Ch offset = 20E_092Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

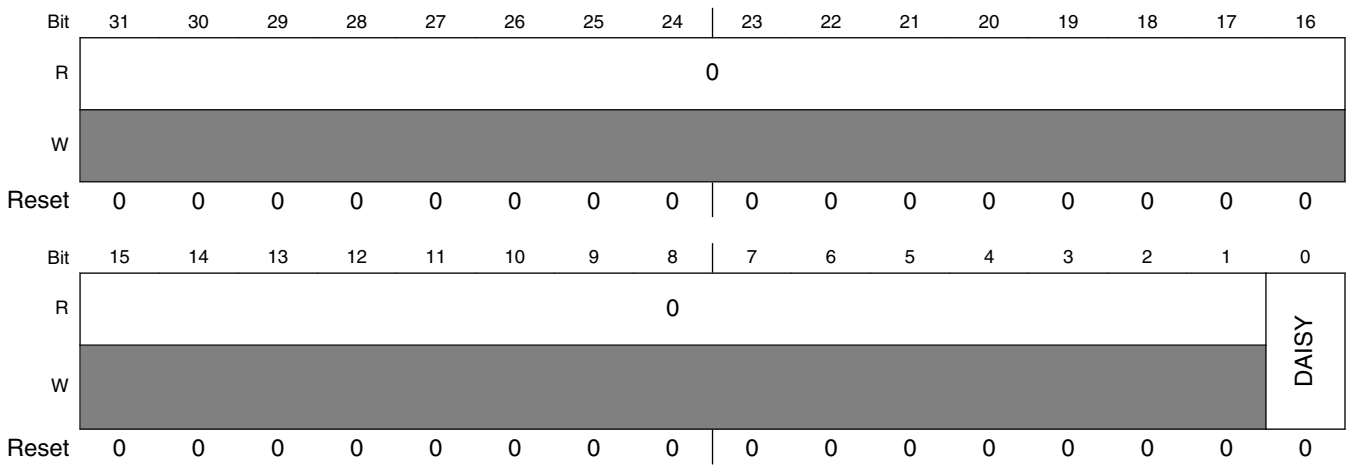
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_USDHC1_WP_ON_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **DI0_PIN04_ALT3** — Selecting ALT3 mode of pad DI0_PIN4 for SD1_WP. 1 **GPIO09_ALT6** — Selecting ALT6 mode of pad GPIO_9 for SD1_WP.

37.4.582 Select Input Register (IOMUXC_USDHC2_CARD_CLK_IN_SELECT_INPUT)

Address: 20E_0000h base + 930h offset = 20E_0930h



IOMUXC_USDHC2_CARD_CLK_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **RESERVED0** — This field value is reserved. 1 **SD2_CLK_ALT0** — Selecting ALT0 mode of pad SD2_CLK for SD2_CLK.

37.4.583 Select Input Register (IOMUXC_USDHC3_CARD_CLK_IN_SELECT_INPUT)

Address: 20E_0000h base + 934h offset = 20E_0934h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

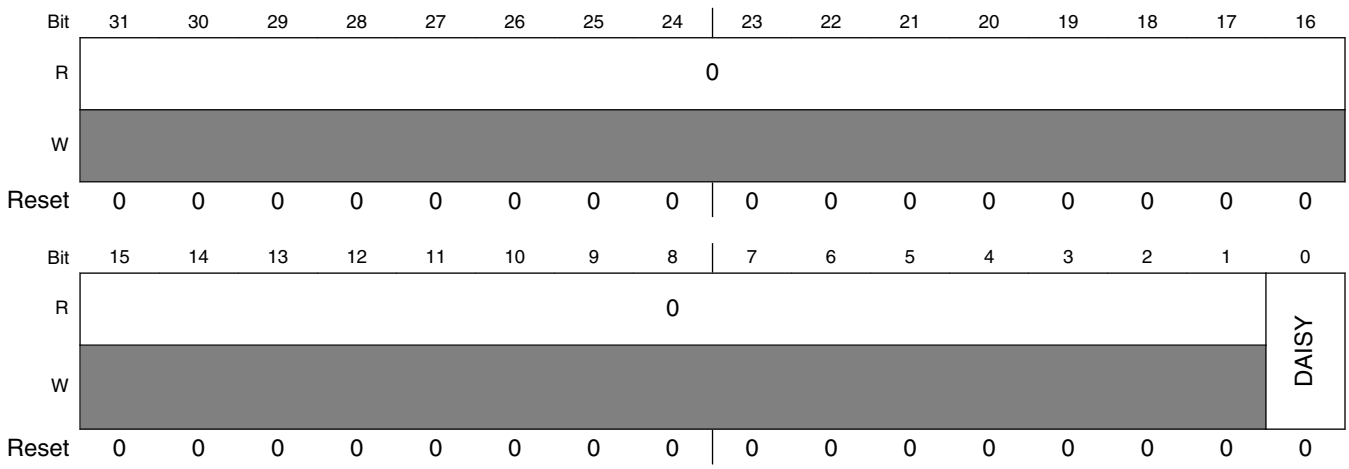
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_USDHC3_CARD_CLK_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **RESERVED0** — This field value is reserved. 1 **SD3_CLK_ALT0** — Selecting ALT0 mode of pad SD3_CLK for SD3_CLK.

37.4.584 Select Input Register (IOMUXC_USDHC4_CARD_CLK_IN_SELECT_INPUT)

Address: 20E_0000h base + 938h offset = 20E_0938h



IOMUXC_USDHC4_CARD_CLK_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain. 0 **RESERVED0** — This field value is reserved. 1 **SD4_CLK_ALT0** — Selecting ALT0 mode of pad SD4_CLK for SD4_CLK.

Chapter 38
Image Processing Unit (IPU)

38.1 Overview

The IPU is planned to be a part of the video and graphics subsystem in an application processor.

The goal of the IPU is to provide comprehensive support for the flow of data from an image sensor and/or to a display device. This support covers all aspects of these activities:

- Connectivity to relevant devices - cameras, displays, graphics accelerators, TV encoders and decoders.
- Related image processing and manipulation: sensor image signal processing, display processing, image conversions, etc.
- Synchronization and control capabilities (to avoid tearing artifacts).

This integrative approach leads to several significant advantages:

- Automation: The involvement of the ARM platform in image management is minimized. In particular, display refresh/update and a camera preview (displaying the input from an image sensor) can be performed completely autonomously. The resulting benefits are reducing the overhead due to SW-HW synchronization, freeing the ARM platform to perform other tasks and reduced power consumption (when the ARM core is idle and can be powered down).
- Optimal data path: Access to system memory is minimized. In particular, significant processing can be performed on-the-fly while receiving data from an image sensor and/or sending data to a display. System memory is used essentially only when a change in pixel order or frame rate is needed. The resulting benefits are reduced load on the system bus and further reduction of power consumption.
- Resource sharing: Maximal HW reuse for different applications, resulting with the support of a wide range of requirements with minimal HW.

The HW reuse mentioned above is enabled by a sophisticated configurability of each HW block. This configurability also allows the support of a wide range of external devices, data formats and operation modes. The resulting flexibility is important also because the support requirements are evolving significantly, so expected future changes need to be anticipated and accounted for.

The following further principles guided the choice of support provided by the IPU:

- For key applications that deserve and need HW support (for acceleration or low power), provide the best support (leading to an optimal implementation).
- For additional applications that can benefit from the HW, consider cost vs. benefit of making minor modifications/extensions to support them.
- For all other relevant applications (to be supported by SW), verify that their support is not degraded.
- Whenever possible, let the operating system (and its windowing system) act as it would without the IPU.

38.1.1 Architecture

A simplified block diagram of the IPU can be found here. The role of each block is described in IPU.

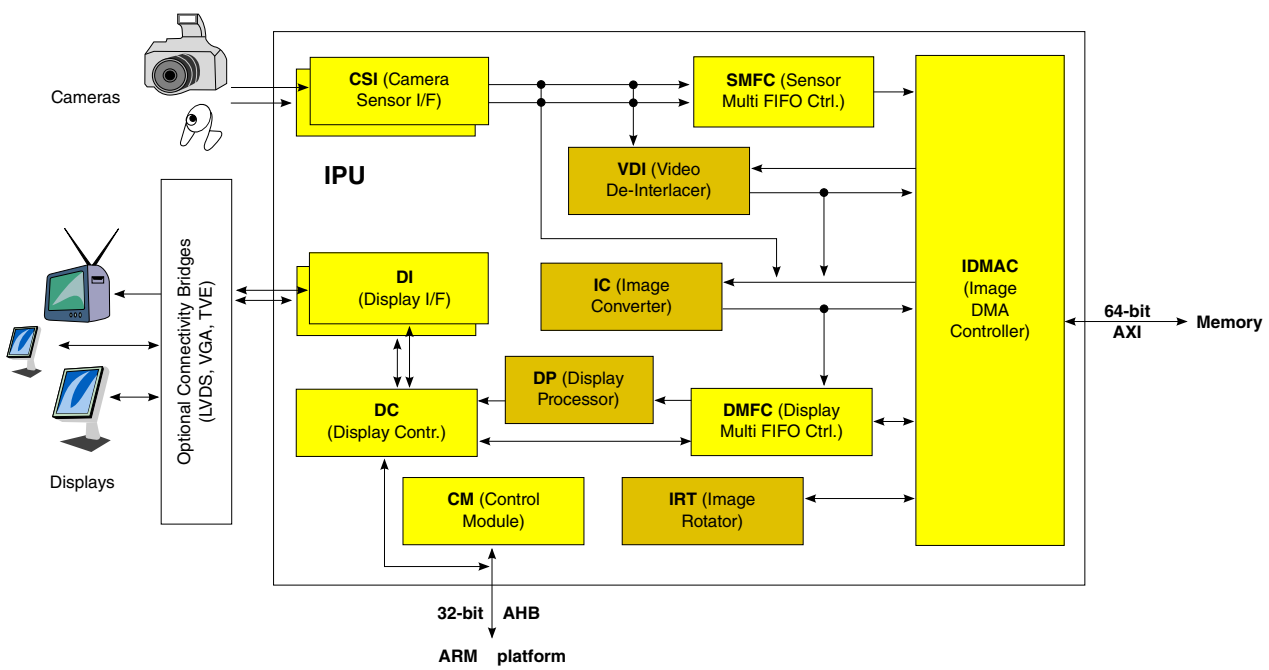


Figure 38-1. IPU Block Diagram

Table 38-1. IPU - Block Description

Block	Description
CSI - Camera Sensor Interface	Controls a camera port; provides interface to an image sensor or a related device. IPU includes 2 such blocks.
DI - Display Interface	Provides interface to displays, display controllers and related devices. IPU includes 2 such blocks.
DC - Display Controller	Controls the display ports.
DP - Display Processor	Performs the processing required for data sent to display.
IC - Image Converter	Performs resizing, color conversion/correction, combining with graphics, and horizontal inversion.
VDIC - Video De Interlacer	Performs video de interlacing (Interlaced -> progressive) or combining.
IRT - Image Rotator	Performs rotation (90 or 180 degrees) and inversion (vertical/horizontal).
IDMAC - Image DMA Controller	Controls the memory port; transfers data to/from system memory.
DMFC - Display Multi FIFO Controller	Controls FIFOs for IDMAC channels related to the display system.
CM - Control Module	Provides control and synchronization.

38.1.2 Features And Functionality

38.1.2.1 External Ports

IPU has the following ports:

- Two camera ports - each controlled by a CSI sub-block, providing a connection to image sensors and related devices.
- Two display ports - each controlled by a DI sub-block, providing a connection to displays and related devices.
- Memory port - AXI (AHB V3.0) master, controlled by the IDMAC - providing connection to the system memory.
- AHB-lite slave port, providing connection to the ARM Platform (and to any other master connected to the ARM's cross-bar switch).
- Additional ports for control and debug.

38.1.2.1.1 Camera Ports

The role of these ports is to receive input from image sensors (or TV decoders) and to provide support for time-sensitive control signals to the camera.

(Non-time-sensitive controls; configuration, reset are performed by the ARM platform through I2C I/F or GPIO signals).

Each of the camera ports includes the following features:

- Direct connectivity to most relevant image sensors and to TV decoders.
- Interface types
  - Parallel interface
    - Up to 20-bit input data bus.
    - A single value in each cycle, except for special cases listed in the table below (comments column).
    - Programmable polarity.
  - High-speed serial interface - MIPI (Mobile Industry Processor Interface) CSI-2 (Camera Serial Interface) (implemented partly in the IPU and partly in the HSC).
    - Up to four data lanes; up to 800 Mbps per lane
    - Class 1 compliancy (supporting all primary formats)
- The data formats
  - Interleaved color components, up to 16 bits per value (component).
  - The supported formats are listed in the table below.

Table 38-2. Data Formats Supported By The Camera Port

Format	Resolution	On-The-Fly Processing	Direct path to memory	Comments
Bayer RGB	8 bits/value	No	8- or 16-bit values	MIPI mandatory format
	9-10 bits/value	No	Written to the MSB of a 16-bit word	10 bits/value is a MIPI mandatory format
	16 bits/value	No		
Full RGB or YUV 4:4:4	444/555 mode	Yes, starting with color extension to 8 bits/ sample	Yes	MIPI optional formats In parallel I/F: through an 8-bit or 16-bit bus
	565 mode			MIPI mandatory format In parallel I/F: through an 8-bit or 16-bit bus
	8 bits/value (888 mode)	Yes	Yes	MIPI mandatory format
	8-16 bits/value	No	8- or 16-bit components are written to the MSB of a 16-bit word 10 bits/ value can also be packed in a 32-bit word	
YUV 4:2:2 Component order: UY1VY2... or Y1UY2V...	8 bits/value	Yes	Yes	MIPI mandatory format (UY1VY2...) In parallel I/F: through an 8-bit bus (such as BT.656) or 16-bit bus (such as BT.1120)

Table continues on the next page...

Table 38-2. Data Formats Supported By The Camera Port (continued)

Format	Resolution	On-The-Fly Processing	Direct path to memory	Comments
	9-10 bits/value	No	Written to the MSB of a 16-bit word	In parallel I/F: through a 10-bit bus (such as BT.656) or 20-bit bus (such as BT.1120)
	16 bits/value	No	Written to the MSB of a 16-bit word	
Gray scale	8 bits/value	No	Yes	
	16 bits/value	No	Written to the MSB of a 16-bit word	
Generic data		No	Yes In a parallel I/F, if wider than 8 bits, each bus word is written to the MSB of a 16-bit word	MIPI mandatory format May be used for any other formant, such as JPEG/MPEG4

- Scan order: progressive or interlaced data (expected only for YUV 4:2:2) is sent directly to system memory, where it can be read back for further processing.
- Frame size: up to 8192 x 4096 pixels
- Synchronization: video mode
  - The sensor is the master of the pixel clock (PIXCLK) & synchronization signals
  - Synchronization signals are received using either of the following methods:
    - Dedicated control signals -VSYNC, HSYNC - with programmable pulse width & polarity
    - Controls embedded in the data stream, following loosely the BT.656 protocol, with flexibility in code values and location.
- Synchronization : still image capture
  - The image capture is triggered by the ARM platform or by an external signal (such as a mechanical shutter).
  - Synchronized strobes are generated for up to 6 outputs - the sensor and camera peripherals (such as flash, mechanical shutter).
- Additional features
  - Frame rate reduction, by the periodic skipping of frames
    - The supported reduction ratios are: m:n, where m,n<=5
    - This is supported independently for the different destinations - IC, SMFC.
  - Window-of-interest selection
  - Pre-flash - for red-eye reduction and for measurements (such as focus) in low-light conditions

Several sensors can be connected to each of the CSIs. Simultaneous functionality (sending data) is supported as follows:

- Two sensors can send data independently, each through a different port, each using either parallel or fast serial interface.
- Several sensors can send data to the same port, using the MIPI interface (through a HUB), each sensor being identified by different ID's.
- Unpacking and companding capabilities are provided for up to two streams (either through same or different interfaces), while the other ones are treated as generic data.
- Only one of the (non-generic) streams can be transferred to theVDI C or IC for on-the-fly processing, while the others are sent directly to system memory.

The input rate supported by the camera port is as follows:

For parallel interface, the maximum speed of the interface is 240Mhz. The required operating frequency of the interface is calculated in the following way:

F = FH * FW * FPS * BI * DF

Where

- **FH** = frame height (in pixels)
- **FW** = Frame width (in pixels)
- **fps** = frame rate (frames per second)
- **BI** = typically 35% overhead, should be assumed as 1.35. The actual blanking intervals are a parameter of the attached device.
- **DF** = data format, defines the number of cycles needed to send a single pixel.

The number of cycles needed to send a single pixel depends on the interface and the data format.

Data format examples:

- YUV422 over 16 bit = 1 cycle/pixel
- RGB888 over 8 bit = 3 cycles/pixel
- RGB565 over 16 bit = 1 cycle/pixel
- Bayer/Generic data = 1 cycle/pixel
- YUV422 over 8 bit = 2 cycles/pixel
- BT.656, YUV422 format = 2 cycles/pixel
- BT.1120, YUV422 format = 1 cycle/pixel

Examples of supported interfaces:

- 3.2MP camera, 15fps, yuv422 format, 8 bit interface
- 1080P30, yuv422, 8 bit interface

Fast serial interface (MIPI-CSI2):

- IPU receives 2 components per cycle from the MIPI-CSI2 interface
- The maximum speed of the interface is:
  - 200Mhz for 4 data lanes configuration
  - 250Mhz for 2 data lanes configuration

The maximum bandwidth of the interface is as follows:

- 200Mhz for 4 data lanes configuration (800Mbps/lane, 400MByte/sec)
- 187.5Mhz for 3 data lanes configuration (1000Mbps/lane, 375MByte/sec)
- 125Mhz for 2 data lanes configuration (1000Mbps/lane, 250MByte/sec)
- 62.5Mhz for 1 data lane configuration (1000Mbps/lane, 125Mbyte/sec)

The required operating frequency of the interface is calculated in the same way as for parallel interface above. The DF parameter is different.

- YUV422 = 1 cycle/pixel
- RGB888 = 1.5 cycles/pixel
- Generic data = 2 bytes/pixel

Examples of supported interfaces:

- 3.2MP camera, 2 lanes configuration, 15fps, yuv422 format (~65Mhz)
- 6MP camera, 4 lanes configuration, 15fps, RGB888 format (~182Mhz)

38.1.2.1.2 Display Ports

The role of these ports is to communicate with display devices, either directly or through a controller (such as a graphics accelerator) or a bridge (such as a TV encoder or an LVDS interface bridge).

38.1.2.1.2.1 Access Modes

Two access modes are supported.

38.1.2.1.2.1.1 Synchronous Access

In this mode, the IPU transfers a two-dimensional block of pixels to the display device, in synchronization with the screen refresh cycle.

It is called "video mode" in the MIPI standards.

This mode has a dual role:

- For a RAM-less display or a TV screen, this mode is used to perform the screen refresh process from a display buffer in system memory.
- For a "smart" display, this mode is used to transfer a rectangular block of pixels to the display's screen and, in some cases, also to the display buffer
  - The transferred block may be only part of the screen (the rest of the screen being refreshed by the integrated controller, from the internal buffer). Moreover, a mask can be used to transfer to the display only parts of the block, such as a window partly hidden by other windows.
  - If the block is transferred only to the screen, the transfer rate must be equal to the refresh rate. If, however, the transfer is also to the display's memory, the rate can be reduced to the rate at which the input buffer is updated.

In all cases (including the last one), the IPU sends to the display all the synchronization signals controlling the screen refresh and the block transfer is synchronized with these signals. This synchronization means that tearing effects are avoided when using this mode.

38.1.2.1.2.1.2 Asynchronous Access

This is the main mode used for communicating with an external display controller (possibly in a smart display or a graphics accelerator).

It is called "command mode" in the MIPI standards. In this mode, the IPU performs random access - read/write - to the memory and registers of the controller.

Two types of addressing methods are supported

- Generic linear addressing of pixels and generic data
- 2-dimensional (X/Y) addressing of pixels

The following access types are provided:

- Data transfer to the external device, after on-the-fly processing in the IPU.
- Data transfer (DMA) - read/write - between the host's system memory and the external device, through the IPU's memory port (controlled by the IDMAC), such as the transfer of a rectangular block of pixels (possibly full screen).
- Host access - read/write - to an external device, through the AHB-slave port
  - Access types

    - Direct access - emulating a directly-addressed access (see below) This includes burst access (incremental; up to 8 words/burst)
    - Low-level access - leaving to the host the explicit generation of the access protocol
  - The possible accessing modules include the ARM platform and the system DMA controller (as well as any other AHB master connected to the ARM's cross-bar switch).

Transfer of video/graphics data stream to controller's display buffer is performed using one of the first two modes above. Unlike in the synchronous mode, this process is not tightly-synchronized with the screen refresh cycle. However, a loose synchronization - to avoid tearing - is still possible: the appropriate timing for the transfer can be derived from the VSYNC signal of the screen refresh - either generated by the IPU's display controller or received from the external controller.

The asynchronous access requires the specification of an address. The display interface uses "indirect addressing", namely, there is no address bus, and the address, as well as control and configuration commands, are embedded in the data stream. The access procedure - including writing addresses and commands - is managed autonomously by the interface, in one of two ways:

- Automatic emulation of transparent access, following microcode ("access template") generated by the ARM Platform. This mechanism is very flexible, supporting a wide variety of devices.
- Streaming commands/addresses from a buffer stored (by the ARM Platform) in system memory.

Note that direct access requires the use of the first method - automatic emulation.

38.1.2.1.2.2 Display Interface

The display interface is very flexible and supports a wide variety of devices from major manufacturers. The following interface types are provided (in each of the two display ports)

- Parallel video interface (for synchronous access) - up to 24-bit data bus.
  - Compatible with MIPI-DPI standard .
  - Control protocol - follows Sharp HR and generic TFT definitions
  - Supports BT.656 (8-bit) and BT.1120 (16-bit) protocols
  - Supports HDTV standards SMPTE274 (1080i/p) and SMPTE296 (720p)
- A parallel bidirectional bus interface (for asynchronous access) - up to 32-bit data bus.
  - Compatible with MIPI-DBI standard.

    - Control protocol - either system-80 or system-68K The timing and polarity of the signals are programmable.
    - Byte-enable - optional, for a 16-bit interface
    - Burst access for direct access, the burst is determined by the corresponding signal in the AHB interface.
- A serial interface - 3-wire, 4-wire and 5-wire (two flavors) (for asynchronous access)
- High-speed serial interface: MIPI (Mobile Industry Processor Interface) - DSI (Display Serial Interface) full support, with up to 2 data lanes, up to 4 virtual channels (implemented party in the IPU and partly in the HSC).

The supported formats for pixel data are

- RGB - color depth fully configurable; up to 8 bits/value (color component)
- YUV 4:2:2, 8 bits/value (for TV encoder)
- All mandatory formats in MIPI's DBI, DPI and DSI.

In the parallel interfaces, the data bus has up to 32 bits. Non-trivial mapping of pixels to the bus is restricted to the 24 LSB's. This mapping is fully configurable and very flexible. In the serial interfaces, the data is mapped in the same way as in the parallel interfaces and then serialized.

The interface also supports "generic data". Such data is transferred - byte-by-byte, without modification - between the system memory and the display device (through a serial interface or 8/16-bit parallel interface). Non-conventional pixel formats can be supported by considering them as "generic data".

For the interface clock, there are the following options (independently for each port)

- Derived from the IPU internal clock (master mode)
- Provided by an external source (slave mode)

The transfer rate supported:

- For single port (for on-chip interfaces):
    - 240 Mega-Accesses/Sec if the DI clock is derived from an external to IPU source (like another PLL)
    - 264 Mega-Accesses/Sec if the DI clock is derived from the IPU clock (HSP_CLK)
    - When off-chip interfaces are involved the rate may be limited by IO capabilities. Please refer to the device's data-sheet for exact numbers.

For synchronous access with one cycle/pixel, this enables, e.g (including 35% blanking intervals)

- 1080p (1920x1080) @ 60 fps

- WSXGA+ (1680x1050) @ 60 fps
- The combined rate for the two ports is up to 240 MP/sec

The interface includes the following additional features:

- Screen size: up to 4096 x 2048 pixels, programmable by software.
- Scan Order: progressive or interlaced
- Synchronization
  - Programmable horizontal and vertical synchronization output signals (for synchronous access)
  - Data enabling output signal
- Software contrast control using 8-bit programmable pulse-width modulation (PWM) Two dedicated PWM outputs are provided

Connecting To Display Devices

IPU allows the connectivity to multiple display devices. In particular, it supports the following setup:

- Primary LCD display; can be smart, dumb (RAM-less) or dual-port; may use fast serial, or the parallel interface or (through an integrated bridge) LVDS interface.
- Second LCD display; can be smart or dumb (RAM-less); may use fast serial, parallel or serial interface or (through an integrated bridge) LVDS interface.

Each of the above connections has independent settings - interface timing, access template, chip-select, etc.

Simultaneous functionality of the above devices is possible in each of the following ways:

- Two devices can be accessed (synchronously or asynchronously) independently, each through a different port: each using any of the available interfaces.
- Two devices can time-share asynchronous accesses through the legacy serial & parallel interfaces, using the CS signals.
- Two devices can be accessed - synchronously or asynchronously - through the same port, using the MIPI interface (through a HUB), each device being identified by different ID's.
- An asynchronous access can be performed during vertical blanking intervals of a synchronous access (screen refresh; to the same or other device).

The possibilities for simultaneous functionality by time-sharing the legacy interfaces in a single port are summarized in the following table.

Table 38-3. Simultaneous Functionality of Display Port By Time-Sharing Legacy Interfaces

	Second/Third Display Type		
Primary Display Type	**Smart Display Serial Interface**	**Smart Display Parallel Interface Asynchronous access**	**RAM-less display or TV screen**
Smart Display Parallel Interface Asynchronous access	Yes	Yes	Yes; access to the smart display is restricted to blanking intervals
Dual Port Smart Display Synchronous access	Yes	Yes, access to the secondary display is restricted to blanking intervals	Not Available
TFT RAM-less Display Or TV screen	Yes	Yes, access to the smart display is restricted to blanking intervals	Not Available
Graphics Accelerator	Yes	Yes, if the accelerator supports a chip select functionality	Not Available

38.1.2.1.3 Memory Port

The memory port is an AXI (AHB V3.0) master port, used to read/write data - typically two-dimensional blocks from/to system memory.

The interface supports the following features

- Clock rate up to 264 MHz (equal to the internal clock)
- 64-bit data bus
- The supported data formats are listed in the table below.

Table 38-4. Data Formats Supported By The Memory Interface

Format	Resolution	Input/Output	Comments
Non-interleaved YUV (in three separate buffers)	8 bits/value	both	4:4:4, 4:2:2, 4:2:0 formats
Partially-interleaved YUV (in two separate buffers)	8 bits/value	both	4:2:2, 4:2:0 formats Y buffer and UV buffer
Interleaved YUV (all color components in a single buffer)	8 bits/value	both	4:4:4 format (YUV...) 4:2:2 format (UY1VY2... or UY2VY1... or Y1UY2V...or Y2UY1V...)
Interleaved true color	8, 12,16, 18, 24, 32 bits/ pixel	both	Flexible component location A: translucency value

Table continues on the next page...

Table 38-4. Data Formats Supported By The Memory Interface (continued)

Format	Resolution	Input/Output	Comments
	0 - 8 bits per R/G/B/A value		(only in input)
Coded color (using a palette)	4,8 bits/pixel	input only	
Gray scale	8 bits/pixel	both	
	4 bits/pixel	input only	Transferred to display port
Generic data (Transparent M)	8 bits/unit	both	E.g.: From CSI to DI Compressed data to/from DP Translucency for combining

- The pixel formats are translated to/from a uniform internal format: RGBA/YUVA 8:8:8:8
- The supported ordering of bytes and pixels is little endian. For 4 bits/pixel, big endian is also supported.
- Addressing modes include:
  - Sequential access (to a contiguous memory buffer) - for generic data.
  - Raster-scan within a two-dimensional window of a video/display buffer - for both pixel and generic data.
  - Raster-scan of two-dimensional blocks within a two-dimensional window (for rotation of pixel data)
- Additional features
  - Scan order: progressive or interlaced Interlaced access is supported for fields which are stored either in separate memory buffers or with rows interleaved in a single buffer.
  - Reordered scan, implementing inversion and rotation.
    - Rotation and horizontal inversion - only when transferring two-dimensional blocks (to/from the IRT)
    - Vertical inversion - also in row-by-row raster-scan
  - Scrolling
    - Applications Panning within a frame Frame scrolling
    - Not supported for non-interleaved and partially-interleaved formats
    - Resolution Vertical: single pixel Horizontal: 18 BPP - 4 pixels; 12, 4 BPP or YUV 422 - 2 pixels; other formats - one pixel
  - Conditional read (for combining): fully-transparent or hidden pixels are not read.
    - This is supported, for graphics. by reading the transparency (alpha) from a separate buffer
  - Input/output FIFOs (in the SMFC, DMFC and in the processing sub-blocks) - size adjusted to provide resilience for latency of up to 1500 cycles.

38.1.2.1.4 Processing

The IPU processes rectangular blocks of pixels. The processing is performed in these sub-blocks - VDIC, DP, IC and IRT.

(see the IPU block diagram and Table 38-1).

38.1.2.1.4.1 Processing flows

Several time-shared data flows are supported, as described in the following table.

Table 38-5. Time-Shared Data Flows through the IPU

Name	Number	Type	Flow	Target	Restrictions
Display Refresh/ Update	5 flows (at most two of them of type DS1)	DS1	Fmem -> DP -> Display	Synchronous Access (e.g. display refresh; controlled by the DI)	
		DS2	Fmem -> DP -> Display	Asynchronous Access (e.g. display update)	
		DS3	Fmem <-> Display	Generic Data Transfer	
	1 flow	DS4	ARM Platform<-> Display	Direct Access	
Video Playback	flows	PL1	Bmem -> VDIC -> IC -> Bmem -> IRT -> Fmem + DSx	Main option	
		PL2	Fmem -> IRT -> Bmem -> IC -> DP	Low power (branching to DSx, as a video plane)	Large enough window No other video flows
		PL3	Fmem -> VDIC -> IC -> DP	Low power (branching to DSx, as a video plane)	Interlaced source Large enough window No other video flows
Camera Preview	1 flow (VF2 may be used also as a playback flow)	VF1	Sensor -> IC -> Bmem -> IRT -> Fmem+DSx	main option	Single progressive input
		VF2	Sensor -> Fmem -> VDIC -> IC -> Bmem -> IRT -> Fmem+ DSx	two inputs and/or interlaced input	When the VDIC is used, one of the three input fields can go directly from the sensor to the VDIC. In that case the sensor output goes to the memory via the VDIC and not the SMFC
		VF4	Sensor -> IC -> Fmem + DS1	Low power RAM-less Display Single Display Buffer (in internal memory) Tearing-less	Single progressive input Refresh rate = 2x sensor frame rate Large enough window No other video flows
Video Record	1 flow	RCx	IC -> Bmem -> IRT -> Fmem	(branching from VFx)	

Table continues on the next page...

Table 38-5. Time-Shared Data Flows through the IPU (continued)

Name	Number	Type	Flow	Target	Restrictions
Graphic Overlays	2 flows	GF1	Fmem -> IC	(combining with the main flow	
	2 flow	GF2	Fmem -> DP		

Comments

- System memory usage - legend
  - Fmem: frame double-buffer (page-flip) in system memory (typically external memory: DDR DRAM [address 0x1000_0000 - 0xFFFF_FFFF])
  - Bmem: two possibilities
    - A frame double buffer, as above
    - A band (4-256 rows) double-buffer (page-flip) in system memory (could be internal memory: OCRAM [address 0x0900_0000 - 0x093F_FFFF])
  - Direct arrow between two processing stages represents an internal pipeline
- Time-sharing
  - IC can time-share tightly three flows: one VFx, one RCx and one PLx (with independent processing parameters)
  - DP can time-share one DS1 flow and a one DS2 flow (each with different destinations and independent processing parameters)
  - Direct access to display (DS4) time-shares tightly the display port with other active DSx flows.
  - Other time-sharing (between PLx flow and DS2 and DS3 flows in IRT) is frame-by-frame
- Any of the processing stages in the above flows can be skipped.
- Triggering and synchronization
  - Flow segments starting from a sensor are triggered and synchronized by input from the sensor
  - Flow segments ending with display refresh are triggered and synchronized by the refresh control mechanism in the DI.
  - Flow segments starting and ending in system memory, are triggered either by the double buffering mechanism or by explicit configuration and are processed continuously without delay, at a rate determined by the available resources. These flow segments have a lower priority than the sensor/display-driven flows.

The functionality of each of the processing blocks is described below.

38.1.2.1.4.2 Display Processor (DP)

The Display Processor performs all the processing required for data sent to a display.

- Input: from the IC and/or from system memory

    - Order: rows, progressive or interlaced
    - Format: YUVA/RGBA, non-decimated, 8 bits/value
- Processing chain
    - Combining 2 video/graphics planes
    - Overlaying a simple HW cursor 32 x 32 pixels, uniform color; may be combined logically with the full plane.
    - Color conversion/correction - linear (multiplicative & additive) programmable including:
        - YUV <-> RGB, YUV<->YUV conversions where YUV stands for any one of the color formats defined in the MPEG-4 standard
        - Adjustments: brightness, contrast, color saturation, etc.
        - Special effects: gray-scale, color inversion, sephia, blue-tone, etc.
        - Color-preserving clipping, for gamut mapping
        - Hue-preserving gamut mapping - for minimal color distortion
        - Applied to the output of combining or to one of the inputs
    - Gamma correction and contrast stretching - programmable piecewise-linear map
- Output: to display (through the DC)
    - Rate: up to 240M pixels/sec
    - Format: YUV/RGB, non-decimated, 8 bits/value

The DP processes a single data flow at any given time, but supports up to three data flows by time sharing.

- A Primary flow:
    - The input is loaded periodically using a timer (e.g. for a synchronous access)
    - Optionally, frames are skipped if the content has not changed (as appropriate for a smart display)
- Two secondary flows:
    - Asynchronous; processed when the DP is not needed for the primary flow (during blanking intervals or when a primary frame is skipped).
    - The two secondary flows are switched frame-by-frame.

38.1.2.1.5 Video De-Interlacer or Combiner (VDIC)

The Video De-Interlacer as well as the Combiner have two operation modes

- De-interlacing: converts an interlaced video stream to progressive order.
- Combining: combines two video/graphics planes and a background color

38.1.2.1.5.1 De-interlacing in the VDIC

The Video De-Interlacer converts an interlaced video stream to progressive order, using a high-quality 3-field motion-adaptive filter.

- Video source - SDTV: 480i30 (720x480 @ 30 fps) or 576i25 (720x576 @ 25 fps) and HDTV: 1080i/30 (1920x1080 @ 30 fps)
- Input: -three consecutive fields
  - Source
    - The most recent field may come from the CSI or from system memory
    - The other two fields are read from memory
  - Field size: up to 968x1024 pixels (may be a vertical stripe of a wider field; e.g. 1920 pixels)
  - Pixel format: YUV 4:2:2/4:2:0, 8 bits/value
- Output: progressive frame
  - Destination: to system memory or to the Image Converter.
  - Frame size: up to 968x2048 pixels
  - Rate: up to 240 MP/sec (e.g., 1920x1080 @ 85 fps)
  - Format: same as input format

The de-interlacing is performed using a high-quality 3-field filter which is motion adaptive:

- For slow motion - retains the full resolution (of both top and bottom fields)
- For fast motion - prevents motion artifacts

The VDIC supports a single video stream at any given time.

38.1.2.1.5.2 Combining in the VDIC

- Input for combining: two progressive video/graphics planes
  - Source: system memory
  - Plane size: up to 1920x1200 pixels.
  - Pixel format: RGB/YUV 4:2:2, 8 bits/value

In this mode:

- The two input planes are read from system memory, using the previous field and next field input FIFOs.
- Their relative height, up/down is configurable
- Each of them may cover only part of the output frame. For the remaining part of the frame, the chip uses the following values:
  - Down plane: a 24-bit background color, stored in an internal register
  - Up plane: a transparent pixel
- The combining method is identical to the one in the DP, IC.

38.1.2.1.5.3 Image Converter (IC)

The Image Converter performs various operations on a video stream.

- Input: from sensor or from system memory
  - Frame size: up to 4096x4096 pixels
  - Rate: up to 200M pixels/sec (e.g. 5 MP @ 30 fps + 35% blanking intervals)
  - Order: rows, progressive. If resizing is not used, interlaced order is also acceptable.
  - Pixel format: YUV/RGB, 8 bits/value
- Processing chain:
  - Resizing
    - Fully flexible resizing ratio Maximal downsizing ratio: 8:1. Subject to this limitation, any N->M resizing can be performed.
    - Independent horizontal and vertical resizing ratios.
  - Color conversion/correction - linear (multiplicative & additive) programmable, including:
    - YUV <-> RGB, YUV<->YUV conversions where YUV stands for any one of the color formats defined in the MPEG-4 standard
    - Adjustments: brightness, contrast, color saturation...
    - Special effects: gray-scale, color inversion, sephia, blue-tone...
  - Combining with a graphics plane (e.g. application-specific overlay)
  - Horizontal inversion
- Output: to system memory or (for a single active flow) to a display device (through the DP).
  - Frame size: up to 1024x1024 pixels
  - Rate: up to 100Mpixels/sec (e.g. 1920x1080 @ 30 fps)
  - Order: rows, progressive. If resizing is not used, interlaced order is also acceptable.
  - Format: YUV/RGB, 8 bits/value

The IC supports three time-shared data flows: record, camera preview and playback (the first two share a common input).

38.1.2.1.5.4 Image Rotator (IRT)

- Input/output: from/to system memory
  - Rate
    - Up to 120M pixels/sec (when a single task is active).
    - Up to 100M pixels/sec (when more than one task is active).
  - Order: raster scan of 8x8-pixel blocks
  - Format: YUV/RGB, non-decimated, 8 bits/value
- Transformation: combination of the following
  - 90-degree rotation
  - Horizontal inversion
  - Vertical inversion

38.1.2.1.6 Automatic Procedures

The IPU is equipped with powerful control and synchronization capabilities to perform its tasks with minimal involvement of the ARM Core and minimal use of memory.

In particular, it includes:

- An integrated DMA controller with an AXI master port, allowing autonomous access to the system memory.
- An integrated display controller, performing screen refresh of a RAM-less display.
- A page-flip double buffering mechanism, synchronizing read and write access to system memory, to prevent tearing effects.
- A double/triple buffer synchronization mechanism with a video/graphics source.
- Internal synchronization, e.g., between input from sensor and output to display

As a result, in most cases, the ARM platform is involved only when it also performs part of the processing (e.g. video coding). In particular, the following procedures are performed by the IPU completely autonomously:

- Screen refresh for RAM-less displays
- Update of the ("partial plane") display buffer used for screen refresh (located either in system memory or in an external display controller, e.g. of a smart display or graphics accelerator), when the content is generated in a different ("full plane") buffer.

Typically, there are extended periods of times in which there is no other activity in the system. The ARM platform, being idle, can be put to a low-power mode, reducing the power consumption and extending significantly the battery life.

The IPU supports several techniques to reduce further the power consumption of the display system:

- Dynamically optimized screen refresh rate (see Screen Refresh below)
- Optimized update of the display buffer (see Update Of The Display Buffer below)
- Dynamic backlight control, with low-light compensation by image enhancement

Further features and capabilities of the automatic procedures are outlined below

38.1.2.1.6.1 Screen Refresh

- The refresh rate may vary within a predefined range. Within this range, the rate is dynamically adjusted to the content update rate.
- An indication about the availability of new content is obtained as follows:

    - If the page-flip double buffering is used, the mechanism provides this indication
    - If only a single buffer is used (and incrementally updated), the IPU can receive an indication of a modification from the ARM platform (by setting an internal flag).
- The IPU counts the refresh cycles: the total and those with new content. The ARM platform can use these counters to optimize display management (e.g. switching display buffer compression on/off). The counters are reset by the ARM platform.
- The transferred data may be processed on the way, using the IC and DP.

38.1.2.1.6.2 Update Of The Display Buffer

- Conditional update The IPU can receive an external "snooping" signal indicating a modification of the full plane buffer (as during screen refresh above). It monitors the signal and, upon detection, it performs one of the following:
    - Performs an update, without any SW intervention
    - Interrupts the ARM core, that can initiate some more involved procedure (e.g. selective update)
- Automatic display of a changing image (animation) or moving image (scrolling) This is implemented by reading frames (from a full plane buffer) with incremental offset. When the IPU reaches the last programmed frame, it can perform one of the following:
    - Return to the first frame, without any SW intervention
    - Interrupt the ARM platform, to generate the next content.
- The timing of the update can be adjusted to avoid tearing.
- The transferred data may be processed on the way, using the IC and DP

38.1.2.1.6.3 Camera Preview

- Tearing artifacts can be prevented by (automatic) page-flip double buffering in system memory
- Alternatively, the video stream from an image sensor can be sent directly to the display buffer used for screen refresh. The significance of this option is that only a single frame buffer is needed (and not two). This buffer may be located either in system memory or in an external display controller.
    - This option is useful, e.g., in a low frame rate, when tearing is not visible.
    - When tearing must be prevented, the refresh cycle in the display can be synchronized with the timing signals from the sensor (two refresh cycles for each input frame): the IPU receives a VSYNC signal from the sensor and generates from it synchronization signals for the display.

38.2 External Signals

The table found here describes the external signals of IPU1.

Table 38-6. IPU1 External Signals

Signal	Description	Pad	Mode	Direction
IPU1_CSI0_DATA00	-	EIM_D27	ALT2	I
IPU1_CSI0_DATA01	-	EIM_D26	ALT2	I
IPU1_CSI0_DATA02	-	EIM_D31	ALT3	I
IPU1_CSI0_DATA03	-	EIM_D30	ALT3	I
IPU1_CSI0_DATA04	-	CSI0_DAT4	ALT0	I
IPU1_CSI0_DATA05	-	CSI0_DAT5	ALT0	I
IPU1_CSI0_DATA06	-	CSI0_DAT6	ALT0	I
IPU1_CSI0_DATA07	-	CSI0_DAT7	ALT0	I
IPU1_CSI0_DATA08	-	CSI0_DAT8	ALT0	I
IPU1_CSI0_DATA09	-	CSI0_DAT9	ALT0	I
IPU1_CSI0_DATA10	-	CSI0_DAT10	ALT0	I
IPU1_CSI0_DATA11	-	CSI0_DAT11	ALT0	I
IPU1_CSI0_DATA12	-	CSI0_DAT12	ALT0	I
IPU1_CSI0_DATA13	-	CSI0_DAT13	ALT0	I
IPU1_CSI0_DATA14	-	CSI0_DAT14	ALT0	I
IPU1_CSI0_DATA15	-	CSI0_DAT15	ALT0	I
IPU1_CSI0_DATA16	-	CSI0_DAT16	ALT0	I
IPU1_CSI0_DATA17	-	CSI0_DAT17	ALT0	I
IPU1_CSI0_DATA18	-	CSI0_DAT18	ALT0	I
IPU1_CSI0_DATA19	-	CSI0_DAT19	ALT0	I
IPU1_CSI0_DATA_EN	-	CSI0_DATA_EN	ALT0	I
IPU1_CSI0_HSYNC	-	CSI0_MCLK	ALT0	I
IPU1_CSI0_PIXCLK	-	CSI0_PIXCLK	ALT0	I
IPU1_CSI0_VSYNC	-	CSI0_VSYNC	ALT0	I
IPU1_CSI1_DATA00	-	EIM_DA9	ALT2	I
IPU1_CSI1_DATA01	-	EIM_DA8	ALT2	I
IPU1_CSI1_DATA02	-	EIM_DA7	ALT2	I
IPU1_CSI1_DATA03	-	EIM_DA6	ALT2	I
IPU1_CSI1_DATA04	-	EIM_DA5	ALT2	I
IPU1_CSI1_DATA05	-	EIM_DA4	ALT2	I
IPU1_CSI1_DATA06	-	EIM_DA3	ALT2	I
IPU1_CSI1_DATA07	-	EIM_DA2	ALT2	I
IPU1_CSI1_DATA08	-	EIM_DA1	ALT2	I
IPU1_CSI1_DATA09	-	EIM_DA0	ALT2	I
IPU1_CSI1_DATA10		EIM_D22	ALT3	I
		EIM_EB1	ALT2	

Table continues on the next page...

Table 38-6. IPU1 External Signals (continued)

Signal	Description	Pad	Mode	Direction
IPU1_CSI1_DATA11		EIM_D21	ALT3	I
		EIM_EB0	ALT2	
IPU1_CSI1_DATA12		EIM_A17	ALT2	I
		EIM_D28	ALT3	
IPU1_CSI1_DATA13		EIM_A18	ALT2	I
		EIM_D27	ALT3	
IPU1_CSI1_DATA14		EIM_A19	ALT2	I
		EIM_D26	ALT3	
IPU1_CSI1_DATA15		EIM_A20	ALT2	I
		EIM_D20	ALT3	
IPU1_CSI1_DATA16		EIM_A21	ALT2	I
		EIM_D19	ALT3	
IPU1_CSI1_DATA17		EIM_A22	ALT2	I
		EIM_D18	ALT3	
IPU1_CSI1_DATA18		EIM_A23	ALT2	I
		EIM_D16	ALT3	
IPU1_CSI1_DATA19		EIM_A24	ALT2	I
		EIM_EB2	ALT3	
IPU1_CSI1_DATA_EN		EIM_DA10	ALT2	I
		EIM_D23	ALT4	
IPU1_CSI1_HSYNC		EIM_DA11	ALT2	I
		EIM_EB3	ALT4	
IPU1_CSI1_PIXCLK		EIM_A16	ALT2	I
		EIM_D17	ALT3	
IPU1_CSI1_VSYNC		EIM_DA12	ALT2	I
		EIM_D29	ALT6	
IPU1_DI0_D0_CS	-	EIM_D23	ALT1	O
IPU1_DI0_D1_CS	-	EIM_A25	ALT4	O
IPU1_DI0_DISP_CLK	-	DI0_DISP_CLK	ALT0	O
IPU1_DI0_PIN01	-	EIM_D22	ALT2	IO
IPU1_DI0_PIN02	-	DI0_PIN2	ALT0	O
IPU1_DI0_PIN03	-	DI0_PIN3	ALT0	O
IPU1_DI0_PIN04	-	DI0_PIN4	ALT0	O
IPU1_DI0_PIN05	-	EIM_D16	ALT2	O
IPU1_DI0_PIN06	-	EIM_D17	ALT2	O
IPU1_DI0_PIN07	-	EIM_D18	ALT2	O
IPU1_DI0_PIN08	-	EIM_D19	ALT2	O
IPU1_DI0_PIN11	-	EIM_D30	ALT2	O
IPU1_DI0_PIN12	-	EIM_D31	ALT2	O

Table continues on the next page...

Table 38-6. IPU1 External Signals (continued)

Signal	Description	Pad	Mode	Direction
IPU1_DI0_PIN13	-	EIM_D28	ALT7	O
IPU1_DI0_PIN14	-	EIM_D29	ALT7	O
IPU1_DI0_PIN15	-	DI0_PIN15	ALT0	O
IPU1_DI0_PIN16	-	EIM_D20	ALT2	O
IPU1_DI0_PIN17	-	EIM_D21	ALT2	O
IPU1_DI1_D0_CS		EIM_DA13	ALT1	O
		EIM_D18	ALT4	
IPU1_DI1_D1_CS	-	EIM_DA14	ALT1	O
IPU1_DI1_DISP_CLK	-	EIM_A16	ALT1	O
IPU1_DI1_PIN01	-	EIM_DA15	ALT1	IO
IPU1_DI1_PIN02		EIM_DA11	ALT1	O
		EIM_D23	ALT6	
IPU1_DI1_PIN03		EIM_DA12	ALT1	O
		EIM_EB3	ALT6	
IPU1_DI1_PIN04	-	EIM_DA15	ALT2	O
IPU1_DI1_PIN05	-	EIM_CS0	ALT1	O
IPU1_DI1_PIN06	-	EIM_CS1	ALT1	O
IPU1_DI1_PIN07	-	EIM_OE	ALT1	O
IPU1_DI1_PIN08	-	EIM_RW	ALT1	O
IPU1_DI1_PIN11	-	EIM_D26	ALT1	O
IPU1_DI1_PIN12	-	EIM_A25	ALT3	O
IPU1_DI1_PIN13	-	EIM_D27	ALT1	O
IPU1_DI1_PIN14	-	EIM_D23	ALT7	O
IPU1_DI1_PIN15		EIM_DA10	ALT1	O
		EIM_D29	ALT1	
IPU1_DI1_PIN16	-	EIM_BCLK	ALT1	O
IPU1_DI1_PIN17	-	EIM_LBA	ALT1	O
IPU1_DISP0_DATA00	-	DISP0_DAT0	ALT0	IO
IPU1_DISP0_DATA01	-	DISP0_DAT1	ALT0	IO
IPU1_DISP0_DATA02	-	DISP0_DAT2	ALT0	IO
IPU1_DISP0_DATA03	-	DISP0_DAT3	ALT0	IO
IPU1_DISP0_DATA04	-	DISP0_DAT4	ALT0	IO
IPU1_DISP0_DATA05	-	DISP0_DAT5	ALT0	IO
IPU1_DISP0_DATA06	-	DISP0_DAT6	ALT0	IO
IPU1_DISP0_DATA07	-	DISP0_DAT7	ALT0	IO
IPU1_DISP0_DATA08	-	DISP0_DAT8	ALT0	IO
IPU1_DISP0_DATA09	-	DISP0_DAT9	ALT0	IO
IPU1_DISP0_DATA10	-	DISP0_DAT10	ALT0	IO
IPU1_DISP0_DATA11	-	DISP0_DAT11	ALT0	IO

Table continues on the next page...

Table 38-6. IPU1 External Signals (continued)

Signal	Description	Pad	Mode	Direction
IPU1_DISP0_DATA12	-	DISP0_DAT12	ALT0	IO
IPU1_DISP0_DATA13	-	DISP0_DAT13	ALT0	IO
IPU1_DISP0_DATA14	-	DISP0_DAT14	ALT0	IO
IPU1_DISP0_DATA15	-	DISP0_DAT15	ALT0	IO
IPU1_DISP0_DATA16	-	DISP0_DAT16	ALT0	IO
IPU1_DISP0_DATA17	-	DISP0_DAT17	ALT0	IO
IPU1_DISP0_DATA18	-	DISP0_DAT18	ALT0	IO
IPU1_DISP0_DATA19	-	DISP0_DAT19	ALT0	IO
IPU1_DISP0_DATA20	-	DISP0_DAT20	ALT0	IO
IPU1_DISP0_DATA21	-	DISP0_DAT21	ALT0	IO
IPU1_DISP0_DATA22	-	DISP0_DAT22	ALT0	IO
IPU1_DISP0_DATA23	-	DISP0_DAT23	ALT0	IO
IPU1_DISP1_DATA00	-	EIM_DA9	ALT1	IO
IPU1_DISP1_DATA01	-	EIM_DA8	ALT1	IO
IPU1_DISP1_DATA02	-	EIM_DA7	ALT1	IO
IPU1_DISP1_DATA03	-	EIM_DA6	ALT1	IO
IPU1_DISP1_DATA04	-	EIM_DA5	ALT1	IO
IPU1_DISP1_DATA05	-	EIM_DA4	ALT1	IO
IPU1_DISP1_DATA06	-	EIM_DA3	ALT1	IO
IPU1_DISP1_DATA07	-	EIM_DA2	ALT1	IO
IPU1_DISP1_DATA08	-	EIM_DA1	ALT1	IO
IPU1_DISP1_DATA09	-	EIM_DA0	ALT1	IO
IPU1_DISP1_DATA10	-	EIM_EB1	ALT1	IO
IPU1_DISP1_DATA11	-	EIM_EB0	ALT1	IO
IPU1_DISP1_DATA12	-	EIM_A17	ALT1	IO
IPU1_DISP1_DATA13	-	EIM_A18	ALT1	IO
IPU1_DISP1_DATA14	-	EIM_A19	ALT1	IO
IPU1_DISP1_DATA15	-	EIM_A20	ALT1	IO
IPU1_DISP1_DATA16	-	EIM_A21	ALT1	IO
IPU1_DISP1_DATA17	-	EIM_A22	ALT1	IO
IPU1_DISP1_DATA18	-	EIM_A23	ALT1	IO
IPU1_DISP1_DATA19	-	EIM_A24	ALT1	IO
IPU1_DISP1_DATA20	-	EIM_D31	ALT1	IO
IPU1_DISP1_DATA21	-	EIM_D30	ALT1	IO
IPU1_DISP1_DATA22	-	EIM_D26	ALT7	IO
IPU1_DISP1_DATA23	-	EIM_D27	ALT7	IO
IPU1_EXT_TRIG	-	EIM_D28	ALT6	I
IPU1_SISG0	-	NANDF_CS2	ALT1	O
IPU1_SISG1	-	NANDF_CS3	ALT1	O

Table continues on the next page...

Table 38-6. IPU1 External Signals (continued)

Signal	Description	Pad	Mode	Direction
IPU1_SISG2		EIM_A24	ALT4	O
		EIM_D26	ALT6	
IPU1_SISG3		EIM_A23	ALT4	O
		EIM_D27	ALT6	
IPU1_SISG4	-	KEY_COL4	ALT1	O
IPU1_SISG5	-	KEY_ROW4	ALT1	O

38.3 Clocks

The table found here describes the clock sources for IPU.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 38-7. IPU Clocks

Clock name	Clock Root	Description
hsp_clk	ipu1_ipu_hsp_clk_root	HSP clock
ipp_di_0_ext_clk	ipu1_di0_clk_root	IPU DI0 interface pixel clock
ipp_di_1_ext_clk	ipu1_di1_clk_root	IPU DI1 interface pixel clock
ipu_master_hclk	ahb_clk_root	IPU master clock

38.4 Functional Description

This section provides a complete functional description of the block.

38.4.1 IPU detailed block diagram

The following figure is the IPU top level block diagram.

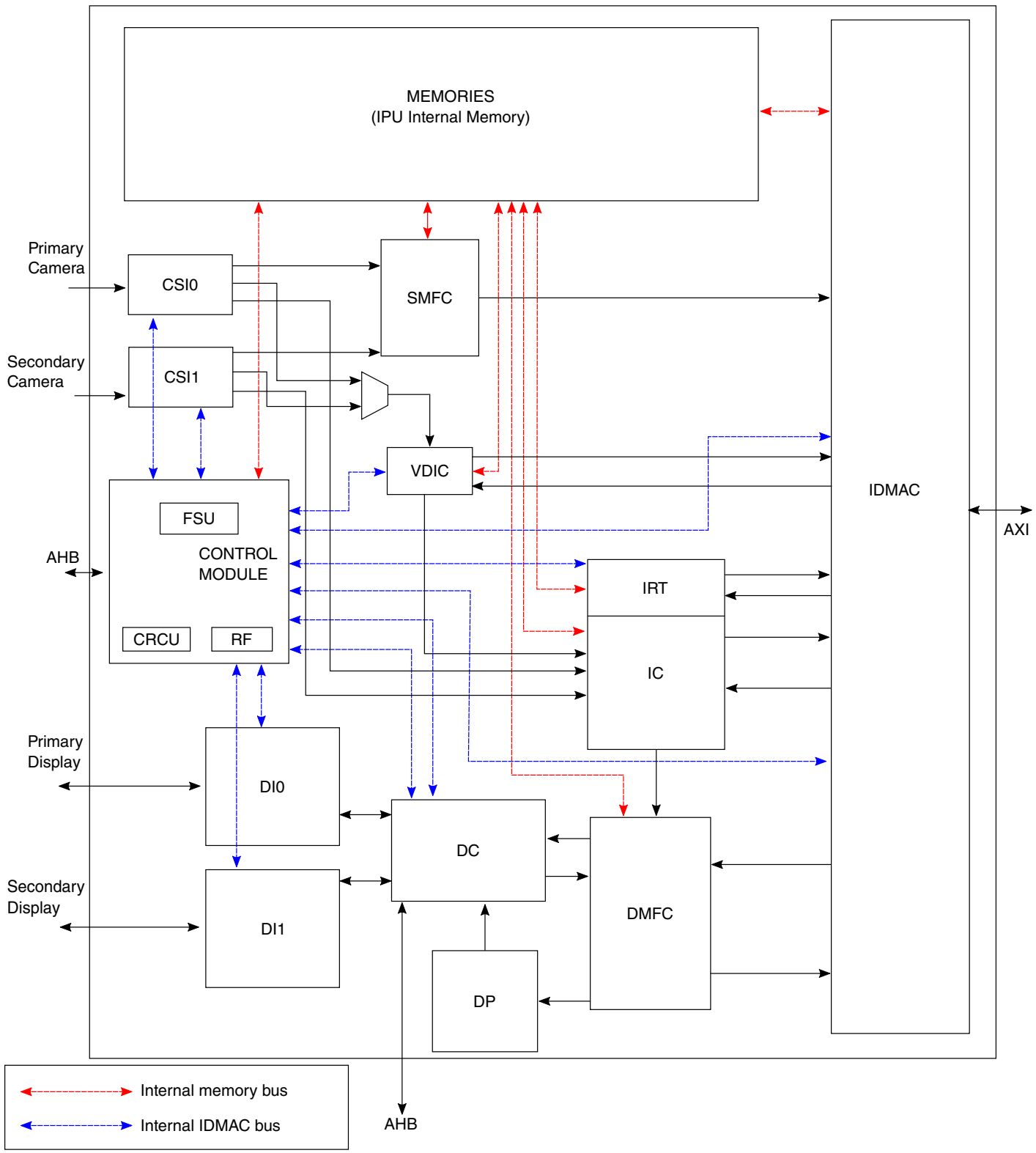


Figure 38-2. IPU Detailed Block Diagram

38.4.2 Image DMA Controller (IDMAC)

The following diagram is the IDMAC's block diagram.

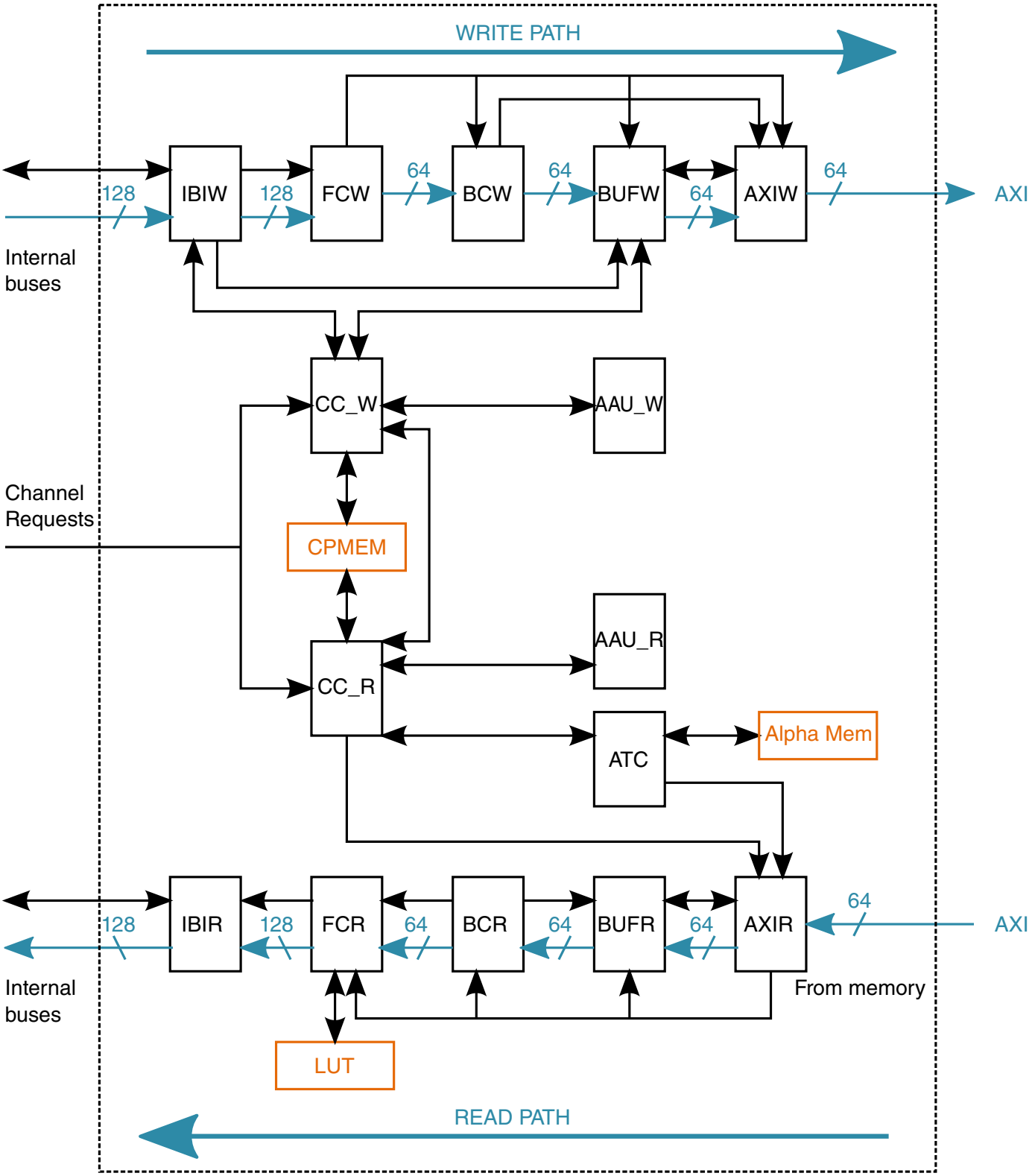


Figure 38-3. IDMAC Block Diagram

The following table describes the IDMAC's sub-block glossary

Table 38-8. IDMAC's sub modules glossary

Sub Module	Description
IBIW	Internal Bus Interface Write
IBIR	Internal Bus Interface Read
FCW	Format Converter Write
FCR	Format Converter Read
BCW	Buffer Controller Write
BCR	Buffer Controller Read
BUFW	Buffer Write
BUFR	Buffer Read
AXIW	AXI Write
AXIR	AXI Read
CC_W	Channel Control Write
CC_R	Channel Control Read
AAU_W	Address Arithmetic Unit Write
AAU_R	Address Arithmetic Unit Read
ATC	Alpha Transparency Controller
LUT	Look up table
CPMEM	Channel Parameter Memory

38.4.2.1 IDMAC's channels

The table below summarizes the IDMAC's channels.

Enabling a channel is done via the channel's corresponding IDMAC_CH_EN bit.

Table 38-9. IDMAC DMA channels list

Channel #	Source	Destination	Alternate Destination	Purpose	Data type
0	CSI (via SMFC)	Fmem		VF2 - Bayer; BPP>8; JPEG; MIPI additional channels	Generic or Pixel
1	CSI (via SMFC)	Fmem		VF2 - Bayer; BPP>8; JPEG; MIPI additional channels	Generic or Pixel
2	CSI (via SMFC)	Fmem		VF2 - Bayer; BPP>8; JPEG; MIPI additional channels	Generic or Pixel
3	CSI (via SMFC)	Fmem		VF2 - Bayer; BPP>8; JPEG; MIPI additional channels	Generic or Pixel
5	VDIC	Bmem	IC	VF1/VF2	Pixel
8	Fmem	VDIC		Previous field	Pixel
9	Fmem	VDIC		Current field	Pixel
10	Fmem	VDIC		Next field	Pixel

Table continues on the next page...

Table 38-9. IDMAC DMA channels list (continued)

Channel #	Source	Destination	Alternate Destination	Purpose	Data type
11	Bmem	IC		video plane for post processing task	Pixel
12	Bmem	IC		video plane for PrP tasks (view finder or encoding)	Pixel
13	VDIC	Fmem		Recent field from CSI	Pixel
14	Fmem	IC		graphics plane for PrP task (view finder or encoding)	Pixel
15	Fmem	IC		graphics plane for post processing task	Pixel
16	Reserved				
17	Fmem	IC		Transparency (alpha for channel 14)	Generic
18	Fmem	IC		Transparency (alpha for channel 15)	Generic
19	Fmem	VDIC		Transparency (alpha for channel 25)	Generic
20	IC	Bmem		Preprocessing data from IC (encoding task) to memory	Pixel
21	IC	Bmem	DMFC	Preprocessing data from IC (viewfinder task) to memory; This channel can be configured to send the data directly to the DMFC. This is done by programming the IC_DMFC_SEL bit.	Pixel
22	IC	Bmem		Postprocessing data from IC to memory	Pixel
23	Fmem	DP		DP primary flow - main plane	Pixel
24	Fmem	DP		DP secondary flow - main plane	Pixel
25	Fmem	VDIC		Plane #1 of the VDIC for combining	pixel
26	Fmem	VDIC		Plane #3 of the VDIC for combining	pixel
27	Fmem	DP		DP primary flow - auxiliary plane	Pixel
28	Fmem	DC		DC channel for both sync and async flows	Pixel
29	Fmem	DP		DP secondary flow - auxiliary plane	Pixel
30	Reserved				
31	Fmem	DP		Transparency (alpha for channel 27)	Generic
32	Reserved				
33	Fmem	DP		Transparency (alpha for channel 29)	Generic
34	Reserved				
35	Reserved				
36	Reserved				
37	Reserved				
38	Reserved				
39	Reserved				
40	DC	Fmem		DC read channel	Generic
41	Fmem	DC		DC async flow	Generic
42	Fmem	DC		DC command stream	Generic
43	Fmem	DC		DC command stream	Generic
44	Fmem	DC		DC output mask	Generic
45	Bmem	IRT		Rotation for post Encoding task	Pixel
46	Bmem	IRT		Rotation for viewfinder task	Pixel

Table continues on the next page...

Table 38-9. IDMAC DMA channels list (continued)

Channel #	Source	Destination	Alternate Destination	Purpose	Data type
47	Bmem	IRT		Rotation for post processing task	Pixel
48	IRT	Bmem		Rotation for Encoding task	Pixel
49	IRT	Bmem		Rotation for viewfinder task	Pixel
50	IRT	Bmem		Rotation for post processing task	Pixel
51	Fmem	DP		Transparency (alpha for channel 23)	Generic
52	Fmem	DP		Transparency (alpha for channel 24)	Generic
53-63	Reserved				

38.4.2.2 IBIW & IBIR - Internal bus interface for write and read

The Internal Bus Interface handles the internal IPU protocol communicating between the IDMAC and the IPU's sub modules.

The IBIR handles channels that perform read from external memory. The IBIW handles channels that perform write accesses to external memory.

38.4.2.3 FCW & FCR - Format converter write and read

The format converter performs packing ("write direction") / unpacking ("read" direction) of pixels with programmable position and width of color components, decoding 4- or 8-bits coded pixels according to a loaded look-up table, panning of an image read from the system memory according to a panning offset (start pixel address).

The format converter supports formats with a pixel width of 4, 8, 12, 16, 18, 24 or 32 bits. The format converter unit handles two pixels simultaneously.

The IPU sub modules can handle only the formats, presented below. Each component is 8 bit:



Figure 38-4. IPU internal pixel formats

The pixel can be stored in the memory in the formats presented below. (R/G/B means that this could component can be R or G or B; A is the location of the alpha component).

For Read:

R/G/B/A	R/G/B/A	R/G/B/A	R/G/B/A

For Write:

R/G/B	R/G/B	R/G/B	A

A	R/G/B	R/G/B	R/G/B

Figure 38-5. IPU external pixel formats

Formatting parameters are written in the channel parameter memory (CPMEM - Channel Parameter Memory). The following parameters are used:

- Offset OFS0 between MSB position of the color component 0 and MSB position of packed pixel. The color component 0 occupies the most significant bits of the unpacked pixel (mostly this is the R component). The OFS0 range is from 0 to 31.
- Color component 0 width (WID0 minus 1).
- Offset OFS1 between MSB position of the color component 1 and MSB position of packed pixel. The color component 1 occupies the middle left bits of the unpacked pixel (mostly this is the G component). The OFS1 range is from 0 to 31.
- Color component 1 width (WID1 minus 1).
- Offset OFS2 between MSB position of the color component 2 and MSB position of packed pixel. The color component 2 occupies the middle right bits of the unpacked pixel (mostly this is the B component). The OFS2 range is from 0 to 31.
- Color component 2 width (WID2 minus 1).
- Offset OFS3 between MSB position of the color component 3 and MSB position of packed pixel. The color component 3 occupies the least significant bits of the unpacked pixel (mostly this is the A component). The OFS3 range is from 0 to 31. For write specified DMA channels, the OFS3 value is set to 24 or 0 bits.
- In cases of read with separate alpha (alpha is located in a separate buffer in the system's memory than the pixel data), the alpha component size is defined according to WID3.

The figures below show examples of data packing and unpacking.

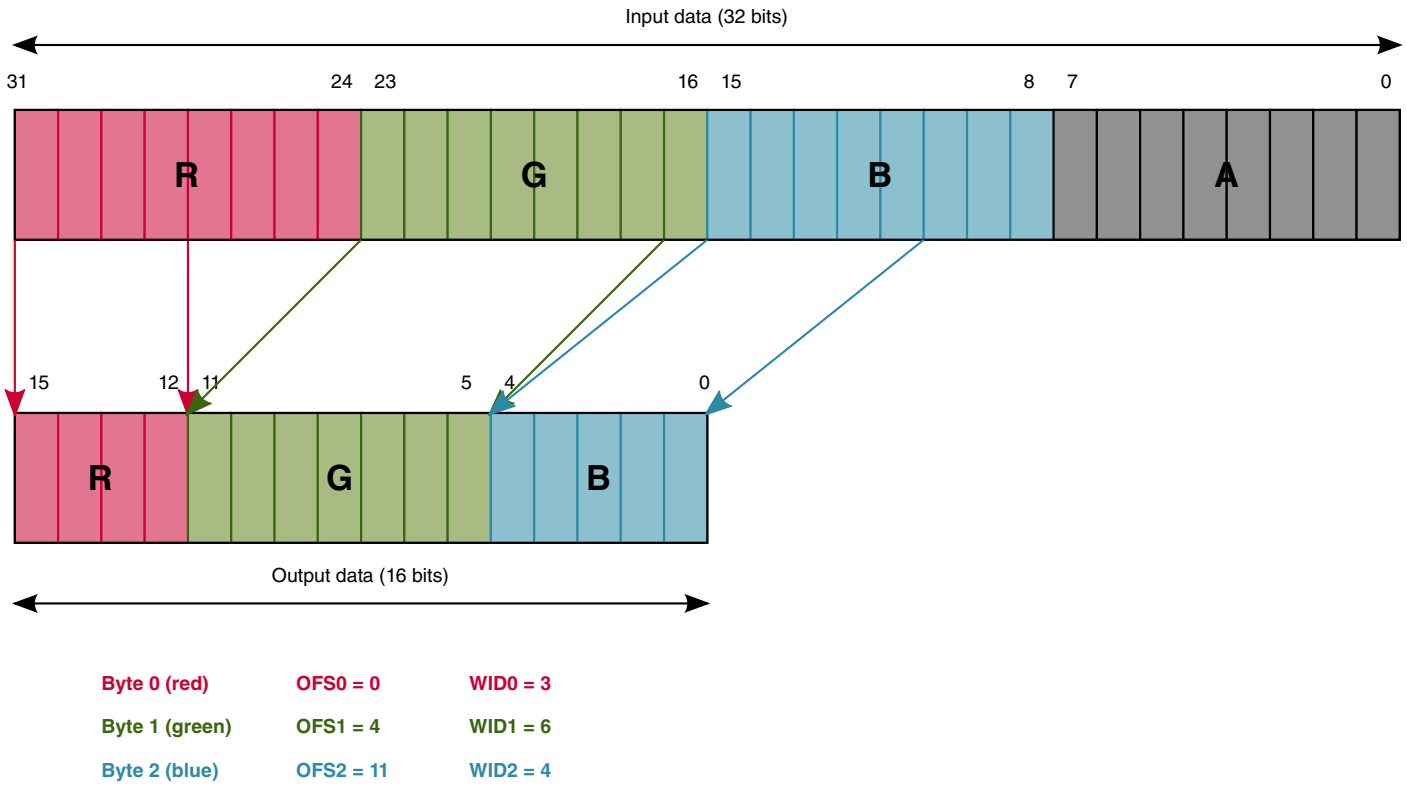


Figure 38-6. Data Packing Example

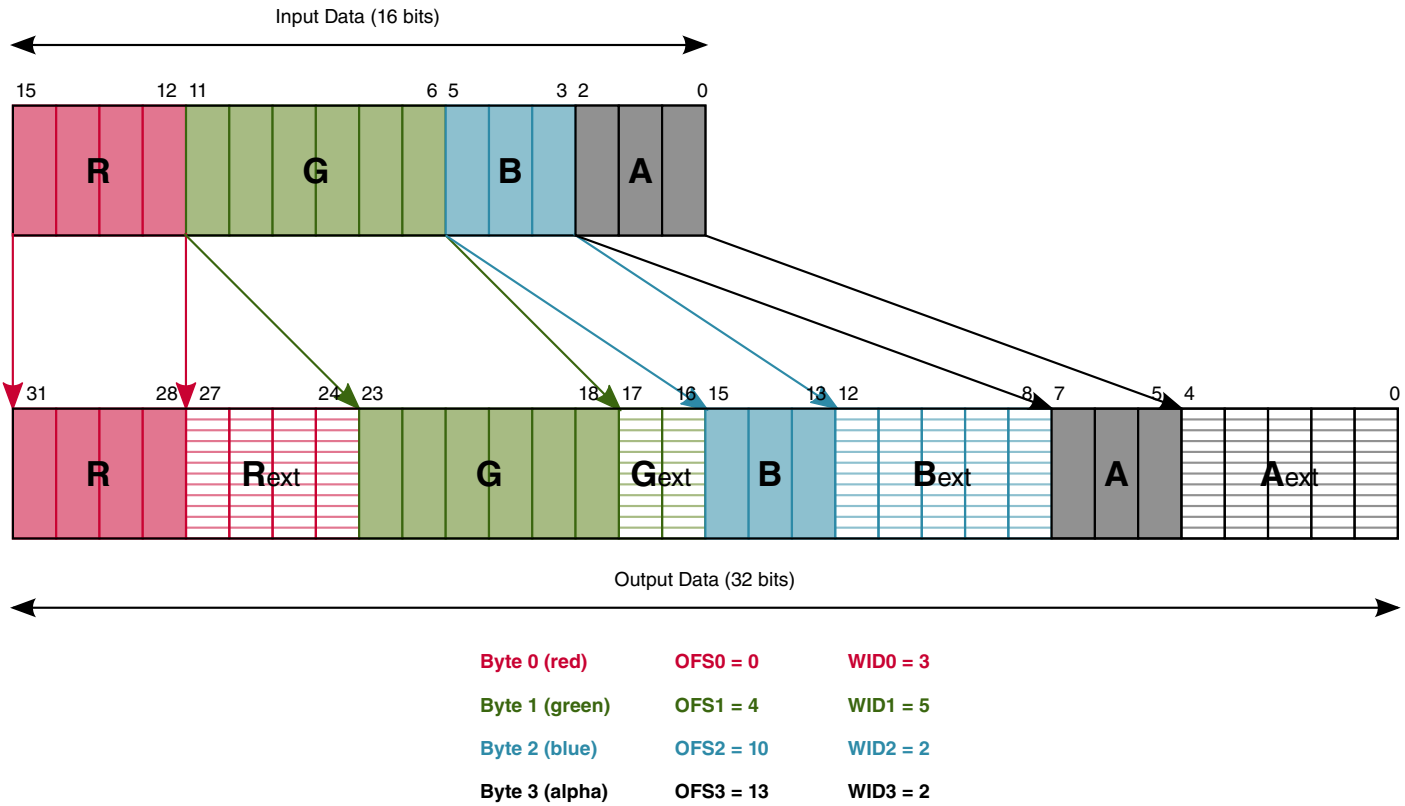


Figure 38-7. Data Unpacking Example

If read data has the coded format, it is decoded via the look-up table. The Look-Up Table Memory (LUT- Look Up Table) must be loaded at the IDMAC initialization step. The LUT output format must match the IPU internal format RGBA 8888 where R is placed in MSB and A is placed in LSB. The A field is used only for graphics data.

38.4.2.4 Buffering units

The buffering units (BUFW, BUFR) are used to store the data consisting of different coded color components.

- On write transactions (BUFW), before writing to memory & after the format has been coded.
- On read transactions (BUFR), after reading from memory & before the format has been decoded.

Each buffering unit includes 4 X 64 bytes buffers. Each buffer, for each direction, can handle any kind of color component (interleaved - Y / partial interleaved - Y, UV / non interleaved - Y, U, V).

The Write buffers are controlled by the buffer controller for write (BCW) and AXIW units.

The Read buffers are controlled by the buffer controller for read (BCR) and AXIR units.

38.4.2.4.1 Handling real time channels

The memory controller connected to the AXI bus of the IPU can use the AXI ID associated with each burst in order to distinguish between real time and non real time channels.

In order to do that, the user has to set the channel's ID according to the settings in the memory controller and set the priority of the channel according to its nature. The buffer controller (BCW/BCR) holds all the pending requests that won the arbitration. However, as the memory controller can distinguish between the real time channels and non real time channels within the IPU, there could be a situation where the real time requests are blocked as the IPU's queue is filled with non real time requests. To avoid that, the user can limit the number of non-real time requests in the queue.

The queue for read requests can handle up to 8 requests. The queue for write requests can handle up to 6 requests. The user can limit the number of non real time requests by setting the USED_BUFS_MAX_W for write requests and USED_BUFS_MAX_R for read requests. The feature that limits the number of requests is enabled by setting the USED_BUFS_EN_R bit for the read requests and USED_BUFS_EN_W for the write requests.

38.4.2.5 AXIW - AXI Write and AXIR - AXI Read

The AXI Master Interfaces are responsible for data transfer from/to the system memory. The Interface supports only 64-bits burst accesses of 1-8 words, with nonalignment of a byte resolution.

2 separate & independent AXI masters are used for "read" (AXIR) & "write" (AXIW), each can be programmed (via CPMEM) with 4 different IDs to support out-of-order accesses within bursts.

38.4.2.6 CC_W & CC_R - Channel Control Write and Read

The Channel Control unit is the main control unit of the IDMAC.

- It arbitrates the channels according to the priority.
- Controls the address arithmetic unit.

- Functions as a memory interface to the CPMEM. It reads the parameters from the CPMEM, prepares the controls accordingly and writes back updated parameters to the CPMEM.
- The read unit provides the parameters to the IBIR unit
- The write unit provides the parameters to the AXIW unit

The CC calculates all the parameters related to the access except the address and BS (burst size), which are calculated at the AAU.

The priority is set according to:

- The channel's corresponding bit in the IDMAC_CH_PRI_1 & IDMAC_CH_PRI_2 registers.
- The watermark signal generated from the sub module. The watermark signal is ignored unless the channel's corresponding IDMAC_WM_EN bit is set.
- Special priority for alpha channels

A priority value is calculated for each of the enabled channels according to the above conditions. Then, the CC unit selects between the channels with the same priority value in a round-robin fashion.

Table 38-10. Calculated priority value

alpha channel	Channel's priority bit	watermark signal	Priority Value
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	1
1	0	1	2
1	1	0	3
1	1	1	4

38.4.2.6.1 Locking the arbitration and reordering the AXI bursts

The performance of the overall system can be improved by sending AXI bursts of consecutive addresses. This can be done by reordering the AXI bursts in a way that accesses that belong to the same channel will be sent one after the other. This can be done by controlling the IDMAC_LOCK_# bits of the corresponding channel.

An IDMAC request that won the arbitration will be served for the next bursts according to the settings of the IDMAC_LOCK_# bits. The block that issued the request (DMFC on IPU) will assert the request only if it has enough room in its FIFO to accept the number

of bursts defined by the IDMAC_LOCK_# bits. IPU provides this capability to channels that serve real time screen refresh to synchronous display (23,27,28). In addition, it provides this capability to channels that may generate very short AXI bursts (IC and IRT)

38.4.2.7 AAU_W & AAU_R- Address Arithmetic Unit for Write and Read

The AAU_R & AAU_W units calculate the address in the system memory to be accessed by the IPU. These units also calculate the burst size (BS). The address calculation is done according to parameters stored in the CPMEM.

The following main addressing parameters are used:

- XB-Horizontal pixel position in frame
- YB-Vertical pixel position in frame
- SL-Stride line minus 1 (gap in bytes between two pixels in the same column in two consecutive rows).
- SX-Horizontal pixel scrolling offset
- SY-Vertical pixel scrolling offset
- EBA-Frame buffer base address in bytes (there are two such parameters to support double buffering)
- BPP-Bits per pixel
- FW-Frame width minus 1
- FH-Frame height minus 1

Relations between the addressing parameters and image frame are shown in the table below.

The system memory address in bytes is calculated as:

ADDR = EBA + (XB + SX) * BPP + (YB + SY) * (SL + 1)

with 0 < XB <= FW and 0 < YB <= FH.

For non-interleaved formats the 4 LSB bits of SX are defined according to the IOX parameter.

When double buffering is used, the EBA0 is the base address of the buffer 0 and the EBA1 is the base address of the buffer 1. The IPU_CHA_CUR_BUF Register is a status register. It contains 1-bit pointers to the current working buffers for all IPU DMA channels. The IPU automatically toggles a pointer after completion of the current buffer processing. If the ARM platform is a data source for specific double-buffered channel, it should check this status bit in order to know what is the IPU current buffer. The ARM platform is allowed to write to the buffer only when a working DMA channel does not

use it. After the ARM platform has been fill the buffer, it has to set the corresponding bit in the IPU_CHA_BUF0_RDY and IPU_CHA_BUF1_RDY Registers. If needed, the ARM platform can only clear the pointer by writing 1 but not set it.

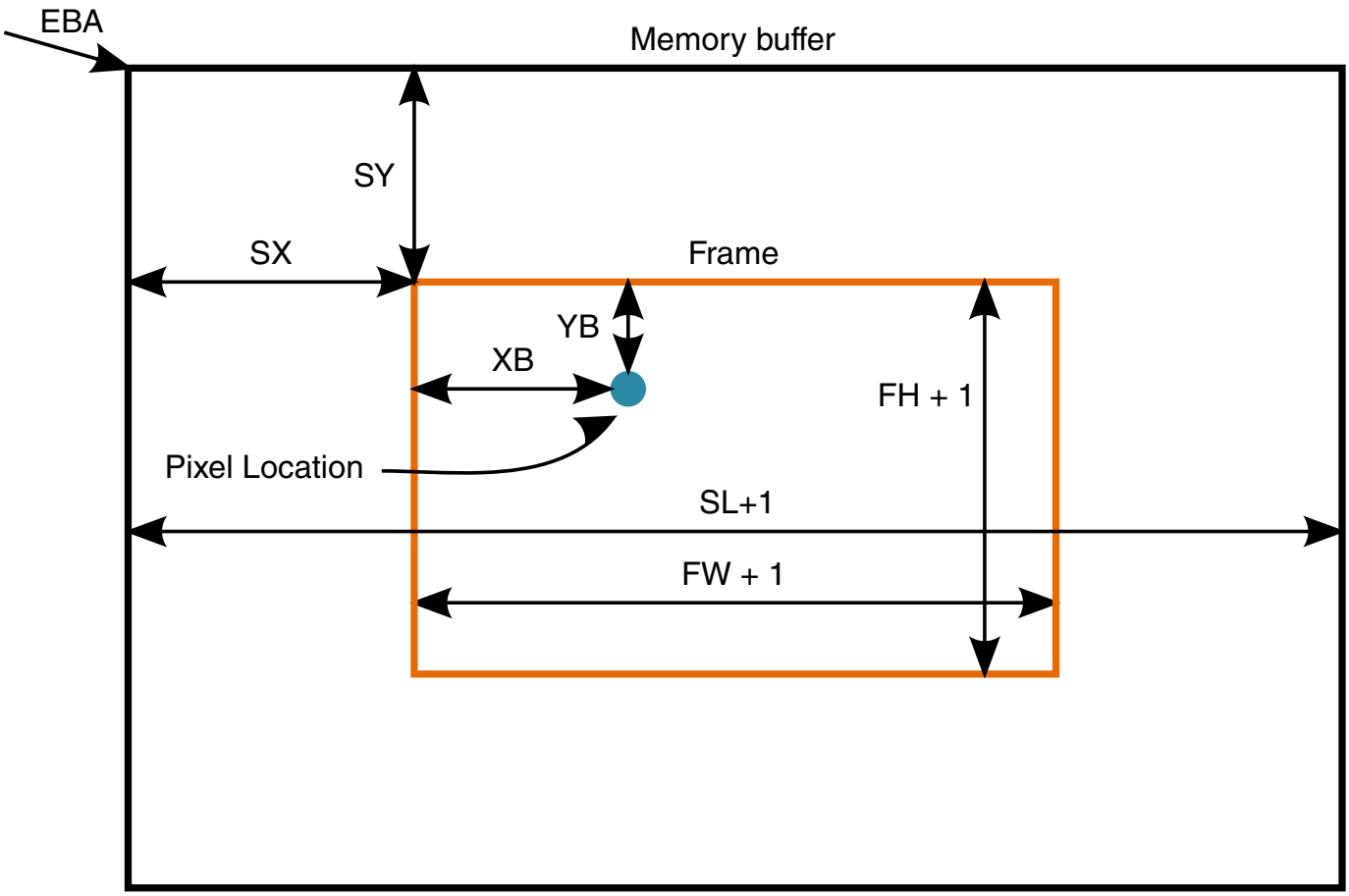


Figure 38-8. Addressing Parameters and Image Frame

The XB and YB coordinates are calculated according to addressing mode. There are two addressing modes:

- 2D mode
- Block mode

In 2D mode the pixel data is transferred to the memory row-by-row. There are two ways to use 2D mode: start from YB = 0 and finish at YB <= FH (YB is incremented) or start from YB = FH and finish at YB <= 0 (YB is decremented). The second option provides vertical flip of the image.

In block mode the frame is divided into blocks. This is needed for rotation or post-filtering, where the order used for data transfers is block-by-block. The order of the block transfer is according to the VF,HF and ROT bits in the IDMAC Channel Parameter

Memory. The order within the block is row-by-row where the block size is limited by the block width (BW) and block height (BH) parameters. The BW and BH parameters are set by IC rotation section and cannot be configured through the Channel Parameter Memory.

The Channel Control is responsible for the address calculation flow. It takes channel parameters from the Channel Parameter Memory, updates them and controls the Address Arithmetic Unit.

38.4.2.7.1 Scrolling support

Automatic display of a changing image (animation) or moving image (scrolling) is implemented by reading frames (from a background buffer) with incremental offset. Enabling the scrolling feature is done by setting the channel's corresponding SCE bit.

The scrolling step is controlled by channel's corresponding SDX and SDY parameters, and the scrolling direction is defined by the channel's corresponding SDRX and SDRY parameters. The maximum number of scrolled frames to be read is defined by the channel's corresponding SM parameter.

When the last programmed frame is reached (IDMAC's internal counter reached SM), IDMAC can perform one of the following (controlled by the SCC bit):

- Return to the first frame, without any SW intervention. The return point is defined by SX0 and SY0 parameters.
- Interrupt the ARM platform, to generate the next content.

38.4.2.8 ATC - Alpha Transparency Controller

The Alpha transparency controller (ATC) handles the alpha buffers on the external memory for cases where the pixel data and the alpha data are located on separate buffers (separate alpha mode).

In that case the IDMAC reads the alpha data and the pixel data, merge them together and provides a pixel that includes the alpha information to the relevant sub module. The ATC's main functions are:

- Generates requests for alpha channels, following a request to pixel data from the module.
- Maintain an internal alpha memory buffer for each channel.
- When there isn't enough alphas in the memory the ATC blocks the corresponding pixel channels.
- When there is a request of pixel channel the ATC load and accumulate its alphas in a register.

- ATC memory controller can manage 8 channels of alphas.
- ATC supports synchronous new frame before end of frame errors.

In order to configure the channel to use separate alpha the channel's corresponding IDMAC_SEP_AL bit should be set in addition to the ALU bit in the CPMEM of the corresponding channel.

The following figure is the ATC's block diagram.

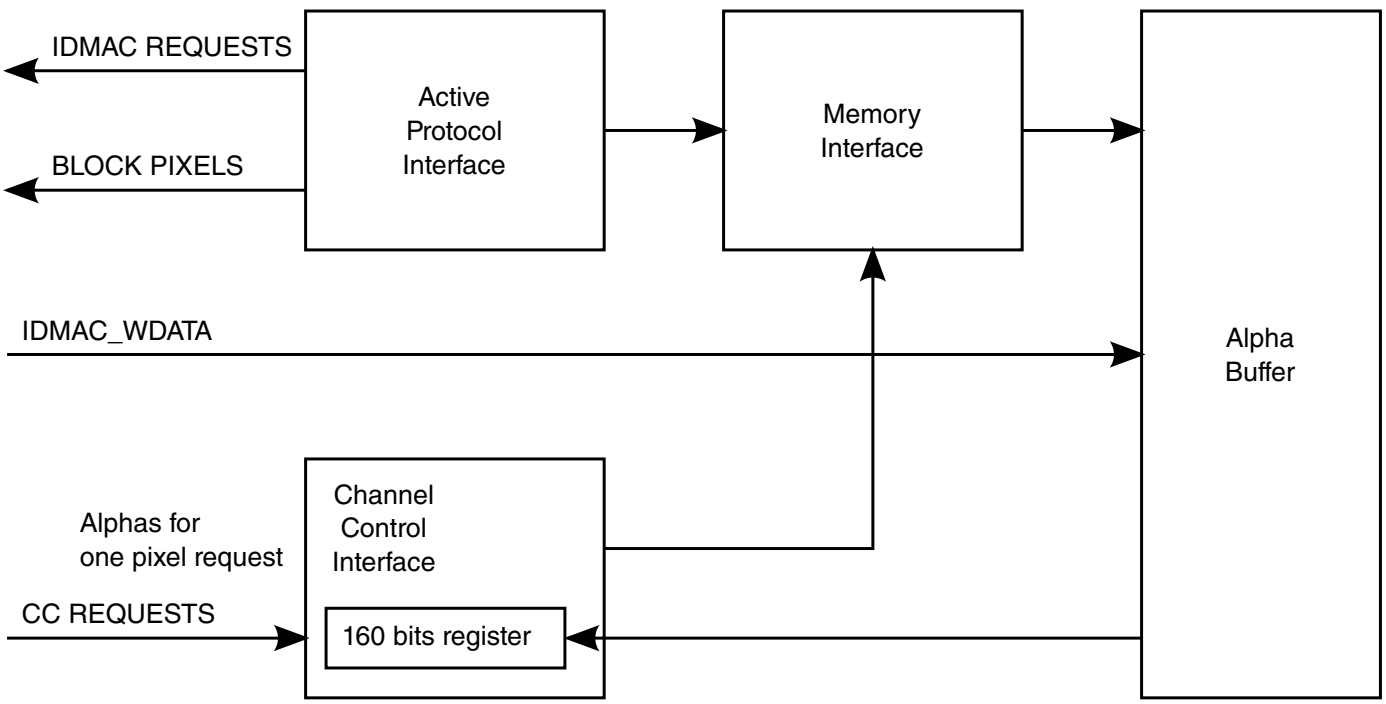


Figure 38-9. ATC block diagram

The ATC alpha buffer memory can hold up to 8 buffers of alphas. A pointer to a buffer in the ATC memory is defined according to the ALBM parameter in the CPMEM. The table below describes the relations between a data channel, an alpha channel and the pointer in the alpha buffer memory.

Table 38-11. Alpha channels mapping

data channel number	associated alpha channel number	Alpha buffer memory (ALBM)
14	17	0
15	18	1
27	31	2
29	33	3
23	51	4
24	52	5
25	19	6

38.4.2.8.1 Conditional read

The alpha data can be used to reduce reads from the memory of pixels that are going to be transparent (alpha = 0). The conditional read feature is enabled by the CRE bit in the CPMEM.

If all of the corresponding alpha values for a single burst of pixels are equal to zero, the IDMAC will block the access to the external memory and provide a data of all zeros to the corresponding channel. This way some of the accesses to the memory can be prevented, thus reducing the load on the memory.

38.4.2.9 LUT- Look Up Table

When working in coded pixel format, the data read from the memory is the decoded value of pixel according to address given. In case of 8 bit code, the data read from the memory is the decoded value of the pixel according to the address given.

In case of 4 bit code configuration, The address of the 4 bit decoded values is set according to DEC_SEL parameter in the CPMEM

00 = addresses 0 to 15

01 = addresses 64 to 79

10 = addresses 128 to 143

11 = addresses 192 to 207

Table 38-12. Look-Up Table Memory Structure

Address	Word	DEC_SEL	Description
0	Word0	4 BPP = 00 8 BPP = don't care	Decoded Pixels [15:0] for both 8 and 4 bit coded configuration
...	...		
15	Word15		
16	Word16	don't care	Decoded Pixels [63:16] for 8 bit coded configuration only
...	...		
63	Word63		
64	Word64	4 BPP = 01 8 BPP = don't care	Decoded Pixels [79:64] for both 8 and 4 bit coded configuration
...	...		
79	Word79		
80	Word80	don't care	Decoded Pixels [127:80] for 8 bit coded configuration only
...	...		
127	Word127		

Table continues on the next page...

Table 38-12. Look-Up Table Memory Structure (continued)

Address	Word	DEC_SEL	Description
128	Word128	4 BPP = 10 8 BPP = don't care	Decoded Pixels [143:128] for both 8 and 4 bit coded configuration
...	...		
143	Word143		
144	Word144	don't care	Decoded Pixels [191:144] for 8 bit coded configuration only
...	...		
191	Word191		
192	Word192	4 BPP = 11 8 BPP = don't care	Decoded Pixels [207:192] for both 8 and 4 bit coded configuration
...	...		
207	Word207		
208	Word208	don't care	Decoded Pixels [255:208] for 8 bit coded configuration only
...	...		
255	Word255		

38.4.2.10 CPMEM - Channel Parameter Memory

The CPMEM holds the configuration parameters for each IDMAC channel. The CPMEM can holds the settings of 80 channels.

Each channel's settings are defined by a two mega-words. Each mega-word is 160 bits wide. The following diagram illustrates the CPMEM's structure.

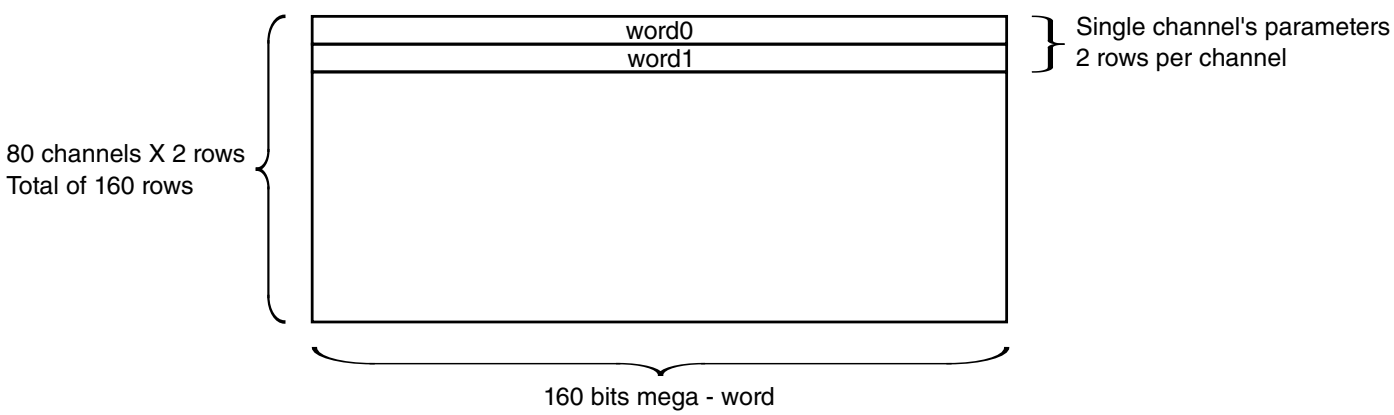


Figure 38-10. CPMEM structure

Each IDMAC channel can be configured to work in one of two different modes:

- Non Interleaved mode where the Y:U:V data is organized in 3 separate buffers in the system's memory
- Interleaved mode where the Y:U:V data is organized in a single buffer in the system's memory

The parameters and the way they are organized are different for each mode. The Pixel Format Select (PFS) value of the CPMEM words are used by the IPU to determine if the words should be interpreted as interleaved or non-interleaved format.

NOTE

CPMEM is "Memory Mapped" and can be accessed by the AHB bus through the Control Module (see Memory Access Unit).

The following tables describe the IDMAC parameters and their organization in each mode.

38.4.2.10.1 CPMEM's words' structure for non interleaved mode

The table below describes the CPMEM's words' structure for non interleaved mode. Each CPMEM word consist of 160 bits. The bits that are not listed in the table below are reserved bits.

Table 38-13. Channel Parameters Memory for non-interleaved

Name	Mnemonic	Size	Location	Description
Word 0				
XV Virtual Coordinate	XV	10 bits	W0[9:0]	Variable coordinates for determining next block address. {X1,Y1} and {X2,Y2} coordinates will be determined according to {XV,YV} upon restart of channel. These coordinates are used for Y:U:V (Y pointer) and RGB formats.
YV Virtual Coordinate	YV	9 bits	W0[18:10]	
XB inner Block Coordinate	XB	13 bits	W0[31:19]	Variable coordinates for determining address within the block. These coordinates are used for Y:U:V (Y pointer) and RGB formats. Need 24 bits for 2D transfer support.
YB inner Block Coordinate	YB	12 bits	W0[43:32]	
New Sub Block	NSB_B	1 bit	W0[44]	This bit determines if the next value for {XB,YB} should be taken from {XB,YB} saved in channel parameter memory or from new {x1,y1}/{x2,y2}.
Current Field	CF	1 bit	W0[45]	CF = 0 Current field is even CF = 1 Current field is odd
Mem U Buffer Offset	UBO	22 bits	W0[67:46]	Double buffer destination address offset for Y:U:V (U pointer) formats.The actual physical address value is divided by 8 (i.e. this parameter includes bits [24:3] of the actual address)

Table continues on the next page...

Table 38-13. Channel Parameters Memory for non-interleaved (continued)

Mem V Buffer Offset	VBO	22 bits	W0[89:68]	Double buffer destination address offset for Y:U:V (V pointer) format. The actual physical address value is divided by 8 (i.e. this parameter includes bits [24:3] of the actual address)
Initial Offset X	IOX	4 bits	W0[93:90]	The IOX parameter, is the offset in pixels for a frame that starts at a non aligned address. for 42x formats must be even.
Reduce Double Read or Writes	RDRW	1 bits	W0[94:94]	This bit is relevant for YUV4:2:0 formats. For read channels: U and V components are not read from odd rows. (read - supported only for the VDIC) For write channels: U and V components are not written to odd rows. (write - supported for all write channels)
Scan Order	SO	1 bit	W0[113]	SO = 0 Scan order is progressive SO = 1 Scan order is interlaced
Band Mode	BNDM	3 bits	W0[116:114]	BNDM = 000 bands disable. BNDM = 001 bands enable. Band height = 4 lines. BNDM = 010 bands enable. Band height = 8 lines. BNDM = 011 bands enable. Band height = 16 lines. BNDM = 100 bands enable. Band height = 32 lines. BNDM = 101 bands enable. Band height = 64 lines. BNDM = 110 bands enable. Band height = 128 lines. BNDM = 111 bands enable. Band height = 256 When working in band mode, the channel's corresponding IDMAC_BNDM_EN bit has to be set.
Block Mode	BM	2 bits	W0[118:117]	BM = 00 block mode disable. BW = FW, BH = FH BM = 01 block mode enable. BW = 8, BH = 8 BM = 10 block mode enable. BW = 16, BH = 16 (this mode is reserved for future use) BM = 11 not used
Rotation	ROT	1 bit	W0[119]	ROT = 0 -> No rotation ROT = 1 -> 90 degree rotation clockwise
Horizontal Flip	HF	1 bit	W0[120]	HF = 0 -> No flip HF = 1 -> Horizontal flip enable
Vertical Flip	VF	1 bit	W0[121]	VF = 0 -> No flip VF = 1 -> Vertical flip enable

Table continues on the next page...

Table 38-13. Channel Parameters Memory for non-interleaved (continued)

Threshold Enable	THE	1 bit	W0[122]	THE = 0 -> Threshold disable THE = 1 -> Threshold enable
Conditional Access Polarity	CAP	1 bit	W0[123]	CAP = 0 -> If conditional bit in CM register is low skip the access CAP = 1 -> If conditional bit in CM register is high skip the access. This mode is reserved for future use.
Conditional Access Enable	CAE	1 bit	W0[124]	CAE = 0 -> Conditional access disable CAE = 1 -> Conditional access enable This mode is reserved for future use.
Frame Width	FW	13 bits	W0[137:125]	Number of pixels in one row, of the channel frame. FW 000000000000 = 0001 pixels 000000000001 = 0002 pixels ..... 111111111111 = 8192 pixels
Frame Height	FH	12 bits	W0[149:138]	Number of pixels in one column, of the channel frame. FH 000000000000 = 0001 line 000000000001 = 0002 lines ..... 111111111111 = 4096 lines For progressive YUV 4:2:0 (non interleaved and partial interleaved formats) the FH value should be a multiple of 2. For interlaced YUV 4:2:0 (non interleaved and partial interleaved formats) the FH value should be a multiple of 4.
End of Line interrupt	EOLI	1 bit	W0[150]	End of line interrupt enable. The end of line indication is asserted once the last data of the line has been written. 1 - generate an end of line interrupt 0 - no affect
Word 1				
Ext Mem Buffer 0 Address	EBA0	29 bits	W1[28:0]	1st double buffer destination address for RGB and Y:U:V (Y pointer) formats. This parameter must not be changed if the corresponding channel is enabled. The actual physical address value is divided by 8 (i.e. this parameter includes bits [31:3] of the actual address)

Table continues on the next page...

Table 38-13. Channel Parameters Memory for non-interleaved (continued)

Ext Mem Buffer 1 Address	EBA1	29 bits	W1[57:29]	2nd double buffer destination address for RGB and Y:U:V (Y pointer) formats. This parameter must not be changed if the corresponding channel is enabled. The actual physical address value is divided by 8 (i.e. this parameter includes bits [31:3] of the actual address)
Interlace Offset	ILO	20 bits	W1[77:58]	2nd double buffer destination address for RGB and Y:U:V (Y pointer) formats. An interlaced data stored in the memory can be read as a consecutive progressive only if FW*BPP is a multiplication of 8. The actual physical address value is divided by 8 (i.e. this parameter includes bits [22:3] of the actual address). For YUV420 formats, the ILO is relevant only to the Y component as the U and V components do not exist for the even lines. This value is signed
Number of Pixels in Whole Burst Access	NPB	7 bits	W1[84:78]	Number of pixels per burst access. The following are valid numbers of pixels in a memory burst access according to the BPP parameter: NPB 0000111 = 08 pixels in each burst 0001111 = 16 pixels in each burst ..... 0111111 = 64 pixels in each burst Range: 16BPP => 1 -> 32 pixels (for YUV444 NI) 08BPP => 1 -> 64 pixels (For YUV420 NI/PI and YUV422 NI/PI) In NI/PI formats the NPB has to be a multiplication of 8
Pixel Format Select	PFS	4 bits	W1[88:85]	4'h0 = non-interleaved 4:4:4 4'h1 = non-interleaved 4:2:2 4'h2 = non-interleaved 4:2:0 4'h3 = partial interleaved 4:2:2 4'h4 = partial interleaved 4:2:0 4'h5 to 4'hF = NA
Alpha Used	ALU	1 bit	W1[89]	1 = the alpha associated with the data of this channel resides on another channel (separate buffer) 0 = the alpha associated with the data of this channel resides along with the pixel data (same buffer) The corresponding alpha channel must be enabled to assure correct behavior.

Table continues on the next page...

Table 38-13. Channel Parameters Memory for non-interleaved (continued)

Alpha Channel Mapping	ALBM	2 bits	W1[92:90]	Alpha channel mapping - This parameter is a pointer to a buffer in the ATC memory. This parameter is relevant only to data channels that are associated with a separate alpha buffer (like graphic plane channels). The parameter should be programmed on the data channels' ALBM. Setting this parameter to any other channel has no meaning. See Table 38-11 for exact ALBM mapping.
AXI Id	ID	2 bits	W1[94:93]	AXI protocol id
Threshold	TH	7 bits	W1[101:95]	0000000 = 32 lines 0000001 = 64 lines ..... 1111111 = 4096 lines
Stride Line	SLY	14 bits	W1[115:102]	Address vertical scaling factor in bytes for memory access. Also number of maximum bytes in the "Y" component row according to memory limitations. SLY 00000000000000 = 00001 bytes 00000000000001 = 00002 bytes ..... 11111111111111 = 16384 bytes
Width3	WID3	3 bits	W1[127:125]	Fourth color component size of the input-unpacking/ output-packing pixel. WID3 000 = 1 bits 001 = 2 bits ..... 111 = 8 bits As this is a non-interleaved format, this field is relevant only to the alpha associated with this pixel channel.
Stride Line	SLUV	14 bits	W1[141:128]	Address vertical scaling factor in bytes for memory access. Also number of maximum bytes in the "U" or "V" component row according to memory limitations. SLUV 00000000000000 = 00001 bytes 00000000000001 = 00002 bytes ..... 11111111111111 = 16384 bytes
Conditional Read Enable	CRE	1 bit	W1[149:149]	This bit enables the conditional read feature.

38.4.2.10.2 CPMEM's words' structure for interleaved mode

The table below describes the CPMEM's words' structure for interleaved mode. Each CPMEM word consist of 160 bits. The bits that are not listed in the table below are reserved bits

Table 38-14. Channel Parameters Memory for interleaved

Name	Mnemonic	Size	Location	Description
Word 0				
XV Virtual Coordinate	XV	10 bits	W0[9:0]	Variable coordinates for determining next block address. {X1,Y1} and {X2,Y2} coordinates will be determined according to {XV,YV} upon restart of channel. These coordinates are used for Y:U:V (Y pointer) and RGB formats.
YV Virtual Coordinate	YV	9 bits	W0[18:10]	
YB inner Block Coordinate	XB	13 bits	W0[31:19]	Variable coordinates for determining address within the block. These coordinates are used for Y:U:V (Y pointer) and RGB formats. Need 24 bits for 2D transfer support.
XB inner Block Coordinate	YB	12 bits	W0[43:32]	
New Sub Block	NSB_B	1 bit	W0[44]	This bit determines if the next value for {XB,YB} should be taken from {XB,YB} saved in channel parameter memory or from new {x1,y1}/{x2,y2}.
Current Field	CF	1 bit	W0[45]	CF = 0 Current field is even CF = 1 Current field is odd
Scroll X counter	SX	12 bits	W0[57:46]	Holds the temporary count for the Scroll X in between frame For interleaved YUV4:2:2 formats the SX should be a multiple of 2.
Scroll Y counter	SY	11 bits	W0[68:58]	Holds the temporary count for the Scroll Y in between frame
Number of Scroll	NS	10 bits	W0[78:69]	This variable holds the total number of Scrolls
Scroll Delta X	SDX	7 bits	W0[85:79]	Frame start row offset, compared to last frame. SDX 0000000 = 00 pixels 0000001 = 01 pixels 0000010 = 02 pixels ..... 1111110 = 126 pixels 1111111 = 127 pixels For interleaved YUV4:2:2 formats the SDX should be a multiple of 2.

Table continues on the next page...

Table 38-14. Channel Parameters Memory for interleaved (continued)

Scroll Max	SM	10 bits	W0[95:86]	Frame maximum row and column increment offset in frame. SM 0000000000 = 0001 0000000001 = 0002 ..... 1111111111 = 1024
Scrolling Configuration	SCC	1 bit	W0[96]	Determines if scrolling will continue from zero when NS counter has reached SM or stop at the current value for SX and SY for the next frames to come. SCC 0 => Scrolling will stop at NS = SM 1 => Scrolling will start from "0" at NS = SM
Scrolling Enable	SCE	1 bit	W0[97]	SCE = 0 Scrolling disable SCE = 1 Scrolling enable
Scroll Delta Y	SDY	7 bits	W0[104:98]	Frame start column offset, compared to last frame. SDY 0000000 = 00 pixels 0000001 = 01 pixels 0000010 = 02 pixels ..... 1111110 = 30 pixels 1111111 = 127 pixels
Scroll Horizontal Direction	SDRX	1 bit	W0[105]	Determines if the next frame will move right or left compared to the current frame. SDRX 0 => Next frame will be right of current 1 => Next frame will be left of current
Scroll Vertical Direction	SDRY	1 bit	W0[106]	Determines if the next frame will move down or up compared to the current frame. SDRY 0 => Next frame will be down of current 1 => Next frame will be up of current
Bits Per Pixel	BPP	3 bits	W0[109:107]	3'h0 = 32 Bits per pixel 3'h1 = 24 Bits per pixel 3'h2 = 18 Bits per pixel

Table continues on the next page...

Table 38-14. Channel Parameters Memory for interleaved (continued)

				3'h3 = 16 Bits per pixel 3'h4 = 12 Bits per pixel 3'h5 = 08 Bits per pixel 3'h6 = 04 Bits per pixel
Decode Address Select	DEC_SEL	2 bits	W0[111:110]	Upon 4BPP, selects between two look-up tables DEC_SEL 00 = addresses 0 to 15 01 = addresses 64 to 79 10 = addresses 128 to 143 11 = addresses 192 to 207
Access Dimension	DIM	1 bit	W0[112]	DIM = 0 Access Dimension is 2d DIM = 1 Access Dimension is 1d
Scan Order	SO	1 bit	W0[113]	SO = 0 Scan order is progressive SO = 1 Scan order is interlaced
Band Mode	BNDM	3 bits	W0[116:114]	BNDM = 000 bands disable. BNDM = 001 bands enable. Band height = 4 lines. BNDM = 010 bands enable. Band height = 8 lines. BNDM = 011 bands enable. Band height = 16 lines. BNDM = 100 bands enable. Band height = 32 lines. BNDM = 101 bands enable. Band height = 64 lines. BNDM = 110 bands enable. Band height = 128 lines. BNDM = 111 bands enable. Band height = 256 When working in band mode, the channel's corresponding IDMAC_BNDM_EN bit has to be set.
Block Mode	BM	2 bits	W0[118:117]	BM = 00 block mode disable. BW = FW, BH = FH BM = 01 block mode enable. BW = 8, BH = 8 BM = 10 block mode enable. BW = 16, BH = 16 (this mode is reserved for future use) BM = 11 not used
Rotation	ROT	1 bit	W0[119]	ROT = 0 -> No rotation ROT = 1 -> 90 degree rotation clockwise

Table continues on the next page...

Table 38-14. Channel Parameters Memory for interleaved (continued)

Horizontal Flip	HF	1 bit	W0[120]	HF = 0 -> No flip HF = 1 -> Horizontal flip enable
Vertical Flip	VF	1 bit	W0[121]	VF = 0 -> No flip VF = 1 -> Vertical flip enable
Threshold Enable	THE	1 bit	W0[122]	THE = 0 -> Threshold disable THE = 1 -> Threshold flip enable
Conditional Access Polarity	CAP	1 bit	W0[123]	CAP = 0 -> If conditional bit in CM register is low skip the access CAP = 1 -> If conditional bit in CM register is high skip the access
Conditional Access Enable	CAE	1 bit	W0[124]	CAE = 0 -> Conditional access disable CAE = 1 -> Conditional access enable
Frame Width	FW	13 bits	W0[137:125]	Number of pixels in one row, of the channel frame. FW 0000000000000 = 0001 pixels 0000000000001 = 0002 pixels ..... 1111111111111 = 8192 pixels For interleaved YUV4:2:2 formats the FW should be a multiple of 2.
Frame Height	FH	12 bits	W0[149:138]	Number of pixels in one column, of the channel frame. FH 000000000000 = 0001 line 000000000001 = 0002 lines ..... 111111111111 = 4096 lines
End of Line interrupt	EOLI	1 bit	W0[150]	End of line interrupt enable. The end of line indication is asserted once the last data of the line has been written. 1 - generate an end of line interrupt 0 - no affect
Word 1				
Ext Mem Buffer 0 Address	EBA0	29 bits	W1[28:0]	1st double buffer destination address for RGB and Y:U:V (Y pointer) formats. This parameter must not be changed if the corresponding channel is enabled.The actual physical address value is divided by 8 (i.e. this parameter includes bits [31:3] of the actual address)

Table continues on the next page...

Table 38-14. Channel Parameters Memory for interleaved (continued)

Ext Mem Buffer 1 Address	EBA1	29 bits	W1[57:29]	2nd double buffer destination address for RGB and Y:U:V (Y pointer) formats. This parameter must not be changed if the corresponding channel is enabled.The actual physical address value is divided by 8 (i.e. this parameter includes bits [31:3] of the actual address)
Interlace Offset	ILO	20 bits	W1[77:58]	2nd double buffer destination address for RGB and Y:U:V (Y pointer) formats. An interlaced data stored in the memory can be read as a consecutive progressive only if FW*BPP is a multiplication of 8.The actual physical address value is divided by 8 (i.e. this parameter includes bits [22:3] of the actual address). This value is signed
Number of Pixels in Whole Burst Access	NPB	7 bits	W1[84:78]	Number of pixels per burst access. The following are valid numbers of pixels in a memory burst access according to the BPP parameter: NPB 0000000 = 01 pixels in each burst 0000001 = 02 pixels in each burst ..... 1111111 = 128 pixels in each burst Range: 32BPP => 1 -> 16 pixels 24BPP => 1 -> 20 pixels 16BPP => 1 -> 32 pixels 12BPP => 1 -> 40 pixels 08BPP => 1 -> 64 pixels 04BPP => 1 -> 128 pixels
Pixel Format Select	PFS	4 bits	W1[88:85]	4'h0 to 4'h4 = NA 4'h5 = Code (LUT) 4'h6 = Generic data 4'h7 = RGB (& also YUV interleaved 4:4:4) 4'h8 = interleaved 4:2:2 Y1U1Y2V1[1] 4'h9 = interleaved 4:2:2 Y2U1Y1V1[2] 4'hA = interleaved 4:2:2 U1Y1V1Y2[3] 4'hB = interleaved 4:2:2 U1Y2V1Y1[4] 4'hC to 4'hF = NA

Table continues on the next page...

Table 38-14. Channel Parameters Memory for interleaved (continued)

Alpha Used	ALU	1 bit	W1[89]	1 = the alpha associated with the data of this channel resides on another channel (separate buffer) 0 = the alpha associated with the data of this channel resides along with the pixel data (same buffer) The corresponding alpha channel must be enabled to assure correct behavior.
Alpha Channel Mapping	ALBM	3 bits	W1[92:90]	Alpha channel mapping - This parameter is a pointer to a buffer in the ATC memory. This parameter is relevant only to data channels that are associated with a separate alpha buffer (like graphic plane channels). The parameter should be programmed on the data channels' ALBM. Setting this parameter to any other channel has no meaning. See Table 38-11 for exact ALBM mapping.
AXI Id	ID	2 bits	W1[94:93]	AXI protocol id; IPU is targeted to an AXI slave that can handle up to 2 requests with 2 different IDs + one request with a third ID. In case that IPU is going to be used on a system that can handle more than 2 requests with different IDs, the number of different IDs programmed in the CPMEM for different channels is limited for 2. This limitation is relevant for read channels only. For write channels there's no such limitation
Threshold	TH	7 bits	W1[101:95]	0000000 = 32 lines 0000001 = 64 lines ..... 1111111 = 4096 lines
Stride Line	SL	14 bits	W1[115:102]	Address vertical scaling factor in bytes for memory access. Also number of maximum bytes in row according to memory limitations. SL 00000000000000 = 00001 bytes 00000000000001 = 00002 bytes ..... 11111111111111 = 16384 bytes
Width0	WID0	3 bits	W1[118:116]	First color component size of the input-unpacking/ output-packing pixel. WID0 000 = 1 bits

Table continues on the next page...

Table 38-14. Channel Parameters Memory for interleaved (continued)

				001 = 2 bits ..... 111 = 8 bits This field is relevant only for interleaved RGB format (PFS = 4'h7)
Width1	WID1	3 bits	W1[121:119]	Second color component size of the input-unpacking/ output-packing pixel. WID1 000 = 1 bits 001 = 2 bits ..... 111 = 8 bits This field is relevant only for interleaved RGB format (PFS = 4'h7)
Width2	WID2	3 bits	W1[124:122]	Third color component size of the input-unpacking/ output-packing pixel. WID2 000 = 1 bits 001 = 2 bits ..... 111 = 8 bits This field is relevant only for interleaved RGB format (PFS = 4'h7)
Width3	WID3	3 bits	W1[127:125]	Fourth color component size of the input-unpacking/ output-packing pixel. WID3 000 = 1 bits 001 = 2 bits ..... 111 = 8 bits This field is relevant only for interleaved RGB format (PFS = 4'h7)
Offset0	OFS0	5 bits	W1[132:128]	Number of bits between MSB of pixel and MSB of color component, on input. 1 states that the color component will be the first color component aligned to MSB of output pixel. OFS0 00000 = No offset 00001 = u => 1 bit sleft, p => 1 bit sright .....

Table continues on the next page...

Table 38-14. Channel Parameters Memory for interleaved (continued)

				11111 = u => 31 bit sleft, p => 31 bit sright * u = unpacking * p = packing * sleft = shift left * sright = shift right This field is relevant only for interleaved RGB format (PFS = 4'h7)
Offset1	OFS1	5 bits	W1[137:133]	Number of bits between MSB of pixel and MSB of color component on input. 2 states that the color component will be the second color component aligned to MSB output pixel. OFS1 00000 = No offset 00001 = u => 1 bit sleft, p => 1 bit sright ..... 11111 = u => 31 bit sleft, p => 31 bit sright * u = unpacking * p = packing * sleft = shift left * sright = shift right This field is relevant only for interleaved RGB format (PFS = 4'h7)
Offset2	OFS2	5 bits	W1[142:138]	Number of bits between MSB of pixel and MSB of color component on input. 3 states that the color component will be the third color component aligned to MSB output pixel. OFS2 00000 = No offset 00001 = u => 1 bit sleft, p => 1 bit sright ..... 11111 = u => 31 bit sleft, p => 31 bit sright * u = unpacking * p = packing * sleft = shift left * sright = shift right

Table continues on the next page...

Table 38-14. Channel Parameters Memory for interleaved (continued)

				This field is relevant only for interleaved RGB format (PFS = 4'h7)
Offset3	OFS3	5 bits	W1[147:143]	Number of bits between MSB of pixel and MSB of color component on input. 4 states that the color component will be the fourth color component aligned to MSB output pixel. OFS3 00000 = No offset 00001 = u => 1 bit sleft, p => 1 bit sright ..... 11111 = u => 31 bit sleft, p => 31 bit sright * u = unpacking * p = packing * sleft = shift left * sright = shift right This field is relevant only for interleaved RGB format (PFS = 4'h7)
Select SX SY Set	SXYS	1 bit	W1[148:148]	This bit selects between the settings on: SC_CORD and SC_CORD1
Conditional Read Enable	CRE	1 bit	W1[149:149]	This bit enables the conditional read feature.
Decode Address Select bit[2]	DEC_SEL2	1 bit	W1[150:150]	This field is reserved

1. Y1U1Y2V1 means byte0 = bits [7:0] =Y1; byte1 = bits [15:8] =U1; byte2 = bits [23:16] =Y2; byte3 = bits [31:24] = V1
2. Y2U1Y1V1 means byte0 =Y2; byte1 =U1; byte2 =Y1; byte3 = V1
3. U1Y1V1Y2 means byte0 =U1; byte1 =Y1; byte2 =V1; byte3 = Y2
4. U1Y2V1Y1 means byte0 =U1; byte1 =Y2; byte2 =V1;byte3 = Y1

38.4.2.10.3 Accessing the CPMEM for programming

Each IDMAC's channel's parameters are located on 2 CPMEM entries. Each Entry is 160 bit. The CPMEM is memory mapped and is accessible via the AHB bus. The AHB bus's accesses are 32bit wide. A CPMEM entry is composed of 5x32bit words. The next CPMEM entry starts at the next 8x32bit words (0x0, 0x20,0x40, etc.).

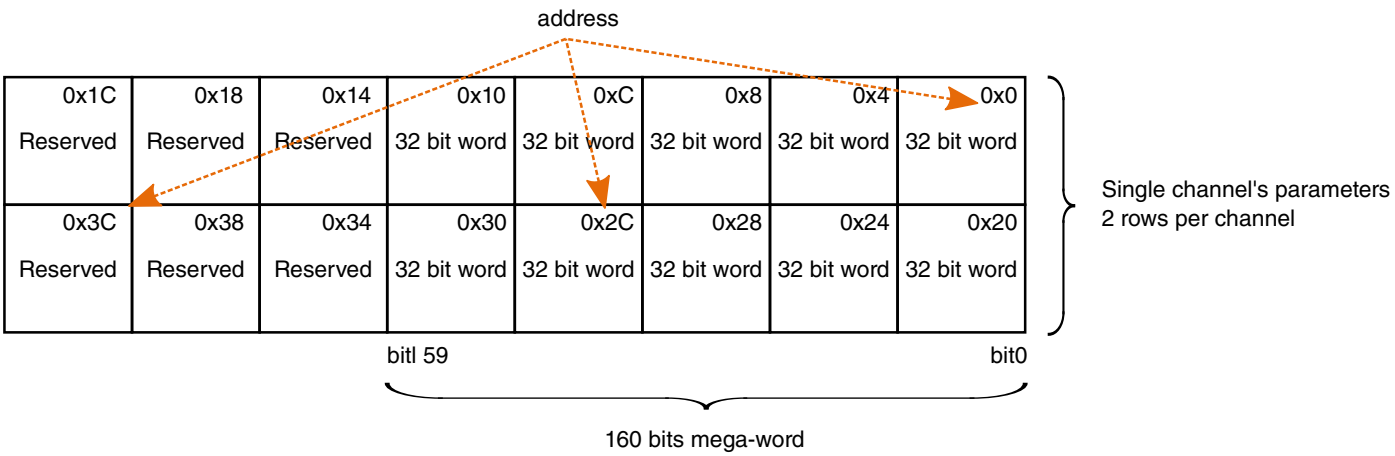


Figure 38-11. CPMEM's word structure

38.4.2.10.4 Alternate IDMAC settings

Some of the IDMAC channels support alternate flow. This means that a physical IDMAC channel can use alternate set of parameters. Switching between the flows is controlled by the CM.

The primary flow the IDMAC's settings are read from the channel's corresponding entry in the CPMEM. The alternate settings can be stored in another entry on the CPMEM. The pointer to the alternate entry on the CPMEM is stored on the physical channel's corresponding IDMAC_SUB_ADDR parameters.

38.4.2.11 IDMAC's modes of operation

38.4.2.11.1 Rotation modes

Rotation is performed by the IDMAC and the Rotation unit inside the IC.

The frame is partitioned into 8X8 pixels blocks. The IC reorders the pixels within a block. The IDMAC reorders the block. The reordering is done according to the ROT, VF & HF parameters in the CPMEM.

The following diagram illustrate various options for reordering of blocks.

- ROTATE means that the ROT bit is set
- HORIZONTAL means that the HF bit is set
- VERTICAL means that the VF bit is set

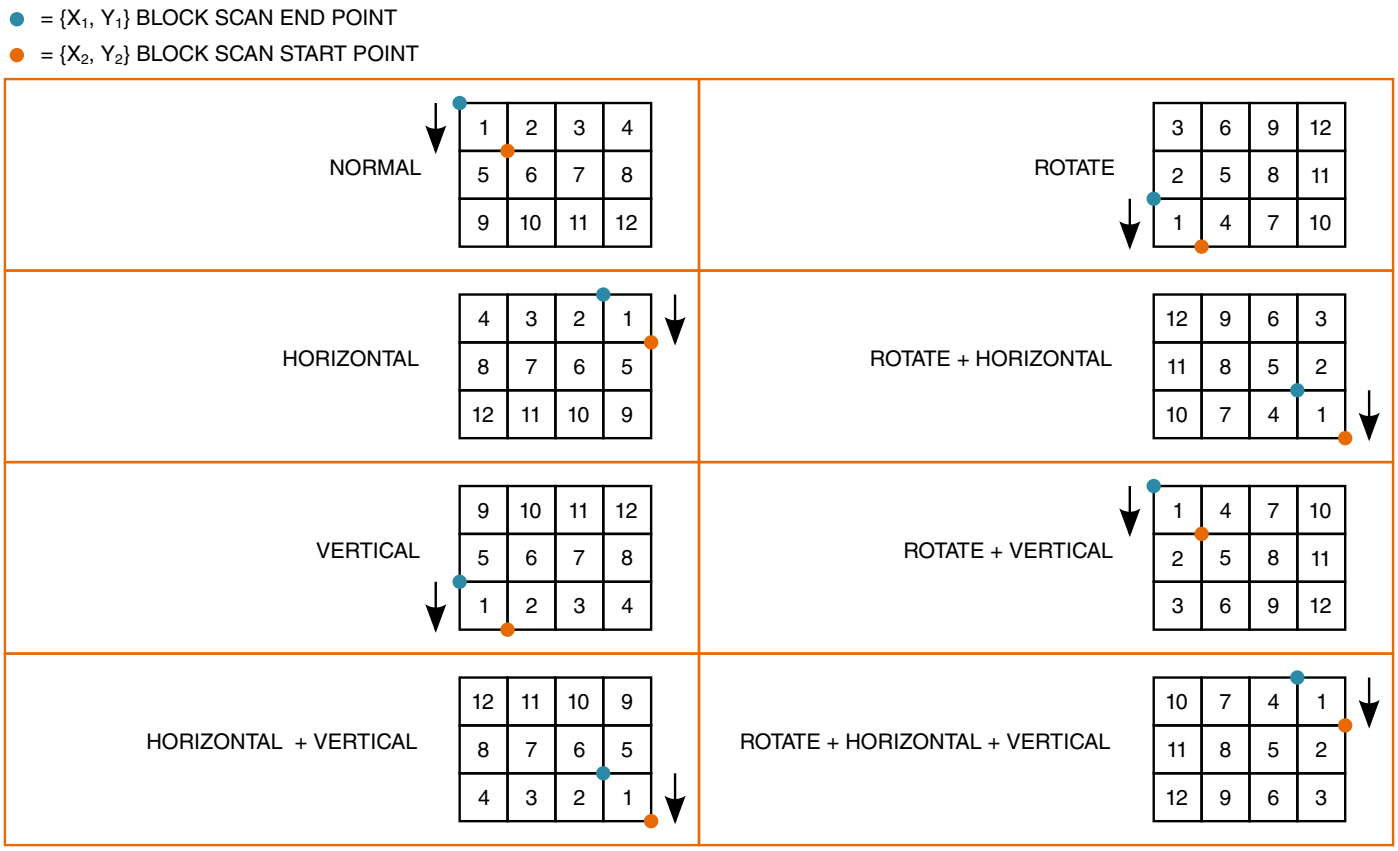


Figure 38-12. Rotation options

38.4.2.11.2 Frame size

The IPU supports various non-interleaved modes; the Frame Height (FH) and Frame Width (FW).

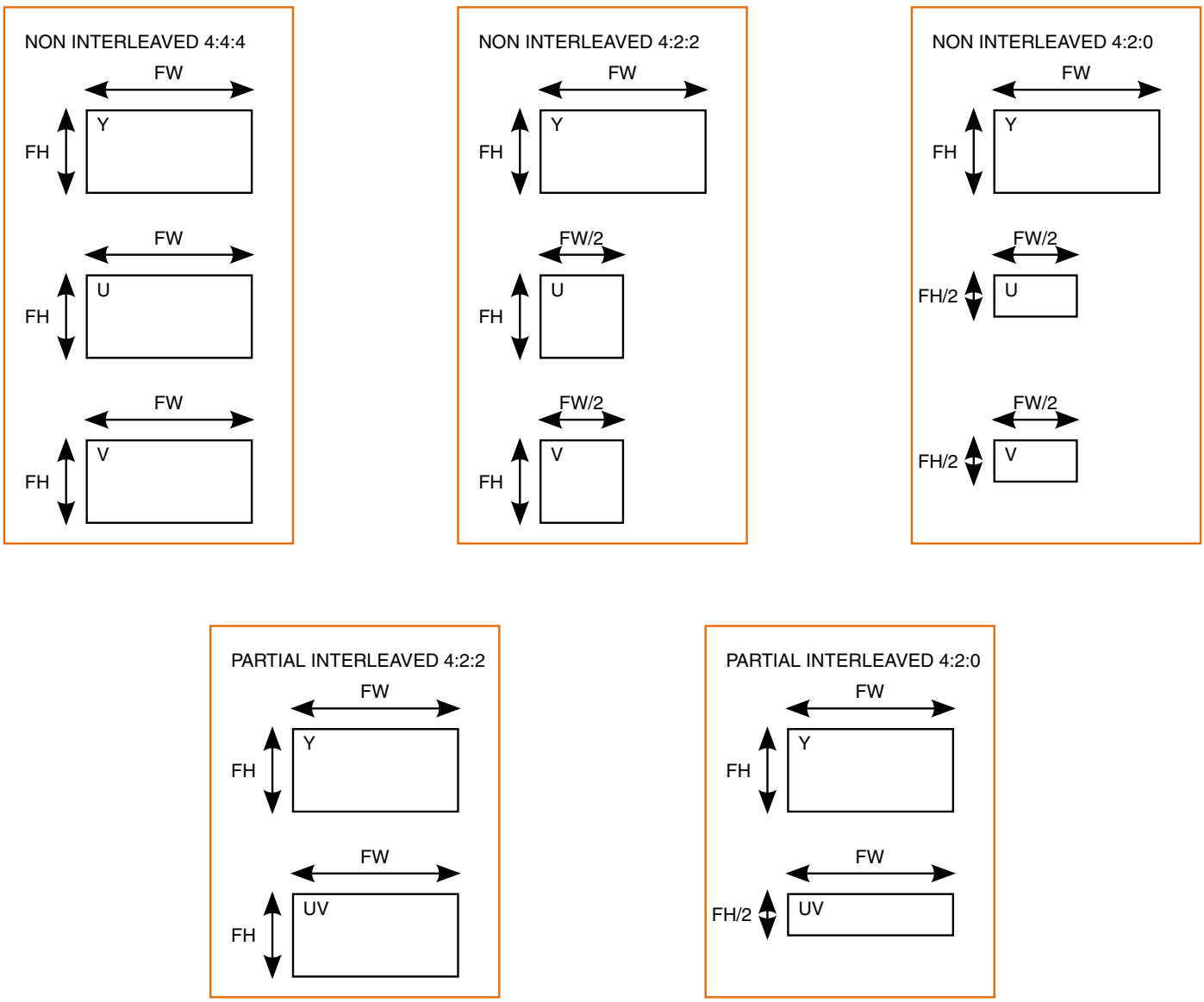


Figure 38-13. Frame size in various modes

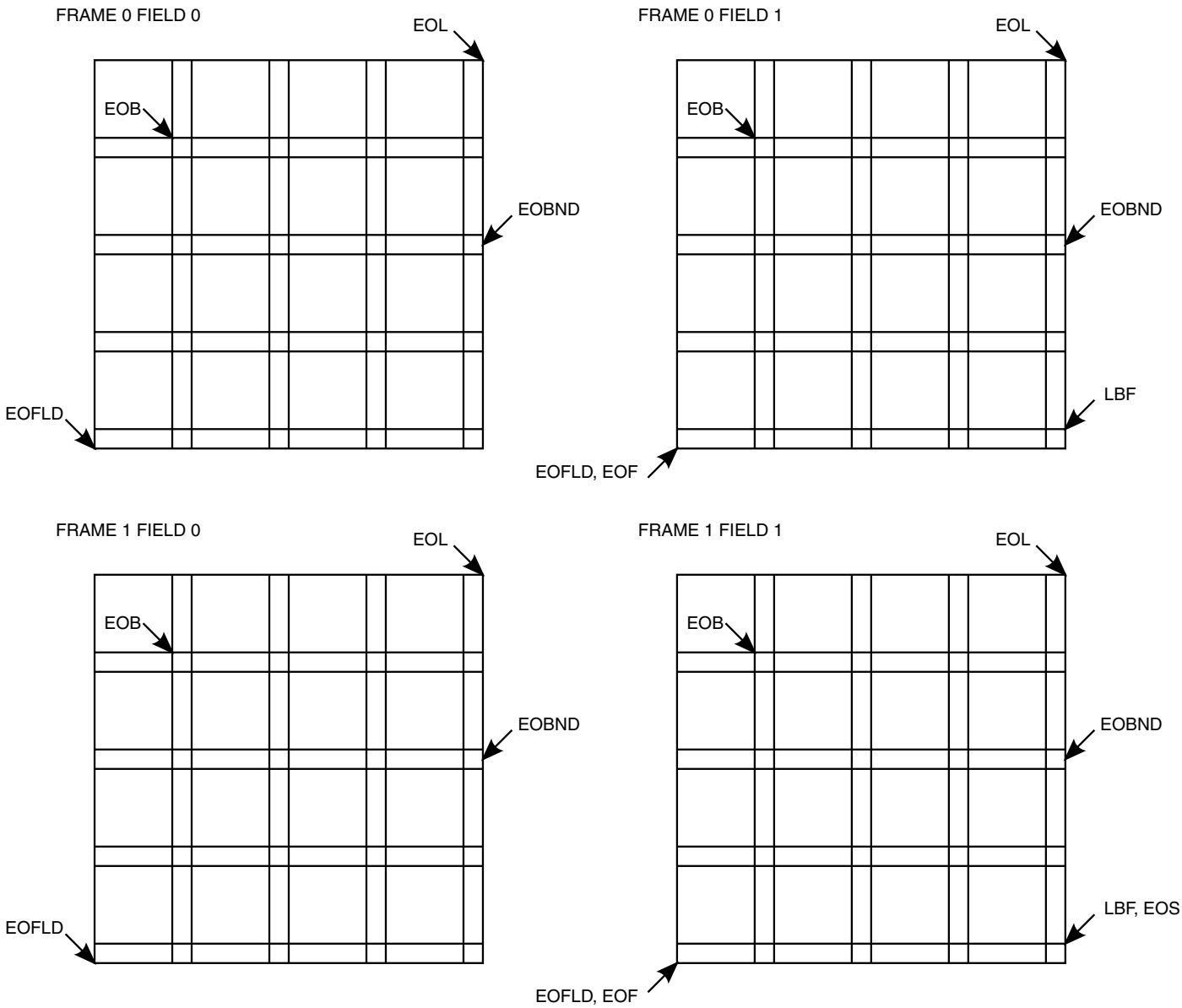


Figure 38-14. Frame's controls

38.4.2.12 IDMAC's restriction

The data must be received from the system's memory through the AXIR interface ("read" direction) "in-order" within a single burst. The entire burst can occur "out-of-order."

38.4.2.13 IDMAC's Endianness support

Byte Endianness - only LE (little endian) is supported

Pixel Endianness - both LE & BE are supported, only for read direction (only 4 BPP case is meaningful, supported only for the "read" direction)

38.4.2.14 IDMAC's internal events

Some of the IDMAC's internal signals can be used for monitor the progress of flows. These bits can be polled by software. Some of these bits can be used to trigger an interrupt or an SDMA event.

The following table describes the available events and their meaning

Table 38-15. IDMAC's internal events

IDMAC event's type	Monitored on	Event's meaning
end of frame	IDMAC_EOF	This is the channel's end of frame indication. This indication is asserted once the entire frame was read/written via the IDMAC. This indication is normally used as an interrupt or SDMA event
new frame acknowledge	IDMAC_NFACK	This indication means that the IDMAC acknowledge the new frame's request from the module. It can be used to track the starting point of a flow or a frame.
new frame before end of frame error	IDMAC_NFB4EOF_ERR	This error indication may indicate on data lost. This indication is asserted when a new frame starts before the completion of the previous frame. For example if a real time input (from camera) was not written properly to the memory due to FIFO full condition.
end of scroll	IDMAC_EOS	end of scroll; This indication is asserted when the scroll counter finished counting its pre defined value
end of band	IDMAC_EOBND	This is the end of band indication. Any time IDMAC complete reading/writing a band it will assert this indication. This is useful to manually control a flow via channels working in band mode.
treshold	IDMAC_TH	Threshold crossing indication. The treshold is defined according to the TH parameter in the IDMAC.
channel busy	IDMAC_CH_BUSY	This signal is asserted when a channel is between NFACK event to EOF event. Negation of these indications is one of the conditions for low power modes handshake.

38.4.3 Camera Sensor Interface (CSI)

The IPU has 2 identical camera sensor interfaces (CSI). The CSI description below refers to a single CSI.

38.4.3.1 CSI Block Diagram

The CSI consists of synchronizer, interface logic, Data packing unit and Sensor Interface Control.

The CSI is controlled via the peripheral bus registers. All programming parameters for the CSI are double buffered with synchronous change at the frame start.

The CSI gets data from the sensor, synchronizes the data and the control signals to the IPU clock (HSP_CLK), and transfer it according to configuration of DATA_DEST register to one or more of the following: IC, SMFC.

This figure shows the CSI Block Diagram.

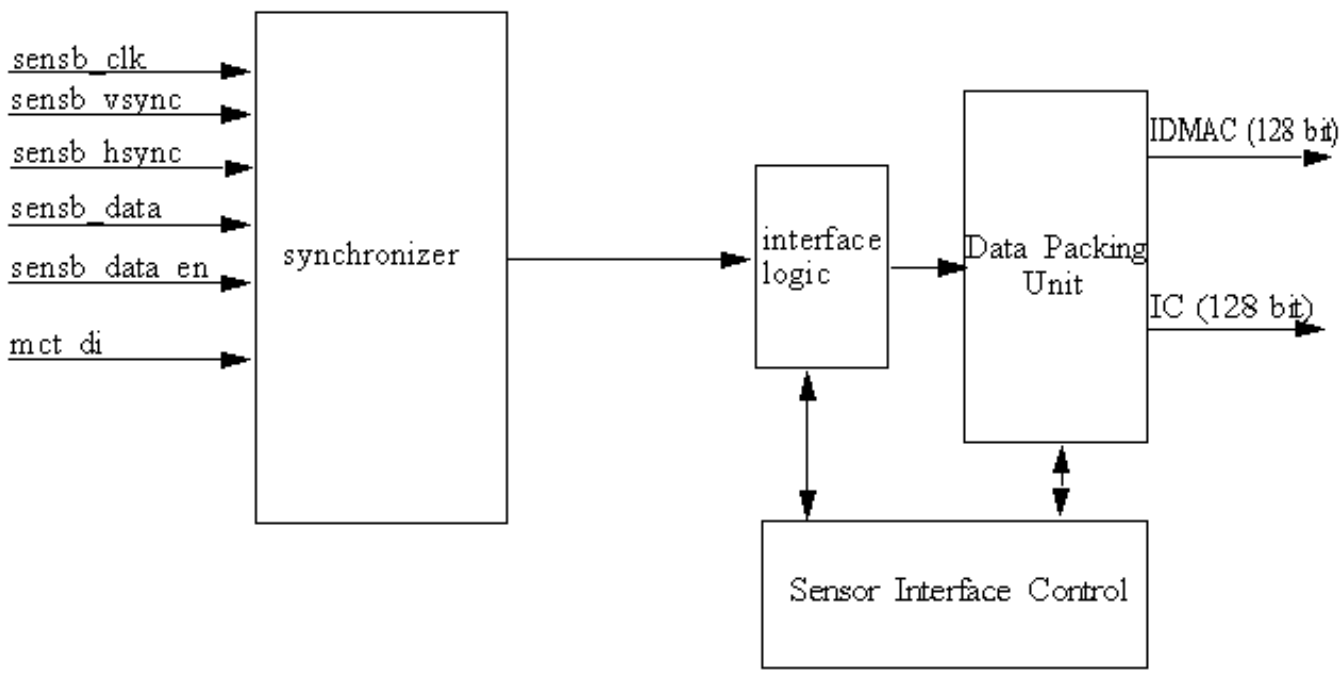


Figure 38-15. CSI Block Diagram

38.4.3.2 CSI Interface

CSI supports two types of interfaces. The interface is determined via the DATA_SOURCE register.

38.4.3.2.1 Parallel Interface

In parallel interface a single value arrives in each clock, except when working in BT.1120 mode, in which two values arrive in each cycle. Each value can be 8-16 bit wide according to configuration of DATA_WIDTH. If DATA_WIDTH is configured to N, then 20-N LSB bits are ignored.

CSI can work with several data formats according to SENS_DATA_FORMAT configuration. In case the data format is YUV, the output of the CSI is always YUV444 (even if the data arrives in YUV422 format).

The polarity of the inputs can be configured using the registers SENS_PIX_CLK_POL, DATA_POL, HSYNC_POL and VSYNC_POL.

38.4.3.2.2 High-speed serial interface - MIPI (Mobile Industry Processor Interface).

In MIPI interface two values arrive in each cycle. Each value is 8 bit wide, which means 16 MSB bits of the data bus input are treated, while 4 LSB bits are ignored.

When working in this mode, the CSI can handle up to 4 streams of data. Each stream is identified with DI (data identifier) that includes the virtual channel and the data type of this stream. Each stream that is handled is defined in registers MIPI_DI0-3. Only the main stream (MIPI_DI0) can be sent to all destination units while the other streams are sent only to the SMFC as generic data.

In this mode SENS_DATA_FORMAT and DATA_WIDTH registers are ignored, since this information is coming to the CSI via the MCT_DI bus.

38.4.3.3 Test Mode

When TEST_GEN_MODE register is configured to 1, the TEST MODE which is a debugging mode, is operated.

The CSI generates the frame by itself and sends it to one of the destination units. The sent frame is a chess board composed of black and configured color squares. The configured color is set with the registers PG_B_VALUE, PG_G_VALUE and PG_R_VALUE. The data can be sent in different frequencies according to the configuration of DIV_RATIO register.

CSI Test Mode requires the following CSIx_SENS_CONF settings:

CSIx_EXT_VSYNC = 0x1 (External VSYNC mode)

CSIx_DATA_WIDTH = 0x1 (8 bits per color)

CSIx_SENS_DATA_FORMAT = 0x0 (Full RGB or YUV444)

CSIx_PACK_TIGHT = 0x0

CSIx_SENS_PRTCL = 0x1 (Non-gated clock sensor timing/data mode)

CSIx_SENS_PIX_CLK_POL = 0x1 Pixel clock is inverted before applied to internal circuitry

CSIx_DATA_POL = 0x0 (Data lines are directly applied to internal circuitry.)

CSIx_HSYNC_POL = 0x0 (HSYNC is directly applied to internal circuitry)

CSIx_VSYNC_POL = 0x0 (VSYNC is directly applied to internal circuitry)

38.4.3.4 Sensor Image Frame Relations

The figure found here illustrates the generalized relations between image frames produced by a sensor and accepted by the CSI.

Generally, four frame definitions exist. The virtual frame A starts with the VSYNC signal. After vertical blanking starts frame B. The HSYNC signal indicates boundaries of the frame B. The frame B includes both the active sensor frame C and horizontal blanking intervals. A size of the blanking intervals depends on sensor type and programming. The size of the frame sent by the sensor (the actual pixels) has to be configured in the registers SENS_FRM_WIDTH and SENS_FRM_HEIGHT. The CSI selects a window (the frame D) inside the frame C by skipping rows and columns according to parameters defined in registers HSC, VSC, ACT_FRM_HEIGHT and ACT_FRM_WIDTH. frame D is sent to the rest of the IPU.

NOTE

The following limitation must exist:

SENS_FRM_HEIGHT >= VSC + ACT_FRM_HEIGHT

SENS_FRM_WIDTH >= HSC + ACT_FRM_WIDTH

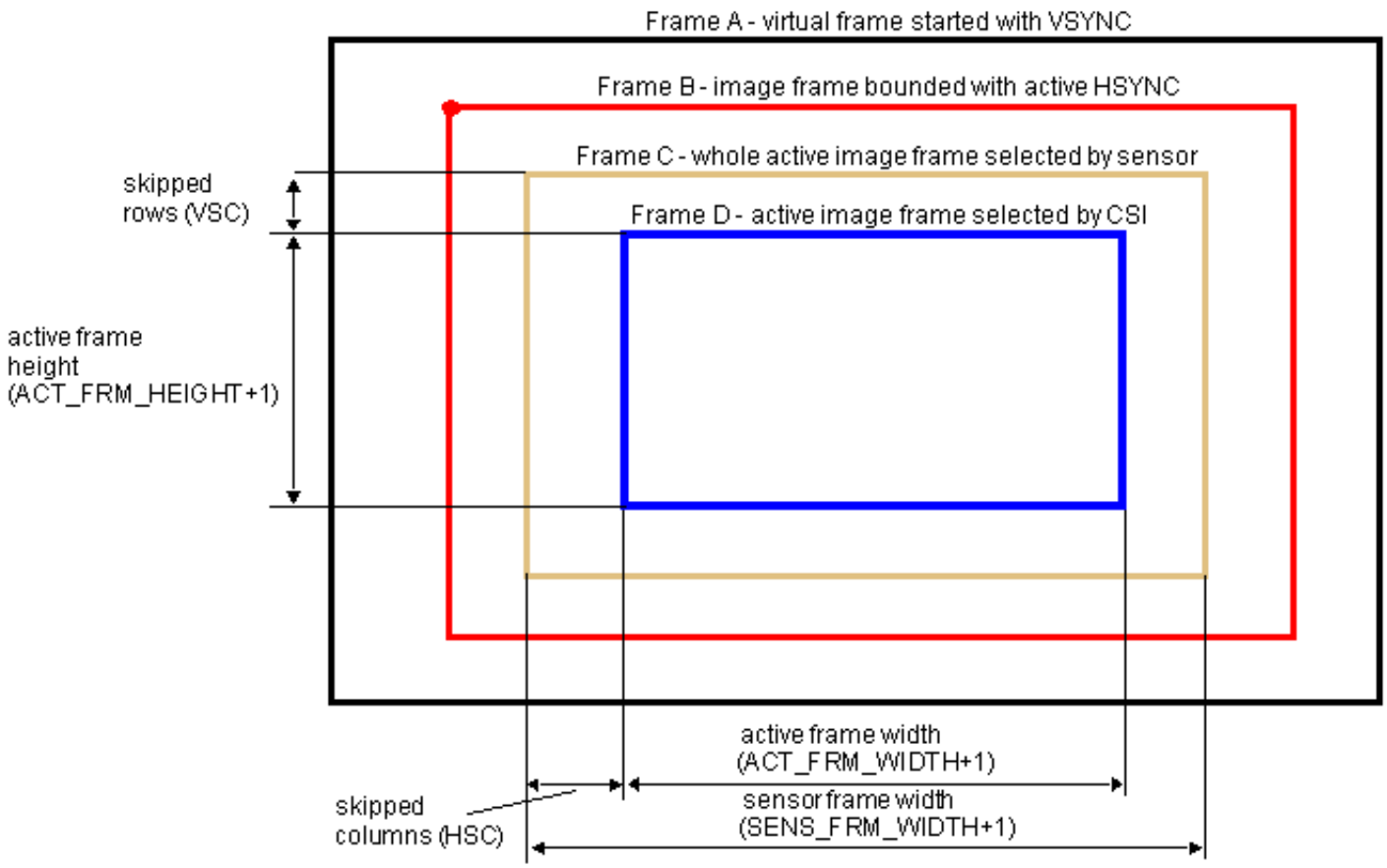


Figure 38-16. Sensor Image Frames

38.4.3.5 Companding

Companding is performed as follows:

input: 10-bit unsigned number.In case that component's size is bigger than 10 bit, 10 MSB bits are taken. In case that component's size is less than 10 bit, color extraction is performed.

First step: shift:

x-> clip(x + offset, 0, 1023)

where the offset is a 10 bit signed number that is configured in CPD_OFFSET1 and CPD_OFFSET2 registers.

second step - piecewise-linear map:

y = Min[255, (y1[k]+(((x-x1[k])*slope[k])>>6 + 1))>>1]

where:

- the input range 0..1023 is partitioned to 16 equal segments: x1[k] .. x1[k+1] - 1 where x1[k] = 0, 64, 128, 192, 256, .. . 960.

- The linear map, in each segment, is characterized by y1[k] (9-bit unsigned number) and slope[k] (8-bit unsigned number).

Each destination unit can get the data after being companded depending on the configuration of CPD register. If this register is configured to 3'h0, the compander units are disable in order to save power. Parameters of the companding are equal for all destinations and can be set in the CPD registers.

2 companding units are located in the CSI. This is because when working in MIPI orBT. 1120 modes 2 components arrive in each cycle and the companding for each 2 components has to be in parallel. When CSI works in other mode, only one compander is needed, so the other one is disabled to save power.

38.4.3.6 Timing/Data mode protocols

CSI can work in several timing/data mode protocols, according to SENS_PRTCL configuration.

38.4.3.6.1 Gated Mode

In this mode, VSYNC is asserted at the beginning frame. HSYNC is asserted at the beginning of a row and remains asserted until the end of the row. The sensor clock is ticking all the time.

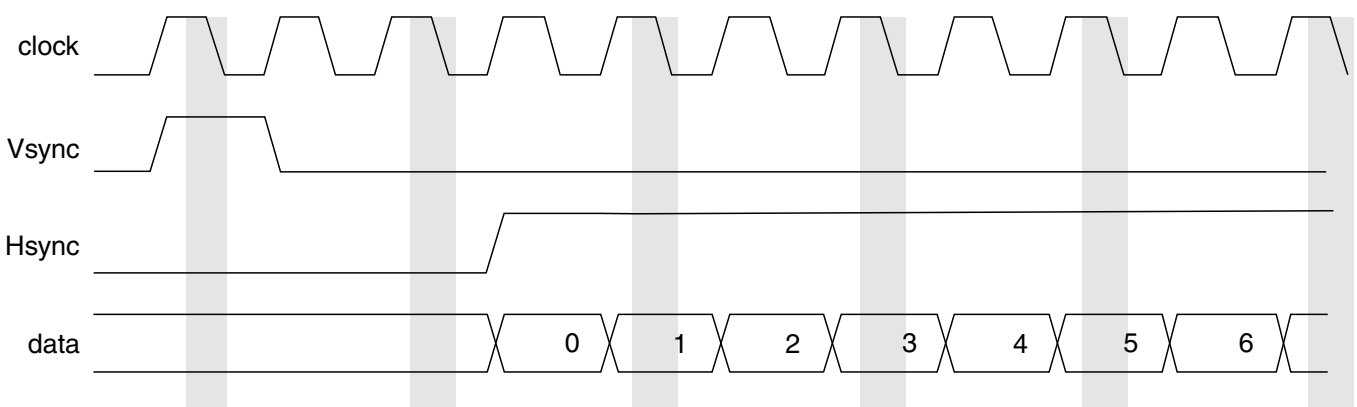


Figure 38-17. gated mode

38.4.3.6.2 Non-Gated Mode

In this mode VSYNC is used to indicate beginning of a frame. Sensor clock is ticking only when data is valid. HSYNC is not used.

When working with MIPI, the non-gated mode should be configured.

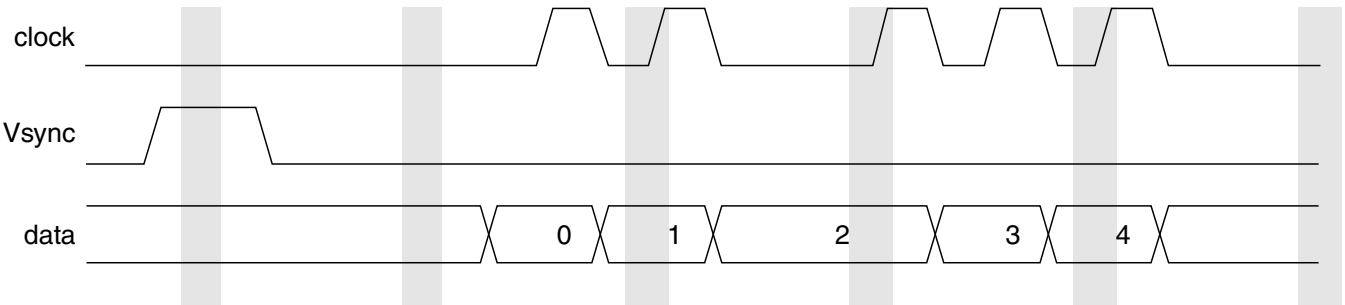


Figure 38-18. non-gated mode

38.4.3.6.3 BT.656 mode

In this mode the CSI works in compliance with recommendation ITU-R BT.656. The timing reference signals (frame start, frame end, line start, line end) are embedded in the data bus input. Each timing reference signal consists of a four word sequence. The first three words are fixed and configured in the CCIR_PRECOM register. The fourth word contains information defining field, the state of field blanking and the state of line blanking. these states are configured in registers CCIR_CODE_1 (for field 0) and CCIR_CODE_2 (for field 1).

In this mode in each cycle one value of data arrives

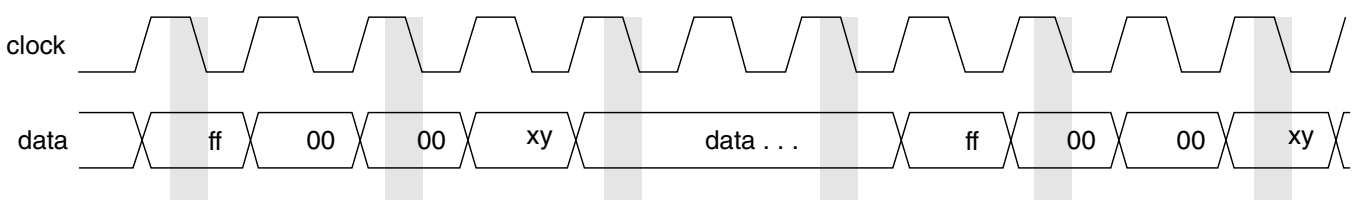


Figure 38-19. BT.656 mode

In this diagram the first three words are 0xff, 0x00, 0x00. The fourth word is XY and it includes the timing reference.

38.4.3.6.4 BT.1120 mode

In this mode the CSI works in compliance with recommendation ITU-R BT.1120. The timing reference signals (frame start, frame end, line start, line end) are embedded in the data bus input. Each timing reference signal consists of a four word sequence. The first three words are fixed and configured in the CCIR_PRECOM register. The fourth word contains information defining field, the state of field blanking and the state of line blanking. these states are configured in registers CCIR_CODE_1 (for field 0) and CCIR_CODE_2 (for field 1).

In this mode, the CSI can also work in DDR mode - data arrives on every edge of the clock. In addition, in each cycle two value of data arrive.

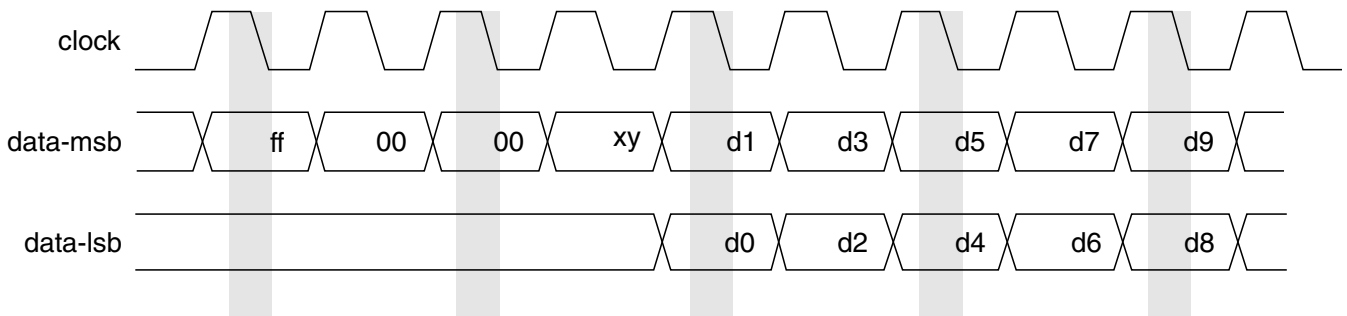


Figure 38-20. BT.1120 mode - SDR mode

In the above diagram each data arrives with the positive edge of the clock.

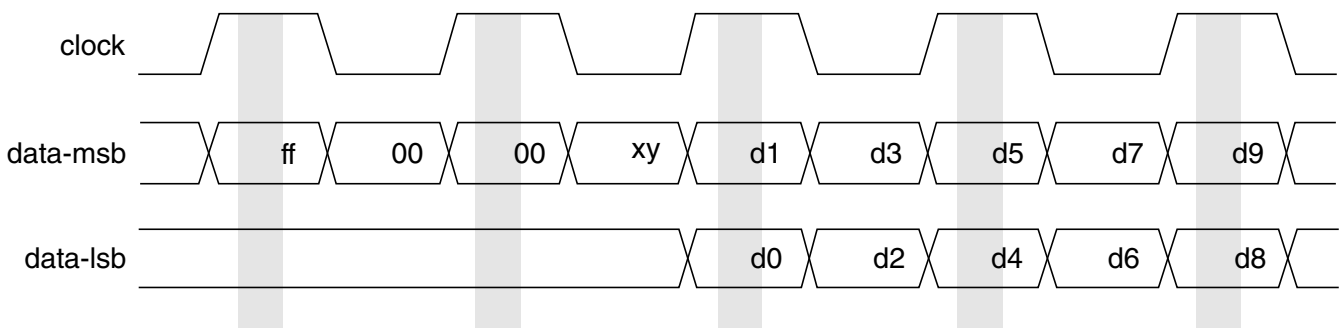


Figure 38-21. BT.1120 mode - DDR mode

In the above diagram, data arrives in the positive edge and the negative edge of the clock.

38.4.3.7 Packing to memory

The data bus output to the SMFC is 128-bit.

The following table shows the how the CSI performs the packing before sending the data to the SMFC.

The data bus output to the SMFC is 128-bit. The following table shows the how the CSI performs the packing before sending the data to the SMFC.

Table 38-16. Packing Unit

data format	component size	companded	regular packing	tight packing
Bayer, Generic data, JPEG	8	{8DC[1] ,8DC,8DC,...,8DC}	{8D,8D,8D,...,8D[2] }	NA
	9-16	{8DC,8DC,8DC,...,8DC}	{16DE[3] ,16DE,...,16DE}	NA
RGB, YUV	8	{8DC,8DC,8DC,8R}	{8D,8D,8D,8R}	NA
	9-10	{8DC,8DC,8DC,8R}	{16DE,16DE,16DE,16R[4] }	{10DE,10DE,10DE,2R}
	11-16	{8DC,8DC,8DC,8R}	{16DE,16DE,16DE,16R}	{10DT[5] ,10DT,10DT,2R}

1. DC - data after being companded
2. D - data arrived from sensor.
3. DE - data after being extended.
4. R- reserved bits
5. DT - data after being truncated.

The tight packing functionality is enabled when PACK_TIGHT register is set. It is only used when data format is RGB or YUV and the data width is bigger than 8.

38.4.3.8 Skipping frames

Some of the frames that are sent to the SMFC can be skipped. Skipped frames are ignore by the CSI and are not sent to the corresponding unit.

Using SKIP_SMFC and MAX_RATIO_SKIP_SMFC registers the user can define the frames for the SMFC that will be skipped.

38.4.3.9 16 bit camera support

Devices that support 16 bit data bus can be connected to the CSI. This can be done in one of the following ways.

16 bit YUV422

In this mode the CSI receives 2 components per cycle. The CSI is programmed to accept the data as 16 bit generic data. The captured data will be stored in the memory through the SMFC. The IDMAC needs to be programmed to store 16bit generic data. When the data is read back from the memory for further processing in the IPU it will be read as YUV422 data.

16 bit RGB as generic data

In this mode the CSI receives 3 components per cycle. If the external device is 24bit - the user can get connect a 16 bit sample of it (such as RGB565) The CSI is programmed to accept the data as 16 bit generic data. The captured data will be stored in the memory through the SMFC. The IDMAC needs to be programmed to store 16bit generic data. When the data is read back from the memory for further processing in the IPU it will be read as 16 bit RGB data. The IDMAC's mapping unit will be used to remap the 16 bit data to the internal 24bpp RGB format In this mode on the fly processing is can't be performed. The data has to be sent to the memory first and then further processed by the IPU.

16 bit RGB565

This is the only mode that allows on the fly processing of 16 bit data. In this mode the CSI is programmed to receive 16 bit generic data. In this mode the interface is restricted to be in "non-gated mode" and the CSI#_DATA_SOURCE bit has to be set If the external device is 24bit - the user can connect a 16 bit sample of it (RGB565 format). The IPU has to be configured in the same way as the case of CSI#_SENS_DATA_FORMAT=RGB565

38.4.3.10 CSI Restrictions

The frequency of the sensor clock must not be greater than the IPU clock (HSP_CLK)

SENS_FRM_HEIGHT >= VSC + ACT_FRM_HEIGHT

SENS_FRM_WIDTH >= HSC + ACT_FRM_WIDTH

38.4.4 Sensor Multi FIFO Controller (SMFC)

The Sensor Multifile Controller used as buffer between CSI and IDMAC. Two masters (CSIs) can be connected to SMFC. Both masters can be active simultaneously.

Each master can send up to 4 frames, distinguished by csi_id bus. The frame can be mapped to one of four IDMAC channels via SMFC mapping registers. Each DMA channel have dedicated FIFO.

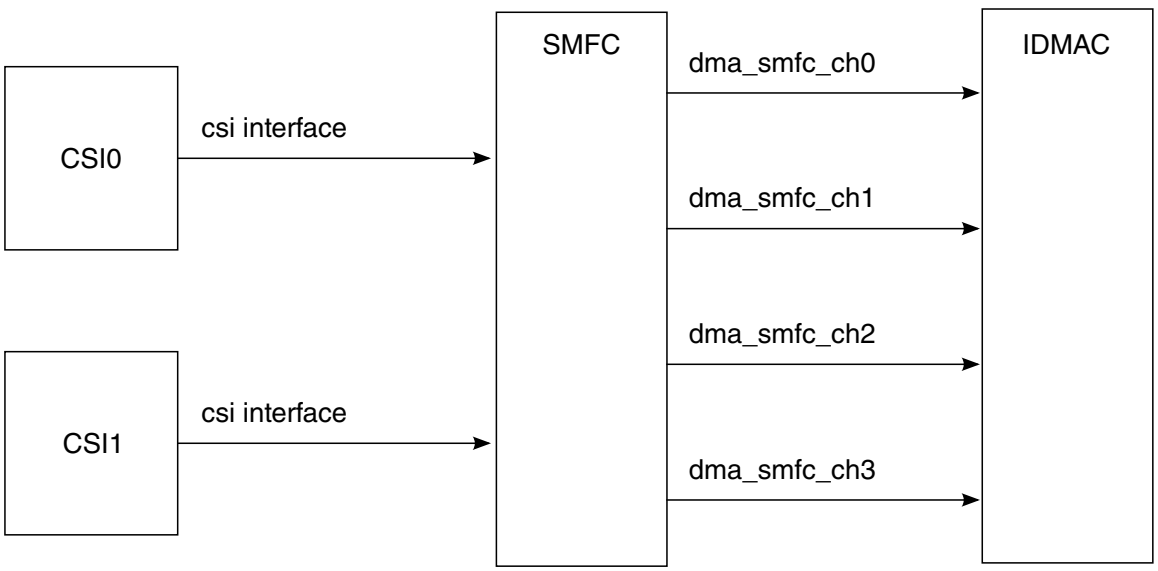


Figure 38-22. SMFC data flow

38.4.4.1 SMFC's Features

- Support two CSI masters and four DMA channels
- Automatic FIFO size calculation

38.4.4.2 SMFC's Functional description

SMFC supports up to four DMA channels. Each channel has a dedicated FIFO controller, as shown in the following figure. Sampled data and frame ID are kept in the buffers until the buffer is selected by Round Robin Priority Mechanism. Then, the content of ID buffer is compared to CH#_MAP bits and the corresponding FIFO controller is activated. As a result, the content of the buffer is copied to the RAM. The wptr and the "base" are used to calculate the location in the RAM. The rptr are following after wptr during dma_active signal, initiated by DMA.

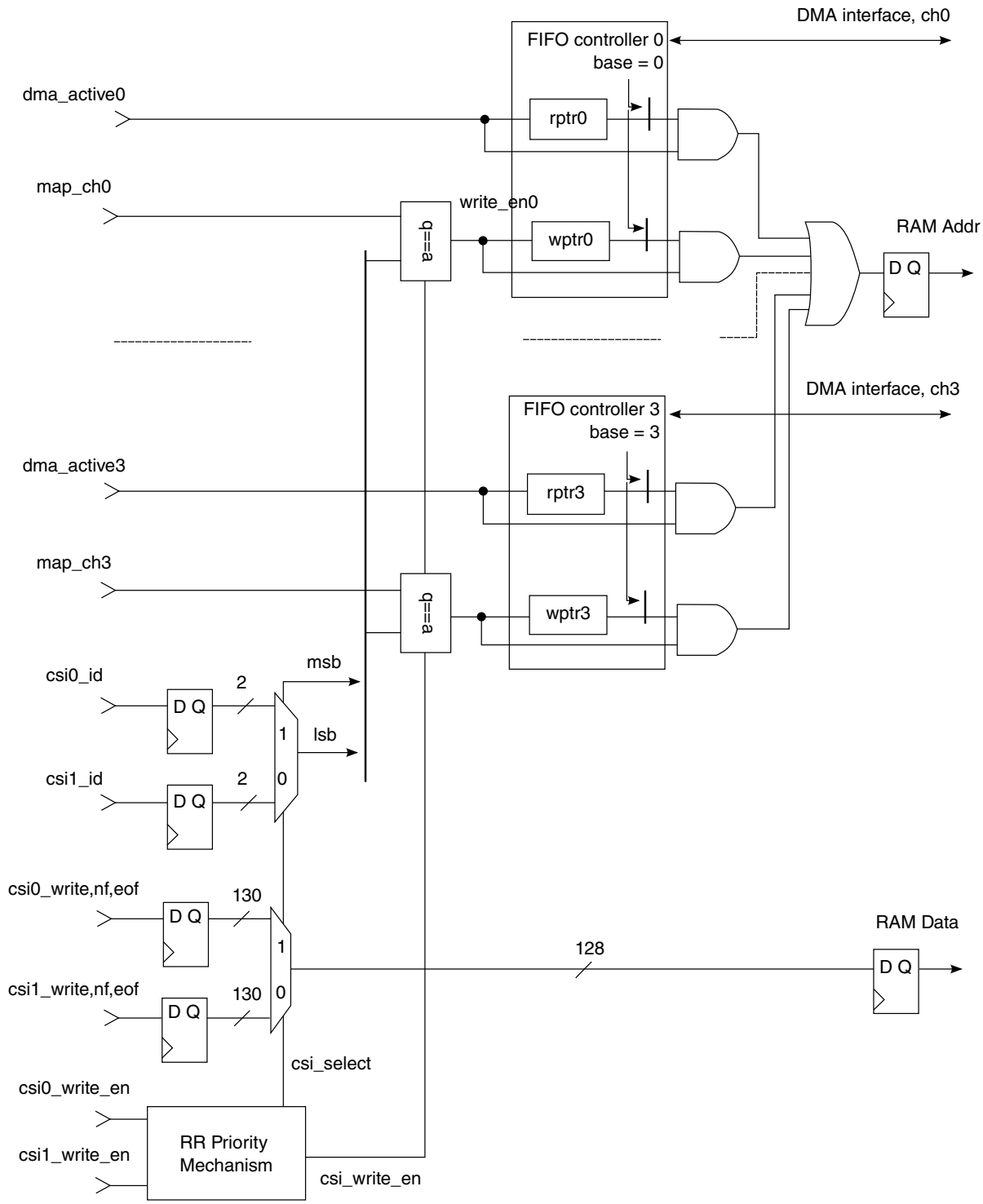


Figure 38-23. Sensor Multi FIFO Controller

Four FIFOs are implemented with single RAM due to the fact that the channels can't be active simultaneously. The memory space of SMFC is divided into four equal sectors. Each FIFO has a fixed base address - "base". The "base" used as MSB for corresponding pointer in order to calculate absolute address of the RAM. FIFO size of channels 1 and 3 are fixed and is equal to size of one sector. FIFO size of channels 0 and 2 depend from other channels as shown in the table below. Other configurations are not allowed.

Table 38-17. FIFO Channels

Number of DMA channels required	enable/disable of channels 3,2,1,0	FIFO size (sectors) per channels 3,2,1,0
1	0 0 0 1	0 0 0 4
2	0 1 0 1	0 2 0 2
3	1 1 0 1	1 1 0 2
4	1 1 1 1	1 1 1 1

NOTE

Channels should not be enabled after activation of channel 0 or/and channel This can cause to overlapping of FIFO areas and other malfunctions.

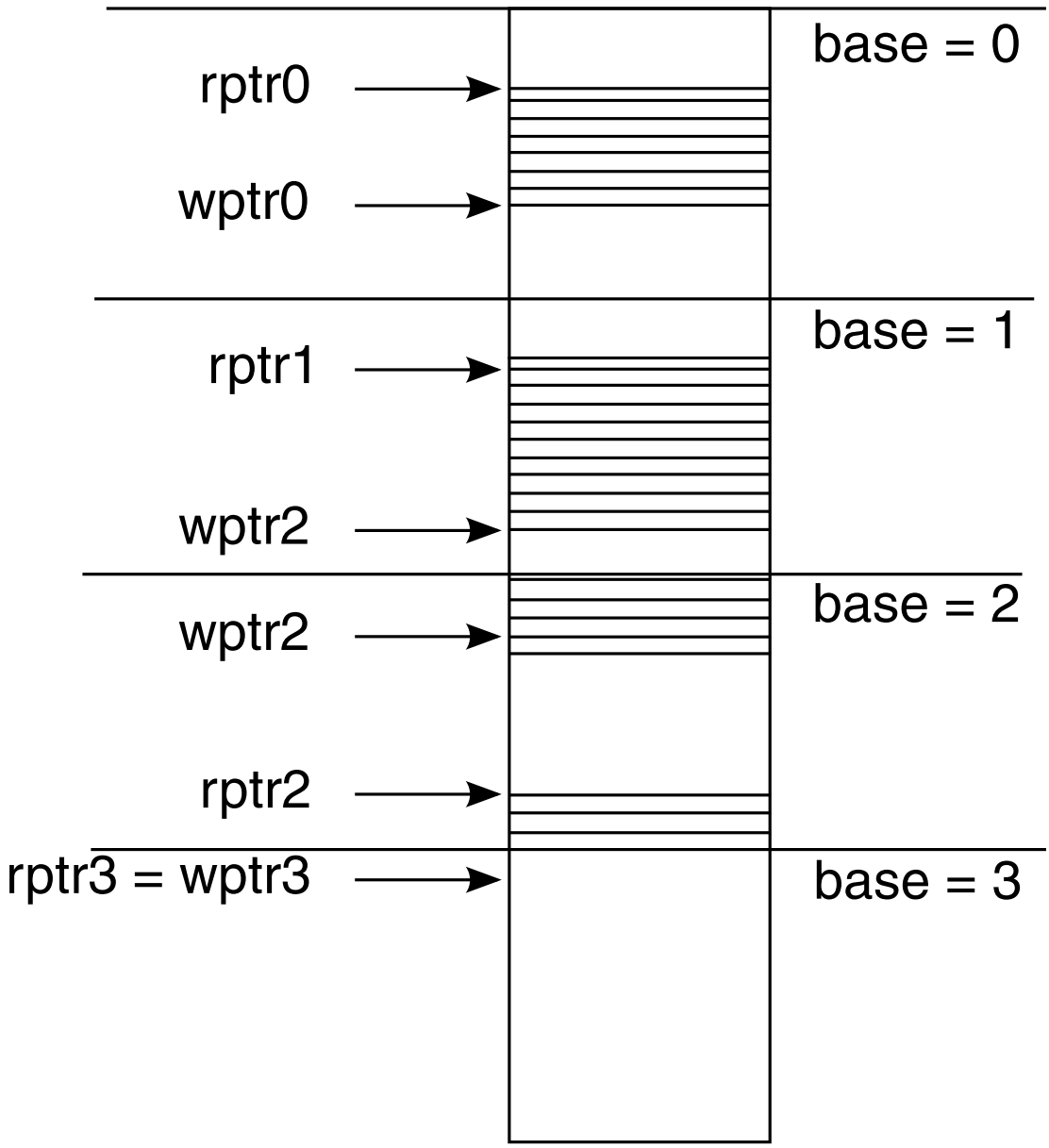


Figure 38-24. SMFC memory map when DMA channels 3,2,1,0 are enabled

38.4.4.2.1 SMFC Master interface.

SMFC Master interface is shown in the following figure.

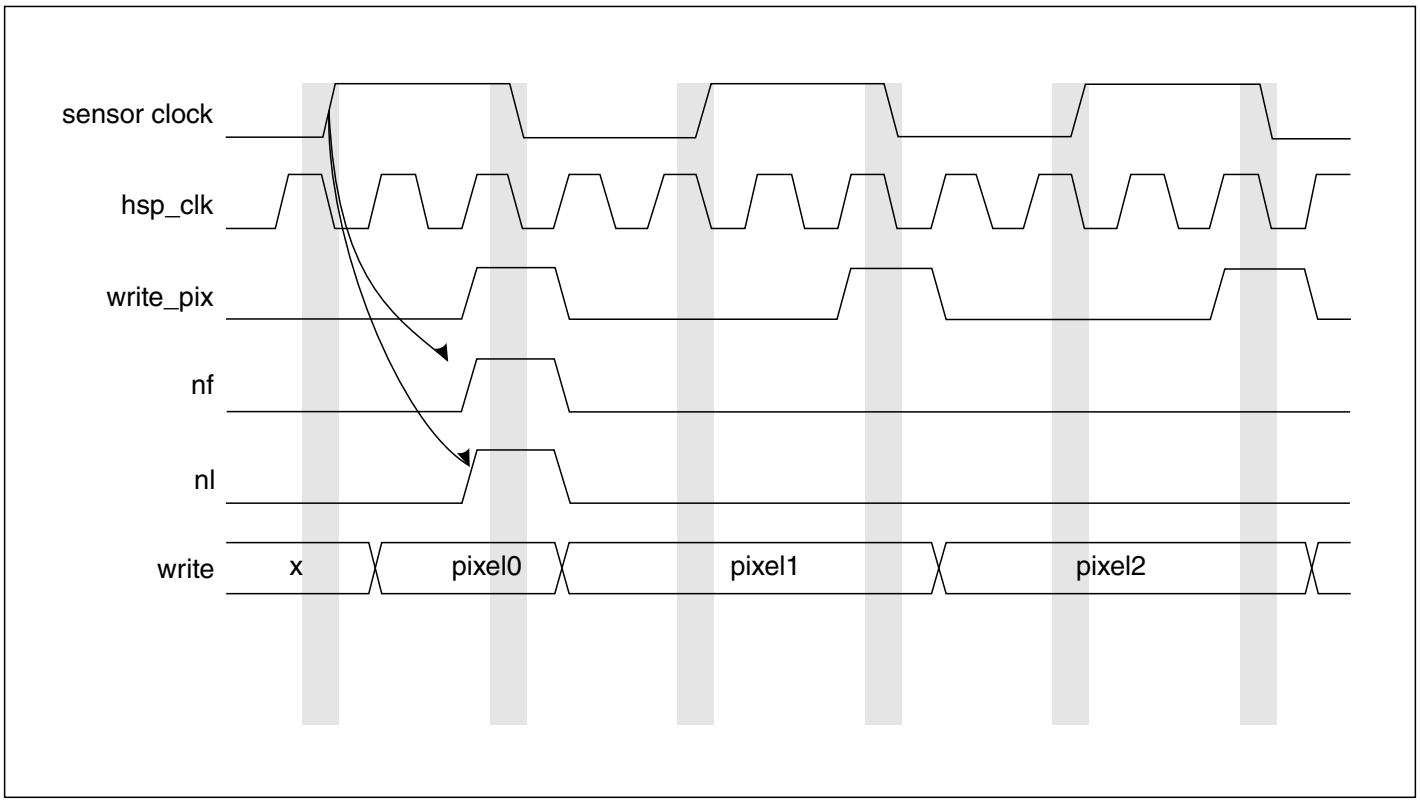


Figure 38-25. Timing diagram of SMFC slave interface

Sensor clock can be asynchronous to IPU clock (hsp_clk). The CSI synchronize data arrived from the sensor to hsp_clk. The csi_write signal indicates when data on csi_pix bus can be sampled by hsp_clk clock.

38.4.4.2.2 Restrictions

1. DMA channels should not be enabled after activation of channel 0 or/and channel 2. This can cause to overlapping of FIFO area and other malfunctions.
2. Watermark set value should be greater than watermark clear value.
3. One frame should not be mapped to few DMA channel.

38.4.5 Image Converter

38.4.5.1 IC Block Diagram

The IC contains three processing sections: downsizing, main processing and rotation.

The block is controlled via the peripheral bus registers. Some processing parameters should be written by the ARM core to the Task Parameter Memory. Writing to the memory is performed via the AHB bus.

The IC Block Diagram is shown in the following figure.

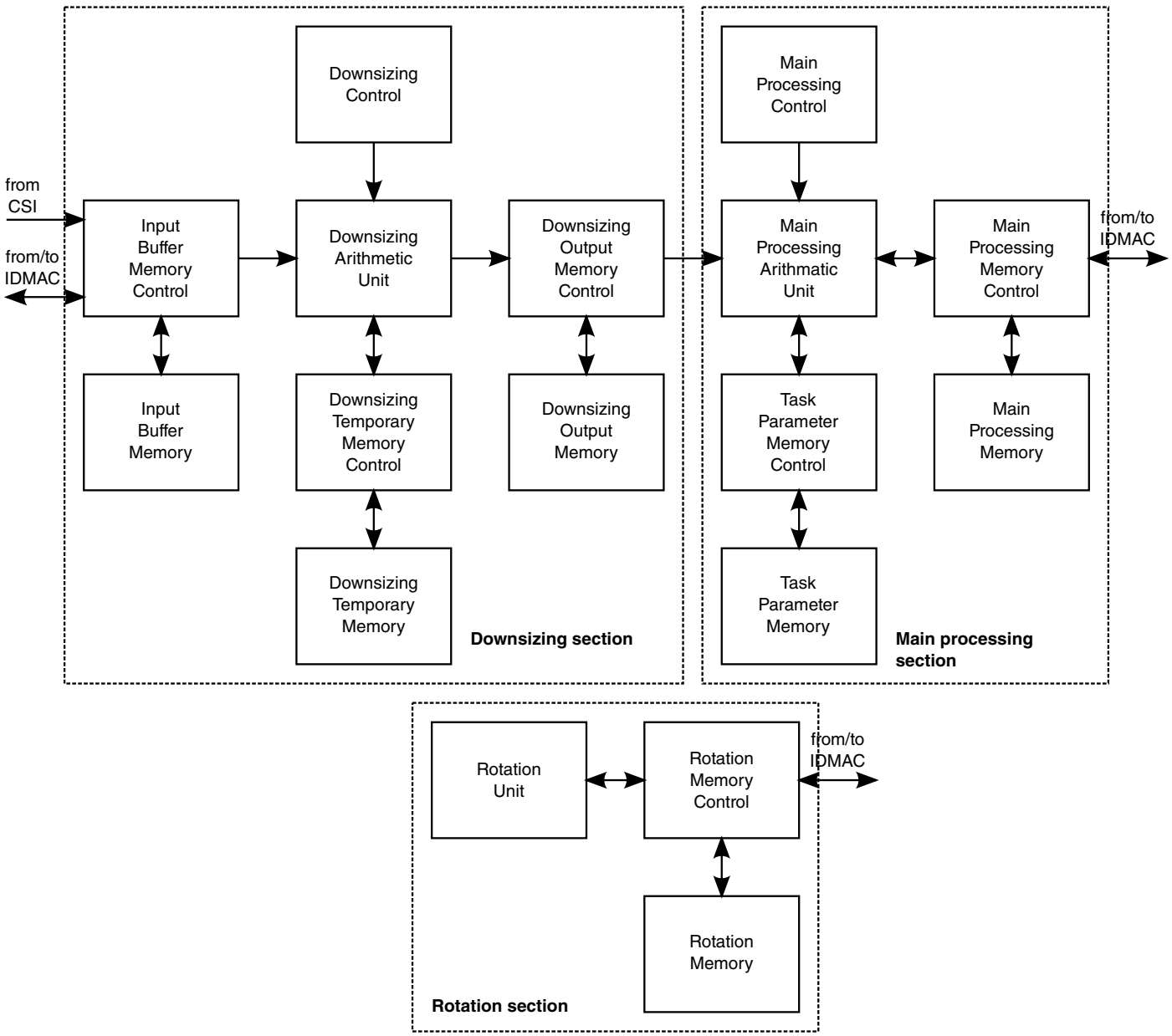


Figure 38-26. IC Block Diagram

38.4.5.2 Processing tasks

Each of the three processing sections performs up to three processing tasks with time sharing:

1. Preprocessing task for encoding.
2. Preprocessing task for displaying image from sensor (viewfinder).

3. Postprocessing task.

The tasks are performed by single hardware. The ARM platform configures each task before enabling it.

Tasks switching is transparent for the ARM platform. The time unit for task switching in the downsizing section corresponds to a processing time of one burst of eight pixels, in the main processing section - to a processing time of one image line, in the rotation section - to a processing time of one image frame.

All three tasks include similar operations controlled by commands. Task configuring consists in definition of commands for each task, as described in the following table.

Table 38-18. Task Commands

Command code	Command	Processing unit	Command parameters	Description
EN	Task Enable	DSU, MPU		Task will enabled from next frame. Task 1 is preprocessing for encoding. Task 2 is preprocessing for viewfinder. Task 3 is postprocessing.
	Downsizing	DSU	Downsizing ratio (GCR)	Downsizing ratio 1:1, 2:1, 4:1
	Resizing	MPU	Resizing ratio (GCR)	Resizing ratio from 2:1 to 1:M Resizing ratio = N:M; M =2^13; N = floor(M*(SI-1)/(SO-1)); SI - input size; SO - output size
CSC1	Color space conversion 1	MPU	Color conversion coefficients and offsets (TPM)	Color conversion matrix 1
GLOB_A	Global alpha	MPU		Used only for Task 2 and Task 3
CMB	Combining	MPU		Combining video with graphics. Used only for Task2 and Task3.

The ARM platform writes the commands to the IC_CONF Register. Because there is no double buffering for all the IC parameters, the ARM platform must disable a task before changing its parameters. After being disabled, the task is still allowed to complete its current frame execution. At frame finish, the IPU sends an interrupt to the ARM platform indicating that the ARM platform can change task parameters. The ARM platform enables the task again and task execution is resumed from start of the next frame.

38.4.5.3 Downsizing Section

The sensor data from the CSI is written into a FIFO located in the Input Buffer Memory. Depending on programmed processing flow, the FIFO data can be sent to the system memory via the IDMAC or straight forward to the Downsizing Unit.

In the first case, the data is processed by the ARM platform and returned by the IDMAC to another FIFO located in the Input Buffer Memory.

For postprocessing, the IDMAC transfers a data from the system memory to the third FIFO. The data is read by the Downsizing Unit when the postprocessing is performed.

Each or three FIFOs has 128 pages. Each page can store one burst of 2 or 4 words which corresponds to 8 or 16 pixels. The size of the burst is defined according to the channel's corresponding CB#_BURST_SIZE bit. The memory word width is 128 bits. Each memory word contains color components of four adjacent pixels or 16 bytes of generic data (e.g. Bayer). Access to the FIFOs is controlled by the Input Buffer Memory Control.

The Downsizing Unit performs averaging and decimation of image pixels both in horizontal and vertical directions according to the following equations:

$$HP_{R,c} = \frac{1}{DS_R_H} \sum_{k=0}^{DS_R_H-1} IP_{r+k,c}$$

$$VP_{R,C} = \frac{1}{DS_R_V} \sum_{l=0}^{DS_R_V-1} HP_{R,c+l}$$

where IPr,c - the input pixel, HPR,c - the pixel after horizontal downsizing, VPR,C - the pixel after vertical downsizing, DS_R_H and DS_R_V - the horizontal and vertical downsizing ratios according to the IC_PRP_ENC_RSC, IC_PRP_VF_RSC and IC_PP_RSC Registers. The final calculation result is rounded to 8 bits.

Each of three downsizing tasks processes the data by bursts of 8 pixels. Normally, the current task runs until emptying the corresponding input FIFO. After finishing burst processing, the Downsizing Unit may switch between the current task and another task with higher priority, if the Input Buffer Memory has received a burst for this new task.

Averaging is performed firstly in the horizontal direction. All color components of a pixel are processed in parallel. After horizontal averaging has finished for a single output pixel, the new pixel value is added to the corresponding pixel value of a temporary row derived from previous averaging steps. This provides vertical averaging of the pixels. The temporary row is stored in the Downsizing Temporary Memory. The memory word width matches one accumulated pixel width (36 bits). There are three temporary rows stored in this memory - one per downsizing task.

After vertical averaging has been finished, the output row is written to the Downsizing Output Memory. The memory word of 48 bits includes two output pixels. For each task, the memory has a double buffer of one row. When the Downsizing Unit fills the foreground part of the double buffer, the Main Processing Unit takes pair or pixels from the background part. After the new downsized row has been ready, the foreground and background memory pointers are swapped.

38.4.5.4 Main Processing Section

The Main Processing Unit reads pairs of pixels from the Downsizing Output Memory background part. It processes the complete pixel row for the current task and after that switches to another task if the input data for this new task is ready.

For each task, the Main Processing Unit is able to perform the following sequence of operations:

1. Horizontal flipping the image (optional) performed with reading from the Downsizing Output Memory. Flipping is enabled via the VF, HF & ROT parameters of the corresponding DMA channels (Table 38-13 and Table 38-14) responsible for output of the task results. The preprocessing task for encoding uses the VF, HF & ROT parameters from IDMAC channel #20, the preprocessing task for viewfinder - from the IDMAC channel #21, the postprocessing task - from the IDMAC channel #22.
2. Horizontal resizing by bilinear interpolation between two adjacent pixels received from the Downsizing Output Memory according to the equation:

$$HP_{R,c} = IP_{r,c} + RS_C_H \cdot (IP_{r+1,c} - IP_{r,c})$$

where RS_C_H - the current horizontal resizing coefficient. The calculation result is rounded to 8 bits. The resizing coefficient is calculated as

$$RS_C_H = \left(\sum_{k=0}^{R-1} RS_R_H \right) \bmod(8196)$$

where RS_R_H - the horizontal resizing ratio from the IC_PRP_ENC_RSC, IC_PRP_VF_RSC and IC_PP_RSC Registers. The RS_R_H parameter is equal to a numerator N of the resizing ratio N:M with M =2^13.

The resulting row of the horizontal resizing is stored in the Task Parameter Memory.

3. Vertical resizing by bilinear interpolation between the current and previous results of horizontal resizing. Both current and previous results of horizontal resizing is stored in the Task Parameter Memory. Resizing is accomplished according to the equation:

$$VP_{R,C} = HP_{R,c} + RS_C_V \cdot (HP_{R,c+1} - HP_{R,c})$$

where RS_C_V - the current vertical resizing coefficient. The calculation result is rounded to 8 bits. The resizing coefficient is calculated as

$$RS_C_V = \left(\sum_{k=0}^{C-1} RS_R_V \right) \bmod(8196)$$

where RS_R_V - the horizontal resizing ratio from the IC_PRP_ENC_RSC, IC_PRP_VF_RSC and IC_PP_RSC Registers. The RS_R_V parameter is equal to a numerator N of the resizing ratio N:M with M =2^13.

At completion of vertical resizing, this row is updated - the current result of horizontal resizing replaces the previous one.

4. First color space conversion YUV to RGB or RGB to YUV with the conversion matrix CSC1. The conversion matrix coefficients are programmable. They are stored in the Task Parameter Memory. The conversion equations are:

$$Z_0 = 2^{SCALE-1} \cdot (X_0 \cdot C_{00} + X_1 \cdot C_{01} + X_2 \cdot C_{02} + A_0)$$
$$Z_1 = 2^{SCALE-1} \cdot (X_0 \cdot C_{10} + X_1 \cdot C_{11} + X_2 \cdot C_{12} + A_1)$$
$$Z_2 = 2^{SCALE-1} \cdot (X_0 \cdot C_{20} + X_1 \cdot C_{21} + X_2 \cdot C_{22} + A_2)$$

where for YUV to RGB: X_0=Y, X_1=U, X_2=V, Z_0=R, Z_1=G, Z_2=B, for RGB to YUV: X_0=R, X_1=G, X_2=B, Z_0=Y, Z_1=U, Z_2=V.

All the parameters of the conversion matrix are written by the ARM platform to the Task Parameter Memory (IC Task Parameter Memory). The final calculation result is limited according to the SAT_MODE parameter and rounded to 8 bits.

5. Combining video with graphics. There are the following combining options:
   - local alpha blending,
   - global alpha blending,
   - use of key color.

   If both alpha blending and color keying are enabled, color keying has higher priority (graphic pixels of the key color are fully transparent independently on the alpha value).

   Combining mode is selected via the IC_CONF Register. The combining equation is:

$$OP = IGP \cdot \alpha + IVP \cdot (1 - \alpha)$$

   where IGP - an input graphics pixel, IVP - an input video pixel, α=(A+floor(A/128))/256 - an alpha value, A - a global or local transparency parameter. The global A is written in the IC_CMBP_1 Register, the local A arrives together with the graphics pixel.

   A graphics pixel becomes transparent when color keying is enabled and a pixel color matches a key color (independently on an alpha parameter).

   The graphics data is read from a FIFO located in the Main Processing Memory. The FIFO contains 32 pages of size of 8 pixels. The size of the burst is defined according to the channel's corresponding CB#_BURST_SIZE bit. The graphics pixel format in the FIFO is RGB or RGBA or YUV 4:4:4 or YUVA. The graphics data is loaded by the IDMAC to the FIFO from the system memory.

All the operation are executed by an unified processing unit sequentially. Steps 1 and 2 cannot be interrupted by another task. All other steps can be interrupted by a task with higher priority if an input row is ready for this task. Preprocessing tasks priority is higher than postprocessing task priority.

The processing unit consists of three identical parts for each color component. All three color components are processed in parallel. Each of the processing operations can be enabled or disabled by an appropriate command according to.

The processing results are written to an output FIFO located in the Main Processing Output Memory row-by-row. The FIFO contains 32 pages, each page can include one burst of 8 or 16 pixels. The size of the burst is defined according to the channel's corresponding CB#_BURST_SIZE bit. The IDMAC transfers the output bursts to the system memory or to the display via DMFC (Channel 21only). The Main Processing Memory contains three buffers for each tasks: the temporary row buffer, the graphics FIFO and the output FIFO. Each memory word (128 bits) stores 4 adjacent pixels in formats RGB or RGBA or YUV 4:4:4.

38.4.5.5 Rotation Section

The rotation section includes the Rotation Memory, which stores an input rectangular block of 8x8 pixels and an output FIFO containing four pages of 8 pixels each one. The Rotation Memory word width corresponds to four adjacent pixels - 96 bits. The input block is loaded to the memory by the IDMAC like to a FIFO.

The Rotation Unit rewrites pixels from the input block to the output FIFO with corresponding relocation of a pixel inside the block. Rotation and/or left/right flipping and/or up/down flipping are enabled separately for each of three tasks. Configuring the rotation and flipping options is performed via the VF, HF & ROT parameters of the corresponding DMA channels (Table 38-13 and Table 38-14) responsible for task data input. The preprocessing task for encoding uses the VF, HF & ROT parameters from the IDMAC channel #45, the preprocessing task for viewfinder - from the IDMAC channel #46, the postprocessing task - from the IDMAC channel #47.

Rotation and flip options are shown in the following table.

Table 38-19. Rotation and Flip Options

ROT	FLR	FUD	Image
0	0	0	

Table continues on the next page...

Table 38-19. Rotation and Flip Options (continued)

ROT	FLR	FUD	Image
			
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	

Table continues on the next page...

Table 38-19. Rotation and Flip Options (continued)

ROT	FLR	FUD	Image
			
1	1	0	
1	1	1	

After finishing the rotation task, the IDMAC returns the output FIFO content to the system memory. When writing to the system memory, the IDMAC changes a location of the block relative to an input block location in order to provide proper rotation of the whole frame. Rotation tasks switching is performed after completion of rotation of the whole frame.

38.4.5.6 IC Task Parameter Memory

The following table presents IC task parameter memory details.

Table 38-20. IC Parameters

Address[1]	Word[2]	Parameter	Field	Description
x2008	Encoding CSC1 word0 and word1	C22	8:0	Coefficients of color conversion matrix1 for encoding task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2;
		C11	17:9	
		C00	26:18	

Table continues on the next page...

Table 38-20. IC Parameters (continued)

Address[1]	Word[2]	Parameter	Field	Description
				Coefficients format is s.xxxxxxxx[3];
		A0	39:27	Offset of color conversion matrix1 for encoding task: Offset format is sxxxxxxxxxx.xx
		SCALE	41:40	Scale of coefficients for color conversion matrix1 for encoding task: 0 --> coefficients *0.5 1 --> coefficients*1 2 --> coefficients*2 3 --> coefficients*4
		SAT_MODE	42:42	Saturation mode for color conversion matrix1 for encoding task: 0 --> (min, max) = (0, 255) 1 --> (min, max) = (16, 240) for Z1,Z2 1 --> (min, max) = (16, 235) for Z0
x2010	Encoding CSC1 word2 and word3	C20	8:0	Coefficients of color conversion matrix1 for encoding task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C10	17:9	
		C01	26:18	
		A1	39:27	Offset of color conversion matrix1 for encoding task: Offset format is sxxxxxxxxxx.xx
x2018	Encoding CSC1 word4 and word5	C21	8:0	Coefficients of color conversion matrix1 for encoding task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C12	17:9	
		C02	26:18	
		A2	39:27	Offset of color conversion matrix1 for encoding task: Offset format is sxxxxxxxxxx.xx
x4028	Viewfinder CSC1 word0 and word1	C22	8:0	Coefficients of color conversion matrix1 for viewfinder task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C11	17:9	
		C00	26:18	
		A0	39:27	Offset of color conversion matrix1 for viewfinder task: Offset format is sxxxxxxxxxx.xx
		SCALE	41:40	Scale of coefficients for color conversion matrix1 for viewfinder task: 0 -->coefficients *0.5

Table continues on the next page...

Table 38-20. IC Parameters (continued)

Address[1]	Word[2]	Parameter	Field	Description
				1--> coefficients*1 2--> coefficients*2 3-->coefficients*4
		SAT_MODE	42:42	Saturation mode for color conversion matrix1 for viewfinder task: 0 --> (min, max) = (0, 255) 1 --> (min, max) = (16, 240) for Z1,Z2 1 --> (min, max) = (16, 235) for Z0
x4030	Viewfinder CSC1 word2 and word3	C20	8:0	Coefficients of color conversion matrix1 for viewfinder task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C10	17:9	
		C01	26:18	
		A1	39:27	Offset of color conversion matrix1 for viewfinder task: Offset format is sxxxxxxxxxx.xx
x4028	Viewfinder CSC1 word4 and word5	C21	8:0	Coefficients of color conversion matrix1 for viewfinder task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C12	17:9	
		C02	26:18	
		A2	39:27	Offset of color conversion matrix1 for viewfinder task: Offset format is sxxxxxxxxxx.xx
x6060	Postprocessing CSC1 word0 and word1	C22	8:0	Coefficients of color conversion matrix1 for viewfinder task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C11	17:9	
		C00	26:18	
		A0	39:27	Offset of color conversion matrix1 for viewfinder task: Offset format is sxxxxxxxxxx.xx
		SCALE	41:40	Scale of coefficients for color conversion matrix1 for postprocessing task: 0 -->coefficients *0.5 1--> coefficients*1 2--> coefficients*2 3-->coefficients*4
		SAT_MODE	42:42	Saturation mode for color conversion matrix1 for postprocessing task:

Table continues on the next page...

Table 38-20. IC Parameters (continued)

Address[1]	Word[2]	Parameter	Field	Description
				0 --> (min, max) = (0, 255) 1 --> (min, max) = (16, 240) for Z1,Z2 1 --> (min, max) = (16, 235) for Z0
x6068	Postprocessing CSC1 word2 and word3	C20	8:0	Coefficients of color conversion matrix1 for postprocessing task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C10	17:9	
		C01	26:18	
		A1	39:27	Offset of color conversion matrix1 for post-processing task: Offset format is sxxxxxxxxxx.xx
x6070	Postprocessing CSC1 word4 and word5	C21	8:0	Coefficients of color conversion matrix1 for post-processing task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C12	17:9	
		C02	26:18	
		A2	39:27	Offset of color conversion matrix1 for postprocessing task: Offset format is sxxxxxxxxxx.xx
x6078	Postprocessing CSC1 word0 and word1	C22	8:0	Coefficients of color conversion matrix2 for viewfinder task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C11	17:9	
		C00	26:18	
		A0	39:27	Offset of color conversion matrix2 for viewfinder task: Offset format is sxx.xxxxxxxxx
		SCALE	41:40	Scale of coefficients for color conversion matrix1 for viewfinder task: 0 -->coefficients *2 1--> coefficients*1 2--> coefficients*0.5 3-->coefficients*0.25
		SAT_MODE	42:42	Saturation mode for color conversion matrix2 for viewfinder task: 0 --> (min, max) = (0, 255) 1 --> (min, max) = (16, 240) for Z1,Z2 1 --> (min, max) = (16, 235) for Z0

Table continues on the next page...

Table 38-20. IC Parameters (continued)

Address[1]	Word[2]	Parameter	Field	Description
x6080	Postprocessing CSC1 word2 and word3	C20	8:0	Coefficients of color conversion matrix2 for viewfinder task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C10	17:9	
		C01	26:18	
		A1	39:27	Offset of color conversion matrix2 for viewfinder task: Offset format is sxx.xxxxxxxxxx
x6088	Postprocessing CSC1 word4 and word5	C21	8:0	Coefficients of color conversion matrix2 for viewfinder task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C12	17:9	
		C02	26:18	
		A2	39:27	Offset of color conversion matrix2 for viewfinder task: Offset format is sxx.xxxxxxxxxx

1. The address documented in this table is the relative address within the TPM. This is the address that should be accessed when writing or reading from this memory via the AHB bus.
2. Each word is aligned to 64 bit accessible via the AHB bus in 2 separate 32bit accesses.
3. s - sign position, x - binary digit position

38.4.5.7 IC's DMA channels

The table below mas the IDMAC channels of the IC and the IRT to the corresponding tasks.

The IC's channel name is the name of the channel at IC level. The IC channel name is referred on the IC's programming model.

Table 38-21. IC's DMA Channels

IDMAC's channel number	IC's channel name	Read/ Write	Source	Destination	Processing Flow Purpose
20	CB0	Write	IC ENC	Memory	Preprocessing data from IC (encoding task) to memory
21	CB1	Write	IC VF	Memory/DMFC	Preprocessing data from IC (viewfinder task) to memory; This channel can be configured to send the data directly to the DMFC. This is done by programming the ic_dmfc_sel bit.
22	CB2	Write	IC PP	Memory	Postprocessing data from IC to memory

Table continues on the next page...

Table 38-21. IC's DMA Channels (continued)

IDMAC's channel number	IC's channel name	Read/ Write	Source	Destination	Processing Flow Purpose
14	CB3	Read	Memory	IC VF	Graphics data for combining (viewfinder task)
15	CB4	Read	Memory	IC PP	Graphics data for combining (post-processing task)
11	CB5	Read	Memory	IC PP	Postprocessing data from memory
12	CB6	Read	Memory	IC VF	Preprocessing data from sensor stored in memory (for example Bayer)
5	CB7	Write	IC	Memory	Direct data from IC (sensor data) to memory
48	CB8	Write	ENC ROT	Memory	Preprocessing data after rotation (encoding task)
49	CB9	Write	VF ROT	Memory	Preprocessing data after rotation (viewfinder task)
45	CB10	Read	Memory	ENC ROT	Preprocessing data for rotation (encoding task)
46	CB11	Read	Memory	VF ROT	Preprocessing data for rotation (viewfinder task)
50	CB12	Write	PP ROT	Memory	Postprocessing data after rotation
47	CB13	Read	Memory	PP ROT	Postprocessing data for rotation

38.4.5.8 IC restrictions

- The input's frame width to the IC must be a multiplication of 8 pixels
- When performing resizing the frame width must be multiple of burst size - 8 or 16 pixels as defined by CB#_BURST_16 parameter.

38.4.5.9 IC bridge

The IC sub-block utilizes a single memory to serve read and write channels. These memories are the IBM, RM and MPM. The IDMAC has separate mechanism to handle read and write channels. As a result, a contention between read and write channels may occur on each one of the memories. In order to resolve the contention, an IC bridge sub-block is connected between the channels associated with this memory.

The bridge prioritizes a read channel over a write channel. In order to avoid starvation of the write channels, the user can limit the maximum consecutive requests of the same channel that will be served. The limitation is done by programming the memory's corresponding <>_brdg_max_rq field.

38.4.6 Display port

The display port handles all the IPU features targeted for controlling and sending data to the display. The display port consists of 4 modules.

DC - a display controller,

DP - a display processor,

DMFC - a display multi-FIFO controller

DI - a display interface. The DI is instantiated twice to provide two symmetrical display interfaces.

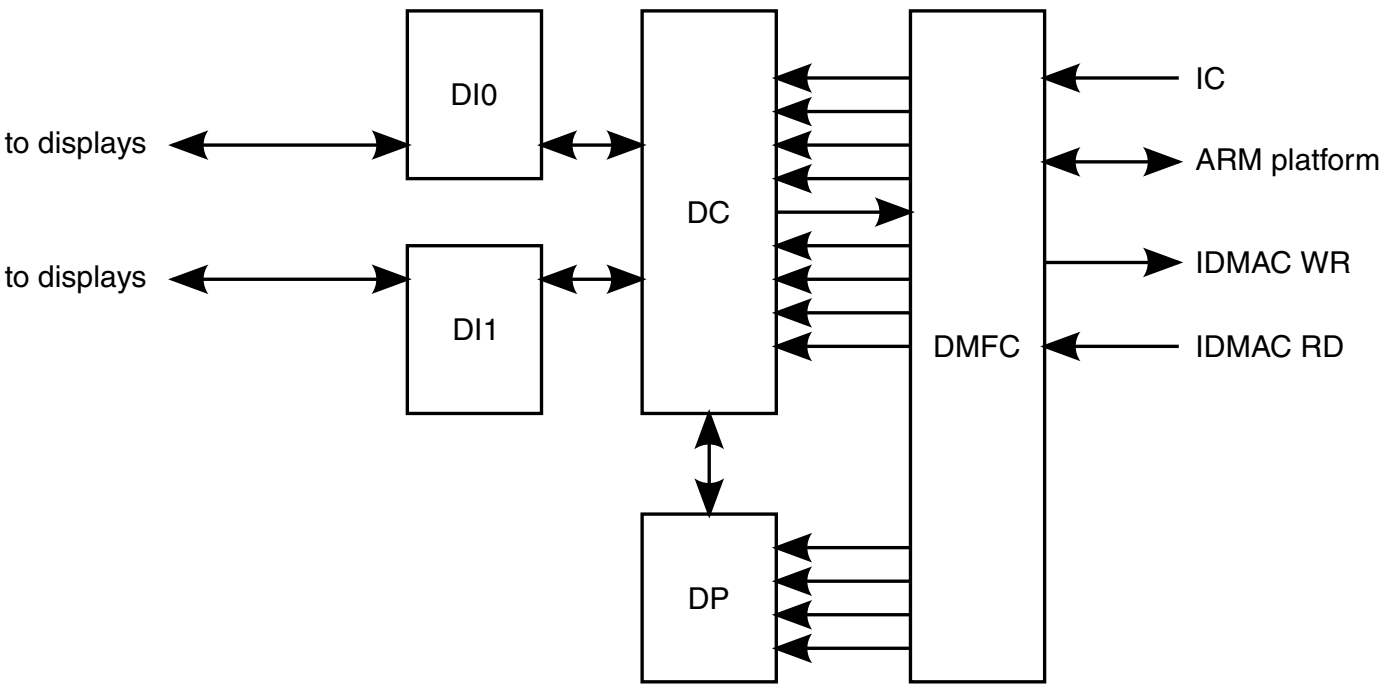


Figure 38-27. The display port

38.4.6.1 Display ports channels

The display port send data to the display over several channels. The data source may be the system's memory (via the IDMAC) the IC block and the ARM platform. The display port can read data from the display and send it to the ARM platform or IDMAC.

The data is routed over channels. The table below maps the IDMAC channels to DMFC, DC, DP channels and describes each channel.

Table 38-22. Display port channels

IDMAC's channel number	Display port's destination	DMFC channel number	DC channel number	Corresponding alpha channel	channel usage
21	DC	Programmable using dmfc_ic_in_port	Programmable using dmfc_ic_in_port	NA	This channel is coming from the IC module. When data is coming from the IC, the IC channel replaces one of the IDMAC's channels connected to the DMFC. When the IC_DMFC_SEL bit is set the output of the IC is routed to the DMFC. This channel can be routed to the DC channels 1,2,5B,5F,6B,6F. Routing this channel to the DC channel is done with the dmfc_ic_in_port bits. The DC's channel allocated to the IC channel can't be used for data coming from other source.
23	DP	5B	5	51	This channel is for the DP's primary flow. When a single plane is used, the data should be sent over this channel. When 2 planes are combined in this flow, the second plane should come on channel 27.
24	DP	6B	6	52	This channel is for the DP's secondary flow. When a single plane is used, the data should be sent over this channel. When 2 planes are combined in this flow, the second plane should come on channel 29.
27	DP	5F	5	31	This channel is for the DP's main flow. When a single plane is used, the data should be sent over channel 23 and this channel should not be used. When 2 planes are combined in this flow, one plane should come on channel 23 and the other plane on this channel.
28	DC	1	1	NA	This channel can serve sync and async flows. When channel 28 is connected to DI0, channel 23 must be connected to DI1 even if ch23 is not used. This is done by programming the PROG_DISP_ID_5 field
29	DP	6F	6	33	This channel is for the DP's secondary flow. When a single plane is used, the data should be sent over channel 24 and this channel should not be used. When 2 planes are combined in this flow, one plane should come on channel 24 and the other plane on this channel.
40	DC	0	0	NA	This is a read channel
41	DC	2	2	NA	This channel can serve only async flow
42	DC	1C		NA	Command stream. See Display port's restrictions
43	DC	2C		NA	Command stream. See Display port's restrictions
44	DC	3		NA	Mask channel. This mask channel can be associated with channel 23 or channel 28

38.4.6.2 Supported display interfaces

- The display port has 2 DI interfaces. Each interface can handle up to 3 displays.

- The total number of supported displays is 4.
- Each DI can handle up to 2 async interface - only one of them can be serial interface.
- Each DI can handle one synchronous interface. Asynchronous displays that are accessed in synchronous mode are considered synchronous interface.

38.4.6.2.1 Synchronous Interfaces

The DI supports the following synchronous display interfaces.

1. Synchronous generic interfaces to TFT dumb displays or RGB interfaces of smart displays.
2. Synchronous interfaces to Sharp displays.
3. Synchronous interfaces to TV encoders:
   - PAL
   - NTSC

   TV interfaces can operate in progressive or interlaced modes.
4. Synchronous interface to a graphic accelerator
5. BT.656
6. BT.1120

   BT.1120 and BT.655 support

   BT.1120 and BT.656 are supported. Only video data is supported, sending data during blanking intervals is not supported. The component size is always 8 bit.

38.4.6.2.2 Asynchronous Parallel Interfaces

The DI has a flexible asynchronous interface. The interface include 2 chip selects (CS) and 7 general purpose control signals.

The user can decide which of the 7 signals will be associated to each one of the asynchronous displays (up to 2 displays per DI). The using can configured some of the control signals to be shared by more than one display.

38.4.6.3 Display port's bandwidth

When the IPU clock (HSP_CLK) is equal to 264Mhz, the peak bandwidth supported by the display port is as follows

For on chip devices (like on chip MIPI-DPI bridge)

- 240 Mega-Accesses/Sec if the DI clock is derived from an external to IPU source (like another PLL)
- 264 Mega-Accesses/Sec if the DI clock is derived from the IPU clock (HSP_CLK)

For off chip devices (Like an external LCD)

- 170 Mega-Accesses/Sec if the DI clock is derived from an external to IPU source (like another PLL)
- 180 Mega-Accesses/Sec if the DI clock is derived from the IPU clock (HSP_CLK)

An access can be a pixel, component or generic data.

38.4.6.4 Display Dual Mode

This mode is useful for smart display with synchronous interface. The data is sent to the display only when the data has changed or when the display processor's settings are changed. The physical interface to this display has synchronous interface's characteristics.

38.4.6.5 Display Errors

There are a few types of errors that may impact the image sent to the display. The IPU provides some automatic mechanisms, which allow the system to overcome these errors.

38.4.6.5.1 Data starvation errors

These types of errors may happen for synchronous displays if the data is ready at the DI to be sent to the display at the time point it is required. This anomaly may happen if the system is very loaded and the IDMAC was not able to read data from the external memory and provide it to the DMFC.

Frame boundary errors

In case that a new frame should be sent to the display but the data of the previous frame was not sent completely, the IPU will reset the display modules and flush the internal buffers, as the IPU expects that the new frame indication will be triggered at least 2 lines (blanking interval) before the actual data is to be sent. The should recover and be ready with the data of the new frame within the blanking interval period.

Error within a frame

In case that a pixel was missed within a frame i.e. the pixel did not arrived to the DI on time. The IPU has 2 ways to handle the situation. The mode is selected by configuring the DI#_ERR_TREATMENT bit.

Redo the last access. The IPU will keep sending the last access to the display. The IPU will drop any pixel that arrive till the end of the line. If the data of the next line arrives correctly the IPU will continue working normally as overcame the problem. In case that the data of the next line is also incorrect the IPU will perform the same procedure as described on frame boundary errors.

Freeze the clock. In this mode, the IPU freezes the clock sent to the display till the correct data arrives. In order to avoid a case where the clock is frozen forever and the system is stuck: when the clock is frozen, a watchdog timer starts counting. When the timer completed counting the IPU will perform the same procedure as described on frame boundary errors. The number of cycles to be counted by the watchdog timer is defined according to the DI#_WATCHDOG_MODE bits.

38.4.6.5.2 Anti tearing errors

In case of anti tearing errors, the IPU indicates about the error by asserting the corresponding DC_TEARING_ERR_# bit.

See also Antitearing control.

38.4.6.6 Display port's restrictions

- In case where 2 synchronous flows are used and additional asynchronous flow via DP is used. The asynchronous flow via DP can be targeted to the same DI that the synchronous flow via DP is targeted.
- There are only 2 command channels (42 and 43). Command channels are associated with data channels (24,28,41).
  - When channel 28 is associated with a command, the command stream will come from channel 42
  - When channel 41 is associated with a command, the command stream will come from channel 43
  - Channel 24 can be associated with a command stream only if channel 28 or channel 41 do not use a command stream. If channel 28 is not associated with a command stream then channel 24 can be associated with channel 42. If channel 41 is not associated with a command stream and channel 28 is associated with a command stream then channel 24 can be associated with channel 43.

- A channel that uses an alternate flow, cannot be associated with a command stream (ch. 24 or ch. 41)
- In case of a synchronous display using external clock, The DI where the synchronous display is connected to can be connected to another async display. But, it can support only the write direction. Read via the asynchronous interface cannot be performed if this DI uses external clock

38.4.7 DC - Display Controller

IPU handles few display flows supporting few displays.

The IPU's flows' data sources can be the ARM platform, the system's memory, a camera or an external device connected on the display's port such as an external graphic accelerator.

The data's destination can be any device connected on one of the DI ports.

The DC controls the flows coming to and from the DI port. The DC manages the flows, decides which flows are currently active and when each flow is activated. The arbitrates between the active flows, gets the data from the predefined source and distribute it to the correct DI.

The DC's core is the microcode. The microcode contains a set of routine. A routine is built of a set of commands stored in the template's (microcode) memory. For each event (like new frame, end of frame etc.) a specific routine is executed. The user writes the routines and map them to a specific events. The routine contains instructions to the DC about the way of handling the data/address/commands associated with the display. The routine may contain information about the data's mapping, about waveform's characteristics, and more.

The figure below shows the micro architecture diagram for the DC block

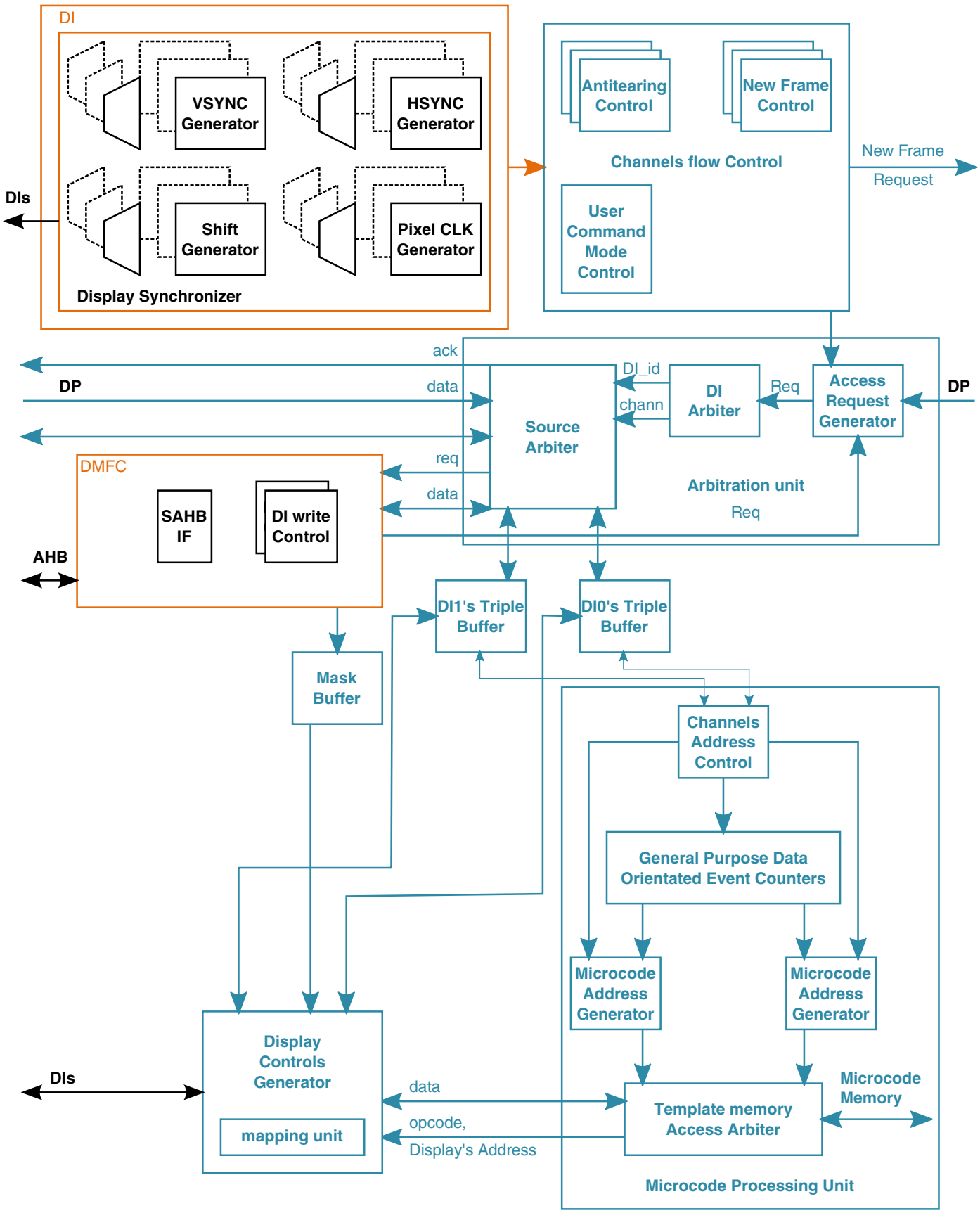


Figure 38-28. DC - Display Controller Block Diagram

38.4.7.1 Channels flow control

38.4.7.1.1 New Frame control

The channels flow control schedules the flows handled by the DC

For asynchronous flows the scheduling is done according to a request coming from the Frame Synchronization Unit (FSU) on the control sub-block (CM).

For synchronous flows the scheduling is done according to a trigger that is generated by a timer located on the one of the display's interface blocks (DI)

38.4.7.1.2 Antitearing control

Anti tearing mechanism uses a signal indicating on a display's refresh of a frame.

The supported tearing elimination triggers can be:

- An internally generated VSYNC signal
- A VSYNC signal generated by the display. The data source is the memory.
- A VSYNC signal coming from the CSI

The DC has the capability to avoid image tearing. For asynchronous flows where the source of the data is the system's memory or postprocessing, the DC monitors the position of a display's refresh pointer. Writing to the display is started only after crossing the window start point by the display refresh pointer. After that, writing to the display is allowed only when a write pointer does not advance beyond the refresh pointer. To provide anti-tearing mode, a window start time (in rows) must be defined in the PROG_START_TIME_1, PROG_START_TIME_2, PROG_START_TIME_5, PROG_START_TIME_6 registers for the corresponding channels.

The antitearing mechanism is limited to a case where only asynchronous flows are handle via the target DI.

In the case when tearing cannot be avoided (when the refresh rate is too high and the refresh pointer overtakes the write pointer after full refresh cycle), an error interrupt is generated. Tearing elimination mode can be disabled via the PROG_CHAN_TYP field for the corresponding channel.

38.4.7.1.3 User command mode control

A user may prepare in the system's memory a buffer that holds commends to be sent to the external device. The command buffer includes the same amount of lines as the data buffer. The line of commands are sent to the display line by line. A line of data is sent following each line of commands. This mode is activated by programming the PROG_CHAN_TYP of the corresponding channel.

The structure of the command is as follows:

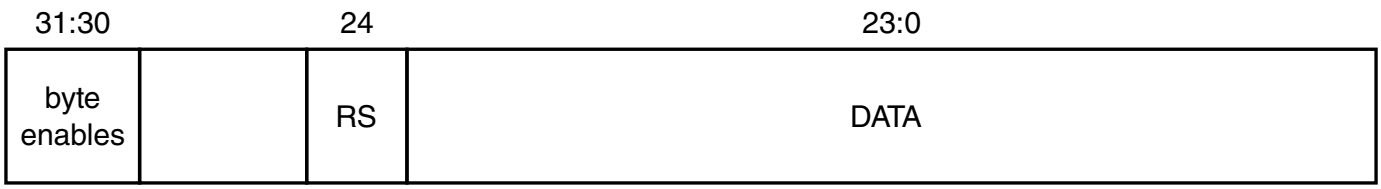


Figure 38-29. Structure of a command word

38.4.7.2 Arbitration Unit

This unit arbitrates the requests coming from the DMFC and from the DP and sends them to corresponding DI.

38.4.7.2.1 Access request generator

The requests coming from the DMFC or DP are sorted according to the target DI and then served according to a hard coded priority. The priority order is Sync flow, ARM platform access, IDMAC's Async flows.

38.4.7.2.2 DI arbiter

This units arbitrates between the DIs. The priority is hard coded. The priority order is Sync flow, ARM platform access, IDMAC's Async flows.

In case of 2 simultaneous requests to different DIs with the same priority the requests are served in a Round-Robin fashion. In case of 2 simultaneous Sync flow requests to different DI, the user can bypass the Round Robin mechanism and prioritize one DI on the other according to the SYNC_PRIORITY_1 & SYNC_PRIORITY_5 bits. Setting low priority to both of these channels is forbidden.

38.4.7.2.3 Source arbiter

Once the source of the request to be served was selected and the target DI was chosen, this unit routes the request and all the signals, associated with it, to the correct triple buffer and to the correct source of the data.

38.4.7.3 Microcode processing unit

The main control unit of the DC is the Microcode processing unit (MPU).

The data coming to the DC may be associated with some additional information like new frame, new line, new address etc. The information is processed in the MPU. The MPU executes the associated routine. The routine includes a set of instructions of the actions to be performed by the DC and DI.

38.4.7.3.1 Channels address control

This unit controls the display's address for each channel. This unit is responsible for defining the next address to be accessed (by incrementing or jumping). Stores special events flags (like EOF, EOL etc.). Based on the display's address and the special events, the type of the routines to be executed is defined.

38.4.7.3.2 General purpose Data oriented events counters

A user may define up to 4 general purpose events. The events are triggered by a standard event (NF, NL). The standard event restarts a counter, when the counter completes counting the user's general purpose event is asserted. This event activates a routine like any other events

38.4.7.3.3 Microcode address generator

This unit calculates the physical address of the template memory where the event associated routine resides. In addition, these unit arbitrates between simultaneous events and select the event to be served.

The arbitration is done according to a user defined priority. The priority of each event is set according to the #_CHAN_PRIORITY_CHAN_# bits of each channel. There are 2 modes of arbitration:

- Serving all the pending events according to the priority
- Serving only the highest priority event while ignoring all the other events

The arbitration mode is defined according to the CHAN_MASK_DEFAULT bit of each channel.

38.4.7.3.4 Template's Memory Access Arbiter

This unit gets memory access requests from the 2 microcode address generator units and arbitrates between them in a Round Robin fashion. In case that only one of the requests belongs to a synchronous flow, this request is selected.

38.4.7.4 DC's Template structure

The template memory contains 256 template words. Each template word is a 42 bits words. Accessing a template word require 2 accesses (32bit each).

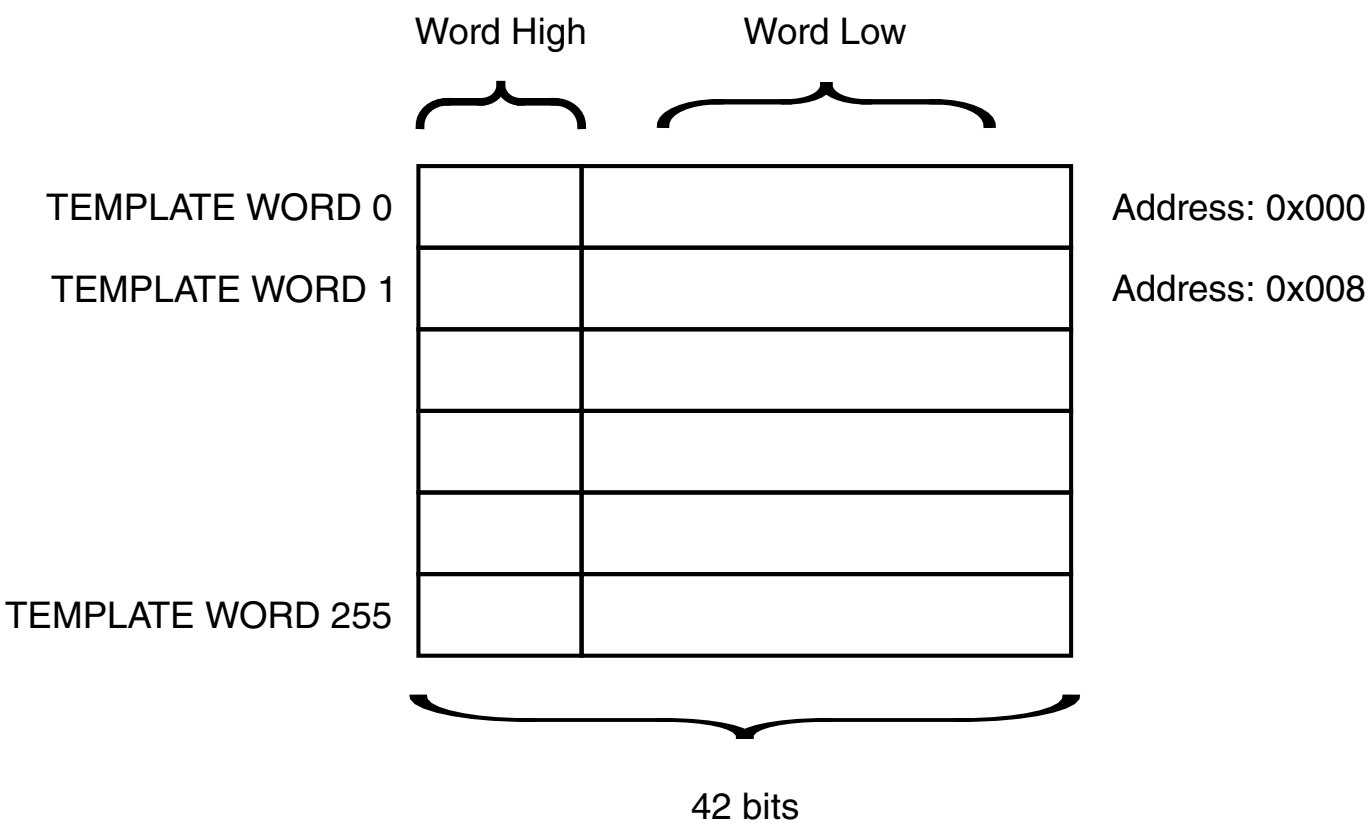


Figure 38-30. Template's structure

38.4.7.4.1 DC template's memory map

Table 38-23. DC template's memory map

0x1F80000 DC_MICROCODE_W0_L																
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OPERAND												MAPPING			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAP PING	WAVEFORM				GLUELOGIC							SYNC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1F80004 DC_MICROCODE_W0_H																
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	STOP	OPCODE								
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DC template's fields description

Table 38-24. DC template's fields description

Field	Description
STOP	Stop bit - This bit should be set in order to indicate that the current command is the last command of the routine
OPCODE	The command's code
OPERAND	The command's operand - for some of the commands this field can hold a parameter associated with the command
MAPPING	The MAPPING field holds a pointer to a register holding 3 fields: MAPPING_PNTR_BYTE0_X, MAPPING_PNTR_BYTE1_X, MAPPING_PNTR_BYTE2_X. This pointers point to sets of OFFSET and MASK parameters that define the mapping scheme. MAPPING = 0 means that mapping is disabled. The value in this field should be incremented by 1 to get the correct X pointer value In order to point to MAPPING_PNTR_BYTE2_0, MAPPING_PNTR_BYTE1_0, MAPPING_PNTR_BYTE0_0 the user should write 1 to the MAPPING field
WAVEFORM	For data oriented output pins. The IPU has 4 waveform generator units. The IPU holds 12 sets of waveforms' configuration registers called DI0_DW_GEN_<i> and DI1_DW_GEN_<i>

Table continues on the next page...

Table 38-24. DC template's fields description (continued)

Field	Description
	The WAVEFORM field defines which one of the 12 waveforms' configuration registers is used. The DI1_DW_GEN_X register holds a pointer to one of the 4 waveform generators units for each of the data oriented pins. 0 - The waveform of the data oriented output pins is not affected 1 -Points to DI0_DW_GEN_0 or DI1_DW_GEN_0 2 - Points to DI0_DW_GEN_1 or DI1_DW_GEN_1 ... 12 - Points to DI0_DW_GEN_11 or DI1_DW_GEN_11
GLUELOGIC	For signals generated by waveform generator #3; This field provides extra flexibility on the signals waveform
	GLUELOGIC[6] - This bit defines if we are in clock mode (1) or CS mode(0). 1- clock mode When the command is related to the display clock's pin then only if we are in clock mode, GLUELOGIC[5:4] are valid. 0- CS mode When the command is related to the CS pin then only if we are in CS mode, GLUELOGIC[3:0] are valid.
	GLUELOGIC[5:4] - clock mode settings 00 - Freeze the display clock following the execution of the current command 01 - Freeze the display clock before the execution of the next command to be executed 10 - Enable (unfreeze) the display clock following the execution of the current command 11 - Enable (unfreeze) the display clock before the execution of the next command to be executed
	GLUELOGIC[3] - CS mode settings 1 - Once the signal is asserted then it remains asserted (high or low according to the polarity) 0 - No impact on the waveform
	GLUELOGIC[2] - CS mode settings 1 - Once the signal is negated then it remains negated (high or low according to the polarity) 0 - No impact on the waveform
	GLUELOGIC[1] - CS mode settings 1- The current waveform can be attached to the previous waveform. If the previous waveform was asserted and the current waveform start asserted the two waveforms will be attached so the signals' waveforms will be consecutive. This impact the behavior of the previous waveform. This can be done only if GLUELOGIC[0] of the previous waveform is set to 1 0 - No impact on the waveform
	GLUELOGIC[0] - CS mode settings 1 - this bit allows the next waveform to be attached to the current waveform.
SYNC	The data associated with this command should be synchronized to the DI's one of gen_time_sync generators' output. 0000 - No sync. The data is sent without any synchronization to any event 0001 - Sync with unit #1 0010 - Sync with unit #2 ...

Table 38-24. DC template's fields description

Field	Description
	1010 - Sync with unit #10 1011 - 1111 Reserved

DC template's command description

The diagram below illustrates the command processing flow. The command is being fetched from the microcode memory. The Opcode is decoded. According to the decoding several controls are sent to the processing unit. In addition the processing unit gets the pixel's data and the display's address. The processed data/ address is stored in an internal register. There are 2 types of commands

HOLD - In this type of commands the data/address is stored in the register and not sent to the DI. The data/address is held in the register for further processing in the following commands.

WRITE - In this type of commands the data/address is stored in the register and also sent to the DI.

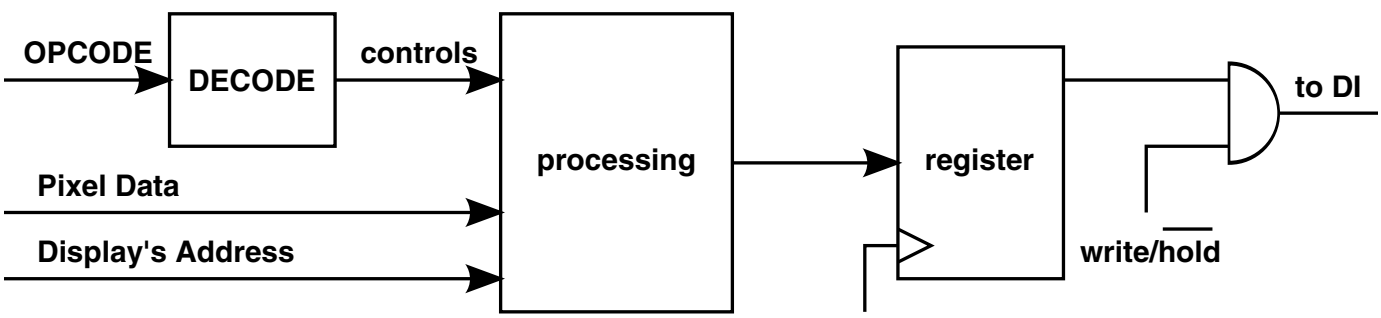


Figure 38-31. Opcode processing

The table below describes the DC's Commands.

Table 38-25. DC template's commands description

| Command | COMMAND[41:0] | | |

Table 38-25. DC template's commands description (continued)

Command	COMMAND[41:0]																																													
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
WRG	S	0	1	DATA																								WAVEFORM				GLUELOGIC							SYNC							
	Write 24bit word to the DI and Hold the word in register. DATA is a general purpose data to be written.																																													
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
HLOA	S	1	0	1	0	af	DATA																MAPPPING					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
	Hold the display's address in an internal register for further processing. af=0: No shift af=1: 8 bit right shift DATA is a 16bit general purpose data. This data is ORed with the 16 MSB of the mapped address.																																												
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
WROA	S	1	1	1	0	af	DATA																MAPPPING					WAVEFORM				GLUELOGIC							SYNC							
	Write address to the display and Hold address in register. af=0: No shift af=1: 8 bit right shift																																													
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
HLOD	s	1	0	0	0	0	DATA																MAPPPING					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
	Hold data in register. DATA is a 16bit general purpose data. This data is ORed with the 16 MSB of the mapped data coming from the data's source IDMAC or MCU.																																												
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
WROD	s	1	1	0	0	0	DATA																MAPPPING					WAVEFORM				GLUELOGIC							SYNC							
	Write data to DI and Hold data in register. DATA is a 16bit general purpose data. This data is ORed with the 16 MSB of the mapped data coming from the data's source IDMAC or MCU.																																												
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
HLOAR	S	1	0	0	0	1	1	1	af	0	0	0	0	0	0	0	0	0	0	0	0	0	MAPPPING					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
	Adding Mapped Address to held data and hold in an internal register. af=0: No shift af=1: 8 bit right shift																																												

Table continues on the next page...

Table 38-25. DC template's commands description (continued)

Command	COMMAND[41:0]																																											
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WROAR	S	1	1	0	0	1	1	1	af	0	0	0	0	0	0	0	0	0	0	0	0	0	MAPPPING					WAVEFORM				GLUELOGIC										SYNC			
	Adding Mapped Address to held data. Write to DI and hold in register af=0: No shift af=1: 8 bit right shift																																										
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HLODR	S	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MAPPPING					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Adding Mapped Data to held data and hold in register.																																										
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRODR	S	1	1	0	0	1	1	0	0	M2	M1	M0	0	0	0	0	0	0	0	0	0	0	MAPPPING					WAVEFORM				GLUELOGIC										SYNC			
	Adding Mapped Data to held data. Write to DI and hold in register. M0: 0: a new data[7:0] before mapping 1: a previous access data [7:0] before mapping M1: 0: a new data[15:8] before mapping 1: a previous access data [15:8] before mapping M2: 0: a new data[31:16] before mapping 1: a previous access data [31:16] before mapping If (M2 = M1 = M0 = 0) than the command performs: Adding a new Mapped Data to a held data in an internal register and write it to display else a display's data will be combined from a new data and previous data according to M-flags. The combined data will be mapped according to MAPPING and sent to a display. Examples: previous_data = 0x89ABCDEF new_data = 0x12345678 held_data (previous_data after mapping) = 0x0000EF Current MAPPING mode: new_data & 0x00ffff00 Output: If M2 = M1 = M0 = 0) than: Output = new_data OR held_data = 0x3456EF If M0 = 0, M1= 1,M2 = 1 than: Output = MAPPING ({previous_data[31:8], new_data[7:0]}) = MAPPING(0x89ABCD78) = 0x00ABCD00)																																										

Table continues on the next page...

Table 38-25. DC template's commands description (continued)

Command	COMMAND[41:0]																																														
	If M0 = 0, M1= 1,M2 = 0 than: Output = MAPPING ({new_data[31:16], previous_data[15:8], new_data[7:0]}) = MAPPING(0x1234CD78) = 0x0034CD00) if M0 = 1, M1= 0, M2 = 0 than: Output = MAPPING ({new_data[31:8], previous_data[7:0]}) = MAPPING(0x123456EF) = 0x00345600)																																											
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
WRBC	S	1	1	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	MAPPPING					WAVEFORM					GLUELOGIC										SYNC			
	Merge 1 bit mask channel with mapped data. Hold in register and write to DI. Mask data is coming from the DC's mask channel.																																													
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
WCLK	S	1	1	0	0	1	0	0	1	N_CLK_OPERAND													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	Wait N_CLK_OPERAND clocks. N_CLK_OPERNAD is the number of DI_CLK cycles to wait.																																													
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
WSTS-I	0	1	0	0	0	1	0	0	1	N_CLK_OPERAND													MAPPPING					WAVEFORM					GLUELOGIC										SYNC			
	Wait for Status I Read from the display and compare to a predefined PASSWORD. If the read data is equal to the PASSWORD, then continue. If not, redo the read & compare cycle. N_CLK_OPERAND is the number of DI_CLK cycles to wait before latching the data coming from the DI. DI_READ_DATA_ACK_VALUE_0 DI_READ_DATA_MASK_0																																													
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
WSTS-II	0	1	0	0	0	1	0	1	0	N_CLK_OPERAND													MAPPPING					WAVEFORM					GLUELOGIC										SYNC			
	Wait for Status II Read from the display and compare to a predefined PASSWORD. If the read data is equal to the PASSWORD, then continue. If not, redo the read & compare cycle. N_CLK_OPERAND is the number of DI_CLK cycles to wait before latching the data coming from the DI. WSTS-II command must be followed by a WSTS-I command. This command is useful in case that the PASSWORD is read in 2 accesses. The comparison will be done only after the execution of the WSTS-I command.																																												
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
WSTS-III	S	1	0	0	0	1	0	1	1	N_CLK_OPERAND													MAPPPING					WAVEFORM					GLUELOGIC										SYNC			
	Wait for Status III																																													

Table continues on the next page...

Table 38-25. DC template's commands description (continued)

Command	COMMAND[41:0]
	Read from the display and compare to a predefined PASSWORD. If the read data is equal to the PASSWORD, then continue. If not redo the read & compare cycle. N_CLK_OPERAND is the number of DI_CLK cycles to wait before latching the data coming from the DI. WSTS-III command must be followed by a WSTS-II command. This command is useful in case that the PASSWORD is read in 3 accesses. The comparison will be done only after the execution of the WSTS-I command. In case of a PASSWORD mismatch the read is done again. The entire cycle (WSTS-III -> WSTS-II -> WSTS-I) will be performed.
	41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RD	S 1 0 0 0 1 0 0 0 N_CLK_OPERAND MAPPPING WAVEFORM GLUELOGIC SYNC
	Read data from DI N_CLK_OPERAND - means delay value in DI_CLK for display's data latching by DI, defined by user For serial display the N_CLK_OPERAND is fixed and should be set to 1 value.
	41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
WACK	S 1 0 0 0 1 1 0 1 0 N_CLK_OPERAND 0 0 0 0 0 WAVEFORM GLUELOGIC SYNC
	Wait for acknowledge N_CLK_OPERAND - Number of DI_CLK cycles to count before monitoring the ACK received from the display.
	41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSK	S 1 1 0 0 1 0 0 0 0 0 0 0 0 e0m e1m e2m e3m nfm nlm nfldm eofm eolm eofldm nadm ncm dm 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	Mask - Mask a specific event; In case of more than one pending events the user can mask some of the event and serve the others according to the priority. e0m - event 0 mask, defined by user e1m - event 1 mask, defined by user e2m - event 2 mask, defined by user e3m - event 3 mask, defined by user nfm - new frame mask, defined by user nlm - new line mask, defined by user nfldm - new field mask, defined by user eofm - end of frame mask, defined by user eolm - end of line mask, defined by user eofldm - end of field mask, defined by user nadm - new address mask, defined by user

Table continues on the next page...

Table 38-25. DC template's commands description (continued)

Command	COMMAND[41:0]
	ncm - new channel mask, defined by user dm - data mask, defined by user
	41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
HMA	S 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ADDRESS 0 0 0 0 0
	HOLD_MICROCODE_ADDRESS - hold operand in special Microcode address register 0. This register is used to store a pointer in the microcode to be used by the BMA command ADDRESS is a microcode address in the microcode memory defined by the user.
	41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
HMA1	S 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ADDRESS 0 0 0 0 0
	HOLD_MICROCODE_ADDRESS - hold operand in special Microcode address register 1. This register is used to store a pointer in the microcode to be used by the BMA command ADDRESS is a microcode address in the microcode memory defined by the user.
	41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
BMA	S 0 0 1 1 LF AF 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 N 0 0 SYNC
	BRANCH_MICROCODE_ADDRESS jump to Microcode address which is stored at an internal Microcode address register. The branch will be done to the ADDRESS stored with the HMA/HMA1 command. The BMA command manages the repeat cycles by one of the auto decrement counters (internal counters), the counter's id is defined by LF-bit (0/1) The N operand defines the N-number of routine's repetitions. In case of N=0. The BMA command functions as a JUMP command. As long as the internal counter didn't reached its predefined value, the microcode will branch to the pre defined address stored by the HMA/HMA1 commands. Selecting between HMA and HMA1 is done by the AF bit (it is calculated according to the "1 - AF" value). - That's the subroutine's microcode start address When the counter reached its predefined value, the microcode will branch to the pre defined address stored by the HMA/HMA1 commands. Selecting between HMA and HMA1 is done by the AF bit (it is calculated according to the "AF" value). - That's the subroutine's microcode return address BMA synchronization (SYNC field): The BMA command can be used for context switching between 2-channels. When switching between flows the user can run a routine for a new channel, which actually affects the previous flow. This feature is useful when a new flow breaks the current flow (post command for example): SYNC is used for display_id selection // 0 0 0 - current display // 0 0 1 - previous display // 1 0 0 - display 0 // 1 0 1 - display 1 // 1 1 0 - display 2 // 1 1 1 - display 3

38.4.7.5 Display controls' generator

This unit generates the control signals associated with the data that are sent to the DI block.

38.4.7.5.1 Bus Mapping Unit

The Bus Mapping Unit is responsible for programmable mapping of the input data and commands to the display interface format and vice versa. Address mapping is done by this unit as well (programmable via micro code).

The internal DI format for data and commands is a 24-bits word divided into three byte components (eight zeroes are added to MSB for 16-bits words from the DC). This word can be output or input in one, two, three or four cycles of the display clock.

The word coming from the memory is a 32bit word. The mapping operation is done on 24 bits only. The 24 bit are selected from the received 32 bit according to the W_SIZE_# field. The 24 bit input word is partitioned to a 3 bytes (3X8bits). Each byte can be mapped to any position at the 24bit output word. The exact position is set according to the MD_MASK & MD_OFFSET fields.

The MAPPING_PNTR_BYTE0_X, MAPPING_PNTR_BYTE1_X, MAPPING_PNTR_BYTE2_X fields holds the pointers for the MD_MASK & MD_OFFSET for each byte.

The MAPPING field holds a pointer to a register holding 3 fields: MAPPING_PNTR_BYTE0_X, MAPPING_PNTR_BYTE1_X, MAPPING_PNTR_BYTE2_X.

0 - no mapping i.e. 32 bits are sent as is.

1 - points to MAPPING_PNTR_BYTE0_0, MAPPING_PNTR_BYTE1_0, MAPPING_PNTR_BYTE2_0

2 - points to MAPPING_PNTR_BYTE0_1, MAPPING_PNTR_BYTE1_1, MAPPING_PNTR_BYTE2_1

...

30 - points to MAPPING_PNTR_BYTE0_29, MAPPING_PNTR_BYTE1_29, MAPPING_PNTR_BYTE2_29

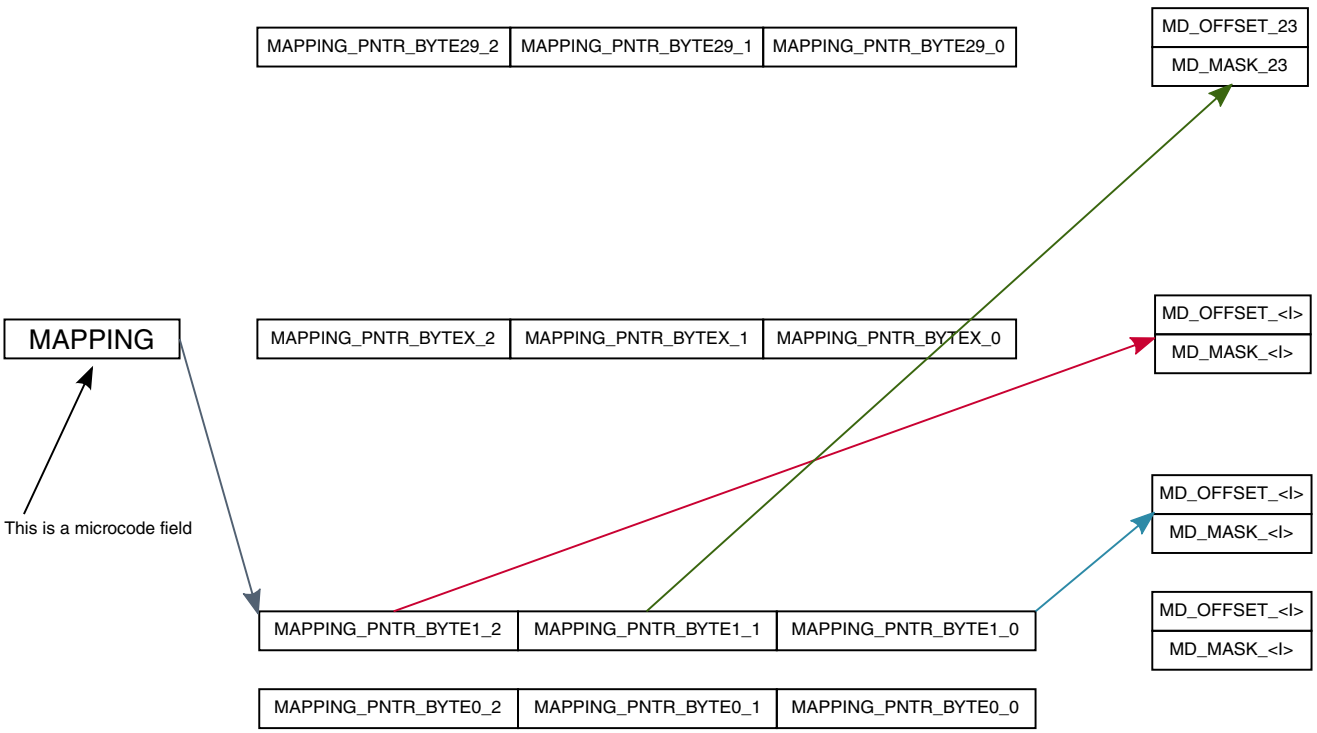
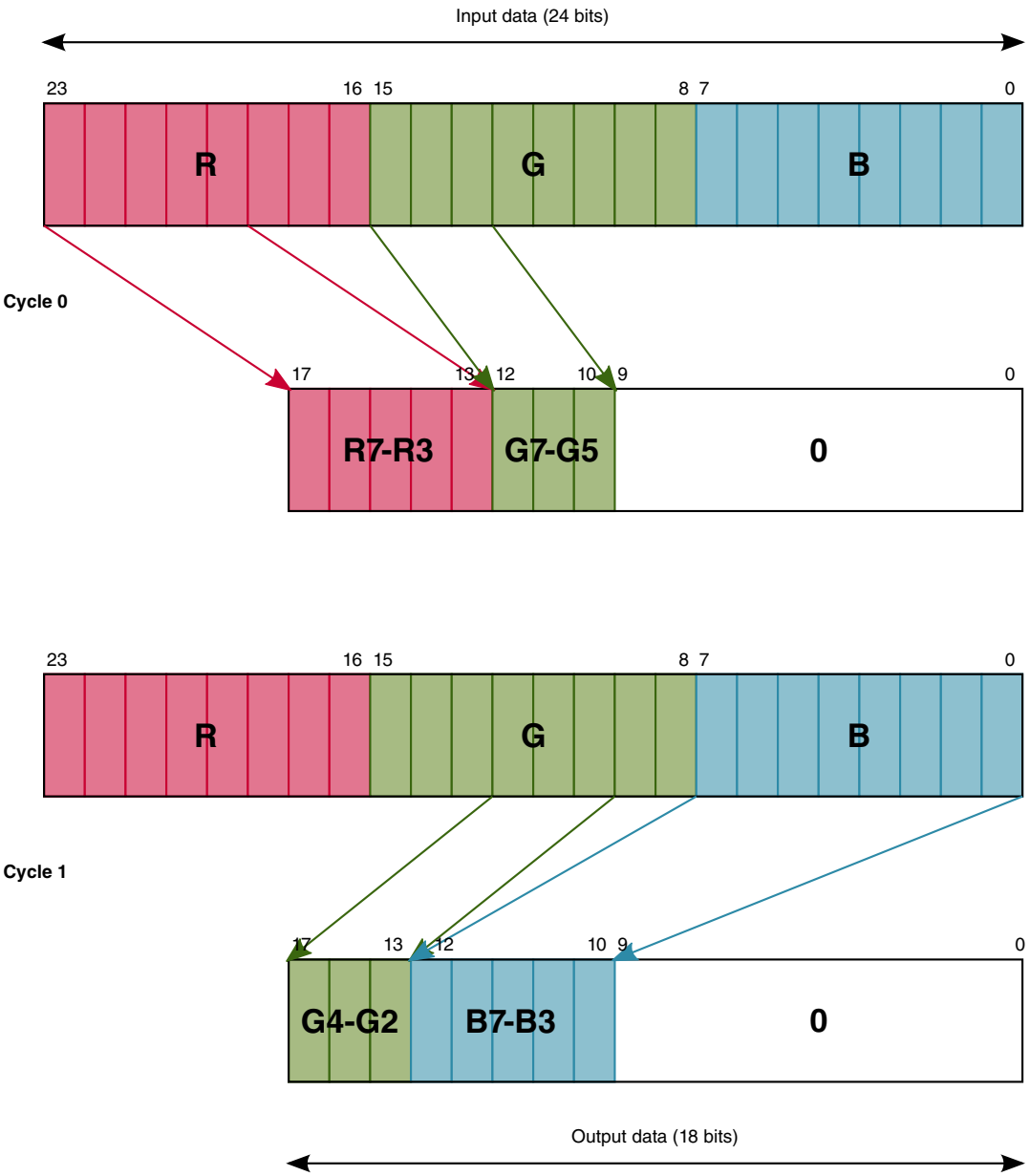


Figure 38-32. Mapping scheme

The mapping rule written in each of the Registers defines two types of parameters for the specific byte component and display:

1. Offsets of the byte component MSB relative to the output word LSB. Because the offsets can change dynamically, they are defined separately for the display clock cycles zero, one and two.
2. Numbers of the display clock cycles at which every bit of the byte component should be valid on the display bus. There are eight such 2-bit numbers in the Register.

Figure 38-33 presents an example of programming data packing for one of displays. Cycle 0 and Cycle1 in the diagram represent two separate atomic operations performed by the microcode template.



Cycle 0

MD_OFFSET_[R] - 0x11; MD_MASK_[R] - 0xF8

MD_OFFSET_[G] - 0xC; MD_MASK_[G] - 0xE0

MD_OFFSET[B] = 0x0; MD_MASK_[B] = 0x0

Cycle 1

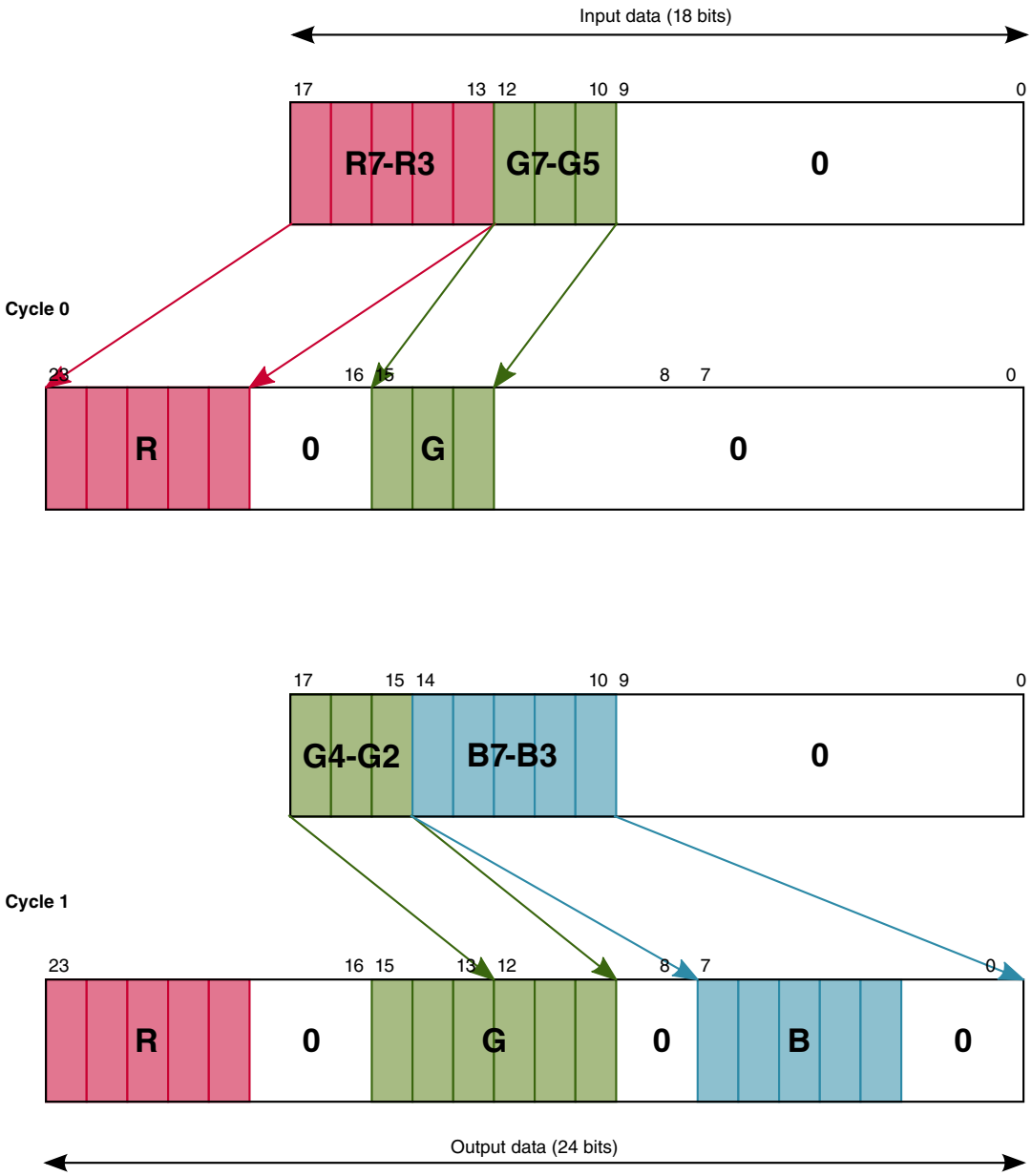
MD_OFFSET_[R] = 0x0; MD_MASK_[R] = 0x0

MD_OFFSET_[G] = 0x14; MD_MASK_[G] = 0x1C

MD_OFFSET_[B] = 0XE; MD_MASK_[B] = 0xF8

Figure 38-33. Example of Data Packing for Writing Data to the Display

The following figure presents an example of programming data packing for one of displays. Cycle 0 and Cycle1 in the diagram represent two separate atomic operations performed by the microcode template.



Cycle 0

MD_OFFSET_[R] - 0x11; MD_MASK_[R] - 0xF8

MD_OFFSET_[G] - 0xC; MD_MASK_[G] - 0xE0

MD_OFFSET[B] = 0x0; MD_MASK_[B] = 0x0

Cycle 1

MD_OFFSET_[R] = 0x0; MD_MASK_[R] = 0x0

MD_OFFSET_[G] = 0x14; MD_MASK_[G] = 0x1C

MD_OFFSET_[B] = 0XE; MD_MASK_[B] = 0xF8

Figure 38-34. Example of Data Unpacking for Reading Data from the Display

The same packing/unpacking registers are used for parallel and serial interface.

38.4.8 DMFC - Display Multi FIFO Controller

The following figure shows the block diagram for the DMFC block.

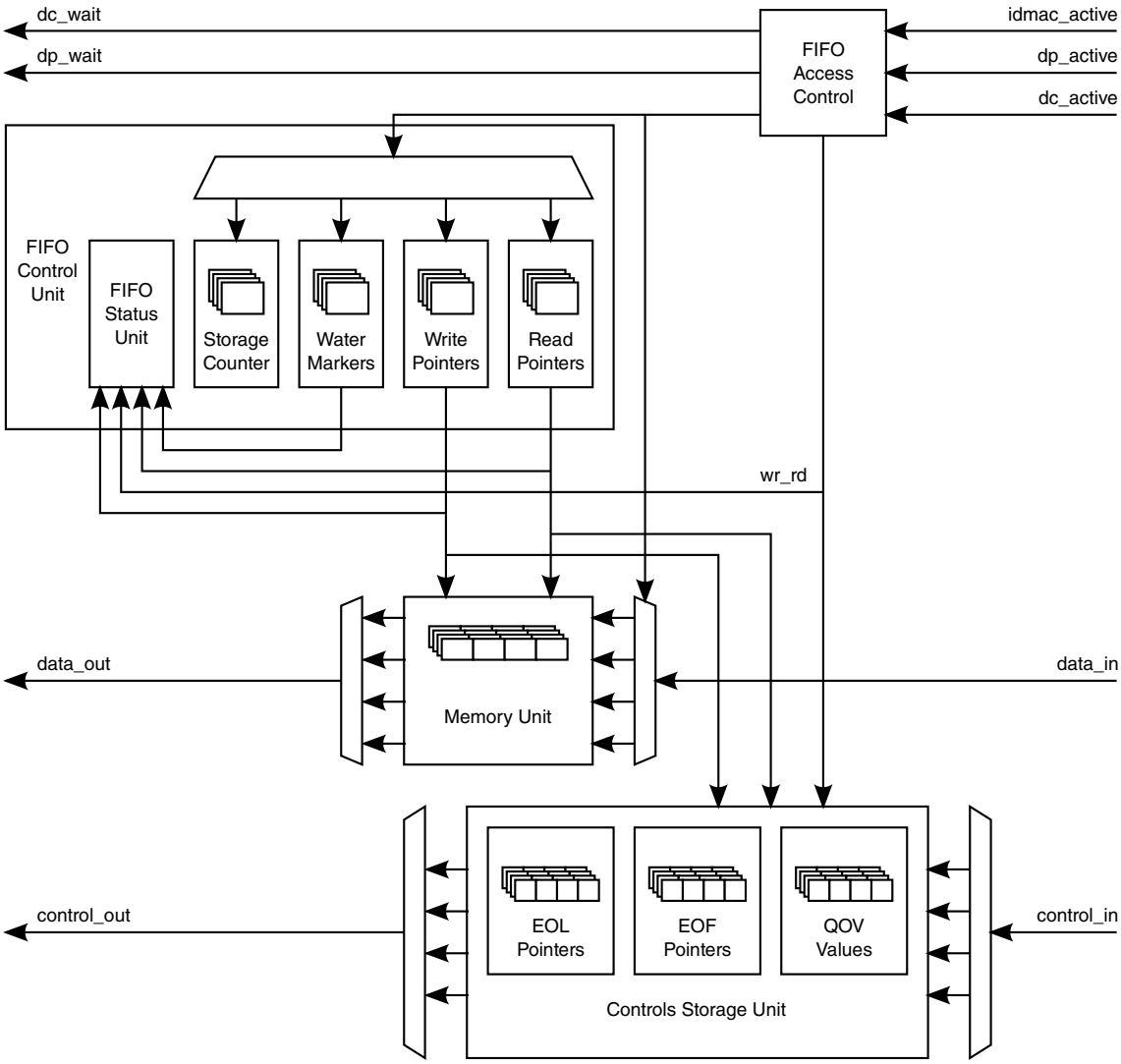


Figure 38-35. Display Multi Fifo Controller Block Diagram

The Display Multi Fifo Control manages Multi channels FIFOs. The DMFC serves the following clients:

- IDMAC - both read and write
- DP - read only
- DC - both read and write

- IC - write only
- AHB - both read and write

The DP and the DC read channels are physically attached to an IDMAC or an IC channel. As the IC has only one output channel connected to the DMFC. When the input is coming from the IC it replaces a channel that was physically attached to the IDMAC. The DMFC uses a single physical memory that serves the DP and DC read channels. The AHB accesses to the DC and the DC's write channel (read from display) use a separate physical memory.

The DMFC's write FIFO is built of 1024 entries of 128-bits each.

38.4.8.1 DP and DC read channels

Each one of the DP and the DC read (read from memory) channels is physically attached to an IDMAC read channel. A portion of the physical memory is allocated for each channel. The DMFC arbitrates between channels according to a predefined priority.

The DMFC controls each of the FIFOs

- Assert a request any time there's available place on the FIFO.
- Make sure that there's available place on the FIFO to accept the coming data.
- Optionally assert a watermark indication to avoid starvation

38.4.8.1.1 FIFO allocation to channels

The physical memory is partitioned to 8 segments. For each channel the user has to define the start address at a segment's boundary using the DMFC_ST_ADDR parameter.

The size of the FIFO allocated to a channel is defined by the DMFC_FIFO_SIZE parameter. The user must allocate the FIFO and avoid overlapping between FIFOs.

The DMFC hold few special indications like EOF, EOL, EOFILD. The most important one is end of line (EOL). If the size of the FIFO is shorter than or equal to the IDMAC line's length (FW) than no special restrictions on the DMFC usage.

If the size of the FIFO is greater than the IDMAC line's length (FW) than the user has to be aware to the following restriction for each channel.

The DMFC has two operation modes which are distinguished by the wait4eot bit. For each channel the DMFC can store a maximum number of EOL indication. The maximum number of EOL indications of EOL is given in the table below.

Table 38-26. DMFC's number of EOL indications

IDMAC's Channel	Maximum lines on the FIFO
23	Up to 3 lines
27	Up to 2 lines
28	Up to 2 lines
Other	Up to 1 line

If the use case is that the number of EOL indications cannot exceed the maximum number of EOL indications than the user should have the wait4eot cleared.

If the use case is that the number of EOL indication can exceed the maximum number of EOL indications than the user should have the wait4eot set.

Having the wait4eot bit set has performance impact as the DMFC analyzes the data prior to sending it to the destination. In addition the DMFC cannot utilize the entire FIFO allocated to this channel.

The user need to specify the burst size of the IDMAC by setting the DMFC_BURST_SIZE field. This field must match the IDMAC settings. In case that the IDMAC's burst size is not a power-of-2 number, the value of this field should be rounded up to the nearest power-of-2 number. The burst size must not be greater than the FIFO's size.

38.4.8.1.2 Arbitration between channels

The arbitration between channels is fully hardware controlled. IDMAC has the highest priority. Then the synchronous channels. Then the asynchronous channels.

38.4.8.1.3 Watermark

The DMFC can generate a water mark signal for each channel. The watermark signal is sent to the IDMAC and dynamically increases the channels priority on the IDMAC's arbitration.

The watermark feature is enabled by the DMFC_WM_EN bit. The FIFO is partitioned to bursts. The user can set the watermark level at a burst boundary.

The watermark signal is set when the number of bursts on the FIFO + the number of already requested burst is smaller than the value specified on DMFC_WM_SET bit.

The watermark signal is cleared when the number of bursts on the FIFO + the number of already requested burst is greater than the value specified on DMFC_WM_CLR bit.

DMFC_WM_SET must be smaller than DMFC_WM_CLR.

38.4.8.2 IC interface

One of the IDMAC channels can be replaced by a flow coming from the IC using the DMFC_IC_IN_PORT. The user has to provide the IC's setting to the DMFC by programming the DMFC_IC_FRAME_WIDTH_RD, DMFC_IC_FRAME_HEIGHT_RD and DMFC_IC_FRAME_PPW_C fields. The burst size of the channel coming from the IC should always be programmed to 4 words.

38.4.8.3 DC write channel and AHB accesses

The second physical memory of the DMFC serves the

- IDMAC write channel (read from display)
- 2 AHB channels that can be read or write

The IDMAC write channel is programmed using the DMFC_RD_CHAN register in a similar way to the DC and DP read channels described above. The user has to provide the frame width and height for this channel and the pixel per word parameter.

The AHB channels are used from accesses via the AHB port to the display. The accesses are distributed between channels according to the MCU_T parameter.

38.4.9 DP - Display Processor

The display processor processes the image prior to sending it to the display. The main task performed by the DP is combining between 2 planes.

The DP has 2 input FIFOs holding the data of full plane and the partial plane. In addition the DP performs some image enhancement functions like gamma correction, Color space conversion including Gamut mapping.

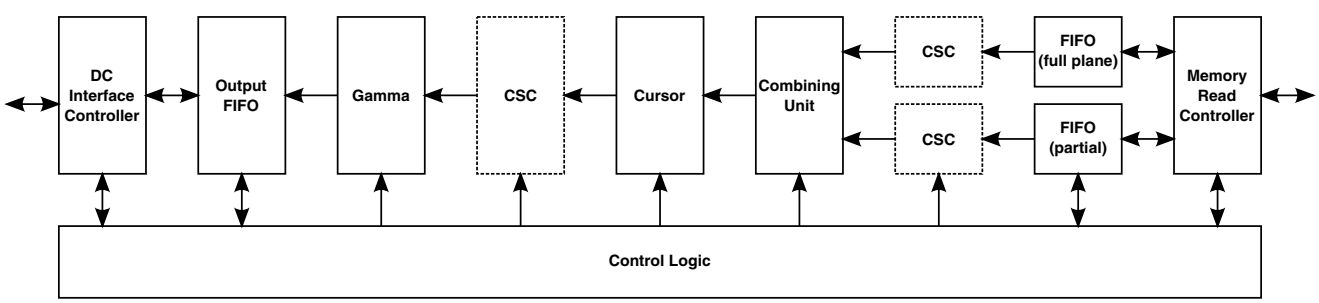


Figure 38-36. DP Micro architecture diagram

38.4.9.1 The DP programming model

The DP supports 3 flows. One sync flow and 2 Async flows. The DP holds 3 sets of registers. one set for each flow. Hence when referring to a register in this section the information is applicable to all the 3 sets. for example when referring to DP_COM_CONF, the information is applicable to DP_COM_CONF_SYNC, DP_COM_CONF_ASYNC0 and DP_COM_CONF_ASYNC1.

38.4.9.2 Displayed Planes

The following figure shows the planes displayed on a display.

There are full and partial planes. The partial plane's position is defined relatively to the upper left corner of the full plane (FGXP and FGYP parameters on the corresponding IPU_DP_FG_POS register). The size of the partial and full planes is defined on the corresponding IDMAC's channels' FW and FH parameters. The cursor position and parameters are set in the DP_CUR_POS register.

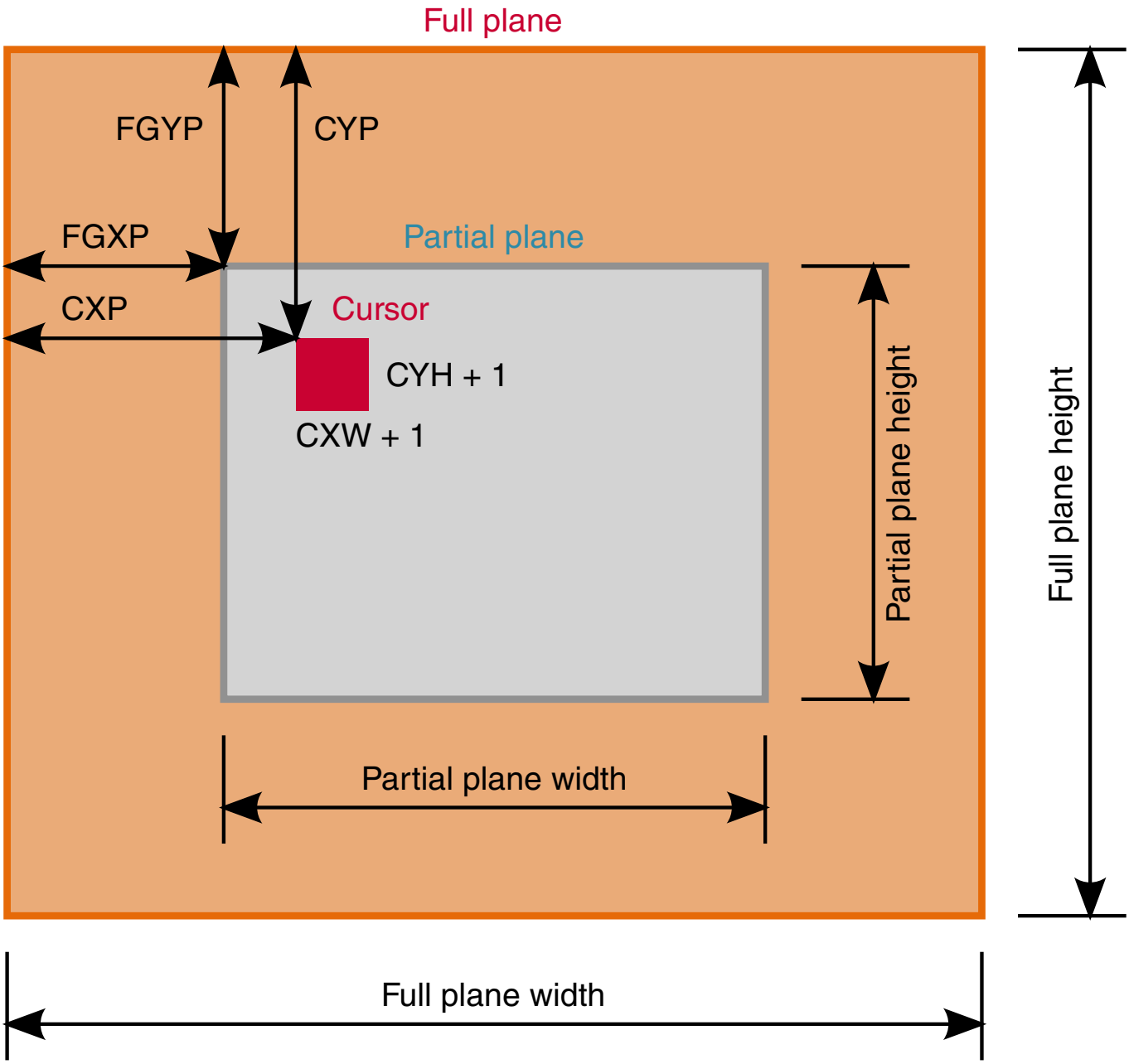


Figure 38-37. Displayed Planes

38.4.9.3 Combining Unit

The Combining Unit performs combining between the full and the partial planes. Each one of the planes may be graphics or video plane.

There are the following combining options:

- local alpha blending,
- global alpha blending,
- use of key color.
- order of the planes (full is presented over the partial plane and vice versa)

Combining mode is selected via the DP_COM_CONF Register. The combining equation is:

OP = BG*(1 - a) + FG*a

Where BG and FG are 2 input pixels; The DP_GWSEL bit defines if the BG is the pixel coming from the full plane or the partial plane.

a= (A+floor(A/128))/256 - an alpha value

A - a global or local transparency parameter.

The global A is written in the DP_GWAV field, the local A arrives together with the pixel.

A pixel becomes transparent when color keying is enabled and a pixel color matches a key color (independently on an alpha parameter). The color keying is defined on:DP_GWCKR, DP_GWCKG, DP_GWCKB

Combining takes 1 cycle per pixel. The Combining Unit outputs 24-bit words in the RGB/YUV format.

38.4.9.4 Cursor Generator

The Combining Unit output is passes through the Cursor Generator. The cursor's size and position are set via the DP_CUR_POS Register, a cursor color - via the DP_CUR_MAP Register. Different logic functions of combining the cursor with the image are supported as defined by the DP_COC field.

The cursor can be blinking. The blinking mechanism resides on the display controller sub-block. The blinking parameters are defined on the DC_BK_EN and DC_BKDVIV fields.

38.4.9.5 Color Space Conversion unit - CSC

The DP can get 2 input pixels from 2 different color spaces (YUV or RGB) and convert one of them to a common color space (YUV or RGB). In addition the 2 inputs can be of the same color space where the result is converted to another color space (YUV or RGB).

The DP has a single CSC unit that can be placed on one of 3 locations:

- At the output of one of the 2 input FIFOs
- At the output of the cursor generator.

Placing is done according to the DP_CSC_DEF field.

The color conversion implements a 3x3 matrix multiplication between the full RGB pixels and the color conversion constants, in order to obtain a YCC format image.

The conversion formula is:

x -> Clip(Round(S* 2^E)), S=Ax+B

where

A is a 3x3-dimensional matrix of weights, each a 10-bit signed number with 8 fractional digits

$$A = \begin{bmatrix} CSC_A0 & CSC_A1 & CSC_A2 \\ CSC_A3 & CSC_A4 & CSC_A5 \\ CSC_A6 & CSC_A7 & CSC_A8 \end{bmatrix}$$

B is a 3-dimensional vector of offsets, each a 14-bit signed number with 2 fractional digits

$$B = \begin{bmatrix} CSC_B0 & CSC_B1 & CSC_B2 \end{bmatrix}$$

S is a 3 dimensional vector of sums, each a 16-bit signed number with 4 fractional digits

S=Ax+B

E is an exponent, assuming one of the following values: -1,0,1,2 (allowing weights up to 8). The CSC_S parameters are encoded by 2 bits, please refer to the CSC_S parameter description.

$$E = [CSC_S0\ CSC_S1\ CSC_S2]$$

A more explicit formula:

S[i] = (sum(A[i][j]*In[j]) >> 4) + (B[i] << 2) + (1 << (3-E[i]))

Out[i] = Clip(S[i] >> 4-E[i])

Where Clip() performs clipping to the range 0..255 (either per-component clipping or more sophisticated clipping that preserves the hue of the pixel)

38.4.9.5.1 Gamut mapping

When the color transformation produces colors outside the allowed range, they must be mapped back. This is called gamut mapping. The DP supports 2 clipping algorithms. (controlled by GAMUT_SAT_EN bit)

Hue preserving clipping algorithm is suitable only for RGB components. For YUV components, the per-component clipping algorithm is used.

Per component Clipping

This mapping is performed by clipping each of the components independently to its allowed range of values (the final value being uint8):

•Y: to 0..255 or 16..235 according to the SAT_MODE bit

•U/V: to 0.255 or 16..240 according to the SAT_MODE bit

Hue Preserving Clipping

Hue Preserving clipping is done in the following way

•First stage - eliminating negative values

```
N = min(R,G,B)
if (N<0) X-> X-N, where X=R,G,B
```

•At this stage, all components are non-negative and the MSB's beyond d=11 bits are ignored (assumed 0).

•Second stage - eliminating large values

```
M = max(R,G,B) (d-bit integer)
if (M>255)
```

```
M' = M >> (m-7), where m=8..d-1 is the index of the most-significant non-zero
bit in M
D' = Ceil(256*255/M') = 256..510 (since M' = 128..255)
Ceil(x) = the smallest integer which is not smaller than x
Implemented by a hard-wired 128x9-bit LUT
X -> min(255, (X*D')>>(m+1)), where X=R,G,B (8x9 multiplier)
else
X -> X
```

38.4.9.6 Gamma correction

The DP includes a gamma correction function. It is approximated by the piece-wise polynomial:

gammar = GAMMA_C<i> + ((R[4:0] * GAMMA_S<i>) >> 4 + 1)>>1

Where R is the input red component, that's a 9 bit input composed pixel_in*2+pixel_in[7].

Single approximation slope is used for Gamma Correction of red, green and blue color components. However the Gamma Correction block instantiated three times since processing for color components should be done in parallel. The gamma transform is also available for changing the contrast of the luminance component only.

The required Gamma correction slope for a specific display should by provided by the display manufacture. This information can be provided in various forms, as graph or formula. The gamma correction input pixel level (Gin) should be normalized to a maximum of 383. The gamma correction output pixel level (Gout) should be normalized to a maximum of 255. Then a following data should be collected:

Table 38-27. Gamma correction values

Gin	Gout
0	Gout0
2	Gout1
4	Gout2
8	Gout3
16	Gout4
32	Gout5
64	Gout6
96	Gout7
128	Gout8
160	Gout9
192	Gout10
224	Gout11
256	Gout12

Table continues on the next page...

Table 38-27. Gamma correction values (continued)

Gin	Gout
288	Gout13
320	Gout14
352	Gout15

Based on the table above the values of DP_GAMMA_S_SYNC<i> and DP_GAMMA_C_SYNC<i> fields for gamma correction control registers are calculated as following:

Table 38-28. Gamma correction values

i	DP_GAMMA_C_SYNC<i>	DP_GAMMA_S_SYNC<i>
0	Gout0	16*(Gout1-Gout0)
1	2*Gout1-Gout2	16*(Gout2-Gout1)
2	2*Gout2-Gout3	8*(Gout3-Gout2)
3	2*Gout3-Gout4	4*(Gout4-Gout3)
4	2*Gout4-Gout5	2*(Gout5-Gout4)
5	Gout5	Gout6-Gout5
6	Gout6	Gout7-Gout6
7	Gout7	Gout8-Gout7
8	Gout8	Gout9-Gout8
9	Gout9	Gout10-Gout9
10	Gout10	Gout11-Gout10
11	Gout11	Gout12-Gout11
12	Gout12	Gout13-Gout12
13	Gout13	Gout14-Gout13
14	Gout14	Gout15-Gout14
15	Gout15	255-Gout15

38.4.9.7 DC interface

The DC interface unit performs 2 tasks.

- Starts the flow via the DP by getting a request from the DC and once the DP is ready send the new frame request to the IDMAC.
- Control the DP's output FIFO and send the data to the DC using an handshake mechanism.

38.4.9.8 DP's flows management

The DP can manage up to 3 flows: one synchronous flow and 2 asynchronous flows. However the DP can handle only one flow simultaneously. The DP has an automatic mechanism to control and switch between flows.

The DP's highest priority flow is the sync flow. An async flow can be executed during the blanking interval of the sync flow. An async flow can be stopped in order to serve the sync flow. The sync flow cannot be broken. The DP holds 3 sets of registers, one set for each flow. The registers are located in the SRM memory. According to the flow that needs to be executed the correct set of registers is loaded to the DP.

The figure below illustrates the flow management done by the DP.

A flow starts when a request from the DC arrives and the internal pipe is empty. For sync flows, the full frame NF (new frame) indication arrives immediately. In case that a partial frame is also used the NF indication will be sent one row before the row that the partial plane is actually positioned.

In case of ASYNC flow the DP first check if the previous async flow was broken. If yes, the DP restores the last settings of the previous flow. If this is a new async flow the DP will first reload the flow's parameters from the SRM. In case that a partial frame is also used the NF indication will be sent one row before the row that the partial plane is actually positioned.

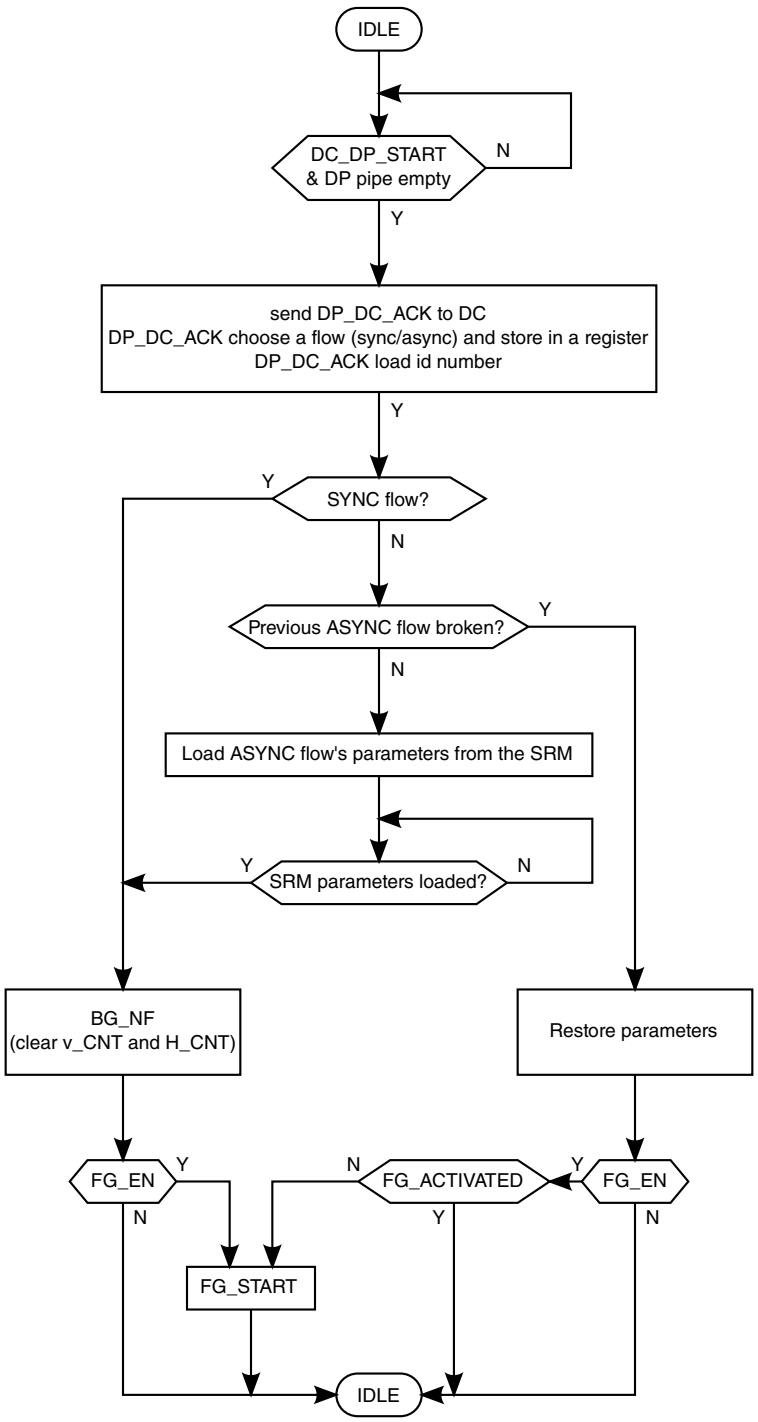


Figure 38-38. DP flow management chart

38.4.9.9 DP debug unit

The DP supports synchronous and asynchronous flows using the same hardware. The asynchronous flow can be broken by the synchronous flow. The DP's debug unit provides the ability to know on which row exactly the async flow has been broken. This is done by providing an interrupt (with DP_DEBUG_CNT register) and providing row status flags (on DP_DEBUG_STAT register).

As the async flow can broken multiple times within a specific frame The user can control the breaking point the issues the debug event by programming BRAKE_CNT field.

38.4.9.10 Restriction

When both full and partial planes are processed, the full plane's minimal frame width is 13 pixels.

The Minimum frame height supported by the DP is 2 lines.

38.4.10 Display Interface (DI)

The DI provides arbitrates access to up to three displays with time multiplexing. It converts a data from the DC or the ARM platform (low level access for serial interface only) to a format suitable for the specific display interface.

The DI generates display clocks and other display control signals with programmable timings. The DI outputs data to or inputs from parallel and/or serial interfaces.

This block generates all the control signals sent to the display. The DC sends to the DI; the data for the display and a set of control signals. The controls coming from the DC are used in order to generate the control signals sent to the display. One exception is serial low level access (LLA), where the DC is bypassed and the data is coming directly from the ARM platform. The figure below is the DI's block diagram.

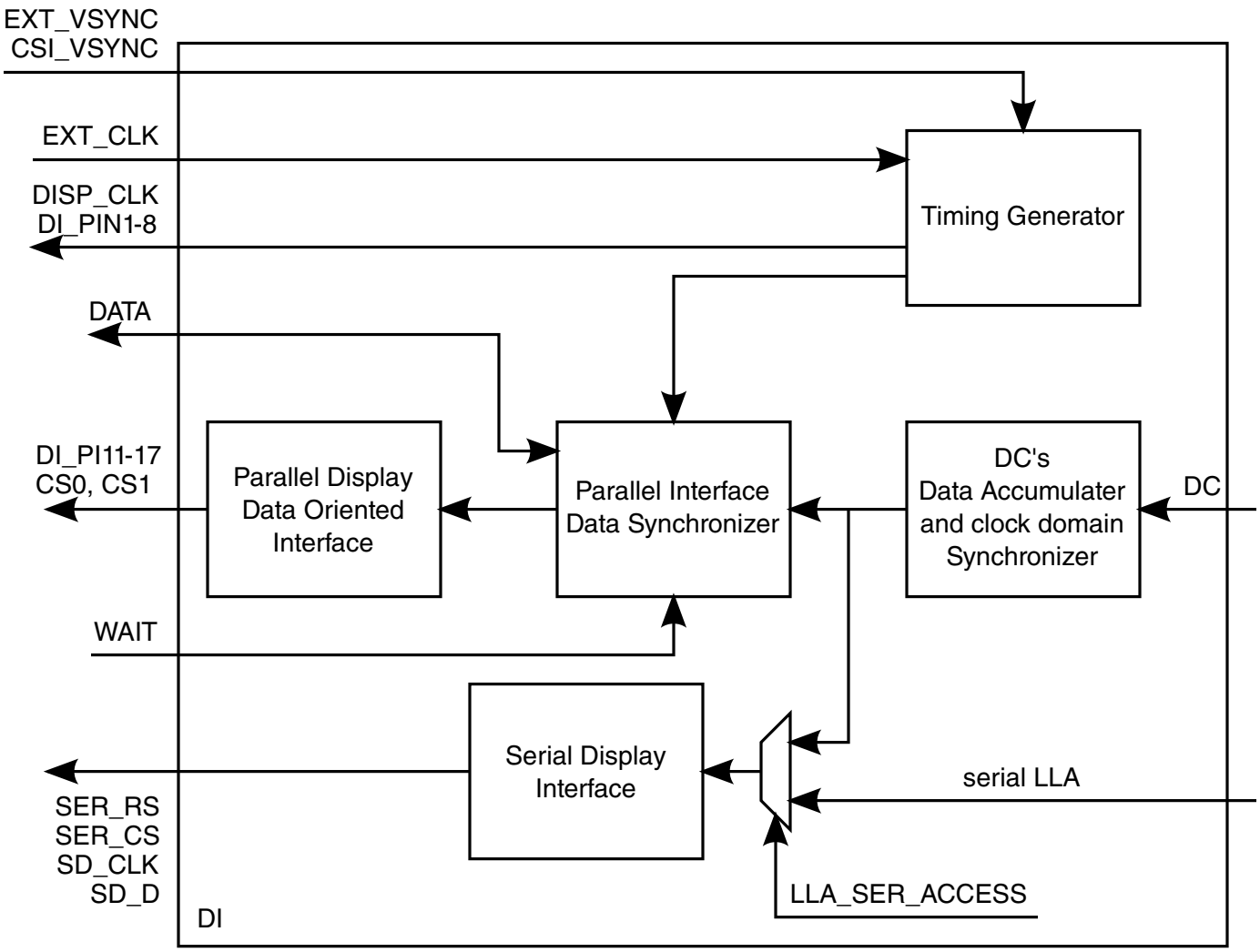


Figure 38-39. DI's block diagram

NOTE

The Serial Display Interface is not supported on this product.

The display interface includes 2 groups of control signals:

- Time oriented signals - These type of signals are generated according to the DI's internal timers. These are free running signals that change their state according to a pre-defined waveform. For example VSYNC, HSYNC, display's clock (pixel clock) etc.
- Data oriented signals - The DC may add markers to data sent to the DI. These markers are used to indicate a specific attribute of the data (for example: end-of-line, end-of-frame, chip-select etc.). The marker coming from the DC triggers a specific waveform of one or more signals on the display's interface. The specific waveform will be seen on the bus along with the associated data. The markers may be synced to

a time oriented signal. For example: attach the end-of-frame signal to the next VSYNC.

38.4.10.1 DC interface, data accumulator and clock domain synchronizer

The data accumulator is the DI's input buffer. It receives the data from the DC along with a set of control signals. The data accumulator receives the data from the DC's clock domain and synchronize it to the DI's clock domain.

38.4.10.2 Parallel interface data synchronizer and data oriented interface

The data accumulated in the DI's input buffer will be sent to the display according to the DI's internal events. Each event is generated by a counter. A tag is attached to each data by the DC's microcode using the SYNC field in the DC's microcode. The tag selects the event that the data will be synced to.

Once the corresponding event is generated, the data will be sent to the display. A data can be a pixel or a component (part of a pixel).The data synchronization occurs separately for each component. For asynchronous displays the tag will be equal to 0 i.e. the data is not synchronized to any event. The data will be sent out of the buffer immediately.

38.4.10.3 Timing generator

The timing generator is used for generating the waveforms' of each pin of the display's interface.

The timing generator is built of 10 counters. One counter functions as a time base This timer is called the BASE TIMER (BS). The other 9 counters are used in order to generate the control signals' waveforms. The last counter (counter #9) is special see Counter number 9.

The DI clock can be derived from IPU's clock (HSP_CLK) or from an external source (via the DIn_DISP_CLK - ipp_di_#_ext_clk pin). The clock's source is statically selected by configuring the DI#_CLK_EXT bit.

Figure 38-41 illustrates the main parameters of a waveform. A waveform's segment is built of 5 parameters. Each segment can be has 2 phase "ACTIVE PHASE" and "OFFSET PHASE".

- TIMEBASE - this is the base timer (fundamental timebase); all the other parameters are derived from this timer. The timebase is generated by counting DI clock cycles. The amount of cycles is defined according to DI#_DISP_CLK_PERIOD. This field defines the Display interface clock period, This parameter contains an integer part (bits 11:4) and a fractional part (bits 3:0). It defines a fractional division ratio of the Di's source clock for generation of the display's interface clock. The timebase is generated from edge aligned pulses of the DI clock. The user can delay the timebase starting point by defining an offset to the timebase. This is done by configuring the DI#_DISP_CLK_OFFSET field. The offset is calculated from the point where the DI is enabled to the point where the timebase starts ticking. The display clock waveform is generated between 2 edges of the timebase. The waveform is defined according to the DI#_DISP_CLK_UP and DI#_DISP_CLK_DOWN. Each pin has a specific timebase that is derived from the fundamental timebase. The following figure illustrates the relations between the TIMEBASE and the DI's clock

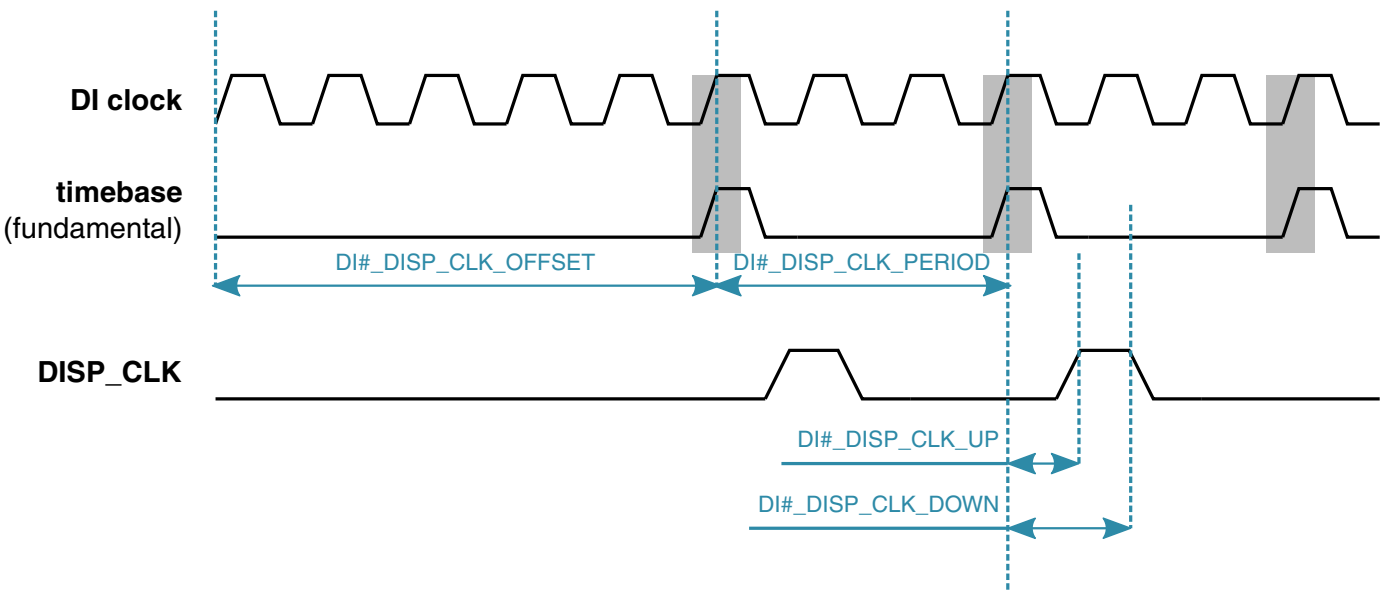


Figure 38-40. Timebase, DI's clock and display's clock relations

- OFFSET - this parameter defines when the length of the "OFFSET PHASE" it is defined by di#_offset_value_<N> field, where N is the counter's index.
- STEP - This parameter defines the length of the "ACTIVE PHASE"; it is defined by the di#_step_repeat_<N> field, where N is the counter's index. If the counter is in auto reload mode (di#_cnt_auto_reload_<N> bit is set) then the counter will be automatically reloaded forever. The value of di#_step_repeat_<N> is ignored in that case.

- RUN - The "ACTIVE PHASE" is partitioned to several "RUN sections"; this parameter defines the length of the "RUN section"; it is defined by the di#_run_value_m1_<N> field, where N is the counter's index.
- UP - Each "RUN section" contains the waveform of a single pulse. This parameter defines the offset from the beginning of the "RUN section" to the assertion of the signal; it is defined by the di#_cnt_up_<N> field, where N is the counter's index.
- DOWN - This parameter defines the offset from the beginning of the "RUN section" to the negation of the signal; it is defined by the di#_cnt_down_<N> field, where N is the counter's index. In case where DOWN < UP the waveform will have a 50% duty cycle.

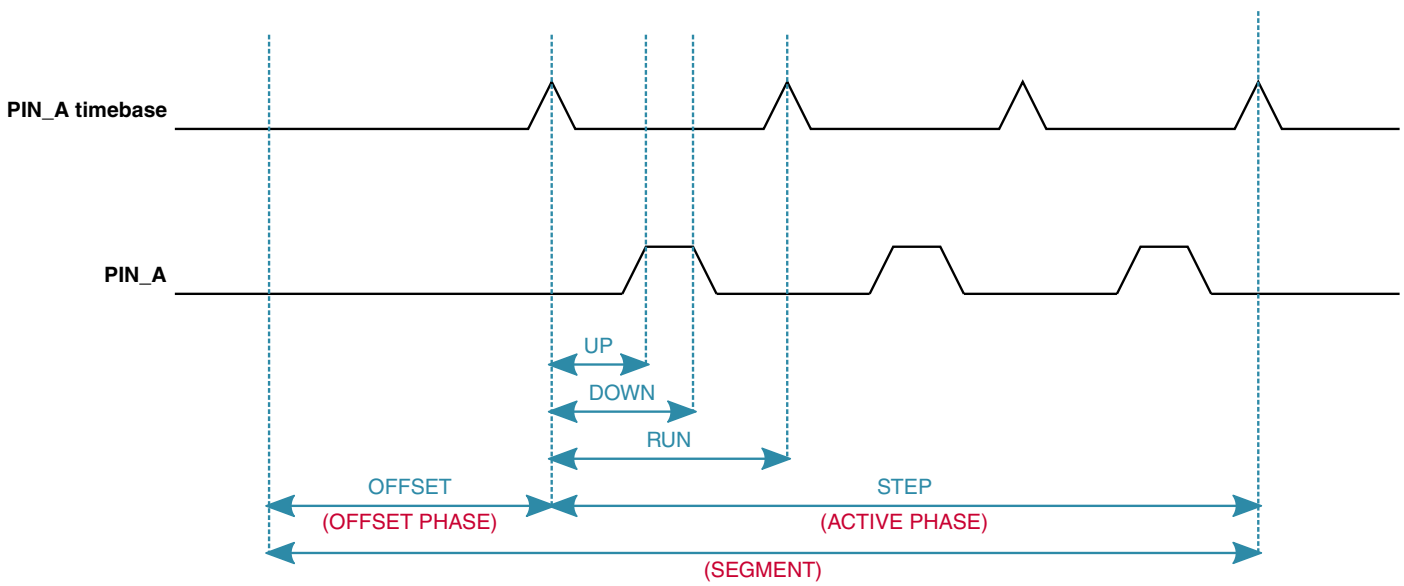


Figure 38-41. DI waveform's main parameters

38.4.10.3.1 Waveform concatenation

The DI provides the ability to derive the waveform from the fundamental timebase or from another PIN. In that case, one pin's waveform is used as another pin's timebase.

The following figure provides an example. PIN_A and PIN_C are derived from the fundamental timebase. However, PIN_B is derived from PIN_A's wavefore. The trigger is selected by DI#_RUN_RESOLUTION_<N>, where <N> is the index of the counter. A counter can be triggered by a counter with lowered index. For example: counter #5 can be triggered by counter #3 but can't be triggered by counter #7.

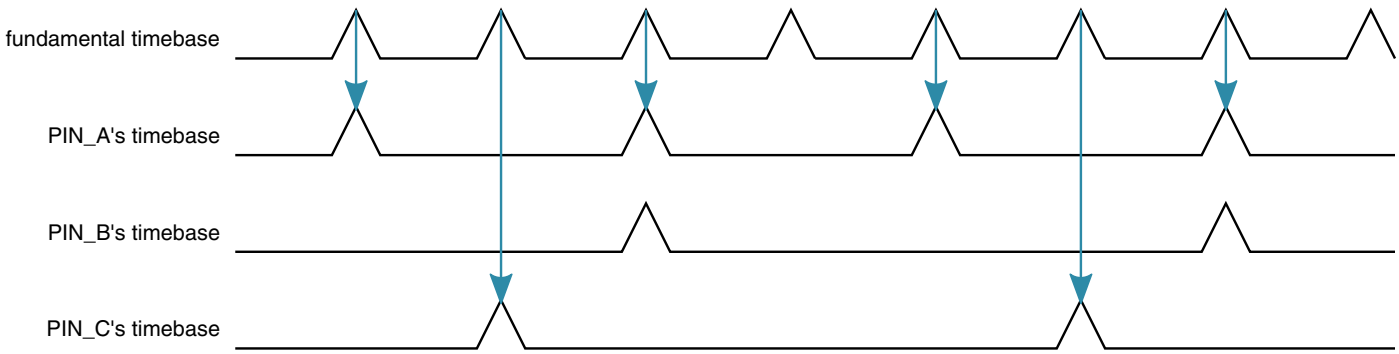


Figure 38-42. DI pins - Waveform's time bases concatenation

38.4.10.3.2 The basic counter

The DI has 9 counters. A counter is built of 3 units: timebase generator, waveform generator and polarity generator.

Figure 38-43 illustrates the counter's structure.

The timebase generator

The timebase generator gets 3 triggers. Clear, offset and run trigger. The Clear is the trigger that resets the counter. It is selected by programming the DI#_CNT_CLR_SEL_<N>.

The Offset trigger is the trigger used for counting the OFFSET_PHASE. The user can select the source of the trigger by programming the DI#_OFFSET_RESOLUTION_<N>. The offset's value is defined by programming the DI#_OFFSET_VALUE_<N>

The RUN trigger is the trigger used for counting the RUN period. The user can select the source of the trigger by programming the DI#_RUN_RESOLUTION_<N>. The RUN's value is defined by programming the DI#_RUN_VALUE_<N>

The timebase generator counts according to the DI_CLK or according to another counter's output. In order to use a source different than DI_CLK, the user should set the POLARITY_GEN_EN bits to 01.

Waveform generator

This unit generates the waveform. It gets the values of RUN, UP, DOWN and STEP and build the waveform accordingly. The waveform is counted according to the signal generated by the timebase generator.

Polarity generator

The waveform's polarity is controlled by 2 units. The static polarity is changed according to the POLARITY_<#> bit of each pin. This bit defines if the waveform of the pin is active high or active low.

The other unit controlling the polarity is the polarity generator. This unit is enabled by setting the POLARITY_GEN_EN[1] to 1. The polarity generator has 2 modes: "toggle mode" and "normal polarity mode". The mode is defined according to POLARITY_GEN_EN[0].

- Normal polarity mode - In this mode the polarity is changed according to 2 other counters. The counter selected by POLARITY_TRIGGER_SEL define the sampling point. The counter selected by POLARITY_CLR_SEL defines the polarity value. At the sampling point the polarity value is defined. The current polarity value defines the current polarity of the waveform.
- Toggle mode - In this mode, the output of the timebase generator's output causes the polarity to toggle. Any tick of the timebase generator inverts the polarity. When this mode is enabled the ticks generated by the counter selected by POLARITY_TRIGGER_SEL initialize the polarity generator and the timebase generator causes the polarity to toggle.

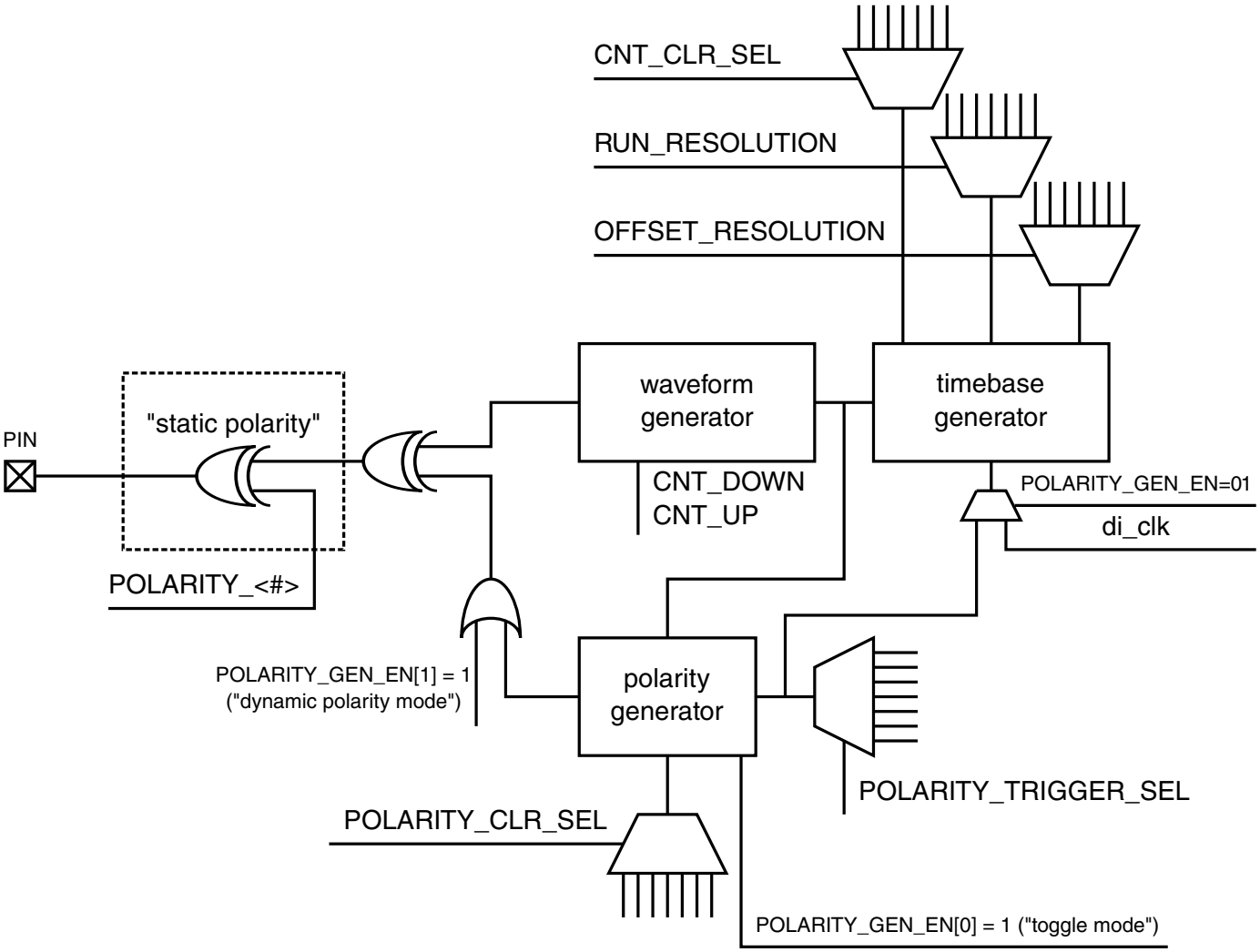


Figure 38-43. DI's counter's structure

The following figure provides an example for waveform generation using different trigger sources. The waveform is generated for PIN_D. PIN_D counter is cleared by the PIN_C's timebase. The offset period is calculated by counting cycles of PIN_B's timebase. The RUN period is calculated by counting cycles of PIN_A's timebase. The UP and DOWN periods of the waveform are calculated by counting cycles of PIN_D's timebase.

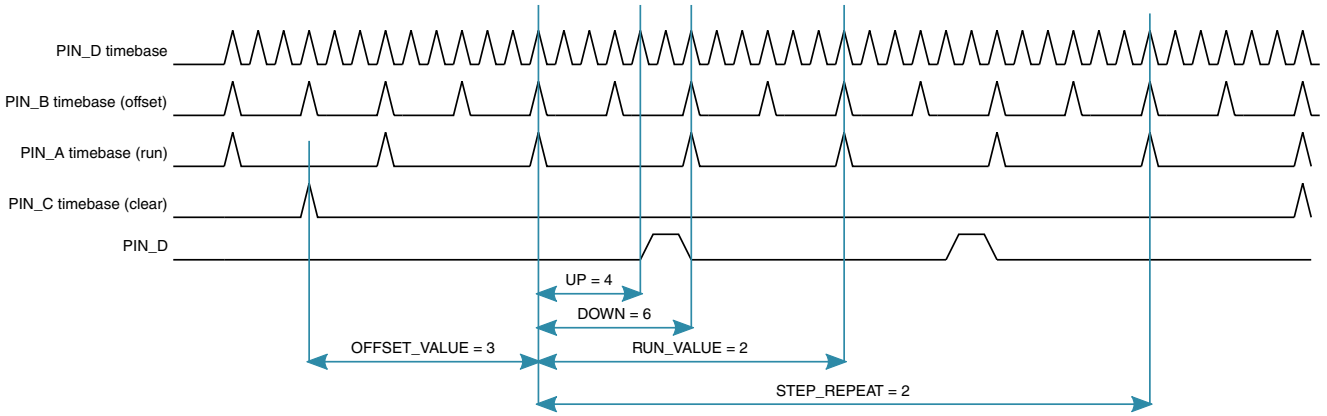


Figure 38-44. Clear, offset and run triggers and values - Example

38.4.10.3.3 Counter number 9

The last counter (counter #9) is an auxiliary counter and it is not used for generating a waveform for a specific pin. It can be used in order to attach another waveform to an existing one.

The user define the waveform for a specific pin (main pin). The user defines the auxiliary waveform using counter #9. The 2 waveforms are logically ORed. The combined waveform will be routed to the main pin. The main waveform that this counter is attached to is defined according to DI#_GENTIME_SEL_9.

The tag of counter #9 can be generated from counter #9 or from the main waveform. This is selected according to the DI#_TAG_SEL_9.

38.4.10.3.4 DI's active window

The DI provides an alternative way to define the synchronous display setting by using an active window and thus, needs to program less counters. The synchronous display's active window is a rectangle on the display where IPU sends data. It is set by programming the parameters defined on DI#_AW0 and DI#_AW1 registers.

The following figure illustrates the different parameters defining the active window. The display's vertical and horizontal position are defined according to counters. The DI#_AW_HCOUNT_SEL selects the counter which the horizontal position is calculated according to. DI#_AW_VCOUNT_SEL selects the counter which the vertical position is calculated according to. The Active data is sent according to a trigger. DI#_AW_TRIG_SEL selects the counter which calculates this trigger. This trigger usually functions as a data enable signal.

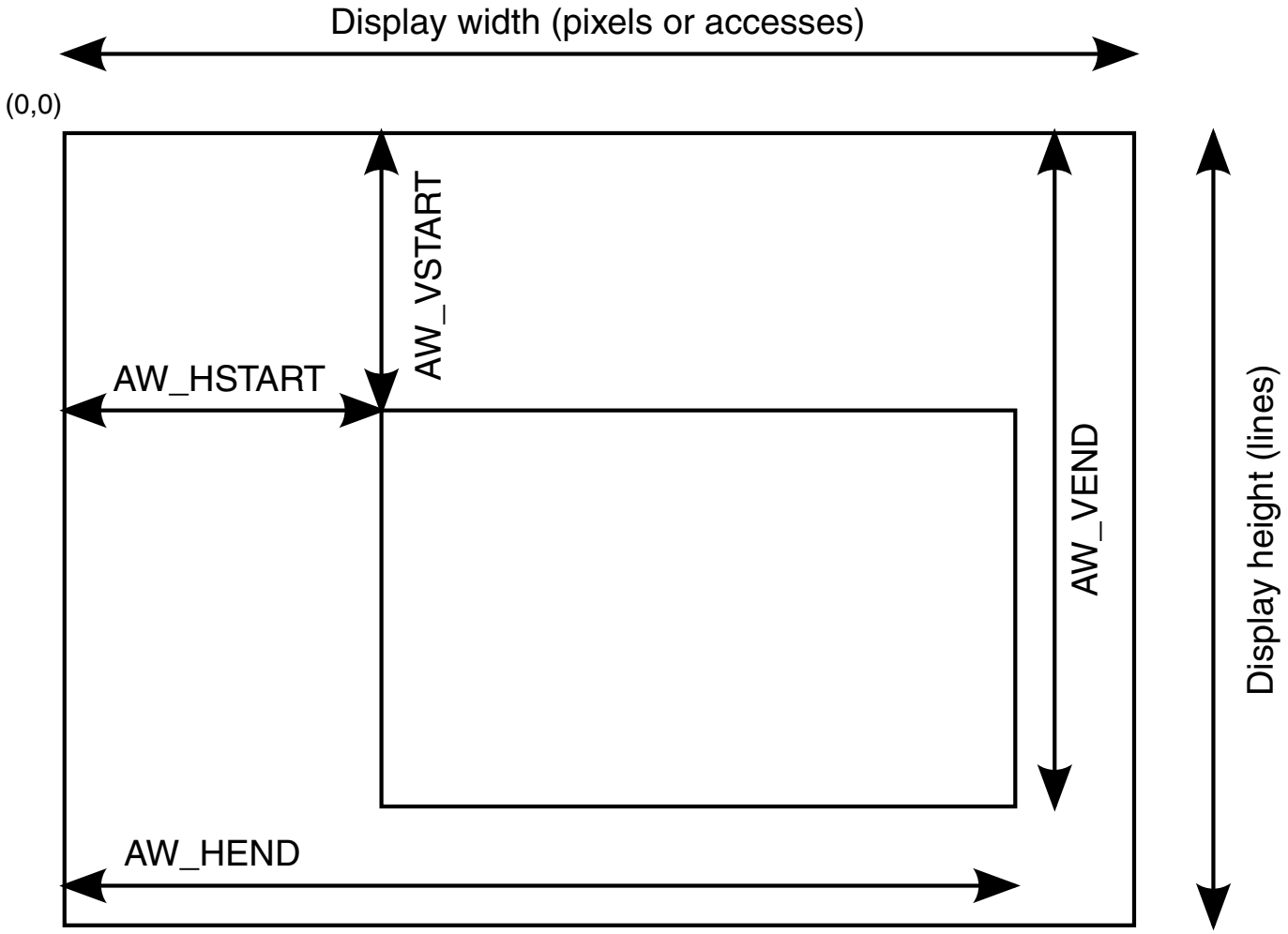


Figure 38-45. DI's Active Window

38.4.10.4 Waveform settings for asynchronous interface pins

The DI provides 8 signals that are used for asynchronous interface. These signals are PIN11 through PIN17 (ipp_di_#_pin_11 through ipp_di_0_pin_17) and the CS (ipp_di_0_do_dispb_d0_cs).

The DI holds 12 wave set quartets. Each quartet includes 4 registers (DI#_DW_SET<j>_<i>). Each DW_SET register holds the UP and DOWN values of the waveform. The DW_SET register is selected in the following way. The WAVEFORM field in the DC template is a pointer that points to one of the 12 quartets. In addition the WAVEFORM field points to one of 12 DI#_DW_GEN_<i> registers. The DI#_DW_GEN_<i> holds 9 pointers. The pointers are 2 bits field that points to one of the registers from the DI#_DW_SET<j>_<i> quartet. Each one of the 8 pointers in the

DI#_DW_GEN_<i> registers is related to a specific pin of the DI's asynchronous interface. The following figure illustrates the relations between the registers controlling the asynchronous interface's signals.

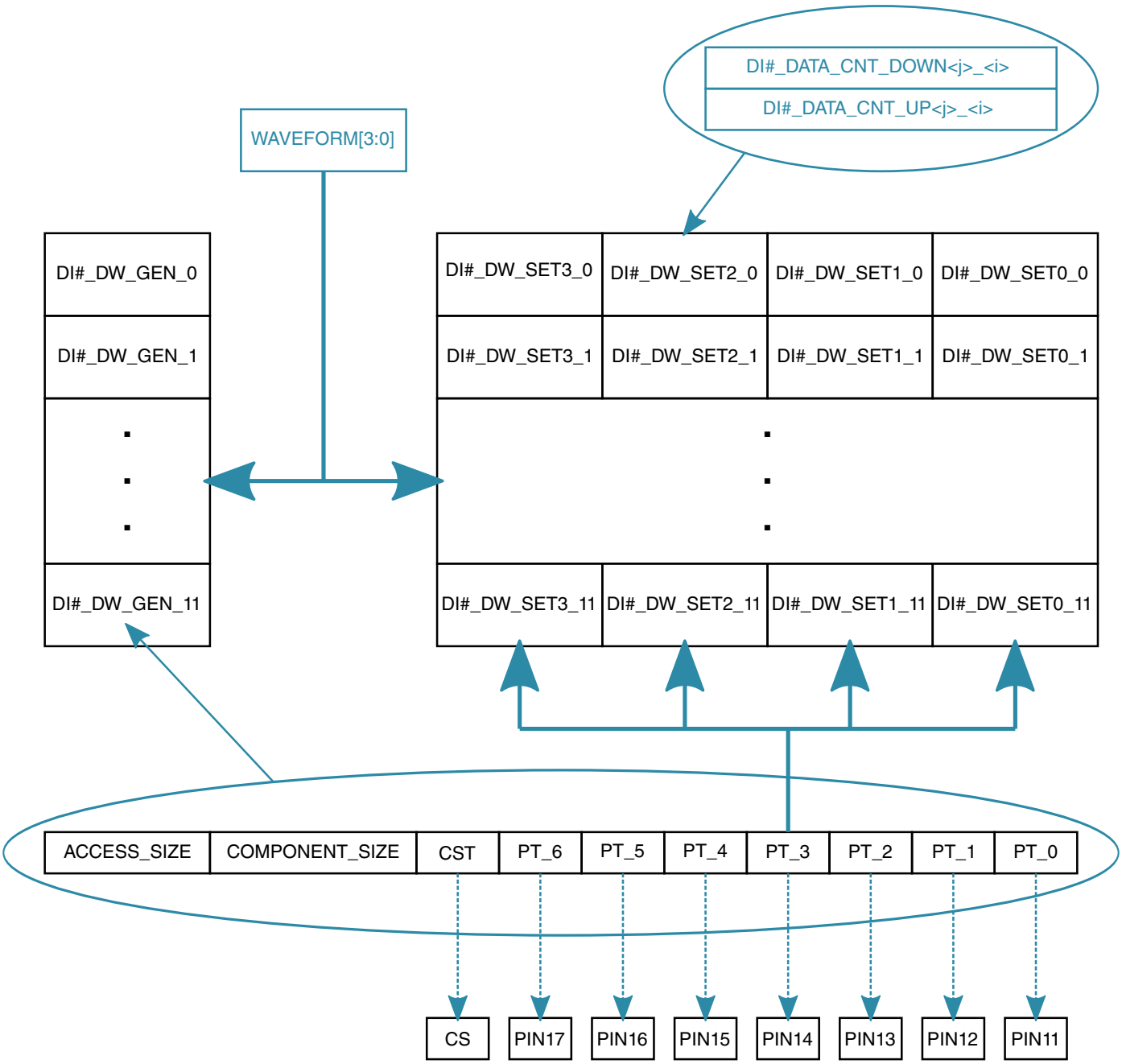


Figure 38-46. Waveform settings for asynchronous interface pins - parallel interface

The DI#_DW_GEN_<i> register includes the data's waveform settings as well. ACCESS_SIZE defines the amount of DI clock cycles that a pixel is valid on the bus. When generic data is sent, this field defines the amount of cycles that the generic data is

valid on the bus. A pixel may be broken into few components. The COMPONNENT_SIZE field defines the amount of cycles that each component is valid on the bus.

The COMPONNENT_SIZE is always smaller or equal to ACCESS_SIZE. For synchronous interface COMPONNENT_SIZE is always equal to ACCESS_SIZE. In case that there's a need for some gap between one type of accesses to another having the COMPONNENT_SIZE smaller than ACCESS_SIZE can be useful (for example read after write accesses that require some gap between them).

38.4.10.5 Low Level Access - LLA

LLA is a ARM platform direct access to the display. For parallel displays the DI's behavior for LLA is the same as any other access to a parallel display. For serial displays the DI allows a DI arbitration bypass. This is done if the DI#_LLA_SER_ACCESS bit is set.

In that case there is a direct access from the DMFC to the DI which bypasses the DC allowing simultaneous access to both serial and parallel interfaces. When DI#_LLA_SER_ACCESS bit is set, the user must not do any other kind of accesses to the serial interface except LLA. When DI#_LLA_SER_ACCESS bit is set the corresponding DI#_WAIT4SERIAL must be clear.

38.4.10.6 Using a mask channel

The IPU is able to provide the windowing function on displays that have data enable control. This is achieved by masking of some screen regions according to a 1-bit/pixel mask read from the memory via IDMAC through channel #44.

When the mask value is zero, the pixel is not displayed. This feature can be used for dual-port smart displays.

38.4.11 Video De Interlacing or Combining Block (VDIC)

The Video De-Interlacer as well as the Combiner have two operation modes:

- De-interlacing: converts an interlaced video stream to progressive order.
- Combining: combines two video/graphics planes and a background color.

The Video De-interlace block (VDIC) deinterlaces standard interlaced video to produce progressive video, that is used for upsizing to HD formats or for display on progressive displays. For VDIC operation three fields are necessary F(n-1),F(n),F(n+1). The F(n-1) field arrived through CSI interface in real time mode or through channel 1 and then stored in FIFO1. The F(n) arrived through channel 2. At least three lines of F(n) are stored in Line Store memory. The F(n+1) arrived through channel 3 and stored in FIFO3. FIFO controllers read data from FIFOs and then data aligned in pixels buffers. From the buffers the data arrived to Line Padding Controller (LPC). The LPC padding missing line at the beginning and end of the frame. The DeInterlacing sub-block (DI) perform the processing. Then data send to IC sub-block for processing or for transferring to external memory.

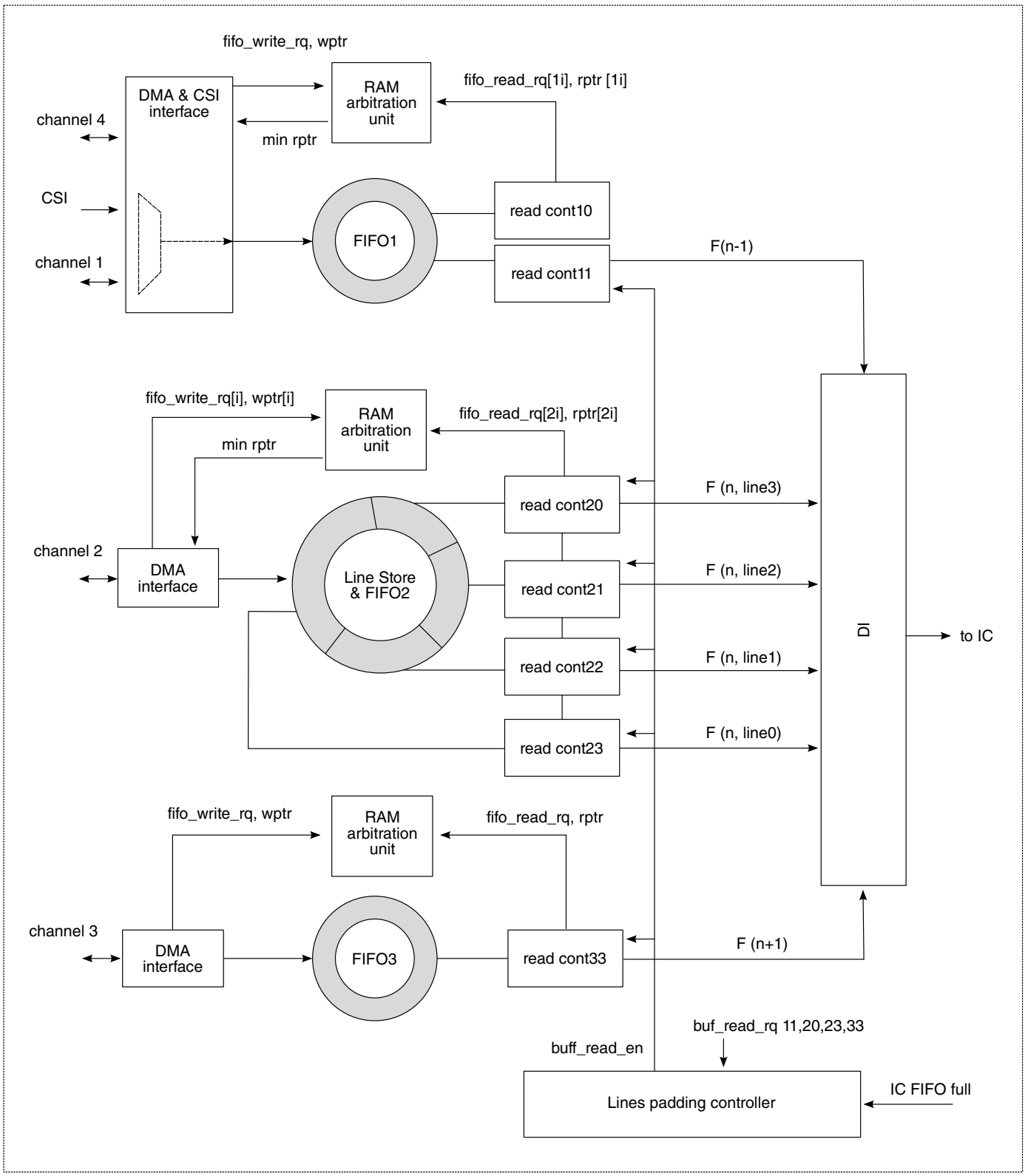


Figure 38-47. VDIC Block Diagram

38.4.11.1 VDIC Features

Key features of the VDIC block include:

- Deintarlacing
  - maximum horizontal resolution 968 pixels
  - maximum pixel rate 75100MP/s
  - Support YUV422 and YUV420 formats
- CSI FIFO mode
- Combining

38.4.11.2 De interlacer (DI) sub-block

The block diagram of the DI block is shown in figure below.

Pipelining is inserted at all stages, so that the design may run at a fast clock speed, if needed.

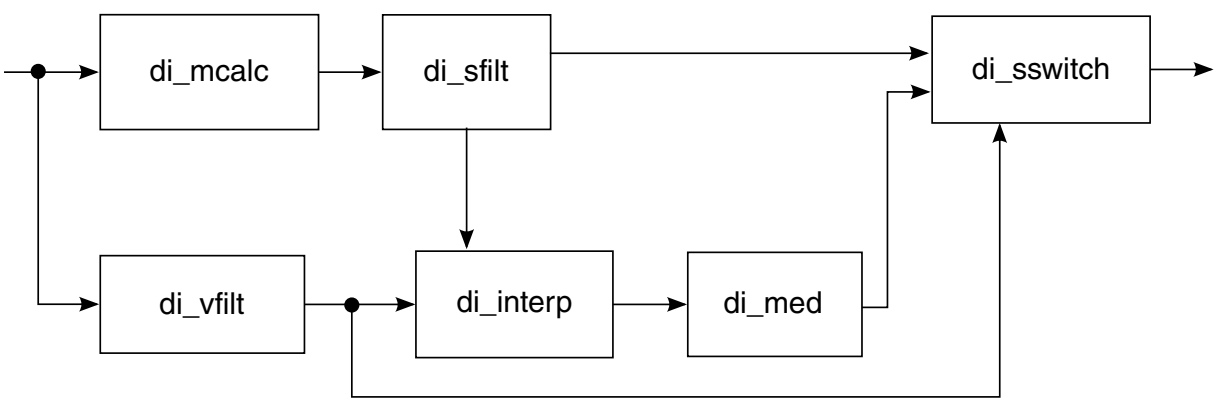


Figure 38-48. DI block diagram

38.4.11.2.1 Vertical Filter Block (di_vfilt)

The di_vfilt block performs spatial vertical filtering of pixels.

It is a four tap vertical filter:

vfilt_out = (-3.0*pix1 + 19.0*pix2 + 19.0*pix3 - 3.0*pix4)/32.0

Where pix1, pix2, pix3, pix4 are four pixels in same horizontal location on four consecutive lines of a field.

vfilt_out is pixel being predicted (between pix2 and pix3)

38.4.11.2.2 Motion Calculator Block (di_mcalc)

The mcalc block estimates the amount of motion for any given pixel by looking at pixel values in current, previous and next fields.

The generic formula used to calculate the estimated motion is:

m = SAT(Ks*(|e-w|/(|e-w|+|n-s|+SPA_DETAIL)))

- Where, n is the pixel above the pixel being predicted
- s is the pixel below the pixel being predicted
- e is the pixel in previous field at same spatial location as the pixel being predicted
- w is the pixel in next field at same spatial location as the pixel being predicted
- m is motion estimation for the pixel being predicted (range from 0 to 1)
- Ks is slope control and decides how quickly the algorithm switches from no motion (m=0) to full motion (m=1)
- SPA_DETAIL is a constant (50) that is added to |n-s|.
- SAT() is saturate at 1 function

The motion calculator block is simplified in a certain respect, by removing the need for a divider, while providing a degree of flexibility. The main motivators for this are the following observations:

1. the above equation defines a set of curves based on the value of |n-s|, but the curves are fairly closely spaced, so that using a granularity of 8 in |n-s| to define specific curves to use gives pretty much the same quality of picture as using all of the curves; and once |n-s| reaches about 120, the effect of an increased |n-s| value is hard to observe
2. once |e-w| gets to about 15, the motion is usually saturated to a value of 1

Based on the above observations, the motion calculator is now implemented with the use of two "ROM"s, using the 4 LSBs of |e-w| and bits 6:3 of |n-s|.

The motion calculator has 3 modes of operation. These modes are defined by the VDI_MOT_SEL field. In case that the user has an information about the motion (For example SW analyzing motion vectors provided by a video decoder) he can select one of the modes listed below. Changing the value of the VDI_MOT_SEL has an affect only on the next frame.

- When VDI_MOT_SEL==2'b01, m_calc is 0 (no motion - use weave)

- When VDI_MOT_SEL==2'b10, We assume high motion and use min(15,delta_t).
- When VDI_MOT_SEL==2'b00, We assume low motion and use sat([0,15],delta_t-8) (delta_t-8 is signed operation).

38.4.11.2.3 Spatial Motion Filter (di_sfilt)

The di_sfilt block spreads motion signal over five pixels:

Mspread = MAX(m3, (0.5*m1 + m2 + m3 + m4 + 0.5*m5)/4.0)

- Where, m3 is motion estimate for current pixel
- m2 is motion estimate for previous pixel
- m4 is motion estimate for next pixel
- m1 is motion estimate for pixel before previous pixel
- m5 is motion estimate for pixel after next pixel

38.4.11.2.4 Interpolated Pixel Calculator Block (di_interp)

The di_interp block uses the motion estimated by the di_sfilt block and computes an interpolated pixel that is weighted sum of the surrounding four pixels (n, s, e, w).

The block performs the following calculations:

```
          if (Mspread <= 0.5) {
i = (1-2*Mspread)*(e+w)/2 + 2*Mspread*vfilt_out
} else {
          i = vfilt_out
}
```

Where, i is the interpolated pixel

n, s, e, w are surrounding pixels as explained earlier

38.4.11.2.5 Median Filter Block (di_med)

The di_med block performs a 5-point median of n, s, e, w and i pixels.

It should be noted that the median required here is not a true 5-point, but can be implemented more efficiently as a 3-point median:

med = MEDIAN(min(max(n,s),max(e,w)), max(min(n,s),min(e,w)), i)

38.4.11.2.6 Soft Switch Block (di_sswitch)

The final output of the deinterlacer is a blend of the median value and the vertical filter, assuming that the pixel data n,s are uncorrelated with the pixel data e,w. By uncorrelated, we mean (max(n,s) < min(e,w)) or (min(n,s) > max(e,w)).

```
          if (Mspread <= 0.5 || not(F)) {
pix_out =    med
} else { /* Mspread > 0.5 */
pix_out = (1-2*(Mspread-0.5))*med + 2*(Mspread-0.5)*vfilt_out
}
```

38.4.11.3 DMA only Mode

In DMA only mode the data is coming from IDMAC only.

38.4.11.4 Real Time Mode

In Real Time Mode the F(n-1) are coming from CSI. The CSI write to FIFO1. The DI sub-block read F(n-1) from processing. In addition IDMAC read the field from FIFO1 and store in external memory. Then stored frames are used as F(n) and F(n+1).

38.4.11.5 CSI only Mode

In CSI only mode VDIC do not perform any processing and used as FIFO only. The data arrived from CSI written to FIFO1. The IDMAC read data from the FIFO1. The FIFO3 and Line store memory are not used. The ID sub-blocks are turned OFF. The CSI only mode can be used as alternative to SMFC, when appropriate.

38.4.11.6 Using Combining in the VDIC

As an alternate function to the de interacing function, the VDIC can perform combining of two planes.

- Both planes have to be at the same color space.
- Overlaying of a single plane over a unified background is supported.
- The planes can be at a different sizes - one plane can be smaller than the other.
- Combining requires a single cycle per output pixel.
- Alpha blending (global or local) is supported.
- Color keying is supported.

The plane's location and size is programmable as shown in the following figure.

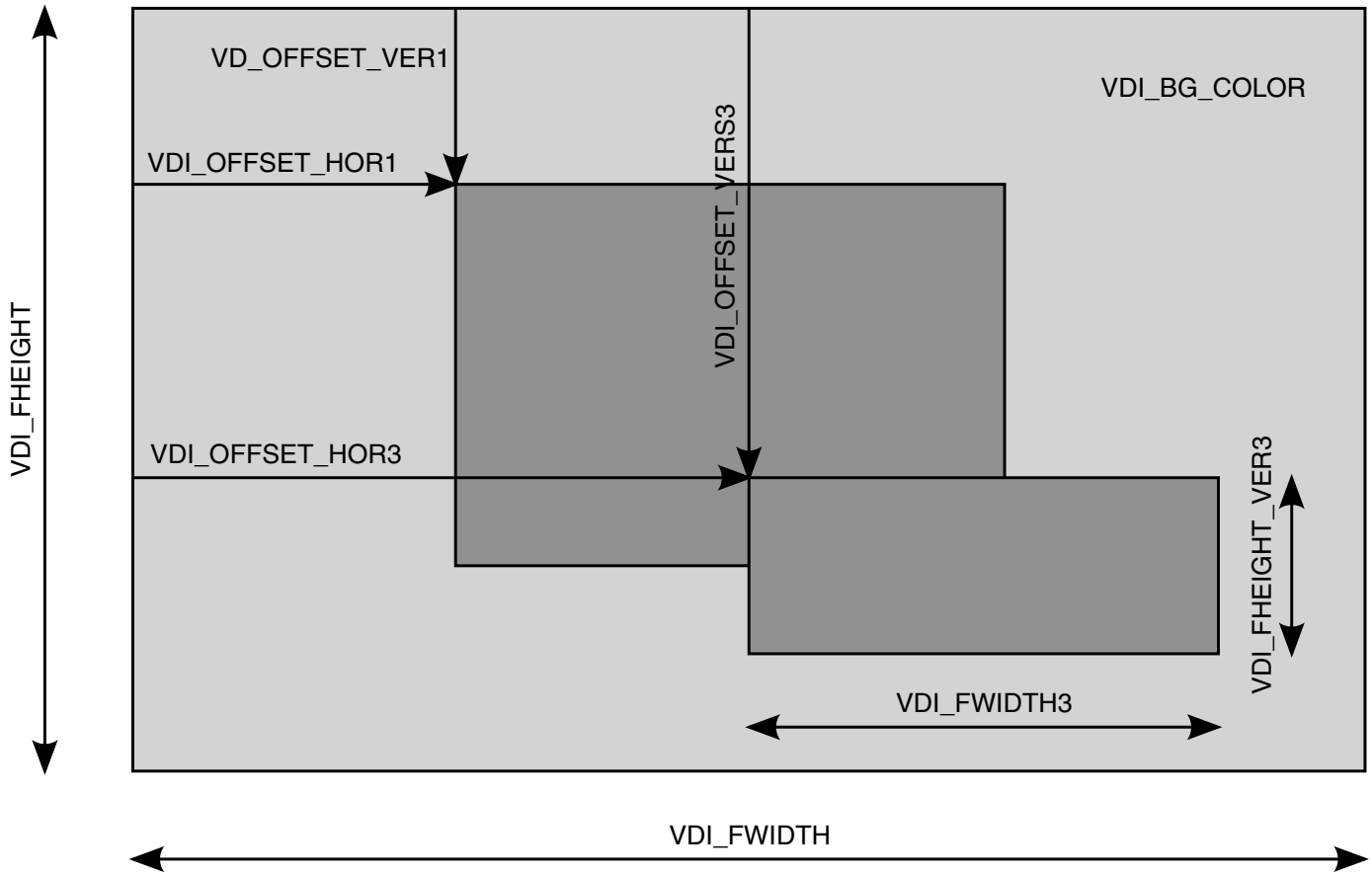


Figure 38-49. The relations between planes of the combining unit

When local alpha is used it should be arrived through channel3 of the VDIC.

The combining equation is:

OP = BG*(1 - alpha) + FG*alpha

Where "OP" is output pixel, "FG" is input pixel of foreground plan arrived thought channel3, BG is input pixel of background plan arrived thought channel1, "alpha" is global or local transparency.

38.4.11.7 VDIC Restrictions

- Maximum output pixel rate is 100 MP/s.
- The output pixel format (YUV422 or YUV420) are equal to input format.
- The VDIC can perform combining or de interlacing. It cannnot perform both functions at the same time.

38.4.12 Control Module (CM)

38.4.12.1 Block Diagram

The CM Block Diagram is shown in the figure below.

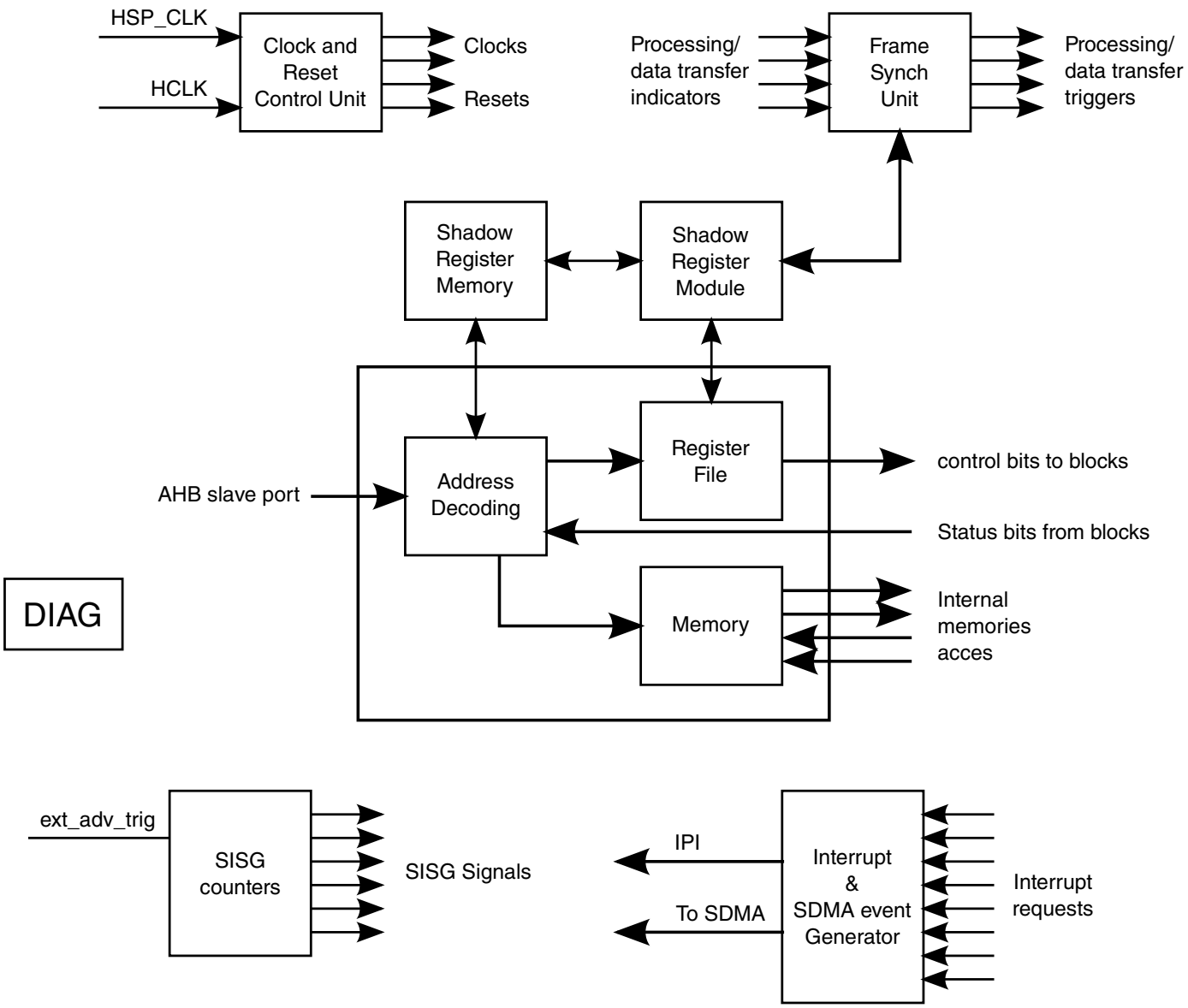


Figure 38-50. CM Block Diagram

The CM consists of the Frame Synchronization Unit (FSU), the Interrupt Generator (IG), the General Configuration Registers (GCR), the Clock and Reset Control Unit (CRCU) and the Shadow Registers Block (SRM).

38.4.12.2 Frame Synchronization Unit

This section details the frame synchronization unit.

38.4.12.2.1 General Description

The FSU provides synchronization of tasks performed by different IPU sub-blocks and ARM platform tasks. This allows to build complex processing flows which are performed automatically (without ARM platform involvement in synchronization of the IPU's tasks).

The FSU supports double buffering of image frames stored in the external memory and allows chaining IPU processing flows in automatic mode.

38.4.12.2.2 Frame Synchronization Flow

1. Initialization

   The ARM platform initializes all the parameters for a task by writing to the GCR and to the parameters memories of each IPU sub-blocks. The initialization must occur before the ARM platform enables the task.

2. Enabling

   After the initialization step has been completed, the ARM platform enables the task by setting its enable bit in an appropriate register.

3. Triggering

   After the task is enabled, the FSU waits for triggering signal. The triggering signals is a combination of the enable bit and the buffer ready signal which can be driven by the ARM platform (DMA_CH_BUF<0/1>_RDY_<#>) and/or by the preceding task (IDMAC_EOF_# or Frame Complete signal from the DC).

   The trigger causes the FSU to invoke the relevant unit to start by assertion of the NEW_FRM_RDY signal. In some cases triggering occurs at the enabling step (when the enable bit is the trigger for the task).

4. Operating

   The triggering step cause the task to move to active mode, this is the operating step. In this step, the FSU monitors the synchronization signals from ARM platform, IDMAC and the corresponding processing units, and controls the units operation. The FSU also controls the IDMAC buffer toggling when double buffer page flipping is used.

The FSU checks at end of each frame if the next frame can be served. If the answer is yes, the FSU stays in active mode with re-sending the <TASK>_NEW_FRM_RDY signal and updating the relevant flags (e.g. DMA<BL>_<#>_CUR_BUF and DMA<BL>_<#>_BUF_RDY). If the answer is no, the FSU moves to pause mode and pauses the task waiting until the next frame can be served.

5. Disabling

   When the task is disabled by the ARM platform (by negating the enable bit), it moves back to non-active mode.

38.4.12.2.3 FSU's fundamentals

Trigger source select

A flow is triggered by a trigger. The trigger may be asserted manually by the ARM platform or may be a result of the completion of the preceding task. Trigger's source select choose the source of the trigger. The trigger means that the data is ready to be processed by the sub-blocks. The trigger source select is defined by the corresponding SRC_SEL bits of the block or task.

Trigger destination select

A block or task that process data needs to know that the following task in the chain is ready to receive the processed data. The user needs to specify what is the destination of the processed data. This is done by setting the corresponding DEST_SEL bits of the block or task.

Double buffering

The IPU supports double buffering in the system's memory. When a flow is processed frame by frame the first frame will be read from one location (BUF0) in the memory, the next frame will be read from another location in the memory (BUF1). The location in the memory of the buffers is defined by the EBA0 and EBA1 parameters of the corresponding channel in the CPMEM. The IDMAC use the correct buffer according to the DMA_CH_CUR_BUF_# signal. The FSU automatically toggles the DMA_CH_CUR_BUF_# to point on the correct buffer to be used by the channel. In order to work in double buffer mode the corresponding DMA_CH_DB_MODE_SEL_# needs to be set.

Alternative flow

Some of the IPU sub-blocks can handle 2 flows via them one is the main flow and the other is the alt flow. In order to support an alternate flow via the same sub-block an alternative configuration should be used by the sub-block. This includes

- Alternate registers including an alternative sub-block's setup - this is handled by the SRM
- Alternate IDMAC settings: parameters in the CPMEM, separate alpha
- Alternate SRC_SEL as the source of the trigger may come from a different function.
- Alternate FSU settings (CUR_BUF, BUF_RDY, DB_MODE_SEL)
- Some of the display sub-blocks alternate settings are handled by programming an alternative set of registers.

The FSU handles the switch to the alternate flow, it controls the update of the alternate configuration and send the appropriate signals to other sub-blocks in the IPU indicating about a need to switch to the alternate configuration.

Once a frame is completed there is a chance that there are 2 buffers ready. One is the next buffer of the current flow and the other is the a buffer in an alternate flow. The FSU will automatically select between the next buffer to handle in a round-robin fashion.

IPU task chaining - Single flow

The diagram below illustrates task chaining in the In this example a single flow is handled

The Frames are coming from the same camera sub-block. The frames are handled frame by Frame. The first frame arriving Frame0 is stored in BUF0 of the "INPUT BUFFER". Once the Frame0 is ready in BUF0, the processing sub-block is triggered and data is read from BUF0 to the processing sub-block. In the meantime, Frame1 coming from the camera is written to BUF1 of the "INPUT BUFFER". The processing sub-block processes the data and stores the first frame on BUF0 of the "OUTPUT BUFFER". Once the processing is done and the frame is ready, the display sub-block is triggered to pick the data and send it to the display. The processing sub-block will start working on the next frame once the data is ready in the "INPUT_BUFFER" and there's a free buffer in the "OUTPUT_BUFFER".

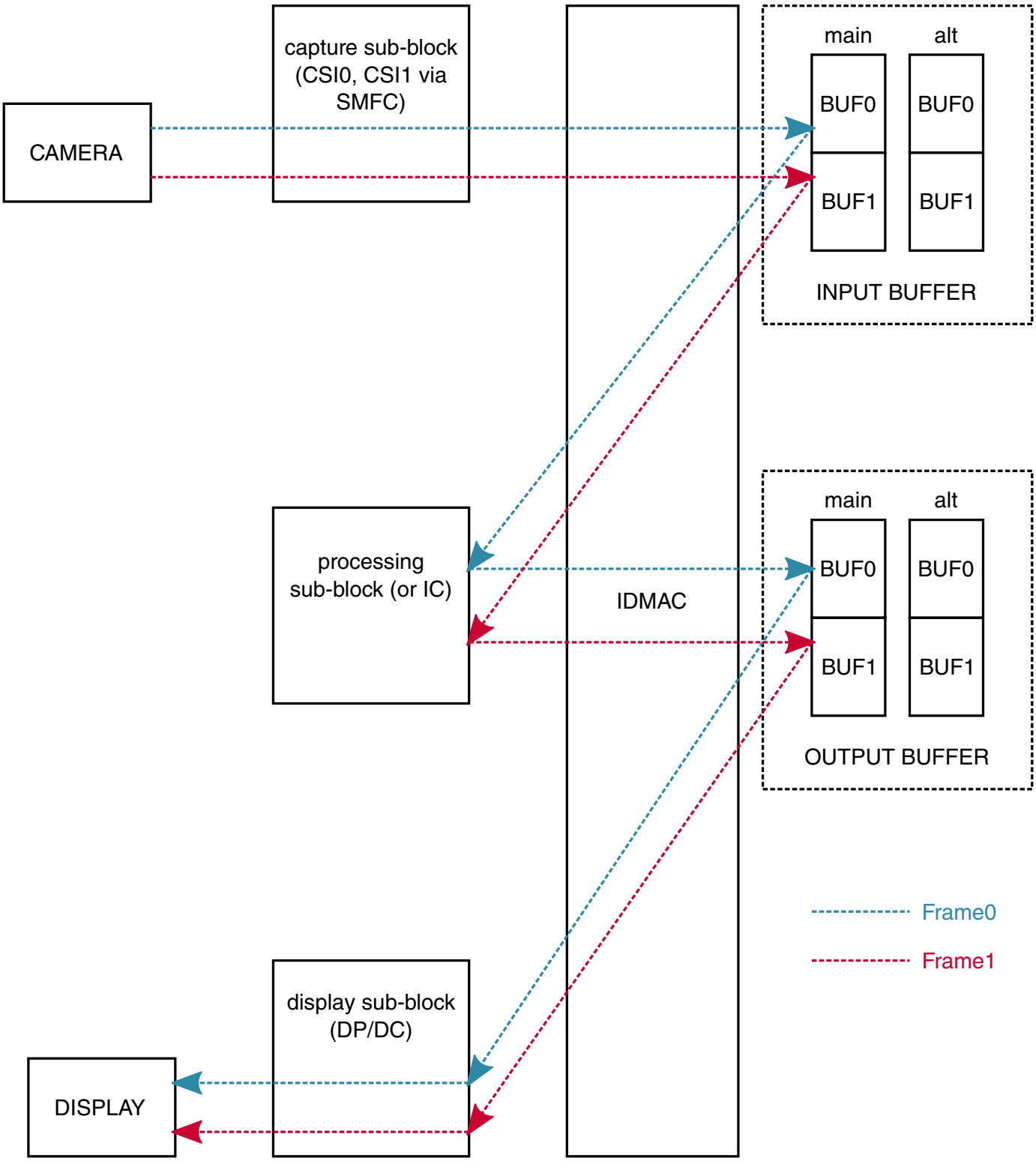


Figure 38-51. IPU tasks' chaining illustration - single flow

IPU task chaining - double flow

The diagram below illustrates task chaining in the In this example double flow is handled. In addition to the flow described on the previous example, another flow is handled via the display sub-blocks. In that case the DC will handle 2 flows. Once sending Frame0 to the display is complete, the FSU will decide in whether to handle Frame1 or Frame0_ALT. The decision is made according the readiness of the other buffers in the memory, in case of 2 ready buffers (main flow and alternate) the FSU will switch between them in a round robin fashion.

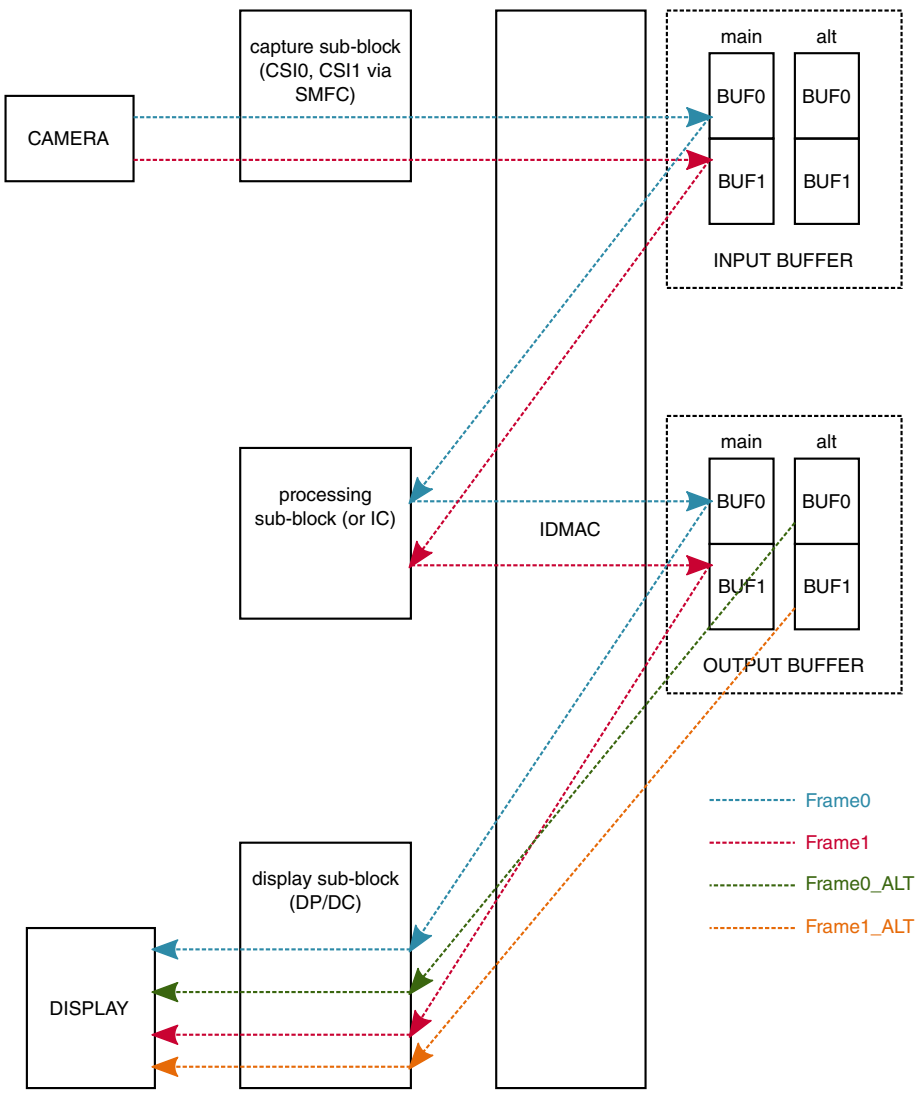


Figure 38-52. IPU tasks' chaining illustration - double flow

38.4.12.2.4 IPU main flows

IPU's flows can be partitioned into 5 groups.

- CF - capture flows

- PF - processing flows
- SF - synchronous display flows
- AF - Asynchronous display flows
- DF - Direct flow from IC to the display

Tasks from different groups can be chained as illustrated below.

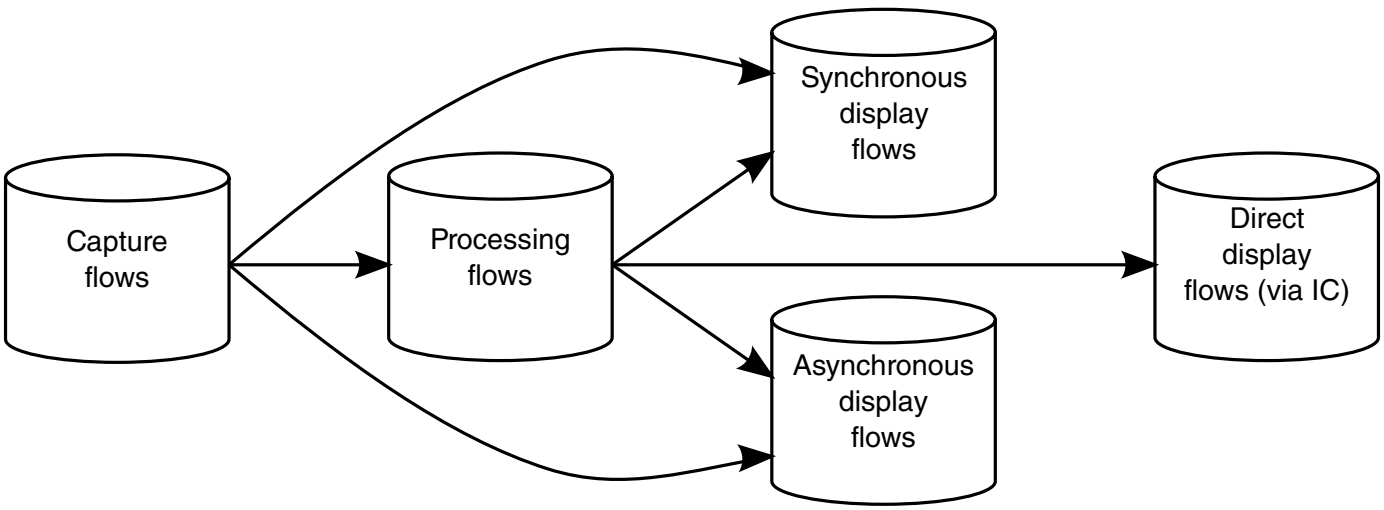


Figure 38-53. IPU task flows chaining

The tables below describe the most important use cases of chaining the IPU tasks. The tables show the main task for each group.

The physical DMA channel is the DMA channel that is used for the main flow - this is the IDMAC channel that is physically connected to the block - the CPMEM parameters should be configured according to the physical channel number. In case of an alternate flow then an alternate entry in the CPMEM should be configured as well

The following table describes capturing flows where data is captured from the sensor and sent to the memory without processing. This flows can be chained to the processing and display flows described on: Table 38-30, Table 38-31, Table 38-32 and on Table 38-33.

Table 38-29. IPU's capture flows

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Capturing image from sensor and storing it in the memory (via SMFC) without processing	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3

Table continues on the next page...

Table 38-29. IPU's capture flows (continued)

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Capturing image from sensor and storing it in the memory (via IC) without processing	CSI0 or CSI1 --> MEM	---	---	IDMAC_CH_5
	Capturing interlaced input and storing it in the memory (via VDIC) while performing video de-interlacing in the VDIC.	CSI0 or CSI1 -> VDIC --> MEM		IDMAC_CH_9 IDMAC_CH_10	IDMAC_CH_5 IDMAC_CH_13
		Interlaced input coming from one of the CSIs is sent to the memory without processing via channel #13. In addition 2 more inputs are read from the memory via channels 9 and 10. The processed image is written to the memory via ch 5 in progressive scan mode.			

NOTE

Register IPUx_SMFC_MAP is used to map CSI0 and CSI1 inputs to one of the four IDMAC channels. See Sensor Multi FIFO Controller (SMFC) for additional information.

The following table describe processing flows via the IC . This flows can start from the memory, can be chained to a capturing flow described on Table 38-29. The target of this flow can be chained to the display flows described on Table 38-31 and on Table 38-32.

Table 38-30. IPU's Processing flows

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Preprocessing image from sensor for encoding	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (PRP ENC) --> MEM	IDMAC_CH_12	---	IDMAC_CH_20
	Preprocessing image from memory for encoding	MEM --> IC (PRP ENC) --> MEM	IDMAC_CH_12	---	IDMAC_CH_20
	Preprocessing image from sensor for encoding	CSI0 or CSI1 --> IC (PRP ENC) --> MEM	---	---	IDMAC_CH_20
	Preprocessing + rotation of image from sensor for encoding	CSI0 or CSI1 -->SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (PRP ENC) --> MEM	IDMAC_CH_12	---	IDMAC_CH_20
		MEM --> IC (ROT ENC) --> MEM	IDMAC_CH_45	---	IDMAC_CH_48

Table continues on the next page...

Table 38-30. IPU's Processing flows (continued)

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Preprocessing + rotation of image from memory for encoding	MEM --> IC (PRP ENC) --> MEM	IDMAC_CH_12	---	IDMAC_CH_20
		MEM --> IC (ROT ENC) --> MEM	IDMAC_CH_45	---	IDMAC_CH_48
	Rotation and preprocessing of image from sensor for encoding	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (ROT ENC) --> MEM	IDMAC_CH_45	---	IDMAC_CH_48
		MEM --> IC (PRP ENC) --> MEM	IDMAC_CH_12	---	IDMAC_CH_20
	Preprocessing image from sensor for viewfinder	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (PRP VF) --> MEM	IDMAC_CH_12	IDMAC_CH_14	IDMAC_CH_21
	Preprocessing image from memory for viewfinder	MEM --> IC (PRP VF) --> MEM	IDMAC_CH_12	IDMAC_CH_14	IDMAC_CH_21
	Preprocessing and rotation of image from sensor for viewfinder	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (PRP VF) --> MEM	IDMAC_CH_12	IDMAC_CH_14	IDMAC_CH_21
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
	Rotation and preprocessing of image from sensor for viewfinder	CSI0 or CSI1 -->SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
		MEM --> IC (PRP VF) --> MEM	IDMAC_CH_12	IDMAC_CH_14	IDMAC_CH_21
	Preprocessing and rotation of image from sensor for viewfinder	MEM --> IC (PRP VF) --> MEM	IDMAC_CH_12	IDMAC_CH_14	IDMAC_CH_21
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
	Preprocessing image from sensor	CSI0 or CSI1 --> IC (PRP VF) --> MEM	---	IDMAC_CH_14	IDMAC_CH_21
	Preprocessing and rotation of image from sensor	CSI0 or CSI1 --> IC (PRP VF) --> MEM	---	IDMAC_CH_14	IDMAC_CH_21
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
	Postprocessing image	MEM --> IC (PP) --> MEM	IDMAC_CH_11	IDMAC_CH_15	IDMAC_CH_22
	Postprocessing and rotation of image	MEM --> IC (PP) --> MEM	IDMAC_CH_11	IDMAC_CH_15	IDMAC_CH_22
		MEM --> IC (ROT PP) -->MEM	IDMAC_CH_47	---	IDMAC_CH_50

Table continues on the next page...

Table 38-30. IPU's Processing flows (continued)

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Rotation and postprocessing of image	MEM --> IC (ROT PP) -->MEM	IDMAC_CH_47	---	IDMAC_CH_50
		MEM --> IC (PP) --> MEM	IDMAC_CH_11	IDMAC_CH_15	IDMAC_CH_22
	Postprocessing image from sensor	CSI0 or CSI1 -->SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (PP) --> MEM	IDMAC_CH_11	IDMAC_CH_15	IDMAC_CH_22
	video de-interlacing in the VDIC.	MEM -> VDIC --> MEM	IDMAC_CH_8 IDMAC_CH_9 IDMAC_CH_10	--	IDMAC_CH_5
		Interlaced input coming from the memory via 3 channels 8, 9 and 10. The processed image is written to the memory via ch 5 in progressive scan mode.			
	video de-interlacing in the VDIC. Then processing in the IC.	MEM -> VDIC --> IC (PRP VF) --> MEM	IDMAC_CH_8 IDMAC_CH_9 IDMAC_CH_10	IDMAC_CH_14	IDMAC_CH_21
		Interlaced input coming from the memory via 3 channels 8, 9 and 10. The processed image is sent to the IC for further processing then it is sent to the memory via ch 21			
	video de-interlacing in the VDIC, then preprocessing and rotation of image coming from memory	MEM --> VDIC --> IC (PRP VF) --> MEM	IDMAC_CH_8 IDMAC_CH_9 IDMAC_CH_10	IDMAC_CH_14	IDMAC_CH_21
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
	video de-interlacing in the VDIC, then preprocessing and rotation of image coming from CSI	CSI0 or CSI1 --> VDIC--> IC (PRP VF) --> MEM	IDMAC_CH_9 IDMAC_CH_10	IDMAC_CH_14	IDMAC_CH_13 IDMAC_CH_21
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
	Combining in the VDIC.	MEM -> VDIC --> MEM	IDMAC_CH_25I DMAC_CH_26	--	IDMAC_CH_5
		Plane3 is coming on IDMAC_CH_25; Additional plane (plane1) may come from IDMAC_CH_26.			
	Combining in the VDIC. Then processing in the IC.	MEM -> VDIC --> IC (PRP VF) --> MEM	IDMAC_CH_25I DMAC_CH_26	IDMAC_CH_14	IDMAC_CH_21
		Plane3 is coming on IDMAC_CH_25; Additional plane (plane1) may come from IDMAC_CH_26.			
	Combining in the VDIC, then preprocessing and rotation	MEM --> VDIC --> IC (PRP VF) --> MEM	IDMAC_CH_25I DMAC_CH_26	IDMAC_CH_14	IDMAC_CH_21
		MEM --> IC (ROT VF) --> MEM		---	IDMAC_CH_49
		Plane3 is coming on IDMAC_CH_25; Additional plane (plane1) may come from IDMAC_CH_26.			
	Rotation of an image and Combining in the VDIC	MEM --> IC (ROT VF) -->MEM	IDMAC_CH_46	---	IDMAC_CH_49
		MEM -> VDIC --> MEM	IDMAC_CH_25I DMAC_CH_26	--	IDMAC_CH_5

Table continues on the next page...

Table 38-30. IPU's Processing flows (continued)

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
		Plane3 is rotated on the VF task. Then it sent to the VDIC IDMAC_CH_25; Additional plane (plane1) may come from IDMAC_CH_26.			
	Rotation of an image and Combining in the VDIC	MEM --> IC (ROT ENC) -->MEM	IDMAC_CH_45	---	IDMAC_CH_48
		MEM -> VDIC --> MEM	IDMAC_CH_25I DMAC_CH_26	--	IDMAC_CH_5
		Plane1 is rotated on the ENC task. Then it sent to the VDIC IDMAC_CH_26; In additional plane3 is coming from IDMAC_CH_25			
	Rotation of two images and Combining in the VDIC	MEM --> IC (ROT VF) -->MEM	IDMAC_CH_46	---	IDMAC_CH_49
		MEM --> IC (ROT ENC) -->MEM	IDMAC_CH_45	---	IDMAC_CH_48
		MEM -> VDIC --> MEM	IDMAC_CH_25I DMAC_CH_26	--	IDMAC_CH_5
		Plane3 is rotated on the VF task. Plane1 is rotated on the ENC task.			
	Rotation of two images and Combining in the VDIC. Then processing in the IC.	MEM --> IC (ROT VF) -->MEM	IDMAC_CH_46	---	IDMAC_CH_49
		MEM --> IC (ROT ENC) -->MEM	IDMAC_CH_45	---	IDMAC_CH_48
		MEM -> VDIC --> IC (PRP VF) --> MEM	IDMAC_CH_25I DMAC_CH_26	IDMAC_CH_14	IDMAC_CH_21
		Plane3 is rotated on the VF task. Plane1 is rotated on the ENC task.			

The following table describes the synchronous display flows, this flows can be chained to the capture, processing and direct flows described on Table 38-29,Table 38-29and on Table 38-33

Table 38-31. IPU synchronous display flows

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Synchronous display refresh via DC	MEM --> DC	IDMAC_CH_28	---	---
	Synchronous display refresh via DP	MEM --> DP	IDMAC_CH_23	---	---
	Synchronous display refresh via DC	MEM --> DC	IDMAC_CH_28	IDMAC_CH_44	---
		Comment: IDMAC_CH_44 is an optional mask Channel			
	Synchronous display refresh via DP	MEM --> DP	IDMAC_CH_23	IDMAC_CH_44	---
		Comment: IDMAC_CH_44 is an optional mask Channel			
	Synchronous display refresh via DP + combining in the DP	MEM --> DP SYNC(BG/FG)	IDMAC_CH_23	IDMAC_CH_27	---
		Comment: IDMAC_CH_23 is the main channel; IDMAC_CH_27 is used for combining of another plane.			

Table continues on the next page...

Table 38-31. IPU synchronous display flows (continued)

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Synchronous display refresh via DP	MEM --> DP SYNC(BG/FG)	IDMAC_CH_23/ IDMAC_CH_27	IDMAC_CH_44 (mask)	---
		Comment: IDMAC_CH_23 is the main channel; IDMAC_CH_27 is optional and can be used for combining of another plane. IDMAC_CH_44 is an optional mask Channel			

The following table describes the asynchronous display flows, this flows can be chained to the capture, processing and direct flows described onTable 38-29,Table 38-29and on Table 38-33

Table 38-32. IPU Asynchronous display flows

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Asynchronous display refresh via DP	MEM --> DC	IDMAC_CH_24	---	---
	Asynchronous display refresh via DP	MEM --> DC	IDMAC_CH_24/ IDMAC_CH_29	---	---
		Comment: IDMAC_CH_29 is another input to the DP to be combined with data coming from IDMAC_CH_24			
	Asynchronous display refresh via DP in command buffer mode	MEM --> DC	IDMAC_CH_24/ IDMAC_CH_29	IDMAC_CH_42 (command)	---
		Comment: IDMAC_CH_29 is another input to the DP to be combined with data coming from IDMAC_CH_24			
	Asynchronous display refresh via DC	MEM --> DC	IDMAC_CH_28	---	---
	Asynchronous display refresh via DC	MEM --> DC	IDMAC_CH_41	---	---
	Asynchronous display refresh via DC	MEM --> DC	IDMAC_CH_28	IDMAC_CH_42(command)	---
		Comment: IDMAC_CH_42 can be optionally used as command channel associated with IDMAC_CH_28			
	Asynchronous display refresh via DC	MEM --> DC	IDMAC_CH_41	IDMAC_CH_43(command)	---
		Comment: IDMAC_CH_43 can be optionally used as command channel associated with IDMAC_CH_41			
	Asynchronous display refresh via DP	MEM --> DP ASYNC	IDMAC_CH_24	IDMAC_CH_42(command)	---
		Comment: IDMAC_CH_42 can be optionally used as command channel associated with IDMAC_CH_24			

Table continues on the next page...

Table 38-32. IPU Asynchronous display flows (continued)

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Asynchronous display refresh via DP	MEM --> DC	IDMAC_CH_24	IDMAC_CH_43 (command)	---
		Comment: IDMAC_CH_43 can be optionally used as command channel associated with IDMAC_CH_24			
	Reading data from asynchronous display	DC --> MEM	---	---	IDMAC_CH_40

The table below describes direct flows via the IC. These are processing flows via the IC where the output is sent directly to the DMFC. Direct camera to display flow is possible when the frame rate of the source and the destination is the same (typically the target display will be asynchronous display, where the display is updated at the rate of the source). From that point, any of the display flows described on Table 38-31 and on Table 38-32 can be chained.

Table 38-33. IPU direct flows to the display via the IC

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Rotation and preprocessing of image from sensor for viewfinder and displaying it on synchronous display via DP	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
		MEM --> IC (PRP VF) --> DMFC	IDMAC_CH_12	IDMAC_CH_14	Direct to the DMFC
		MEM --> DP SYNC (BG/FG)	IDMAC_CH_23/ IDMAC_CH_27	IDMAC_CH_44 (mask)	---
	Rotation and preprocessing of image from sensor for viewfinder and displaying it on asynchronous display via DP	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
		MEM --> IC (PRP VF) --> DMFC	IDMAC_CH_12	IDMAC_CH_14	Direct to the DMFC
		MEM --> DP SYNC (BG/FG)	IDMAC_CH_24/ IDMAC_CH_29	command channel	---
	Preprocessing image from sensor for viewfinder and direct displaying it on asynchronous display	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3

Table continues on the next page...

Table 38-33. IPU direct flows to the display via the IC (continued)

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
		MEM --> IC (PRP VF) --> DC	IDMAC_CH_12	IDMAC_CH_14	---
	Preprocessing image from sensor for viewfinder and direct displaying it on asynchronous display	CSI0 or CSI1 --> IC (PRP VF) --> DC	---	IDMAC_CH_14	IDMAC_CH_21
	Preprocessing image from sensor for viewfinder and direct displaying it on asynchronous display via DP	CSI0 or CSI1 --> IC (PRP VF) --> DP ASYNC	---	IDMAC_CH_14	IDMAC_CH_21
	Capturing interlaced input via CSI, then perform video de-interlacing in the VDIC. Then processing in the IC and direct displaying it on asynchronous display via DP	CSI0 or CSI1 -> VDIC --> IC (PRP VF) --> DP ASYNC	--	IDMAC_CH_9 IDMAC_CH_10 IDMAC_CH_14	IDMAC_CH_13 IDMAC_CH_21
		Interlaced input coming from one of the CSIs is sent to the memory without processing via channel #13. In addition 2 more inputs are read from the memory via channels 9 and 10. The processed image is sent to the IC for further processing then it is sent to the display			
	Capturing interlaced input via CSI, then perform video de-interlacing in the VDIC. Then processing in the IC and direct displaying it on asynchronous display	CSI0 or CSI1 -> VDIC --> IC (PRP VF) --> DC	--	IDMAC_CH_9 IDMAC_CH_10 IDMAC_CH_14	IDMAC_CH_13 IDMAC_CH_21
		Interlaced input coming from one of the CSIs is sent to the memory without processing via channel #13. In addition 2 more inputs are read from the memory via channels 9 and 10. The processed image is sent to the IC for further processing then it is sent to the display			

38.4.12.2.5 Sub-Frame Double-Buffering (Band Mode)

Page-flip double buffering is performed using full-frame buffers.

In addition IPU supports also page-flip double buffering using smaller buffers, each containing 4/8/16/32/64/128/256 rows of pixels. This allows the use of internal memory for buffering.

This mode can be supported by the following modifications (relative to full-frame buffers):

The address in system memory is generated by the same formula, but inserting the full row number, only k LSB's (for a band of 2^k rows) are inserted (e.g. [2:0] for 8 rows).

Page flip is triggered at the end of each band (when the k LSB's are all 1) and not only at the end of the frame. This flip may be accompanied by an ARM/SDMA interrupt, if synchronization with other modules (e.g. VPU or ARM) is needed.

The channels that can work in this mode are controlled by the corresponding IDMAC_BNDM_EN_<i>

bit. The BNDM parameter of the corresponding IDMAC channel has to be set as well.

38.4.12.2.6 Automatic Window Refresh

By programming the <>_SRC_SEL bits of the corresponding flow to autoref mode, automatic refresh of a window on the smart display is enabled. This means that the flow is triggered any time the refresh counter completes its counting. The refresh period is defined via the AUTOREF_PER field (the time unit is 2^{17} periods of the HSP_CLK clock).

The actual value of refresh period is equal to:

$T_{HSP}*2^{17}$ *(AUTO_REF_PER+1).

38.4.12.2.7 VPU Sub Frame Synchronization

IPU can synchronize with the VPU at a sub frame level. This allows the VPU to start encoding a frame before IPU completes writing the whole frame of the source video (from camera sensor) into the frame buffer and by that reduce latency. This feature is called IPU-VPU sub-frame synchronization.

The IPU sends an end of line indication to the VPU, so the VPU can synchronize to the IPU on a line by line basis.

The IDMAC asserts an end-of-line indication of the corresponding IDMAC channel. The assertion of this indication is enabled by the EOLI bit at the CPMEM. The indication is asserted once the entire row is written to the memory. The signal sent to the VPU is selected by the IPU_VPU_DEST_SEL field. VPU sub frame synchronization feature is available for IDMAC channels 0,2 and 21.

38.4.12.2.8 IPU VDOA synchronization

IPU can interact to a companion block called VDOA.

VDOA (Video Data Order Adapter) is used to re-order video data from the "tiled" order used by the VPU to the conventional raster-scan order needed by the IPU.

The VDOA can transfer its output to the IPU through internal memory, containing a band double buffer.

This tight double-buffering synchronization is performed without MCU involvement, using dedicated signals between the VDOA and IPU and the same protocol as used between two IPU DMA channels.

VDOA notifies the IPU which is the current band to read. IPU notifies the VDOA when the current band is read.

This synchronization is supported for VDI and IC(PP). The target destination of the VDOA data is defined by VDOA_DEST_SEL parameter. The PP_SRC_SEL or VDI_SRC_SEL has to be programmed as well to select VDOA as the source of the task.

For more details about IDMAC's band mode support Sub-Frame Double-Buffering (Band Mode).

The IPU settings and the VDOA settings has to be the same. In particular

- The band height defined at the IPU and at the VDOA must match
- The Number of Frames used by the VDI as defined by the VDI_MOT_SEL has to match the VDOA settings.

38.4.12.3 Interrupt Generator

The IG produces two interrupts to the ARM platform - the functional interrupt and the error interrupt. All of the interrupts are maskable.

The following table describes the functional interrupts.

Table 38-34. Functional Interrupts Summary

Location	Sub-blocks	Interrupt status bit name	Description
IPU_INT_STAT_1[0]	IDMAC	IDMAC_EOF_0	-
IPU_INT_STAT_1[1]	IDMAC	IDMAC_EOF_1	-
IPU_INT_STAT_1[2]	IDMAC	IDMAC_EOF_2	-
IPU_INT_STAT_1[3]	IDMAC	IDMAC_EOF_3	-
IPU_INT_STAT_1[5]	IDMAC	IDMAC_EOF_5	-
IPU_INT_STAT_1[11]	IDMAC	IDMAC_EOF_11	-
IPU_INT_STAT_1[12]	IDMAC	IDMAC_EOF_12	-
IPU_INT_STAT_1[14]	IDMAC	IDMAC_EOF_14	-
IPU_INT_STAT_1[15]	IDMAC	IDMAC_EOF_15	-
IPU_INT_STAT_1[17]	IDMAC	IDMAC_EOF_17	-
IPU_INT_STAT_1[18]	IDMAC	IDMAC_EOF_18	-
IPU_INT_STAT_1[20]	IDMAC	IDMAC_EOF_20	-
IPU_INT_STAT_1[21]	IDMAC	IDMAC_EOF_21	-

Table continues on the next page...

Table 38-34. Functional Interrupts Summary (continued)

Location	Sub-blocks	Interrupt status bit name	Description
IPU_INT_STAT_1[22]	IDMAC	IDMAC_EOF_22	-
IPU_INT_STAT_1[23]	IDMAC	IDMAC_EOF_23	-
IPU_INT_STAT_1[24]	IDMAC	IDMAC_EOF_24	-
IPU_INT_STAT_1[27]	IDMAC	IDMAC_EOF_27	-
IPU_INT_STAT_1[28]	IDMAC	IDMAC_EOF_28	-
IPU_INT_STAT_1[29]	IDMAC	IDMAC_EOF_29	-
IPU_INT_STAT_1[31]	IDMAC	IDMAC_EOF_31	-
IPU_INT_STAT_2[1]	IDMAC	IDMAC_EOF_33	-
IPU_INT_STAT_2[8]	IDMAC	IDMAC_EOF_40	-
IPU_INT_STAT_2[9]	IDMAC	IDMAC_EOF_41	-
IPU_INT_STAT_2[10]	IDMAC	IDMAC_EOF_42	-
IPU_INT_STAT_2[11]	IDMAC	IDMAC_EOF_43	-
IPU_INT_STAT_2[12]	IDMAC	IDMAC_EOF_44	-
IPU_INT_STAT_2[13]	IDMAC	IDMAC_EOF_45	-
IPU_INT_STAT_2[14]	IDMAC	IDMAC_EOF_46	-
IPU_INT_STAT_2[15]	IDMAC	IDMAC_EOF_47	-
IPU_INT_STAT_2[16]	IDMAC	IDMAC_EOF_48	-
IPU_INT_STAT_2[17]	IDMAC	IDMAC_EOF_49	-
IPU_INT_STAT_2[18]	IDMAC	IDMAC_EOF_50	-
IPU_INT_STAT_2[19]	IDMAC	IDMAC_EOF_51	-
IPU_INT_STAT_2[20]	IDMAC	IDMAC_EOF_52	-
IPU_INT_STAT_3[0]	IDMAC	IDMAC_NFACK_0	-
IPU_INT_STAT_3[1]	IDMAC	IDMAC_NFACK_1	-
IPU_INT_STAT_3[2]	IDMAC	IDMAC_NFACK_2	-
IPU_INT_STAT_3[3]	IDMAC	IDMAC_NFACK_3	-
IPU_INT_STAT_3[5]	IDMAC	IDMAC_NFACK_5	-
IPU_INT_STAT_3[8]	IDMAC	IDMAC_NFACK_8	-
IPU_INT_STAT_3[9]	IDMAC	IDMAC_NFACK_9	-
IPU_INT_STAT_3[10]	IDMAC	IDMAC_NFACK_10	-
IPU_INT_STAT_3[11]	IDMAC	IDMAC_NFACK_11	-
IPU_INT_STAT_3[12]	IDMAC	IDMAC_NFACK_12	-
IPU_INT_STAT_3[13]	IDMAC	IDMAC_NFACK_13	-
IPU_INT_STAT_3[14]	IDMAC	IDMAC_NFACK_14	-
IPU_INT_STAT_3[15]	IDMAC	IDMAC_NFACK_15	-
IPU_INT_STAT_3[17]	IDMAC	IDMAC_NFACK_17	-
IPU_INT_STAT_3[18]	IDMAC	IDMAC_NFACK_18	-
IPU_INT_STAT_3[20]	IDMAC	IDMAC_NFACK_20	-
IPU_INT_STAT_3[21]	IDMAC	IDMAC_NFACK_21	-
IPU_INT_STAT_3[22]	IDMAC	IDMAC_NFACK_22	-

Table continues on the next page...

Table 38-34. Functional Interrupts Summary (continued)

Location	Sub-blocks	Interrupt status bit name	Description
IPU_INT_STAT_3[23]	IDMAC	IDMAC_NFACK_23	-
IPU_INT_STAT_3[24]	IDMAC	IDMAC_NFACK_24	-
IPU_INT_STAT_3[27]	IDMAC	IDMAC_NFACK_27	-
IPU_INT_STAT_3[28]	IDMAC	IDMAC_NFACK_28	-
IPU_INT_STAT_3[29]	IDMAC	IDMAC_NFACK_29	-
IPU_INT_STAT_3[31]	IDMAC	IDMAC_NFACK_31	-
IPU_INT_STAT_4[1]	IDMAC	IDMAC_NFACK_33	-
IPU_INT_STAT_4[8]	IDMAC	IDMAC_NFACK_40	-
IPU_INT_STAT_4[9]	IDMAC	IDMAC_NFACK_41	-
IPU_INT_STAT_4[10]	IDMAC	IDMAC_NFACK_42	-
IPU_INT_STAT_4[11]	IDMAC	IDMAC_NFACK_43	-
IPU_INT_STAT_4[12]	IDMAC	IDMAC_NFACK_44	-
IPU_INT_STAT_4[13]	IDMAC	IDMAC_NFACK_45	-
IPU_INT_STAT_4[14]	IDMAC	IDMAC_NFACK_46	-
IPU_INT_STAT_4[15]	IDMAC	IDMAC_NFACK_47	-
IPU_INT_STAT_4[16]	IDMAC	IDMAC_NFACK_48	-
IPU_INT_STAT_4[17]	IDMAC	IDMAC_NFACK_49	-
IPU_INT_STAT_4[18]	IDMAC	IDMAC_NFACK_50	-
IPU_INT_STAT_4[19]	IDMAC	IDMAC_NFACK_51	-
IPU_INT_STAT_4[20]	IDMAC	IDMAC_NFACK_52	-
IPU_INT_STAT_7[23]	IDMAC	IDMAC_EOS_23	-
IPU_INT_STAT_7[24]	IDMAC	IDMAC_EOS_24	-
IPU_INT_STAT_7[27]	IDMAC	IDMAC_EOS_27	-
IPU_INT_STAT_7[28]	IDMAC	IDMAC_EOS_28	-
IPU_INT_STAT_7[29]	IDMAC	IDMAC_EOS_29	-
IPU_INT_STAT_7[31]	IDMAC	IDMAC_EOS_31	-
IPU_INT_STAT_8[2]	IDMAC	IDMAC_EOS_33	-
IPU_INT_STAT_8[9]	IDMAC	IDMAC_EOS_41	-
IPU_INT_STAT_8[10]	IDMAC	IDMAC_EOS_42	-
IPU_INT_STAT_8[11]	IDMAC	IDMAC_EOS_43	-
IPU_INT_STAT_8[12]	IDMAC	IDMAC_EOS_44	-
IPU_INT_STAT_8[19]	IDMAC	IDMAC_EOS_51	-
IPU_INT_STAT_8[20]	IDMAC	IDMAC_EOS_52	-
IPU_INT_STAT_11[0]	IDMAC	IDMAC_EOBND_0	-
IPU_INT_STAT_11[1]	IDMAC	IDMAC_EOBND_1	-
IPU_INT_STAT_11[2]	IDMAC	IDMAC_EOBND_2	-
IPU_INT_STAT_11[3]	IDMAC	IDMAC_EOBND_3	-
IPU_INT_STAT_11[5]	IDMAC	IDMAC_EOBND_5	-
IPU_INT_STAT_11[11]	IDMAC	IDMAC_EOBND_11	-

Table continues on the next page...

Table 38-34. Functional Interrupts Summary (continued)

Location	Sub-blocks	Interrupt status bit name	Description
IPU_INT_STAT_11[12]	IDMAC	IDMAC_EOBND_12	-
IPU_INT_STAT_11[20]	IDMAC	IDMAC_EOBND_20	-
IPU_INT_STAT_11[21]	IDMAC	IDMAC_EOBND_21	-
IPU_INT_STAT_11[22]	IDMAC	IDMAC_EOBND_22	-
IPU_INT_STAT_12[13]	IDMAC	IDMAC_EOBND_45	-
IPU_INT_STAT_12[14]	IDMAC	IDMAC_EOBND_46	-
IPU_INT_STAT_12[15]	IDMAC	IDMAC_EOBND_47	-
IPU_INT_STAT_12[16]	IDMAC	IDMAC_EOBND_48	-
IPU_INT_STAT_12[17]	IDMAC	IDMAC_EOBND_49	-
IPU_INT_STAT_12[18]	IDMAC	IDMAC_EOBND_50	-
IPU_INT_STAT_13[0]	IDMAC	IDMAC_TH_0	-
IPU_INT_STAT_13[1]	IDMAC	IDMAC_TH_1	-
IPU_INT_STAT_13[2]	IDMAC	IDMAC_TH_2	-
IPU_INT_STAT_13[3]	IDMAC	IDMAC_TH_3	-
IPU_INT_STAT_13[5]	IDMAC	IDMAC_TH_5	-
IPU_INT_STAT_13[8]	IDMAC	IDMAC_TH_8	-
IPU_INT_STAT_13[9]	IDMAC	IDMAC_TH_9	-
IPU_INT_STAT_13[10]	IDMAC	IDMAC_TH_10	-
IPU_INT_STAT_13[11]	IDMAC	IDMAC_TH_11	-
IPU_INT_STAT_13[12]	IDMAC	IDMAC_TH_12	-
IPU_INT_STAT_13[13]	IDMAC	IDMAC_TH_13	-
IPU_INT_STAT_13[14]	IDMAC	IDMAC_TH_14	-
IPU_INT_STAT_13[15]	IDMAC	IDMAC_TH_15	-
IPU_INT_STAT_13[17]	IDMAC	IDMAC_TH_17	-
IPU_INT_STAT_13[18]	IDMAC	IDMAC_TH_18	-
IPU_INT_STAT_13[20]	IDMAC	IDMAC_TH_20	-
IPU_INT_STAT_13[21]	IDMAC	IDMAC_TH_21	-
IPU_INT_STAT_13[22]	IDMAC	IDMAC_TH_22	-
IPU_INT_STAT_13[23]	IDMAC	IDMAC_TH_23	-
IPU_INT_STAT_13[24]	IDMAC	IDMAC_TH_24	-
IPU_INT_STAT_13[27]	IDMAC	IDMAC_TH_27	-
IPU_INT_STAT_13[28]	IDMAC	IDMAC_TH_28	-
IPU_INT_STAT_13[29]	IDMAC	IDMAC_TH_29	-
IPU_INT_STAT_13[31]	IDMAC	IDMAC_TH_31	-
IPU_INT_STAT_14[1]	IDMAC	IDMAC_TH_33	-
IPU_INT_STAT_14[8]	IDMAC	IDMAC_TH_40	-
IPU_INT_STAT_14[9]	IDMAC	IDMAC_TH_41	-
IPU_INT_STAT_14[10]	IDMAC	IDMAC_TH_42	-
IPU_INT_STAT_14[11]	IDMAC	IDMAC_TH_43	-

Table continues on the next page...

Table 38-34. Functional Interrupts Summary (continued)

Location	Sub-blocks	Interrupt status bit name	Description
IPU_INT_STAT_14[12]	IDMAC	IDMAC_TH_44	-
IPU_INT_STAT_14[13]	IDMAC	IDMAC_TH_45	-
IPU_INT_STAT_14[14]	IDMAC	IDMAC_TH_46	-
IPU_INT_STAT_14[15]	IDMAC	IDMAC_TH_47	-
IPU_INT_STAT_14[16]	IDMAC	IDMAC_TH_48	-
IPU_INT_STAT_14[17]	IDMAC	IDMAC_TH_49	-
IPU_INT_STAT_14[18]	IDMAC	IDMAC_TH_50	-
IPU_INT_STAT_14[19]	IDMAC	IDMAC_TH_51	-
IPU_INT_STAT_14[20]	IDMAC	IDMAC_TH_52	-
IPU_INT_STAT_15[0]	CM	IPU_SNOOPING1_INT	-
IPU_INT_STAT_15[1]	CM	IPU_SNOOPING2_INT	-
IPU_INT_STAT_15[2]	DP	DP_SF_START	-
IPU_INT_STAT_15[3]	DP	DP_SF_END	-
IPU_INT_STAT_15[4]	DP	DP_ASF_START	-
IPU_INT_STAT_15[5]	DP	DP_ASF_END	-
IPU_INT_STAT_15[6]	DP	DP_SF_BRAKE	-
IPU_INT_STAT_15[7]	DP	DP_ASF_BRAKE	-
IPU_INT_STAT_15[8]	DC	DC_FC_0	-
IPU_INT_STAT_15[9]	DC	DC_FC_1	-
IPU_INT_STAT_15[10]	DC	DC_FC_2	-
IPU_INT_STAT_15[11]	DC	DC_FC_3	-
IPU_INT_STAT_15[12]	DC	DC_FC_4	-
IPU_INT_STAT_15[13]	DC	DC_FC_6	-
IPU_INT_STAT_15[14]	DC	DI_VSYNC_PRE_0	-
IPU_INT_STAT_15[15]	DC	DI_VSYNC_PRE_1	-
IPU_INT_STAT_15[16]	DC	DC_DP_START	-
IPU_INT_STAT_15[17]	DC	DC_ASYNC_STOP	-
IPU_INT_STAT_15[18]	DI0	DI0_DISP_CLK_EN_PRE	-
IPU_INT_STAT_15[19]	DI0	DI0_CNT_EN_PRE_1	-
IPU_INT_STAT_15[20]	DI0	DI0_CNT_EN_PRE_2	-
IPU_INT_STAT_15[21]	DI0	DI0_CNT_EN_PRE_3	-
IPU_INT_STAT_15[22]	DI0	DI0_CNT_EN_PRE_4	-
IPU_INT_STAT_15[23]	DI0	DI0_CNT_EN_PRE_5	-
IPU_INT_STAT_15[24]	DI0	DI0_CNT_EN_PRE_6	-
IPU_INT_STAT_15[25]	DI0	DI0_CNT_EN_PRE_7	-
IPU_INT_STAT_15[26]	DI0	DI0_CNT_EN_PRE_8	-
IPU_INT_STAT_15[27]	DI0	DI0_CNT_EN_PRE_9	-
IPU_INT_STAT_15[28]	DI0	DI0_CNT_EN_PRE_10	-
IPU_INT_STAT_15[29]	DI1	DI1_DISP_CLK_EN_PRE	-

Table continues on the next page...

Table 38-34. Functional Interrupts Summary (continued)

Location	Sub-blocks	Interrupt status bit name	Description
IPU_INT_STAT_15[30]	DI1	DI1_CNT_EN_PRE_3	-
IPU_INT_STAT_15[31]	DI1	DI1_CNT_EN_PRE_8	-

The table below describes the error interrupts. The panic column indicates if this signal is part of the logic generating the ipu_panic signal. The ipu_panic signal can be used for indicating about errors that are result of data rate problems. Such problems may be a result of the IPU running in slower clock then required by the use case. This signal can be used in order to indicate the system that the IPU can't handle the desired data rate. In that case the system may need to increase the clock to the IPU or simplify the use case.

Table 38-35. Error Interrupts Summary (continued)

Location	Sub-blocks	Interrupt Status name	panic	Description
IPU_INT_STAT_5[0]	IDMAC	IDMAC_NFB4EOF_ERR_0	YES	-
IPU_INT_STAT_5[1]	IDMAC	IDMAC_NFB4EOF_ERR_1	YES	-
IPU_INT_STAT_5[2]	IDMAC	IDMAC_NFB4EOF_ERR_2	YES	-
IPU_INT_STAT_5[3]	IDMAC	IDMAC_NFB4EOF_ERR_3	YES	-
IPU_INT_STAT_5[5]	IDMAC	IDMAC_NFB4EOF_ERR_5	YES	-
IPU_INT_STAT_5[8]	IDMAC	IDMAC_NFB4EOF_ERR_8	YES	-
IPU_INT_STAT_5[9]	IDMAC	IDMAC_NFB4EOF_ERR_9	YES	-
IPU_INT_STAT_5[10]	IDMAC	IDMAC_NFB4EOF_ERR_10	YES	-
IPU_INT_STAT_5[11]	IDMAC	IDMAC_NFB4EOF_ERR_11	YES	-
IPU_INT_STAT_5[12]	IDMAC	IDMAC_NFB4EOF_ERR_12	YES	-
IPU_INT_STAT_5[13]	IDMAC	IDMAC_NFB4EOF_ERR_13	YES	-
IPU_INT_STAT_5[14]	IDMAC	IDMAC_NFB4EOF_ERR_14	YES	-
IPU_INT_STAT_5[15]	IDMAC	IDMAC_NFB4EOF_ERR_15	YES	-
IPU_INT_STAT_5[17]	IDMAC	IDMAC_NFB4EOF_ERR_17	YES	-
IPU_INT_STAT_5[18]	IDMAC	IDMAC_NFB4EOF_ERR_18	YES	-
IPU_INT_STAT_5[20]	IDMAC	IDMAC_NFB4EOF_ERR_20	YES	-
IPU_INT_STAT_5[21]	IDMAC	IDMAC_NFB4EOF_ERR_21	YES	-
IPU_INT_STAT_5[22]	IDMAC	IDMAC_NFB4EOF_ERR_22	YES	-
IPU_INT_STAT_5[23]	IDMAC	IDMAC_NFB4EOF_ERR_23	YES	-
IPU_INT_STAT_5[24]	IDMAC	IDMAC_NFB4EOF_ERR_24	YES	-
IPU_INT_STAT_5[27]	IDMAC	IDMAC_NFB4EOF_ERR_27	YES	-
IPU_INT_STAT_5[28]	IDMAC	IDMAC_NFB4EOF_ERR_28	YES	-
IPU_INT_STAT_5[29]	IDMAC	IDMAC_NFB4EOF_ERR_29	YES	-
IPU_INT_STAT_5[31]	IDMAC	IDMAC_NFB4EOF_ERR_31	YES	-
IPU_INT_STAT_6[1]	IDMAC	IDMAC_NFB4EOF_ERR_33	YES	-
IPU_INT_STAT_6[8]	IDMAC	IDMAC_NFB4EOF_ERR_40	YES	-

Table continues on the next page...

Table 38-35. Error Interrupts Summary (continued) (continued)

Location	Sub-blocks	Interrupt Status name	panic	Description
IPU_INT_STAT_6[9]	IDMAC	IDMAC_NFB4EOF_ERR_41	YES	-
IPU_INT_STAT_6[10]	IDMAC	IDMAC_NFB4EOF_ERR_42	YES	-
IPU_INT_STAT_6[11]	IDMAC	IDMAC_NFB4EOF_ERR_43	YES	-
IPU_INT_STAT_6[12]	IDMAC	IDMAC_NFB4EOF_ERR_44	YES	-
IPU_INT_STAT_6[13]	IDMAC	IDMAC_NFB4EOF_ERR_45	YES	-
IPU_INT_STAT_6[14]	IDMAC	IDMAC_NFB4EOF_ERR_46	YES	-
IPU_INT_STAT_6[15]	IDMAC	IDMAC_NFB4EOF_ERR_47	YES	-
IPU_INT_STAT_6[16]	IDMAC	IDMAC_NFB4EOF_ERR_48	YES	-
IPU_INT_STAT_6[17]	IDMAC	IDMAC_NFB4EOF_ERR_49	YES	-
IPU_INT_STAT_6[18]	IDMAC	IDMAC_NFB4EOF_ERR_50	YES	-
IPU_INT_STAT_6[19]	IDMAC	IDMAC_NFB4EOF_ERR_51	YES	-
IPU_INT_STAT_6[20]	IDMAC	IDMAC_NFB4EOF_ERR_52	YES	-
IPU_INT_STAT_9[0]	IC	VDI_FIFO1_OVF	YES	-
IPU_INT_STAT_9[26]	IC	IC_BAYER_BUF_OVF	YES	-
IPU_INT_STAT_9[27]	IC	IC_ENC_BUF_OVF	YES	-
IPU_INT_STAT_9[28]	IC	IC_VF_BUF_OVF	YES	-
IPU_INT_STAT_9[30]	CSI0	CSI0_PUPE	YES	-
IPU_INT_STAT_9[31]	CSI0	CSI1_PUPE	YES	-
IPU_INT_STAT_10[0]	SMFC	SMFC0_FRM_LOST	YES	-
IPU_INT_STAT_10[1]	SMFC	SMFC1_FRM_LOST	YES	-
IPU_INT_STAT_10[2]	SMFC	SMFC2_FRM_LOST	YES	-
IPU_INT_STAT_10[3]	SMFC	SMFC3_FRM_LOST	YES	-
IPU_INT_STAT_10[16]	DC	DC_TEARING_ERR_1	YES	-
IPU_INT_STAT_10[17]	DC	DC_TEARING_ERR_2	YES	-
IPU_INT_STAT_10[18]	DC	DC_TEARING_ERR_6	YES	-
IPU_INT_STAT_10[19]	DI0	DI0_SYNC_DISP_ERR	YES	-
IPU_INT_STAT_10[20]	DI1	DI1_SYNC_DISP_ERR	YES	-
IPU_INT_STAT_10[21]	DI0	DI0_TIME_OUT_ERR	YES	-
IPU_INT_STAT_10[22]	DI1	DI1_TIME_OUT_ERR	YES	-
IPU_INT_STAT_10[24]	IC	IC_VF_FRM_LOST_ERR	YES	-
IPU_INT_STAT_10[25]	IC	IC_ENC_FRM_LOST_ERR	YES	-
IPU_INT_STAT_10[26]	IC	IC_BAYER_FRM_LOST_ERR	YES	-
IPU_INT_STAT_10[28]	CM	NON_PRIVILEGED_ACC_ERR	NO	-
IPU_INT_STAT_10[29]	IDMAC	AXIW_ERR	NO	-
IPU_INT_STAT_10[30]	IDMAC	AXIR_ERR	NO	-

38.4.12.4 SDMA event generator

The IPU provides an SDMA event signal that can be used as trigger to the SoC's SDMA. IPU routes an internal event to the SDMA event signal. The internal event causing the assertion of the SDMA signal is enabled by setting the corresponding bit on the SDMA_EVENT_# registers.

The user is allowed to enable multiple events. When one of these events occurs the ipu_sdma_event signal will be asserted. Similar to interrupts, the ipu_sdma_event signal is cleared by writing one to the corresponding bit in the INT_STAT_# registers.

It is not recommended to use the same internal event for a simultaneous generation of an interrupt signal and an SDMA event. This will require special software care when clearing the corresponding bit in the INT_STAT_# registers.

38.4.12.5 General Configuration Registers

The GCR contains a set of control/status/data registers. It provides IPU interface to the AHB slave bus.

The HCLK rate is equal to the HSP_CLK rate. The detail description of the registers is found in the Programmable Registers section.

38.4.12.6 Shadow Registers Module (SRM)

IPU supports frame by frame task switching. This means that a sub-block can handle a frame with one configuration and handle the following frame with different configuration. Changing the configuration is done by updating the sub-block's parameters.

In order to allow automatic flow without a need of the SW to update all the parameters at the frame boundaries. A Register of a sub-block that has the shadowing capabilities has a shadow register file that resides in the Shadow Register Memory.

The sub-blocks supporting this function may use it in one of the following ways:

38.4.12.6.1 Switching between 2 flows

Upon request from the FSU the SRM switches between the registers and the content in the Shadow Register Memory. When a sub-block uses one of the configuration SW, it is allowed to update the parameters in the memory. This is normally used when 2 flows are supported via one module.

- The current flow's configuration is stored in the module's registers
- The alternate flow configuration is stored in the SRM.
- When switching between flows the current flow's configuration is stored in the SRM. The alternate flow's configuration is written to the module's registers. When the alternate flow ends the configurations are swapped again.

This process is fully controlled by the FSU

38.4.12.6.2 Updating parameters between frames

This mode is used when the user needs to update the parameters of the current task being processed by the module. The updates can't affect the frame that is currently being processed. The update is effective only on the next frame. The SRM performs the parameters update only when there's a specific request by the user.

- The current frame's configuration is stored in the sub-block's registers
- The next frame's configuration is stored in the SRM.
- On frame's boundary the SRM reads the new configuration from the SRM and writes it to the sub-block's registers. The old configuration is lost.

38.4.12.6.3 Updating the memory

In order to avoid data coherency problems, the user should set a flag indicating that he is currently updating the memory region of a specific sub-block. When the flag is set the SRM will not attempt to replace the parameters relevant to the specific sub-block currently being updated by the user.

The user should clear this flag when he completes the update, and the parameters are ready to be used.

The flag is the corresponding **SRM_MODE** field for each sub-block. This field controls the SRM logic that handles the sub-blocks registers

- 00 - ARM platform is allowed to access the sub-block's region in the RAM; The automatic swapping mechanism is disabled.
- 01 - The SRM logic is controlled by the FSU. The update will be done of the next frame.
- 10 - The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame
- 11 - Update now. The SRM is controlled by the ARM platform. The Register will be update now

Each sub-block uses the SRM mechanism according to the sub-block's behavior, the table below summarizes the SRM support for each sub-block

Table 38-36. SRM support per Sub-block

Sub-block	Switching between 2 flows support	Updating parameters between frames	Comment
CSI1, CSI0	NO	YES	SRM_MODE can be 00 or 01. The update will happen only once. When set to 01: after the update the state machine is automatically moved to 00 mode. It is recommended to make sure that the first frame has started by polling the corresponding NFACK bit before setting the SRM_MODE
DI1, DI0	NO	YES	DI0, DI1 parameters are needed when clock change is performed (Clock Change procedure). SRM_MODE can be 00 or 01. The update will happen only once. When set to 01: after the update the state machine is automatically moved to 00 mode
DC	YES	NO	SRM_MODE can be 00 or 10.
DP	YES	NO	SRM_MODE can be 00, 10 or 11. When set to 11: after the update the state machine is automatically moved to 10 mode 10 is not supported for SYNC flows

In order to update parameters the user should monitor the SRM_BUSY bit of the corresponding sub-block. When the SRM is not busy the user should set the **SRM_MODE to 00. The user will now update the register file in the memory. When done the user should switch the SRM_MODE field to the desired mode.**

38.4.12.6.4 SRM priority

The SRM updates the registers according to a pre defined priority. The priority is set according to the SRM_PRI bits of each sub-block. The user must set a unique value for each sub-block.

38.4.12.6.5 SRM entries mapping

The table below maps any IPU register to an address in the SRM. The registers marked as NONE do not have an SRM entry

PG column indicates if this register is saved during power gating mode

LPSR column indicates if this register is swapped during low power screen refresh mode (LPSR)

Table 38-37. IPU SRM entries mapping

Register's name	Register's address	SRM entry	PG	LPSR
IPU_CONF	0x0000_0000	NONE	YES	NO
IPU_SISG_CTRL0	0x00000004	NONE	YES	NO
IPU_SISG_CTRL1	0x00000008	NONE	YES	NO
IPU_SISG_SET_1	0x0000000C	NONE	YES	NO
IPU_SISG_SET_2	0x00000010	NONE	YES	NO
IPU_SISG_SET_3	0x00000014	NONE	YES	NO
IPU_SISG_SET_4	0x00000018	NONE	YES	NO
IPU_SISG_SET_5	0x0000001C	NONE	YES	NO
IPU_SISG_SET_6	0x00000020	NONE	YES	NO
IPU_SISG_CLR_1	0x00000024	NONE	YES	NO
IPU_SISG_CLR_2	0x00000028	NONE	YES	NO
IPU_SISG_CLR_3	0x0000002C	NONE	YES	NO
IPU_SISG_CLR_4	0x00000030	NONE	YES	NO
IPU_SISG_CLR_5	0x00000034	NONE	YES	NO
IPU_SISG_CLR_6	0x00000038	NONE	YES	NO
IPU_INT_CTRL_1	0x0000003C	NONE	YES	NO
IPU_INT_CTRL_2	0x00000040	NONE	YES	NO
IPU_INT_CTRL_3	0x00000044	NONE	YES	NO
IPU_INT_CTRL_4	0x00000048	NONE	YES	NO
IPU_INT_CTRL_5	0x0000004C	NONE	YES	NO
IPU_INT_CTRL_6	0x00000050	NONE	YES	NO
IPU_INT_CTRL_7	0x00000054	NONE	YES	NO
IPU_INT_CTRL_8	0x00000058	NONE	YES	NO
IPU_INT_CTRL_9	0x0000005C	NONE	YES	NO
IPU_INT_CTRL_10	0x00000060	NONE	YES	NO
IPU_INT_CTRL_11	0x00000064	NONE	YES	NO
IPU_INT_CTRL_12	0x00000068	NONE	YES	NO
IPU_INT_CTRL_13	0x0000006C	NONE	YES	NO
IPU_INT_CTRL_14	0x00000070	NONE	YES	NO
IPU_INT_CTRL_15	0x00000074	NONE	YES	NO
IPU_SDMA_EVENT_1	0x00000078	NONE	YES	NO
IPU_SDMA_EVENT_2	0x0000007C	NONE	YES	NO
IPU_SDMA_EVENT_3	0x00000080	NONE	YES	NO
IPU_SDMA_EVENT_4	0x00000084	NONE	YES	NO
IPU_SDMA_EVENT_7	0x00000088	NONE	YES	NO
IPU_SDMA_EVENT_8	0x0000008C	NONE	YES	NO
IPU_SDMA_EVENT_11	0x00000090	NONE	YES	NO
IPU_SDMA_EVENT_12	0x00000094	NONE	YES	NO
IPU_SDMA_EVENT_13	0x00000098	NONE	YES	NO

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_SDMA_EVENT_14	0x0000009C	NONE	YES	NO
IPU_SRM_PRI1	0x000000A0	NONE	YES	NO
IPU_SRM_PRI2	0x000000A4	NONE	YES	NO
IPU_FS_PROC_FLOW1	0x000000A8	NONE	YES	NO
IPU_FS_PROC_FLOW2	0x000000AC	NONE	YES	NO
IPU_FS_PROC_FLOW3	0x000000B0	NONE	YES	NO
IPU_FS_DISP_FLOW1	0x000000B4	NONE	YES	NO
IPU_FS_DISP_FLOW2	0x000000B8	NONE	YES	NO
IPU_SKIP	0x000000BC	NONE	YES	NO
IPU_DISP_ALT_CONF	0x000000C0	NONE	YES	NO
IPU_DISP_GEN	0x000000C4	NONE	YES	NO
IPU_DISP_ALT1	0x000000C8	NONE	YES	NO
IPU_DISP_ALT2	0x000000CC	NONE	YES	NO
IPU_DISP_ALT3	0x000000D0	NONE	YES	NO
IPU_DISP_ALT4	0x000000D4	NONE	YES	NO
IPU_SNOOP	0x000000D8	NONE	YES	NO
IPU_MEM_RST	0x000000DC	NONE	YES	NO
IPU_PM	0x000000E0	NONE	YES	NO
IPU_GPR	0x000000E4	NONE	YES	NO
IPU_CH_DB_MODE_SEL_0	0x00000150	NONE	YES	NO
IPU_CH_DB_MODE_SEL_1	0x00000154	NONE	YES	NO
IPU_ALT_CH_DB_MODE_SEL_0	0x00000168	NONE	YES	NO
IPU_ALT_CH_DB_MODE_SEL_1	0x0000016C	NONE	YES	NO
IPU_CH_TRB_MODE_SEL_0	0x00000178	NONE	YES	NO
IPU_CH_TRB_MODE_SEL_1	0x0000017C	NONE	YES	NO
IPU_INT_STAT_1	0x00000200	NONE	NO	NO
IPU_INT_STAT_2	0x00000204	NONE	NO	NO
IPU_INT_STAT_3	0x00000208	NONE	NO	NO
IPU_INT_STAT_4	0x0000020C	NONE	NO	NO
IPU_INT_STAT_5	0x00000210	NONE	NO	NO
IPU_INT_STAT_6	0x00000214	NONE	NO	NO
IPU_INT_STAT_7	0x00000218	NONE	NO	NO
IPU_INT_STAT_8	0x0000021C	NONE	NO	NO
IPU_INT_STAT_9	0x00000220	NONE	NO	NO
IPU_INT_STAT_10	0x00000224	NONE	NO	NO
IPU_INT_STAT_11	0x00000228	NONE	NO	NO
IPU_INT_STAT_12	0x0000022C	NONE	NO	NO
IPU_INT_STAT_13	0x00000230	NONE	NO	NO
IPU_INT_STAT_14	0x00000234	NONE	NO	NO

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_INT_STAT_15	0x00000238	NONE	NO	NO
IPU_CUR_BUF_0	0x0000023C	NONE	NO	NO
IPU_CUR_BUF_1	0x00000240	NONE	NO	NO
IPU_ALT_CUR_BUF_0	0x00000244	NONE	NO	NO
IPU_ALT_CUR_BUF_1	0x00000248	NONE	NO	NO
IPU_SRM_STAT	0x0000024C	NONE	NO	NO
IPU_PROC_TASKS_STAT	0x00000250	NONE	NO	NO
IPU_DISP_TASKS_STAT	0x00000254	NONE	NO	NO
IPU_TRIPLE_CUR_BUF_0	0x00000258	NONE	NO	NO
IPU_TRIPLE_CUR_BUF_1	0x0000025C	NONE	NO	NO
IPU_TRIPLE_CUR_BUF_2	0x00000260	NONE	NO	NO
IPU_TRIPLE_CUR_BUF_3	0x00000264	NONE	NO	NO
IPU_CH_BUF0_RDY0	0x00000268	NONE	NO	NO
IPU_CH_BUF0_RDY1	0x0000026C	NONE	NO	NO
IPU_CH_BUF1_RDY0	0x00000270	NONE	NO	NO
IPU_CH_BUF1_RDY1	0x00000274	NONE	NO	NO
IPU_ALT_CH_BUF0_RDY0	0x00000278	NONE	NO	NO
IPU_ALT_CH_BUF0_RDY1	0x0000027C	NONE	NO	NO
IPU_ALT_CH_BUF1_RDY0	0x00000280	NONE	NO	NO
IPU_ALT_CH_BUF1_RDY1	0x00000284	NONE	NO	NO
IPU_CH_BUF2_RDY0	0x00000288	NONE	NO	NO
IPU_CH_BUF2_RDY1	0x0000028C	NONE	NO	NO
IPU_IDMAC_CONF	0x00008000	NONE	YES	NO
IPU_IDMAC_CH_EN_1	0x00008004	NONE	YES	NO
IPU_IDMAC_CH_EN_2	0x00008008	NONE	YES	NO
IPU_IDMAC_SEP_ALPHA	0x0000800C	NONE	YES	NO
IPU_IDMAC_ALT_SEP_ALPHA	0x00008010	NONE	YES	NO
IPU_IDMAC_CH_PRI_1	0x00008014	NONE	YES	NO
IPU_IDMAC_CH_PRI_2	0x00008018	NONE	YES	NO
IIPU_DMAC_WM_EN_1	0x0000801C	NONE	YES	NO
IPU_IDMAC_WM_EN_2	0x00008020	NONE	YES	NO
IPU_IDMAC_LOCK_EN_1	0x00008024	NONE	YES	NO
IPU_IDMAC_LOCK_EN_2	0x00008028	NONE	YES	NO
IPU_IDMAC_SUB_ADDR_0	0x0000802C	NONE	YES	NO
IPU_IDMAC_SUB_ADDR_1	0x00008030	NONE	YES	NO
IPU_IDMAC_SUB_ADDR_2	0x00008034	NONE	YES	NO
IPU_IDMAC_SUB_ADDR_3	0x00008038	NONE	YES	NO
IPU_IDMAC_SUB_ADDR_4	0x0000803C	NONE	YES	NO
IPU_IDMAC_BNDM_EN_1	0x00008040	NONE	YES	NO

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_IDMAC_BNDM_EN_2	0x00008044	NONE	YES	NO
IPU_IDMAC_SC_CORD	0x00008048	NONE	YES	NO
IPU_IDMAC_SC_CORD1	0x0000804C	NONE	YES	NO
IPU_IDMAC_CH_BUSY_1	0x00008100	NONE	NO	NO
IPU_IDMAC_CH_BUSY_2	0x00008104	NONE	NO	NO
IPU_DP_COM_CONF_SYNC	0x1F40000	0x1F40000	YES	YES
IPU_DP_GRAPH_WIND_CTRL_SYNC	0x1F40004	0x1F40004	YES	YES
IPU_DP_FG_POS_SYNC	0x1F40008	0x1F40008	YES	YES
IPU_DP_CUR_POS_SYNC	0x1F4000C	0x1F4000C	YES	YES
IPU_DP_CUR_MAP_SYNC	0x1F40010	0x1F40010	YES	YES
IPU_DP_GAMMA_C_SYNC_0	0x1F40014	0x1F40014	YES	YES
IPU_DP_GAMMA_C_SYNC_1	0x1F40018	0x1F40018	YES	YES
IPU_DP_GAMMA_C_SYNC_2	0x1F4001C	0x1F4001C	YES	YES
IPU_DP_GAMMA_C_SYNC_3	0x1F40020	0x1F40020	YES	YES
IPU_DP_GAMMA_C_SYNC_4	0x1F40024	0x1F40024	YES	YES
IPU_DP_GAMMA_C_SYNC_5	0x1F40028	0x1F40028	YES	YES
IPU_DP_GAMMA_C_SYNC_6	0x1F4002C	0x1F4002C	YES	YES
IPU_DP_GAMMA_C_SYNC_7	0x1F40030	0x1F40030	YES	YES
IPU_DP_GAMMA_S_SYNC_0	0x1F40034	0x1F40034	YES	YES
IPU_DP_GAMMA_S_SYNC_1	0x1F40038	0x1F40038	YES	YES
IPU_DP_GAMMA_S_SYNC_2	0x1F4003C	0x1F4003C	YES	YES
IPU_DP_GAMMA_S_SYNC_3	0x1F40040	0x1F40040	YES	YES
IPU_DP_CSCA_SYNC_0	0x1F40044	0x1F40044	YES	YES
IPU_DP_CSCA_SYNC_1	0x1F40048	0x1F40048	YES	YES
IPU_DP_CSCA_SYNC_2	0x1F4004C	0x1F4004C	YES	YES
IPU_DP_CSCA_SYNC_3	0x1F40050	0x1F40050	YES	YES
IPU_DP_CSC_SYNC_0	0x1F40054	0x1F40054	YES	YES
IPU_DP_CSC_SYNC_1	0x1F40058	0x1F40058	YES	YES
IPU_DP_CUR_POS_ALT	0x1F4005C	0x1F4005C	YES	YES
IPU_DP_COM_CONF_ASYNC0	0x1F40060	0x1F40060	YES	YES
IPU_DP_GRAPH_WIND_CTRL_ASYNC0	0x1F40064	0x1F40064	YES	YES
IPU_DP_FG_POS_ASYNC0	0x1F40068	0x1F40068	YES	YES
IPU_DP_CUR_POS_ASYNC0	0x1F4006C	0x1F4006C	YES	YES
IPU_DP_CUR_MAP_ASYNC0	0x1F40070	0x1F40070	YES	YES
IPU_DP_GAMMA_C_ASYNC0_0	0x1F40074	0x1F40074	YES	YES
IPU_DP_GAMMA_C_ASYNC0_1	0x1F40078	0x1F40078	YES	YES
IPU_DP_GAMMA_C_ASYNC0_2	0x1F4007C	0x1F4007C	YES	YES
IPU_DP_GAMMA_C_ASYNC0_3	0x1F40080	0x1F40080	YES	YES
IPU_DP_GAMMA_C_ASYNC0_4	0x1F40084	0x1F40084	YES	YES

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DP_GAMMA_C_ASYNC0_5	0x1F40088	0x1F40088	YES	YES
IPU_DP_GAMMA_C_ASYNC0_6	0x1F4008C	0x1F4008C	YES	YES
IPU_DP_GAMMA_C_ASYNC0_7	0x1F40090	0x1F40090	YES	YES
IPU_DP_GAMMA_S_ASYNC0_0	0x1F40094	0x1F40094	YES	YES
IPU_DP_GAMMA_S_ASYNC0_1	0x1F40098	0x1F40098	YES	YES
IPU_DP_GAMMA_S_ASYNC0_2	0x1F4009C	0x1F4009C	YES	YES
IPU_DP_GAMMA_S_ASYNC0_3	0x1F400A0	0x1F400A0	YES	YES
IPU_DP_CSCA_ASYNC0_0	0x1F400A4	0x1F400A4	YES	YES
IPU_DP_CSCA_ASYNC0_1	0x1F400A8	0x1F400A8	YES	YES
IPU_DP_CSCA_ASYNC0_2	0x1F400AC	0x1F400AC	YES	YES
IPU_DP_CSCA_ASYNC0_3	0x1F400B0	0x1F400B0	YES	YES
IPU_DP_CSC_ASYNC0_0	0x1F400B4	0x1F400B4	YES	YES
IPU_DP_CSC_ASYNC0_1	0x1F400B8	0x1F400B8	YES	YES
IPU_DP_COM_CONF_ASYNC1	0x1F400BC	0x1F400BC	YES	YES
IPU_DP_GRAPH_WIND_CTRL_ASYNC1	0x1F400C0	0x1F400C0	YES	YES
IPU_DP_FG_POS_ASYNC1	0x1F400C4	0x1F400C4	YES	YES
IPU_DP_CUR_POS_ASYNC1	0x1F400C8	0x1F400C8	YES	YES
IPU_DP_CUR_MAP_ASYNC1	0x1F400CC	0x1F400CC	YES	YES
IPU_DP_GAMMA_C_ASYNC1_0	0x1F400D0	0x1F400D0	YES	YES
IPU_DP_GAMMA_C_ASYNC1_1	0x1F400D4	0x1F400D4	YES	YES
IPU_DP_GAMMA_C_ASYNC1_2	0x1F400D8	0x1F400D8	YES	YES
IPU_DP_GAMMA_C_ASYNC1_3	0x1F400DC	0x1F400DC	YES	YES
IPU_DP_GAMMA_C_ASYNC1_4	0x1F400E0	0x1F400E0	YES	YES
IPU_DP_GAMMA_C_ASYNC1_5	0x1F400E4	0x1F400E4	YES	YES
IPU_DP_GAMMA_C_ASYNC1_6	0x1F400E8	0x1F400E8	YES	YES
IPU_DP_GAMMA_C_ASYNC1_7	0x1F400EC	0x1F400EC	YES	YES
IPU_DP_GAMMA_S_ASYNC1_0	0x1F400F0	0x1F400F0	YES	YES
IPU_DP_GAMMA_S_ASYNC1_1	0x1F400F4	0x1F400F4	YES	YES
IPU_DP_GAMMA_S_ASYNC1_2	0x1F400F8	0x1F400F8	YES	YES
IPU_DP_GAMMA_S_ASYNC1_3	0x1F400FC	0x1F400FC	YES	YES
IPU_DP_CSCA_ASYNC1_0	0x1F40100	0x1F40100	YES	YES
IPU_DP_CSCA_ASYNC1_1	0x1F40104	0x1F40104	YES	YES
IPU_DP_CSCA_ASYNC1_2	0x1F40108	0x1F40108	YES	YES
IPU_DP_CSCA_ASYNC1_3	0x1F4010C	0x1F4010C	YES	YES
IPU_DP_CSC_ASYNC1_0	0x1F40110	0x1F40110	YES	YES
IPU_DP_CSC_ASYNC1_1	0x1F40114	0x1F40114	YES	YES
IPU_DP_DEBUG_CNT	0x000180BC	NONE	NO	NO
IPU_DP_DEBUG_STAT	0x000180C0	NONE	NO	NO
IPU_IC_CONF	0x00020000	NONE	YES	NO

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_IC_PRP_ENC_RSC	0x00020004	NONE	YES	NO
IPU_IC_PRP_VF_RSC	0x00020008	NONE	YES	NO
IPU_IC_PP_RSC	0x0002000C	NONE	YES	NO
IPU_IC_CMBP_1	0x00020010	NONE	YES	NO
IPU_IC_CMBP_2	0x00020014	NONE	YES	NO
IPU_IC_IDMAC_1	0x00020018	NONE	YES	NO
IPU_IC_IDMAC_2	0x0002001C	NONE	YES	NO
IPU_IC_IDMAC_3	0x00020020	NONE	YES	NO
IPU_IC_IDMAC_4	0x00020024	NONE	YES	NO
IPU_CSI0_SENS_CONF	0x00030000	NONE	YES	NO
IPU_CSI0_SENS_FRM_SIZE	0x00030004	NONE	YES	NO
IPU_CSI0_ACT_FRM_SIZE	0x00030008	NONE	YES	NO
IPU_CSI0_OUT_FRM_CTRL	0x0003000C	NONE	YES	NO
IPU_CSI0_TST_CTRL	0x00030010	NONE	YES	NO
IPU_CSI0_CCIR_CODE_1	0x00030014	NONE	YES	NO
IPU_CSI0_CCIR_CODE_2	0x00030018	NONE	YES	NO
IPU_CSI0_CCIR_CODE_3	0x0003001C	NONE	YES	NO
IPU_CSI0_DI	0x00030020	NONE	YES	NO
IPU_CSI0_SKIP	0x00030024	NONE	YES	NO
IPU_CSI0_CPD_CTRL	0x00030028	0x1F40314	YES	NO
IPU_CSI0_CPD_RC_0	0x0003002C	0x1F40318	YES	NO
IPU_CSI0_CPD_RC_1	0x00030030	0x1F4031C	YES	NO
IPU_CSI0_CPD_RC_2	0x00030034	0x1F40320	YES	NO
IPU_CSI0_CPD_RC_3	0x00030038	0x1F40324	YES	NO
IPU_CSI0_CPD_RC_4	0x0003003C	0x1F40328	YES	NO
IPU_CSI0_CPD_RC_5	0x00030040	0x1F4032C	YES	NO
IPU_CSI0_CPD_RC_6	0x00030044	0x1F40330	YES	NO
IPU_CSI0_CPD_RC_7	0x00030048	0x1F40334	YES	NO
IPU_CSI0_CPD_RS_0	0x0003004C	0x1F40338	YES	NO
IPU_CSI0_CPD_RS_1	0x00030050	0x1F4033C	YES	NO
IPU_CSI0_CPD_RS_2	0x00030054	0x1F40340	YES	NO
IPU_CSI0_CPD_RS_3	0x00030058	0x1F40344	YES	NO
IPU_CSI0_CPD_GRC_0	0x0003005C	0x1F40348	YES	NO
IPU_CSI0_CPD_GRC_1	0x00030060	0x1F4034C	YES	NO
IPU_CSI0_CPD_GRC_2	0x00030064	0x1F40350	YES	NO
IPU_CSI0_CPD_GRC_3	0x00030068	0x1F40354	YES	NO
IPU_CSI0_CPD_GRC_4	0x0003006C	0x1F40358	YES	NO
IPU_CSI0_CPD_GRC_5	0x00030070	0x1F4035C	YES	NO
IPU_CSI0_CPD_GRC_6	0x00030074	0x1F40360	YES	NO

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_CSI0_CPD_GRC_7	0x00030078	0x1F40364	YES	NO
IPU_CSI0_CPD_GRS_0	0x0003007C	0x1F40368	YES	NO
IPU_CSI0_CPD_GRS_1	0x00030080	0x1F4036C	YES	NO
IPU_CSI0_CPD_GRS_2	0x00030084	0x1F40370	YES	NO
IPU_CSI0_CPD_GRS_3	0x00030088	0x1F40374	YES	NO
IPU_CSI0_CPD_GBC_0	0x0003008C	0x1F40378	YES	NO
IPU_CSI0_CPD_GBC_1	0x00030090	0x1F4037C	YES	NO
IPU_CSI0_CPD_GBC_2	0x00030094	0x1F40380	YES	NO
IPU_CSI0_CPD_GBC_3	0x00030098	0x1F40384	YES	NO
IPU_CSI0_CPD_GBC_4	0x0003009C	0x1F40388	YES	NO
IPU_CSI0_CPD_GBC_5	0x000300A0	0x1F4038C	YES	NO
IPU_CSI0_CPD_GBC_6	0x000300A4	0x1F40390	YES	NO
IPU_CSI0_CPD_GBC_7	0x000300A8	0x1F40394	YES	NO
IPU_CSI0_CPD_GBS_0	0x000300AC	0x1F40398	YES	NO
IPU_CSI0_CPD_GBS_1	0x000300B0	0x1F4039C	YES	NO
IPU_CSI0_CPD_GBS_2	0x000300B4	0x1F403A0	YES	NO
IPU_CSI0_CPD_GBS_3	0x000300B8	0x1F403A4	YES	NO
IPU_CSI0_CPD_BC_0	0x000300BC	0x1F403A8	YES	NO
IPU_CSI0_CPD_BC_1	0x000300C0	0x1F403AC	YES	NO
IPU_CSI0_CPD_BC_2	0x000300C4	0x1F403B0	YES	NO
IPU_CSI0_CPD_BC_3	0x000300C8	0x1F403B4	YES	NO
IPU_IPU_CSI0_CPD_BC_4	0x000300CC	0x1F403B8	YES	NO
IPU_CSI0_CPD_BC_5	0x000300D0	0x1F403BC	YES	NO
IPU_CSI0_CPD_BC_6	0x000300D4	0x1F403C0	YES	NO
IPU_CSI0_CPD_BC_7	0x000300D8	0x1F403C4	YES	NO
IPU_CSI0_CPD_BS_0	0x000300DC	0x1F403C8	YES	NO
IPU_CSI0_CPD_BS_1	0x000300E0	0x1F403CC	YES	NO
IPU_CSI0_CPD_BS_2	0x000300E4	0x1F403D0	YES	NO
IPU_CSI0_CPD_BS_3	0x000300E8	0x1F403D4	YES	NO
IPU_CSI0_CPD_OFFSET1	0x000300EC	0x1F403D8	YES	NO
IPU_CSI0_CPD_OFFSET2	0x000300F0	0x1F403DC	YES	NO
IPU_CSI1_SENS_CONF	0x00038000	NONE	YES	NO
IPU_CSI1_SENS_FRM_SIZE	0x00038004	NONE	YES	NO
IPU_CSI1_ACT_FRM_SIZE	0x00038008	NONE	YES	NO
IPU_CSI1_OUT_FRM_CTRL	0x0003800C	NONE	YES	NO
IPU_CSI1_TST_CTRL	0x00038010	NONE	YES	NO
IPU_CSI1_CCIR_CODE_1	0x00038014	NONE	YES	NO
IPU_CSI1_CCIR_CODE_2	0x00038018	NONE	YES	NO
IPU_CSI1_CCIR_CODE_3	0x0003801C	NONE	YES	NO

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_CSI1_DI	0x00038020	NONE	YES	NO
IPU_CSI1_SKIP	0x00038024	NONE	YES	NO
IPU_CSI1_CPD_CTRL	0x00038028	0x1F403E0	YES	NO
IPU_CSI1_CPD_RC_0	0x0003802C	0x1F403E4	YES	NO
IPU_CSI1_CPD_RC_1	0x00038030	0x1F403E8	YES	NO
IPU_CSI1_CPD_RC_2	0x00038034	0x1F403EC	YES	NO
IPU_CSI1_CPD_RC_3	0x00038038	0x1F403F0	YES	NO
IPU_CSI1_CPD_RC_4	0x0003803C	0x1F403F4	YES	NO
IPU_CSI1_CPD_RC_5	0x00038040	0x1F403F8	YES	NO
IPU_CSI1_CPD_RC_6	0x00038044	0x1F403FC	YES	NO
IPU_CSI1_CPD_RC_7	0x00038048	0x1F40400	YES	NO
IPU_CSI1_CPD_RS_0	0x0003804C	0x1F40404	YES	NO
IPU_CSI1_CPD_RS_1	0x00038050	0x1F40408	YES	NO
IPU_CSI1_CPD_RS_2	0x00038054	0x1F4040C	YES	NO
IPU_CSI1_CPD_RS_3	0x00038058	0x1F40410	YES	NO
IPU_CSI1_CPD_GRC_0	0x0003805C	0x1F40414	YES	NO
IPU_CSI1_CPD_GRC_1	0x00038060	0x1F40418	YES	NO
IPU_CSI1_CPD_GRC_2	0x00038064	0x1F4041C	YES	NO
IPU_CSI1_CPD_GRC_3	0x00038068	0x1F40420	YES	NO
IPU_CSI1_CPD_GRC_4	0x0003806C	0x1F40424	YES	NO
IPU_CSI1_CPD_GRC_5	0x00038070	0x1F40428	YES	NO
IPU_CSI1_CPD_GRC_6	0x00038074	0x1F4042C	YES	NO
IPU_CSI1_CPD_GRC_7	0x00038078	0x1F40430	YES	NO
IPU_CSI1_CPD_GRS_0	0x0003807C	0x1F40434	YES	NO
IPU_CSI1_CPD_GRS_1	0x00038080	0x1F40438	YES	NO
IPU_CSI1_CPD_GRS_2	0x00038084	0x1F4043C	YES	NO
IPU_CSI1_CPD_GRS_3	0x00038088	0x1F40440	YES	NO
IPU_CSI1_CPD_GBC_0	0x0003808C	0x1F40444	YES	NO
IPU_CSI1_CPD_GBC_1	0x00038090	0x1F40448	YES	NO
IPU_CSI1_CPD_GBC_2	0x00038094	0x1F4044C	YES	NO
IPU_CSI1_CPD_GBC_3	0x00038098	0x1F40450	YES	NO
IPU_CSI1_CPD_GBC_4	0x0003809C	0x1F40454	YES	NO
IPU_CSI1_CPD_GBC_5	0x000380A0	0x1F40458	YES	NO
IPU_CSI1_CPD_GBC_6	0x000380A4	0x1F4045C	YES	NO
IPU_CSI1_CPD_GBC_7	0x000380A8	0x1F40460	YES	NO
IPU_CSI1_CPD_GBS_0	0x000380AC	0x1F40464	YES	NO
IPU_CSI1_CPD_GBS_1	0x000380B0	0x1F40468	YES	NO
IPU_CSI1_CPD_GBS_2	0x000380B4	0x1F4046C	YES	NO
IPU_CSI1_CPD_GBS_3	0x000380B8	0x1F40470	YES	NO

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_CSI1_CPD_BC_0	0x000380BC	0x1F40474	YES	NO
IPU_CSI1_CPD_BC_1	0x000380C0	0x1F40478	YES	NO
IPU_CSI1_CPD_BC_2	0x000380C4	0x1F4047C	YES	NO
IPU_CSI1_CPD_BC_3	0x000380C8	0x1F40480	YES	NO
IPU_CSI1_CPD_BC_4	0x000380CC	0x1F40484	YES	NO
IPU_CSI1_CPD_BC_5	0x000380D0	0x1F40488	YES	NO
IPU_CSI1_CPD_BC_6	0x000380D4	0x1F4048C	YES	NO
IPU_CSI1_CPD_BC_7	0x000380D8	0x1F40490	YES	NO
IPU_CSI1_CPD_BS_0	0x000380DC	0x1F40494	YES	NO
IPU_CSI1_CPD_BS_1	0x000380E0	0x1F40498	YES	NO
IPU_CSI1_CPD_BS_2	0x000380E4	0x1F4049C	YES	NO
IPU_CSI1_CPD_BS_3	0x000380E8	0x1F404A0	YES	NO
IPU_CSI1_CPD_OFFSET1	0x000380EC	0x1F404A4	YES	NO
IPU_CSI1_CPD_OFFSET2	0x000380F0	0x1F404A8	YES	NO
IPU_DI0_GENERAL	0x00040000	0x1F404E4	YES	YES
IPU_DI0_BS_CLKGEN0	0x00040004	0x1F404E8	YES	YES
IPU_DI0_BS_CLKGEN1	0x00040008	0x1F404EC	YES	YES
IPU_DI0_SW_GEN0_1	0x0004000C	0x1F404F0	YES	YES
IPU_DI0_SW_GEN0_2	0x00040010	0x1F404F4	YES	YES
IPU_DI0_SW_GEN0_3	0x00040014	0x1F404F8	YES	YES
IPU_DI0_SW_GEN0_4	0x00040018	0x1F404FC	YES	YES
IPU_DI0_SW_GEN0_5	0x0004001C	0x1F40500	YES	YES
IPU_DI0_SW_GEN0_6	0x00040020	0x1F40504	YES	YES
IPU_DI0_SW_GEN0_7	0x00040024	0x1F40508	YES	YES
IPU_DI0_SW_GEN0_8	0x00040028	0x1F4050C	YES	YES
IPU_DI0_SW_GEN0_9	0x0004002C	0x1F40510	YES	YES
IPU_DI0_SW_GEN1_1	0x00040030	0x1F40514	YES	YES
IPU_DI0_SW_GEN1_2	0x00040034	0x1F40518	YES	YES
IPU_DI0_SW_GEN1_3	0x00040038	0x1F4051C	YES	YES
IPU_DI0_SW_GEN1_4	0x0004003C	0x1F40520	YES	YES
IPU_DI0_SW_GEN1_5	0x00040040	0x1F40524	YES	YES
IPU_DI0_SW_GEN1_6	0x00040044	0x1F40528	YES	YES
IPU_DI0_SW_GEN1_7	0x00040048	0x1F4052C	YES	YES
IPU_DI0_SW_GEN1_8	0x0004004C	0x1F40530	YES	YES
IPU_DI0_SW_GEN1_9	0x00040050	0x1F40534	YES	YES
IPU_DI0_SYNC_AS_GEN	0x00040054	0x1F40538	YES	YES
IPU_DI0_DW_GEN_0	0x00040058	0x1F4053C	YES	YES
IPU_DI0_DW_GEN_1	0x0004005C	0x1F40540	YES	YES
IPU_DI0_DW_GEN_2	0x00040060	0x1F40544	YES	YES

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DI0_DW_GEN_3	0x00040064	0x1F40548	YES	YES
IPU_DI0_DW_GEN_4	0x00040068	0x1F4054C	YES	YES
IPU_DI0_DW_GEN_5	0x0004006C	0x1F40550	YES	YES
IPU_DI0_DW_GEN_6	0x00040070	0x1F40554	YES	YES
IPU_DI0_DW_GEN_7	0x00040074	0x1F40558	YES	YES
IPU_DI0_DW_GEN_8	0x00040078	0x1F4055C	YES	YES
IPU_DI0_DW_GEN_9	0x0004007C	0x1F40560	YES	YES
IPU_DI0_DW_GEN_10	0x00040080	0x1F40564	YES	YES
IPU_DI0_DW_GEN_11	0x00040084	0x1F40568	YES	YES
IPU_DI0_DW_SET0_0	0x00040088	0x1F4056C	YES	YES
IPU_DI0_DW_SET0_1	0x0004008C	0x1F40570	YES	YES
IPU_DI0_DW_SET0_2	0x00040090	0x1F40574	YES	YES
IPU_DI0_DW_SET0_3	0x00040094	0x1F40578	YES	YES
IPU_DI0_DW_SET0_4	0x00040098	0x1F4057C	YES	YES
IPU_DI0_DW_SET0_5	0x0004009C	0x1F40580	YES	YES
IPU_DI0_DW_SET0_6	0x000400A0	0x1F40584	YES	YES
IPU_DI0_DW_SET0_7	0x000400A4	0x1F40588	YES	YES
IPU_DI0_DW_SET0_8	0x000400A8	0x1F4058C	YES	YES
IPU_DI0_DW_SET0_9	0x000400AC	0x1F40590	YES	YES
IPU_DI0_DW_SET0_10	0x000400B0	0x1F40594	YES	YES
IPU_DI0_DW_SET0_11	0x000400B4	0x1F40598	YES	YES
IPU_DI0_DW_SET1_0	0x000400B8	0x1F4059C	YES	YES
IPU_DI0_DW_SET1_1	0x000400BC	0x1F405A0	YES	YES
IPU_DI0_DW_SET1_2	0x000400C0	0x1F405A4	YES	YES
IPU_DI0_DW_SET1_3	0x000400C4	0x1F405A8	YES	YES
IPU_DI0_DW_SET1_4	0x000400C8	0x1F405AC	YES	YES
IPU_DI0_DW_SET1_5	0x000400CC	0x1F405B0	YES	YES
IPU_DI0_DW_SET1_6	0x000400D0	0x1F405B4	YES	YES
IPU_DI0_DW_SET1_7	0x000400D4	0x1F405B8	YES	YES
IPU_DI0_DW_SET1_8	0x000400D8	0x1F405BC	YES	YES
IPU_DI0_DW_SET1_9	0x000400DC	0x1F405C0	YES	YES
IPU_DI0_DW_SET1_10	0x000400E0	0x1F405C4	YES	YES
IPU_DI0_DW_SET1_11	0x000400E4	0x1F405C8	YES	YES
IPU_DI0_DW_SET2_0	0x000400E8	0x1F405CC	YES	YES
IPU_DI0_DW_SET2_1	0x000400EC	0x1F405D0	YES	YES
IPU_DI0_DW_SET2_2	0x000400F0	0x1F405D4	YES	YES
IPU_DI0_DW_SET2_3	0x000400F4	0x1F405D8	YES	YES
IPU_DI0_DW_SET2_4	0x000400F8	0x1F405DC	YES	YES
IPU_DI0_DW_SET2_5	0x000400FC	0x1F405E0	YES	YES

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DI0_DW_SET2_6	0x00040100	0x1F405E4	YES	YES
IPU_DI0_DW_SET2_7	0x00040104	0x1F405E8	YES	YES
IPU_DI0_DW_SET2_8	0x00040108	0x1F405EC	YES	YES
IPU_DI0_DW_SET2_9	0x0004010C	0x1F405F0	YES	YES
IPU_DI0_DW_SET2_10	0x00040110	0x1F405F4	YES	YES
IPU_DI0_DW_SET2_11	0x00040114	0x1F405F8	YES	YES
IPU_DI0_DW_SET3_0	0x00040118	0x1F405FC	YES	YES
IPU_DI0_DW_SET3_1	0x0004011C	0x1F40600	YES	YES
IPU_DI0_DW_SET3_2	0x00040120	0x1F40604	YES	YES
IPU_DI0_DW_SET3_3	0x00040124	0x1F40608	YES	YES
IPU_DI0_DW_SET3_4	0x00040128	0x1F4060C	YES	YES
IPU_DI0_DW_SET3_5	0x0004012C	0x1F40610	YES	YES
IPU_DI0_DW_SET3_6	0x00040130	0x1F40614	YES	YES
IPU_DI0_DW_SET3_7	0x00040134	0x1F40618	YES	YES
IPU_DI0_DW_SET3_8	0x00040138	0x1F4061C	YES	YES
IPU_DI0_DW_SET3_9	0x0004013C	0x1F40620	YES	YES
IPU_DI0_DW_SET3_10	0x00040140	0x1F40624	YES	YES
IPU_DI0_DW_SET3_11	0x00040144	0x1F40628	YES	YES
IPU_DI0_STP_REP_1	0x00040148	0x1F4062C	YES	YES
IPU_DI0_STP_REP_2	0x0004014C	0x1F40630	YES	YES
IPU_DI0_STP_REP_3	0x00040150	0x1F40634	YES	YES
IPU_DI0_STP_REP_4	0x00040154	0x1F40638	YES	YES
IPU_DI0_STP_REP_9	0x00040158	0x1F4063C	YES	YES
IPU_DI0_SER_CONF	0x0004015C	0x1F40640	YES	YES
IPU_DI0_SSC	0x00040160	0x1F40644	YES	YES
IPU_DI0_POL	0x00040164	0x1F40648	YES	YES
IPU_DI0_AW0	0x00040168	0x1F4064C	YES	YES
IPU_DI0_AW1	0x0004016C	0x1F40650	YES	YES
IPU_DI0_SCR_CONF	0x00040170	0x1F40654	YES	YES
IPU_DI0_STAT	0x00040174	NONE	NO	NO
IPU_DI1_GENERAL	0x00048000	0x1F40658	YES	YES
IPU_DI1_BS_CLKGEN0	0x00048004	0x1F4065C	YES	YES
IPU_DI1_BS_CLKGEN1	0x00048008	0x1F40660	YES	YES
IPU_DI1_SW_GEN0_1	0x0004800C	0x1F40664	YES	YES
IPU_DI1_SW_GEN0_2	0x00048010	0x1F40668	YES	YES
IPU_DI1_SW_GEN0_3	0x00048014	0x1F4066C	YES	YES
IPU_DI1_SW_GEN0_4	0x00048018	0x1F40670	YES	YES
IPU_DI1_SW_GEN0_5	0x0004801C	0x1F40674	YES	YES
IPU_DI1_SW_GEN0_6	0x00048020	0x1F40678	YES	YES

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DI1_SW_GEN0_7	0x00048024	0x1F4067C	YES	YES
IPU_DI1_SW_GEN0_8	0x00048028	0x1F40680	YES	YES
IPU_DI1_SW_GEN0_9	0x0004802C	0x1F40684	YES	YES
IPU_DI1_SW_GEN1_1	0x00048030	0x1F40688	YES	YES
IPU_DI1_SW_GEN1_2	0x00048034	0x1F4068C	YES	YES
IPU_DI1_SW_GEN1_3	0x00048038	0x1F40690	YES	YES
IPU_DI1_SW_GEN1_4	0x0004803C	0x1F40694	YES	YES
IPU_DI1_SW_GEN1_5	0x00048040	0x1F40698	YES	YES
IPU_DI1_SW_GEN1_6	0x00048044	0x1F4069C	YES	YES
IPU_DI1_SW_GEN1_7	0x00048048	0x1F406A0	YES	YES
IPU_DI1_SW_GEN1_8	0x0004804C	0x1F406A4	YES	YES
IPU_DI1_SW_GEN1_9	0x00048050	0x1F406A8	YES	YES
IPU_DI1_SYNC_AS_GEN	0x00048054	0x1F406AC	YES	YES
IPU_DI1_DW_GEN_0	0x00048058	0x1F406B0	YES	YES
IPU_DI1_DW_GEN_1	0x0004805C	0x1F406B4	YES	YES
IPU_DI1_DW_GEN_2	0x00048060	0x1F406B8	YES	YES
IPU_DI1_DW_GEN_3	0x00048064	0x1F406BC	YES	YES
IPU_DI1_DW_GEN_4	0x00048068	0x1F406C0	YES	YES
IPU_DI1_DW_GEN_5	0x0004806C	0x1F406C4	YES	YES
IPU_DI1_DW_GEN_6	0x00048070	0x1F406C8	YES	YES
IPU_DI1_DW_GEN_7	0x00048074	0x1F406CC	YES	YES
IPU_DI1_DW_GEN_8	0x00048078	0x1F406D0	YES	YES
IPU_DI1_DW_GEN_9	0x0004807C	0x1F406D4	YES	YES
IPU_DI1_DW_GEN_10	0x00048080	0x1F406D8	YES	YES
IPU_DI1_DW_GEN_11	0x00048084	0x1F406DC	YES	YES
IPU_DI1_DW_SET0_0	0x00048088	0x1F406E0	YES	YES
IPU_DI1_DW_SET0_1	0x0004808C	0x1F406E4	YES	YES
IPU_DI1_DW_SET0_2	0x00048090	0x1F406E8	YES	YES
IPU_DI1_DW_SET0_3	0x00048094	0x1F406EC	YES	YES
IPU_DI1_DW_SET0_4	0x00048098	0x1F406F0	YES	YES
IPU_DI1_DW_SET0_5	0x0004809C	0x1F406F4	YES	YES
IPU_DI1_DW_SET0_6	0x000480A0	0x1F406F8	YES	YES
IPU_DI1_DW_SET0_7	0x000480A4	0x1F406FC	YES	YES
IPU_DI1_DW_SET0_8	0x000480A8	0x1F40700	YES	YES
IPU_DI1_DW_SET0_9	0x000480AC	0x1F40704	YES	YES
IPU_DI1_DW_SET0_10	0x000480B0	0x1F40708	YES	YES
IPU_DI1_DW_SET0_11	0x000480B4	0x1F4070C	YES	YES
IPU_DI1_DW_SET1_0	0x000480B8	0x1F40710	YES	YES
IPU_DI1_DW_SET1_1	0x000480BC	0x1F40714	YES	YES

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DI1_DW_SET1_2	0x000480C0	0x1F40718	YES	YES
IPU_DI1_DW_SET1_3	0x000480C4	0x1F4071C	YES	YES
IPU_DI1_DW_SET1_4	0x000480C8	0x1F40720	YES	YES
IPU_DI1_DW_SET1_5	0x000480CC	0x1F40724	YES	YES
IPU_DI1_DW_SET1_6	0x000480D0	0x1F40728	YES	YES
IPU_DI1_DW_SET1_7	0x000480D4	0x1F4072C	YES	YES
IPU_DI1_DW_SET1_8	0x000480D8	0x1F40730	YES	YES
IPU_DI1_DW_SET1_9	0x000480DC	0x1F40734	YES	YES
IPU_DI1_DW_SET1_10	0x000480E0	0x1F40738	YES	YES
IPU_DI1_DW_SET1_11	0x000480E4	0x1F4073C	YES	YES
IPU_DI1_DW_SET2_0	0x000480E8	0x1F40740	YES	YES
IPU_DI1_DW_SET2_1	0x000480EC	0x1F40744	YES	YES
IPU_DI1_DW_SET2_2	0x000480F0	0x1F40748	YES	YES
IPU_DI1_DW_SET2_3	0x000480F4	0x1F4074C	YES	YES
IPU_DI1_DW_SET2_4	0x000480F8	0x1F40750	YES	YES
IPU_DI1_DW_SET2_5	0x000480FC	0x1F40754	YES	YES
IPU_DI1_DW_SET2_6	0x00048100	0x1F40758	YES	YES
IPU_DI1_DW_SET2_7	0x00048104	0x1F4075C	YES	YES
IPU_DI1_DW_SET2_8	0x00048108	0x1F40760	YES	YES
IPU_DI1_DW_SET2_9	0x0004810C	0x1F40764	YES	YES
IPU_DI1_DW_SET2_10	0x00048110	0x1F40768	YES	YES
IPU_DI1_DW_SET2_11	0x00048114	0x1F4076C	YES	YES
IPU_DI1_DW_SET3_0	0x00048118	0x1F40770	YES	YES
IPU_DI1_DW_SET3_1	0x0004811C	0x1F40774	YES	YES
IPU_DI1_DW_SET3_2	0x00048120	0x1F40778	YES	YES
IPU_DI1_DW_SET3_3	0x00048124	0x1F4077C	YES	YES
IPU_DI1_DW_SET3_4	0x00048128	0x1F40780	YES	YES
IPU_DI1_DW_SET3_5	0x0004812C	0x1F40784	YES	YES
IPU_DI1_DW_SET3_6	0x00048130	0x1F40788	YES	YES
IPU_DI1_DW_SET3_7	0x00048134	0x1F4078C	YES	YES
IPU_DI1_DW_SET3_8	0x00048138	0x1F40790	YES	YES
IPU_DI1_DW_SET3_9	0x0004813C	0x1F40794	YES	YES
IPU_DI1_DW_SET3_10	0x00048140	0x1F40798	YES	YES
IPU_DI1_DW_SET3_11	0x00048144	0x1F4079C	YES	YES
IPU_DI1_STP_REP_1	0x00048148	0x1F407A0	YES	YES
IPU_DI1_STP_REP_2	0x0004814C	0x1F407A4	YES	YES
IPU_DI1_STP_REP_3	0x00048150	0x1F407A8	YES	YES
IPU_DI1_STP_REP_4	0x00048154	0x1F407AC	YES	YES
IPU_DI1_STP_REP_9	0x00048158	0x1F407B0	YES	YES

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DI1_SER_CONF	0x0004815C	0x1F407B4	YES	YES
IPU_DI1_SSC	0x00048160	0x1F407B8	YES	YES
IPU_DI1_POL	0x00048164	0x1F407BC	YES	YES
IPU_DI1_AW0	0x00048168	0x1F407C0	YES	YES
IPU_DI1_AW1	0x0004816C	0x1F407C4	YES	YES
IPU_DI1_SCR_CONF	0x00048170	0x1F407C8	YES	YES
IPU_DI1_STAT	0x00048174	NONE	NO	NO
IPU_SMFC_MAP	0x00050000	NONE	YES	NO
IPU_SMFC_WMC	0x00050004	NONE	YES	NO
IPU_SMFC_BS	0x00050008	NONE	YES	NO
IPU_DC_READ_CH_CONF	0x00058000	NONE	YES	NO
IPU_DC_READ_CH_ADDR	0x00058004	NONE	YES	NO
IPU_DC_RL0_CH_0	0x00058008	NONE	YES	NO
IPU_DC_RL1_CH_0	0x0005800C	NONE	YES	NO
IPU_DC_RL2_CH_0	0x00058010	NONE	YES	NO
IPU_DC_RL3_CH_0	0x00058014	NONE	YES	NO
IPU_DC_RL4_CH_0	0x00058018	NONE	YES	NO
IPU_DC_WR_CH_CONF_1	0x0005801C	NONE	YES	NO
IPU_DC_WR_CH_ADDR_1	0x00058020	NONE	YES	NO
IPU_DC_RL0_CH_1	0x00058024	NONE	YES	NO
IPU_DC_RL1_CH_1	0x00058028	NONE	YES	NO
IPU_DC_RL2_CH_1	0x0005802C	NONE	YES	NO
IPU_DC_RL3_CH_1	0x00058030	NONE	YES	NO
IPU_DC_RL4_CH_1	0x00058034	NONE	YES	NO
IPU_DC_WR_CH_CONF_2	0x00058038	0x1F404AC	YES	NO
IPU_DC_WR_CH_ADDR_2	0x0005803C	0x1F404B0	YES	NO
IPU_DC_RL0_CH_2	0x00058040	0x1F404B4	YES	NO
IPU_DC_RL1_CH_2	0x00058044	0x1F404B8	YES	NO
IPU_DC_RL2_CH_2	0x00058048	0x1F404BC	YES	NO
IPU_DC_RL3_CH_2	0x0005804C	0x1F404C0	YES	NO
IPU_DC_RL4_CH_2	0x00058050	0x1F404C4	YES	NO
IPU_DC_CMD_CH_CONF_3	0x00058054	NONE	YES	NO
IPU_DC_CMD_CH_CONF_4	0x00058058	NONE	YES	NO
IPU_DC_WR_CH_CONF_5	0x0005805C	NONE	YES	NO
IPU_DC_WR_CH_ADDR_5	0x00058060	NONE	YES	NO
IPU_DC_RL0_CH_5	0x00058064	NONE	YES	NO
IPU_DC_RL1_CH_5	0x00058068	NONE	YES	NO
IPU_DC_RL2_CH_5	0x0005806C	NONE	YES	NO
IPU_DC_RL3_CH_5	0x00058070	NONE	YES	NO

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DC_RL4_CH_5	0x00058074	NONE	YES	NO
IPU_DC_WR_CH_CONF_6	0x00058078	0x1F404C8	YES	NO
IPU_DC_WR_CH_ADDR_6	0x0005807C	0x1F404CC	YES	NO
IPU_DC_RL0_CH_6	0x00058080	0x1F404D0	YES	NO
IPU_DC_RL1_CH_6	0x00058084	0x1F404D4	YES	NO
IPU_DC_RL2_CH_6	0x00058088	0x1F404D8	YES	NO
IPU_DC_RL3_CH_6	0x0005808C	0x1F404DC	YES	NO
IPU_DC_RL4_CH_6	0x00058090	0x1F404E0	YES	NO
IPU_DC_WR_CH_CONF1_8	0x00058094	NONE	YES	NO
IPU_DC_WR_CH_CONF2_8	0x00058098	NONE	YES	NO
IPU_DC_RL1_CH_8	0x0005809C	NONE	YES	NO
IPU_DC_RL2_CH_8	0x000580A0	NONE	YES	NO
IPU_DC_RL3_CH_8	0x000580A4	NONE	YES	NO
IPU_DC_RL4_CH_8	0x000580A8	NONE	YES	NO
IPU_DC_RL5_CH_8	0x000580AC	NONE	YES	NO
IPU_DC_RL6_CH_8	0x000580B0	NONE	YES	NO
IPU_DC_WR_CH_CONF1_9	0x000580B4	NONE	YES	NO
IPU_DC_WR_CH_CONF2_9	0x000580B8	NONE	YES	NO
IPU_DC_RL1_CH_9	0x000580BC	NONE	YES	NO
IPU_DC_RL2_CH_9	0x000580C0	NONE	YES	NO
IPU_DC_RL3_CH_9	0x000580C4	NONE	YES	NO
IPU_DC_RL4_CH_9	0x000580C8	NONE	YES	NO
IPU_DC_RL5_CH_9	0x000580CC	NONE	YES	NO
IPU_DC_RL6_CH_9	0x000580D0	NONE	YES	NO
IPU_DC_GEN	0x000580D4	NONE	YES	NO
IPU_DC_DISP_CONF1_0	0x000580D8	NONE	YES	NO
IPU_DC_DISP_CONF1_1	0x000580DC	NONE	YES	NO
IPU_DC_DISP_CONF1_2	0x000580E0	NONE	YES	NO
IPU_DC_DISP_CONF1_3	0x000580E4	NONE	YES	NO
IPU_DC_DISP_CONF2_0	0x000580E8	NONE	YES	NO
IPU_DC_DISP_CONF2_1	0x000580EC	NONE	YES	NO
IPU_DC_DISP_CONF2_2	0x000580F0	NONE	YES	NO
IPU_DC_DISP_CONF2_3	0x000580F4	NONE	YES	NO
IPU_DC_DI0_CONF_1	0x000580F8	NONE	YES	NO
IPU_DC_DI0_CONF_2	0x000580FC	NONE	YES	NO
IPU_DC_DI1_CONF_1	0x00058100	NONE	YES	NO
IPU_DC_DI1_CONF_2	0x00058104	NONE	YES	NO
IPU_DC_MAP_CONF_0	0x00058108	NONE	YES	NO
IPU_DC_MAP_CONF_1	0x0005810C	NONE	YES	NO

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DC_MAP_CONF_2	0x00058110	NONE	YES	NO
IPU_DC_MAP_CONF_3	0x00058114	NONE	YES	NO
IPU_DC_MAP_CONF_4	0x00058118	NONE	YES	NO
IPU_DC_MAP_CONF_5	0x0005811C	NONE	YES	NO
IPU_DC_MAP_CONF_6	0x00058120	NONE	YES	NO
IPU_DC_MAP_CONF_7	0x00058124	NONE	YES	NO
IPU_DC_MAP_CONF_8	0x00058128	NONE	YES	NO
IPU_DC_MAP_CONF_9	0x0005812C	NONE	YES	NO
IPU_DC_MAP_CONF_10	0x00058130	NONE	YES	NO
IPU_DC_MAP_CONF_11	0x00058134	NONE	YES	NO
IPU_DC_MAP_CONF_12	0x00058138	NONE	YES	NO
IPU_DC_MAP_CONF_13	0x0005813C	NONE	YES	NO
IPU_DC_MAP_CONF_14	0x00058140	NONE	YES	NO
IPU_DC_MAP_CONF_15	0x00058144	NONE	YES	NO
IPU_DC_MAP_CONF_16	0x00058148	NONE	YES	NO
IPU_DC_MAP_CONF_17	0x0005814C	NONE	YES	NO
IPU_DC_MAP_CONF_18	0x00058150	NONE	YES	NO
IPU_DC_MAP_CONF_19	0x00058154	NONE	YES	NO
IPU_DC_MAP_CONF_20	0x00058158	NONE	YES	NO
IPU_DC_MAP_CONF_21	0x0005815C	NONE	YES	NO
IPU_DC_MAP_CONF_22	0x00058160	NONE	YES	NO
IPU_DC_MAP_CONF_23	0x00058164	NONE	YES	NO
IPU_DC_MAP_CONF_24	0x00058168	NONE	YES	NO
IPU_DC_MAP_CONF_25	0x0005816C	NONE	YES	NO
IPU_DC_MAP_CONF_26	0x00058170	NONE	YES	NO
IPU_DC_UGDE0_0	0x00058174	NONE	YES	NO
IPU_DC_UGDE0_1	0x00058178	NONE	YES	NO
IPU_DC_UGDE0_2	0x0005817C	NONE	YES	NO
IPU_DC_UGDE0_3	0x00058180	NONE	YES	NO
IPU_DC_UGDE1_0	0x00058184	NONE	YES	NO
IPU_DC_UGDE1_1	0x00058188	NONE	YES	NO
IPU_DC_UGDE1_2	0x0005818C	NONE	YES	NO
IPU_DC_UGDE1_3	0x00058190	NONE	YES	NO
IPU_DC_UGDE2_0	0x00058194	NONE	YES	NO
IPU_DC_UGDE2_1	0x00058198	NONE	YES	NO
IPU_DC_UGDE2_2	0x0005819C	NONE	YES	NO
IPU_DC_UGDE2_3	0x000581A0	NONE	YES	NO
IPU_DC_UGDE3_0	0x000581A4	NONE	YES	NO
IPU_DC_UGDE3_1	0x000581A8	NONE	YES	NO

Table continues on the next page...

Table 38-37. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DC_UGDE3_2	0x000581AC	NONE	YES	NO
IPU_DC_UGDE3_3	0x000581B0	NONE	YES	NO
IPU_DC_LLA0	0x000581B4	NONE	YES	NO
IPU_DC_LLA1	0x000581B8	NONE	YES	NO
IPU_DC_R_LLA0	0x000581BC	NONE	YES	NO
IPU_DC_R_LLA1	0x000581C0	NONE	YES	NO
IPU_DC_WR_CH_ADDR_5_ALT	0x000581C4	NONE	YES	NO
IPU_DC_STAT	0x000581C8	NONE	NO	NO
IPU_DMFC_RD_CHAN	0x00060000	NONE	YES	NO
IPU_DMFC_WR_CHAN	0x00060004	NONE	YES	NO
IPU_DMFC_WR_CHAN_DEF	0x00060008	NONE	YES	NO
IPU_DMFC_DP_CHAN	0x0006000C	NONE	YES	NO
IPU_DMFC_DP_CHAN_DEF	0x00060010	NONE	YES	NO
IPU_DMFC_GENERAL1	0x00060014	NONE	YES	NO
IPU_DMFC_GENERAL2	0x00060018	NONE	YES	NO
IPU_DMFC_IC_CTRL	0x0006001C	NONE	YES	NO
IPU_DMFC_WR_CHAN_ALT	0x00060020	NONE	YES	NO
IPU_DMFC_WR_CHAN_DEF_ALT	0x00060024	NONE	YES	NO
IPU_DMFC_DP_CHAN_ALT	0x00060028	NONE	YES	NO
IPU_DMFC_DP_CHAN_DEF_ALT	0x0006002C	NONE	YES	NO
IPU_DMFC_GENERAL1_ALT	0x00060030	NONE	YES	NO
IPU_DMFC_STAT	0x00060034	NONE	NO	NO
IPU_VDI_FSIZE	0x00068000	NONE	YES	NO
IPU_VDI_C	0x00068004	NONE	YES	NO
IPU_VDI_C2	0x00068008	NONE	YES	NO
IPU_VDI_CMBP_1	0x0006800C	NONE	YES	NO
IPU_VDI_CMBP_2	0x00068010	NONE	YES	NO
IPU_VDI_PS_1	0x00068014	NONE	YES	NO
IPU_VDI_PS_2	0x00068018	NONE	YES	NO
IPU_VDI_PS_3	0x0006801C	NONE	YES	NO
IPU_VDI_PS_4	0x00068020	NONE	YES	NO

38.4.12.7 Memory Access Unit

The Memory Access Unit (MA) supports ARM platform access to the IPU internal memories.

Some of the IPU internal memories are memory mapped. This unit handles accessing these memories. The table below describe the accessible memories and their limitations.

Table 38-38. Internal Memories Access Support and Limitations

Memory	Function	Support and Limitations
lut	IDMAC's look up table	Accessible only when All the IDMAC channels that use the LUT are disabled
cpmem	IDMAC's Channel parameter Memory	This memory can be accessed while tasks are enabled. Must be configured before enabling processing tasks. IDMAC channel parameters must not be changed in the CPM when the corresponding DMA channel is enabled excluding the base addresses (EBA0 and EBA1). One of these parameters can be changed during channel operation if it relates to the non-active double buffer.
tpm	IC's task parameter memory	Must be configured before enabling processing tasks. This memory can be accessed while tasks are enabled. IC task parameters must not be changed in the TPM when the corresponding task is active.
dc_template	DC's template memory	Can be configured before enabling tasks. Must not be accessed while operation. Both read and write access to the DC's template memory are forbidden when there is any enabled channel which can use the template

38.4.12.8 SISG - Still Image Synchronization Generator

The IPU incudes a "Still Image Synchronization Generator" (SISG), providing time-sensitive control signals synchronizing the image sensor with camera peripherals, such as a flash lamp and a mechanical shutter.

The SISG is implemented using a single time base counter, and six Time Compare Units - as described in the following figure.

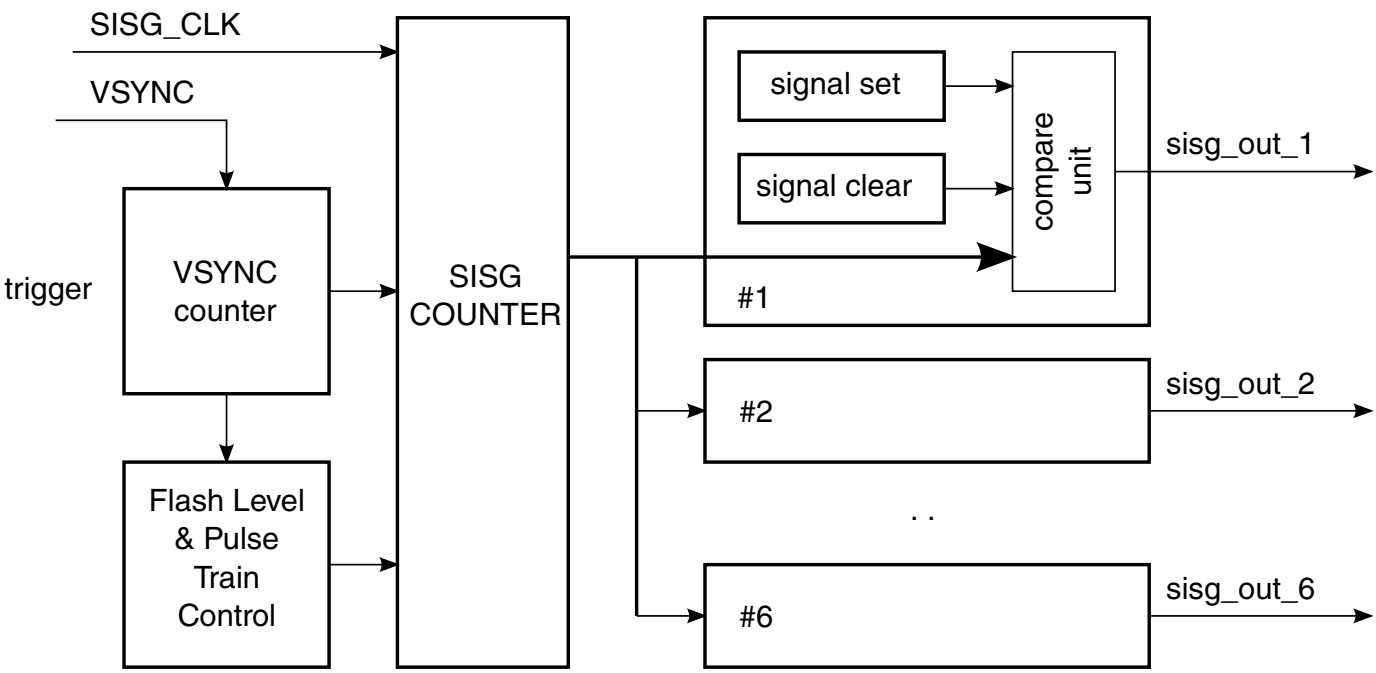


Figure 38-54. Still Image Signal Generator

The SISG inputs are:

- Activation trigger
- VSYNC: the frame-boundary signal from the sensor

The SISG is activated by one of the following triggers:

- The ARM platform by setting the MCU_ACTV_TRIG bit
- A signal generated by an appropriate short packet received through the MIPI/CSI-2 I/F
- An external signal (GPIO pin)

Upon activation, the counter is reset and then there are the following two possibilities:

- Counting starts immediately
- Counting start is delayed until one of the next 7 VSYNC signals (programmable via the NO_OF_VSYNC bits)

During the counting period, the SISG can generate up to 6 output strobes:

- Each strobe can be individually enabled or disabled and has a programmable polarity

- The edges of the strobes are generated at specified counter values - to achieve pixel-level resolution - as specified by programmable SISG_SET& SISG_CLR time tag registers
- The clock has 25 bits, to allow strobe generation during a time period of up to two 12M pixel frames

The SISG can repeat the above sequence for up to 32 cycles (this is provided to generate a train of flash pulses, for anti-red-eye or for measurements in low-light conditions). The repetition is implemented by resetting the counter, which can be triggered by one of the following events:

- A VSYNC signal
- A pre-defined value reached by the counter

After the last sequence, when the counter reaches its maximal value, it stops counting and the SISG remains in idle mode until the next activation.

38.4.12.9 Clock Change procedure

The IPU supports dynamic clock rate changes.

Types of change:

- DVFS transitions: frequent, initiated by the SoC's power modes controller (GPC) and the SoC's clock controller module (CCM)
- Other: infrequent, initiated by SW.

IPU may have on-going activities at this stage.

The display interface clocks may either change or remain unchanged. During screen refresh, the display clock would typically not change. During asynchronous access, it may be appropriate to change also the display interface clock. The choice between these options would be made in advance by the user

If the IPU display interface uses the external clock (DI0_DISP_CLK - ipp_di_0_ext_clk or DI1_DISP_CLK - ipp_di_1_ext_clk) source, it remains unchanged. A change in the rate of this clock is performed fully by SW, without the special HW support described below. In particular, the SW may have to stop explicitly any interaction with the display (e.g. screen refresh) before performing the change.

The user is responsible to make sure that the lowest planned clock (in DVFS transitions) is still high enough to support the expected activities (e.g. data rate through the display bus)

The procedure below describes the IPU handshaking with the CCM.

1. The user prepares 2 sets of clock modes CLOCK_MODE_0 is the default clock mode, CLOCK_MODE_1 is the alternate clock mode. The IPU toggles between these two settings following the next assertion of ipg_clk_change_rq. If the user sets the SRM_CLOCK_CHANGE_MODE bit then he should also prepare the registers in the SRM for each of the DIs. These registers include all the DI settings adjusted to the new clock.
2. CCM asserts the ipg_clk_change_rq signal when a clock change is needed
3. The CM calculates the new clock frequency and send it to the DI (signals are di0_clk_freq, and di1_clk_freq). These signals should be sent to the DI only after getting the di_clk_change_ack signal from the DIs. The values of this field could be.
   - 00 - 1/4 of full frequency
   - 01 - 1/2 of full frequency
   - 10 - full frequency
   - 11 - illegal
4. The CM sends a cm_clk_change_rq signal to the DIs.
5. The DI stops the clock to the display (freeze mode) according to DI0_CLOCK_STOP_MODE & DI1_CLOCK_STOP_MODE bits. If the DI is disable, the ACK from the DI is not needed and the CM will assume that the DI sent an ACK.
6. Once the clock to the display is stopped, the DI sends a signal to the CM called di_clk_change_ack
7. The CM wait for the clock change signals from both of the DIs
8. If the SRM_CLOCK_CHANGE_MODE bit is set the CM should read the new DI settings from the SRM and override the previous DI settings. Then the CM clears the SRM_CLOCK_CHANGE_MODE bit
9. Once the above is complete the CM asserts the ipg_clk_change_ack signal to the CCM
10. The CM sends the signals di0_clk_freq, and di1_clk_freq to the DIs.
11. The CCM will negate the cm_clk_change_rq
12. The CCM will now change the clocks
13. When the new clock arrives the ipu_clk_changed signal will be asserted.
14. The state machine on each DI will now move out of freeze mode and continue working with the new clock.

38.4.12.10 Low Power Modes

IPU supports the following low power modes.

- STOP: on this mode the clock to the IPU is stopped
- LPSR: low power screen refresh.The clock to the IPU is changed to a slower frequency, the IPU performs only screen refresh.

The user should not request LPSR. at the same time. This case is not supported and the results are not predictable.

The CCM may assert one of the 2 signals: stop_clk_at_stop_req OR stop_clk_at_wait_req

The IPU OR them internally as there's no difference between them with regards to IPU's behavior.

In all these modes the clock to the IPU is going to be stopped (assertion of stop_clk_at_stop_req).

IPU should complete all his tasks:

- CSIs complete transferring the last frame. Wait for csi_busy = 0
- IDMAC completes all the flows (all CH_BUSY = 0)
- All the flows in the FSU are complete. New ones do not start.
- CM sends stop request to the DIs
- Once the DI sent all the data to the display, it asserts an ACK signal

Only when all the above occurs, the CM can assert an internal signal called "IPU_IDLE"

IPU_IDLE is the starting point for any of the low power modes.

Note that if the VDIC was in use prior to entrance the low power mode the viewfinder task of the IC will be in WAIT_FOR_READY state (the task's status is reflected in the VF_TSTAT field). The user will need to manually switch that task to IDLE. This is done by performing the following steps:

- Wait for EOF of the viewfinder output channel (IDMAC_EOF_21)
- Set RSW_EN bit
- resume one frame via the viewfinder task.
- Disable the VDI and IC and the corresponding IDMAC channels

38.4.12.10.1 STOP Mode

In this mode the IPU sends the ipu_stby_ack after getting to IPU_IDLE state. The CCM will gate off the clock to the IPU.

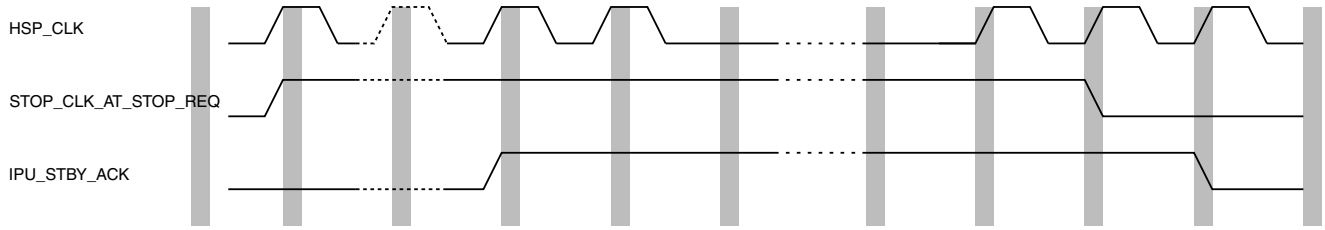


Figure 38-55. Entering and Exiting STOP Mode

Wake up from STOP mode

When the SoC decides to wake up from STOP mode it should:

- Resume the clock to the IPU.
- Negate the stop_clk_at_stop_req or stop_clk_at_wait_req signal.
- The IPU will then negate the ipu_stby_ack signal.
- The IPU will resume screen refresh.

38.4.12.10.2 Low Power Screen Refresh mode - LPSR

In Low Power Screen Refresh mode, the clock to the IPU is changed to a slower frequency, the IPU performs only screen refresh to a single display via the DP (channels 23 & 27).

Preparations

1. The user sets the LPSR_MODE bit indicating that the next assertion of stop_clk_at_stop_req OR stop_clk_at_wait_req activates the LPSR procedure.
2. The user moves the IPU to screen refresh flow. This means that if other tasks are active (flows via CSI or multiple flows to multiple displays) - this tasks needs to be complete and disabled. This is step is fully done by the user (SW). The only flow that remains active is screen refresh to a single display done via the DP (channels 23 & 27)
3. The user stores in the SRM the planned configuration for the sub-blocks involved in the LPSR flow. These configuration will e switched with the active registers' settings on later stage. The relevant sub-blocks are:
   - DI0 & DI1
   - DC
   - DP
   - DMFC
   - IDMAC
   - CM

Entering LPSR

The flow for entering LPSR is the same as stop mode.

1. The IPU follows the same procedure as on STOP mode till getting into IPU_IDLE state.
2. IPU swaps the registers of the relevant blocks with the pre stored content from the SRM. The content of the registers of the current flow is stored in the SRM. The IPU will switch back to this configuration after exiting from LPSR.
3. The IPU sends the ipu_stby_ack
4. The CCM will gate off the clock to the IPU.
5. The CCM will switch to the new clock
6. After changing the clock, the CCM asserts ccm_lpsr_ipu, this signal is synched inside the IPU to the hsp_clk
7. The IPU will resume screen refresh with the new settings.

Exit from LPSR

1. The CCM negates ccm_lpsr_ipu
2. The CM sends standby request to the DI
3. The DI should complete processing the current frame and stop the clock to the display and send an acknowledge signal to the CM.
4. The SRM swaps the registers' configuration. current configuration (LPSR) is saved in the SRM, the previous (original) configuration is stored in the blocks' registers.
5. IPU asserts the ipu_wakeup_ack signal indicating that it is now safe to leave LPSR mode.
6. CCM stops the clocks to IPU
7. CCM resumes the clock to the IPU - This clock is the same clock used prior to entering the LPSR mode.
8. CCM negates the stop_clk_at_stop_req (or stop_clk_at_wait_req)
9. IPU negates the ipu_stby_ack and the ipu_wakeup_ack signals
10. The IPU resumes screen refresh with the original settings.

The diagram below illustrates the procedure for entering and leaving LPSR mode

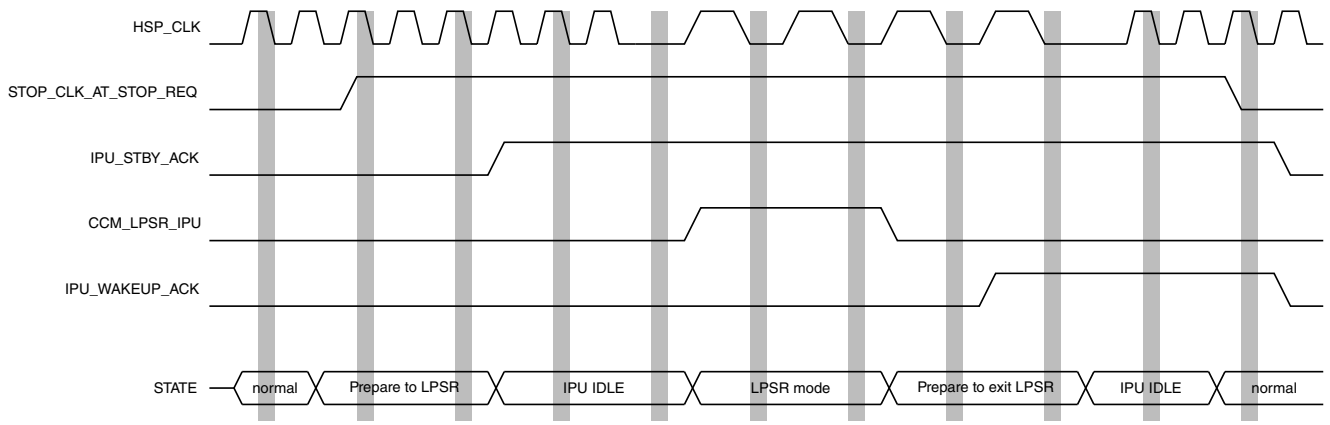


Figure 38-56. Entering and Exiting LPSR mode

38.5 IPU Memory Map/Register Definition

The address space for accesses through the AHB-lite slave port is 4 MB and it is split internally (with 2MB resolution) according to bit [21] of the address. Using the following notation

Address = (IPU_ID[31:25], 1,1,1,MSB[21], LSB[20:0])

the address is used as follows :

1. MSB=0: Low-level access to an external device, with LSB[3:0] = (Lock, CS, RS[1:0])

- LSB[5:4] = RS[1:0] (the address on the display interface)
- LSB[6] = Choice of display's channel (0=channel 8, 1=channel 9)
- LSB[7] = Lock (Lock=1 prevents the use of the display port until the next ARM platform access)

1. MSB=1: access to internal IPU registers, with address LSB

NOTE

The addresses given in the table are relative to the IPU base address defined at SoC's level.

IPU memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
260_0000	Configuration Register (IPU_CONF)	32	R/W	0000_0000h	38.5.1/3014
260_0004	SISG Control 0 Register (IPU_SISG_CTRL0)	32	R/W	0000_0000h	38.5.2/3017
260_0008	SISG Control 1 Register (IPU_SISG_CTRL1)	32	R/W	0000_0000h	38.5.3/3018
260_000C	SISG Set<i> Register (IPU_SISG_SET_i)	32	R/W	0000_0000h	38.5.4/3018
260_0024	SISG Clear <i> Register (IPU_SISG_CLR_i)	32	R/W	0000_0000h	38.5.5/3019
260_003C	Interrupt Control Register 1 (IPU_INT_CTRL_1)	32	R/W	0000_0000h	38.5.6/3020
260_0040	Interrupt Control Register 2 (IPU_INT_CTRL_2)	32	R/W	0000_0000h	38.5.7/3024
260_0044	Interrupt Control Register 3 (IPU_INT_CTRL_3)	32	R/W	0000_0000h	38.5.8/3026
260_0048	Interrupt Control Register 4 (IPU_INT_CTRL_4)	32	R/W	0000_0000h	38.5.9/3030
260_004C	Interrupt Control Register 5 (IPU_INT_CTRL_5)	32	R/W	0000_0000h	38.5.10/ 3033
260_0050	Interrupt Control Register 6 (IPU_INT_CTRL_6)	32	R/W	0000_0000h	38.5.11/ 3038
260_0054	Interrupt Control Register 7 (IPU_INT_CTRL_7)	32	R/W	0000_0000h	38.5.12/ 3041
260_0058	Interrupt Control Register 8 (IPU_INT_CTRL_8)	32	R/W	0000_0000h	38.5.13/ 3043
260_005C	Interrupt Control Register 9 (IPU_INT_CTRL_9)	32	R/W	0000_0000h	38.5.14/ 3045
260_0060	Interrupt Control Register 10 (IPU_INT_CTRL_10)	32	R/W	0000_0000h	38.5.15/ 3047
260_0064	Interrupt Control Register 11 (IPU_INT_CTRL_11)	32	R/W	0000_0000h	38.5.16/ 3049
260_0068	Interrupt Control Register 12 (IPU_INT_CTRL_12)	32	R/W	0000_0000h	38.5.17/ 3052
260_006C	Interrupt Control Register 13 (IPU_INT_CTRL_13)	32	R/W	0000_0000h	38.5.18/ 3054
260_0070	Interrupt Control Register 14 (IPU_INT_CTRL_14)	32	R/W	0000_0000h	38.5.19/ 3058
260_0074	Interrupt Control Register15 (IPU_INT_CTRL_15)	32	R/W	0000_0000h	38.5.20/ 3061
260_0078	SDMA Event Control Register 1 (IPU_SDMA_EVENT_1)	32	R/W	0000_0000h	38.5.21/ 3065
260_007C	SDMA Event Control Register 2 (IPU_SDMA_EVENT_2)	32	R/W	0000_0000h	38.5.22/ 3069
260_0080	SDMA Event Control Register 3 (IPU_SDMA_EVENT_3)	32	R/W	0000_0000h	38.5.23/ 3072
260_0084	SDMA Event Control Register 4 (IPU_SDMA_EVENT_4)	32	R/W	0000_0000h	38.5.24/ 3077
260_0088	SDMA Event Control Register 7 (IPU_SDMA_EVENT_7)	32	R/W	0000_0000h	38.5.25/ 3080
260_008C	SDMA Event Control Register 8 (IPU_SDMA_EVENT_8)	32	R/W	0000_0000h	38.5.26/ 3082

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
260_0090	SDMA Event Control Register 11 (IPU_SDMA_EVENT_11)	32	R/W	0000_0000h	38.5.27/ 3083
260_0094	SDMA Event Control Register 12 (IPU_SDMA_EVENT_12)	32	R/W	0000_0000h	38.5.28/ 3086
260_0098	SDMA Event Control Register 13 (IPU_SDMA_EVENT_13)	32	R/W	0000_0000h	38.5.29/ 3088
260_009C	SDMA Event Control Register 14 (IPU_SDMA_EVENT_14)	32	R/W	0000_0000h	38.5.30/ 3092
260_00A0	Shadow Registers Memory Priority 1 Register (IPU_SRM_PRI1)	32	R/W	0000_0100h	38.5.31/ 3095
260_00A4	Shadow Registers Memory Priority 2 Register (IPU_SRM_PRI2)	32	R/W	0605_0803h	38.5.32/ 3096
260_00A8	FSU Processing Flow 1 Register (IPU_FS_PROC_FLOW1)	32	R/W	0000_0000h	38.5.33/ 3098
260_00AC	FSU Processing Flow 2 Register (IPU_FS_PROC_FLOW2)	32	R/W	0000_0000h	38.5.34/ 3102
260_00B0	FSU Processing Flow 3 Register (IPU_FS_PROC_FLOW3)	32	R/W	0000_0000h	38.5.35/ 3105
260_00B4	FSU Displaying Flow 1 Register (IPU_FS_DISP_FLOW1)	32	R/W	0000_0000h	38.5.36/ 3108
260_00B8	FSU Displaying Flow 2 Register (IPU_FS_DISP_FLOW2)	32	R/W	0000_0000h	38.5.37/ 3111
260_00BC	SKIP Register (IPU_SKIP)	32	R/W	0000_0000h	38.5.38/ 3113
260_00C0	Display Alternate Configuration Register (IPU_DISP_ALT_CONF)	32	R/W	0000_0000h	38.5.39/ 3115
260_00C4	Display General Control Register (IPU_DISP_GEN)	32	R/W	0040_0000h	38.5.39/ 3116
260_00C8	Display Alternate Flow Control Register 1 (IPU_DISP_ALT1)	32	R/W	0040_0000h	38.5.40/ 3119
260_00CC	Display Alternate Flow Control Register 2 (IPU_DISP_ALT2)	32	R/W	0000_0000h	38.5.41/ 3120
260_00D0	Display Alternate Flow Control Register 3 (IPU_DISP_ALT3)	32	R/W	0040_0000h	38.5.42/ 3121
260_00D4	Display Alternate Flow Control Register 4 (IPU_DISP_ALT4)	32	R/W	0000_0000h	38.5.43/ 3123
260_00DC	Memory Reset Control Register (IPU_MEM_RST)	32	R/W	0000_0000h	38.5.44/ 3124
260_00E0	Power Modes Control Register (IPU_PM)	32	R/W	0810_0810h	38.5.45/ 3126
260_00E4	General Purpose Register (IPU_GPR)	32	R/W	0000_0000h	38.5.46/ 3129
260_0150	Channel Double Buffer Mode Select 0 Register (IPU_CH_DB_MODE_SEL0)	32	R/W	0000_0000h	38.5.47/ 3131

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
260_0154	Channel Double Buffer Mode Select 1 Register (IPU_CH_DB_MODE_SEL1)	32	R/W	0000_0000h	38.5.48/ 3135
260_0168	Alternate Channel Double Buffer Mode Select 0 Register (IPU_ALT_CH_DB_MODE_SEL0)	32	R/W	0000_0000h	38.5.49/ 3138
260_016C	Alternate Channel Double Buffer Mode Select1 Register (IPU_ALT_CH_DB_MODE_SEL1)	32	R/W	0000_0000h	38.5.50/ 3140
260_0178	Alternate Channel Triple Buffer Mode Select 0 Register (IPU_ALT_CH_TRB_MODE_SEL0)	32	R/W	0000_0000h	38.5.51/ 3141
260_017C	Alternate Channel Triple Buffer Mode Select 1 Register (IPU_ALT_CH_TRB_MODE_SEL1)	32	R/W	0000_0000h	38.5.52/ 3143
260_0200	Interrupt Status Register 1 (IPU_INT_STAT_1)	32	w1c	0000_0000h	38.5.52/ 3144
260_0204	Interrupt Status Register2 (IPU_INT_STAT_2)	32	w1c	0000_0000h	38.5.53/ 3149
260_0208	Interrupt Status Register 3 (IPU_INT_STAT_3)	32	w1c	0000_0000h	38.5.54/ 3152
260_020C	Interrupt Status Register 4 (IPU_INT_STAT_4)	32	w1c	0000_0000h	38.5.55/ 3156
260_0210	Interrupt Status Register 5 (IPU_INT_STAT_5)	32	w1c	0000_0000h	38.5.56/ 3159
260_0214	Interrupt Status Register 6 (IPU_INT_STAT_6)	32	w1c	0000_0000h	38.5.57/ 3165
260_0218	Interrupt Status Register7 1 (IPU_INT_STAT_7)	32	w1c	0000_0000h	38.5.58/ 3169
260_021C	Interrupt Status Register 8 (IPU_INT_STAT_8)	32	w1c	0000_0000h	38.5.59/ 3172
260_0220	Interrupt Status Register 9 (IPU_INT_STAT_9)	32	w1c	0000_0000h	38.5.60/ 3175
260_0224	Interrupt Status Register 10 (IPU_INT_STAT_10)	32	w1c	0000_0000h	38.5.61/ 3177
260_0228	Interrupt Status Register 11 (IPU_INT_STAT_11)	32	w1c	0000_0000h	38.5.62/ 3180
260_022C	Interrupt Status Register 12 (IPU_INT_STAT_12)	32	w1c	0000_0000h	38.5.63/ 3184
260_0230	Interrupt Status Register 13 (IPU_INT_STAT_13)	32	w1c	0000_0000h	38.5.64/ 3186
260_0234	Interrupt Status Register 14 (IPU_INT_STAT_14)	32	w1c	0000_0000h	38.5.65/ 3191
260_0238	Interrupt Status Register 15 (IPU_INT_STAT_15)	32	w1c	0000_0000h	38.5.66/ 3194
260_023C	Current Buffer Register 0 (IPU_CUR_BUF_0)	32	R	0000_0000h	38.5.67/ 3198
260_0240	Current Buffer Register 1 (IPU_CUR_BUF_1)	32	R	0000_0000h	38.5.68/ 3203

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
260_0244	Alternate Current Buffer Register 0 (IPU_ALT_CUR_0)	32	R	0000_0000h	38.5.69/ 3207
260_0248	Alternate Current Buffer Register 1 (IPU_ALT_CUR_1)	32	R	0000_0000h	38.5.70/ 3209
260_024C	Shadow Registers Memory Status Register (IPU_SRM_STAT)	32	R	0000_0000h	38.5.71/ 3212
260_0250	Processing Status Tasks Register (IPU_PROC_TASKS_STAT)	32	R	0000_0000h	38.5.72/ 3214
260_0254	Display Tasks Status Register (IPU_DISP_TASKS_STAT)	32	R	0000_0000h	38.5.73/ 3216
260_0258	Triple Current Buffer Register 0 (IPU_TRIPLE_CUR_BUF_0)	32	R	0000_0000h	38.5.74/ 3218
260_025C	Triple Current Buffer Register 1 (IPU_TRIPLE_CUR_BUF_1)	32	R	0000_0000h	38.5.75/ 3220
260_0260	Triple Current Buffer Register 2 (IPU_TRIPLE_CUR_BUF_2)	32	R	0000_0000h	38.5.76/ 3221
260_0264	Triple Current Buffer Register 3 (IPU_TRIPLE_CUR_BUF_3)	32	R	0000_0000h	38.5.76/ 3222
260_0268	IPU Channels Buffer 0 Ready 0 Register (IPU_CH_BUF0_RDY0)	32	R/W	0000_0000h	38.5.76/ 3222
260_026C	IPU Channels Buffer 0 Ready 1 Register (IPU_CH_BUF0_RDY1)	32	R/W	0000_0000h	38.5.77/ 3226
260_0270	IPU Channels Buffer 1 Ready 0 Register (IPU_CH_BUF1_RDY0)	32	R/W	0000_0000h	38.5.78/ 3228
260_0274	IPU Channels Buffer 1 Ready 1Register (IPU_CH_BUF1_RDY1)	32	R/W	0000_0000h	38.5.79/ 3231
260_0278	IPU Alternate Channels Buffer 0 Ready 0 Register (IPU_ALT_CH_BUF0_RDY0)	32	R/W	0000_0000h	38.5.80/ 3234
260_027C	IPU Alternate Channels Buffer 0 Ready 1 Register (IPU_ALT_CH_BUF0_RDY1)	32	R/W	0000_0000h	38.5.81/ 3235
260_0280	IPU Alternate Channels Buffer1 Ready 0 Register (IPU_ALT_CH_BUF1_RDY0)	32	R/W	0000_0000h	38.5.82/ 3236
260_0284	IPU Alternate Channels Buffer 1 Ready 1 Register (IPU_ALT_CH_BUF1_RDY1)	32	R/W	0000_0000h	38.5.83/ 3237
260_0288	IPU Channels Buffer 2 Ready 0 Register (IPU_CH_BUF2_RDY0)	32	R/W	0000_0000h	38.5.84/ 3238
260_028C	IPU Channels Buffer 2 Ready 1 Register (IPU_CH_BUF2_RDY1)	32	R/W	0000_0000h	38.5.85/ 3240
260_8000	IDMAC Configuration Register (IPU_IDMAC_CONF)	32	R/W	0000_002Fh	38.5.86/ 3241
260_8004	IDMAC Channel Enable 1 Register (IPU_IDMAC_CH_EN_1)	32	R/W	0000_0000h	38.5.87/ 3243
260_8008	IDMAC Channel Enable 2 Register (IPU_IDMAC_CH_EN_2)	32	R/W	0000_0000h	38.5.88/ 3246

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
260_800C	IDMAC Separate Alpha Indication Register (IPU_IDMAC_SEP_ALPHA)	32	R/W	0000_0000h	38.5.89/ 3248
260_8010	IDMAC Alternate Separate Alpha Indication Register (IPU_IDMAC_ALT_SEP_ALPHA)	32	R/W	0000_0000h	38.5.90/ 3250
260_8014	IDMAC Channel Priority 1 Register (IPU_IDMAC_CH_PRI_1)	32	R/W	0000_0000h	38.5.91/ 3252
260_8018	IDMAC Channel Priority 2 Register (IPU_IDMAC_CH_PRI_2)	32	R/W	0000_0000h	38.5.92/ 3255
260_801C	IDMAC Channel Watermark Enable 1 Register (IPU_IDMAC_WM_EN_1)	32	R/W	0000_0000h	38.5.93/ 3257
260_8020	IDMAC Channel Watermark Enable 2 Register (IPU_IDMAC_WM_EN_2)	32	R/W	0000_0000h	38.5.94/ 3259
260_8024	IDMAC Channel Lock Enable 1Register (IPU_IDMAC_LOCK_EN_1)	32	R/W	0000_0000h	38.5.95/ 3260
260_8028	IDMAC Channel Lock Enable 2Register (IPU_IDMAC_LOCK_EN_2)	32	R/W	0000_0000h	38.5.96/ 3262
260_802C	IDMAC Channel Alternate Address 0 Register (IPU_IDMAC_SUB_ADDR_0)	32	R/W	0000_0000h	38.5.97/ 3263
260_8030	IDMAC Channel Alternate Address 1 Register (IPU_IDMAC_SUB_ADDR_1)	32	R/W	0000_0000h	38.5.98/ 3264
260_8034	IDMAC Channel Alternate Address 2 Register (IPU_IDMAC_SUB_ADDR_2)	32	R/W	0000_0000h	38.5.99/ 3265
260_8038	IDMAC Channel Alternate Address 3 Register (IPU_IDMAC_SUB_ADDR_3)	32	R/W	0000_0000h	38.5.100/ 3266
260_803C	IDMAC Channel Alternate Address 4 Register (IPU_IDMAC_SUB_ADDR_4)	32	R/W	0000_0000h	38.5.101/ 3268
260_8040	IDMAC Band Mode Enable 1 Register (IPU_IDMAC_BNDM_EN_1)	32	R/W	0000_0000h	38.5.102/ 3269
260_8044	IDMAC Band Mode Enable 2 Register (IPU_IDMAC_BNDM_EN_2)	32	R/W	0000_0000h	38.5.103/ 3272
260_8048	IDMAC Scroll Coordinations Register (IPU_IDMAC_SC_CORD)	32	R/W	0000_0000h	38.5.104/ 3273
260_804C	IDMAC Scroll Coordinations Register 1 (IPU_IDMAC_SC_CORD_1)	32	R/W	0000_0000h	38.5.105/ 3274
260_8100	IDMAC Channel Busy 1 Register (IPU_IDMAC_CH_BUSY_1)	32	R	0000_0000h	38.5.106/ 3275
260_8104	IDMAC Channel Busy 2 Register (IPU_IDMAC_CH_BUSY_2)	32	R	0000_0000h	38.5.107/ 3281
261_8000	DP Common Configuration Sync Flow Register (IPU_DP_COM_CONF_SYNC)	32	R/W	0000_0000h	38.5.108/ 3285
261_8004	DP Graphic Window Control Sync Flow Register (IPU_DP_Graph_Wind_CTRL_SYNC)	32	R/W	0000_0000h	38.5.109/ 3287
261_8008	DP Partial Plane Window Position Sync Flow Register (IPU_DP_FG_POS_SYNC)	32	R/W	0000_0000h	38.5.110/ 3288

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
261_800C	DP Cursor Position and Size Sync Flow Register (IPU_DP_CUR_POS_SYNC)	32	R/W	0000_0000h	38.5.111/ 3288
261_8010	DP Color Cursor Mapping Sync Flow Register (IPU_DP_CUR_MAP_SYNC)	32	R/W	0000_0000h	38.5.112/ 3289
261_8014	DP Gamma Constants Sync Flow Register i (IPU_DP_GAMMA_C_SYNC_i)	32	R/W	0000_0000h	38.5.113/ 3290
261_8034	DP Gamma Correction Slope Sync Flow Register i (IPU_DP_GAMMA_S_SYNC_i)	32	R/W	0000_0000h	38.5.114/ 3291
261_8044	DP Color Space Conversion Control Sync Flow Registers (IPU_DP_CSCA_SYNC_i)	32	R/W	0000_0000h	38.5.115/ 3291
261_8054	DP Color Conversion Control Sync Flow Register 0 (IPU_DP_SCS_SYNC_0)	32	R/W	0000_0000h	38.5.116/ 3292
261_8058	DP Color Conversion Control Sync Flow Register 1 (IPU_DP_SCS_SYNC_1)	32	R/W	0000_0000h	38.5.117/ 3293
261_805C	DP Cursor Position and Size Alternate Register (IPU_DP_CUR_POS_ALT)	32	R/W	0000_0000h	38.5.118/ 3294
261_8060	DP Common Configuration Async 0 Flow Register (IPU_DP_COM_CONF_ASYNC0)	32	R/W	0000_0000h	38.5.119/ 3295
261_8064	DP Graphic Window Control Async 0 Flow Register (IPU_DP_GRAPH_WIND_CTRL_ASYNC0)	32	R/W	0000_0000h	38.5.120/ 3297
261_8068	DP Partial Plane Window Position Async 0 Flow Register (IPU_DP_FG_POS_ASYNC0)	32	R/W	0000_0000h	38.5.121/ 3298
261_806C	DP Cursor Position and Size Async 0 Flow Register (IPU_DP_CUR_POS_ASYNC0)	32	R/W	0000_0000h	38.5.122/ 3298
261_8070	DP Color Cursor Mapping Async 0 Flow Register (IPU_DP_CUR_MAP_ASYNC0)	32	R/W	0000_0000h	38.5.123/ 3299
261_8074	DP Gamma Constant Async 0 Flow Register i (IPU_DP_GAMMA_C_ASYNC0_i)	32	R/W	0000_0000h	38.5.124/ 3300
261_8094	DP Gamma Correction Slope Async 0 Flow Register i (IPU_DP_GAMMA_S_ASYNC0_i)	32	R/W	0000_0000h	38.5.125/ 3301
261_80A4	DP Color Space Conversion Control Async 0 Flow Register i (IPU_DP_CSCA_ASYNC0_i)	32	R/W	0000_0000h	38.5.126/ 3301
261_80B4	DP Color Conversion Control Async 0 Flow Register 0 (IPU_DP_CSC_ASYNC0_0)	32	R/W	0000_0000h	38.5.127/ 3302
261_80B8	DP Color Conversion Control Async 1 Flow Register (IPU_DP_CSC_ASYNC_1)	32	R/W	0000_0000h	38.5.128/ 3303
261_80BC	DP Common Configuration Async 1 Flow Register (IPU_DP_COM_CONF_ASYNC1)	32	R/W	0000_0000h	38.5.129/ 3304
261_80BC	DP Debug Control Register (IPU_DP_DEBUG_CNT)	32	R/W	0000_0000h	38.5.130/ 3306
261_80C0	DP Graphic Window Control Async 1 Flow Register (IPU_DP_GRAPH_WIND_CTRL_ASYNC1)	32	R/W	0000_0000h	38.5.131/ 3307
261_80C0	DP Debug Status Register (IPU_DP_DEBUG_STAT)	32	R	0000_0000h	38.5.132/ 3308

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
261_80C4	DP Partial Plane Window Position Async 1 Flow Register (IPU_DP_FG_POS_ASYNC1)	32	R/W	0000_0000h	38.5.133/ 3310
261_80C8	DP Cursor Postion and Size Async 1 Flow Register (IPU_DP_CUR_POS_ASYNC1)	32	R/W	0000_0000h	38.5.134/ 3310
261_80CC	DP Color Cursor Mapping Async 1 Flow Register (IPU_DP_CUR_MAP_ASYNC1)	32	R/W	0000_0000h	38.5.135/ 3311
261_80D0	DP Gamma Constants Async 1 Flow Register i (IPU_DP_GAMMA_C_ASYNC1_i)	32	R/W	0000_0000h	38.5.136/ 3312
261_80F0	DP Gamma Correction Slope Async 1 Flow Register i (IPU_DP_GAMMA_S_ASYN1_i)	32	R/W	0000_0000h	38.5.137/ 3313
261_8100	DP Color Space Converstion Control Async 1 Flow Register i (IPU_DP_CSCA_ASYNC1_i)	32	R/W	0000_0000h	38.5.138/ 3313
261_8110	DP Color Conversion Control Async 1 Flow Register 0 (IPU_DP_CSC_ASYNC1_0)	32	R/W	0000_0000h	38.5.139/ 3314
261_8114	DP Color Conversion Control Async 1 Flow Register 1 (IPU_DP_CSC_ASYNC1_1)	32	R/W	0000_0000h	38.5.140/ 3315
262_0000	IC Configuration Register (IPU_IC_CONF)	32	R/W	0000_0000h	38.5.141/ 3316
262_0004	IC Preprocessing Encoder Resizing Coefficients Register (IPU_IC_PRP_ENC_RSC)	32	R/W	2000_2000h	38.5.142/ 3318
262_0008	IC Preprocessing View-Finder Resizing Coefficients Register (IPU_IC_PRP_VF_RSC)	32	R/W	2000_2000h	38.5.143/ 3319
262_000C	IC Postprocessing Encoder Resizing Coefficients Register (IPU_IC_PP_RSC)	32	R/W	2000_2000h	38.5.144/ 3320
262_0010	IC Combining Parameters Register 1 (IPU_IC_CMBP_1)	32	R/W	0000_0000h	38.5.145/ 3321
262_0014	IC Combining Parameters Register 2 (IPU_IC_CMBP_2)	32	R/W	0000_0000h	38.5.146/ 3321
262_0018	IC IDMAC Parameters 1 Register (IPU_IC_IDMAC_1)	32	R/W	0000_0000h	38.5.147/ 3322
262_001C	IC IDMAC Parameters 2 Register (IPU_IC_IDMAC_2)	32	R/W	0000_0000h	38.5.148/ 3325
262_0020	IC IDMAC Parameters 3Register (IPU_IC_IDMAC_3)	32	R/W	0000_0000h	38.5.149/ 3326
262_0024	IC IDMAC Parameters 4 Register (IPU_IC_IDMAC_4)	32	R/W	0000_0000h	38.5.150/ 3326
263_0000	CSI0 Sensor Configuration Register (IPU_CSI0_SENS_CONF)	32	R/W	0000_0000h	38.5.151/ 3327
263_0004	CSI0 Sense Frame Size Register (IPU_CSI0_SENS_FRM_SIZE)	32	R/W	0000_0000h	38.5.152/ 3330
263_0008	CSI0 Actual Frame Size Register (IPU_CSI0_ACT_FRM_SIZE)	32	R/W	0000_0000h	38.5.153/ 3330
263_000C	CSI0 Output Control Register (IPU_CSI0_OUT_FRM_CTRL)	32	R/W	0000_0000h	38.5.154/ 3331

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
263_0010	CSI0 Test Control Register (IPU_CSI0_TST_CTRL)	32	R/W	0000_0000h	38.5.155/ 3332
263_0014	CSI0 CCIR Code Register 1 (IPU_CSI0_CCIR_CODE_1)	32	R/W	0000_0000h	38.5.156/ 3333
263_0018	CSI0 CCIR Code Register 2 (IPU_CSI0_CCIR_CODE_2)	32	R/W	0000_0000h	38.5.157/ 3334
263_001C	CSI0 CCIR Code Register 3 (IPU_CSI0_CCIR_CODE_3)	32	R/W	0000_0000h	38.5.158/ 3335
263_0020	CSI0 Data Identifier Register (IPU_CSI0_DI)	32	R/W	FFFF_FFFFh	38.5.159/ 3335
263_0024	CSI0 SKIP Register (IPU_CSI0_SKIP)	32	R/W	0000_0000h	38.5.160/ 3336
263_0028	CSI0 Compander Control Register (IPU_CSI0_CPD_CTRL)	32	R/W	0000_0000h	38.5.161/ 3337
263_002C	CSI0 Red Component Compander Constants Register <i> (IPU_CSI0_CPD_RC_i)	32	R/W	0000_0000h	38.5.162/ 3338
263_004C	CSI0 Red Component Compander SLOPE Register <i> (IPU_CSI0_CPD_RS_i)	32	R/W	0000_0000h	38.5.163/ 3339
263_005C	CSI0 GR Component Compander Constants Register <i> (IPU_CSI0_CPD_GRC_i)	32	R/W	0000_0000h	38.5.164/ 3339
263_007C	CSI0 GR Component Compander SLOPE Register <i> (IPU_CSI0_CPD_GRS_i)	32	R/W	0000_0000h	38.5.165/ 3340
263_008C	CSI0 GB Component Compander Constants Register <i> (IPU_CSI0_CPD_GBC_i)	32	R/W	0000_0000h	38.5.166/ 3341
263_00AC	CSI0 GB Component Compander SLOPE Register <i> (IPU_CSI0_CPD_GBS_i)	32	R/W	0000_0000h	38.5.167/ 3341
263_00BC	CSI0 Blue Component Compander Constants Register <i> (IPU_CSI0_CPD_BC_i)	32	R/W	0000_0000h	38.5.168/ 3342
263_00DC	CSI0 Blue Component Compander SLOPE Register <i> (IPU_CSI0_CPD_BS_i)	32	R/W	0000_0000h	38.5.169/ 3343
263_00EC	CSI0 Compander Offset Register 1 (IPU_CSI0_CPD_OFFSET1)	32	R/W	0000_0000h	38.5.170/ 3343
263_00F0	CSI0 Compander Offset Register 2 (IPU_CSI0_CPD_OFFSET2)	32	R/W	0000_0000h	38.5.171/ 3344
263_8000	CSI1 Sensor Configuration Register (IPU_CSI1_SENS_CONF)	32	R/W	0000_0000h	38.5.172/ 3345
263_8004	CSI1 Sense Frame Size Register (IPU_CSI1_SENS_FRM_SIZE)	32	R/W	0000_0000h	38.5.173/ 3347
263_8008	CSI1 Actual Frame Size Register (IPU_CSI1_ACT_FRM_SIZE)	32	R/W	0000_0000h	38.5.174/ 3348
263_800C	CSI1 Output Control Register (IPU_CSI1_OUT_FRM_CTRL)	32	R/W	0000_0000h	38.5.175/ 3349
263_8010	CSI1 Test Control Register (IPU_CSI1_TST_CTRL)	32	R/W	0000_0000h	38.5.176/ 3350

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
263_8014	CSI1 CCIR Code Register 1 (IPU_CSI1_CCIR_CODE_1)	32	R/W	0000_0000h	38.5.177/ 3351
263_8018	CSI1 CCIR Code Register 2 (IPU_CSI1_CCIR_CODE_2)	32	R/W	0000_0000h	38.5.178/ 3352
263_801C	CSI1 CCIR Code Register 3 (IPU_CSI1_CCIR_CODE_3)	32	R/W	0000_0000h	38.5.179/ 3353
263_8020	CSI1 Data Identifier Register (IPU_CSI1_DI)	32	R/W	FFFF_FFFFh	38.5.180/ 3353
263_8024	CSI1 SKIP Register (IPU_CSI1_SKIP)	32	R/W	0000_0000h	38.5.181/ 3354
263_8028	CSI1 Compander Control Register (IPU_CSI1_CPD_CTRL)	32	R/W	0000_0000h	38.5.182/ 3355
263_802C	CSI1 Red Component Compander Constants Register <i> (IPU_CSI1_CPD_RC_i)	32	R/W	0000_0000h	38.5.183/ 3356
263_804C	CSI1 Red Component Compander SLOPE Register <i> (IPU_CSI1_CPD_RS_i)	32	R/W	0000_0000h	38.5.184/ 3356
263_805C	CSI1 GR Component Compander Constants Register <i> (IPU_CSI1_CPD_GRC_i)	32	R/W	0000_0000h	38.5.185/ 3357
263_807C	CSI1 GR Component Compander SLOPE Register <i> (IPU_CSI1_CPD_GRS_i)	32	R/W	0000_0000h	38.5.186/ 3358
263_808C	CSI1 GB Component Compander Constants Register <i> (IPU_CSI1_CPD_GBC_i)	32	R/W	0000_0000h	38.5.187/ 3358
263_80AC	CSI1 GB Component Compander SLOPE Register <i> (IPU_CSI1_CPD_GBS_i)	32	R/W	0000_0000h	38.5.188/ 3359
263_80BC	CSI1 Blue Component Compander Constants Register <i> (IPU_CSI1_CPD_BC_i)	32	R/W	0000_0000h	38.5.189/ 3360
263_80DC	CSI1 Blue Component Compander SLOPE Register <i> (IPU_CSI1_CPD_BS_i)	32	R/W	0000_0000h	38.5.190/ 3360
263_80EC	CSI1 Compander Offset Register 1 (IPU_CSI1_CPD_OFFSET1)	32	R/W	0000_0000h	38.5.191/ 3361
263_80F0	CSI1 Compander Offset Register 2 (IPU_CSI1_CPD_OFFSET2)	32	R/W	0000_0000h	38.5.192/ 3362
264_0000	DI0 General Register (IPU_DI0_GENERAL)	32	R/W	0020_0000h	38.5.193/ 3363
264_0004	DI0 Base Sync Clock Gen 0 Register (IPU_DI0_BS_CLKGEN0)	32	R/W	0000_0000h	38.5.194/ 3365
264_0008	DI0 Base Sync Clock Gen 1 Register (IPU_DI0_BS_CLKGEN1)	32	R/W	0000_0000h	38.5.195/ 3366
264_000C	DI0 Sync Wave Gen 1 Register 0 (IPU_DI0_SW_GEN0_1)	32	R/W	0000_0000h	38.5.196/ 3367
264_0010	DI0 Sync Wave Gen 2 Register 0 (IPU_DI0_SW_GEN0_2)	32	R/W	0000_0000h	38.5.197/ 3368
264_0014	DI0 Sync Wave Gen 3 Register 0 (IPU_DI0_SW_GEN0_3)	32	R/W	0000_0000h	38.5.198/ 3369

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
264_0018	DI0 Sync Wave Gen 4 Register 0 (IPU_DI0_SW_GEN0_4)	32	R/W	0000_0000h	38.5.199/ 3370
264_001C	DI0 Sync Wave Gen 5 Register 0 (IPU_DI0_SW_GEN0_5)	32	R/W	0000_0000h	38.5.200/ 3372
264_0020	DI0 Sync Wave Gen 6 Register 0 (IPU_DI0_SW_GEN0_6)	32	R/W	0000_0000h	38.5.201/ 3373
264_0024	DI0 Sync Wave Gen 7 Register 0 (IPU_DI0_SW_GEN0_7)	32	R/W	0000_0000h	38.5.202/ 3374
264_0028	DI0 Sync Wave Gen 8 Register 0 (IPU_DI0_SW_GEN0_8)	32	R/W	0000_0000h	38.5.203/ 3375
264_002C	DI0 Sync Wave Gen 9 Register 0 (IPU_DI0_SW_GEN0_9)	32	R/W	0000_0000h	38.5.204/ 3376
264_0030	DI0 Sync Wave Gen 1 Register 1 (IPU_DI0_SW_GEN1_1)	32	R/W	0000_0000h	38.5.205/ 3378
264_0034	DI0 Sync Wave Gen 2 Register 1 (IPU_DI0_SW_GEN1_2)	32	R/W	0000_0000h	38.5.206/ 3380
264_0038	DI0 Sync Wave Gen 3 Register 1 (IPU_DI0_SW_GEN1_3)	32	R/W	0000_0000h	38.5.207/ 3382
264_003C	DI0 Sync Wave Gen 4 Register 1 (IPU_DI0_SW_GEN1_4)	32	R/W	0000_0000h	38.5.208/ 3384
264_0040	DI0 Sync Wave Gen 5 Register 1 (IPU_DI0_SW_GEN1_5)	32	R/W	0000_0000h	38.5.209/ 3386
264_0044	DI0 Sync Wave Gen 6 Register 1 (IPU_DI0_SW_GEN1_6)	32	R/W	0000_0000h	38.5.210/ 3388
264_0048	DI0 Sync Wave Gen 7 Register 1 (IPU_DI0_SW_GEN1_7)	32	R/W	0000_0000h	38.5.211/ 3390
264_004C	DI0 Sync Wave Gen 8 Register 1 (IPU_DI0_SW_GEN1_8)	32	R/W	0000_0000h	38.5.212/ 3392
264_0050	DI0 Sync Wave Gen 9 Register 1 (IPU_DI0_SW_GEN1_9)	32	R/W	0000_0000h	38.5.213/ 3394
264_0054	DI0 Sync Assistance Gen Register (IPU_DI0_SYNC_AS_GEN)	32	R/W	0000_0000h	38.5.214/ 3395
264_0058	DI0 Data Wave Gen <i> Register (IPU_DI0_DW_GEN_i)	32	R/W	0000_0000h	38.5.215/ 3396
264_0088	DI0 Data Wave Set 0 <i> Register (IPU_DI0_DW_SET0_i)	32	R/W	0000_0000h	38.5.216/ 3398
264_00B8	DI0 Data Wave Set 1 <i> Register (IPU_DI0_DW_SET1_i)	32	R/W	0000_0000h	38.5.217/ 3398
264_00E8	DI0 Data Wave Set 2 <i> Register (IPU_DI0_DW_SET2_i)	32	R/W	0000_0000h	38.5.218/ 3399
264_0118	DI0 Data Wave Set 3 <i> Register (IPU_DI0_DW_SET3_i)	32	R/W	0000_0000h	38.5.219/ 3399
264_0148	DI0 Step Repeat <i> Registers (IPU_DI0_STP_REP_i)	32	R/W	0000_0000h	38.5.220/ 3400

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
264_0158	DI0 Step Repeat 9 Registers (IPU_DI0_STP_REP_9)	32	R/W	0000_0000h	38.5.221/ 3400
264_015C	DI0 Serial Display Control Register (IPU_DI0_SER_CONF)	32	R/W	0000_0000h	38.5.222/ 3401
264_0160	DI0 Special Signals Control Register (IPU_DI0_SSC)	32	R/W	0000_0000h	38.5.223/ 3403
264_0164	DI0 Polarity Register (IPU_DI0_POL)	32	R/W	0000_0000h	38.5.224/ 3406
264_0168	DI0 Active Window 0 Register (IPU_DI0_AW0)	32	R/W	0000_0000h	38.5.225/ 3407
264_016C	DI0 Active Window 1 Register (IPU_DI0_AW1)	32	R/W	0000_0000h	38.5.226/ 3408
264_0170	DI0 Screen Configuration Register (IPU_DI0_SCR_CONF)	32	R/W	0000_0000h	38.5.227/ 3409
264_0174	DI0 Status Register (IPU_DI0_STAT)	32	R	0000_0005h	38.5.228/ 3410
264_8000	DI1General Register (IPU_DI1_GENERAL)	32	R/W	0020_0000h	38.5.229/ 3412
264_8004	DI1 Base Sync Clock Gen 0 Register (IPU_DI1_BS_CLKGEN0)	32	R/W	0000_0000h	38.5.230/ 3414
264_8008	DI1 Base Sync Clock Gen 1 Register (IPU_DI1_BS_CLKGEN1)	32	R/W	0000_0000h	38.5.231/ 3415
264_800C	DI1 Sync Wave Gen 1 Register 0 (IPU_DI1_SW_GEN0_1)	32	R/W	0000_0000h	38.5.232/ 3416
264_8010	DI1 Sync Wave Gen 2 Register 0 (IPU_DI1_SW_GEN0_2)	32	R/W	0000_0000h	38.5.233/ 3417
264_8014	DI1 Sync Wave Gen 3 Register 0 (IPU_DI1_SW_GEN0_3)	32	R/W	0000_0000h	38.5.234/ 3418
264_8018	DI1 Sync Wave Gen 4 Register 0 (IPU_DI1_SW_GEN0_4)	32	R/W	0000_0000h	38.5.235/ 3419
264_801C	DI1 Sync Wave Gen 5 Register 0 (IPU_DI1_SW_GEN0_5)	32	R/W	0000_0000h	38.5.236/ 3421
264_8020	DI1 Sync Wave Gen 6 Register 0 (IPU_DI1_SW_GEN0_6)	32	R/W	0000_0000h	38.5.237/ 3422
264_8024	DI1 Sync Wave Gen 7 Register 0 (IPU_DI1_SW_GEN0_7)	32	R/W	0000_0000h	38.5.238/ 3423
264_8028	DI1 Sync Wave Gen 8 Register 0 (IPU_DI1_SW_GEN0_8)	32	R/W	0000_0000h	38.5.239/ 3424
264_802C	DI1Sync Wave Gen 9 Register 0 (IPU_DI1_SW_GEN0_9)	32	R/W	0000_0000h	38.5.240/ 3425
264_8030	DI1 Sync Wave Gen 1 Register 1 (IPU_DI1_SW_GEN1_1)	32	R/W	0000_0000h	38.5.241/ 3427
264_8034	DI1 Sync Wave Gen 2 Register 1 (IPU_DI1_SW_GEN1_2)	32	R/W	0000_0000h	38.5.242/ 3429

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
264_8038	DI1 Sync Wave Gen 3 Register 1 (IPU_DI1_SW_GEN1_3)	32	R/W	0000_0000h	38.5.243/ 3431
264_803C	DI1 Sync Wave Gen 4 Register 1 (IPU_DI1_SW_GEN1_4)	32	R/W	0000_0000h	38.5.244/ 3433
264_8040	DI1 Sync Wave Gen 5 Register 1 (IPU_DI1_SW_GEN1_5)	32	R/W	0000_0000h	38.5.245/ 3435
264_8044	DI1 Sync Wave Gen 6 Register 1 (IPU_DI1_SW_GEN1_6)	32	R/W	0000_0000h	38.5.246/ 3437
264_8048	DI1Sync Wave Gen 7 Register 1 (IPU_DI1_SW_GEN1_7)	32	R/W	0000_0000h	38.5.247/ 3439
264_804C	DI1 Sync Wave Gen 8 Register 1 (IPU_DI1_SW_GEN1_8)	32	R/W	0000_0000h	38.5.248/ 3441
264_8050	DI1 Sync Wave Gen 9 Register 1 (IPU_DI1_SW_GEN1_9)	32	R/W	0000_0000h	38.5.249/ 3443
264_8054	DI1 Sync Assistance Gen Register (IPU_DI1_SYNC_AS_GEN)	32	R/W	0000_0000h	38.5.250/ 3444
264_8058	DI1 Data Wave Gen <i> Register (IPU_DI1_DW_GEN_i)	32	R/W	0000_0000h	38.5.251/ 3445
264_8088	DI1 Data Wave Set 0 <i> Register (IPU_DI1_DW_SET0_i)	32	R/W	0000_0000h	38.5.252/ 3447
264_80B8	DI1 Data Wave Set 1 <i> Register (IPU_DI1_DW_SET1_i)	32	R/W	0000_0000h	38.5.253/ 3447
264_80E8	DI1 Data Wave Set 2 <i> Register (IPU_DI1_DW_SET2_i)	32	R/W	0000_0000h	38.5.254/ 3448
264_8118	DI1 Data Wave Set 3 <i> Register (IPU_DI1_DW_SET3_i)	32	R/W	0000_0000h	38.5.255/ 3448
264_8148	DI1 Step Repeat <i> Registers (IPU_D1_STP_REP_i)	32	R/W	0000_0000h	38.5.256/ 3449
264_8158	DI1Step Repeat 9 Registers (IPU_DI1_STP_REP_9)	32	R/W	0000_0000h	38.5.257/ 3449
264_815C	DI1 Serial Display Control Register (IPU_DI1_SER_CONF)	32	R/W	0000_0000h	38.5.258/ 3450
264_8160	DI1 Special Signals Control Register (IPU_DI1_SSC)	32	R/W	0000_0000h	38.5.259/ 3452
264_8164	DI1 Polarity Register (IPU_DI1_POL)	32	R/W	0000_0000h	38.5.260/ 3455
264_8168	DI1Active Window 0 Register (IPU_DI1_AW0)	32	R/W	0000_0000h	38.5.261/ 3456
264_816C	DI1 Active Window 1 Register (IPU_DI1_AW1)	32	R/W	0000_0000h	38.5.262/ 3457
264_8170	DI1 Screen Configuration Register (IPU_DI1_SCR_CONF)	32	R/W	0000_0000h	38.5.263/ 3458
264_8174	DI1 Status Register (IPU_DI1_STAT)	32	R	0000_0005h	38.5.264/ 3459

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
265_0000	SMFC Mapping Register (IPU_SMFC_MAP)	32	R/W	0000_0000h	38.5.265/ 3460
265_0004	SMFC Watermark Control Register (IPU_SMFC_WMC)	32	R/W	0000_09A6h	38.5.266/ 3461
265_0008	SMFC Burst Size Register (IPU_SMFC_BS)	32	R/W	0000_0000h	38.5.267/ 3463
265_8000	DC Read Channel Configuration Register (IPU_DC_READ_CH_CONF)	32	R/W	FFFF_0000h	38.5.268/ 3464
265_8004	DC Read Channel Start Address Register (IPU_DC_READ_SH_ADDR)	32	R/W	0000_0000h	38.5.269/ 3466
265_8008	DC Routine Link Register 0 Channel 0 (IPU_DC_RL0_CH_0)	32	R/W	0000_0000h	38.5.270/ 3466
265_800C	DC Routine Link Register 1 Channel 0 (IPU_DC_RL1_CH_0)	32	R/W	0000_0000h	38.5.271/ 3467
265_8010	DC Routine Link Register2 Channel 0 (IPU_DC_RL2_CH_0)	32	R/W	0000_0000h	38.5.272/ 3468
265_8014	DC Routine Link Registe3 Channel 0 (IPU_DC_RL3_CH_0)	32	R/W	0000_0000h	38.5.273/ 3469
265_8018	DC Routine Link Register 4 Channel 0 (IPU_DC_RL4_CH_0)	32	R/W	0000_0000h	38.5.274/ 3470
265_801C	DC Write Channel 1 Configuration Register (IPU_DC_WR_CH_CONF_1)	32	R/W	0000_0000h	38.5.275/ 3471
265_8020	DC Write Channel 1 Address Configuration Register (IPU_DC_WR_CH_ADDR_1)	32	R/W	0000_0000h	38.5.276/ 3473
265_8024	DC Routine Link Register 0 Channel 1 (IPU_DC_RL0_CH_1)	32	R/W	0000_0000h	38.5.277/ 3473
265_8028	DC Routine Link Register 1 Channel 1 (IPU_DC_RL1_CH_1)	32	R/W	0000_0000h	38.5.278/ 3474
265_8030	DC Routine Link Register 2 Channel 1 (IPU_DC_RL2_CH_1)	32	R/W	0000_0000h	38.5.279/ 3475
265_8032	DC Routine Link Register 3 Channel 1 (IPU_DC_RL3_CH_1)	32	R/W	0000_0000h	38.5.280/ 3476
265_8034	DC Routine Link Register 4 Channel 1 (IPU_DC_RL4_CH_1)	32	R/W	0000_0000h	38.5.281/ 3477
265_8038	DC Write Channel 2 Configuration Register (IPU_DC_WR_CH_CONF_2)	32	R/W	0000_0000h	38.5.282/ 3478
265_803C	DC Write Channel 2 Address Configuration Register (IPU_DC_WR_CH_ADDR_2)	32	R/W	0000_0000h	38.5.283/ 3479
265_8040	DC Routine Link Register 0 Channel 2 (IPU_DC_RL0_CH_2)	32	R/W	0000_0000h	38.5.284/ 3480
265_8044	DC Routine Link Register 1 Channel 2 (IPU_DC_RL1_CH_2)	32	R/W	0000_0000h	38.5.285/ 3481
265_8048	DC Routine Link Register 2 Channel 2 (IPU_DC_RL2_CH_2)	32	R/W	0000_0000h	38.5.286/ 3482

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
265_804C	DC Routine Link Register 3 Channel 2 (IPU_DC_RL3_CH_2)	32	R/W	0000_0000h	38.5.287/ 3483
265_8050	DC Routine Link Register 4 Channel 2 (IPU_DC_RL4_CH_2)	32	R/W	0000_0000h	38.5.288/ 3484
265_8054	DC Command Channel 3 Configuration Register (IPU_DC_CMD_CH_CONF_3)	32	R/W	0000_0000h	38.5.289/ 3485
265_8058	DC Command Channel 4 Configuration Register (IPU_DC_CMD_CH_CONF_4)	32	R/W	0000_0000h	38.5.290/ 3486
265_805C	DC Write Channel 5Configuration Register (IPU_DC_WR_CH_CONF_5)	32	R/W	0000_0000h	38.5.291/ 3487
265_8060	DC Write Channel 5Address Configuration Register (IPU_DC_WR_CH_ADDR_5)	32	R/W	0000_0000h	38.5.292/ 3488
265_8064	DC Routine Link Register 0 Channel 5 (IPU_DC_RL0_CH_5)	32	R/W	0000_0000h	38.5.293/ 3489
265_8068	DC Routine Link Register 1 Channel 5 (IPU_DC_RL1_CH_5)	32	R/W	0000_0000h	38.5.294/ 3490
265_806C	DC Routine Link Register 2 Channel 5 (IPU_DC_RL2_CH_5)	32	R/W	0000_0000h	38.5.295/ 3491
265_8070	DC Routine Link Register3 Channel 5 (IPU_DC_RL3_CH_5)	32	R/W	0000_0000h	38.5.296/ 3492
265_8074	DC Routine Link Register 4 Channel 5 (IPU_DC_RL4_CH_5)	32	R/W	0000_0000h	38.5.297/ 3493
265_8078	DC Write Channel 6 Configuration Register (IPU_DC_WR_CH_CONF_6)	32	R/W	0000_0000h	38.5.298/ 3494
265_807C	DC Write Channel 6 Address Configuration Register (IPU_DC_WR_CH_ADDR_6)	32	R/W	0000_0000h	38.5.299/ 3495
265_8080	DC Routine Link Register 0Channel 6 (IPU_DC_RL0_CH_6)	32	R/W	0000_0000h	38.5.300/ 3496
265_8084	DC Routine Link Register 1 Channel 6 (IPU_DC_RL1_CH_6)	32	R/W	0000_0000h	38.5.301/ 3497
265_8088	DC Routine Link Register 2 Channel 6 (IPU_DC_RL2_CH_6)	32	R/W	0000_0000h	38.5.302/ 3498
265_808C	DC Routine Link Register 3 Channel 6 (IPU_DC_RL3_CH_6)	32	R/W	0000_0000h	38.5.303/ 3499
265_8090	DC Routine Link Register 4 Channel 6 (IPU_DC_RL4_CH_6)	32	R/W	0000_0000h	38.5.304/ 3500
265_8094	DC Write Channel 8 Configuration 1Register (IPU_DC_WR_CH_CONF1_8)	32	R/W	0000_0000h	38.5.305/ 3501
265_8098	DC Write Channel 8 Configuration 2 Register (IPU_DC_WR_CH_CONF2_8)	32	R/W	0000_0000h	38.5.306/ 3502
265_809C	DC Routine Link Register 1 Channel 8 (IPU_DC_RL1_CH_8)	32	R/W	0000_0000h	38.5.307/ 3502
265_80A0	DC Routine Link Register 2 Channel 8 (IPU_DC_RL2_CH_8)	32	R/W	0000_0000h	38.5.308/ 3503

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
265_80A4	DC Routine Link Register 3 Channel 8 (IPU_DC_RL3_CH_8)	32	R/W	0000_0000h	38.5.309/ 3504
265_80A8	DC Routine Link Register 4 Channel 8 (IPU_DC_RL4_CH_8)	32	R/W	0000_0000h	38.5.310/ 3504
265_80AC	DC Routine Link Register 5 Channel 8 (IPU_DC_RL5_CH_8)	32	R/W	0000_0000h	38.5.311/ 3505
265_80B0	DC Routine Link Register 6 Channel 8 (IPU_DC_RL6_CH_8)	32	R/W	0000_0000h	38.5.312/ 3506
265_80B4	DC Write Channel 9 Configuration 1Register (IPU_DC_WR_CH_CONF1_9)	32	R/W	0000_0000h	38.5.313/ 3506
265_80B8	DC Write Channel 9Configuration 2Register (IPU_DC_WR_CH_CONF2_9)	32	R/W	0000_0000h	38.5.314/ 3507
265_80BC	DC Routine Link Register 1 Channel 9 (IPU_DC_RL1_CH_9)	32	R/W	0000_0000h	38.5.315/ 3508
265_80C0	DC Routine Link Register 2 Channel 9 (IPU_DC_RL2_CH_9)	32	R/W	0000_0000h	38.5.316/ 3509
265_80C4	DC Routine Link Register 3Channel 9 (IPU_DC_RL3_CH_9)	32	R/W	0000_0000h	38.5.317/ 3510
265_80C8	DC Routine Link Register 4 Channel 9 (IPU_DC_RL4_CH_9)	32	R/W	0000_0000h	38.5.318/ 3510
265_80CC	DC Routine Link Register 5 Channel 9 (IPU_DC_RL5_CH_9)	32	R/W	0000_0000h	38.5.319/ 3511
265_80D0	DC Routine Link Register 6 Channel 9 (IPU_DC_RL6_CH_9)	32	R/W	0000_0000h	38.5.320/ 3512
265_80D4	DC General Register (IPU_DC_GEN)	32	R/W	0000_0060h	38.5.321/ 3512
265_80D8	DC Display Configuration 1 Register 0 (IPU_DC_DISP_CONF1_0)	32	R/W	0000_0042h	38.5.322/ 3514
265_80DC	DC Display Configuration 1 Register 1 (IPU_DC_DISP_CONF1_1)	32	R/W	0000_0042h	38.5.323/ 3515
265_80E0	DC Display Configuration 1 Register 2 (IPU_DC_DISP_CONF1_2)	32	R/W	0000_0042h	38.5.324/ 3517
265_80E4	DC Display Configuration 1 Register 3 (IPU_DC_DISP_CONF1_3)	32	R/W	0000_0042h	38.5.325/ 3518
265_80E8	DC Display Configuration 2 Register 0 (IPU_DC_DISP_CONF2_0)	32	R/W	0000_0000h	38.5.326/ 3519
265_80EC	DC Display Configuration 2 Register 1 (IPU_DC_DISP_CONF2_1)	32	R/W	0000_0000h	38.5.327/ 3520
265_80F0	DC Display Configuration 2 Register 2 (IPU_DC_DISP_CONF2_2)	32	R/W	0000_0000h	38.5.328/ 3520
265_80F4	DC Display Configuration 2 Register 3 (IPU_DC_DISP_CONF2_3)	32	R/W	0000_0000h	38.5.329/ 3520
265_80F8	DC DI0Configuration Register 1 (IPU_DC_DI0_CONF_1)	32	R/W	0000_0000h	38.5.330/ 3521

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
265_80FC	DC DI0Configuration Register 2 (IPU_DC_DI0_CONF_2)	32	R/W	0000_0000h	38.5.331/ 3521
265_8100	DC DI1Configuration Register 1 (IPU_DC_DI1_CONF_1)	32	R/W	0000_0000h	38.5.332/ 3521
265_8104	DC DI1Configuration Register 2 (IPU_DC_DI1_CONF_2)	32	R/W	0000_0000h	38.5.333/ 3522
265_8108	DC Mapping Configuration Register 0 (IPU_DC_MAP_CONF_0)	32	R/W	0000_0000h	38.5.334/ 3522
265_810C	DC Mapping Configuration Register 1 (IPU_DC_MAP_CONF_1)	32	R/W	0000_0000h	38.5.335/ 3523
265_8110	DC Mapping Configuration Register 2 (IPU_DC_MAP_CONF_2)	32	R/W	0000_0000h	38.5.336/ 3524
265_8114	DC Mapping Configuration Register 3 (IPU_DC_MAP_CONF_3)	32	R/W	0000_0000h	38.5.337/ 3525
265_8118	DC Mapping Configuration Register 4 (IPU_DC_MAP_CONF_4)	32	R/W	0000_0000h	38.5.338/ 3526
265_811C	DC Mapping Configuration Register 5 (IPU_DC_MAP_CONF_5)	32	R/W	0000_0000h	38.5.339/ 3527
265_8120	DC Mapping Configuration Register 6 (IPU_DC_MAP_CONF_6)	32	R/W	0000_0000h	38.5.340/ 3528
265_8124	DC Mapping Configuration Register 7 (IPU_DC_MAP_CONF_7)	32	R/W	0000_0000h	38.5.341/ 3529
265_8128	DC Mapping Configuration Register 8 (IPU_DC_MAP_CONF_8)	32	R/W	0000_0000h	38.5.342/ 3530
265_812C	DC Mapping Configuration Register 9 (IPU_DC_MAP_CONF_9)	32	R/W	0000_0000h	38.5.343/ 3531
265_8130	DC Mapping Configuration Register 10 (IPU_DC_MAP_CONF_10)	32	R/W	0000_0000h	38.5.344/ 3532
265_8134	DC Mapping Configuration Register 11 (IPU_DC_MAP_CONF_11)	32	R/W	0000_0000h	38.5.345/ 3533
265_8138	DC Mapping Configuration Register 12 (IPU_DC_MAP_CONF_12)	32	R/W	0000_0000h	38.5.346/ 3534
265_813C	DC Mapping Configuration Register 13 (IPU_DC_MAP_CONF_13)	32	R/W	0000_0000h	38.5.347/ 3535
265_8140	DC Mapping Configuration Register 14 (IPU_DC_MAP_CONF_14)	32	R/W	0000_0000h	38.5.348/ 3536
265_8144	DC Mapping Configuration Register 15 (IPU_DC_MAP_CONF_15)	32	R/W	0000_0000h	38.5.349/ 3537
265_8148	DC Mapping Configuration Register 16 (IPU_DC_MAP_CONF_16)	32	R/W	0000_0000h	38.5.350/ 3537
265_814C	DC Mapping Configuration Register 17 (IPU_DC_MAP_CONF_17)	32	R/W	0000_0000h	38.5.351/ 3538
265_8150	DC Mapping Configuration Register 18 (IPU_DC_MAP_CONF_18)	32	R/W	0000_0000h	38.5.352/ 3539

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
265_8154	DC Mapping Configuration Register 19 (IPU_DC_MAP_CONF_19)	32	R/W	0000_0000h	38.5.353/ 3539
265_8158	DC Mapping Configuration Register 20 (IPU_DC_MAP_CONF_20)	32	R/W	0000_0000h	38.5.354/ 3540
265_815C	DC Mapping Configuration Register 21 (IPU_DC_MAP_CONF_21)	32	R/W	0000_0000h	38.5.355/ 3541
265_8160	DC Mapping Configuration Register 22 (IPU_DC_MAP_CONF_22)	32	R/W	0000_0000h	38.5.356/ 3541
265_8164	DC Mapping Configuration Register 23 (IPU_DC_MAP_CONF_23)	32	R/W	0000_0000h	38.5.357/ 3542
265_8168	DC Mapping Configuration Register 24 (IPU_DC_MAP_CONF_24)	32	R/W	0000_0000h	38.5.358/ 3543
265_816C	DC Mapping Configuration Register 25 (IPU_DC_MAP_CONF_25)	32	R/W	0000_0000h	38.5.359/ 3543
265_8170	DC Mapping Configuration Register 26 (IPU_DC_MAP_CONF_26)	32	R/W	0000_0000h	38.5.360/ 3544
265_8174	DC User General Data Event 0 Register 0 (IPU_DC_UGDE0_0)	32	R/W	0000_0000h	38.5.361/ 3545
265_8178	DC User General Data Event 0 Register 1 (IPU_DC_UGDE0_1)	32	R/W	0000_0000h	38.5.362/ 3546
265_817C	DC User General Data Event 0 Register2 (IPU_DC_UGDE0_2)	32	R/W	0000_0000h	38.5.363/ 3547
265_8180	DC User General Data Event 0 Register 3 (IPU_DC_UGDE0_3)	32	R/W	0000_0000h	38.5.364/ 3547
265_8184	DC User General Data Event 1Register0 (IPU_DC_UGDE1_0)	32	R/W	0000_0000h	38.5.365/ 3548
265_8188	DC User General Data Event 1 Register 1 (IPU_DC_UGDE1_1)	32	R/W	0000_0000h	38.5.366/ 3549
265_818C	DC User General Data Event 1Register 2 (IPU_DC_UGDE1_2)	32	R/W	0000_0000h	38.5.367/ 3550
265_8190	DC User General Data Event 1Register 3 (IPU_DC_UGDE1_3)	32	R/W	0000_0000h	38.5.368/ 3550
265_8194	DC User General Data Event 2 Register 0 (IPU_DC_UGDE2_0)	32	R/W	0000_0000h	38.5.369/ 3551
265_8198	DC User General Data Event 2 Register 1 (IPU_DC_UGDE2_1)	32	R/W	0000_0000h	38.5.370/ 3552
265_819C	DC User General Data Event 2Register 2 (IPU_DC_UGDE2_2)	32	R/W	0000_0000h	38.5.371/ 3553
265_81A0	DC User General Data Event 2Register 3 (IPU_DC_UGDE2_3)	32	R/W	0000_0000h	38.5.372/ 3553
265_81A4	DC User General Data Event 3Register 0 (IPU_DC_UGDE3_0)	32	R/W	0000_0000h	38.5.373/ 3554
265_81A8	DC User General Data Event 3Register 1 (IPU_DC_UGDE3_1)	32	R/W	0000_0000h	38.5.374/ 3555

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
265_81AC	DC User General Data Event 3Register 2 (IPU_DC_UGDE3_2)	32	R/W	0000_0000h	38.5.375/ 3556
265_81B0	DC User General Data Event 3Register 2 (IPU_DC_UGDE3_3)	32	R/W	0000_0000h	38.5.376/ 3556
265_81B4	DC Low Level Access Control Register 0 (IPU_DC_LLA0)	32	R/W	0000_0000h	38.5.377/ 3556
265_81B8	DC Low Level Access Control Register 1 (IPU_DC_LLA1)	32	R/W	0000_0000h	38.5.378/ 3557
265_81BC	DC Read Low Level Read Access Control Register 0 (IPU_DC_R_LLA0)	32	R/W	0000_0000h	38.5.379/ 3557
265_81C0	DC Read Low Level Read Access Control Register1 (IPU_DC_R_LLA1)	32	R/W	0000_0000h	38.5.380/ 3558
265_81C4	DC Write Channel 5 Configuration Register (IPU_DC_WR_CH_ADDR_5_ALT)	32	R/W	0000_0000h	38.5.381/ 3558
265_81C8	DC Status Register (IPU_DC_STAT)	32	R	0000_00AAh	38.5.382/ 3560
266_0000	DMFC Read Channel Register (IPU_DMFC_RD_CHAN)	32	R/W	0000_0200h	38.5.383/ 3562
266_0004	DMFC Write Channel Register (IPU_DMFC_WR_CHAN)	32	R/W	0000_0000h	38.5.384/ 3564
266_0008	DMFC Write Channel Definition Register (IPU_DMFC_WR_CHAN_DEF)	32	R/W	2020_2020h	38.5.385/ 3567
266_000C	DMFC Display Processor Channel Register (IPU_DMFC_DP_CHAN)	32	R/W	0000_0000h	38.5.386/ 3569
266_0010	DMFC Display Processor Channel Definition Register (IPU_DMFC_DP_CHAN_DEF)	32	R/W	2020_2020h	38.5.387/ 3572
266_0014	DMFC General 1 Register (IPU_DMFC_GENERAL_1)	32	R/W	0000_0003h	38.5.388/ 3574
266_0018	DMFC General 2 Register (IPU_DMFC_GENERAL_2)	32	R/W	0000_0000h	38.5.389/ 3576
266_001C	DMFC IC Interface Control Register (IPU_DMFC_IC_CTRL)	32	R/W	0000_0002h	38.5.390/ 3577
266_0020	DMFC Write Channel Alternate Register (IPU_DMFC_WR_CHAN_ALT)	32	R/W	0000_0000h	38.5.391/ 3578
266_0024	DMFC Write Channel Definition Alternate Register (IPU_DMFC_WR_CHAN_DEF_ALT)	32	R/W	0000_2000h	38.5.392/ 3579
266_0028	DMFC MFC Display Processor Channel Alternate Register (IPU_DMFC_DP_CHAN_ALT)	32	R/W	0000_0000h	38.5.393/ 3580
266_002C	DMFC Display Channel Definition Alternate Register (IPU_DMFC_DP_CHAN_DEF_ALT)	32	R/W	2020_0020h	38.5.394/ 3583
266_0030	DMFC General 1 Alternate Register (IPU_DMFC_GENERAL1_ALT)	32	R/W	0000_0000h	38.5.395/ 3585
266_0034	DMFC Status Register (IPU_DMFC_STAT)	32	R	02FF_F000h	38.5.396/ 3587

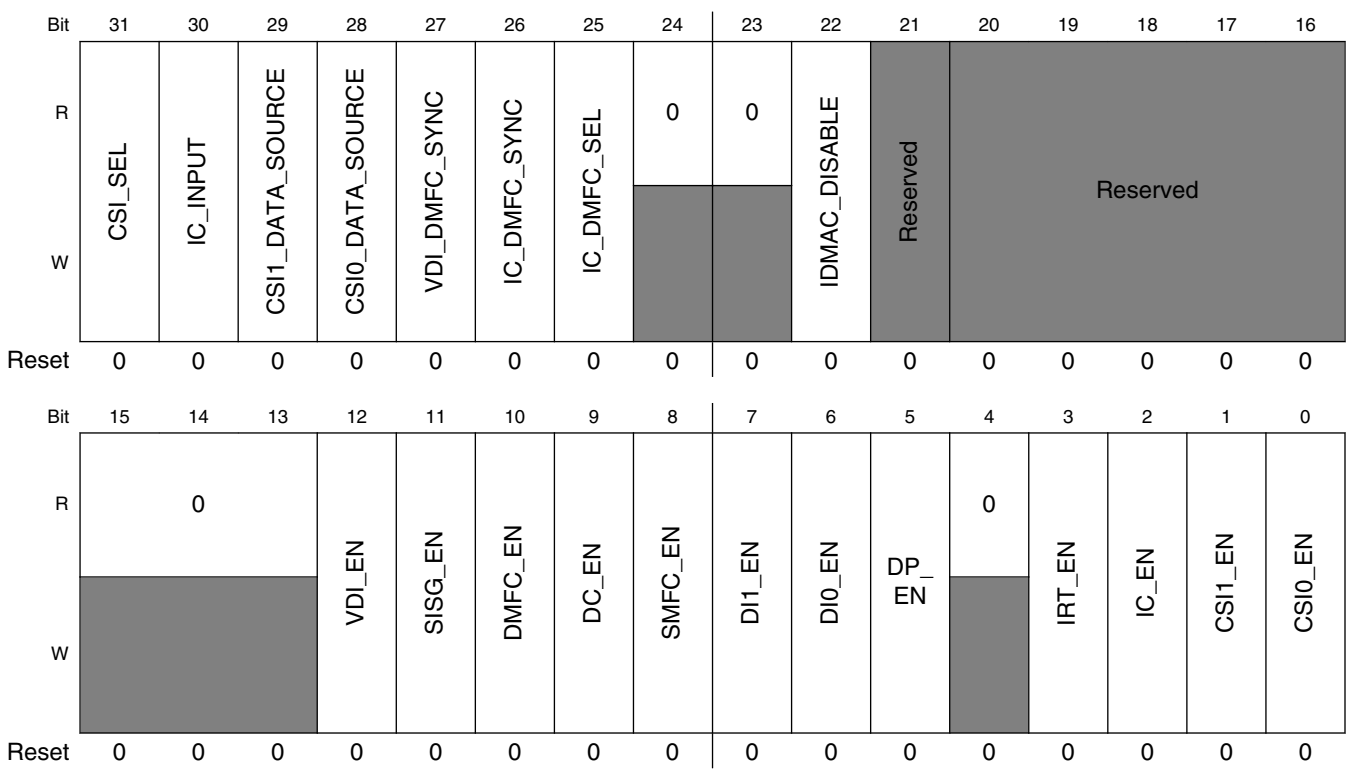
Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
266_8000	VDI Field Size Register (IPU_VDI_FSIZE)	32	R/W	0000_0000h	38.5.397/ 3588
266_8004	VDI Control Register (IPU_VDI_C)	32	R/W	0000_0000h	38.5.398/ 3589
266_8008	VDI Control Register 2 (IPU_VDI_C2_)	32	R/W	0000_0000h	38.5.399/ 3591
266_800C	VDI Combining Parameters Register 1 (IPU_VDI_CMDP_1)	32	R/W	0000_0000h	38.5.400/ 3592
266_8010	VDI Combining Parameters Register 2 (IPU_VDI_CMDP_2)	32	R/W	0000_0000h	38.5.401/ 3593
266_8014	VDI Plane Size Register 1 (IPU_VDI_PS_1)	32	R/W	0000_0000h	38.5.402/ 3593
266_8018	VDI Plane Size Register 2 (IPU_VDI_PS_2)	32	R/W	0000_0000h	38.5.403/ 3594
266_801C	VDI Plane Size Register 3 (IPU_VDI_PS_3)	32	R/W	0000_0000h	38.5.404/ 3595
266_8020	VDI Plane Size Register 4 (IPU_VDI_PS_4)	32	R/W	0000_0000h	38.5.405/ 3595

38.5.1 Configuration Register (IPU_CONF)

Address: 260_0000h base + 0h offset = 260_0000h



IPU_CONF field descriptions

Field	Description
31 CSI_SEL	CSI select bit; This bit selects manually between the 2 CSI's. This bit defines which CSI is the input to the IC. This bit is effective only if IC_INPUT is bit cleared 0 CSI0 is selected 1 CSI1 is selected
30 IC_INPUT	IC Input select bit. This bit selects manually between the 2 inputs to the IC 0 CSI0/1 is selected; In order to select between the CSIs use the CSI_SEL bit. 1 VDI
29 CSI1_DATA_SOURCE	CSI1 data Source This bit selects what is the data source for the CSI1. This is a static mux that should not be changed while CSI1 is working. Data is handles differently if the source is MCT (MIPI) or parallel interface. 0 Parallel interface is connected to CSI1 1 MCT (MIPI) is connected to CSI1
28 CSI0_DATA_SOURCE	CSI0 data Source This bit selects what is the data source for the CSI0. This is a static mux that should not be changed while CSI0 is working. Data is handles differently if the source is MCT (MIPI) or parallel interface. 0 Parallel interface is connected to CSI0 1 MCT (MIPI) is connected to CSI0
27 VDI_DMFC_SYNC	This bit enables the direct path VDIC -> IC_VF -> DMFC for sync flow. If this bit is set IC_DMFC_SEL must be set. 0 the flow is disabled 1 the flow is enabled
26 IC_DMFC_SYNC	IC to DMFC Sync flow This bit defines if the direct flow between IC to DMFC is synchronous or asynchronous 0 async flow 1 Sync flow
25 IC_DMFC_SEL	IC to DMFC select Selects the DMAIC_1 (channel 21) channel's connectivity between the IC and the DMFC 0 DMAIC_1 (channel 21) is routed to the IDMAC 1 DMAIC_1 (channel 21) is routed to DMFC In case DMFC was selected the IDMAC_CH_EN[21] must be clear.
24 Reserved	This read-only field is reserved and always has the value 0.
23 Reserved	This read-only field is reserved and always has the value 0.
22 IDMAC_DISABLE	Image DMA controller (IDMAC) disable bit. This bit allows the user to turn off the clock of the IDMAC if the use case permits it. By default the IDMAC is enabled. 0 IDMAC is enabled 1 IDMAC is disabled
21 -	This field is reserved. Reserved.

Table continues on the next page...

IPU_CONF field descriptions (continued)

Field	Description
20–16 -	This field is reserved. Reserved
15–13 Reserved	This read-only field is reserved and always has the value 0.
12 VDI_EN	VDI enable bit. This bit must be cleared if the ISP_EN bit is set. 0 VDIC is disabled 1 VDIC is enabled
11 SISG_EN	Still Image Synchronization Generator (SISG) Enable bit 0 SISG is disabled 1 SISG is enabled
10 DMFC_EN	Display's Multi FIFO Controller sub-block (DMFC) Enable bit 0 DMFC is disabled 1 DMFC is enabled
9 DC_EN	Display Controller sub-block (DC) Enable bit 0 DC is disabled 1 DC is enabled
8 SMFC_EN	Sensor's Multi FIFO Controller Sub-block (SMFC) Enable bit 0 SMFC is disabled 1 SMFC is enabled
7 DI1_EN	Display Interface Sub-block 1 Enable bit 0 DI1 is disabled 1 DI1 is enabled
6 DI0_EN	Display interface Sub-block 0 Enable bit 0 DI0 is disabled 1 DI0 is enabled
5 DP_EN	Display processor Sub-block Enable bit 0 DP is disabled 1 DP is enabled
4 Reserved	This read-only field is reserved and always has the value 0.
3 IRT_EN	Image Rotation Sub-Block Enable bit 0 IRT is disabled 1 IRT is enabled
2 IC_EN	Image Conversion Sub-Block Enable bit 0 IC is disabled 1 IC is enabled
1 CSI1_EN	Camera Sensor Interface 1 Enable bit

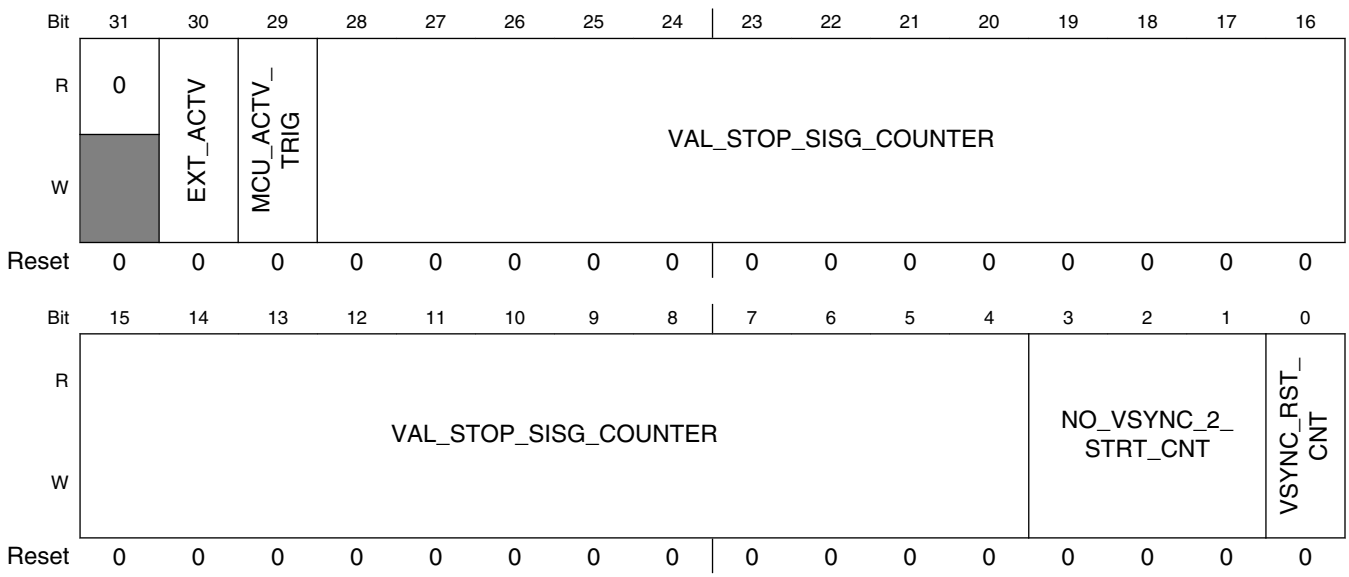
Table continues on the next page...

IPU_CONF field descriptions (continued)

Field	Description
	0 CSI1 is disabled 1 CSI1 is enabled
0 CSI0_EN	Camera Sensor Interface 0 Enable bit 0 CSI0 is disabled 1 CSI0 is enabled

38.5.2 SISG Control 0 Register (IPU_SISG_CTRL0)

Address: 260_0000h base + 4h offset = 260_0004h



IPU_SISG_CTRL0 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30 EXT_ACTV	External Active Define if an external active trigger will start the counters. The external active trigger is an input signal to the IPU called ext_actv_trig
29 MCU_ACTV_TRIG	Reserved, should be cleared.
28–4 VAL_STOP_SISG_COUNTER	SISG Stop Counters value. This is a predefined value that stops the SISG counters. The user should write to this field the N-1 value of the desired value.

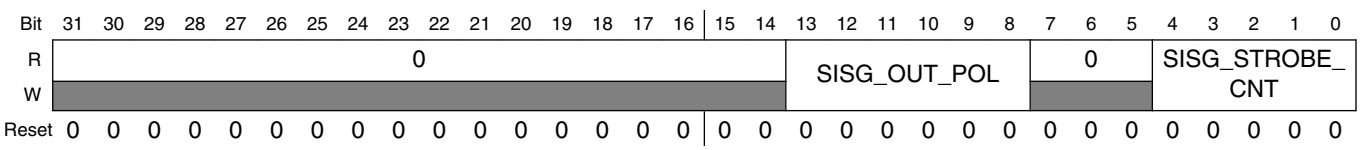
Table continues on the next page...

IPU_SISG_CTRL0 field descriptions (continued)

Field	Description
3–1 NO_VSYNC_2_STRT_CNT	VSYCs to Start Counter This bits define how many VSYNCs signals will be counter before activating the SISG counters. If set to 0 starts immediately. If set to N (1..7) starts after N VSYNCs.
0 VSYNC_RST_CNT	VSYNC Resets counters Defines if the counters are stooped following VSYNC or when the counters reach a pre defined value (VAL_STOP_SISG_COUNTER) 1 The counters are stopped at VSYNC 0 The counters are stooped when the counters reach the VAL_STOP_SISG_COUNTER value.

38.5.3 SISG Control 1 Register (IPU_SISG_CTRL1)

Address: 260_0000h base + 8h offset = 260_0008h

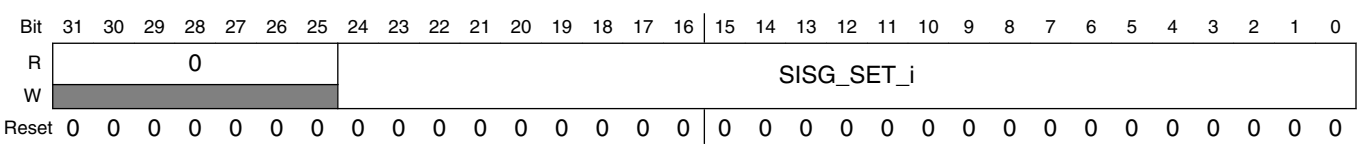


IPU_SISG_CTRL1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
13–8 SISG_OUT_POL	SISG_OUT_POL This bits defines the polarity of the SISG output signals 1 active high 0 active low
7–5 Reserved	This read-only field is reserved and always has the value 0.
SISG_STROBE_CNT	SISG Strobe Count The SISG can repeat the sequence for up to 32 cycles; this is used for generating a train of pulses.

38.5.4 SISG Set<i> Register (IPU_SISG_SET_i)

Address: 260_0000h base + Ch offset = 260_000Ch



IPU_SISG_SET_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
SISG_SET_i	SISG SET <i> value These bits define the set value of the SISG counter #<i>

38.5.5 SISG Clear <i> Register (IPU_SISG_CLR_i)

Address: 260_0000h base + 24h offset = 260_0024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							SISG_CLEAR_i																								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_SISG_CLR_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
SISG_CLEAR_i	SISG CLR <i> value These bits define the clear value of the SISG counter #<i>

38.5.6 Interrupt Control Register 1 (IPU_INT_CTRL_1)

This register contains part of IPU interrupts controls. The controls of EOF (end of frame) of DMA Channels interrupts [31:0] can be found in this register.

Address: 260_0000h base + 3Ch offset = 260_003Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_EOF_EN_31	0	IDMAC_EOF_EN_29	IDMAC_EOF_EN_28	IDMAC_EOF_EN_27	IDMAC_EOF_EN_26	IDMAC_EOF_EN_25	IDMAC_EOF_EN_24	IDMAC_EOF_EN_23	IDMAC_EOF_EN_22	IDMAC_EOF_EN_21	IDMAC_EOF_EN_20	IDMAC_EOF_EN_19	IDMAC_EOF_EN_18	IDMAC_EOF_EN_17	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_EOF_EN_15	IDMAC_EOF_EN_14	IDMAC_EOF_EN_13	IDMAC_EOF_EN_12	IDMAC_EOF_EN_11	IDMAC_EOF_EN_10	IDMAC_EOF_EN_9	IDMAC_EOF_EN_8	0		IDMAC_EOF_EN_5	0	IDMAC_EOF_EN_3	IDMAC_EOF_EN_2	IDMAC_EOF_EN_1	IDMAC_EOF_EN_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_CTRL_1 field descriptions

Field	Description
31 IDMAC_EOF_EN_31	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
30 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
29 IDMAC_EOF_EN_29	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
28 IDMAC_EOF_EN_28	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_1 field descriptions (continued)

Field	Description
27 IDMAC_EOF_EN_27	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
26 IDMAC_EOF_EN_26	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IDMAC_EOF_EN_25	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
24 IDMAC_EOF_EN_24	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
23 IDMAC_EOF_EN_23	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
22 IDMAC_EOF_EN_22	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
21 IDMAC_EOF_EN_21	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_EOF_EN_20	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_EOF_EN_19	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_1 field descriptions (continued)

Field	Description
18 IDMAC_EOF_EN_18	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_EOF_EN_17	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_EOF_EN_15	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_EOF_EN_14	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
13 IDMAC_EOF_EN_13	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_EOF_EN_12	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_EOF_EN_11	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_EOF_EN_10	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_EOF_EN_9	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

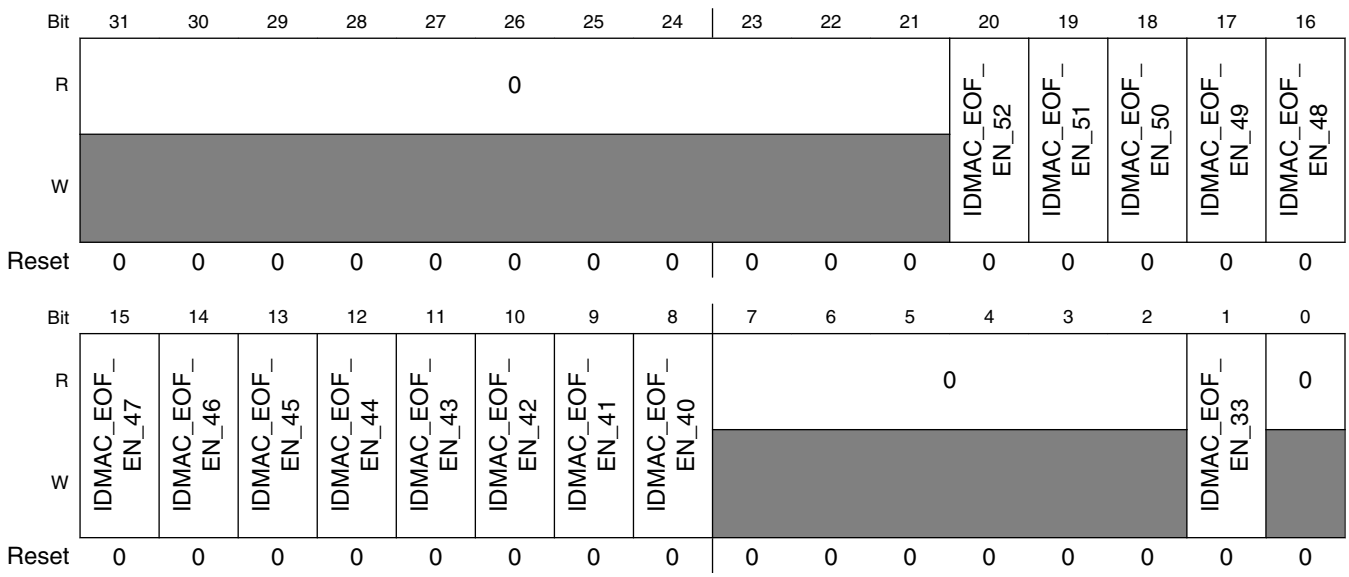
IPU_INT_CTRL_1 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
8 IDMAC_EOF_EN_8	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
7–6 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
5 IDMAC_EOF_EN_5	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
4 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
3 IDMAC_EOF_EN_3	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 IDMAC_EOF_EN_2	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_EOF_EN_1	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 IDMAC_EOF_EN_0	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

38.5.7 Interrupt Control Register 2 (IPU_INT_CTRL_2)

This register contains part of IPU interrupts controls. The controls of EOF (end of frame) of DMA Channels interrupts [63:32] can be found in this register.

Address: 260_0000h base + 40h offset = 260_0040h



IPU_INT_CTRL_2 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_EOF_EN_52	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_EOF_EN_51	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_EOF_EN_50	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_2 field descriptions (continued)

Field	Description
17 IDMAC_EOF_EN_49	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 IDMAC_EOF_EN_48	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_EOF_EN_47	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_EOF_EN_46	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
13 IDMAC_EOF_EN_45	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_EOF_EN_44	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_EOF_EN_43	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_EOF_EN_42	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_EOF_EN_41	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_2 field descriptions (continued)

Field	Description
8 IDMAC_EOF_EN_40	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
7–2 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_EOF_EN_33	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.

38.5.8 Interrupt Control Register 3 (IPU_INT_CTRL_3)

This register contains part of IPU interrupts controls. The controls of NFACK (New Frame Ack) of DMA Channels interrupts [31:0] can be found in this register.

Address: 260_0000h base + 44h offset = 260_0044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_NFACK_EN_31	0	IDMAC_NFACK_EN_29	IDMAC_NFACK_EN_28	IDMAC_NFACK_EN_27	IDMAC_NFACK_EN_26	IDMAC_NFACK_EN_25	IDMAC_NFACK_EN_24	IDMAC_NFACK_EN_23	IDMAC_NFACK_EN_22	IDMAC_NFACK_EN_21	IDMAC_NFACK_EN_20	IDMAC_NFACK_EN_19	IDMAC_NFACK_EN_18	IDMAC_NFACK_EN_17	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_NFACK_EN_15	IDMAC_NFACK_EN_14	IDMAC_NFACK_EN_13	IDMAC_NFACK_EN_12	IDMAC_NFACK_EN_11	IDMAC_NFACK_EN_10	IDMAC_NFACK_EN_9	IDMAC_NFACK_EN_8	0		IDMAC_NFACK_EN_5	0	IDMAC_NFACK_EN_3	IDMAC_NFACK_EN_2	IDMAC_NFACK_EN_1	IDMAC_NFACK_EN_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_CTRL_3 field descriptions

Field	Description
31 IDMAC_NFACK_EN_31	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
30 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
29 IDMAC_NFACK_EN_29	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
28 IDMAC_NFACK_EN_28	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
27 IDMAC_NFACK_EN_27	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
26 IDMAC_NFACK_EN_26	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IDMAC_NFACK_EN_25	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
24 IDMAC_NFACK_EN_24	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
23 IDMAC_NFACK_EN_23	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
22 IDMAC_NFACK_EN_22	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_CTRL_3 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
21 IDMAC_NFACK_EN_21	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_NFACK_EN_20	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_NFACK_EN_19	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_NFACK_EN_18	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_NFACK_EN_17	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_NFACK_EN_15	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_NFACK_EN_14	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
13 IDMAC_NFACK_EN_13	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_3 field descriptions (continued)

Field	Description
12 IDMAC_NFACK_EN_12	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_NFACK_EN_11	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_NFACK_EN_10	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_NFACK_EN_9	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
8 IDMAC_NFACK_EN_8	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
7–6 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
5 IDMAC_NFACK_EN_5	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
4 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
3 IDMAC_NFACK_EN_3	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 IDMAC_NFACK_EN_2	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_CTRL_3 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_NFACK_EN_1	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 IDMAC_NFACK_EN_0	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

38.5.9 Interrupt Control Register 4 (IPU_INT_CTRL_4)

This register contains part of IPU interrupts controls. The controls of NFACK (New Frame Ack) of DMA Channels interrupts [63:32] can be found in this register.

Address: 260_0000h base + 48h offset = 260_0048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											IDMAC_NFACK_EN_52	IDMAC_NFACK_EN_51	IDMAC_NFACK_EN_50	IDMAC_NFACK_EN_49	IDMAC_NFACK_EN_48
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_NFACK_EN_47	IDMAC_NFACK_EN_46	IDMAC_NFACK_EN_45	IDMAC_NFACK_EN_44	IDMAC_NFACK_EN_43	IDMAC_NFACK_EN_42	IDMAC_NFACK_EN_41	IDMAC_NFACK_EN_40	0						IDMAC_NFACK_EN_33	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_CTRL_4 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_INT_CTRL_4 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_NFACK_EN_52	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_NFACK_EN_51	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_NFACK_EN_50	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_NFACK_EN_49	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 IDMAC_NFACK_EN_48	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_NFACK_EN_47	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_NFACK_EN_46	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
13 IDMAC_NFACK_EN_45	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_NFACK_EN_44	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_CTRL_4 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_NFACK_EN_43	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_NFACK_EN_42	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_NFACK_EN_41	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
8 IDMAC_NFACK_EN_40	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
7–2 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_NFACK_EN_33	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.

38.5.10 Interrupt Control Register 5 (IPU_INT_CTRL_5)

This register contains part of IPU interrupts controls. The controls of the New-frame before end-of-frame indication (NFB4EOF) of DMA Channels interrupts [31:0] can be found in this register.

Address: 260_0000h base + 4Ch offset = 260_004Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_NFB4EOF_EN_31	0	IDMAC_NFB4EOF_EN_29	IDMAC_NFB4EOF_EN_28	IDMAC_NFB4EOF_EN_27	IDMAC_NFB4EOF_EN_26	IDMAC_NFB4EOF_EN_25	IDMAC_NFB4EOF_EN_24	IDMAC_NFB4EOF_EN_23	IDMAC_NFB4EOF_EN_22	IDMAC_NFB4EOF_EN_21	IDMAC_NFB4EOF_EN_20	IDMAC_NFB4EOF_EN_19	IDMAC_NFB4EOF_EN_18	IDMAC_NFB4EOF_EN_17	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_NFB4EOF_EN_15	IDMAC_NFB4EOF_EN_14	IDMAC_NFB4EOF_EN_13	IDMAC_NFB4EOF_EN_12	IDMAC_NFB4EOF_EN_11	IDMAC_NFB4EOF_EN_10	IDMAC_NFB4EOF_EN_9	IDMAC_NFB4EOF_EN_8	0		IDMAC_NFB4EOF_EN_5	0	IDMAC_NFB4EOF_EN_3	IDMAC_NFB4EOF_EN_2	IDMAC_NFB4EOF_EN_1	IDMAC_NFB4EOF_EN_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_CTRL_5 field descriptions

Field	Description
31 IDMAC_NFB4EOF_EN_31	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
30 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
29 IDMAC_NFB4EOF_EN_29	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_5 field descriptions (continued)

Field	Description
28 IDMAC_ NFB4EOF_EN_ 28	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
27 IDMAC_ NFB4EOF_EN_ 27	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
26 IDMAC_ NFB4EOF_EN_ 26	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IDMAC_ NFB4EOF_EN_ 25	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
24 IDMAC_ NFB4EOF_EN_ 24	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
23 IDMAC_ NFB4EOF_EN_ 23	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
22 IDMAC_ NFB4EOF_EN_ 22	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
21 IDMAC_ NFB4EOF_EN_ 21	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_5 field descriptions (continued)

Field	Description
20 IDMAC_ NFB4EOF_EN_ 20	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_ NFB4EOF_EN_ 19	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_ NFB4EOF_EN_ 18	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_ NFB4EOF_EN_ 17	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_ NFB4EOF_EN_ 15	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_ NFB4EOF_EN_ 14	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
13 IDMAC_ NFB4EOF_EN_ 13	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_5 field descriptions (continued)

Field	Description
12 IDMAC_ NFB4EOF_EN_ 12	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_ NFB4EOF_EN_ 11	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_ NFB4EOF_EN_ 10	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_ NFB4EOF_EN_9	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
8 IDMAC_ NFB4EOF_EN_8	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
7–6 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
5 IDMAC_ NFB4EOF_EN_5	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
4 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
3 IDMAC_ NFB4EOF_EN_3	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

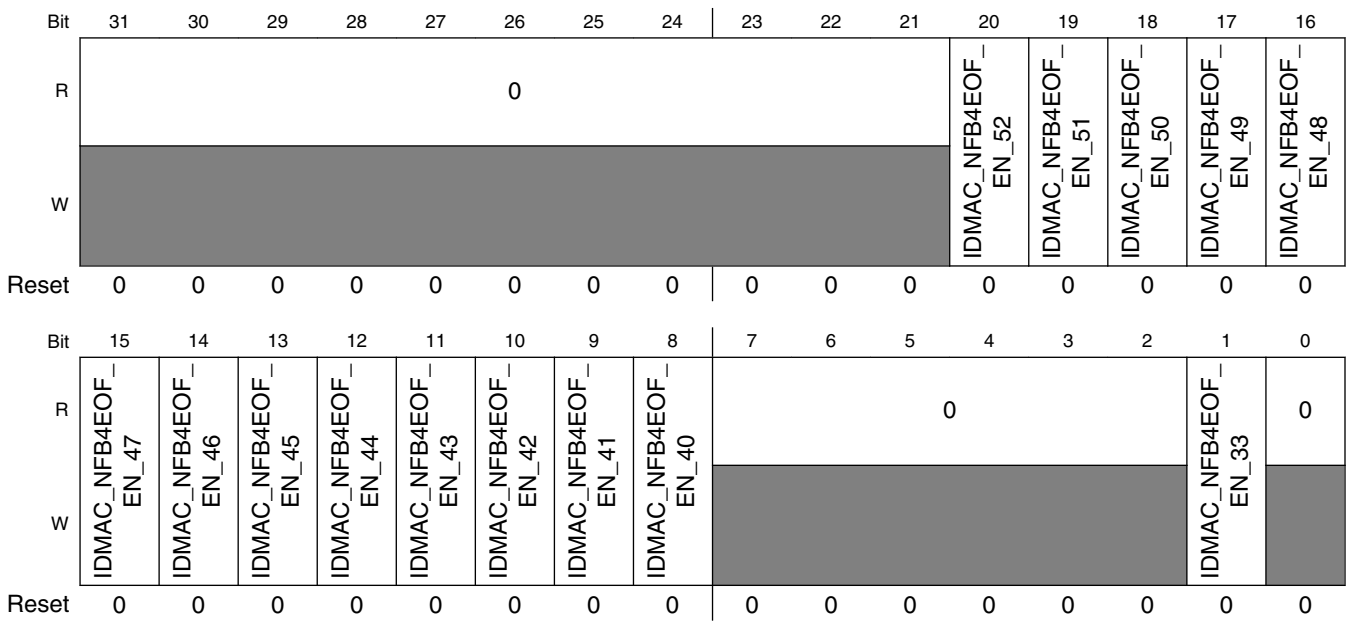
IPU_INT_CTRL_5 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
2 IDMAC_ NFB4EOF_EN_2	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_ NFB4EOF_EN_1	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 IDMAC_ NFB4EOF_EN_0	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

38.5.11 Interrupt Control Register 6 (IPU_INT_CTRL_6)

This register contains part of IPU interrupts controls. The controls of the New-frame before end-of-frame indication (NFB4EOF_ERR) of DMA Channels interrupts [63:32] can be found in this register.

Address: 260_0000h base + 50h offset = 260_0050h



IPU_INT_CTRL_6 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0.
20 IDMAC_ NFB4EOF_EN_ 52	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_ NFB4EOF_EN_ 51	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_6 field descriptions (continued)

Field	Description
18 IDMAC_ NFB4EOF_EN_ 50	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_ NFB4EOF_EN_ 49	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 IDMAC_ NFB4EOF_EN_ 48	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_ NFB4EOF_EN_ 47	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_ NFB4EOF_EN_ 46	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
13 IDMAC_ NFB4EOF_EN_ 45	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_ NFB4EOF_EN_ 44	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_ NFB4EOF_EN_ 43	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

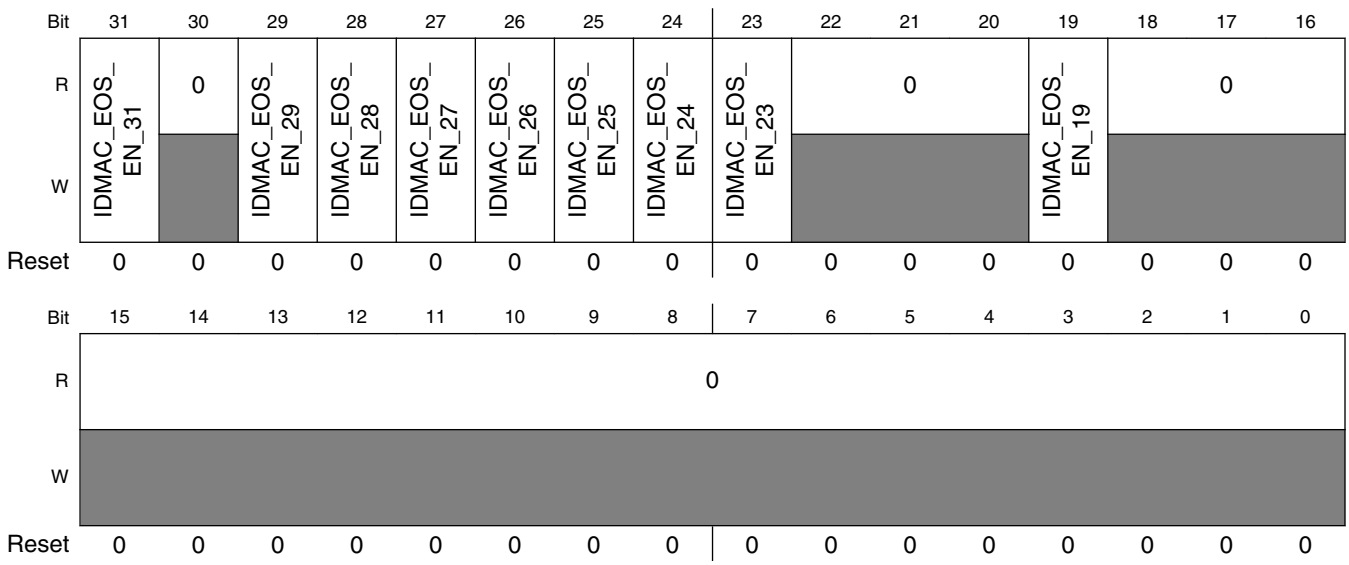
IPU_INT_CTRL_6 field descriptions (continued)

Field	Description
10 IDMAC_ NFB4EOF_EN_ 42	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_ NFB4EOF_EN_ 41	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
8 IDMAC_ NFB4EOF_EN_ 40	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
7–2 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_ NFB4EOF_EN_ 33	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.

38.5.12 Interrupt Control Register 7 (IPU_INT_CTRL_7)

This register contains part of IPUIPU interrupts controls. The controls of the End-of-Scroll indication (EOS) of DMA Channels interrupts [31:0] can be found in this register.

Address: 260_0000h base + 54h offset = 260_0054h



IPU_INT_CTRL_7 field descriptions

Field	Description
31 IDMAC_EOS_EN_31	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
30 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
29 IDMAC_EOS_EN_29	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
28 IDMAC_EOS_EN_28	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

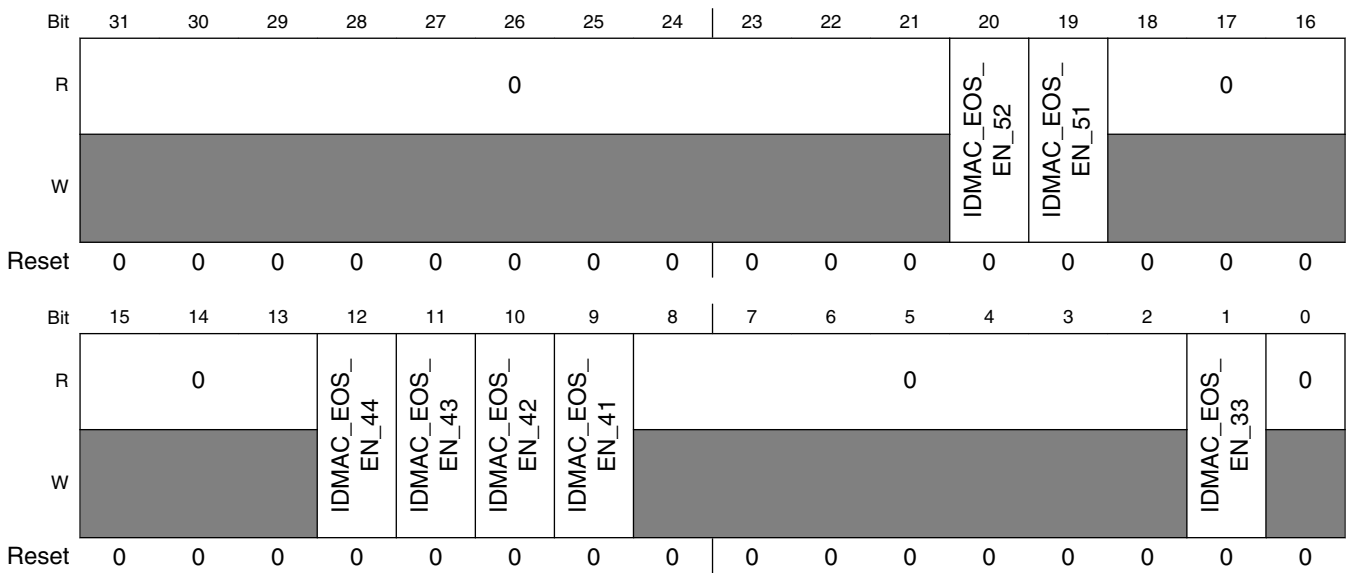
IPU_INT_CTRL_7 field descriptions (continued)

Field	Description
27 IDMAC_EOS_EN_27	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
26 IDMAC_EOS_EN_26	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IDMAC_EOS_EN_25	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
24 IDMAC_EOS_EN_24	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
23 IDMAC_EOS_EN_23	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
22–20 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_EOS_EN_19	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.

38.5.13 Interrupt Control Register 8 (IPU_INT_CTRL_8)

This register contains part of IPU interrupts controls. The controls of the End of Scroll indication (EOS) of DMA Channels interrupts [63:32] can be found in this register.

Address: 260_0000h base + 58h offset = 260_0058h



IPU_INT_CTRL_8 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_EOS_EN_52	End of Scroll of Channel interrupt. This bit is the control of End Of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_EOS_EN_51	End of Scroll of Channel interrupt. This bit is the control of End Of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18–13 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

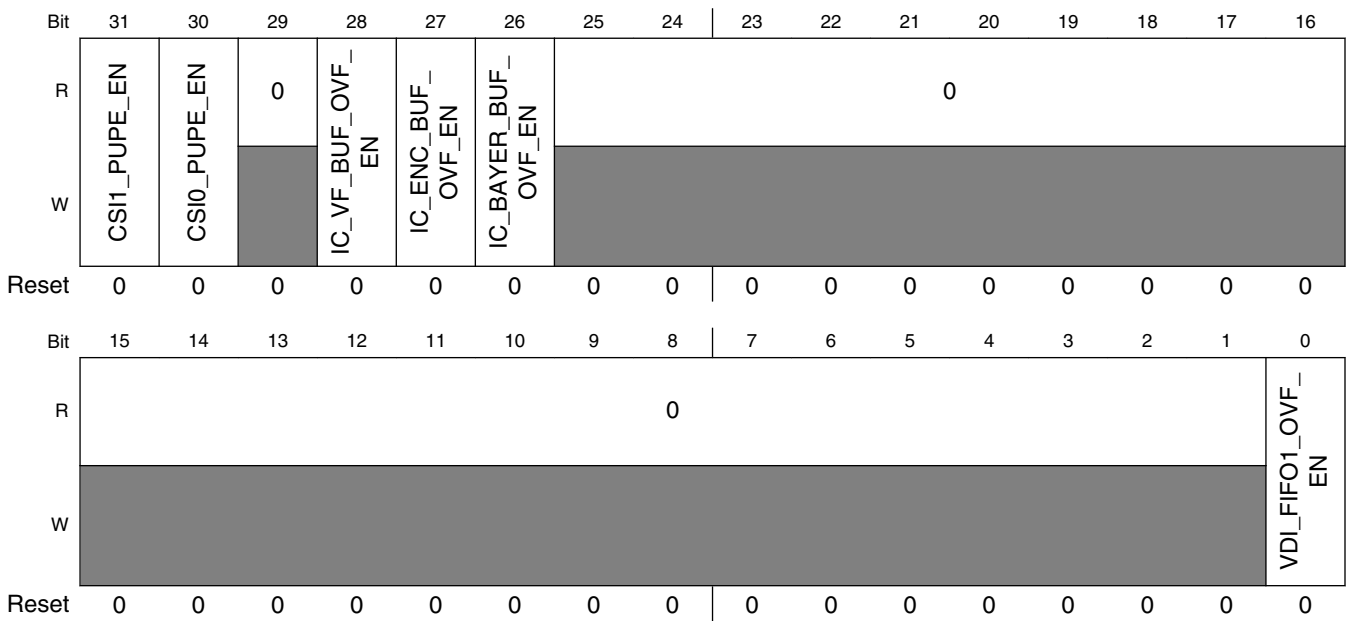
IPU_INT_CTRL_8 field descriptions (continued)

Field	Description
12 IDMAC_EOS_EN_44	End of Scroll of Channel interrupt. This bit is the control of End Of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_EOS_EN_43	End of Scroll of Channel interrupt. This bit is the control of End Of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_EOS_EN_42	End of Scroll of Channel interrupt. This bit is the control of End Of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_EOS_EN_41	End of Scroll of Channel interrupt. This bit is the control of End Of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
8–2 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_EOS_EN_33	End of Scroll of Channel interrupt. This bit is the control of End Of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.

38.5.14 Interrupt Control Register 9 (IPU_INT_CTRL_9)

This register contains part of IPU interrupts controls. This register controls error interrupt signals coming from different sub-blocks within

Address: 260_0000h base + 5Ch offset = 260_005Ch



IPU_INT_CTRL_9 field descriptions

Field	Description
31 CSI1_PUPE_EN	CSI1_PUPE_EN - CSI1 parameters update error interrupt enable. This bit enables an interrupt that is a result of an error generated by the CSI1. The error is generated in case where new frame arrived from the CSI1 before the completion of the CSI1's parameters update by the SRM 0 Interrupt is disabled. 1 Interrupt is enabled.
30 CSI0_PUPE_EN	CSI0_PUPE_EN - CSI0 parameters update error interrupt enable. This bit enables an interrupt that is a result of an error generated by the CSI0. The error is generated in case where new frame arrived from the CSI0 before the completion of the CSI0's parameters update by the SRM 0 Interrupt is disabled. 1 Interrupt is enabled.
29 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

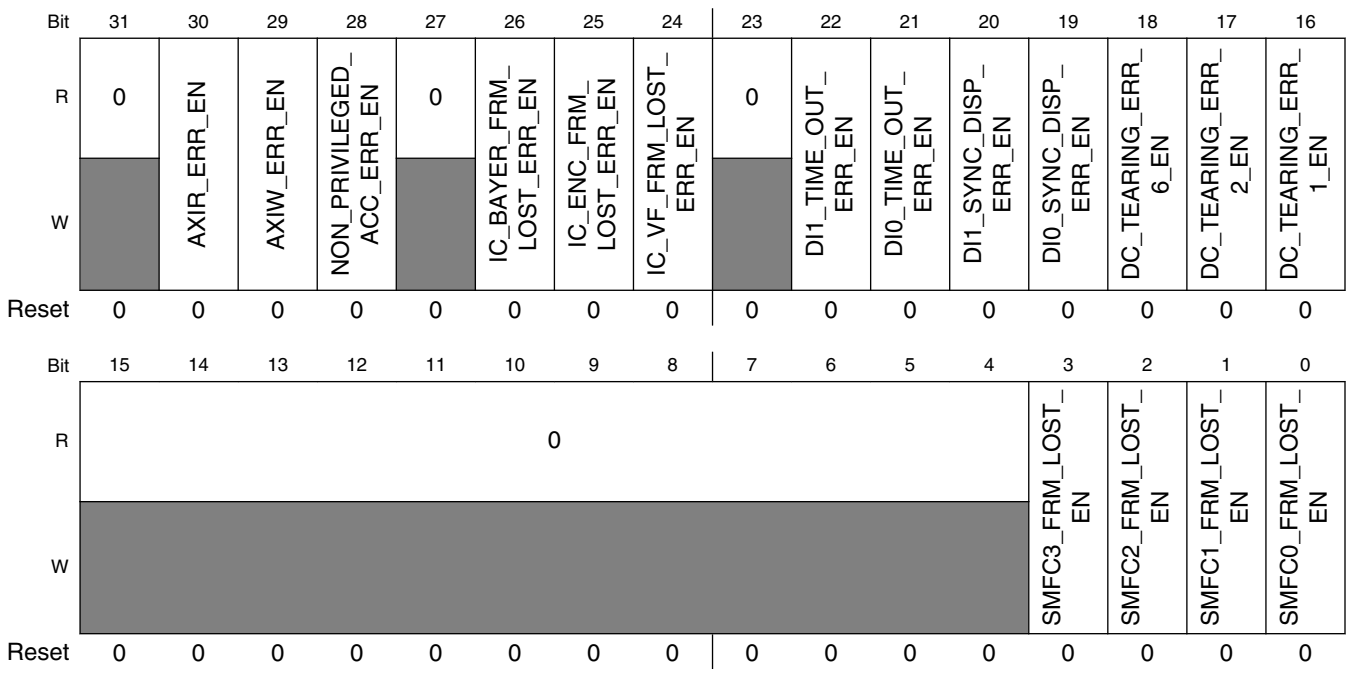
IPU_INT_CTRL_9 field descriptions (continued)

Field	Description
28 IC_VF_BUF_ OVF_EN	This bit enables an interrupt that is a result of the IC Buffer overflow for view finder coming from the IC. The user needs to write 1 to this bit in order to clear it. 0 Interrupt is disabled. 1 Interrupt is enabled.
27 IC_ENC_BUF_ OVF_EN	This bit enables an interrupt that is a result of the IC Buffer overflow for encoding coming from the IC. The user needs to write 1 to this bit in order to clear it. 0 Interrupt is disabled. 1 Interrupt is enabled.
26 IC_BAYER_ BUF_OVF_EN	This bit enables an interrupt that is a result of the IC Buffer overflow for bayer coming from the IC. The user needs to write 1 to this bit in order to clear it. 0 Interrupt is disabled. 1 Interrupt is enabled.
25–1 Reserved	This read-only field is reserved and always has the value 0.
0 VDI_FIFO1_ OVF_EN	FIFO1 overflow Interrupt1 Enable The VDIC generates FIFO1 overflow interrupt1 when the write pointer of FIFO1 overruns read pointer. 0 Interrupt is disabled. 1 Interrupt is enabled.

38.5.15 Interrupt Control Register 10 (IPU_INT_CTRL_10)

This register contains part of IPU interrupts controls. This register controls error interrupt signals coming from different modules within

Address: 260_0000h base + 60h offset = 260_0060h



IPU_INT_CTRL_10 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30 AXIR_ERR_EN	This bit enables an interrupt that is a result of AXI read access resulted with error response. 0 Interrupt is disabled. 1 Interrupt is enabled.
29 AXIW_ERR_EN	This bit enables an interrupt that is a result of AXI write access resulted with error response. 0 Interrupt is disabled. 1 Interrupt is enabled.
28 NON_ PRIVILEGED_ ACC_ERR_EN	Non Privileged Access Error interrupt enable. The CPMEM and the DP can be accessed by the ARM platform in privileged mode only HPROT[1] =1. An attempt to access these regions in user mode will issue an interrupt. This bit enables the interrupt. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_10 field descriptions (continued)

Field	Description
27 Reserved	This read-only field is reserved and always has the value 0.
26 IC_BAYER_ FRM_LOST_ ERR_EN	This bit enables an interrupt that is a result of IC's Bayer frame lost. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IC_ENC_FRM_ LOST_ERR_EN	This bit enables an interrupt that is a result of IC's encoding frame lost. 0 Interrupt is disabled. 1 Interrupt is enabled.
24 IC_VF_FRM_ LOST_ERR_EN	This bit enables an interrupt that is a result of IC's view finder frame lost. 0 Interrupt is disabled. 1 Interrupt is enabled.
23 Reserved	This read-only field is reserved and always has the value 0.
22 DI1_TIME_OUT_ ERR_EN	DI1 time out error interrupt enable This bit enables the interrupt that is a result of a time out error during a read access via DI1 0 Interrupt is disabled. 1 Interrupt is enabled.
21 DI0_TIME_OUT_ ERR_EN	DI0 time outwore interrupt enable This bit enables the interrupt that is a result of a time out error during a read access via DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
20 DI1_SYNC_ DISP_ERR_EN	DI1 Synchronous display error enable This bit enables the interrupt that is a result of an error during access to a synchronous display via DI1 0 Interrupt is disabled. 1 Interrupt is enabled.
19 DI0_SYNC_ DISP_ERR_EN	DI0 Synchronous display error enable This bit enables the interrupt that is a result of an error during access to a synchronous display via DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
18 DC_TEARING_ ERR_6_EN	Tearing Error #6 enable This bit enables the interrupt that is a result of tearing error while the anti tearing mechanism is activated for DC channel 6 0 Interrupt is disabled. 1 Interrupt is enabled.
17 DC_TEARING_ ERR_2_EN	Tearing Error #2 enable This bit enables the interrupt that is a result of tearing error while the anti tearing mechanism is activated for DC channel 2

Table continues on the next page...

IPU_INT_CTRL_10 field descriptions (continued)

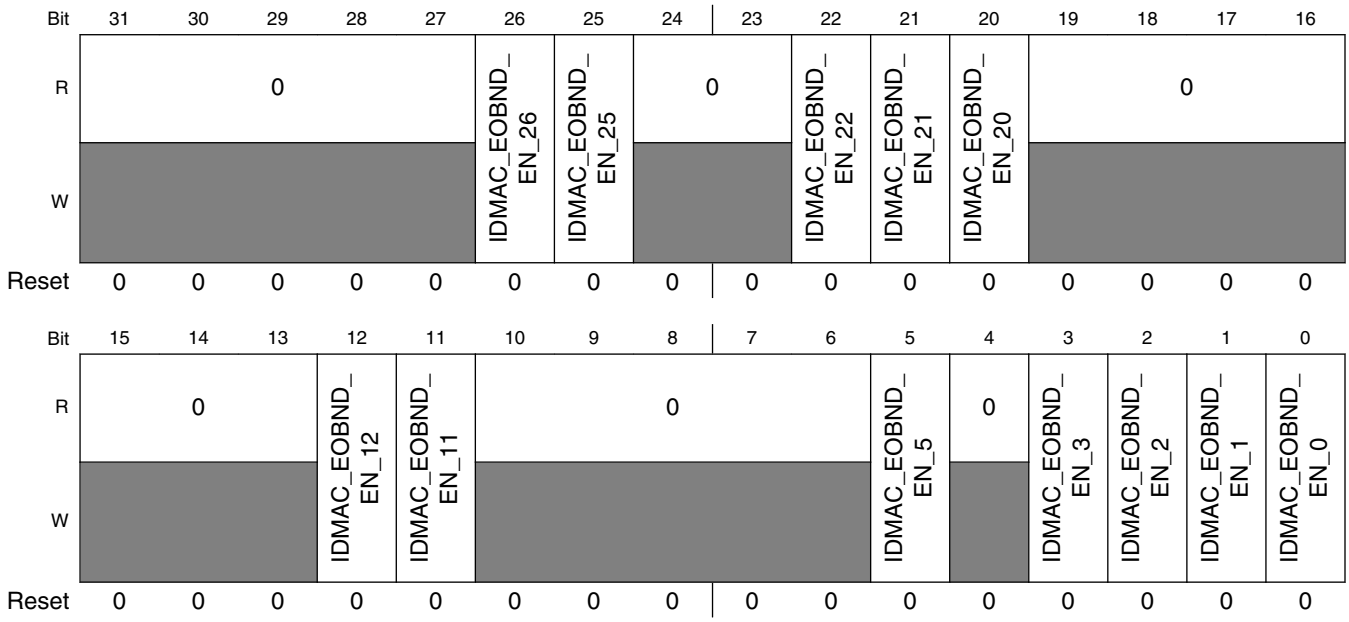
Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
16 DC_TEARING_ERR_1_EN	Tearing Error #1 enable This bit enables the interrupt that is a result of tearing error while the anti tearing mechanism is activated for DC channel 1 0 Interrupt is disabled. 1 Interrupt is enabled.
15–4 Reserved	This read-only field is reserved and always has the value 0.
3 SMFC3_FRM_LOST_EN	Frame Lost of SMFC channel 3 interrupt enable bit This bit enables an interrupt that is a result of a Frame Lost of SMFC channel 3. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 SMFC2_FRM_LOST_EN	Frame Lost of SMFC channel 2 interrupt enable bit This bit enables an interrupt that is a result of a Frame Lost of SMFC channel 2. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 SMFC1_FRM_LOST_EN	Frame Lost of SMFC channel 1 interrupt enable bit This bit enables an interrupt that is a result of a Frame Lost of SMFC channel 1. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 SMFC0_FRM_LOST_EN	Frame Lost of SMFC channel 0 interrupt enable bit This bit enables an interrupt that is a result of a Frame Lost of SMFC channel 0. 0 Interrupt is disabled. 1 Interrupt is enabled.

38.5.16 Interrupt Control Register 11 (IPU_INT_CTRL_11)

This register contains part of IPU interrupts controls. The controls of the end-of-band indication (EOBND) of DMA Channels interrupts [31:0] can be found in this register.

- Hide VDOA_SYNC for all versions
- Show VDOA_SYNC for IPUv3H version.
- The table below tagged with other settings (like IPU3M_only) should be hidden in IPUv3H version.

Address: 260_0000h base + 64h offset = 260_0064h



IPU_INT_CTRL_11 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
26 IDMAC_EOBND_EN_26	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IDMAC_EOBND_EN_25	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
24–23 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
22 IDMAC_EOBND_EN_22	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
21 IDMAC_EOBND_EN_21	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_CTRL_11 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_EOBND_EN_20	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19–13 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_EOBND_EN_12	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_EOBND_EN_11	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10–6 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
5 IDMAC_EOBND_EN_5	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
4 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
3 IDMAC_EOBND_EN_3	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 IDMAC_EOBND_EN_2	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_EOBND_EN_1	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

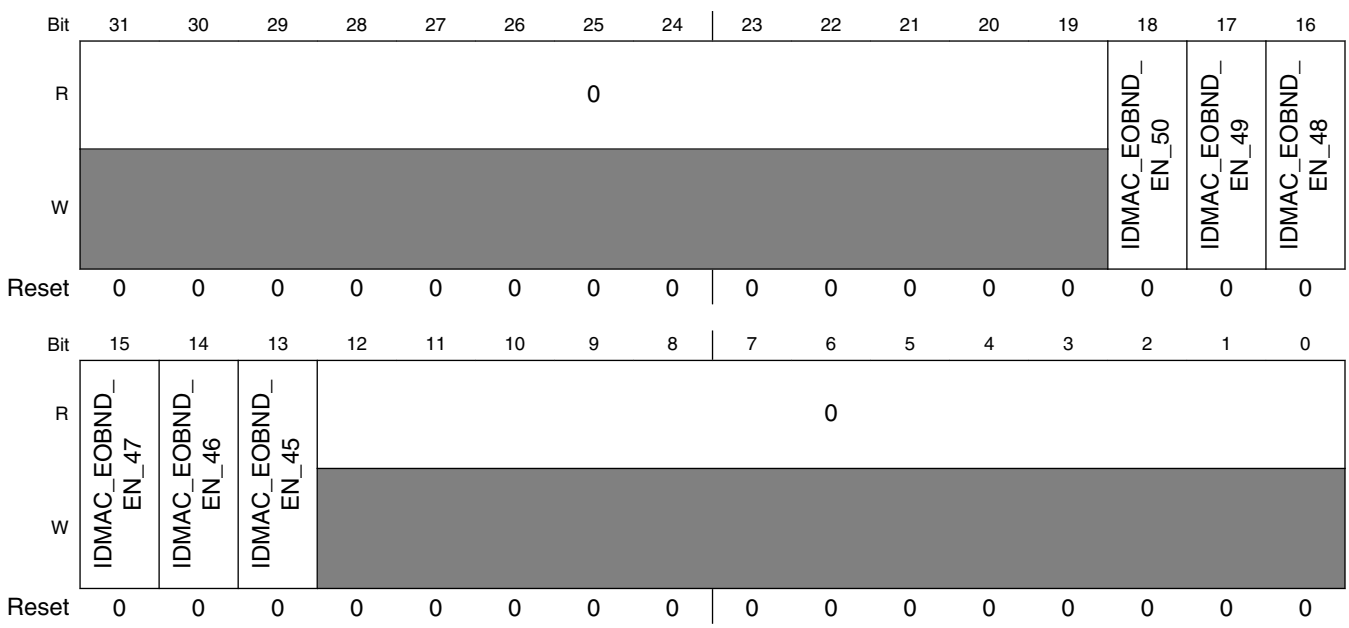
IPU_INT_CTRL_11 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
0 IDMAC_EOBND_EN_0	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

38.5.17 Interrupt Control Register 12 (IPU_INT_CTRL_12)

This register contains part of IPU interrupts controls. The controls of the end-of-band indication (EOBND) of DMA Channels interrupts [63:32] can be found in this register.

Address: 260_0000h base + 68h offset = 260_0068h



IPU_INT_CTRL_12 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_EOBND_EN_50	end-of-band indication of Channel interrupt. This bit is the control end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_CTRL_12 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_EOBND_EN_49	end-of-band indication of Channel interrupt. This bit is the control end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 IDMAC_EOBND_EN_48	end-of-band indication of Channel interrupt. This bit is the control end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_EOBND_EN_47	end-of-band indication of Channel interrupt. This bit is the control end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_EOBND_EN_46	end-of-band indication of Channel interrupt. This bit is the control end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
13 IDMAC_EOBND_EN_45	end-of-band indication of Channel interrupt. This bit is the control end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.

38.5.18 Interrupt Control Register 13 (IPU_INT_CTRL_13)

This register contains part of IPU interrupts controls. The controls of the threshold crossing indication (TH) of DMA Channels interrupts [31:0] can be found in this register.

Address: 260_0000h base + 6Ch offset = 260_006Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_TH_EN_31	0	IDMAC_TH_EN_29	IDMAC_TH_EN_28	IDMAC_TH_EN_27	IDMAC_TH_EN_26	IDMAC_TH_EN_25	IDMAC_TH_EN_24	IDMAC_TH_EN_23	IDMAC_TH_EN_22	IDMAC_TH_EN_21	IDMAC_TH_EN_20	IDMAC_TH_EN_19	IDMAC_TH_EN_18	IDMAC_TH_EN_17	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_TH_EN_15	IDMAC_TH_EN_14	IDMAC_TH_EN_13	IDMAC_TH_EN_12	IDMAC_TH_EN_11	IDMAC_TH_EN_10	IDMAC_TH_EN_9	IDMAC_TH_EN_8	0		IDMAC_TH_EN_5	0	IDMAC_TH_EN_3	IDMAC_TH_EN_2	IDMAC_TH_EN_1	IDMAC_TH_EN_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_CTRL_13 field descriptions

Field	Description
31 IDMAC_TH_EN_31	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
30 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
29 IDMAC_TH_EN_29	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
28 IDMAC_TH_EN_28	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_CTRL_13 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
27 IDMAC_TH_EN_27	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
26 IDMAC_TH_EN_26	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IDMAC_TH_EN_25	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
24 IDMAC_TH_EN_24	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
23 IDMAC_TH_EN_23	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
22 IDMAC_TH_EN_22	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
21 IDMAC_TH_EN_21	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_TH_EN_20	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_CTRL_13 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_TH_EN_19	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_TH_EN_18	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_TH_EN_17	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_TH_EN_15	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_TH_EN_14	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
13 IDMAC_TH_EN_13	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_TH_EN_12	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_13 field descriptions (continued)

Field	Description
11 IDMAC_TH_EN_11	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_TH_EN_10	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_TH_EN_9	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
8 IDMAC_TH_EN_8	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
7–6 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
5 IDMAC_TH_EN_5	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
4 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
3 IDMAC_TH_EN_3	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 IDMAC_TH_EN_2	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

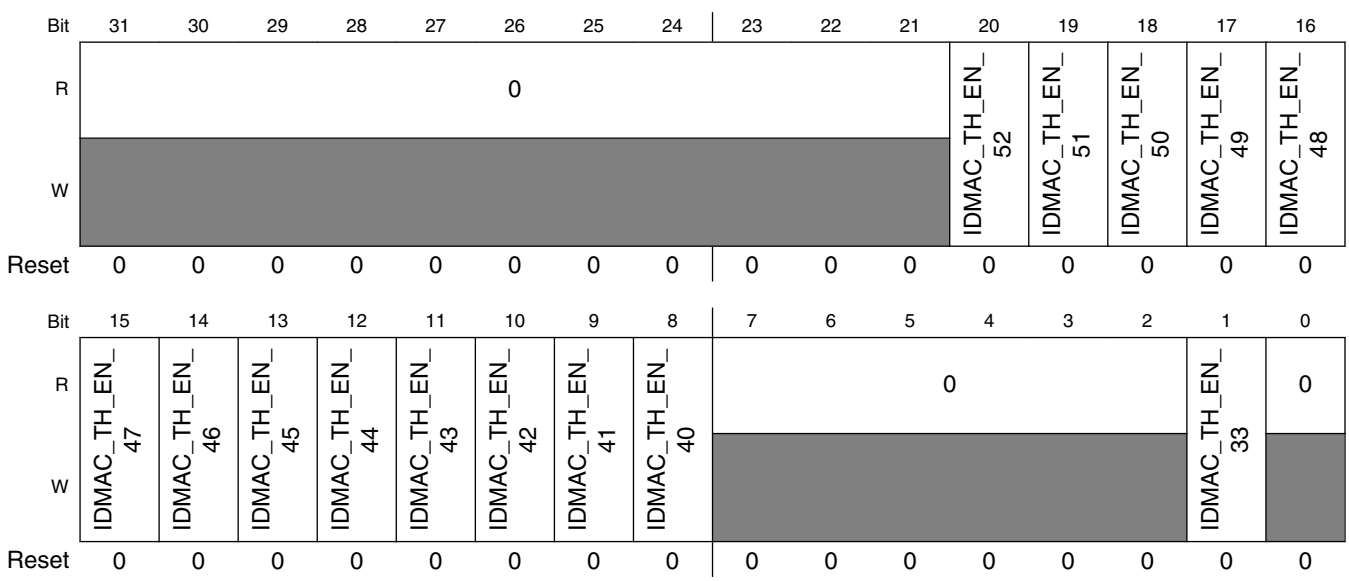
IPU_INT_CTRL_13 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_TH_EN_ 1	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 IDMAC_TH_EN_ 0	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

38.5.19 Interrupt Control Register 14 (IPU_INT_CTRL_14)

This register contains part of IPU interrupts controls. The controls of the Threshold crossing indication (TH) of DMA Channels interrupts [63:32] can be found in this register.

Address: 260_0000h base + 70h offset = 260_0070h



IPU_INT_CTRL_14 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_TH_EN_52	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_TH_EN_51	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_TH_EN_50	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_TH_EN_49	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 IDMAC_TH_EN_48	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_TH_EN_47	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_TH_EN_46	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_14 field descriptions (continued)

Field	Description
13 IDMAC_TH_EN_45	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_TH_EN_44	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_TH_EN_43	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_TH_EN_42	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_TH_EN_41	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
8 IDMAC_TH_EN_40	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
7–2 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_TH_EN_33	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_INT_CTRL_14 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.

38.5.20 Interrupt Control Register15 (IPU_INT_CTRL_15)

This register contains part of IPU interrupts controls. The controls of general purpose interrupts can be found in this register.

Address: 260_0000h base + 74h offset = 260_0074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DI1_CNT_EN_PRE_8_EN	DI1_CNT_EN_PRE_3_EN	DI1_DISP_CLK_EN_PRE_EN	DI0_CNT_EN_PRE_10_EN	DI0_CNT_EN_PRE_9_EN	DI0_CNT_EN_PRE_8_EN	DI0_CNT_EN_PRE_7_EN	DI0_CNT_EN_PRE_6_EN	DI0_CNT_EN_PRE_5_EN	DI0_CNT_EN_PRE_4_EN	DI0_CNT_EN_PRE_3_EN	DI0_CNT_EN_PRE_2_EN	DI0_CNT_EN_PRE_1_EN	DI0_CNT_EN_PRE_0_EN	DC_ASYNC_STOP_EN	DC_DP_START_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DI_VSYNC_PRE_1_EN	DI_VSYNC_PRE_0_EN	DC_FC_6_EN	DC_FC_4_EN	DC_FC_3_EN	DC_FC_2_EN	DC_FC_1_EN	DC_FC_0_EN	DP_ASF_BRAKE_EN	DP_SF_BRAKE_EN	DP_ASF_END_EN	DP_ASF_START_EN	DP_SF_END_EN	DP_SF_START_EN	SNOOPING2_INT_EN	SNOOPING1_INT_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_CTRL_15 field descriptions

Field	Description
31 DI1_CNT_EN_PRE_8_EN	This bit enables the interrupt that is a result of a trigger generated by counter #8 of DI1 0 Interrupt is disabled. 1 Interrupt is enabled.
30 DI1_CNT_EN_PRE_3_EN	This bit enables the interrupt that is a result of a trigger generated by counter #3 of DI1 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_15 field descriptions (continued)

Field	Description
29 DI1_DISP_CLK_EN_PRE_EN	DI1_DISP_CLK_EN_PRE_EN 0 Interrupt is disabled. 1 Interrupt is enabled.
28 DI0_CNT_EN_PRE_10_EN	This bit enables the interrupt that is a result of a trigger generated by counter #10 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
27 DI0_CNT_EN_PRE_9_EN	This bit enables the interrupt that is a result of a trigger generated by counter #9 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
26 DI0_CNT_EN_PRE_8_EN	This bit enables the interrupt that is a result of a trigger generated by counter #8 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
25 DI0_CNT_EN_PRE_7_EN	This bit enables the interrupt that is a result of a trigger generated by counter #7 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
24 DI0_CNT_EN_PRE_6_EN	This bit enables the interrupt that is a result of a trigger generated by counter #6 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
23 DI0_CNT_EN_PRE_5_EN	This bit enables the interrupt that is a result of a trigger generated by counter #5 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
22 DI0_CNT_EN_PRE_4_EN	This bit enables the interrupt that is a result of a trigger generated by counter #4 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
21 DI0_CNT_EN_PRE_3_EN	This bit enables the interrupt that is a result of a trigger generated by counter #3 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
20 DI0_CNT_EN_PRE_2_EN	This bit enables the interrupt that is a result of a trigger generated by counter #2 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
19 DI0_CNT_EN_PRE_1_EN	This bit enables the interrupt that is a result of a trigger generated by counter #1 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
18 DI0_CNT_EN_PRE_0_EN	This bit enables the interrupt that is a result of a trigger generated by counter #0 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_15 field descriptions (continued)

Field	Description
17 DC_ASYNC_ STOP_EN	This bit enables the interrupt asserted anytime the DP stops an async flow and moves to a sync flow 0 Interrupt is disabled. 1 Interrupt is enabled.
16 DC_DP_START_ EN	This bit enables the interrupt asserted anytime the DP start a new sync or async flow or when an async flow is interrupted by a sync flow 0 Interrupt is disabled. 1 Interrupt is enabled.
15 DI_VSYNC_ PRE_1_EN	This bit enables the DI1 interrupt indicating of a VSYNC signal asserted 2 rows before the VSYNC sent to the display 0 Interrupt is disabled. 1 Interrupt is enabled.
14 DI_VSYNC_ PRE_0_EN	This bit enables the DI0 interrupt indicating of a VSYNC signal asserted 2 rows before the VSYNC sent to the display 0 Interrupt is disabled. 1 Interrupt is enabled.
13 DC_FC_6_EN	This bit enables the DC Frame Complete on channel #6 interrupt 0 Interrupt is disabled. 1 Interrupt is enabled.
12 DC_FC_4_EN	This bit enables the DC Frame Complete on channel #4 interrupt 0 Interrupt is disabled. 1 Interrupt is enabled.
11 DC_FC_3_EN	This bit enables the DC Frame Complete on channel #3 interrupt 0 Interrupt is disabled. 1 Interrupt is enabled.
10 DC_FC_2_EN	This bit enables the DC Frame Complete on channel #2 interrupt 0 Interrupt is disabled. 1 Interrupt is enabled.
9 DC_FC_1_EN	This bit enables they'd Frame Complete on channel #1 interrupt 0 Interrupt is disabled. 1 Interrupt is enabled.
8 DC_FC_0_EN	This bit enables they'd Frame Complete on channel #0 interrupt 0 Interrupt is disabled. 1 Interrupt is enabled.
7 DP_ASF_ BRAKE_EN	DP Async Flow Brake enable bit. This bit enables the interrupt that is a result of the async flow brake at the DP 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_15 field descriptions (continued)

Field	Description
6 DP_SF_BRAKE_EN	DP Sync Flow Brake enable bit. This bit enables the interrupt that is a result of the Sync flow brake at the DP 0 Interrupt is disabled. 1 Interrupt is enabled.
5 DP_ASF_END_EN	DP Async Flow End enable bit. This bit enables the interrupt that is a result of the Async flow end at the DP 0 Interrupt is disabled. 1 Interrupt is enabled.
4 DP_ASF_START_EN	DP Async Flow Start enable bit. This bit enables the interrupt that is a result of the Async flow start at the DP 0 Interrupt is disabled. 1 Interrupt is enabled.
3 DP_SF_END_EN	DP Sync Flow End enable bit. This bit enables the interrupt that is a result of the Sync flow end at the DP 0 Interrupt is disabled. 1 Interrupt is enabled.
2 DP_SF_START_EN	DP Sync Flow Start enable bit. This bit enables the interrupt that is a result of the Sync flow start at the DP 0 Interrupt is disabled. 1 Interrupt is enabled.
1 SNOOPING2_INT_EN	IPU snooping 2 interrupt enable bit. This bit enables the interrupt that is a result of the detection of a snooping 2 signal assertion coming to the IPU 0 Interrupt is disabled. 1 Interrupt is enabled.
0 SNOOPING1_INT_EN	IPU snooping 1 interrupt enable bit. This bit enables the interrupt that is a result of the detection of a snooping 1 signal assertion coming to the IPU 0 Interrupt is disabled. 1 Interrupt is enabled.

38.5.21 SDMA Event Control Register 1 (IPU_SDMA_EVENT_1)

This register contains part of IPU SDMA events controls. The controls of EOF (end of frame) of DMA Channels SDMA events [31:0] can be found in this register.

Address: 260_0000h base + 78h offset = 260_0078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_EOF_SDMA_EN_31	0	IDMAC_EOF_SDMA_EN_29	IDMAC_EOF_SDMA_EN_28	IDMAC_EOF_SDMA_EN_27	IDMAC_EOF_SDMA_EN_26	IDMAC_EOF_SDMA_EN_25	IDMAC_EOF_SDMA_EN_24	IDMAC_EOF_SDMA_EN_23	IDMAC_EOF_SDMA_EN_22	IDMAC_EOF_SDMA_EN_21	IDMAC_EOF_SDMA_EN_20	IDMAC_EOF_SDMA_EN_19	IDMAC_EOF_SDMA_EN_18	IDMAC_EOF_SDMA_EN_17	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_EOF_SDMA_EN_15	IDMAC_EOF_SDMA_EN_14	IDMAC_EOF_SDMA_EN_13	IDMAC_EOF_SDMA_EN_12	IDMAC_EOF_SDMA_EN_11	IDMAC_EOF_SDMA_EN_10	IDMAC_EOF_SDMA_EN_9	IDMAC_EOF_SDMA_EN_8	0		IDMAC_EOF_SDMA_EN_5	0	IDMAC_EOF_SDMA_EN_3	IDMAC_EOF_SDMA_EN_2	IDMAC_EOF_SDMA_EN_1	IDMAC_EOF_SDMA_EN_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_SDMA_EVENT_1 field descriptions

Field	Description
31 IDMAC_EOF_SDMA_EN_31	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
30 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
29 IDMAC_EOF_SDMA_EN_29	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
28 IDMAC_EOF_SDMA_EN_28	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_1 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
27 IDMAC_EOF_ SDMA_EN_27	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
26 IDMAC_EOF_ SDMA_EN_26	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
25 IDMAC_EOF_ SDMA_EN_25	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
24 IDMAC_EOF_ SDMA_EN_24	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
23 IDMAC_EOF_ SDMA_EN_23	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
22 IDMAC_EOF_ SDMA_EN_22	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
21 IDMAC_EOF_ SDMA_EN_21	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
20 IDMAC_EOF_ SDMA_EN_20	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_EOF_ SDMA_EN_19	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_1 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
18 IDMAC_EOF_SDMA_EN_18	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
17 IDMAC_EOF_SDMA_EN_17	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
16 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
15 IDMAC_EOF_SDMA_EN_15	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
14 IDMAC_EOF_SDMA_EN_14	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
13 IDMAC_EOF_SDMA_EN_13	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
12 IDMAC_EOF_SDMA_EN_12	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_EOF_SDMA_EN_11	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
10 IDMAC_EOF_SDMA_EN_10	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

Table continues on the next page...

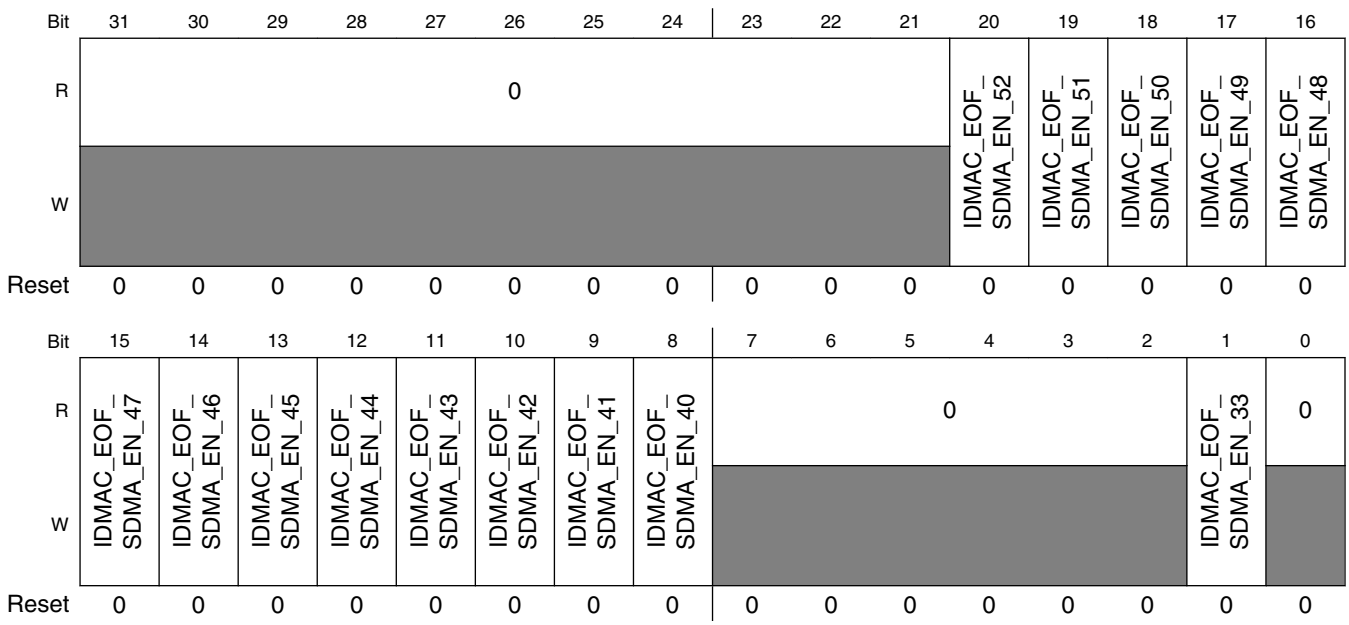
IPU_SDMA_EVENT_1 field descriptions (continued)

Field	Description
9 IDMAC_EOF_ SDMA_EN_9	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
8 IDMAC_EOF_ SDMA_EN_8	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
7–6 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
5 IDMAC_EOF_ SDMA_EN_5	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
4 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
3 IDMAC_EOF_ SDMA_EN_3	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
2 IDMAC_EOF_ SDMA_EN_2	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
1 IDMAC_EOF_ SDMA_EN_1	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
0 IDMAC_EOF_ SDMA_EN_0	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

38.5.22 SDMA Event Control Register 2 (IPU_SDMA_EVENT_2)

This register contains part of IPU SDMA events controls. The controls of EOF (end of frame) of DMA Channels SDMA events [63:32] can be found in this register.

Address: 260_0000h base + 7Ch offset = 260_007Ch



IPU_SDMA_EVENT_2 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
20 IDMAC_EOF_ SDMA_EN_52	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_EOF_ SDMA_EN_51	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
18 IDMAC_EOF_ SDMA_EN_50	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_2 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
17 IDMAC_EOF_ SDMA_EN_49	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
16 IDMAC_EOF_ SDMA_EN_48	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
15 IDMAC_EOF_ SDMA_EN_47	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
14 IDMAC_EOF_ SDMA_EN_46	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
13 IDMAC_EOF_ SDMA_EN_45	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
12 IDMAC_EOF_ SDMA_EN_44	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_EOF_ SDMA_EN_43	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
10 IDMAC_EOF_ SDMA_EN_42	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
9 IDMAC_EOF_ SDMA_EN_41	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

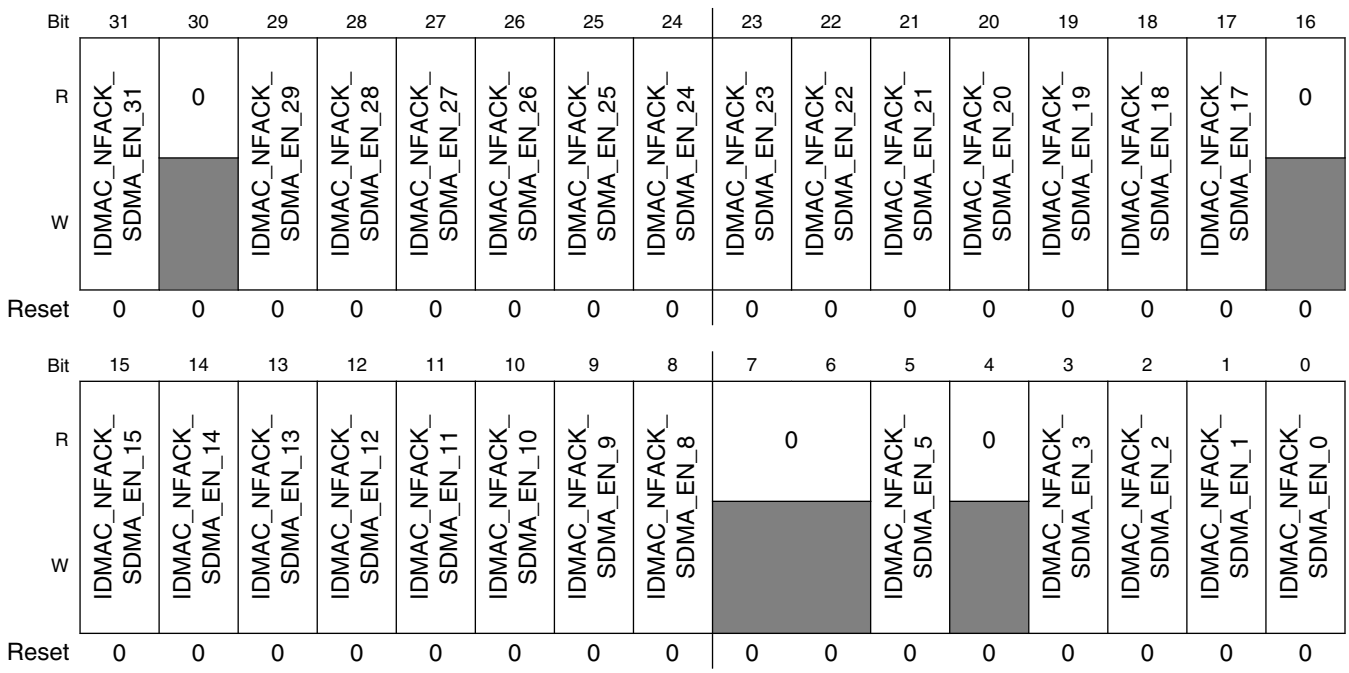
IPU_SDMA_EVENT_2 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
8 IDMAC_EOF_SDMA_EN_40	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
7–2 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
1 IDMAC_EOF_SDMA_EN_33	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
0 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.

38.5.23 SDMA Event Control Register 3 (IPU_SDMA_EVENT_3)

This register contains part of IPU SDMA events controls. The controls of NFACK (New Frame Acknowledge) of DMA Channels SDMA events [31:0] can be found in this register.

Address: 260_0000h base + 80h offset = 260_0080h



IPU_SDMA_EVENT_3 field descriptions

Field	Description
31 IDMAC_NFACK_SDMA_EN_31	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
30 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
29 IDMAC_NFACK_SDMA_EN_29	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_3 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
28 IDMAC_NFACK_SDMA_EN_28	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
27 IDMAC_NFACK_SDMA_EN_27	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
26 IDMAC_NFACK_SDMA_EN_26	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
25 IDMAC_NFACK_SDMA_EN_25	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
24 IDMAC_NFACK_SDMA_EN_24	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
23 IDMAC_NFACK_SDMA_EN_23	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
22 IDMAC_NFACK_SDMA_EN_22	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
21 IDMAC_NFACK_SDMA_EN_21	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_3 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
20 IDMAC_NFACK_SDMA_EN_20	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_NFACK_SDMA_EN_19	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
18 IDMAC_NFACK_SDMA_EN_18	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
17 IDMAC_NFACK_SDMA_EN_17	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
16 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
15 IDMAC_NFACK_SDMA_EN_15	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
14 IDMAC_NFACK_SDMA_EN_14	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
13 IDMAC_NFACK_SDMA_EN_13	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

Table continues on the next page...

IPU_SDMA_EVENT_3 field descriptions (continued)

Field	Description
12 IDMAC_NFACK_SDMA_EN_12	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_NFACK_SDMA_EN_11	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
10 IDMAC_NFACK_SDMA_EN_10	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
9 IDMAC_NFACK_SDMA_EN_9	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
8 IDMAC_NFACK_SDMA_EN_8	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
7–6 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
5 IDMAC_NFACK_SDMA_EN_5	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
4 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
3 IDMAC_NFACK_SDMA_EN_3	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

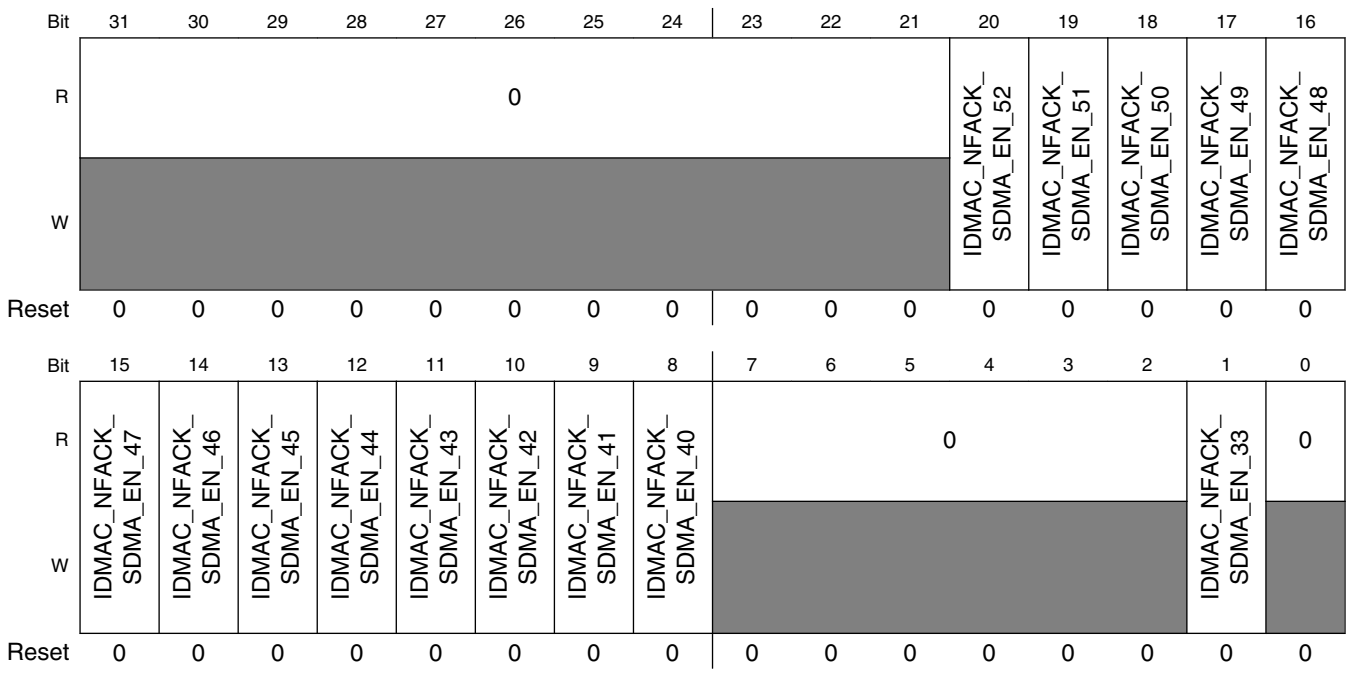
IPU_SDMA_EVENT_3 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
2 IDMAC_NFACK_SDMA_EN_2	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
1 IDMAC_NFACK_SDMA_EN_1	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
0 IDMAC_NFACK_SDMA_EN_0	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

38.5.24 SDMA Event Control Register 4 (IPU_SDMA_EVENT_4)

This register contains part of IPU SDMA events controls. The controls of NFACK (New Frame Acknowledge) of DMA Channels SDMA events [63:32] can be found in this register.

Address: 260_0000h base + 84h offset = 260_0084h



IPU_SDMA_EVENT_4 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
20 IDMAC_NFACK_SDMA_EN_52	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_NFACK_SDMA_EN_51	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_4 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
18 IDMAC_NFACK_SDMA_EN_50	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
17 IDMAC_NFACK_SDMA_EN_49	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
16 IDMAC_NFACK_SDMA_EN_48	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
15 IDMAC_NFACK_SDMA_EN_47	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
14 IDMAC_NFACK_SDMA_EN_46	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
13 IDMAC_NFACK_SDMA_EN_45	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
12 IDMAC_NFACK_SDMA_EN_44	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_NFACK_SDMA_EN_43	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

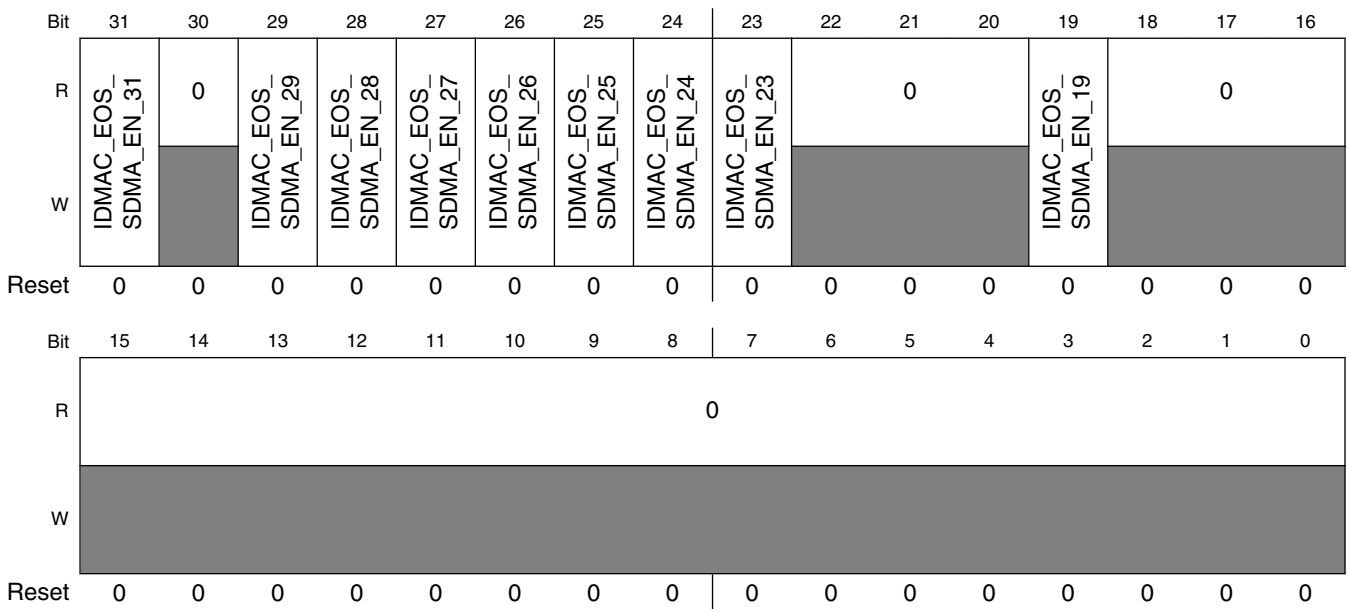
IPU_SDMA_EVENT_4 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
10 IDMAC_NFACK_SDMA_EN_42	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
9 IDMAC_NFACK_SDMA_EN_41	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
8 IDMAC_NFACK_SDMA_EN_40	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
7–2 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
1 IDMAC_NFACK_SDMA_EN_33	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
0 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.

38.5.25 SDMA Event Control Register 7 (IPU_SDMA_EVENT_7)

This register contains part of IPU SDMA events controls. The controls of EOS (End of Scroll) of DMA Channels SDMA events [31:0] can be found in this register.

Address: 260_0000h base + 88h offset = 260_0088h



IPU_SDMA_EVENT_7 field descriptions

Field	Description
31 IDMAC_EOS_SDMA_EN_31	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
30 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
29 IDMAC_EOS_SDMA_EN_29	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
28 IDMAC_EOS_SDMA_EN_28	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

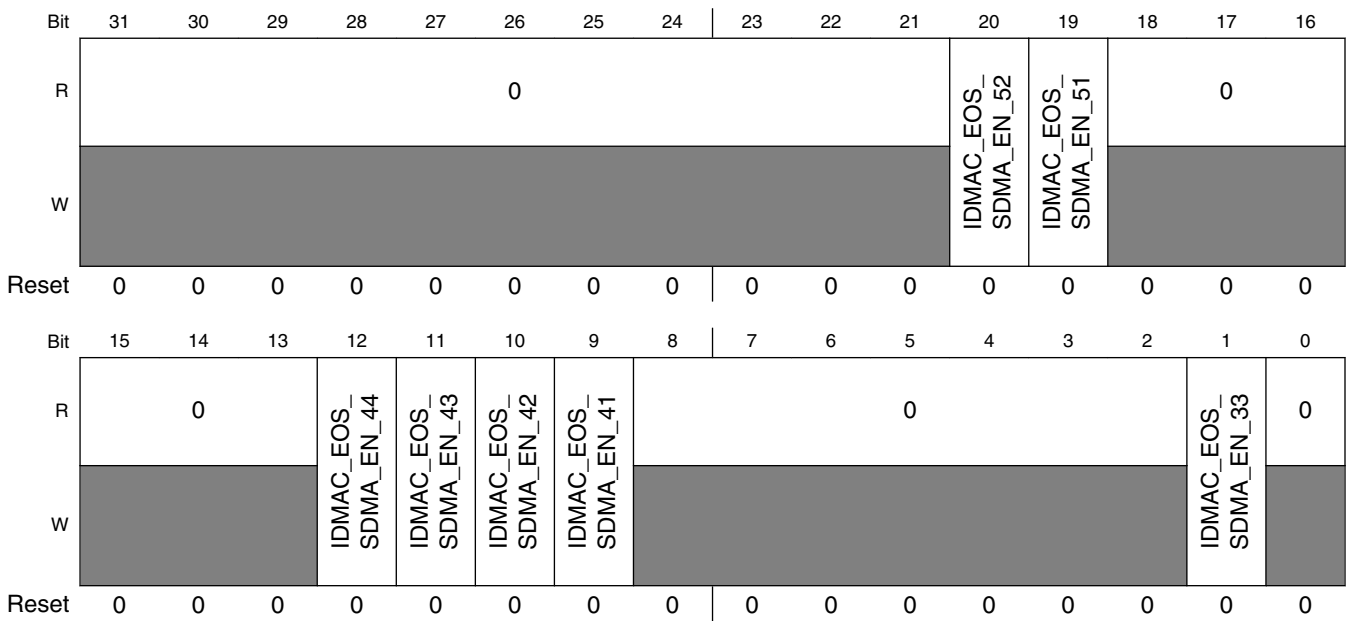
IPU_SDMA_EVENT_7 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
27 IDMAC_EOS_ SDMA_EN_27	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
26 IDMAC_EOS_ SDMA_EN_26	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
25 IDMAC_EOS_ SDMA_EN_25	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
24 IDMAC_EOS_ SDMA_EN_24	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
23 IDMAC_EOS_ SDMA_EN_23	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
22–20 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_EOS_ SDMA_EN_19	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.

38.5.26 SDMA Event Control Register 8 (IPU_SDMA_EVENT_8)

This register contains part of IPU SDMA events controls. The controls of EOS (End of Scroll) of DMA Channels SDMA events [63:32] can be found in this register.

Address: 260_0000h base + 8Ch offset = 260_008Ch



IPU_SDMA_EVENT_8 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
20 IDMAC_EOS_ SDMA_EN_52	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_EOS_ SDMA_EN_51	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
18–13 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.

Table continues on the next page...

IPU_SDMA_EVENT_8 field descriptions (continued)

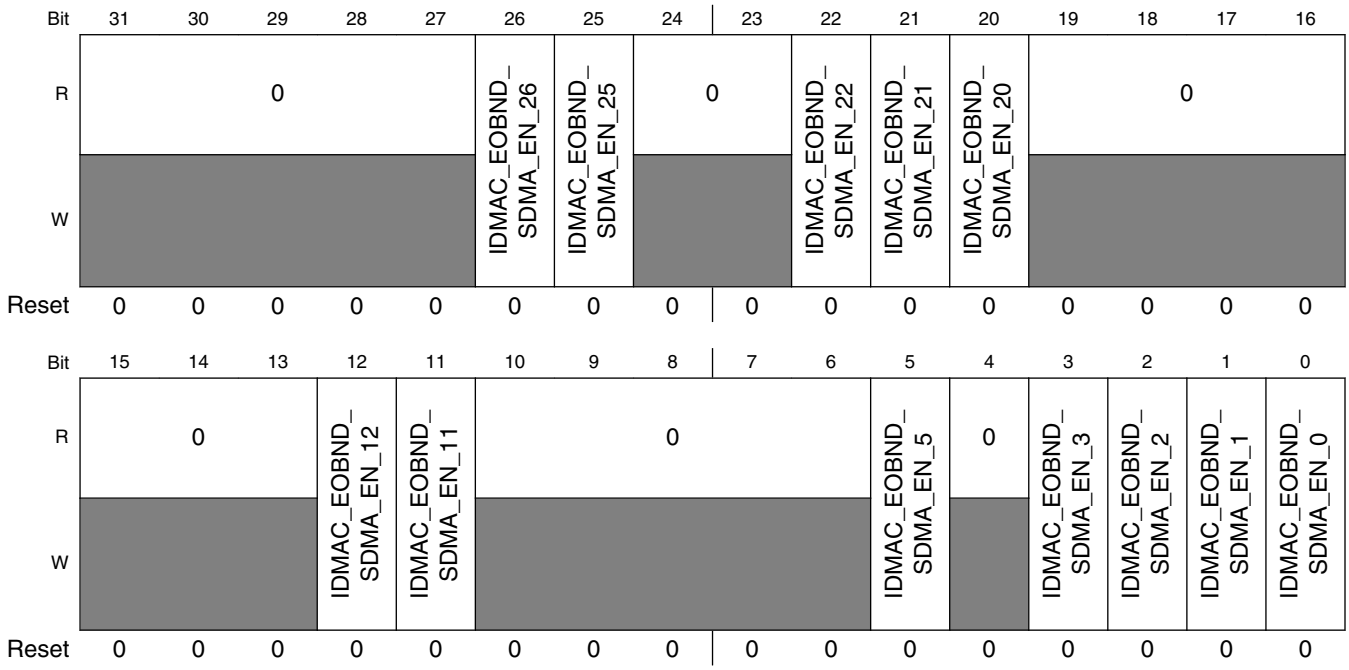
Field	Description
12 IDMAC_EOS_ SDMA_EN_44	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_EOS_ SDMA_EN_43	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
10 IDMAC_EOS_ SDMA_EN_42	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
9 IDMAC_EOS_ SDMA_EN_41	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
8–2 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
1 IDMAC_EOS_ SDMA_EN_33	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
0 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.

38.5.27 SDMA Event Control Register 11 (IPU_SDMA_EVENT_11)

This register contains part of IPU SDMA events controls. The controls of EOBND (End of Band) of DMA Channels SDMA events [31:0] can be found in this register.

- Hide VDOA_SYNC for all versions
- Show VDOA_SYNC for IPUv3H version.
- The table below tagged with other settings (like IPU3M_only) should be hidden in IPUv3H version.

Address: 260_0000h base + 90h offset = 260_0090h



IPU_SDMA_EVENT_11 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
26 IDMAC_EOBND_SDMA_EN_26	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
25 IDMAC_EOBND_SDMA_EN_25	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
24–23 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
22 IDMAC_EOBND_SDMA_EN_22	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
21 IDMAC_EOBND_SDMA_EN_21	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_11 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
20 IDMAC_EOBND_SDMA_EN_20	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19–13 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
12 IDMAC_EOBND_SDMA_EN_12	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_EOBND_SDMA_EN_11	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
10–6 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
5 IDMAC_EOBND_SDMA_EN_5	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
4 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
3 IDMAC_EOBND_SDMA_EN_3	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
2 IDMAC_EOBND_SDMA_EN_2	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
1 IDMAC_EOBND_SDMA_EN_1	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

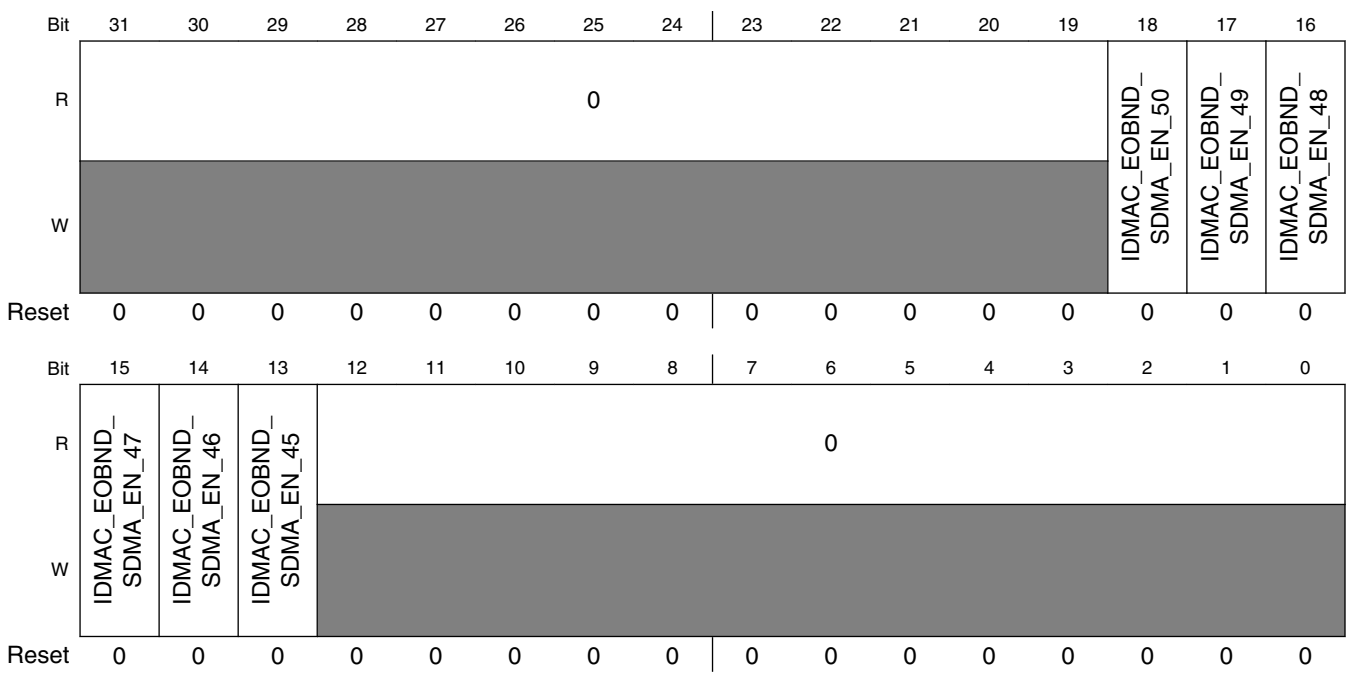
IPU_SDMA_EVENT_11 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
0 IDMAC_EOBND_ SDMA_EN_0	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

38.5.28 SDMA Event Control Register 12 (IPU_SDMA_EVENT_12)

This register contains part of IPU SDMA events controls. The controls of EOBND (End of Band) of DMA Channels SDMA events [63:32] can be found in this register.

Address: 260_0000h base + 94h offset = 260_0094h



IPU_SDMA_EVENT_12 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value 0.
	0 SDMA event is disabled. 1 SDMA event is enabled.

Table continues on the next page...

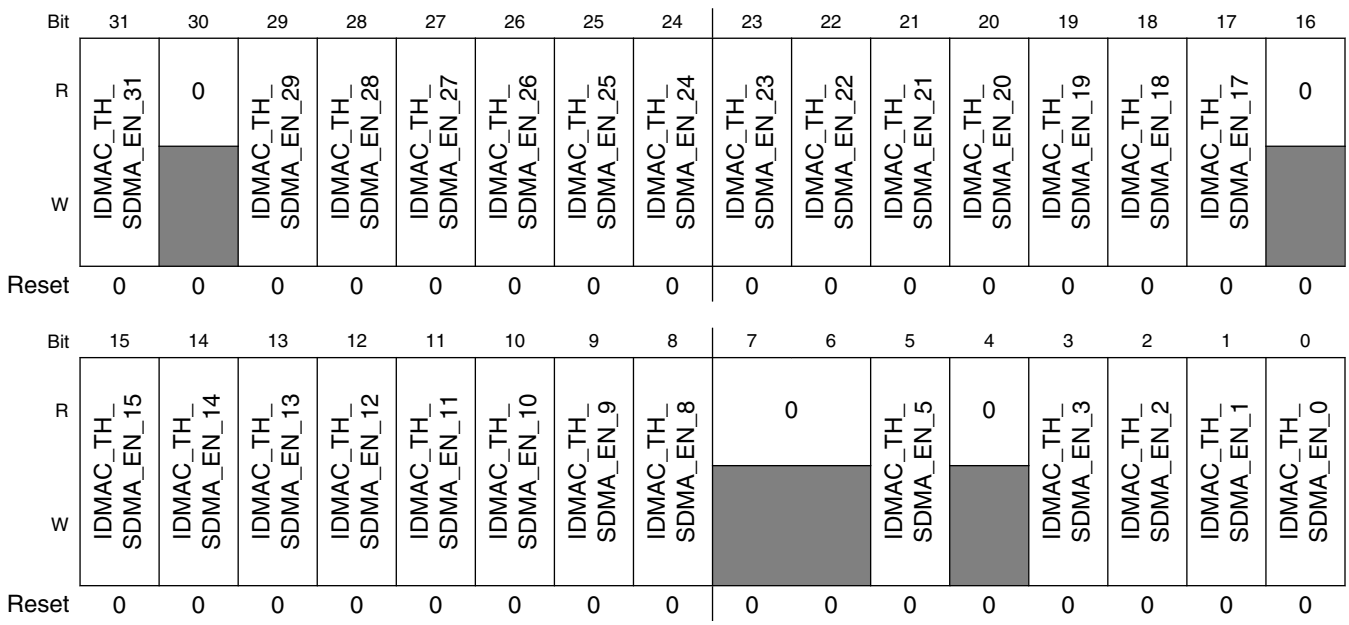
IPU_SDMA_EVENT_12 field descriptions (continued)

Field	Description
18 IDMAC_EOBND_SDMA_EN_50	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
17 IDMAC_EOBND_SDMA_EN_49	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
16 IDMAC_EOBND_SDMA_EN_48	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
15 IDMAC_EOBND_SDMA_EN_47	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
14 IDMAC_EOBND_SDMA_EN_46	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
13 IDMAC_EOBND_SDMA_EN_45	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.

38.5.29 SDMA Event Control Register 13 (IPU_SDMA_EVENT_13)

This register contains part of IPU SDMA events controls. The controls of TH (Threshold) of DMA Channels SDMA events [31:0] can be found in this register.

Address: 260_0000h base + 98h offset = 260_0098h



IPU_SDMA_EVENT_13 field descriptions

Field	Description
31 IDMAC_TH_ SDMA_EN_31	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
30 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
29 IDMAC_TH_ SDMA_EN_29	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
28 IDMAC_TH_ SDMA_EN_28	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_13 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
27 IDMAC_TH_ SDMA_EN_27	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
26 IDMAC_TH_ SDMA_EN_26	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
25 IDMAC_TH_ SDMA_EN_25	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
24 IDMAC_TH_ SDMA_EN_24	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
23 IDMAC_TH_ SDMA_EN_23	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
22 IDMAC_TH_ SDMA_EN_22	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
21 IDMAC_TH_ SDMA_EN_21	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
20 IDMAC_TH_ SDMA_EN_20	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_TH_ SDMA_EN_19	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_13 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
18 IDMAC_TH_ SDMA_EN_18	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
17 IDMAC_TH_ SDMA_EN_17	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
16 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
15 IDMAC_TH_ SDMA_EN_15	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
14 IDMAC_TH_ SDMA_EN_14	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
13 IDMAC_TH_ SDMA_EN_13	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
12 IDMAC_TH_ SDMA_EN_12	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_TH_ SDMA_EN_11	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
10 IDMAC_TH_ SDMA_EN_10	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

Table continues on the next page...

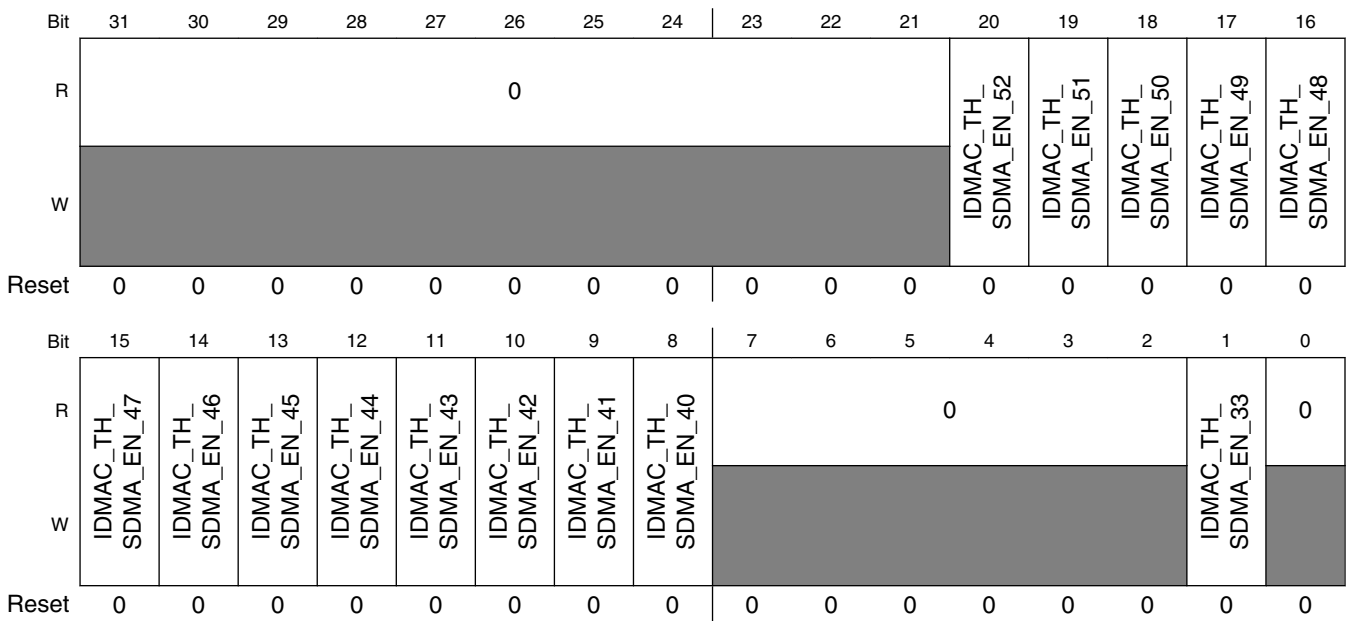
IPU_SDMA_EVENT_13 field descriptions (continued)

Field	Description
9 IDMAC_TH_ SDMA_EN_9	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
8 IDMAC_TH_ SDMA_EN_8	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
7–6 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
5 IDMAC_TH_ SDMA_EN_5	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
4 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
3 IDMAC_TH_ SDMA_EN_3	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
2 IDMAC_TH_ SDMA_EN_2	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
1 IDMAC_TH_ SDMA_EN_1	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
0 IDMAC_TH_ SDMA_EN_0	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

38.5.30 SDMA Event Control Register 14 (IPU_SDMA_EVENT_14)

This register contains part of IPU SDMA events controls. The controls of TH (Threshold) of DMA Channels SDMA events [63:32] can be found in this register.

Address: 260_0000h base + 9Ch offset = 260_009Ch



IPU_SDMA_EVENT_14 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
20 IDMAC_TH_ SDMA_EN_52	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_TH_ SDMA_EN_51	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

Table continues on the next page...

IPU_SDMA_EVENT_14 field descriptions (continued)

Field	Description
18 IDMAC_TH_ SDMA_EN_50	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
17 IDMAC_TH_ SDMA_EN_49	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
16 IDMAC_TH_ SDMA_EN_48	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
15 IDMAC_TH_ SDMA_EN_47	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
14 IDMAC_TH_ SDMA_EN_46	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
13 IDMAC_TH_ SDMA_EN_45	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
12 IDMAC_TH_ SDMA_EN_44	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_TH_ SDMA_EN_43	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

Table continues on the next page...

IPU_SDMA_EVENT_14 field descriptions (continued)

Field	Description
10 IDMAC_TH_ SDMA_EN_42	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
9 IDMAC_TH_ SDMA_EN_41	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
8 IDMAC_TH_ SDMA_EN_40	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
7–2 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.
1 IDMAC_TH_ SDMA_EN_33	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
0 Reserved	This read-only field is reserved and always has the value 0. 0 SDMA event is disabled. 1 SDMA event is enabled.

38.5.31 Shadow Registers Memory Priority 1 Register (IPU_SRM_PRI1)

The register controls the priority of SRM updates. The priority level for each block that has a shadow of its registers in the SRM should be unique. The priority level defines the order of SRM updates. A block with priority set to 010 will be updated before a block with priority set to 001.

Address: 260_0000h base + A0h offset = 260_00A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			CSI0_SRM_MODE		CSI0_SRM_PRI			0			CSI1_SRM_MODE		CSI1_SRM_PRI		
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

IPU_SRM_PRI1 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value 0.
12–11 CSI0_SRM_MODE	CSI0 SRM Mode This field controls the SRM logic that handles the CSI0 registers 00 Automatic swapping is disabled; ARM platform is allowed to access the CSI1's region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done of the next frame. 10 The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame 11 Update now. The SRM is controlled by the ARM Platform. The Register will be update now
10–8 CSI0_SRM_PRI	CSI0 SRM priority This bits define the priority of the CSI1 block
7–5 Reserved	This read-only field is reserved and always has the value 0.
4–3 CSI1_SRM_MODE	CSI1 SRM Mode This field controls the SRM logic that handles the CSI1 registers 00 Automatic swapping is disabled; ARM platform is allowed to access the CSI0's region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done of the next frame. 10 The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame 11 Update now. The SRM is controlled by the ARM platform. The Register will be update now
CSI1_SRM_PRI	CSI1 SRM priority This bits define the priority of the CSI0 module

38.5.32 Shadow Registers Memory Priority 2 Register (IPU_SRM_PRI2)

The register controls the priority of SRM updates. The priority level for each block that has a shadow of its registers in the SRM should be unique. The priority level defines the order of SRM updates. a block with priority set to 010 will be updated before a block with priority set to 001.

Address: 260_0000h base + A4h offset = 260_00A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			DI1_SRM_MODE		DI1_SRM_PRI			0			DI0_SRM_MCU_USE		DI0_SRM_PRI		
W																
Reset	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DC_6_SRM_MODE		DC_2_SRM_MODE		DC_SRM_PRI			DP_A1_SRM_MODE	DP_A0_SRM_MODE			DP_S_SRM_MODE		DP_SRM_PRI		
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1

IPU_SRM_PRI2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–27 DI1_SRM_MODE	DCI1 SRM Mode This field controls the SRM logic that handles the DI1 registers 00 Automatic swapping is disabled; ARM platform is allowed to access the DI1 region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done of the next frame. 10 The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame 11 Update now. The SRM is controlled by the ARM platform. The Register will be update now
26–24 DI1_SRM_PRI	DI1 SRM priority This bits define the priority of the DI1 module
23–21 Reserved	This read-only field is reserved and always has the value 0.
20–19 DI0_SRM_MCU_USE	DI0 SRM is used by ARM platform This bit indicates that the registers of the DI0 are currently being updated by the ARM platform. The ARM platform should set this bit before accessing the SRM part that is relevant to the DI0. The ARM platform should clear this bit when the update procedure is finished. When this bit is set the SRM mechanism will not update the DI0's registers to avoid data coherency problems. 1 DI0 SRM is currently updated by the ARM platform 0 DI0 SRM s currently not updated by the ARM platform
18–16 DI0_SRM_PRI	DI0 SRM priority

Table continues on the next page...

IPU_SRM_PRI2 field descriptions (continued)

Field	Description
	This bits define the priority of the DI0 module
15–14 DC_6_SRM_MODE	DC Group #6 SRM Mode This field controls the SRM logic that handles the DC Group #6 registers 00 Automatic swapping is disabled; ARM platform is allowed to access the DC Group #6's region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done of the next frame. 10 The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame 11 Update now. The SRM is controlled by the ARM platform. The Register will be update now
13–12 DC_2_SRM_MODE	DC Group #2 SRM Mode This field controls the SRM logic that handles the DC Group #2 registers 00 Automatic swapping is disabled; ARM platform is allowed to access the DC Group #2's region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done of the next frame. 10 The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame 11 Update now. The SRM is controlled by the ARM platform. The Register will be update now
11–9 DC_SRM_PRI	DC SRM priority This bits define the priority of the DC module
8–7 DP_A1_SRM_MODE	DP Async flow #1 SRM Mode This field controls the SRM logic that handles the DP Async flow #1 registers 00 Automatic swapping is disabled; ARM platform is allowed to access the DP Async flow #1 region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done of the next frame. 10 The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame 11 Update now. The SRM is controlled by the ARM platform. The Register will be update now
6–5 DP_A0_SRM_MODE	DP Async flow #0 SRM Mode This field controls the SRM logic that handles the DP Async flow #0 registers 00 Automatic swapping is disabled; ARM platform is allowed to access the DP Async flow #0 region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done of the next frame. 10 The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame 11 Update now. The SRM is controlled by the ARM platform. The Register will be update now
4–3 DP_S_SRM_MODE	DP sync flow SRM Mode This field controls the SRM logic that handles the DP sync flow registers 00 Automatic swapping is disabled; ARM platform is allowed to access the DP sync flow region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done on the next frame. 10 Reserved 11 Update now. The SRM is controlled by the ARM platform. The Register will be update now
DP_SRM_PRI	DP SRM priority This bits define the priority of the DP module

38.5.33 FSU Processing Flow 1 Register (IPU_FS_PROC_FLOW1)

This register contain controls for IPU's tasks.

Address: 260_0000h base + A8h offset = 260_00A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VF_IN_VALID	ENC_IN_VALID	VDI_SRC_SEL		PRP_SRC_SEL				VDI3_SRC_SEL		VDI1_SRC_SEL		PP_ROT_SRC_SEL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PP_SRC_SEL				PRPVF_ROT_SRC_SEL				0				PRPENC_ROT_SRC_SEL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_FS_PROC_FLOW1 field descriptions

Field	Description
31 VF_IN_VALID	View-finder Input valid. Setting this bit indicates that the buffer in memory for viewfinder is validated by the ARM platform (valid only when RWS_EN is '1'). 0 View-finder should skip buffer in memory. 1 View-finder should use buffer in memory.
30 ENC_IN_VALID	Encoding Input valid. Setting this bit indicates that the buffer in memory for encoding is validated by the ARM platform (valid only when RWS_EN is '1'). 0 Encoding should skip buffer in memory. 1 Encoding should use buffer in memory.
29–28 VDI_SRC_SEL	Source select for the VDIC This field is relevant if the VDIC works in de-interlacing mode (when VDI_CMB_EN bit is clear) 00 ARM platform 01 CSI direct (cb7) 10 Reserved 10 VDOA 11 Reserved
27–24 PRP_SRC_SEL	Source select for the Pre Processing Task 0000 ARM platform

Table continues on the next page...

IPU_FS_PROC_FLOW1 field descriptions (continued)

Field	Description
	0001 capture0 (smfc0) — — — 0011 capture2 (smfc2) — — — 0101 IC direct (cb7) — 0110 IRT Encoding 0111 IRT viewfinder 1000 Reserved 1001 Reserved 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2
23–22 VDI3_SRC_SEL	Source select for the VDIC plane #3 (IDMAC's CH 25) 00 ARM platform This field is relevant only if the VDIC works in combining mode (VDI_CMB_EN bit is set) 01 IRT viewfinder (ch 49) 10 IRT playback (ch 50) 11 post-processing (ch 22)
21–20 VDI1_SRC_SEL	Source select for the VDIC plane #1 (IDMAC's CH26) This field is relevant only if the VDIC works in combining mode (VDI_CMB_EN bit is set) 00 ARM platform 01 IRT viewfinder 10 IRT playback 11 post-processing
19–16 PP_ROT_SRC_SEL	Source select for the pre processing task of the IRT (CH 50) 0000 ARM platform 0001 capture0 (smfc0) — 0010 Reserved 0011 capture2 (smfc2) — 0100 Reserved 0101 Post-processing 0110 Reserved 0111 Reserved —

Table continues on the next page...

IPU_FS_PROC_FLOW1 field descriptions (continued)

Field	Description
	1000 Reserved — 1001 Reserved 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2
15–12 PP_SRC_SEL	Source select for the pre processing task of the IC 0000 ARM platform 0001 capture0 (smfc0) — 0010 Reserved 0011 capture2 (smfc2) — 0100 Reserved 0101 Reserved 0110 Rotation for post-processing 0111 Reserved — 1000 Reserved 1000 VDOA — 1001 Reserved 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2
11–8 PRPVF_ROT_ SRC_SEL	Source select for the view finder task of the IRT 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture1 (smfc1) — 0011 capture2 (smfc2) — 0100 capture3 (smfc3) — 0101 IC direct (cb7) — 0110 Reserved 0111 Reserved

Table continues on the next page...

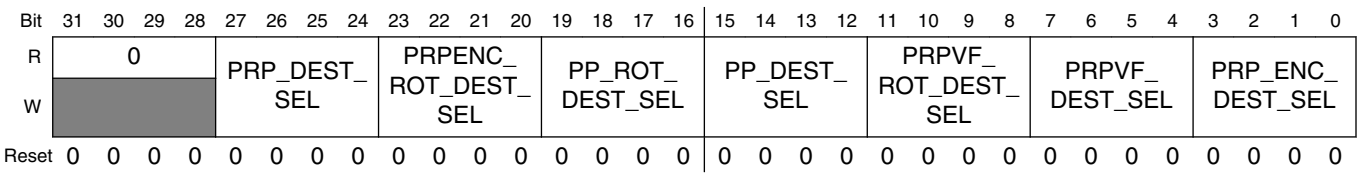
IPU_FS_PROC_FLOW1 field descriptions (continued)

Field	Description
	1000 View-finder 1001 Reserved 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2
7–4 Reserved	This read-only field is reserved and always has the value 0.
PRPENC_ROT_SRC_SEL	Source select for the encoding task of the IRT 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture1 (smfc1) — 0011 capture2 (smfc2) — 0100 capture3 (smfc3) — 0101 IC direct (cb7) — 0110 Reserved 0111 encoding 1000 Reserved 1001 Reserved 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2

38.5.34 FSU Processing Flow 2 Register (IPU_FS_PROC_FLOW2)

This register contains controls for IPU's tasks.

Address: 260_0000h base + ACh offset = 260_00ACh



IPU_FS_PROC_FLOW2 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–24 PRP_DEST_SEL	**Pre processing destination select (for channel DMAIC_7)** 0000 ARM platform 0001 IC input buffer (ch12) 0010 PP (ch11) 0011 PP_ROT (ch47) 0100 DC1 (ch28) 0101 DC2 (ch41) 0110 DP_ASYNC1 (ch24) 0111 DP_ASYNC0 (ch29) 1000 DP_SYNC1 (ch27) 1001 DP_SYNC0 (ch23) 1010 Alt DC2 (ch41) 1011 Alt DP_ASYNC1 (ch24) 1100 Alt DP_ASYNC0 (ch29) 1111 Reserved
23–20 PRPENC_ROT_DEST_SEL	Destination select for Rotation task coming from the Encoding input 0000 ARM platform 0001 Reserved 0010 Reserved — 0011 Reserved — 0100 Reserved 0101 IC Pre Processing 0110 Reserved 0111 DC1 (ch28) 1000 DC2 (ch41)

Table continues on the next page...

IPU_FS_PROC_FLOW2 field descriptions (continued)

Field	Description
	1001 **DP_SYNC0 (ch23)** 1010 **DP_SYNC1 (ch27)** 1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41) 1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**
19–16 PP_ROT_DEST_SEL	Destination select for Rotation task coming from the Post Processing input 0000 ARM platform 0001 Reserved 0010 Reserved 0011 Reserved 0100 IC Playback (Post Processing) — — 0101 VDI_PLANE3 (Ch 25) — — 0110 VDI_PLANE1 (Ch 26) 0111 DC1 (ch28) 1000 DC2 (ch41) 1001 **DP_SYNC0 (ch23)** 1010 **DP_SYNC1 (ch27)** 1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41) 1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**
15–12 PP_DEST_SEL	Destination select for post processing task 0000 ARM platform 0001 Reserved 0010 Reserved 0011 IRT playback — 0100 VDI_PLANE3 (Ch 25) — 0101 VDI_PLANE1 (Ch 26) 0110 Reserved 0111 DC1 (ch28) 1000 DC2 (ch41) 1001 **DP_SYNC0 (ch23)** 1010 **DP_SYNC1 (ch27)** 1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41)

Table continues on the next page...

IPU_FS_PROC_FLOW2 field descriptions (continued)

Field	Description
	1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**
11–8 PRPVF_ROT_ DEST_SEL	Destination select for Rotation task coming from the View finder input 0000 ARM platform 0001 Reserved 0010 Reserved — — 0011 VDI_PLANE3 (Ch 25) — — 0100 VDI_PLANE1 (Ch 26) 0101 IC Pre Processing 0110 Reserved 0111 DC1 (ch28) 1000 DC2 (ch41) 1001 **DP_SYNC0 (ch23)** 1010 **DP_SYNC1 (ch27)** 1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41) 1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**
7–4 PRPVF_DEST_ SEL	Destination select for View finder task 0000 ARM platform 0001 IRT viewfinder 0010 Reserved 0011 Reserved 0100 Reserved 0101 Reserved 0110 Reserved 0111 DC1 (ch28) 1000 DC2 (ch41) 1001 **DP_SYNC0 (ch23)** 1010 **DP_SYNC1 (ch27)** 1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41) 1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**
PRP_ENC_ DEST_SEL	Destination select for Encoding task 0000 ARM platform 0001 IRT Encoding 0010 Reserved

Table continues on the next page...

IPU_FS_PROC_FLOW2 field descriptions (continued)

Field	Description
	0011 Reserved 0100 Reserved 0101 Reserved 0110 Reserved 0111 DC1 (ch28) 1000 DC2 (ch41) 1001 **DP_SYNC0 (ch23)** 1010 **DP_SYNC1 (ch27)** 1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41) 1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**

38.5.35 FSU Processing Flow 3 Register (IPU_FS_PROC_FLOW3)

This register contains controls for IPU's tasks.

- Hide VPU_SUB_FRAME_SYNC for all versions
- Show VPU_SUB_FRAME_SYNC for IPUv3H version.
- The table below tagged with other settings (like IPU3M_only) should be hidden in IPUv3H version.
- This requires some sophisticated conditional tag settings

Address: 260_0000h base + B0h offset = 260_00B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						VPU_DEST_SEL		EXT_SRC2_DEST_SEL		EXT_SRC1_DEST_SEL		0		VDOA_DEST_SEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		SMFC3_DEST_SEL			SMFC2_DEST_SEL			SMFC1_DEST_SEL				SMFC0_DEST_SEL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_FS_PROC_FLOW3 field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.
25–24 VPU_DEST_SEL	This bits selects the corresponding IDMAC channel's EOL indication to be used for sub frame synchronization with the VPU. The corresponding IDMAC channel's EOLI bit at the CPMEM has to be set as well.

Table continues on the next page...

IPU_FS_PROC_FLOW3 field descriptions (continued)

Field	Description
	00 disabled 01 capture0 (smfc0) **(ch0)** 10 capture2 (smfc2) **(ch2)** 11 IC viewfinder (ch21)
23–22 EXT_SRC2_ DEST_SEL	Destination select for External Source 2 00 disabled 01 **DP_SYNC0 (ch23)** 10 **DP_SYNC1 (ch27)** 11 DC1 (ch28)
21–20 EXT_SRC1_ DEST_SEL	Destination select for External Source 1 00 disabled 01 **DP_SYNC0 (ch23)** 10 **DP_SYNC1 (ch27)** 11 DC1 (ch28)
19–18 Reserved	This read-only field is reserved and always has the value 0.
17–16 VDOA_DEST_ SEL	**Destination select for VDOA** 00 disabled 01 IC Playback (Post Processing) 10 VDI (ch8,ch9 & ch10 or ch9 according to VDI_MOT_SEL settings) 11 Reserved
15–14 Reserved	This read-only field is reserved and always has the value 0.
13–11 SMFC3_DEST_ SEL	Destination select for SMFC3 000 ARM platform 001 IRT Encoding 010 IRT viewfinder 011 IRT playback 100 IC Playback (Post Processing) 101 IC Pre Processing — 111 Reserved
10–7 SMFC2_DEST_ SEL	Destination select for SMFC2 0000 ARM platform 0001 IRT Encoding 0010 IRT viewfinder 0011 IRT playback 0100 IC Playback (Post Processing) 0101 IC Pre Processing — 0111 DC1 (ch28) 1000 DC2 (ch41) 1001 **DP_SYNC0 (ch23)**

Table continues on the next page...

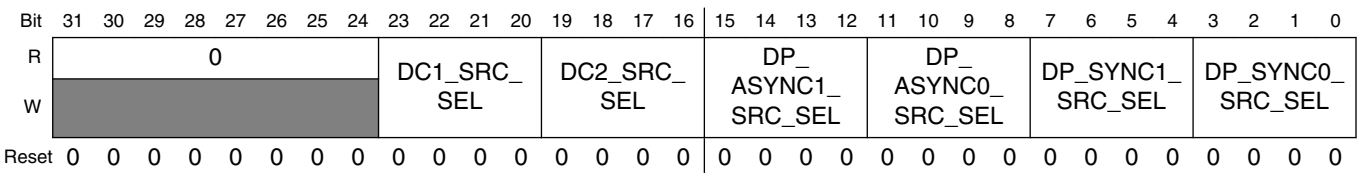
IPU_FS_PROC_FLOW3 field descriptions (continued)

Field	Description
	1010 **DP_SYNC1 (ch27)** 1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41) 1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**
6–4 SMFC1_DEST_SEL	Destination select for SMFC1 000 ARM platform 001 IRT Encoding 010 IRT viewfinder 011 IRT playback 100 IC Playback (Post Processing) 101 IC Pre Processing — 111 Reserved
SMFC0_DEST_SEL	Destination select for SMFC0 0000 ARM platform 0001 IRT Encoding 0010 IRT viewfinder 0011 IRT playback 0100 IC Playback (Post Processing) 0101 IC Pre Processing — 0111 DC1 (ch28) 1000 DC2 (ch41) 1001 **DP_SYNC0 (ch23)** 1010 **DP_SYNC1 (ch27)** 1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41) 1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**

38.5.36 FSU Displaying Flow 1 Register (IPU_FS_DISP_FLOW1)

This register contains controls for IPU's tasks.

Address: 260_0000h base + B4h offset = 260_00B4h



IPU_FS_DISP_FLOW1 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value 0.
23–20 DC1_SRC_SEL	Source select for DS1/DS2 - MG (graphics) plane (ch28) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder 0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved 1011 autoref 1100 autoref+snoop1 — 1101 External source #1 (e.g. an external block like GPU) 1110 snoop1 — 1111 External source #2 (e.g. an external block like GPU)
19–16 DC2_SRC_SEL	Source select for DS3 (ch41) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2)

Table continues on the next page...

IPU_FS_DISP_FLOW1 field descriptions (continued)

Field	Description
	—
	0011 IC encoding
	0100 IC viewfinder
	0101 IC playback
	0110 IRT Encoding
	0111 IRT viewfinder
	1000 IRT playback
	—
	1001 Reserved
	—
	1010 Reserved
	1011 autoref
	1100 autoref+snoop1
	1101 autoref+snoop2
	1110 snoop1
	1111 snoop2
15–12 DP_ASYNC1_ SRC_SEL	Source select for DS1/DS2 - Vx (video) plane (ch24) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder 0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2
11–8 DP_ASYNC0_ SRC_SEL	Source select for DS2 - MG (graphics) plane (ch29) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder

Table continues on the next page...

IPU_FS_DISP_FLOW1 field descriptions (continued)

Field	Description
	0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2
7–4 DP_SYNC1_ SRC_SEL	Source select for DS1/DS2 - Vx (video) plane (ch27) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder 0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved — — — — — — 1110 snoop1 1111 snoop2
DP_SYNC0_ SRC_SEL	Source select for DS2 - MG (graphics) plane (ch23) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder

Table continues on the next page...

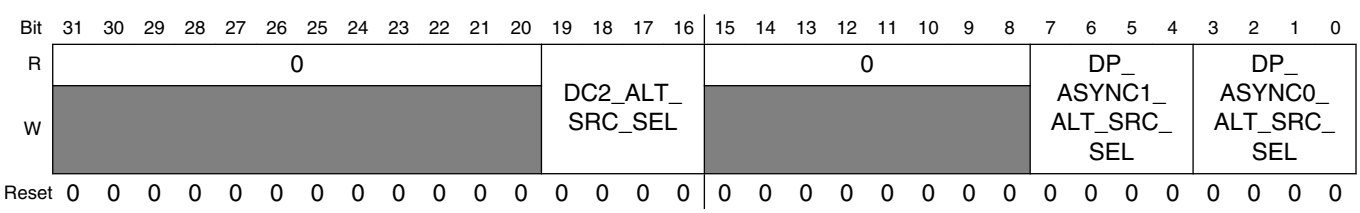
IPU_FS_DISP_FLOW1 field descriptions (continued)

Field	Description
	0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved — — — — — — 1110 snoop1 1111 snoop2

38.5.37 FSU Displaying Flow 2 Register (IPU_FS_DISP_FLOW2)

This register contains controls for IPU's tasks.

Address: 260_0000h base + B8h offset = 260_00B8h



IPU_FS_DISP_FLOW2 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 DC2_ALT_SRC_ SEL	Source select for Alternate DS3 (ch41) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder

Table continues on the next page...

IPU_FS_DISP_FLOW2 field descriptions (continued)

Field	Description
	0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2
15–8 Reserved	This read-only field is reserved and always has the value 0.
7–4 DP_ASYNC1_ ALT_SRC_SEL	Source select for alternate DS1/DS2 - Vx (video) plane (ch24) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder 0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2
DP_ASYNC0_ ALT_SRC_SEL	Source select for alternate DS2 - MG (graphics) plane (ch29) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder 0101 IC playback

Table continues on the next page...

IPU_FS_DISP_FLOW2 field descriptions (continued)

Field	Description
	0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2

38.5.38 SKIP Register (IPU_SKIP)

This register controls the different frame skipping supported by the IPU.

Address: 260_0000h base + BCh offset = 260_00BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VDI_SKIP												VDI_MAX_ RATIO_SKIP				CSI_SKIP_IC_VF					CSI_ MAX_ RATIO_ SKIP_IC_ VF			CSI_SKIP_IC_ ENC					CSI_ MAX_ RATIO_ SKIP_IC_ ENC		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_SKIP field descriptions

Field	Description
31–20 VDI_SKIP	VDI_SKIP These 12 bits define the skipping pattern of the frames send from the VDIC. The VDIC avoids reading fields from the memory if the output frame is skipped. Skipping is relevant only if the source to the VDIC is coming from the CSI. Skipping is done for a set of frames. The number of frames in a set is defined at VDI_MAX_RATIO_SKIP. when VDI_MAX_RATIO_SKIP = 1 => VDI_SKIP[1:0] is used; other bits are ignored when VDI_MAX_RATIO_SKIP = 2 => VDI_SKIP[2:0] are used; other bits are ignored .. .. when VDI_MAX_RATIO_SKIP = 11 => VDI_SKIP[11:0] are used;

Table continues on the next page...

IPU_SKIP field descriptions (continued)

Field	Description
19–16 VDI_MAX_ RATIO_SKIP	Maximum Ratio Skip for VDIC These bits define the number of frames in a skipping set. The maximum value of this bits is 11. When set to 0 the skipping is disabled.
15–11 CSI_SKIP_IC_VF	CSI SKIP IC_VF These 5 bits define the skipping pattern of the frames send to the IC for view finder task from one of the CSIs as defined on the CSI_SEL and IC_INPUT bits Skipping is done for a set of frames. The number of frames in a set is defined at CSI_MAX_RATIO_SKIP_IC_VF. when CSI_MAX_RATIO_SKIP_IC_VF = 1 => CSI_SKIP_IC_VF[1:0] are used; other bits are ignored when CSI_MAX_RATIO_SKIP_IC_VF = 2 => CSI_SKIP_IC_VF[2:0] are used; other bits are ignored when CSI_MAX_RATIO_SKIP_IC_VF =3 => CSI_SKIP_IC_VF[3:0] are used; other bits are ignored when CSI_MAX_RATIO_SKIP_IC_VF = 4 => CSI_SKIP_IC_VF[4:0] are used; Setting bit #n of CSI_SKIP_IC_VF means that the #n frame in the set is skipped. For example: if CSI_MAX_RATIO_SKIP_IC_VF = 4 and CSI_SKIP_IC_VF = 11010 Frames #0 & Frame #2 will not be skipped as bit0 and bit2 are cleared Frames #1 & Frame #3 will be skipped as bit1 and bit3 are set bit #4 is ignored as CSI_MAX_RATIO_SKIP_IC_VF is set to 4
10–8 CSI_MAX_ RATIO_SKIP_ IC_VF	CSI Maximum Ratio Skip for IC (view finder task) These bits define the number of frames in a skipping set. The maximum value of this bits is 4. When set to 0 the skipping is disabled.
7–3 CSI_SKIP_IC_ ENC	CSI SKIP IC_ENC These 5 bits define the skipping pattern of the frames send to the IC for encoding task from one of the CSIs as defined on the CSI_SEL and IC_INPUT bits Skipping is done for a set of frames. The number of frames in a set is defined at CSI_MAX_RATIO_SKIP_IC_ENC. when CSI_MAX_RATIO_SKIP_IC_ENC = 1 => CSI_SKIP_IC_ENC[1:0] are used; other bits are ignored when CSI_MAX_RATIO_SKIP_IC_ENC = 2 => CSI_SKIP_IC_ENC[2:0] are used; other bits are ignored when CSI_MAX_RATIO_SKIP_IC_ENC = 3 => CSI_SKIP_IC_ENC[3:0] are used; other bits are ignored when CSI_MAX_RATIO_SKIP_IC_ENC = 4 => CSI_SKIP_IC_ENC[4:0] are used; Setting bit #n of CSI_SKIP_IC_ENC means that the #n frame in the set is skipped. For example: if CSI_MAX_RATIO_SKIP_IC_ENC = 4 and CSI_SKIP_IC_ENC = 11010 Frames #0 & Frame #2 will not be skipped as bit0 and bit2 are cleared Frames #1 & Frame #3 will be skipped as bit1 and bit3 are set bit #4 is ignored as CSI_MAX_RATIO_SKIP_IC_ENC is set to 4
CSI_MAX_ RATIO_SKIP_ IC_ENC	CSI Maximum Ratio Skip for IC (encoding task) These bits define the number of frames in a skipping set. The maximum value of this bits is 4. When set to 0 the skipping is disabled.

38.5.39 Display Alternate Configuration Register (IPU_DISP_ALT_CONF)

This register controls various parameters that are used for alternate flows related to the display modules.

Address: 260_0000h base + C0h offset = 260_00C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

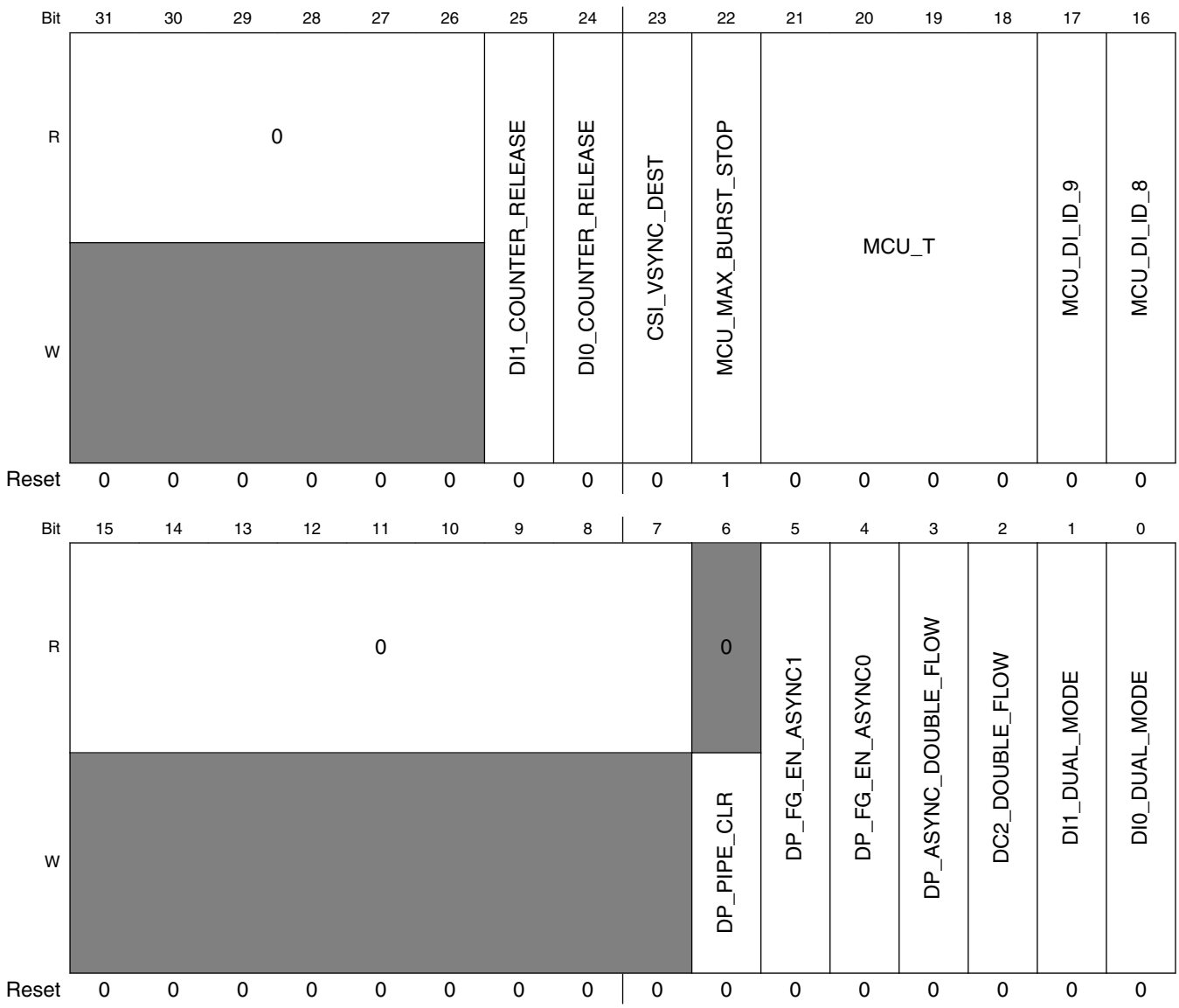
IPU_DISP_ALT_CONF field descriptions

Field	Description
Reserved	This read-only field is reserved and always has the value 0.

38.5.39 Display General Control Register (IPU_DISP_GEN)

This register controls various aspects of the display port.

Address: 260_0000h base + C4h offset = 260_00C4h



IPU_DISP_GEN field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DISP_GEN field descriptions (continued)

Field	Description
25 DI1_COUNTER_RELEASE	DI1 Counter release By default the DI0 counters responsible for waveform generation for sync flow are frozen. For the first attempt to use the DI in sync flow the user should set this bit 1 counter is released and running 0 counter is cleared and stopped
24 DI0_COUNTER_RELEASE	DI1 Counter release By default the DI0 counters responsible for waveform generation for sync flow are frozen. For the first attempt to use the DI in sync flow the user should set this bit 1 counter is released and running 0 counter is cleared and stopped
23 CSI_VSYNC_DEST	CSI_VSYNC destination This bit defines the destination of the VSYNC coming from the CSI's 1 csi1_vsync is connected to DI0; csi0_vsync is connected to DI1 0 csi0_vsync is connected to DI0; csi1_vsync is connected to DI1
22 MCU_MAX_BURST_STOP	ARM platform Maximal burst This bit limit the maximal unspecified length burst. 1 The maximum unspecified burst length is 8-beat 0 The unspecified burst length is unlimited
21–18 MCU_T	The address space for accesses through the AHB-lite slave port is MB and it is split internally (with 32MB resolution) according to bits [28:25] of the address. Using the following notation: Address = (ID[31:29], MSB[28:25], LSB[24:0]) The address is used as follows ("T" is a configurable integer between 0 and 13): MSB<T: access to an external device, with address = (MSB, LSB) T<=MSB<14: access to an external device, with address (MSB-T, LSB)
17 MCU_DI_ID_9	MCU_DI_ID_9 - DI ID via DC channel 9. This bit defines the DI that the ARM platform DC's access via channel #9 1 ARM platform accesses DC's channel #9 via DI1. 0 ARM platform accesses DC's channel #9 via DI0.
16 MCU_DI_ID_8	MCU_DI_ID_8 - DI ID via DC channel 8. This bit defines the DI that the ARM platform DC's access via channel #8 1 ARM platform accesses DC's channel #8 via DI1. 0 ARM platform accesses DC's channel #8 via DI0.
15–7 Reserved	This read-only field is reserved and always has the value 0.
6 DP_PIPE_CLR	DP Pipe Clear This bit clears the internal pipe of the DP. The user may use this bit in case of an error condition This is a self clear bit

Table continues on the next page...

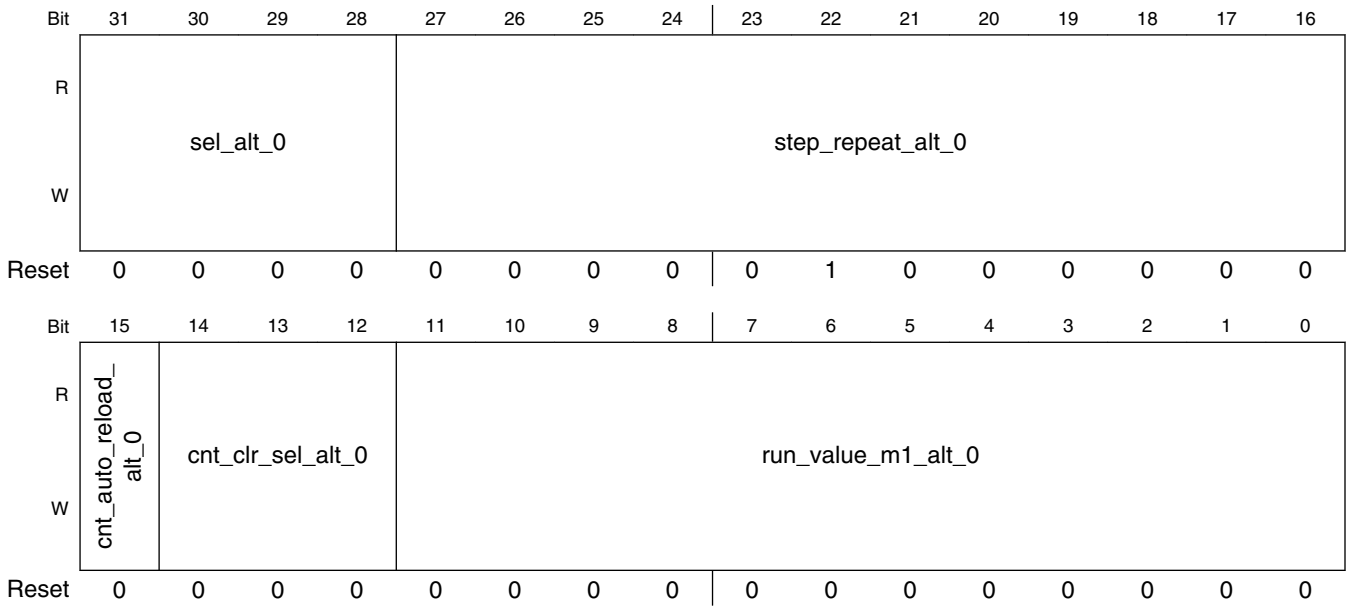
IPU_DISP_GEN field descriptions (continued)

Field	Description
	1 Clear the internal pipe of the DP 0 Idle - does nothing
5 DP_FG_EN_ASYNC1	FG_EN - partial plane Enable for async flow 1. This bit enables the partial plane channel. 1 partial plane channel is enabled. 0 partial plane channel is disabled.
4 DP_FG_EN_ASYNC0	FG_EN - partial plane Enable for async flow 0. This bit enables the partial plane channel. 1 partial plane channel is enabled. 0 partial plane channel is disabled.
3 DP_ASYNC_DOUBLE_FLOW	DP Async Double Flow. This bit define how many async flows are currently handles via DP channel (ch24+29) 1 2 flows are handled via DP 0 single flow is handled via DP
2 DC2_DOUBLE_FLOW	DC2 Double Flow. This bit define how many flows are currently handles via DC2 channel (ch41) 1 2 flows are handled via DC2 0 single flow is handled via DC2
1 DI1_DUAL_MODE	DI1 dual mode control 1 DI1 operates in dual mode 0 DI1 is not in dual mode
0 DI0_DUAL_MODE	DI0 dual mode control 1 DI0 operates in dual mode 0 DI0 is not in dual mode

38.5.40 Display Alternate Flow Control Register 1 (IPU_DISP_ALT1)

This register controls various aspects of the display port.

Address: 260_0000h base + C8h offset = 260_00C8h



IPU_DISP_ALT1 field descriptions

Field	Description
31–28 sel_alt_0	Select alternative parameters instead of DI Sync Wave Gen counter#. The DI is selected according to DP's synchronous channel destination 0000-disable 0001 instead of counter 1 0010 instead of counter 2 1000 instead of counter 8
27–16 step_repeat_alt_0	This fields defines the amount of repetitions that will be performed by the counter
15 cnt_auto_reload_alt_0	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the step_repeat_alt_0 field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the step_repeat_alt_0 field
14–12 cnt_clr_sel_alt_0	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock.

Table continues on the next page...

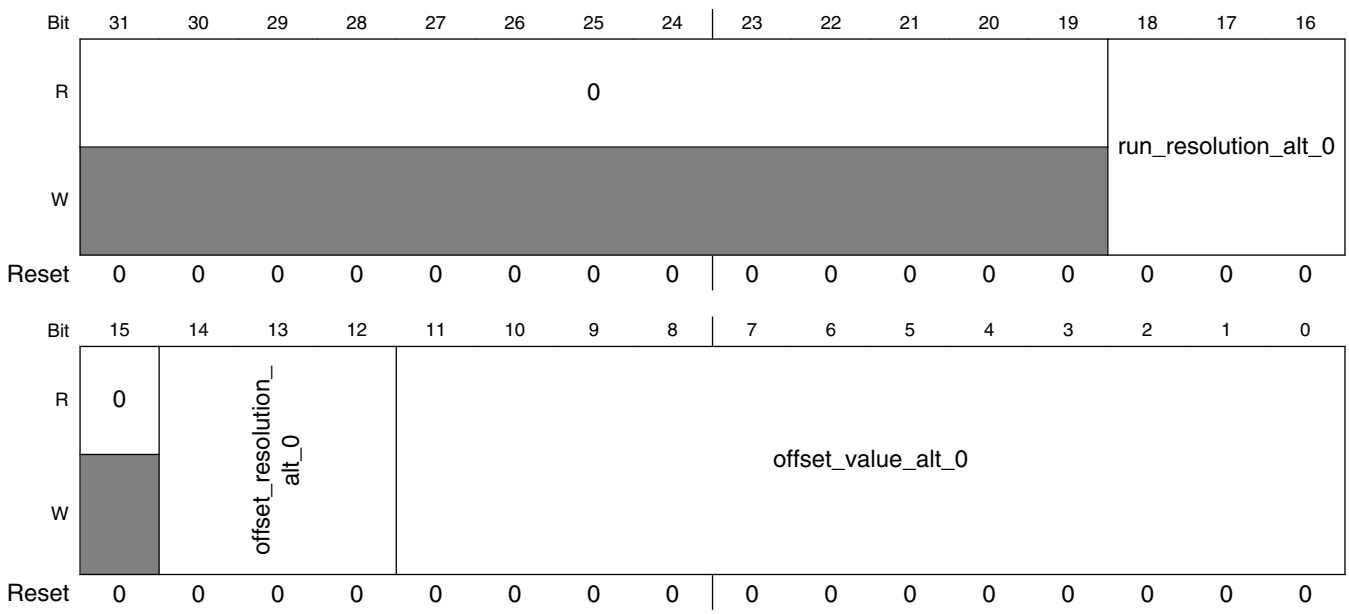
IPU_DISP_ALT1 field descriptions (continued)

Field	Description
	010 Reserved 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
run_value_m1_alt_0	Counter pre defined value This fields defines the counter pre defines value. real value- 1

38.5.41 Display Alternate Flow Control Register 2 (IPU_DISP_ALT2)

This register controls various aspects of the display port.

Address: 260_0000h base + CCh offset = 260_00CCh



IPU_DISP_ALT2 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DISP_ALT2 field descriptions (continued)

Field	Description
18–16 run_resolution_alt_0	Counter Run Resolution This field defines the trigger causing the counter to increment. The counter run resolution should be defined in the same way as in original DI's counter#
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 offset_resolution_alt_0	Counter offset Resolution This field defines the trigger causing the offset counter to increment The counter offset resolution should be defined in the same way as in original DI's counter#
offset_value_alt_0	Counter offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by

38.5.42 Display Alternate Flow Control Register 3 (IPU_DISP_ALT3)

This register controls various aspects of the display port.

Address: 260_0000h base + D0h offset = 260_00D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	sel_alt_1				step_repeat_alt_1											
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	cnt_auto_reload_alt_1	cnt_clr_sel_alt_1							run_value_m1_alt_1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

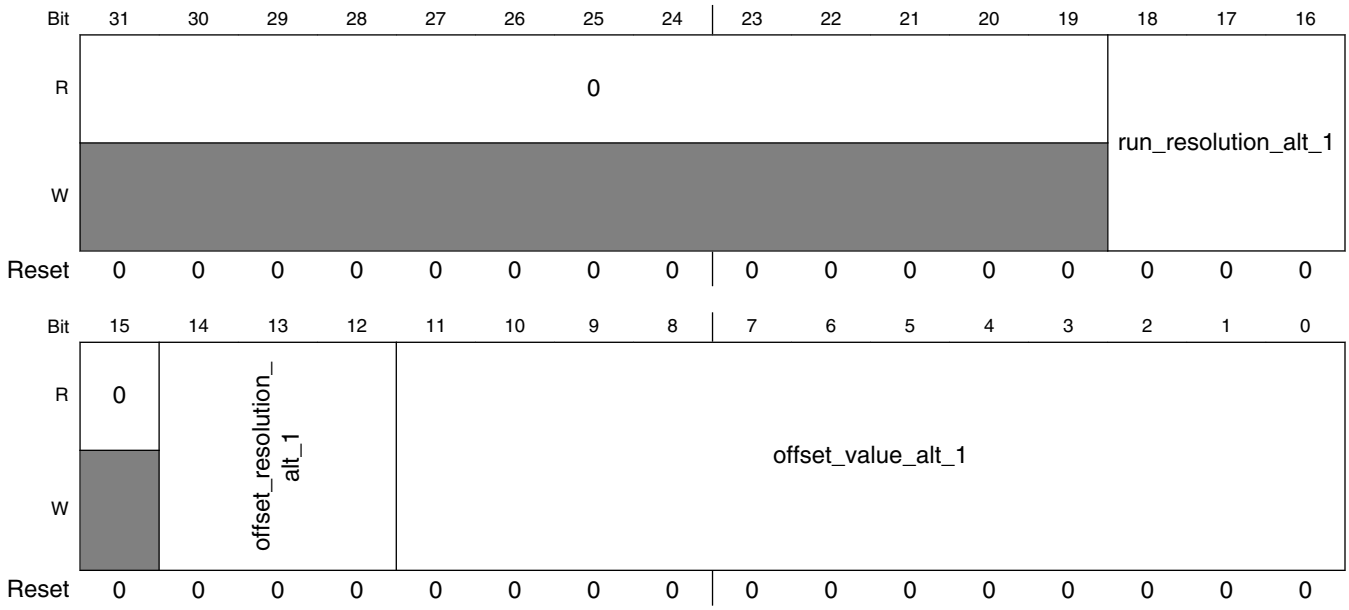
IPU_DISP_ALT3 field descriptions

Field	Description
31–28 sel_alt_1	Select alternative parameters instead of DI Sync Wave Gen counter#. The DI is selected according to DP's synchronous channel destination 0000 disable 0001 instead of counter 1 0010 instead of counter 2 1000 instead of counter 8
27–16 step_repeat_alt_1	This fields defines the amount of repetitions that will be performed by the counter
15 cnt_auto_reload_alt_1	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the step_repeat_alt_0 field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the step_repeat_alt_0 field
14–12 cnt_clr_sel_alt_1	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 Reserved 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
run_value_m1_alt_1	Counter pre defined value This fields defines the counter pre defines value. real value- 1

38.5.43 Display Alternate Flow Control Register 4 (IPU_DISP_ALT4)

This register controls various aspects of the display port.

Address: 260_0000h base + D4h offset = 260_00D4h



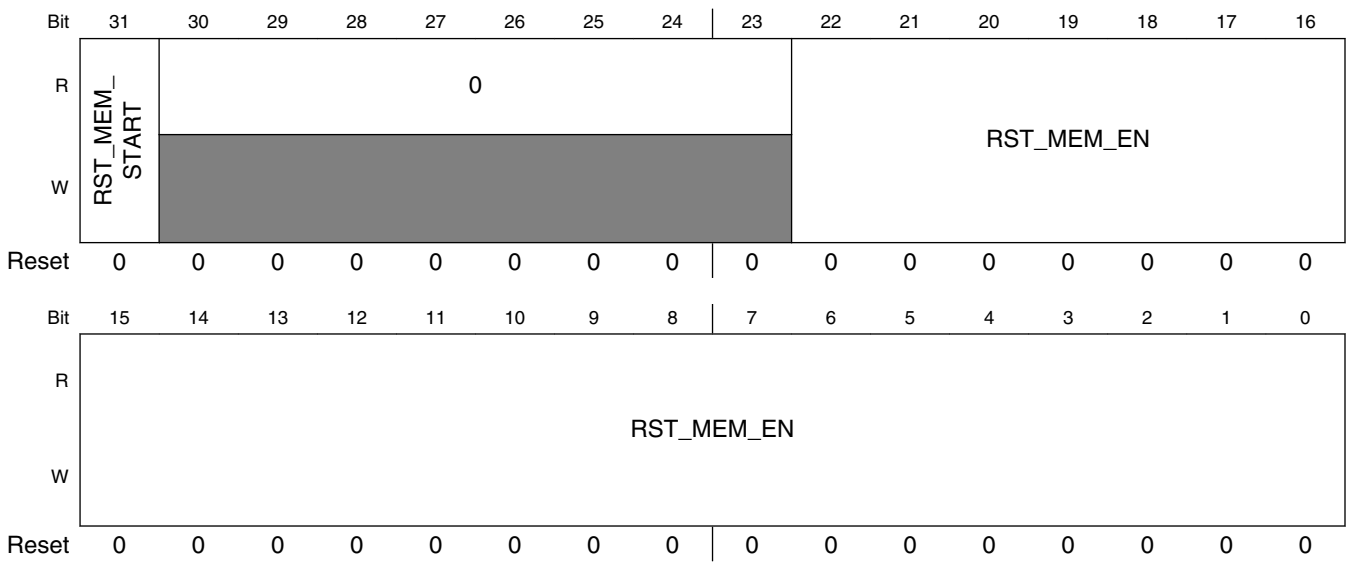
IPU_DISP_ALT4 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value 0.
18–16 run_resolution_alt_1	Counter Run Resolution This field defines the trigger causing the counter to increment. The counter run resolution should be defined in the same way as in original DI's counter#
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 offset_resolution_alt_1	Counter offset Resolution This field defines the trigger causing the offset counter to increment The counter offset resolution should be defined in the same way as in original DI's counter#
offset_value_alt_1	Counter offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by

38.5.44 Memory Reset Control Register (IPU_MEM_RST)

This register controls the memory reset mechanism. IPU has a hardware mechanism for clearing the content of the internal memories. This allows the user to clear the content of or more of the internal memories without the need to perform write accesses to the memories.

Address: 260_0000h base + DCh offset = 260_00DCh



IPU_MEM_RST field descriptions

Field	Description
31 RST_MEM_ START	Memory Reset Start Writing one to this bit activate the memory reset mechanism. The memories that their corresponding RST_MEM_EN bit is set will be cleared. When the memory reset mechanism completes the memory clearing procedure this bit will be automatically cleared. 1 The memory reset mechanism is activated and busy 0 Idle, the memory reset mechanism is not working.
30–23 Reserved	This read-only field is reserved and always has the value 0.
RST_MEM_EN	Reset Memory Enable Each bit on this field enables the memory reset mechanism for a specific memory. The user should set the relevant bits for the memories that need to be cleared. Below is the list of memories and their corresponding bit. srm = rst_mem_en[0] alpha = rst_mem_en[1] cpmem = rst_mem_en[2]

Table continues on the next page...

IPU_MEM_RST field descriptions (continued)

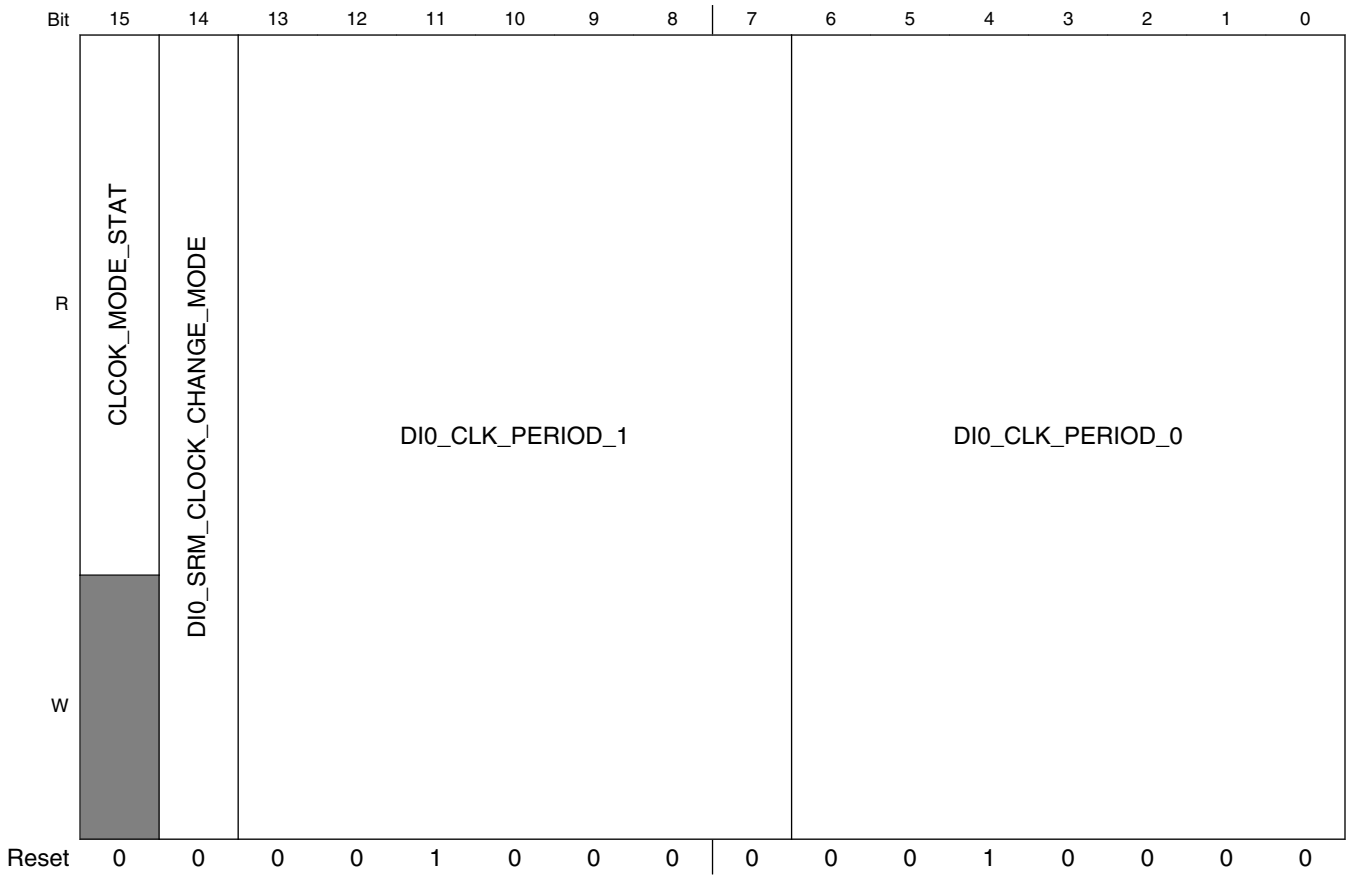
Field	Description
	tpm = rst_mem_en[3] mpm = rst_mem_en[4] bm = rst_mem_en[5] rm = rst_mem_en[6] dstm = rst_mem_en[7] dsom = rst_mem_en[8] lut0 = rst_mem_en[9] lut1 = rst_mem_en[10] ram_smfc = rst_mem_en[11] vdi_fifo2 = rst_mem_en[12] vdi_fifo3 = rst_mem_en[13] icb = rst_mem_en[14] vdi_fifo1 = rst_mem_en[15] dc_template = rst_mem_en[20] dmfc_rd = rst_mem_en[21] dmfc_wr = rst_mem_en[22]

38.5.45 Power Modes Control Register (IPU_PM)

This register controls the automatic transitions of the IPU between different power modes of the SoC and handles the clock change modes.

Address: 260_0000h base + E0h offset = 260_00E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LPSR_MODE	DI1_SRM_CLOCK_CHANGE_MODE	DI1_CLK_PERIOD_1							DI1_CLK_PERIOD_0						
Reset	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0



IPU_PM field descriptions

Field	Description
31 LPSR_MODE	LPSR Mode This bit indicates that the next attempt for entering low power mode is an attempt to move to LPST mode. Setting this bit by the user is essential in order to assure proper response of the IPU to the assertion of the stop request from the CCM. 1 Next low power mode will be LPSR 0 Next low power mode is not LPSR
30 DI1_SRM_ CLOCK_ CHANGE_MODE	SRM clock change mode When the clock is going to be changed to any new ratio other then 1:1, 1:2. 1:4. The user needs to prepare an alternate set of DI setting in the SRM. This bit enable this mode. This bit is self cleared. 1 SRM clock change mode is enabled; the next clock change will be done by updating the DI settings from the SRM 0 SRM clock change mode is disabled.
29–23 DI1_CLK_ PERIOD_1	DI1_CLK period option 1. This parameter defines the period of the clock that the DI1 works with. This parameter contains integer part (bits [6:4]) and fractional part (bits [3:0]). Setting this value to 1.0 (default) means that the DI1 works on the fastest possible clock. Setting a value smaller than 1.0 is not allowed.

Table continues on the next page...

IPU_PM field descriptions (continued)

Field	Description
	The value to be programmed to the DI1_CLK_PERIOD_1 field is equal to: Fast_freq/Target_freq Where: Target_freq = The frequency that the DI clock works with Fast_freq = fastest possible clock that the DI can work with
22–16 DI1_CLK_ PERIOD_0	DI1_CLK period option 0. This parameter defines the period of the clock that the DI1 works with. This parameter contains integer part (bits [6:4]) and fractional part (bits [3:0]). Setting this value to 1.0 (default) means that the DI1 works on the fastest possible clock. Setting a value smaller than 1.0 is not allowed. The value to be programmed to the DI1_CLK_PERIOD_1 field is equal to: Fast_freq/Target_freq Where: Target_freq = The frequency that the DI clock works with Fast_freq = fastest possible clock that the DI can work with
15 CLCOK_MODE_ STAT	Clock mode status This is a read only bit indicating what is the current clock mode 1 current clock mode is 1 0 current clock mode is 0
14 DI0_SRM_ CLOCK_ CHANGE_MODE	SRM clock change mode When the clock is going to be changed to any new ratio other then 1:1, 1:2. 1:4. The user needs to prepare an alternate set of DI setting in the SRM. This bit enable this mode. This bit is self cleared. 1 SRM clock change mode is enabled; the next clock change will be done by updating the DI settings from the SRM 0 SRM clock change mode is disabled.
13–7 DI0_CLK_ PERIOD_1	DI0_CLK period option 1. This parameter defines the period of the clock that the DI0 works with. This parameter contains integer part (bits [6:4]) and fractional part (bits [3:0]). Setting this value to 1.0 (default) means that the DI0 works on the fastest possible clock. Setting a value smaller than 1.0 is not allowed. The value to be programmed to the DI0_CLK_PERIOD_1 field is equal to: Fast_freq/Target_freq Where: Target_freq = The frequency that the DI clock works with Fast_freq = fastest possible clock that the DI can work with
DI0_CLK_ PERIOD_0	DI0_CLK period option 0. This parameter defines the period of the clock that the DI0 works with. This parameter contains integer part (bits [6:4]) and fractional part (bits [3:0]).

Table continues on the next page...

IPU_PM field descriptions (continued)

Field	Description
	Setting this value to 1.0 (default) means that the DI0 works on the fastest possible clock. Setting a value smaller than 1.0 is not allowed. The value to be programmed to the DI0_CLK_PERIOD_1 field is equal to: Fast_freq/Target_freq Where: Target_freq = The frequency that the DI clock works with Fast_freq = fastest possible clock that the DI can work with

38.5.46 General Purpose Register (IPU_GPR)

The register contains general purpose bits.

Address: 260_0000h base + E4h offset = 260_00E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IPU_CH_BUF1_RDY1_CLR	IPU_CH_BUF1_RDY0_CLR	IPU_CH_BUF0_RDY1_CLR	IPU_CH_BUF0_RDY0_CLR	IPU_ALT_CH_BUF1_RDY1_CLR	IPU_ALT_CH_BUF1_RDY0_CLR	IPU_ALT_CH_BUF0_RDY1_CLR	IPU_ALT_CH_BUF0_RDY0_CLR	IPU_DI1_CLK_CHANGE_ACK_DIS	IPU_DI0_CLK_CHANGE_ACK_DIS	IPU_CH_BUF2_RDY1_CLR	IPU_CH_BUF2_RDY0_CLR	IPU_GPn			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IPU_GPn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_GPR field descriptions

Field	Description
31 IPU_CH_BUF1_RDY1_CLR	This bit defines the IPU_CH_BUF1_RDY1 properties. This register can be a write one to clear OR write one to set.

Table continues on the next page...

IPU_GPR field descriptions (continued)

Field	Description
	1 writing one to a bit of this register clears this bit; **IPU_CH_BUF1_RDY1** is w1c register 0 writing one to a bit of this register sets this bit **IPU_CH_BUF1_RDY1** is w1s register
30 IPU_CH_BUF1_RDY0_CLR	This bit defines the **IPU_CH_BUF1_RDY0** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_CH_BUF1_RDY0** is w1c register 0 writing one to a bit of this register sets this bit **IPU_CH_BUF1_RDY0** is w1s register
29 IPU_CH_BUF0_RDY1_CLR	This bit defines the **IPU_CH_BUF0_RDY1** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_CH_BUF0_RDY1** is w1c register 0 writing one to a bit of this register sets this bit **IPU_CH_BUF0_RDY1** is w1s register
28 IPU_CH_BUF0_RDY0_CLR	This bit defines the **IPU_CH_BUF0_RDY0** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_CH_BUF0_RDY0** is w1c register 0 writing one to a bit of this register sets this bit **IPU_CH_BUF0_RDY0** is w1s register
27 IPU_ALT_CH_BUF1_RDY1_CLR	This bit defines the **IPU_ALT_CH_BUF1_RDY1** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_ALT_CH_BUF1_RDY1** is w1c register 0 writing one to a bit of this register sets this bit **IPU_ALT_CH_BUF1_RDY1** is w1s register
26 IPU_ALT_CH_BUF1_RDY0_CLR	This bit defines the **IPU_ALT_CH_BUF1_RDY0** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_ALT_CH_BUF1_RDY0** is w1c register 0 writing one to a bit of this register sets this bit **IPU_ALT_CH_BUF1_RDY0** is w1s register
25 IPU_ALT_CH_BUF0_RDY1_CLR	This bit defines the **IPU_ALT_CH_BUF0_RDY1** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_ALT_CH_BUF0_RDY1** is w1c register 0 writing one to a bit of this register sets this bit **IPU_ALT_CH_BUF0_RDY1** is w1s register
24 IPU_ALT_CH_BUF0_RDY0_CLR	This bit defines the **IPU_ALT_CH_BUF0_RDY0** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_ALT_CH_BUF0_RDY0** is w1c register 0 writing one to a bit of this register sets this bit **IPU_ALT_CH_BUF0_RDY0** is w1s register
23 IPU_DI1_CLK_CHANGE_ACK_DIS	Disable DI1's clock change mechanism. 1 clock change mechanism is disabled. DI automatically acknowledges a clock change request 0 clock change mechanism is disabled. DI performs the clock change procedure
22 IPU_DI0_CLK_CHANGE_ACK_DIS	Disable DI0's clock change mechanism. 1 clock change mechanism is disabled. DI automatically acknowledges a clock change request 0 clock change mechanism is disabled. DI performs the clock change procedure
21 IPU_CH_BUF2_RDY1_CLR	This bit defines the **IPU_CH_BUF2_RDY1** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_CH_BUF2_RDY1** is w1c register 0 writing one to a bit of this register sets this bit **IPU_CH_BUF2_RDY1** is w1s register

Table continues on the next page...

IPU_GPR field descriptions (continued)

Field	Description
20 IPU_CH_BUF2_ RDY0_CLR	This bit defines the **IPU_CH_BUF2_RDY0** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_CH_BUF2_RDY0** is w1c register 0 writing one to a bit of this register sets this bit **IPU_CH_BUF2_RDY0** is w1s register
IPU_GPn	IPU General Purpose bit. *n* Indicates the corresponding DMA channel number. This bits are general Read/Write bits, reserved for future use

38.5.47 Channel Double Buffer Mode Select 0 Register (IPU_CH_DB_MODE_SEL0)

The register contains double buffer mode select control information for 32 IPU's DMA channels.

Address: 260_0000h base + 150h offset = 260_0150h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DMA_CH_DB_MODE_SEL_31	0	DMA_CH_DB_MODE_SEL_29	DMA_CH_DB_MODE_SEL_28	DMA_CH_DB_MODE_SEL_27	DMA_CH_DB_MODE_SEL_26	DMA_CH_DB_MODE_SEL_25	DMA_CH_DB_MODE_SEL_24	DMA_CH_DB_MODE_SEL_23	DMA_CH_DB_MODE_SEL_22	DMA_CH_DB_MODE_SEL_21	DMA_CH_DB_MODE_SEL_20	DMA_CH_DB_MODE_SEL_19	DMA_CH_DB_MODE_SEL_18	DMA_CH_DB_MODE_SEL_17	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DMA_CH_DB_MODE_SEL_15	DMA_CH_DB_MODE_SEL_14	DMA_CH_DB_MODE_SEL_13	DMA_CH_DB_MODE_SEL_12	DMA_CH_DB_MODE_SEL_11	DMA_CH_DB_MODE_SEL_10	DMA_CH_DB_MODE_SEL_9	DMA_CH_DB_MODE_SEL_8	0		DMA_CH_DB_MODE_SEL_5	0	DMA_CH_DB_MODE_SEL_3	DMA_CH_DB_MODE_SEL_2	DMA_CH_DB_MODE_SEL_1	DMA_CH_DB_MODE_SEL_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CH_DB_MODE_SEL0 field descriptions

Field	Description
31 DMA_CH_DB_ MODE_SEL_31	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_CH_DB_MODE_SEL0 field descriptions (continued)

Field	Description
	0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
30 Reserved	This read-only field is reserved and always has the value 0. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
29 DMA_CH_DB_ MODE_SEL_29	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
28 DMA_CH_DB_ MODE_SEL_28	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
27 DMA_CH_DB_ MODE_SEL_27	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
26 DMA_CH_DB_ MODE_SEL_26	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
25 DMA_CH_DB_ MODE_SEL_25	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
24 DMA_CH_DB_ MODE_SEL_24	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
23 DMA_CH_DB_ MODE_SEL_23	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
22 DMA_CH_DB_ MODE_SEL_22	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.

Table continues on the next page...

IPU_CH_DB_MODE_SEL0 field descriptions (continued)

Field	Description
21 DMA_CH_DB_ MODE_SEL_21	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
20 DMA_CH_DB_ MODE_SEL_20	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
19 DMA_CH_DB_ MODE_SEL_19	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
18 DMA_CH_DB_ MODE_SEL_18	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
17 DMA_CH_DB_ MODE_SEL_17	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
16 Reserved	This read-only field is reserved and always has the value 0. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
15 DMA_CH_DB_ MODE_SEL_15	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
14 DMA_CH_DB_ MODE_SEL_14	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
13 DMA_CH_DB_ MODE_SEL_13	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
12 DMA_CH_DB_ MODE_SEL_12	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_CH_DB_MODE_SEL0 field descriptions (continued)

Field	Description
	0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
11 DMA_CH_DB_ MODE_SEL_11	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
10 DMA_CH_DB_ MODE_SEL_10	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
9 DMA_CH_DB_ MODE_SEL_9	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
8 DMA_CH_DB_ MODE_SEL_8	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
7–6 Reserved	This read-only field is reserved and always has the value 0. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
5 DMA_CH_DB_ MODE_SEL_5	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
4 Reserved	This read-only field is reserved and always has the value 0. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
3 DMA_CH_DB_ MODE_SEL_3	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
2 DMA_CH_DB_ MODE_SEL_2	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.

Table continues on the next page...

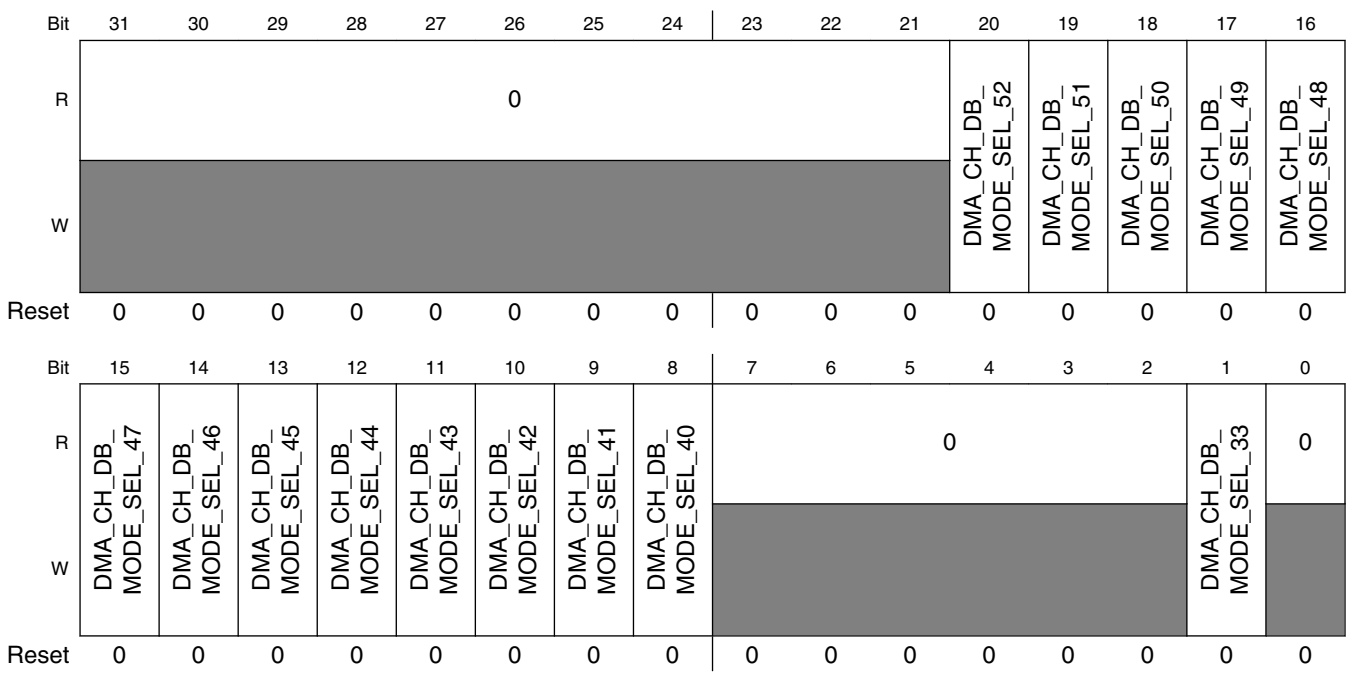
IPU_CH_DB_MODE_SEL0 field descriptions (continued)

Field	Description
1 DMA_CH_DB_ MODE_SEL_1	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
0 DMA_CH_DB_ MODE_SEL_0	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.

38.5.48 Channel Double Buffer Mode Select 1 Register (IPU_CH_DB_MODE_SEL1)

The register contains double buffer mode select control information for 32 IPU's DMA channels.

Address: 260_0000h base + 154h offset = 260_0154h



IPU_CH_DB_MODE_SEL1 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
20 DMA_CH_DB_ MODE_SEL_52	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
19 DMA_CH_DB_ MODE_SEL_51	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
18 DMA_CH_DB_ MODE_SEL_50	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
17 DMA_CH_DB_ MODE_SEL_49	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
16 DMA_CH_DB_ MODE_SEL_48	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
15 DMA_CH_DB_ MODE_SEL_47	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
14 DMA_CH_DB_ MODE_SEL_46	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
13 DMA_CH_DB_ MODE_SEL_45	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
12 DMA_CH_DB_ MODE_SEL_44	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

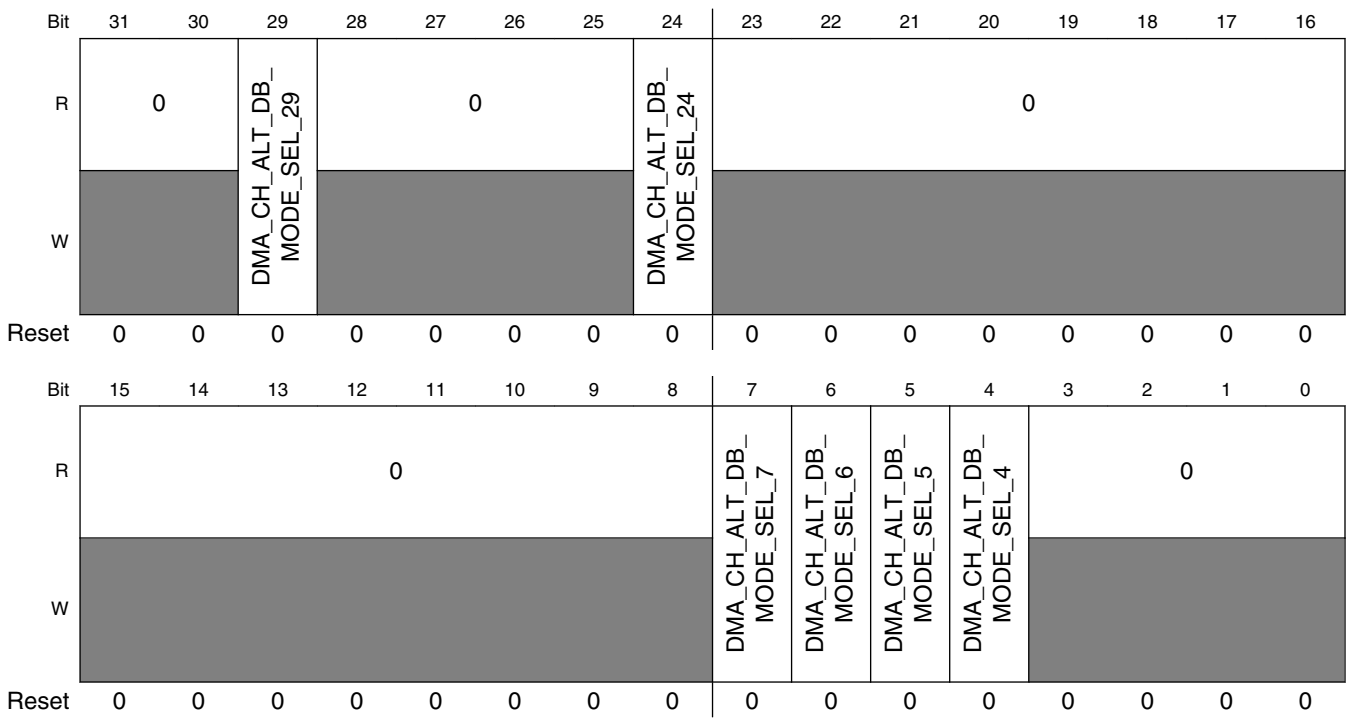
IPU_CH_DB_MODE_SEL1 field descriptions (continued)

Field	Description
	0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
11 DMA_CH_DB_ MODE_SEL_43	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
10 DMA_CH_DB_ MODE_SEL_42	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
9 DMA_CH_DB_ MODE_SEL_41	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
8 DMA_CH_DB_ MODE_SEL_40	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
7–2 Reserved	This read-only field is reserved and always has the value 0. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
1 DMA_CH_DB_ MODE_SEL_33	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
0 Reserved	This read-only field is reserved and always has the value 0. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.

38.5.49 Alternate Channel Double Buffer Mode Select 0 Register (IPU_ALT_CH_DB_MODE_SEL0)

The register contains double buffer mode select control information for 32 IPU's DMA channels.

Address: 260_0000h base + 168h offset = 260_0168h



IPU_ALT_CH_DB_MODE_SEL0 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value 0. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
29 DMA_CH_ALT_DB_MODE_SEL_29	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
28–25 Reserved	This read-only field is reserved and always has the value 0. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.

Table continues on the next page...

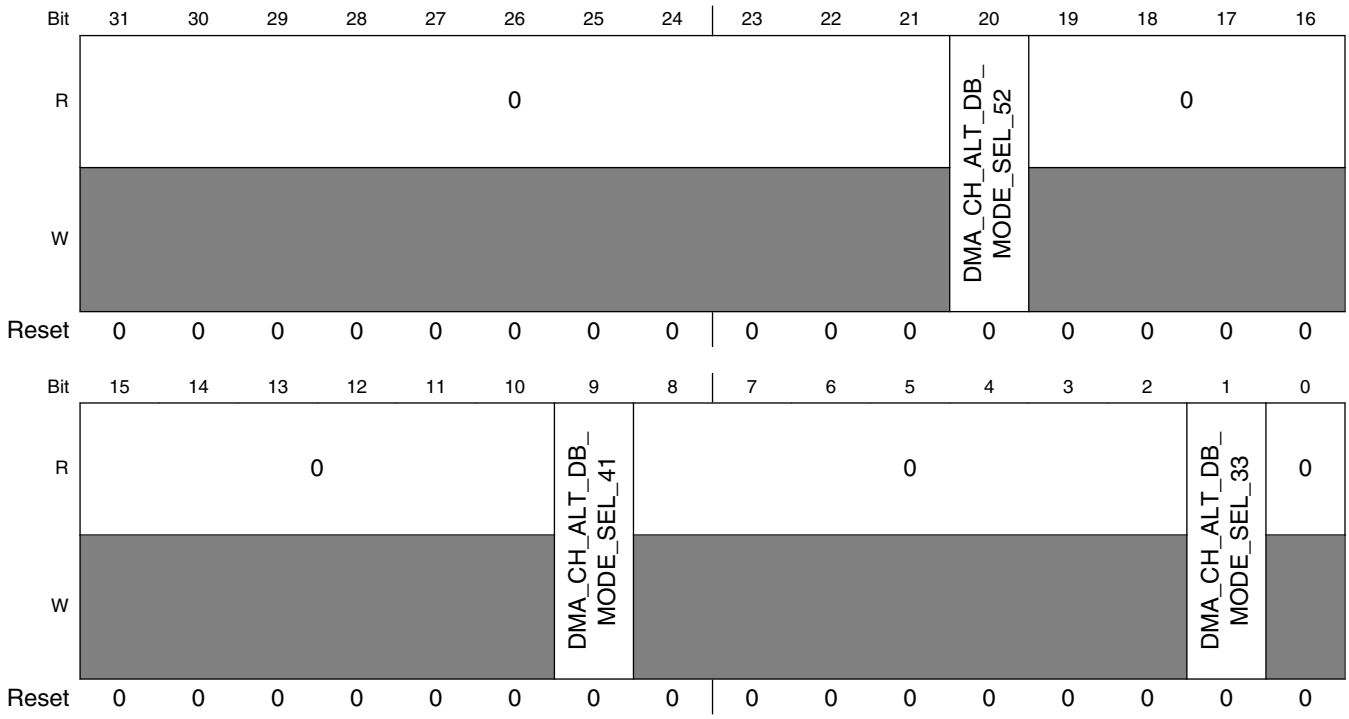
IPU_ALT_CH_DB_MODE_SEL0 field descriptions (continued)

Field	Description
24 DMA_CH_ALT_DB_MODE_SEL_24	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
23–8 Reserved	This read-only field is reserved and always has the value 0. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
7 DMA_CH_ALT_DB_MODE_SEL_7	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
6 DMA_CH_ALT_DB_MODE_SEL_6	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
5 DMA_CH_ALT_DB_MODE_SEL_5	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
4 DMA_CH_ALT_DB_MODE_SEL_4	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
Reserved	This read-only field is reserved and always has the value 0. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.

38.5.50 Alternate Channel Double Buffer Mode Select1 Register (IPU_ALT_CH_DB_MODE_SEL1)

The register contains double buffer mode select control information for 32 IPU's DMA channels.

Address: 260_0000h base + 16Ch offset = 260_016Ch



IPU_ALT_CH_DB_MODE_SEL1 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
20 DMA_CH_ALT_DB_MODE_SEL_52	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
19–10 Reserved	This read-only field is reserved and always has the value 0. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.

Table continues on the next page...

IPU_ALT_CH_DB_MODE_SEL1 field descriptions (continued)

Field	Description
9 DMA_CH_ALT_DB_MODE_SEL_41	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
8–2 Reserved	This read-only field is reserved and always has the value 0. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
1 DMA_CH_ALT_DB_MODE_SEL_33	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
0 Reserved	This read-only field is reserved and always has the value 0. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.

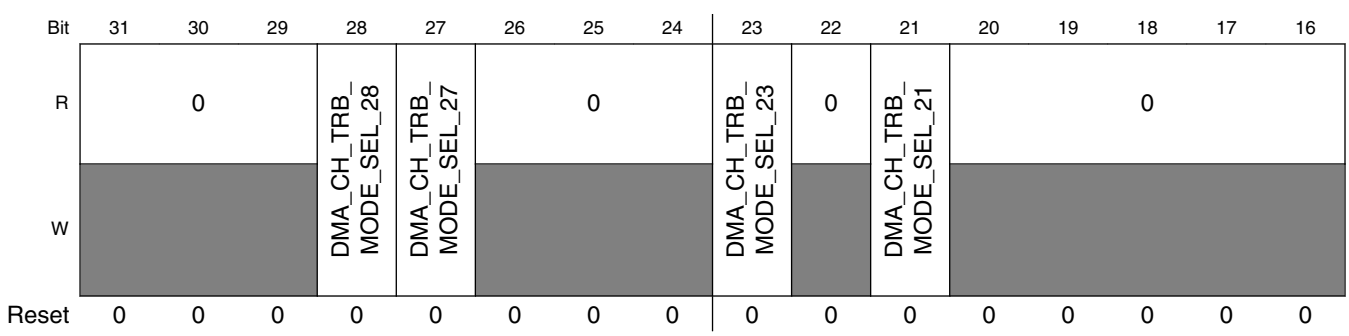
38.5.51 Alternate Channel Triple Buffer Mode Select 0 Register (IPU_ALT_CH_TRB_MODE_SEL0)

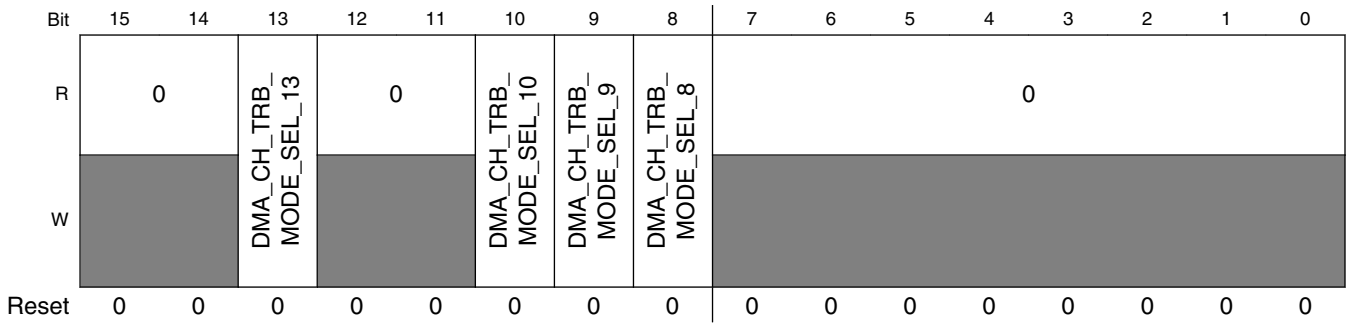
The register contains triple buffer mode select control information for 32 IPU's DMA channels.

When the channel is configured for triple buffer mode. The double buffer mode settings configured on the corresponding DB_MODE_SEL bit are overridden.

- Hide VPU_SUB_FRAME_SYNC for all versions
- Show VPU_SUB_FRAME_SYNC for IPUv3H version.
- The table below tagged with other settings (like IPU3M_only) should be hidden in IPUv3H version.
- This requires some sophisticated conditional tag settings

Address: 260_0000h base + 178h offset = 260_0178h





IPU_ALT_CH_TRB_MODE_SEL0 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
28 DMA_CH_TRB_MODE_SEL_28	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
27 DMA_CH_TRB_MODE_SEL_27	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
26–24 Reserved	This read-only field is reserved and always has the value 0. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
23 DMA_CH_TRB_MODE_SEL_23	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
22 Reserved	This read-only field is reserved and always has the value 0. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
21 DMA_CH_TRB_MODE_SEL_21	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
20–14 Reserved	This read-only field is reserved and always has the value 0. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.

Table continues on the next page...

IPU_ALT_CH_TRB_MODE_SEL0 field descriptions (continued)

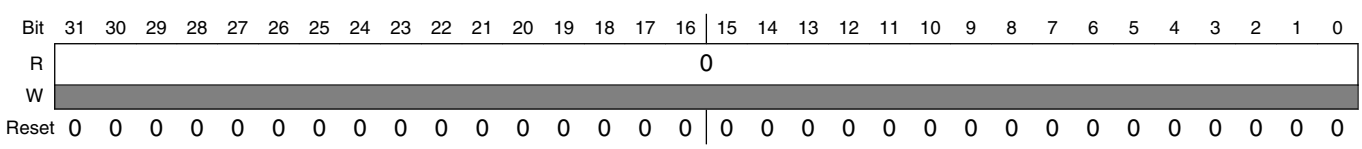
Field	Description
13 DMA_CH_TRB_MODE_SEL_13	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
12–11 Reserved	This read-only field is reserved and always has the value 0. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
10 DMA_CH_TRB_MODE_SEL_10	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
9 DMA_CH_TRB_MODE_SEL_9	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
8 DMA_CH_TRB_MODE_SEL_8	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
Reserved	This read-only field is reserved and always has the value 0. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.

38.5.52 Alternate Channel Triple Buffer Mode Select 1 Register (IPU_ALT_CH_TRB_MODE_SEL1)

The register contains triple buffer mode select control information for 32 IPU's DMA channels.

When the channel is configured for triple buffer mode. The double buffer mode settings configured on the corresponding DB_MODE_SEL bit are overridden.

Address: 260_0000h base + 17Ch offset = 260_017Ch



IPU_ALT_CH_TRB_MODE_SEL1 field descriptions

Field	Description
Reserved	This read-only field is reserved and always has the value 0. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.

38.5.52 Interrupt Status Register 1 (IPU_INT_STAT_1)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of EOF (end of frame) of DMA Channels interrupts [31:0] can be found in this register.

Address: 260_0000h base + 200h offset = 260_0200h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_EOF_31	0	IDMAC_EOF_29	IDMAC_EOF_28	IDMAC_EOF_27	IDMAC_EOF_26	IDMAC_EOF_25	IDMAC_EOF_24	IDMAC_EOF_23	IDMAC_EOF_22	IDMAC_EOF_21	IDMAC_EOF_20	IDMAC_EOF_19	IDMAC_EOF_18	IDMAC_EOF_17	0
W	w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_EOF_15	IDMAC_EOF_14	IDMAC_EOF_13	IDMAC_EOF_12	IDMAC_EOF_11	IDMAC_EOF_10	IDMAC_EOF_9	IDMAC_EOF_8	0		IDMAC_EOF_5	0	IDMAC_EOF_3	IDMAC_EOF_2	IDMAC_EOF_1	IDMAC_EOF_0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c			w1c		w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_1 field descriptions

Field	Description
31 IDMAC_EOF_31	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
30 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
29 IDMAC_EOF_29	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
28 IDMAC_EOF_28	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
27 IDMAC_EOF_27	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
26 IDMAC_EOF_26	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
25 IDMAC_EOF_25	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
24 IDMAC_EOF_24	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
23 IDMAC_EOF_23	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
22 IDMAC_EOF_22	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_1 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
21 IDMAC_EOF_21	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_EOF_20	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_EOF_19	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_EOF_18	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_EOF_17	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_EOF_15	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_EOF_14	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_EOF_13	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_1 field descriptions (continued)

Field	Description
12 IDMAC_EOF_12	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_EOF_11	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
10 IDMAC_EOF_10	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_EOF_9	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_EOF_8	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–6 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
5 IDMAC_EOF_5	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
4 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
3 IDMAC_EOF_3	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
2 IDMAC_EOF_2	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_1 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_EOF_1	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 IDMAC_EOF_0	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

38.5.53 Interrupt Status Register2 (IPU_INT_STAT_2)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of EOF (end of frame) of DMA Channels interrupts [63:32] can be found in this register.

Address: 260_0000h base + 204h offset = 260_0204h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											IDMAC_EOF_52	IDMAC_EOF_51	IDMAC_EOF_50	IDMAC_EOF_49	IDMAC_EOF_48
W												w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_EOF_47	IDMAC_EOF_46	IDMAC_EOF_45	IDMAC_EOF_44	IDMAC_EOF_43	IDMAC_EOF_42	IDMAC_EOF_41	IDMAC_EOF_40	0						IDMAC_EOF_33	0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c							w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_2 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0.
	0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_2 field descriptions (continued)

Field	Description
20 IDMAC_EOF_52	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_EOF_51	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_EOF_50	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_EOF_49	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 IDMAC_EOF_48	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_EOF_47	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_EOF_46	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_EOF_45	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_EOF_44	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_2 field descriptions (continued)

Field	Description
11 IDMAC_EOF_43	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
10 IDMAC_EOF_42	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_EOF_41	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_EOF_40	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–2 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_EOF_33	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.

38.5.54 Interrupt Status Register 3 (IPU_INT_STAT_3)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of NFACK (New Frame Ack) of DMA Channels interrupts [31:0] can be found in this register.

Address: 260_0000h base + 208h offset = 260_0208h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_NFACK_31	0	IDMAC_NFACK_29	IDMAC_NFACK_28	IDMAC_NFACK_27	IDMAC_NFACK_26	IDMAC_NFACK_25	IDMAC_NFACK_24	IDMAC_NFACK_23	IDMAC_NFACK_22	IDMAC_NFACK_21	IDMAC_NFACK_20	IDMAC_NFACK_19	IDMAC_NFACK_18	IDMAC_NFACK_17	0
W	w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_NFACK_15	IDMAC_NFACK_14	IDMAC_NFACK_13	IDMAC_NFACK_12	IDMAC_NFACK_11	IDMAC_NFACK_10	IDMAC_NFACK_9	IDMAC_NFACK_8	0		IDMAC_NFACK_5	0	IDMAC_NFACK_3	IDMAC_NFACK_2	IDMAC_NFACK_1	IDMAC_NFACK_0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c			w1c		w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_3 field descriptions

Field	Description
31 IDMAC_NFACK_31	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
30 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
29 IDMAC_NFACK_29	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_3 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
28 IDMAC_NFACK_28	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
27 IDMAC_NFACK_27	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
26 IDMAC_NFACK_26	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
25 IDMAC_NFACK_25	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
24 IDMAC_NFACK_24	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
23 IDMAC_NFACK_23	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
22 IDMAC_NFACK_22	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
21 IDMAC_NFACK_21	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_NFACK_20	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_3 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_NFACK_19	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_NFACK_18	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_NFACK_17	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_NFACK_15	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_NFACK_14	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_NFACK_13	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_NFACK_12	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_NFACK_11	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_3 field descriptions (continued)

Field	Description
10 IDMAC_NFACK_10	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_NFACK_9	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_NFACK_8	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–6 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
5 IDMAC_NFACK_5	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
4 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
3 IDMAC_NFACK_3	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
2 IDMAC_NFACK_2	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_NFACK_1	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 IDMAC_NFACK_0	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_3 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.

38.5.55 Interrupt Status Register 4 (IPU_INT_STAT_4)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of NFACK (New Frame Ack) of DMA Channels interrupts [63:32] can be found in this register.

Address: 260_0000h base + 20Ch offset = 260_020Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											IDMAC_NFACK_52	IDMAC_NFACK_51	IDMAC_NFACK_50	IDMAC_NFACK_49	IDMAC_NFACK_48
W												w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_NFACK_47	IDMAC_NFACK_46	IDMAC_NFACK_45	IDMAC_NFACK_44	IDMAC_NFACK_43	IDMAC_NFACK_42	IDMAC_NFACK_41	IDMAC_NFACK_40	0						IDMAC_NFACK_33	0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c							w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_4 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0.
	0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_4 field descriptions (continued)

Field	Description
20 IDMAC_NFACK_52	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_NFACK_51	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_NFACK_50	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_NFACK_49	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 IDMAC_NFACK_48	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_NFACK_47	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_NFACK_46	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_NFACK_45	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_NFACK_44	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_4 field descriptions (continued)

Field	Description
11 IDMAC_NFACK_43	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
10 IDMAC_NFACK_42	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_NFACK_41	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_NFACK_40	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–2 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_NFACK_33	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.

38.5.56 Interrupt Status Register 5 (IPU_INT_STAT_5)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of the New-frame before end-of-frame indication (NFB4EOF_ERR) of DMA Channels interrupts [31:0] can be found in this register.

Address: 260_0000h base + 210h offset = 260_0210h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_NFB4EOF_ERR_31	0	IDMAC_NFB4EOF_ERR_29	IDMAC_NFB4EOF_ERR_28	IDMAC_NFB4EOF_ERR_27	IDMAC_NFB4EOF_ERR_26	IDMAC_NFB4EOF_ERR_25	IDMAC_NFB4EOF_ERR_24	IDMAC_NFB4EOF_ERR_23	IDMAC_NFB4EOF_ERR_22	IDMAC_NFB4EOF_ERR_21	IDMAC_NFB4EOF_ERR_20	IDMAC_NFB4EOF_ERR_19	IDMAC_NFB4EOF_ERR_18	IDMAC_NFB4EOF_ERR_17	0
W	w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_NFB4EOF_ERR_15	IDMAC_NFB4EOF_ERR_14	IDMAC_NFB4EOF_ERR_13	IDMAC_NFB4EOF_ERR_12	IDMAC_NFB4EOF_ERR_11	IDMAC_NFB4EOF_ERR_10	IDMAC_NFB4EOF_ERR_9	IDMAC_NFB4EOF_ERR_8	0		IDMAC_NFB4EOF_ERR_5	0	IDMAC_NFB4EOF_ERR_3	IDMAC_NFB4EOF_ERR_2	IDMAC_NFB4EOF_ERR_1	IDMAC_NFB4EOF_ERR_0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c			w1c		w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_5 field descriptions

Field	Description
31 IDMAC_ NFB4EOF_ERR_ 31	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
30 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
29 IDMAC_ NFB4EOF_ERR_ 29	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
28 IDMAC_ NFB4EOF_ERR_ 28	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
27 IDMAC_ NFB4EOF_ERR_ 27	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_5 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
26 IDMAC_ NFB4EOF_ERR_ 26	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
25 IDMAC_ NFB4EOF_ERR_ 25	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
24 IDMAC_ NFB4EOF_ERR_ 24	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
23 IDMAC_ NFB4EOF_ERR_ 23	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
22 IDMAC_ NFB4EOF_ERR_ 22	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
21 IDMAC_ NFB4EOF_ERR_ 21	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_ NFB4EOF_ERR_ 20	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_ NFB4EOF_ERR_ 19	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_5 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_ NFB4EOF_ERR_ 18	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_ NFB4EOF_ERR_ 17	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_ NFB4EOF_ERR_ 15	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_ NFB4EOF_ERR_ 14	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_ NFB4EOF_ERR_ 13	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_ NFB4EOF_ERR_ 12	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_ NFB4EOF_ERR_ 11	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_5 field descriptions (continued)

Field	Description
10 IDMAC_ NFB4EOF_ERR_ 10	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_ NFB4EOF_ERR_ 9	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_ NFB4EOF_ERR_ 8	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–6 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
5 IDMAC_ NFB4EOF_ERR_ 5	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
4 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
3 IDMAC_ NFB4EOF_ERR_ 3	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
2 IDMAC_ NFB4EOF_ERR_ 2	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_ NFB4EOF_ERR_ 1	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_5 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
0 IDMAC_ NFB4EOF_ERR_ 0	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

38.5.57 Interrupt Status Register 6 (IPU_INT_STAT_6)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of the New-frame before end-of-frame indication (NFB4EOF_ERR) of DMA Channels interrupts [63:32] can be found in this register.

Address: 260_0000h base + 214h offset = 260_0214h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											IDMAC_NFB4EOF_ERR_52	IDMAC_NFB4EOF_ERR_51	IDMAC_NFB4EOF_ERR_50	IDMAC_NFB4EOF_ERR_49	IDMAC_NFB4EOF_ERR_48
W												w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_NFB4EOF_ERR_47	IDMAC_NFB4EOF_ERR_46	IDMAC_NFB4EOF_ERR_45	IDMAC_NFB4EOF_ERR_44	IDMAC_NFB4EOF_ERR_43	IDMAC_NFB4EOF_ERR_42	IDMAC_NFB4EOF_ERR_41	IDMAC_NFB4EOF_ERR_40	0						IDMAC_NFB4EOF_ERR_33	0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c							w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_6 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_ NFB4EOF_ERR_ 52	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_ NFB4EOF_ERR_ 51	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_ NFB4EOF_ERR_ 50	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_ NFB4EOF_ERR_ 49	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_6 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
16 IDMAC_ NFB4EOF_ERR_ 48	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_ NFB4EOF_ERR_ 47	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_ NFB4EOF_ERR_ 46	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_ NFB4EOF_ERR_ 45	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_ NFB4EOF_ERR_ 44	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_ NFB4EOF_ERR_ 43	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
10 IDMAC_ NFB4EOF_ERR_ 42	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_ NFB4EOF_ERR_ 41	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_6 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_ NFB4EOF_ERR_ 40	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–2 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_ NFB4EOF_ERR_ 33	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.

38.5.58 Interrupt Status Register7 1 (IPU_INT_STAT_7)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of the End-of-Scroll indication (EOS) of DMA Channels interrupts [31:0] can be found in this register.

Address: 260_0000h base + 218h offset = 260_0218h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_EOS_EN_31	0	IDMAC_EOS_EN_29	IDMAC_EOS_EN_28	IDMAC_EOS_EN_27	IDMAC_EOS_EN_26	IDMAC_EOS_EN_25	IDMAC_EOS_EN_24	IDMAC_EOS_EN_23	0			IDMAC_EOS_EN_19	0		
W	w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c				w1c			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_7 field descriptions

Field	Description
31 IDMAC_EOS_EN_31	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
30 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
29 IDMAC_EOS_EN_29	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
28 IDMAC_EOS_EN_28	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
27 IDMAC_EOS_EN_27	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
26 IDMAC_EOS_EN_26	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
25 IDMAC_EOS_EN_25	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
24 IDMAC_EOS_EN_24	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_7 field descriptions (continued)

Field	Description
23 IDMAC_EOS_EN_23	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
22–20 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_EOS_EN_19	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.

38.5.59 Interrupt Status Register 8 (IPU_INT_STAT_8)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. All the status bits of the End of Scroll indication (EOS) of DMA Channels interrupts [63:32] can be found in this register.

Address: 260_0000h base + 21Ch offset = 260_021Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											IDMAC_EOS_EN_52	IDMAC_EOS_EN_51	0		
W												w1c	w1c			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			IDMAC_EOS_EN_44	IDMAC_EOS_EN_43	IDMAC_EOS_EN_42	IDMAC_EOS_EN_41	0							IDMAC_EOS_EN_33	0
W				w1c	w1c	w1c	w1c								w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_8 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_INT_STAT_8 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_EOS_EN_52	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_EOS_EN_51	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
18–13 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_EOS_EN_44	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_EOS_EN_43	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
10 IDMAC_EOS_EN_42	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_EOS_EN_41	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8–2 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_8 field descriptions (continued)

Field	Description
1 IDMAC_EOS_EN_33	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.

38.5.60 Interrupt Status Register 9 (IPU_INT_STAT_9)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. This register holds the error interrupt indications coming from different modules within All the bits in this register are write one to clear.

Address: 260_0000h base + 220h offset = 260_0220h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CSI1_PUPE	CSI0_PUPE	0	IC_VF_BUF_OVF	IC_ENC_BUF_OVF	IC_BAYER_BUF_OVF	0									
W	w1c	w1c		w1c	w1c	w1c										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															VDI_FIFO1_OVF
W																w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_9 field descriptions

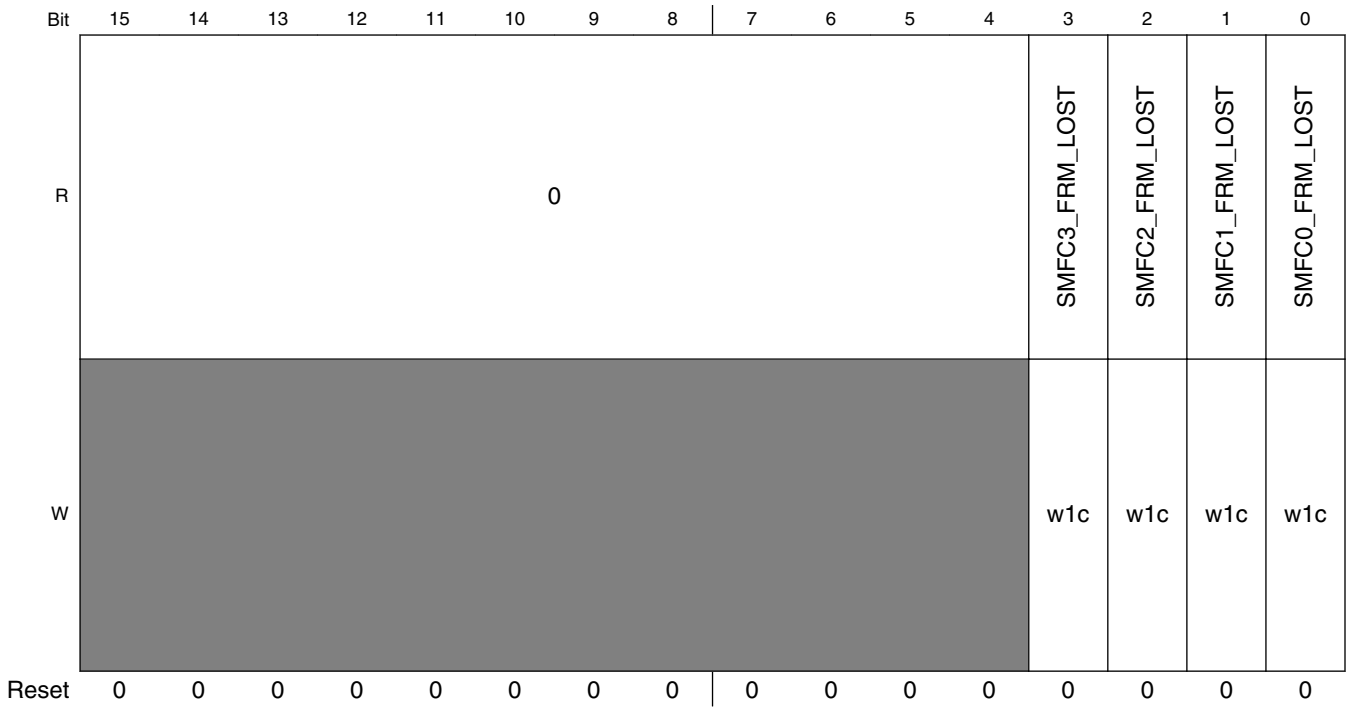
Field	Description
31 CSI1_PUPE	CSI1_PUPE - CSI1 parameters update error interrupt. This bit indicates on an interrupt that is a result of an error generated by the CSI1. The error is generated in case where new frame arrived from the CSI1 before the completion of the CSI1's parameters update by the SRM 0 Interrupt is cleared. 1 Interrupt is requested.
30 CSI0_PUPE	CSI0_PUPE - CSI0 parameters update error interrupt. This bit indicates on an interrupt that is a result of an error generated by the CSI0. The error is generated in case where new frame arrived from the CSI0 before the completion of the CSI0's parameters update by the SRM 0 Interrupt is cleared. 1 Interrupt is requested.
29 Reserved	This read-only field is reserved and always has the value 0.
28 IC_VF_BUF_OVF	This bit indicates on an interrupt that is a result of the IC Buffer overflow for view finder coming from the IC. The user needs to write 1 to this bit in order to clear it. 0 Interrupt is cleared. 1 Interrupt is requested.
27 IC_ENC_BUF_OVF	This bit indicates on an interrupt that is a result of the IC Buffer overflow for encoding coming from the IC. The user needs to write 1 to this bit in order to clear it. 0 Interrupt is cleared. 1 Interrupt is requested.
26 IC_BAYER_BUF_OVF	This bit indicates on an interrupt that is a result of the IC Buffer overflow for Bayer coming from the IC. The user needs to write 1 to this bit in order to clear it. 0 Interrupt is cleared. 1 Interrupt is requested.
25–1 Reserved	This read-only field is reserved and always has the value 0.
0 VDI_FIFO1_OVF	FIFO1 overflow Interrupt1 The VDIC generate FIFO1 overflow interrupt1 when write pointer of FIFO1 overrun read pointer. 0 Interrupt is cleared. 1 Interrupt is requested.

38.5.61 Interrupt Status Register 10 (IPU_INT_STAT_10)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. This register holds error interrupt indications coming from different modules within All the bits in this register are write one to clear.

Address: 260_0000h base + 224h offset = 260_0224h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	AXIR_ERR	AXIW_ERR	NON_PRIVILEGED_ACC_ERR	0	IC_BAYER_FRM_LOST_ERR	IC_ENC_FRM_LOST_ERR	IC_VF_FRM_LOST_ERR	0	DI1_TIME_OUT_ERR	DI0_TIME_OUT_ERR	DI1_SYNC_DISP_ERR	DI0_SYNC_DISP_ERR	DC_TEARING_ERR_6	DC_TEARING_ERR_2	DC_TEARING_ERR_1
W		w1c	w1c	w1c		w1c	w1c	w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



IPU_INT_STAT_10 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30 AXIR_ERR	This bit indicates on an interrupt that is a result of AXI read access resulted with error response. The user needs to write 1 to this bit in order to clear it. 0 Interrupt is cleared. 1 Interrupt is requested.
29 AXIW_ERR	This bit indicates on an interrupt that is a result of AXI write access resulted with error response. The user needs to write 1 to this bit in order to clear it. 0 Interrupt is cleared. 1 Interrupt is requested.
28 NON_ PRIVILEGED_ ACC_ERR	Non Privileged Access Error interrupt. This bit indicates on an interrupt that is a result of access the CPMEM or the DP memory in user mode 0 Interrupt is cleared. 1 Interrupt is requested.
27 Reserved	This read-only field is reserved and always has the value 0.
26 IC_BAYER_ FRM_LOST_ ERR	This bit indicates on an interrupt that is a result of IC's Bayer frame lost. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IC_ENC_FRM_ LOST_ERR	This bit indicates on an interrupt that is a result of IC's encoding frame lost.

Table continues on the next page...

IPU_INT_STAT_10 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
24 IC_VF_FRM_LOST_ERR	This bit indicates on an interrupt that is a result of IC's view finder frame lost. 0 Interrupt is disabled. 1 Interrupt is enabled.
23 Reserved	This read-only field is reserved and always has the value 0.
22 DI1_TIME_OUT_ERR	DI1 time out error interrupt This bit indicates on the interrupt that is a result of a time out error during a read access via DI1
21 DI0_TIME_OUT_ERR	DI0 time out error interrupt This bit indicates on the interrupt that is a result of a time out error during a read access via DI0
20 DI1_SYNC_DISP_ERR	DI1 Synchronous display error interrupt This bit indicates on the interrupt that is a result of an error during access to a synchronous display via DI1
19 DI0_SYNC_DISP_ERR	DI0 Synchronous display error interrupt This bit indicates on the interrupt that is a result of an error during access to a synchronous display via DI0
18 DC_TEARING_ERR_6	Tearing Error #6 interrupt This bit indicates on the interrupt that is a result of tearing error while the anti tearing mechanism is activated for DC channel 6
17 DC_TEARING_ERR_2	Tearing Error #2 interrupt This bit indicates on the interrupt that is a result of tearing error while the anti tearing mechanism is activated for DC channel 2
16 DC_TEARING_ERR_1	Tearing Error #1 interrupt This bit indicates on the interrupt that is a result of tearing error while the anti tearing mechanism is activated for DC channel 1
15–4 Reserved	This read-only field is reserved and always has the value 0.
3 SMFC3_FRM_LOST	Frame Lost of SMFC channel 3 interrupt. This bit indicates on an interrupt that is a result of a result of a Frame Lost of SMFC channel 3 0 Interrupt is cleared. 1 Interrupt is requested.
2 SMFC2_FRM_LOST	Frame Lost of SMFC channel 2 interrupt. This bit indicates on an interrupt that is a result of a result of a Frame Lost of SMFC channel 2 0 Interrupt is cleared. 1 Interrupt is requested.
1 SMFC1_FRM_LOST	Frame Lost of SMFC channel 1 interrupt. This bit indicates on an interrupt that is a result of a result of a Frame Lost of SMFC channel 1 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_10 field descriptions (continued)

Field	Description
0 SMFC0_FRM_LOST	Frame Lost of SMFC channel 0 interrupt. This bit indicates on an interrupt that is a result of a result of a Frame Lost of SMFC channel 0 0 Interrupt is cleared. 1 Interrupt is requested.

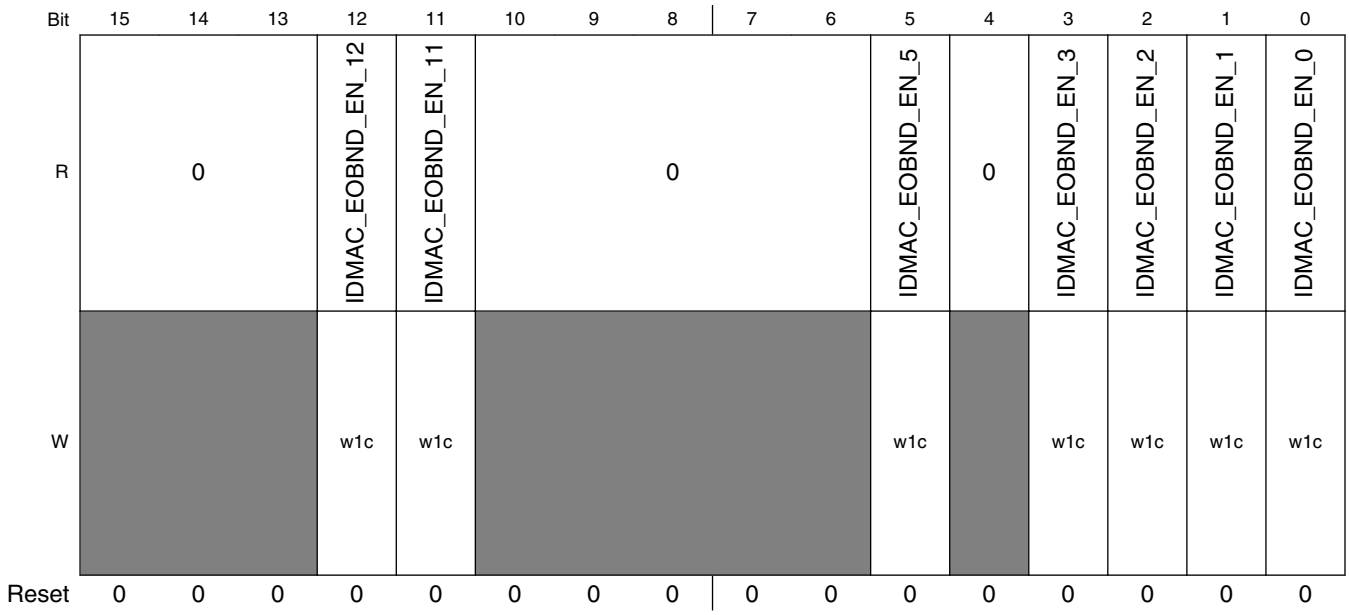
38.5.62 Interrupt Status Register 11 (IPU_INT_STAT_11)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of the end-of-band indication (EOBND) of DMA Channels interrupts [31:0] can be found in this register.

- Hide VDOA_SYNC for all versions
- Show VDOA_SYNC for IPUv3H version.
- The table below tagged with other settings (like IPU3M_only) should be hidden in IPUv3H version.

Address: 260_0000h base + 228h offset = 260_0228h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					IDMAC_EOBND_EN_26	IDMAC_EOBND_EN_25	0		IDMAC_EOBND_EN_22	IDMAC_EOBND_EN_21	IDMAC_EOBND_EN_20	0			
W						w1c	w1c			w1c	w1c	w1c				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



IPU_INT_STAT_11 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
26 IDMAC_EOBND_EN_26	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
25 IDMAC_EOBND_EN_25	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
24–23 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
22 IDMAC_EOBND_EN_22	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
21 IDMAC_EOBND_EN_21	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n.

Table continues on the next page...

IPU_INT_STAT_11 field descriptions (continued)

Field	Description
	n Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_EOBND_EN_20	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
19–13 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_EOBND_EN_12	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_EOBND_EN_11	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
10–6 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
5 IDMAC_EOBND_EN_5	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
4 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
3 IDMAC_EOBND_EN_3	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_11 field descriptions (continued)

Field	Description
2 IDMAC_EOBND_EN_2	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_EOBND_EN_1	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 IDMAC_EOBND_EN_0	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

38.5.63 Interrupt Status Register 12 (IPU_INT_STAT_12)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of the end-of-band indication (EOBND) of DMA Channels interrupts [63:32] can be found in this register.

Address: 260_0000h base + 22Ch offset = 260_022Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													IDMAC_EOBND_EN_50	IDMAC_EOBND_EN_49	IDMAC_EOBND_EN_48
W														w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_EOBND_EN_47	IDMAC_EOBND_EN_46	IDMAC_EOBND_EN_45	0												
W	w1c	w1c	w1c													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_12 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_EOBND_EN_50	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_EOBND_EN_49	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 IDMAC_EOBND_EN_48	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_EOBND_EN_47	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_EOBND_EN_46	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_EOBND_EN_45	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.

38.5.64 Interrupt Status Register 13 (IPU_INT_STAT_13)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of the Threshold crossing indication (TH) of DMA Channels interrupts [31:0] can be found in this register.

Address: 260_0000h base + 230h offset = 260_0230h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_TH_31	0	IDMAC_TH_29	IDMAC_TH_28	IDMAC_TH_27	IDMAC_TH_26	IDMAC_TH_25	IDMAC_TH_24	IDMAC_TH_23	IDMAC_TH_22	IDMAC_TH_21	IDMAC_TH_20	IDMAC_TH_19	IDMAC_TH_18	IDMAC_TH_17	0
W	w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_TH_15	IDMAC_TH_14	IDMAC_TH_13	IDMAC_TH_12	IDMAC_TH_11	IDMAC_TH_10	IDMAC_TH_9	IDMAC_TH_8	0		IDMAC_TH_5	0	IDMAC_TH_3	IDMAC_TH_2	IDMAC_TH_1	IDMAC_TH_0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c			w1c		w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_13 field descriptions

Field	Description
31 IDMAC_TH_31	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_13 field descriptions (continued)

Field	Description
30 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
29 IDMAC_TH_29	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
28 IDMAC_TH_28	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
27 IDMAC_TH_27	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
26 IDMAC_TH_26	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
25 IDMAC_TH_25	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
24 IDMAC_TH_24	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
23 IDMAC_TH_23	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
22 IDMAC_TH_22	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n.

Table continues on the next page...

IPU_INT_STAT_13 field descriptions (continued)

Field	Description
	n Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
21 IDMAC_TH_21	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_TH_20	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_TH_19	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_TH_18	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_TH_17	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_TH_15	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_TH_14	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_13 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_TH_13	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_TH_12	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_TH_11	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
10 IDMAC_TH_10	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_TH_9	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_TH_8	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–6 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
5 IDMAC_TH_5	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

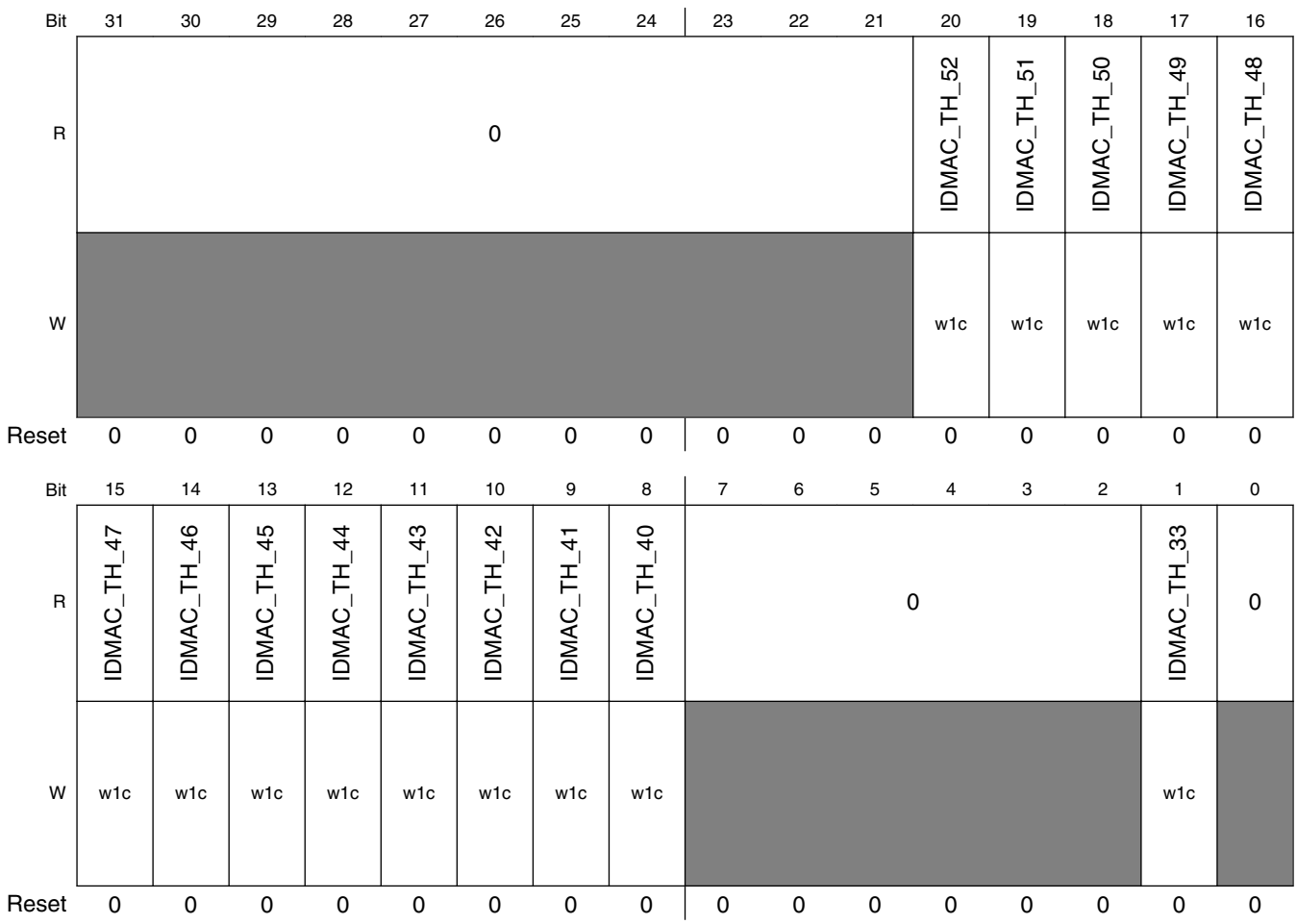
IPU_INT_STAT_13 field descriptions (continued)

Field	Description
4 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
3 IDMAC_TH_3	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
2 IDMAC_TH_2	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_TH_1	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 IDMAC_TH_0	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

38.5.65 Interrupt Status Register 14 (IPU_INT_STAT_14)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of the Threshold crossing indication (TH) of DMA Channels interrupts [63:32] can be found in this register.

Address: 260_0000h base + 234h offset = 260_0234h



IPU_INT_STAT_14 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0.
20 IDMAC_TH_52	0 Interrupt is cleared. 1 Interrupt is requested. Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_14 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_TH_51	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_TH_50	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_TH_49	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 IDMAC_TH_48	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_TH_47	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_TH_46	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_TH_45	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_TH_44	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_14 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_TH_43	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
10 IDMAC_TH_42	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_TH_41	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_TH_40	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–2 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_TH_33	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 Reserved	This read-only field is reserved and always has the value 0. 0 Interrupt is cleared. 1 Interrupt is requested.

38.5.66 Interrupt Status Register 15 (IPU_INT_STAT_15)

IPU status registers are not stored in the SRM during power gating mode.IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of general purpose interrupts can be found in this register.

Address: 260_0000h base + 238h offset = 260_0238h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DI1_CNT_EN_PRE_8	DI1_CNT_EN_PRE_3	DI1_DISP_CLK_EN_PRE	DI0_CNT_EN_PRE_10	DI0_CNT_EN_PRE_9	DI0_CNT_EN_PRE_8	DI0_CNT_EN_PRE_7	DI0_CNT_EN_PRE_6	DI0_CNT_EN_PRE_5	DI0_CNT_EN_PRE_4	DI0_CNT_EN_PRE_3	DI0_CNT_EN_PRE_2	DI0_CNT_EN_PRE_1	DI0_CNT_EN_PRE_0	DC_ASYNC_STOP	DC_DP_START
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DI_VSYNC_PRE_1	DI_VSYNC_PRE_0	DC_FC_6	DC_FC_4	DC_FC_3	DC_FC_2	DC_FC_1	DC_FC_0	DP_ASF_BRAKE	DP_SF_BRAKE	DP_ASF_END	DP_ASF_START	DP_SF_END	DP_SF_START	SNOOPING2_INT	SNOOPING1_INT
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_15 field descriptions

Field	Description
31 DI1_CNT_EN_PRE_8	This bit indicates on the interrupt that is a result of a trigger generated by counter #8 of DI1 0 Interrupt is cleared. 1 Interrupt is requested.
30 DI1_CNT_EN_PRE_3	This bit indicates on the interrupt that is a result of a trigger generated by counter #3 of DI1 0 Interrupt is cleared. 1 Interrupt is requested.
29 DI1_DISP_CLK_EN_PRE	**DI1_DISP_CLK_EN_PRE** 0 Interrupt is cleared. 1 Interrupt is requested.
28 DI0_CNT_EN_PRE_10	This bit indicates on the interrupt that is a result of a trigger generated by counter #10 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
27 DI0_CNT_EN_PRE_9	This bit indicates on the interrupt that is a result of a trigger generated by counter #9 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
26 DI0_CNT_EN_PRE_8	This bit indicates on the interrupt that is a result of a trigger generated by counter #8 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
25 DI0_CNT_EN_PRE_7	This bit indicates on the interrupt that is a result of a trigger generated by counter #7 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
24 DI0_CNT_EN_PRE_6	This bit indicates on the interrupt that is a result of a trigger generated by counter #6 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
23 DI0_CNT_EN_PRE_5	This bit indicates on the interrupt that is a result of a trigger generated by counter #5 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
22 DI0_CNT_EN_PRE_4	This bit indicates on the interrupt that is a result of a trigger generated by counter #4 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
21 DI0_CNT_EN_PRE_3	This bit indicates on the interrupt that is a result of a trigger generated by counter #3 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
20 DI0_CNT_EN_PRE_2	This bit indicates on the interrupt that is a result of a trigger generated by counter #2 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_15 field descriptions (continued)

Field	Description
19 DI0_CNT_EN_PRE_1	This bit indicates on the interrupt that is a result of a trigger generated by counter #1 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
18 DI0_CNT_EN_PRE_0	This bit indicates on the interrupt that is a result of a trigger generated by counter #0 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
17 DC_ASYNC_STOP	This bit indicates on an interrupt asserted anytime the DP stops an async flow and moves to a sync flow 0 Interrupt is cleared. 1 Interrupt is requested.
16 DC_DP_START	This bit indicates on an interrupt asserted anytime the DP start a new sync or async flow or when an async flow is interrupted by a sync flow 0 Interrupt is cleared. 1 Interrupt is requested.
15 DI_VSYNC_PRE_1	DI1 interrupt indicating of a VSYNC signal asserted 2 rows before the VSYNC sent to the display 0 Interrupt is cleared. 1 Interrupt is requested.
14 DI_VSYNC_PRE_0	DI0 interrupt indicating of a VSYNC signal asserted 2 rows before the VSYNC sent to the display 0 Interrupt is cleared. 1 Interrupt is requested.
13 DC_FC_6	DC Frame Complete on channel #6 interrupt indication 0 Interrupt is cleared. 1 Interrupt is requested.
12 DC_FC_4	DC Frame Complete on channel #4 interrupt indication 0 Interrupt is cleared. 1 Interrupt is requested.
11 DC_FC_3	DC Frame Complete on channel #3 interrupt indication 0 Interrupt is cleared. 1 Interrupt is requested.
10 DC_FC_2	DC Frame Complete on channel #2 interrupt indication 0 Interrupt is cleared. 1 Interrupt is requested.
9 DC_FC_1	DC Frame Complete on channel #1 interrupt indication 0 Interrupt is cleared. 1 Interrupt is requested.
8 DC_FC_0	DC Frame Complete on channel #0 interrupt indication 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_15 field descriptions (continued)

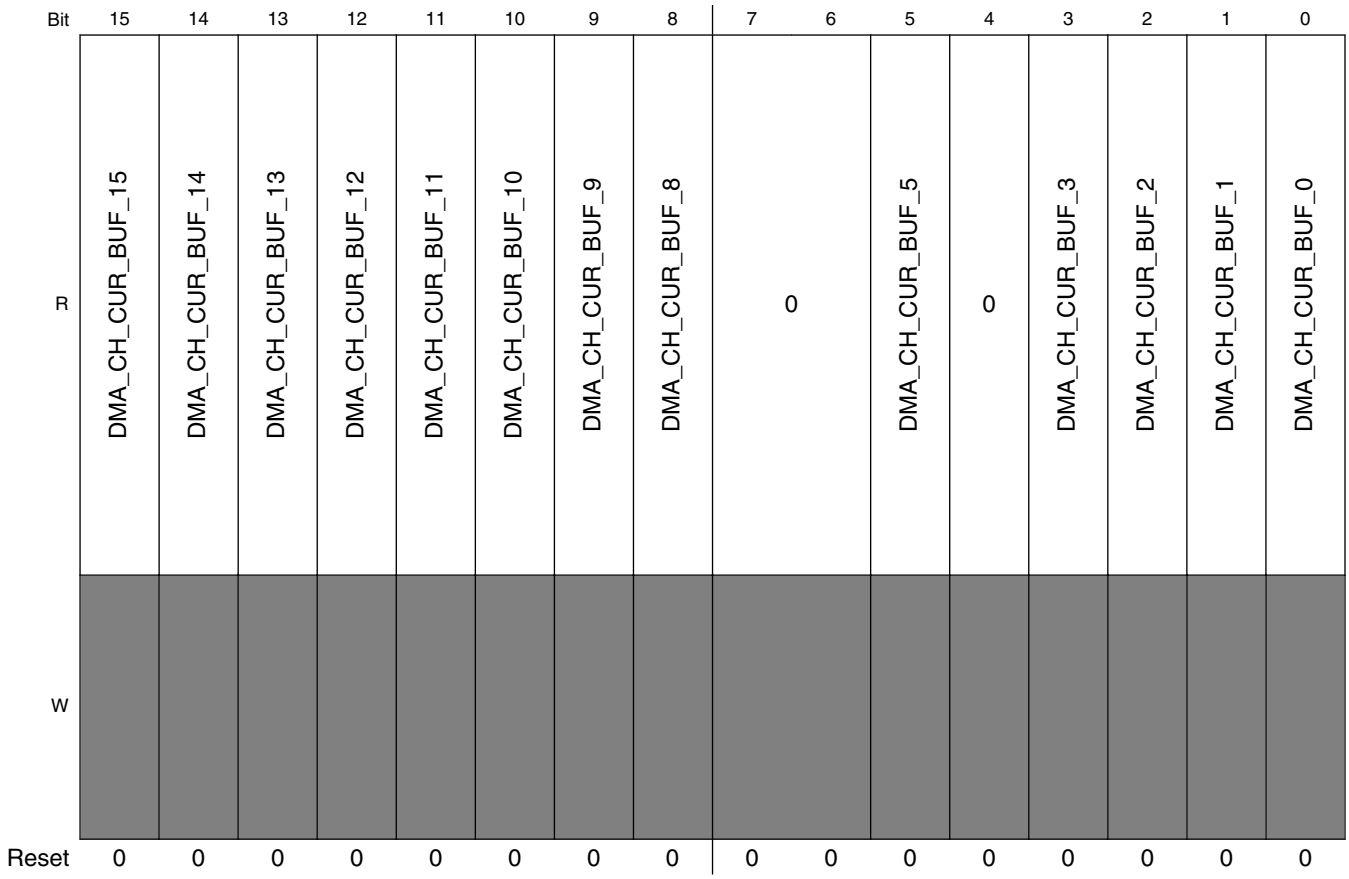
Field	Description
7 DP_ASF_BRAKE	DP Async Flow Brake indication interrupt. This bit indicates on the interrupt that is a result of the async flow brake at the DP 0 Interrupt is cleared. 1 Interrupt is requested.
6 DP_SF_BRAKE	DP Sync Flow Brake indication interrupt. This bit indicates on the interrupt that is a result of the Sync flow brake at the DP 0 Interrupt is cleared. 1 Interrupt is requested.
5 DP_ASF_END	DP Async Flow End indication interrupt. This bit indicates on the interrupt that is a result of the Async flow end at the DP 0 Interrupt is cleared. 1 Interrupt is requested.
4 DP_ASF_START	DP Async Flow Start indication interrupt. This bit indicates on the interrupt that is a result of the Async flow start at the DP 0 Interrupt is cleared. 1 Interrupt is requested.
3 DP_SF_END	DP Sync Flow End indication interrupt. This bit indicates on the interrupt that is a result of the Sync flow end at the DP 0 Interrupt is cleared. 1 Interrupt is requested.
2 DP_SF_START	DP Sync Flow Start indication interrupt. This bit indicates on the interrupt that is a result of the Sync flow start at the DP 0 Interrupt is cleared. 1 Interrupt is requested.
1 SNOOPING2_ INT	IPU snooping 2 event indication interrupt. This bit indicates on the interrupt that is a result of the detection of a snooping 2 signal assertion coming to the IPU 0 Interrupt is cleared. 1 Interrupt is requested.
0 SNOOPING1_ INT	IPU snooping 1 event indication interrupt. This bit indicates on the interrupt that is a result of the detection of a snooping 1 signal assertion coming to the 0 Interrupt is cleared. 1 Interrupt is requested.

38.5.67 Current Buffer Register 0 (IPU_CUR_BUF_0)

This register contains the current buffer status information bit for each DMA channel.

Address: 260_0000h base + 23Ch offset = 260_023Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DMA_CH_CUR_BUF_31	0	DMA_CH_CUR_BUF_29	DMA_CH_CUR_BUF_28	DMA_CH_CUR_BUF_27	DMA_CH_CUR_BUF_26	DMA_CH_CUR_BUF_25	DMA_CH_CUR_BUF_24	DMA_CH_CUR_BUF_23	DMA_CH_CUR_BUF_22	DMA_CH_CUR_BUF_21	DMA_CH_CUR_BUF_20	DMA_CH_CUR_BUF_19	DMA_CH_CUR_BUF_18	DMA_CH_CUR_BUF_17	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



IPU_CUR_BUF_0 field descriptions

Field	Description
31 DMA_CH_CUR_BUF_31	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
30 Reserved	This read-only field is reserved and always has the value 0. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
29 DMA_CH_CUR_BUF_29	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
28 DMA_CH_CUR_BUF_28	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

Table continues on the next page...

IPU_CUR_BUF_0 field descriptions (continued)

Field	Description
27 DMA_CH_CUR_BUF_27	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
26 DMA_CH_CUR_BUF_26	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
25 DMA_CH_CUR_BUF_25	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
24 DMA_CH_CUR_BUF_24	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
23 DMA_CH_CUR_BUF_23	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
22 DMA_CH_CUR_BUF_22	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
21 DMA_CH_CUR_BUF_21	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
20 DMA_CH_CUR_BUF_20	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
19 DMA_CH_CUR_BUF_19	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

Table continues on the next page...

IPU_CUR_BUF_0 field descriptions (continued)

Field	Description
18 DMA_CH_CUR_BUF_18	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
17 DMA_CH_CUR_BUF_17	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
16 Reserved	This read-only field is reserved and always has the value 0. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
15 DMA_CH_CUR_BUF_15	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
14 DMA_CH_CUR_BUF_14	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
13 DMA_CH_CUR_BUF_13	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
12 DMA_CH_CUR_BUF_12	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
11 DMA_CH_CUR_BUF_11	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
10 DMA_CH_CUR_BUF_10	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
9 DMA_CH_CUR_BUF_9	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

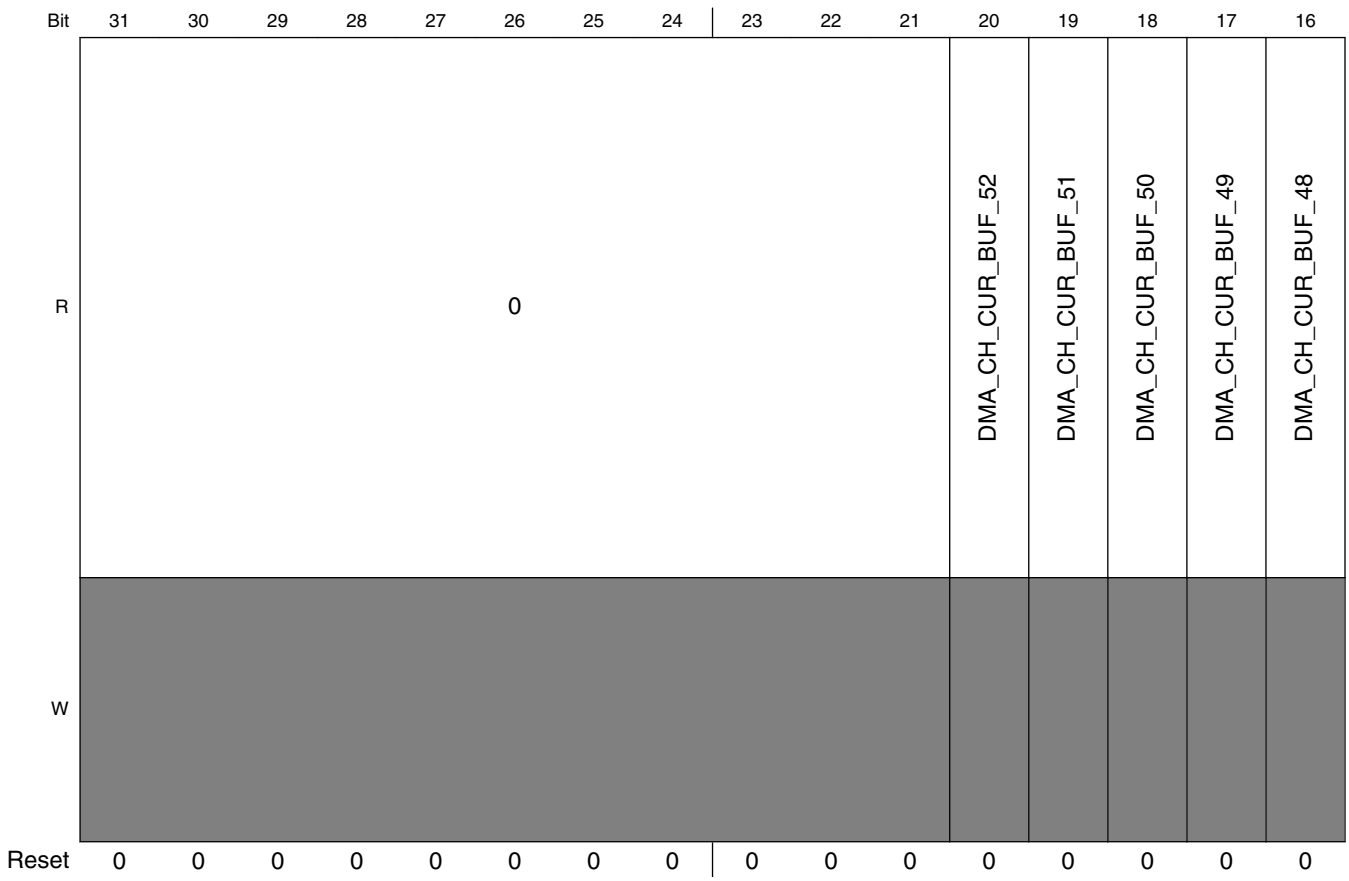
IPU_CUR_BUF_0 field descriptions (continued)

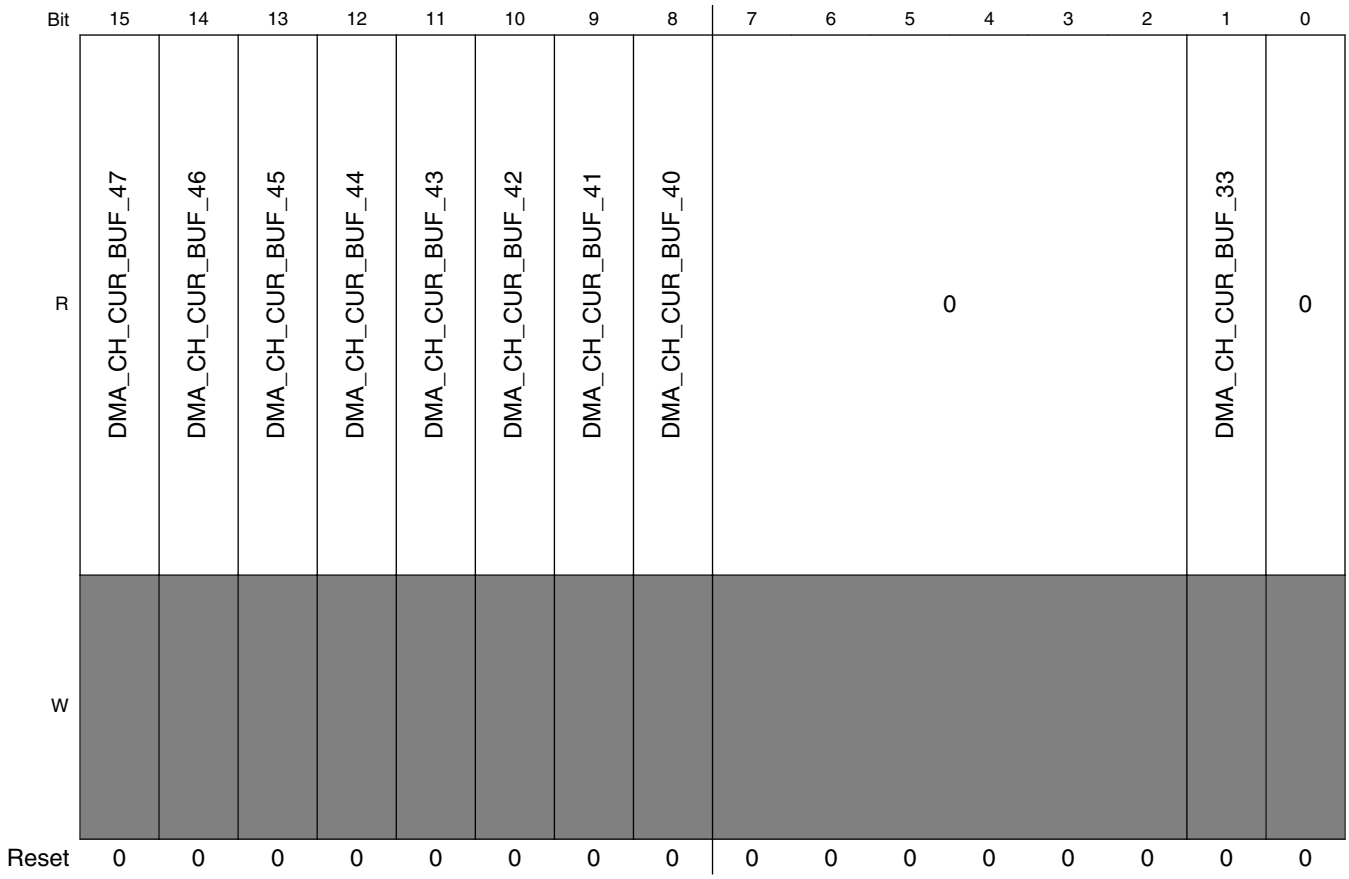
Field	Description
	0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
8 DMA_CH_CUR_BUF_8	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
7–6 Reserved	This read-only field is reserved and always has the value 0. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
5 DMA_CH_CUR_BUF_5	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
4 Reserved	This read-only field is reserved and always has the value 0. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
3 DMA_CH_CUR_BUF_3	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
2 DMA_CH_CUR_BUF_2	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
1 DMA_CH_CUR_BUF_1	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
0 DMA_CH_CUR_BUF_0	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

38.5.68 Current Buffer Register 1 (IPU_CUR_BUF_1)

This register contains the current buffer status information bit for each DMA channel.

Address: 260_0000h base + 240h offset = 260_0240h





IPU_CUR_BUF_1 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0.
20 DMA_CH_CUR_BUF_52	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
19 DMA_CH_CUR_BUF_51	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
18 DMA_CH_CUR_BUF_50	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

Table continues on the next page...

IPU_CUR_BUF_1 field descriptions (continued)

Field	Description
17 DMA_CH_CUR_BUF_49	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
16 DMA_CH_CUR_BUF_48	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
15 DMA_CH_CUR_BUF_47	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
14 DMA_CH_CUR_BUF_46	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
13 DMA_CH_CUR_BUF_45	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
12 DMA_CH_CUR_BUF_44	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
11 DMA_CH_CUR_BUF_43	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
10 DMA_CH_CUR_BUF_42	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
9 DMA_CH_CUR_BUF_41	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

Table continues on the next page...

IPU_CUR_BUF_1 field descriptions (continued)

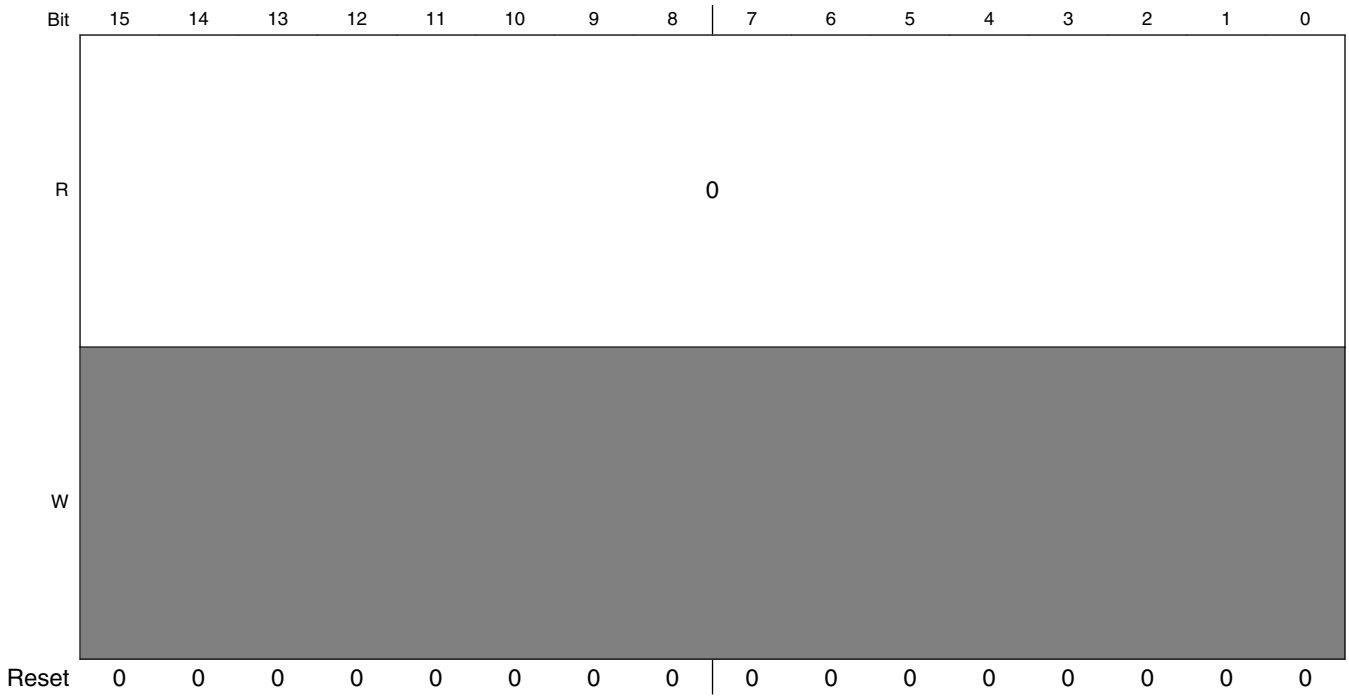
Field	Description
8 DMA_CH_CUR_BUF_40	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
7–2 Reserved	This read-only field is reserved and always has the value 0. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
1 DMA_CH_CUR_BUF_33	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
0 Reserved	This read-only field is reserved and always has the value 0. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

38.5.69 Alternate Current Buffer Register 0 (IPU_ALT_CUR_0)

This register contains the current buffer status information bit for each DMA channel.

Address: 260_0000h base + 244h offset = 260_0244h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		DMA_CH_ALT_CUR_BUF_29	0				DMA_CH_ALT_CUR_BUF_24	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



IPU_ALT_CUR_0 field descriptions

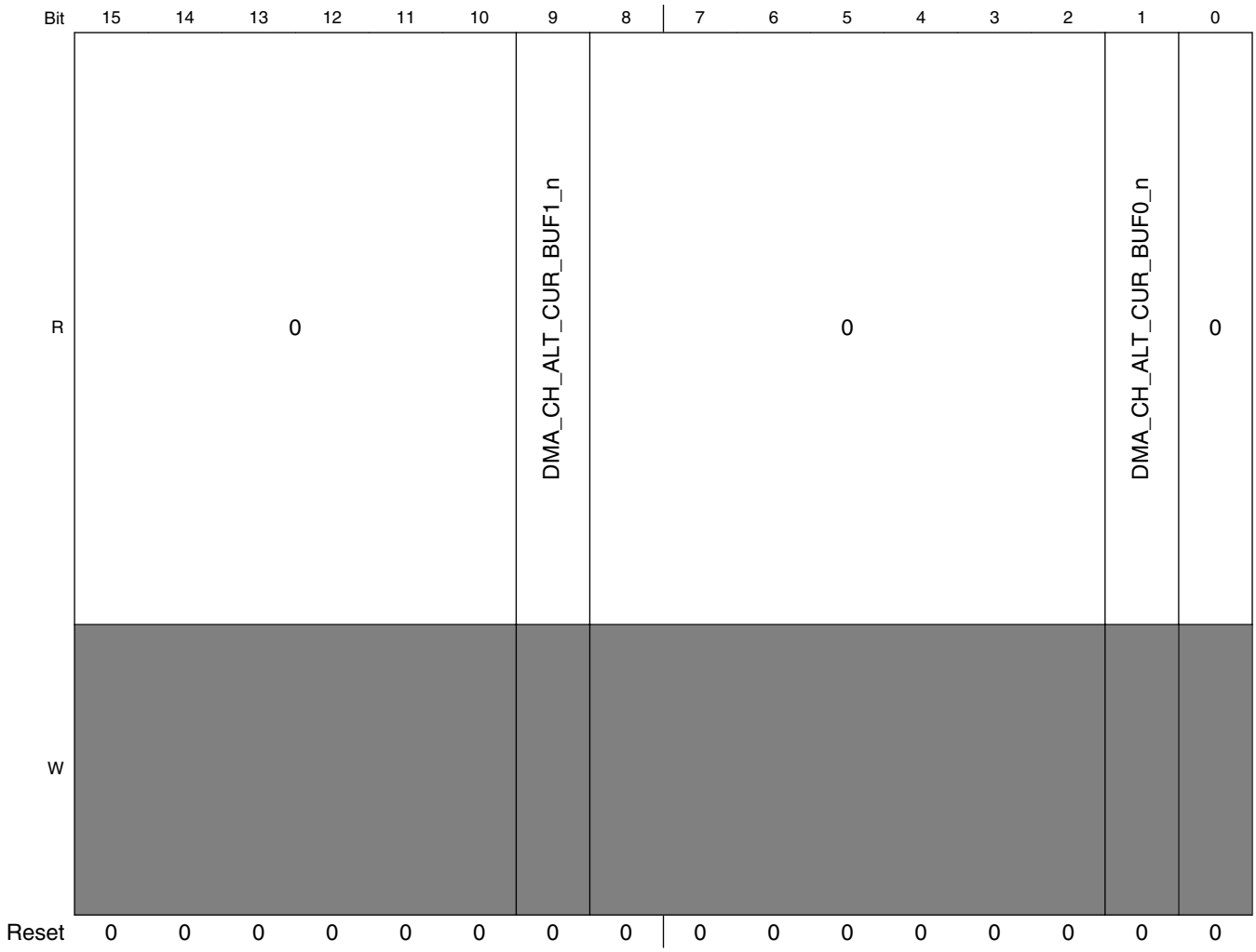
Field	Description
31–30 Reserved	This read-only field is reserved and always has the value 0. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
29 DMA_CH_ALT_ CUR_BUF_29	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
28–25 Reserved	This read-only field is reserved and always has the value 0. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
24 DMA_CH_ALT_ CUR_BUF_24	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
Reserved	This read-only field is reserved and always has the value 0. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

38.5.70 Alternate Current Buffer Register 1 (IPU_ALT_CUR_1)

This register contains the current buffer status information bit for each DMA channel. The register is shown in VDI Plane Size Register 4 , and the register fields are described in VDI Plane Size Register 4 .

Address: 260_0000h base + 248h offset = 260_0248h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											DMA_CH_ALT_CUR_BUF_52	0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



IPU_ALT_CUR_1 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value 0. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
20 DMA_CH_ALT_CUR_BUF_52	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
19–10 Reserved	This read-only field is reserved and always has the value 0. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
9 DMA_CH_ALT_CUR_BUF1_n	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_ALT_CUR_1 field descriptions (continued)

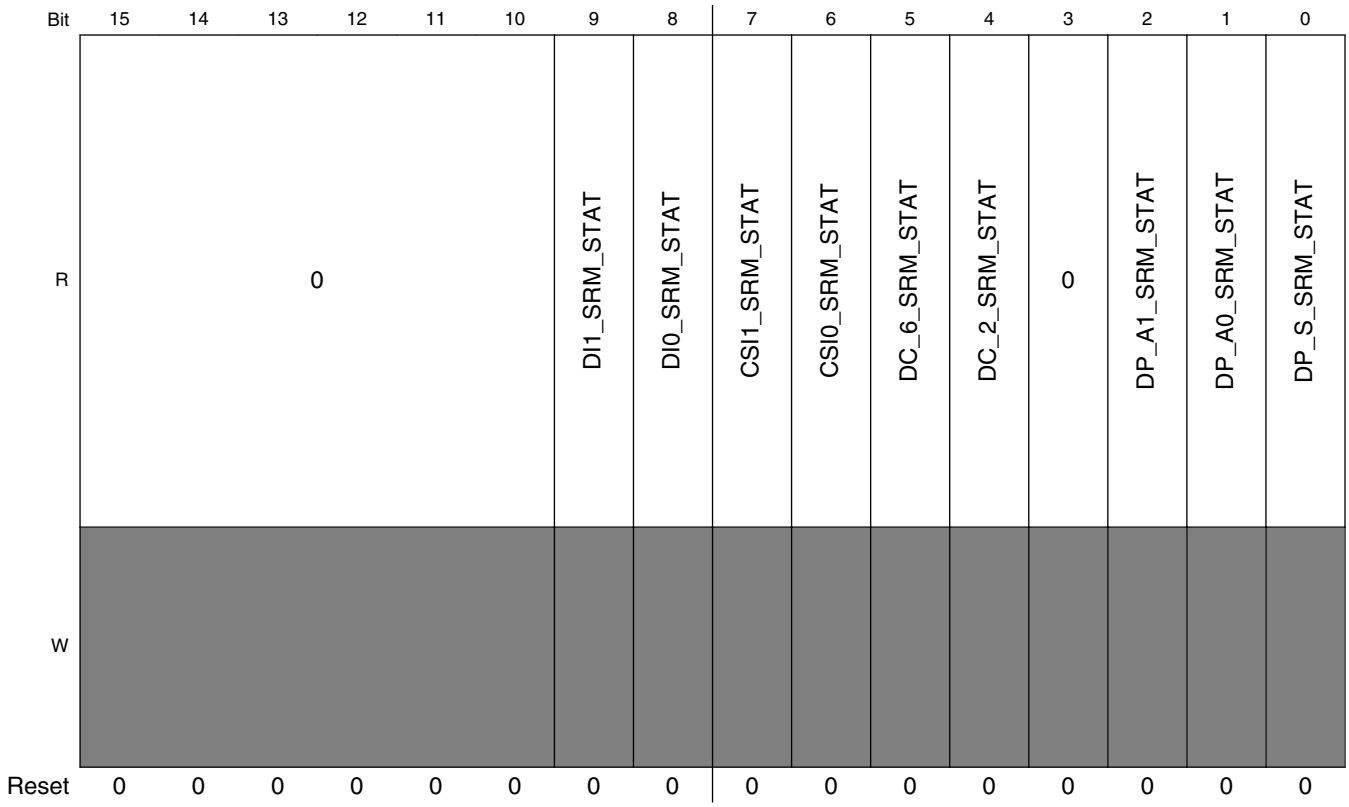
Field	Description
	0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
8–2 Reserved	This read-only field is reserved and always has the value 0. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
1 DMA_CH_ALT_CUR_BUF0_n	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
0 Reserved	This read-only field is reserved and always has the value 0. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

38.5.71 Shadow Registers Memory Status Register (IPU_SRM_STAT)

The register contains status bits of SRM updates. There is a bit for each block. The bit is set when the SRM is currently updating the module's registers. When the SRM completes updating the registers of the block the bit is cleared. SW should not update the block's registers while it is being updated by the SRM.

Address: 260_0000h base + 24Ch offset = 260_024Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



IPU_SRM_STAT field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value 0.
9 DI1_SRM_STAT	DI1 SRM STAT This bit indicates that the SRM is currently updating the DI1 registers 1 SRM is busy updating the DI1 registers 0 SRM is not updating the DI1 registers
8 DI0_SRM_STAT	DI0 SRM STAT This bit indicates that the SRM is currently updating the DI0 registers 1 SRM is busy updating the DI0 registers 0 SRM is not updating the DI0 registers
7 CSI1_SRM_ STAT	CSI1_SRM_STAT
6 CSI0_SRM_ STAT	CSI1_SRM_STAT
5 DC_6_SRM_ STAT	DC group #6 SRM STAT This bit indicates that the SRM is currently updating the DC group #6 registers

Table continues on the next page...

IPU_SRM_STAT field descriptions (continued)

Field	Description
	1 SRM is busy updating the DC registers 0 SRM is not updating the DC registers
4 DC_2_SRM_STAT	DC group #2 SRM STAT This bit indicates that the SRM is currently updating the DC group #2 registers 1 SRM is busy updating the DC group #6 registers 0 SRM is not updating the DC group #2 registers
3 Reserved	This read-only field is reserved and always has the value 0.
2 DP_A1_SRM_STAT	DP ASYNC1 FLOW SRM STAT This bit indicates that the SRM is currently updating the DP async flow 1 registers 1 SRM is busy updating the DP sync flow registers 0 SRM is not updating the DP sync flow registers
1 DP_A0_SRM_STAT	DP ASYNC0 FLOW SRM STAT This bit indicates that the SRM is currently updating the DP async flow 0 registers 1 SRM is busy updating the DP async flow 0 registers 0 SRM is not updating the DP async flow 0 registers
0 DP_S_SRM_STAT	DP SYNC FLOW SRM STAT This bit indicates that the SRM is currently updating the DP sync flow registers 1 SRM is busy updating the DP sync flow registers 0 SRM is not updating the DP sync flow registers

38.5.72 Processing Status Tasks Register (IPU_PROC_TASKS_STAT)

This register contains status bits for IPU's tasks.

Address: 260_0000h base + 250h offset = 260_0250h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MEM2PRP_TSTAT			PP_ROT_TSTAT		VF_ROT_TSTAT		ENC_ROT_TSTAT		PP_TSTAT		VF_TSTAT		ENC_TSTAT	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_PROC_TASKS_STAT field descriptions

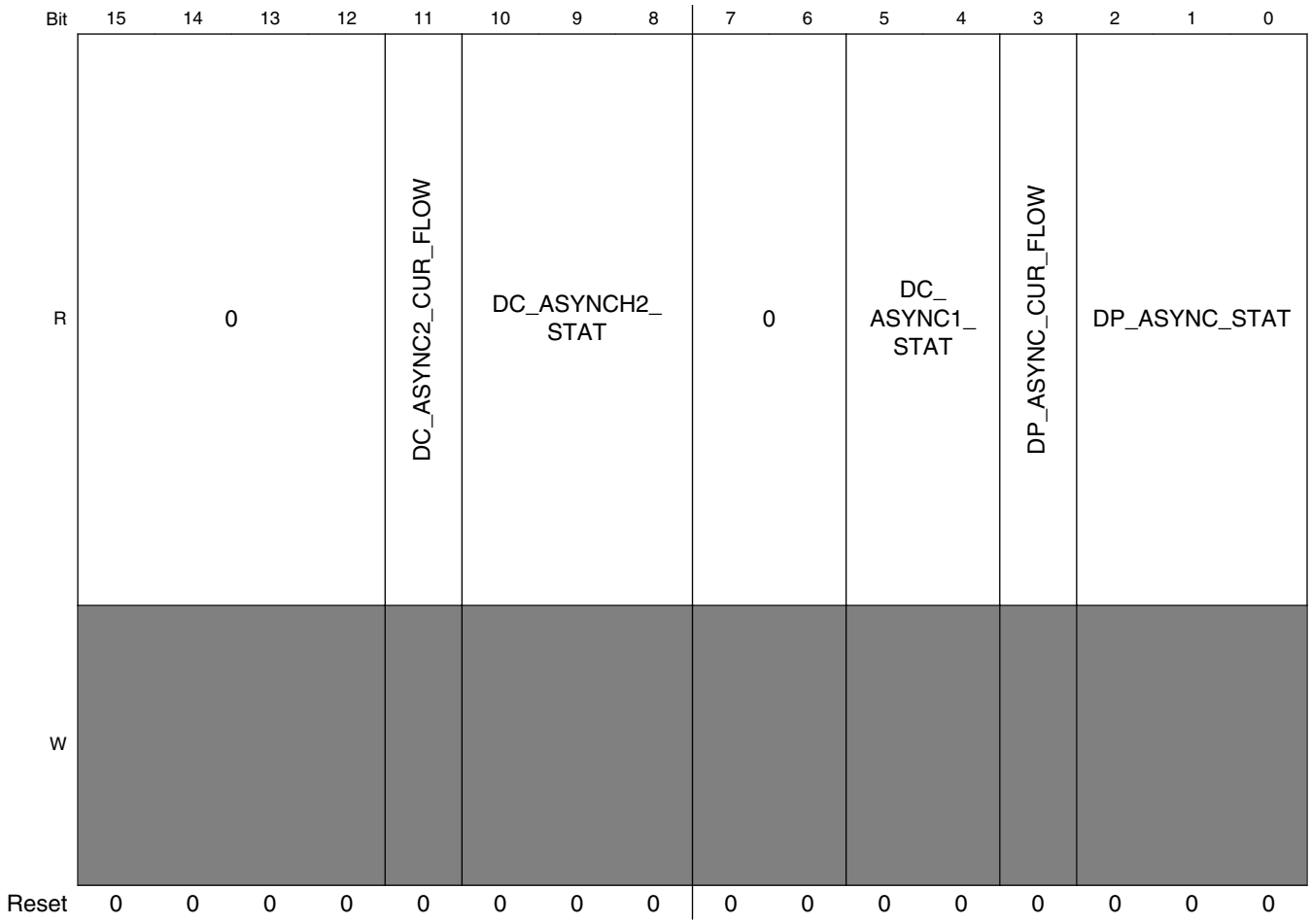
Field	Description
31–15 Reserved	This read-only field is reserved and always has the value 0.
14–12 MEM2PRP_ TSTAT	Status of the pre processing tasks (viewfinder and encoding) when the source is coming from the memory. 000 IDLE - Both pre processing tasks are idle 001 BOTH_ACTIVE - Both pre processing tasks are idle 010 ENC_ACTIVE - Encoding task is active 011 VF_ACTIVE - View finder task is active 100 BOTH_PAUSE - both tasks are paused 101 Reserved 110 Reserved 111 Reserve
11–10 PP_ROT_TSTAT	Status of the rotation for post processing task 00 IDLE - The task is idle 01 ACTIVE - The primary flow of this task is currently active 10 WAIT_FOR_ READY - The task is waiting for a buffer to be ready
9–8 VF_ROT_TSTAT	Status of the rotation for viewfinder task 00 IDLE - The task is idle 01 ACTIVE - The primary flow of this task is currently active 10 WAIT_FOR_ READY - The task is waiting for a buffer to be ready
7–6 ENC_ROT_ TSTAT	Status of the rotation for encoding task 00 IDLE - The task is idle 01 ACTIVE - The primary flow of this task is currently active 10 WAIT_FOR_ READY - The task is waiting for a buffer to be ready
5–4 PP_TSTAT	Status of the post processing task 00 IDLE - The task is idle 01 ACTIVE - The primary flow of this task is currently active 10 WAIT_FOR_ READY - The task is waiting for a buffer to be ready
3–2 VF_TSTAT	Status of the viewfinder task 00 IDLE - The task is idle 01 ACTIVE - The primary flow of this task is currently active 10 WAIT_FOR_ READY - The task is waiting for a buffer to be ready
ENC_TSTAT	Status of the encoding task 00 IDLE - The task is idle 01 ACTIVE - The primary flow of this task is currently active 10 WAIT_FOR_ READY - The task is waiting for a buffer to be ready

38.5.73 Display Tasks Status Register (IPU_DISP_TASKS_STAT)

This register contains status bits for IPU's tasks.

Address: 260_0000h base + 254h offset = 260_0254h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



IPU_DISP_TASKS_STAT field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value 0.
11 DC_ASYNC2_CUR_FLOW	Current asynchronous #2 flow via the DC 1 alternate flow 0 main flow
10–8 DC_ASYNCH2_STAT	Status of the Asynchronous flow #2 through the DC 000 IDLE - the task is idle 001 PRIM_ACTIVE - The primary flow of this task is currently active 010 ALT_ACTIVE - The alternate flow of this task is currently active 011 UPDATE_PARAM - The FSU is busy updating parameters from the SRM 100 PAUSE - The task is paused 101 Reserved 110 Reserved 111 Reserved
7–6 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DISP_TASKS_STAT field descriptions (continued)

Field	Description
5–4 DC_ASYNC1_ STAT	Status of the Asynchronous flow #1 through the DC (ch 28) 00 IDLE - The task is idle 01 ACTIVE - This task is currently active 10 WAIT_FOR_ READY - The task is waiting for a buffer to be ready
3 DP_ASYNC_ CUR_FLOW	Current asynchronous flow via the DP 1 alternate flow 0 main flow
DP_ASYNC_ STAT	Status of the Asynchronous flow through the DP 000 IDLE - the task is idle 001 PRIM_ACTIVE - The primary flow of this task is currently active 010 ALT_ACTIVE - The alternate flow of this task is currently active 011 UPDATE_PARAM - The FSU is busy updating parameters from the SRM 100 PAUSE - The task is paused 101 Reserved 110 Reserved 111 Reserved

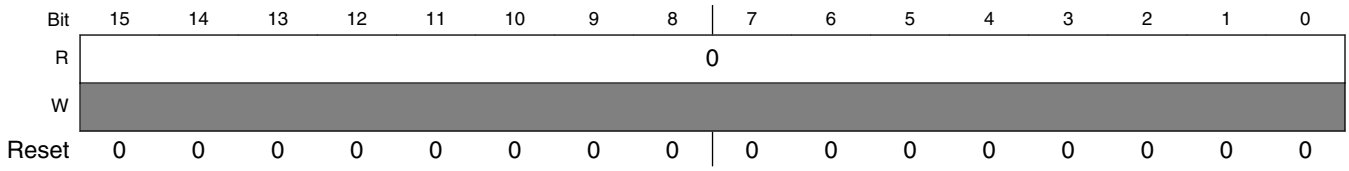
38.5.74 Triple Current Buffer Register 0 (IPU_TRIPLE_CUR_BUF_0)

This register contains the current buffer status information for triple buffer mode for each DMA channel.

- Hide VPU_SUB_FRAME_SYNC for all versions
- Show VPU_SUB_FRAME_SYNC for IPUv3H version.
- The table below tagged with other settings (like IPU3M_only) should be hidden in IPUv3H version.
- This requires some sophisticated conditional tag settings

Address: 260_0000h base + 258h offset = 260_0258h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				DMA_CH_TRIPLE_CUR_BUF_13		0				DMA_CH_TRIPLE_CUR_BUF_10		DMA_CH_TRIPLE_CUR_BUF_9		DMA_CH_TRIPLE_CUR_BUF_8	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



IPU_TRIPLE_CUR_BUF_0 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–26 DMA_CH_ TRIPLE_CUR_ BUF_13	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.
25–22 Reserved	This read-only field is reserved and always has the value 0.
21–20 DMA_CH_ TRIPLE_CUR_ BUF_10	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.
19–18 DMA_CH_ TRIPLE_CUR_ BUF_9	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.
17–16 DMA_CH_ TRIPLE_CUR_ BUF_8	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.
Reserved	This read-only field is reserved and always has the value 0.

38.5.75 Triple Current Buffer Register 1 (IPU_TRIPLE_CUR_BUF_1)

This register contains the current buffer status information for triple buffer mode for each DMA channel.

Address: 260_0000h base + 25Ch offset = 260_025Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						DMA_CH_TRIPLE_CUR_BUF_28		DMA_CH_TRIPLE_CUR_BUF_27		0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DMA_CH_TRIPLE_CUR_BUF_23		0		DMA_CH_TRIPLE_CUR_BUF_21		0									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_TRIPLE_CUR_BUF_1 field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.
25–24 DMA_CH_TRIPLE_CUR_BUF_28	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.
23–22 DMA_CH_TRIPLE_CUR_BUF_27	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.

Table continues on the next page...

IPU_TRIPLE_CUR_BUF_1 field descriptions (continued)

Field	Description
21–16 Reserved	This read-only field is reserved and always has the value 0.
15–14 DMA_CH_ TRIPLE_CUR_ BUF_23	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.
13–12 Reserved	This read-only field is reserved and always has the value 0.
11–10 DMA_CH_ TRIPLE_CUR_ BUF_21	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.
Reserved	This read-only field is reserved and always has the value 0.

38.5.76 Triple Current Buffer Register 2 (IPU_TRIPLE_CUR_BUF_2)

This register contains the current buffer status information for triple buffer mode for each DMA channel.

Address: 260_0000h base + 260h offset = 260_0260h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_TRIPLE_CUR_BUF_2 field descriptions

Field	Description
Reserved	This read-only field is reserved and always has the value 0. 00 Current buffer used by DMA is buffer 0.

IPU_TRIPLE_CUR_BUF_2 field descriptions (continued)

Field	Description
	01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.

38.5.76 Triple Current Buffer Register 3 (IPU_TRIPLE_CUR_BUF_3)

This register contains the current buffer status information for triple buffer mode for each DMA channel.

Address: 260_0000h base + 264h offset = 260_0264h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_TRIPLE_CUR_BUF_3 field descriptions

Field	Description
Reserved	This read-only field is reserved and always has the value 0. 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.

38.5.76 IPU Channels Buffer 0 Ready 0 Register (IPU_CH_BUF0_RDY0)

The register contains buffer 0 ready control information for 32 IPU's DMA channels (31-0). This register can be a write one to set or a write one to clear according to the **IPU_CH_BUF0_RDY0_CLR bit**.

The register is shown in IPU Channels Buffer 0 Ready 0 Register (IPU_CH_BUF0_RDY0), and the register fields are described in IPU Channels Buffer 0 Ready 0 Register (IPU_CH_BUF0_RDY0).

Address: 260_0000h base + 268h offset = 260_0268h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_CH_BUF0_RDY_31	Reserved	DMA_CH_BUF0_RDY_29	DMA_CH_BUF0_RDY_28	DMA_CH_BUF0_RDY_27	Reserved		DMA_CH_BUF0_RDY_24	DMA_CH_BUF0_RDY_23	DMA_CH_BUF0_RDY_22	DMA_CH_BUF0_RDY_21	DMA_CH_BUF0_RDY_20	Reserved	DMA_CH_BUF0_RDY_18	DMA_CH_BUF0_RDY_17	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_CH_BUF0_RDY_15	DMA_CH_BUF0_RDY_14	DMA_CH_BUF0_RDY_13	DMA_CH_BUF0_RDY_12	DMA_CH_BUF0_RDY_11	DMA_CH_BUF0_RDY_10	DMA_CH_BUF0_RDY_9	DMA_CH_BUF0_RDY_8	DMA_CH_BUF0_RDY_7	DMA_CH_BUF0_RDY_6	DMA_CH_BUF0_RDY_5	DMA_CH_BUF0_RDY_4	DMA_CH_BUF0_RDY_3	DMA_CH_BUF0_RDY_2	DMA_CH_BUF0_RDY_1	DMA_CH_BUF0_RDY_0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CH_BUF0_RDY0 field descriptions

Field	Description
31 DMA_CH_BUF0_RDY_31	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
30 -	This field is reserved. Reserved.
29 DMA_CH_BUF0_RDY_29	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
28 DMA_CH_BUF0_RDY_28	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
27 DMA_CH_BUF0_RDY_27	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
26–25 -	This field is reserved. Reserved.
24 DMA_CH_BUF0_RDY_24	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.

Table continues on the next page...

IPU_CH_BUF0_RDY0 field descriptions (continued)

Field	Description
23 DMA_CH_BUF0_RDY_23	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
22 DMA_CH_BUF0_RDY_22	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
21 DMA_CH_BUF0_RDY_21	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
20 DMA_CH_BUF0_RDY_20	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
19 -	This field is reserved. Reserved.
18 DMA_CH_BUF0_RDY_18	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
17 DMA_CH_BUF0_RDY_17	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
16 -	This field is reserved. Reserved.
15 DMA_CH_BUF0_RDY_15	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
14 DMA_CH_BUF0_RDY_14	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
13 DMA_CH_BUF0_RDY_13	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
12 DMA_CH_BUF0_RDY_12	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
11 DMA_CH_BUF0_RDY_11	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.

Table continues on the next page...

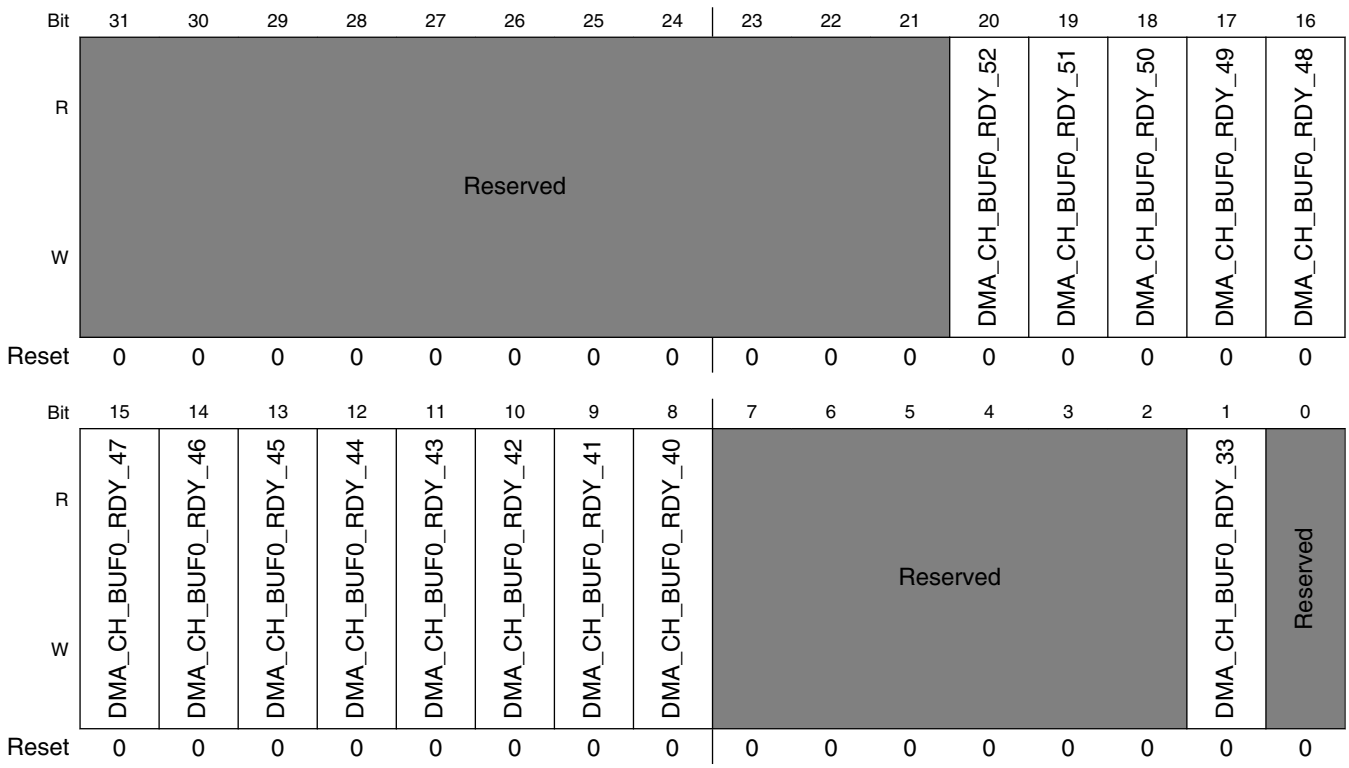
IPU_CH_BUF0_RDY0 field descriptions (continued)

Field	Description
10 DMA_CH_BUF0_RDY_10	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
9 DMA_CH_BUF0_RDY_9	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
8 DMA_CH_BUF0_RDY_8	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
7 DMA_CH_BUF0_RDY_7	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
6 DMA_CH_BUF0_RDY_6	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
5 DMA_CH_BUF0_RDY_5	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
4 DMA_CH_BUF0_RDY_4	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
3 DMA_CH_BUF0_RDY_3	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
2 DMA_CH_BUF0_RDY_2	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
1 DMA_CH_BUF0_RDY_1	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
0 DMA_CH_BUF0_RDY_0	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.

38.5.77 IPU Channels Buffer 0 Ready 1 Register (IPU_CH_BUF0_RDY1)

The register contains buffer 0 ready control information for 32 IPU's DMA channels (63-32). This register can be a write one to set or a write one to clear according to the **IPU_CH_BUF0_RDY1_CLR bit**.

Address: 260_0000h base + 26Ch offset = 260_026Ch



IPU_CH_BUF0_RDY1 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved.
20 DMA_CH_BUF0_RDY_52	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
19 DMA_CH_BUF0_RDY_51	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.

Table continues on the next page...

IPU_CH_BUF0_RDY1 field descriptions (continued)

Field	Description
18 DMA_CH_BUF0_RDY_50	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
17 DMA_CH_BUF0_RDY_49	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
16 DMA_CH_BUF0_RDY_48	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
15 DMA_CH_BUF0_RDY_47	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
14 DMA_CH_BUF0_RDY_46	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
13 DMA_CH_BUF0_RDY_45	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
12 DMA_CH_BUF0_RDY_44	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
11 DMA_CH_BUF0_RDY_43	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
10 DMA_CH_BUF0_RDY_42	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
9 DMA_CH_BUF0_RDY_41	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
8 DMA_CH_BUF0_RDY_40	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
7–2 -	This field is reserved. Reserved.
1 DMA_CH_BUF0_RDY_33	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory.

Table continues on the next page...

IPU_CH_BUF0_RDY1 field descriptions (continued)

Field	Description
	0 Buffer 0 is not ready. 1 Buffer 0 is ready.
0 -	This field is reserved. Reserved.

38.5.78 IPU Channels Buffer 1 Ready 0 Register (IPU_CH_BUF1_RDY0)

The register contains buffer 1 ready control information for 32 IPU's DMA channels (31-0). This register can be a write one to set or a write one to clear according to the **IPU_CH_BUF1_RDY0_CLR bit**.

Address: 260_0000h base + 270h offset = 260_0270h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_CH_BUF1_RDY_31	Reserved	DMA_CH_BUF1_RDY_29	DMA_CH_BUF1_RDY_28	DMA_CH_BUF1_RDY_27	DMA_CH_BUF1_RDY_26	DMA_CH_BUF1_RDY_25	DMA_CH_BUF1_RDY_24	DMA_CH_BUF1_RDY_23	DMA_CH_BUF1_RDY_22	DMA_CH_BUF1_RDY_21	DMA_CH_BUF1_RDY_20	DMA_CH_BUF1_RDY_19	DMA_CH_BUF1_RDY_18	DMA_CH_BUF1_RDY_17	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_CH_BUF1_RDY_15	DMA_CH_BUF1_RDY_14	DMA_CH_BUF1_RDY_13	DMA_CH_BUF1_RDY_12	DMA_CH_BUF1_RDY_	DMA_CH_BUF1_RDY_10	DMA_CH_BUF1_RDY_9	DMA_CH_BUF1_RDY_8	Reserved		DMA_CH_BUF1_RDY_5	Reserved	DMA_CH_BUF1_RDY_3	DMA_CH_BUF1_RDY_2	DMA_CH_BUF1_RDY_1	DMA_CH_BUF1_RDY_0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CH_BUF1_RDY0 field descriptions

Field	Description
31 DMA_CH_BUF1_ RDY_31	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory.

Table continues on the next page...

IPU_CH_BUF1_RDY0 field descriptions (continued)

Field	Description
	0 Buffer 0 is not ready. 1 Buffer 0 is ready.
30 -	This field is reserved. Reserved.
29 DMA_CH_BUF1_RDY_29	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
28 DMA_CH_BUF1_RDY_28	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
27 DMA_CH_BUF1_RDY_27	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
26 DMA_CH_BUF1_RDY_26	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
25 DMA_CH_BUF1_RDY_25	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
24 DMA_CH_BUF1_RDY_24	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
23 DMA_CH_BUF1_RDY_23	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
22 DMA_CH_BUF1_RDY_22	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
21 DMA_CH_BUF1_RDY_21	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
20 DMA_CH_BUF1_RDY_20	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
19 DMA_CH_BUF1_RDY_19	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.

Table continues on the next page...

IPU_CH_BUF1_RDY0 field descriptions (continued)

Field	Description
18 DMA_CH_BUF1_RDY_18	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
17 DMA_CH_BUF1_RDY_17	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
16 -	This field is reserved. Reserved.
15 DMA_CH_BUF1_RDY_15	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
14 DMA_CH_BUF1_RDY_14	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
13 DMA_CH_BUF1_RDY_13	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
12 DMA_CH_BUF1_RDY_12	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
11 DMA_CH_BUF1_RDY_	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
10 DMA_CH_BUF1_RDY_10	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
9 DMA_CH_BUF1_RDY_9	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
8 DMA_CH_BUF1_RDY_8	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
7–6 -	This field is reserved. Reserved.
5 DMA_CH_BUF1_RDY_5	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.

Table continues on the next page...

IPU_CH_BUF1_RDY0 field descriptions (continued)

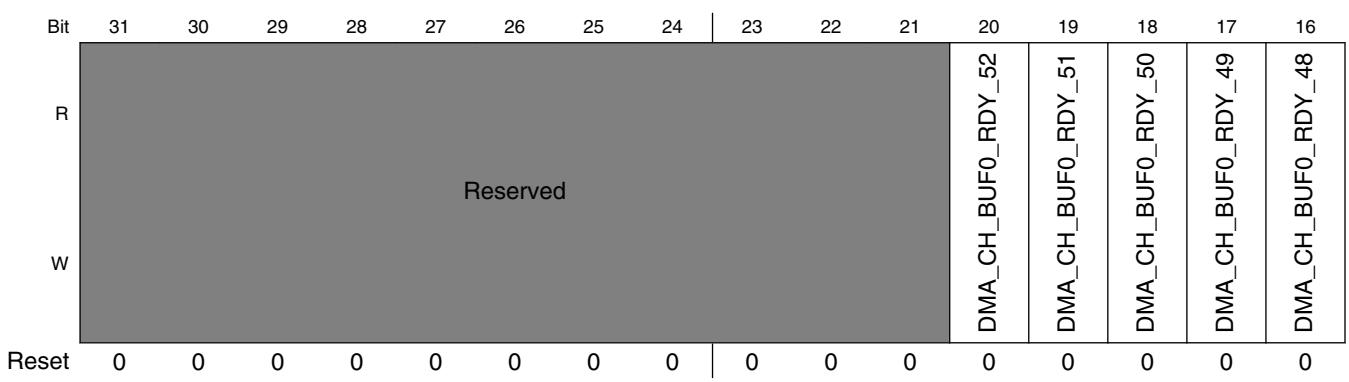
Field	Description
4 -	This field is reserved. Reserved.
3 DMA_CH_BUF1_RDY_3	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
2 DMA_CH_BUF1_RDY_2	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
1 DMA_CH_BUF1_RDY_1	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
0 DMA_CH_BUF1_RDY_0	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.

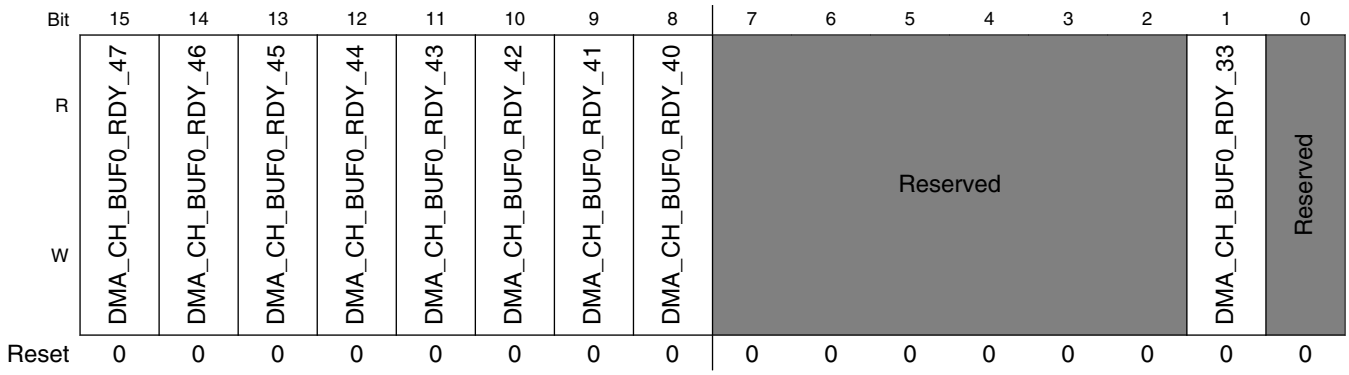
38.5.79 IPU Channels Buffer 1 Ready 1Register (IPU_CH_BUF1_RDY1)

The register contains buffer 0 ready control information for 32 IPU's DMA channels (63-32). This register can be a write one to set or a write one to clear according to the **IPU_CH_BUF1_RDY1_CLR bit**.

The register is shown in IPU Channels Buffer 1 Ready 1Register (IPU_CH_BUF1_RDY1), and the register fields are described in IPU Channels Buffer 1 Ready 1Register (IPU_CH_BUF1_RDY1).

Address: 260_0000h base + 274h offset = 260_0274h





IPU_CH_BUF1_RDY1 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved.
20 DMA_CH_BUF0_RDY_52	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
19 DMA_CH_BUF0_RDY_51	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
18 DMA_CH_BUF0_RDY_50	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
17 DMA_CH_BUF0_RDY_49	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
16 DMA_CH_BUF0_RDY_48	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
15 DMA_CH_BUF0_RDY_47	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
14 DMA_CH_BUF0_RDY_46	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
13 DMA_CH_BUF0_RDY_45	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.

Table continues on the next page...

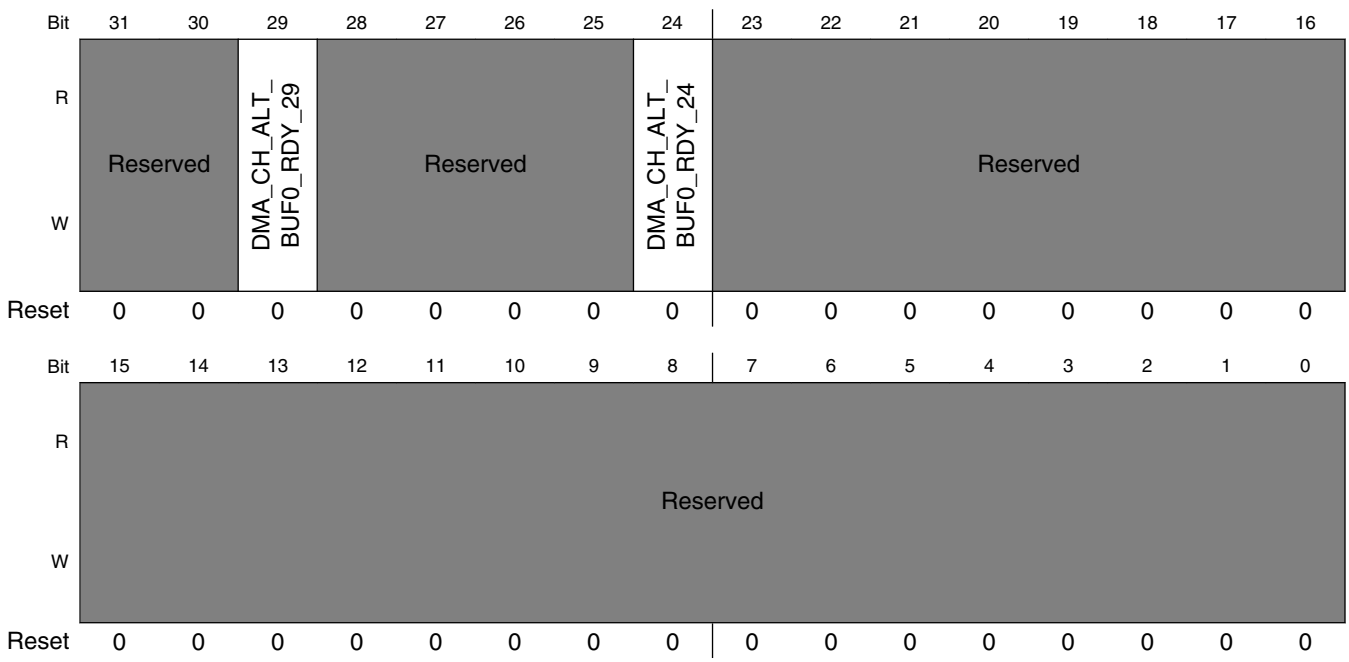
IPU_CH_BUF1_RDY1 field descriptions (continued)

Field	Description
12 DMA_CH_BUF0_RDY_44	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
11 DMA_CH_BUF0_RDY_43	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
10 DMA_CH_BUF0_RDY_42	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
9 DMA_CH_BUF0_RDY_41	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
8 DMA_CH_BUF0_RDY_40	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
7–2 -	This field is reserved. Reserved.
1 DMA_CH_BUF0_RDY_33	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
0 -	This field is reserved. Reserved.

38.5.80 IPU Alternate Channels Buffer 0 Ready 0 Register (IPU_ALT_CH_BUF0_RDY0)

The register contains buffer 0 ready control information for 32 IPU's DMA channels (31-0). Writing "1" to each field will set each bit. Writing "0" to each field simultaneously will clear all the bits.

Address: 260_0000h base + 278h offset = 260_0278h



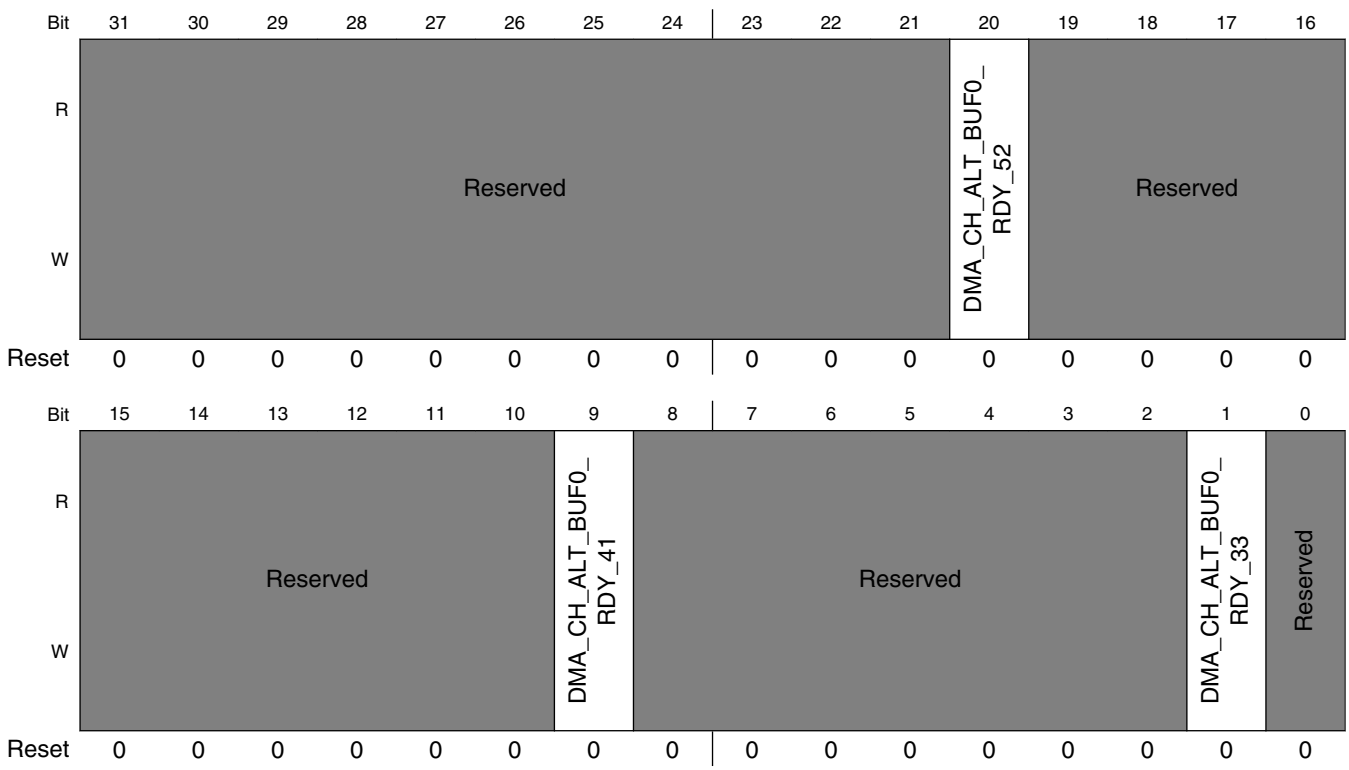
IPU_ALT_CH_BUF0_RDY0 field descriptions

Field	Description
31–30 -	This field is reserved. Reserved.
29 DMA_CH_ALT_BUF0_RDY_29	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
28–25 -	This field is reserved. Reserved.
24 DMA_CH_ALT_BUF0_RDY_24	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
-	This field is reserved. Reserved.

38.5.81 IPU Alternate Channels Buffer 0 Ready 1 Register (IPU_ALT_CH_BUF0_RDY1)

The register contains buffer 0 ready control information for 32 IPU's DMA channels (63-32). Writing "1" to each field will set each bit. Writing "0" to each field simultaneously will clear all the bits.

Address: 260_0000h base + 27Ch offset = 260_027Ch



IPU_ALT_CH_BUF0_RDY1 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved.
20 DMA_CH_ALT_BUF0_RDY_52	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
19–10 -	This field is reserved. Reserved.
9 DMA_CH_ALT_BUF0_RDY_41	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory.

Table continues on the next page...

IPU_ALT_CH_BUF0_RDY1 field descriptions (continued)

Field	Description
	0 Buffer 0 is not ready. 1 Buffer 0 is ready.
8–2 -	This field is reserved. Reserved.
1 DMA_CH_ALT_BUF0_RDY_33	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
0 -	This field is reserved. Reserved.

38.5.82 IPU Alternate Channels Buffer1 Ready 0 Register (IPU_ALT_CH_BUF1_RDY0)

The register contains buffer 1 ready control information for 32 IPU's DMA channels (31-0). Writing "1" to each field will set each bit. Writing "0" to each field simultaneously will clear all the bits.

The register is shown in IPU Alternate Channels Buffer1 Ready 0 Register (IPU_ALT_CH_BUF1_RDY0), and the register fields are described in IPU Alternate Channels Buffer1 Ready 0 Register (IPU_ALT_CH_BUF1_RDY0).

Address: 260_0000h base + 280h offset = 260_0280h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		DMA_CH_ALT_BUF1_RDY_29	Reserved				DMA_CH_ALT_BUF1_RDY_24	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_ALT_CH_BUF1_RDY0 field descriptions

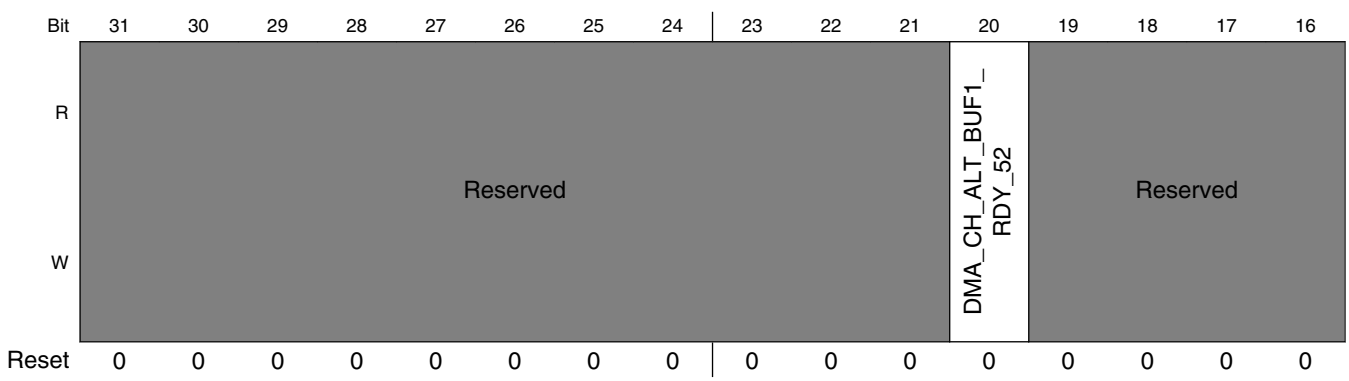
Field	Description
31–30 -	This field is reserved. Reserved.
29 DMA_CH_ALT_BUF1_RDY_29	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 buffer 1 is ready.
28–25 -	This field is reserved. Reserved.
24 DMA_CH_ALT_BUF1_RDY_24	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 buffer 1 is ready.
-	This field is reserved. Reserved.

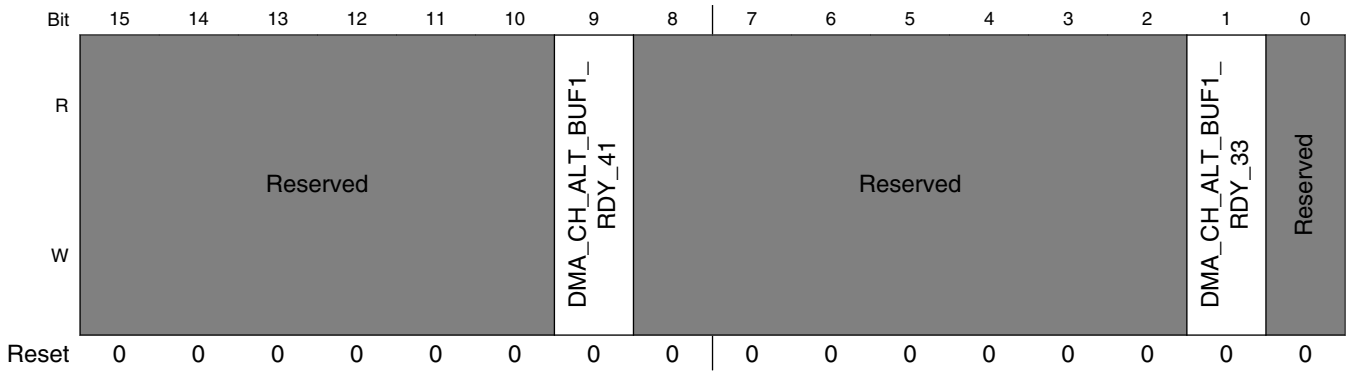
38.5.83 IPU Alternate Channels Buffer 1 Ready 1 Register (IPU_ALT_CH_BUF1_RDY1)

The register contains buffer 0 ready control information for 32 IPU's DMA channels (63-32). Writing "1" to each field will set each bit. Writing "0" to each field simultaneously will clear all the bits.

The register is shown in IPU Alternate Channels Buffer 1 Ready 1 Register (IPU_ALT_CH_BUF1_RDY1), and the register fields are described in IPU Alternate Channels Buffer 1 Ready 1 Register (IPU_ALT_CH_BUF1_RDY1).

Address: 260_0000h base + 284h offset = 260_0284h





IPU_ALT_CH_BUF1_RDY1 field descriptions

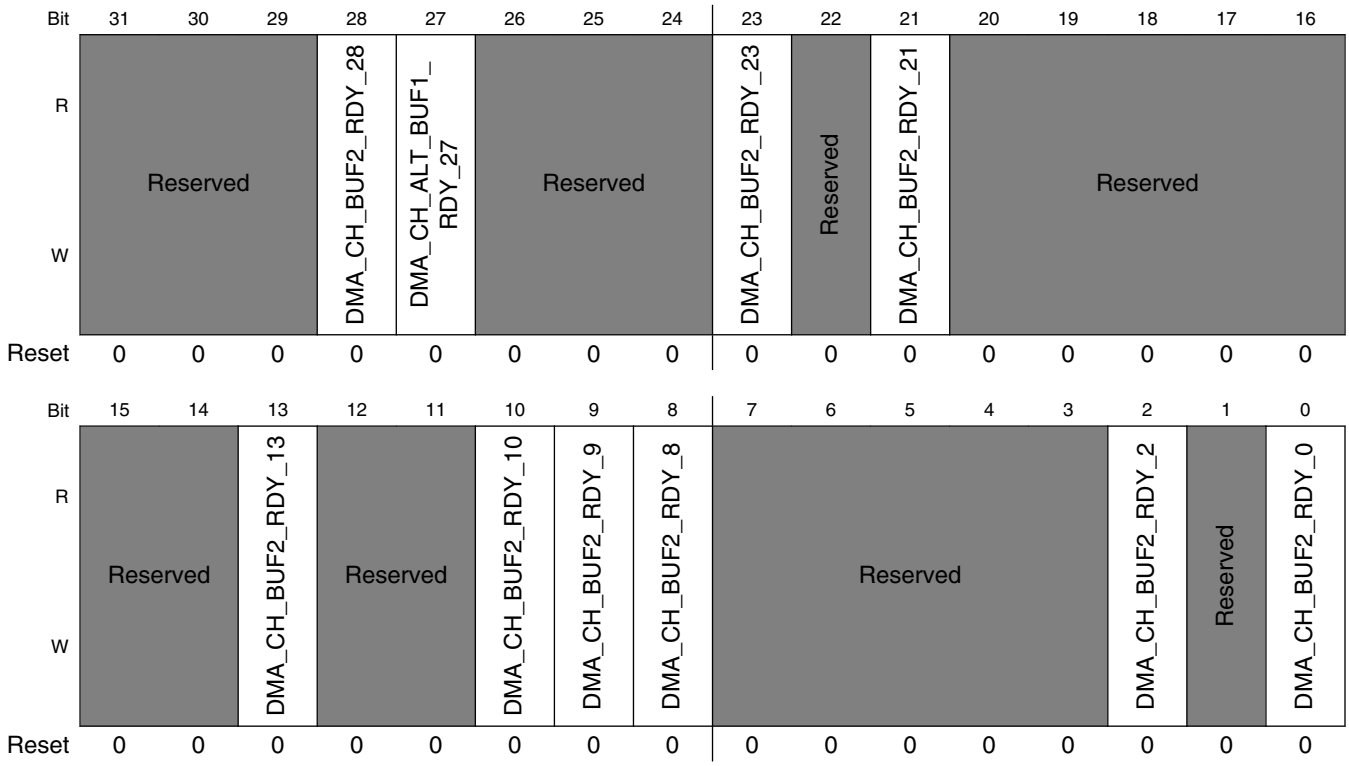
Field	Description
31–21 -	This field is reserved. Reserved.
20 DMA_CH_ALT_BUF1_RDY_52	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 buffer 1 is ready.
19–10 -	This field is reserved. Reserved.
9 DMA_CH_ALT_BUF1_RDY_41	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 buffer 1 is ready.
8–2 -	This field is reserved. Reserved.
1 DMA_CH_ALT_BUF1_RDY_33	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 buffer 1 is ready.
0 -	This field is reserved. Reserved.

38.5.84 IPU Channels Buffer 2 Ready 0 Register (IPU_CH_BUF2_RDY0)

The register contains buffer 2 ready control information for 32 IPU's DMA channels (31-0). This register can be a write one to set or a write one to clear according to the **IPU_CH_BUF2_RDY0_CLR bit**.

The register is shown in IPU Channels Buffer 2 Ready 0 Register (IPU_CH_BUF2_RDY0), and the register fields are described in IPU Channels Buffer 2 Ready 0 Register (IPU_CH_BUF2_RDY0).

Address: 260_0000h base + 288h offset = 260_0288h



IPU_CH_BUF2_RDY0 field descriptions

Field	Description
31–29 -	This field is reserved. Reserved.
28 DMA_CH_BUF2_RDY_28	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.
27 DMA_CH_ALT_BUF1_RDY_27	buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 buffer 2 is not ready. 1 buffer 2 is ready.
26–24 -	This field is reserved. Reserved.
23 DMA_CH_BUF2_RDY_23	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.
22 -	This field is reserved. Reserved.
21 DMA_CH_BUF2_RDY_21	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.

Table continues on the next page...

IPU_CH_BUF2_RDY0 field descriptions (continued)

Field	Description
20–14 -	This field is reserved. Reserved.
13 DMA_CH_BUF2_RDY_13	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.
12–11 -	This field is reserved. Reserved.
10 DMA_CH_BUF2_RDY_10	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.
9 DMA_CH_BUF2_RDY_9	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.
8 DMA_CH_BUF2_RDY_8	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.
7–3 -	This field is reserved. Reserved.
2 DMA_CH_BUF2_RDY_2	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.
1 -	This field is reserved. Reserved.
0 DMA_CH_BUF2_RDY_0	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.

38.5.85 IPU Channels Buffer 2 Ready 1 Register (IPU_CH_BUF2_RDY1)

The register contains buffer 2 ready control information for 32 IPU's DMA channels (63-32). This register can be a write one to set or a write one to clear according to the **IPU_CH_BUF2_RDY1_CLR bit.**

The register is shown in IPU Channels Buffer 2 Ready 1 Register (IPU_CH_BUF2_RDY1), and the register fields are described in IPU Channels Buffer 2 Ready 1 Register (IPU_CH_BUF2_RDY1).

Address: 260_0000h base + 28Ch offset = 260_028Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_CH_BUF2_RDY_x																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CH_BUF2_RDY1 field descriptions

Field	Description
DMA_CH_BUF2_RDY_x	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.

38.5.86 IDMAC Configuration Register (IPU_IDMAC_CONF)

Address: 260_0000h base + 8000h offset = 260_8000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved						USED_BUFS_EN_R		USED_BUFS_MAX_R			USED_BUFS_EN_W	USED_BUFS_MAX_W			P_ENDIAN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										RDI	WIDPT	MAX_REQ_READ			
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1

IPU_IDMAC_CONF field descriptions

Field	Description
31–26 -	This field is reserved. Reserved, should be cleared.
25 USED_BUFS_EN_R	Enables the limit on the number of pending non real time read requests.
24–21 USED_BUFS_MAX_R	Limit the number of pending non real time read requests. The value can be between 0 to 8. This field has no affect if USED_BUFS_EN_R is cleared
20 USED_BUFS_EN_W	Enables the limit on the number of pending non real time write requests.

Table continues on the next page...

IPU_IDMAC_CONF field descriptions (continued)

Field	Description
19–17 USED_BUFS_ MAX_W	Limit the number of pending non real time write requests. The value can be between 0 to 6. This field has no affect if USED_BUFS_EN_W is cleared
16 P_ENDIAN	Pixel Endianness. The pixel Endianness must not be changed while any of the IDMAC channels is enabled. 0 little endian 1 Big endian
15–6 -	This field is reserved. Reserved, should be cleared.
5 RDI	Read Data Interleaving. This bit must match the slave read data interleaving support. If the AXI slave connected to the IPU supports read data interleaving then this bit must be set. If the AXI slave does not support read data interleaving then the IDMAC can utilize this and issue more address phases on read. In that case it is recommended to have this bit cleared. 0 The AXI slave does not support read data interleaving 1 The AXI slave supports read data interleaving
4–3 WIDPT	Write Interleaving Depth These 2 bits define the Write Interleaving Depth of the AXI port. This bits should be configured by the user according to the AXI slave's Write Interleaving Depth. WIDPT defines the maximal number of active bursts (yet to be responded) with different IDs. IDMAC will block data phase if the next data's ID is new (no such ID active) and the number of active IDs is equal to WIDPT. 00 Write Interleaving Depth of 1 01 Write Interleaving Depth of 2 10 Write Interleaving Depth of 3 11 Write Interleaving Depth of 4
MAX_REQ_ READ	Maximum Read Requests. This fields sets the maximum pending requests allowed in the AXI Read requests queue.

38.5.87 IDMAC Channel Enable 1 Register (IPU_IDMAC_CH_EN_1)

Address: 260_0000h base + 8004h offset = 260_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IDMAC_CH_EN_31	Reserved	IDMAC_CH_EN_29	IDMAC_CH_EN_28	IDMAC_CH_EN_27	IDMAC_CH_EN_26	IDMAC_CH_EN_25	IDMAC_CH_EN_24	IDMAC_CH_EN_23	IDMAC_CH_EN_22	IDMAC_CH_EN_21	IDMAC_CH_EN_20	IDMAC_CH_EN_19	IDMAC_CH_EN_18	IDMAC_CH_EN_17	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IDMAC_CH_EN_15	IDMAC_CH_EN_14	IDMAC_CH_EN_13	IDMAC_CH_EN_12	IDMAC_CH_EN_11	IDMAC_CH_EN_10	IDMAC_CH_EN_9	IDMAC_CH_EN_8	Reserved		IDMAC_CH_EN_5	Reserved	IDMAC_CH_EN_3	IDMAC_CH_EN_2	IDMAC_CH_EN_1	IDMAC_CH_EN_0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IDMAC_CH_EN_1 field descriptions

Field	Description
31 IDMAC_CH_EN_31	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
30 -	This field is reserved. Reserved.
29 IDMAC_CH_EN_29	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
28 IDMAC_CH_EN_28	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
27 IDMAC_CH_EN_27	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled

Table continues on the next page...

IPU_IDMAC_CH_EN_1 field descriptions (continued)

Field	Description
26 IDMAC_CH_EN_26	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
25 IDMAC_CH_EN_25	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
24 IDMAC_CH_EN_24	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
23 IDMAC_CH_EN_23	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
22 IDMAC_CH_EN_22	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
21 IDMAC_CH_EN_21	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
20 IDMAC_CH_EN_20	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
19 IDMAC_CH_EN_19	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
18 IDMAC_CH_EN_18	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
17 IDMAC_CH_EN_17	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
16 -	This field is reserved. Reserved.
15 IDMAC_CH_EN_15	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
14 IDMAC_CH_EN_14	IDMAC Channel enable bit [i]

Table continues on the next page...

IPU_IDMAC_CH_EN_1 field descriptions (continued)

Field	Description
	0 IDMAC channel is disabled 1 IDMAC channel is enabled
13 IDMAC_CH_EN_13	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
12 IDMAC_CH_EN_12	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
11 IDMAC_CH_EN_11	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
10 IDMAC_CH_EN_10	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
9 IDMAC_CH_EN_9	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
8 IDMAC_CH_EN_8	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
7–6 -	This field is reserved. Reserved.
5 IDMAC_CH_EN_5	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
4 -	This field is reserved. Reserved.
3 IDMAC_CH_EN_3	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
2 IDMAC_CH_EN_2	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
1 IDMAC_CH_EN_1	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
0 IDMAC_CH_EN_0	IDMAC Channel enable bit [i]

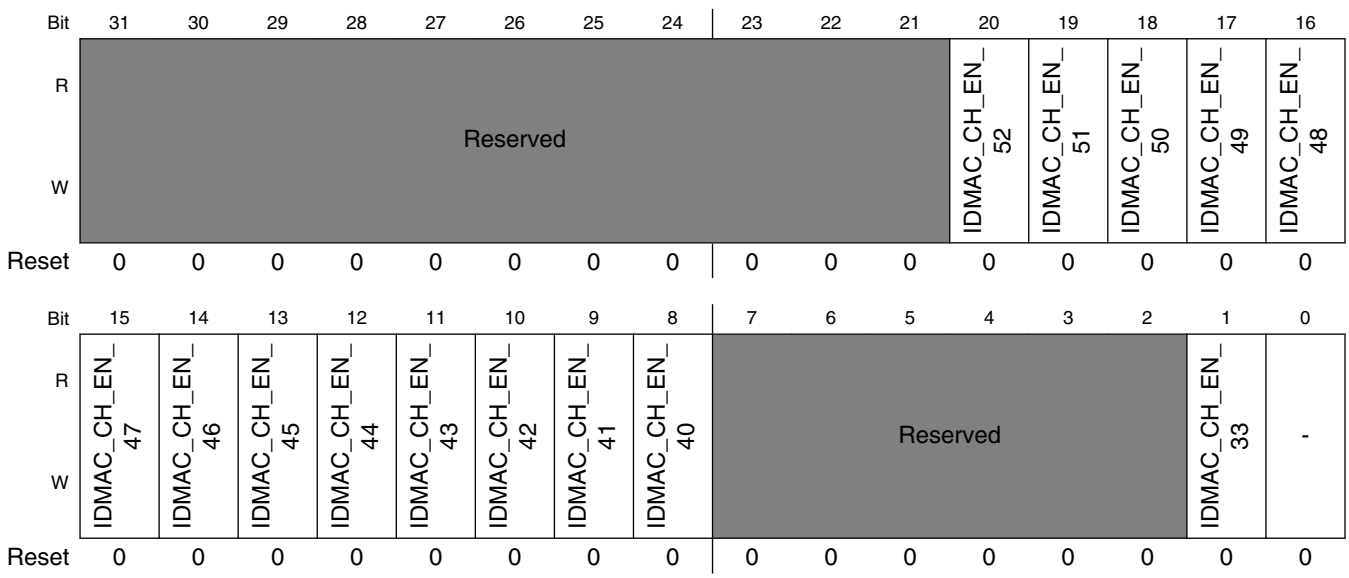
Table continues on the next page...

IPU_IDMAC_CH_EN_1 field descriptions (continued)

Field	Description
	0 IDMAC channel is disabled 1 IDMAC channel is enabled

38.5.88 IDMAC Channel Enable 2 Register (IPU_IDMAC_CH_EN_2)

Address: 260_0000h base + 8008h offset = 260_8008h



IPU_IDMAC_CH_EN_2 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved.
20 IDMAC_CH_EN_52	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
19 IDMAC_CH_EN_51	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
18 IDMAC_CH_EN_50	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
17 IDMAC_CH_EN_49	IDMAC Channel enable bit [i]

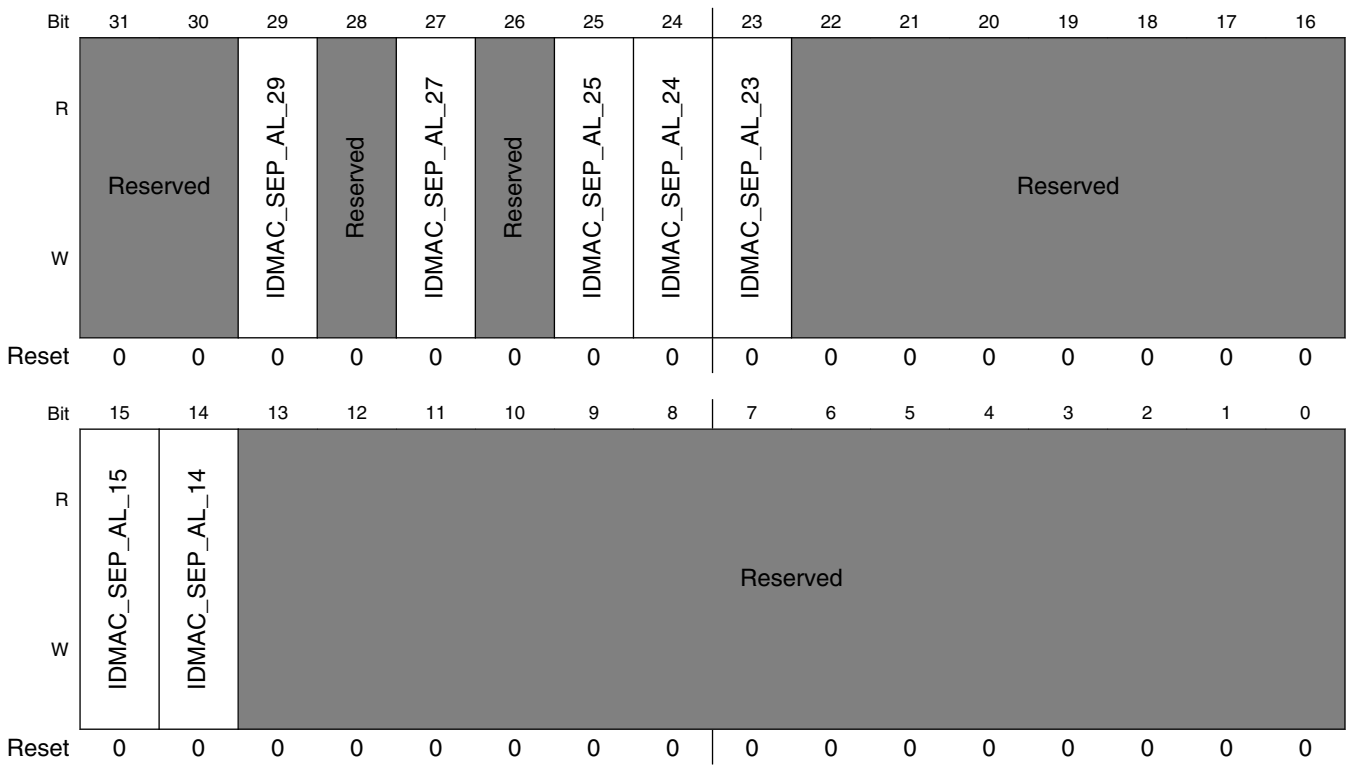
Table continues on the next page...

IPU_IDMAC_CH_EN_2 field descriptions (continued)

Field	Description
	0 IDMAC channel is disabled 1 IDMAC channel is enabled
16 IDMAC_CH_EN_48	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
15 IDMAC_CH_EN_47	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
14 IDMAC_CH_EN_46	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
13 IDMAC_CH_EN_45	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
12 IDMAC_CH_EN_44	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
11 IDMAC_CH_EN_43	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
10 IDMAC_CH_EN_42	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
9 IDMAC_CH_EN_41	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
8 IDMAC_CH_EN_40	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
7–2 -	This field is reserved. Reserved.
1 IDMAC_CH_EN_33	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
0 -	Reserved.

38.5.89 IDMAC Separate Alpha Indication Register (IPU_IDMAC_SEP_ALPHA)

Address: 260_0000h base + 800Ch offset = 260_800Ch



IPU_IDMAC_SEP_ALPHA field descriptions

Field	Description
31–30 -	This field is reserved. Reserved.
29 IDMAC_SEP_AL_29	IDMAC Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
28 -	This field is reserved. Reserved.
27 IDMAC_SEP_AL_27	IDMAC Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel.

Table continues on the next page...

IPU_IDMAC_SEP_ALPHA field descriptions (continued)

Field	Description
	In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
26 -	This field is reserved. Reserved.
25 IDMAC_SEP_AL_25	IDMAC Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
24 IDMAC_SEP_AL_24	IDMAC Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
23 IDMAC_SEP_AL_23	IDMAC Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
22–16 -	This field is reserved. Reserved.
15 IDMAC_SEP_AL_15	IDMAC Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
14 IDMAC_SEP_AL_14	IDMAC Separate alpha indication bit [i]

Table continues on the next page...

IPU_IDMAC_SEP_ALPHA field descriptions (continued)

Field	Description
	A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
-	This field is reserved. Reserved.

38.5.90 IDMAC Alternate Separate Alpha Indication Register (IPU_IDMAC_ALT_SEP_ALPHA)

Address: 260_0000h base + 8010h offset = 260_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		IDMAC_ALT_SEP_AL_29	Reserved				IDMAC_ALT_SEP_AL_24	IDMAC_ALT_SEP_AL_23	Reserved						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IDMAC_ALT_SEP_ALPHA field descriptions

Field	Description
31–30 -	This field is reserved. Reserved.
29 IDMAC_ALT_SEP_AL_29	IDMAC Alternate Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding.

Table continues on the next page...

IPU_IDMAC_ALT_SEP_ALPHA field descriptions (continued)

Field	Description
	For channels that may have alternate flow and may use separate alpha. This bit indicates the mode of the alpha for the alternate flow 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
28–25 -	This field is reserved. Reserved.
24 IDMAC_ALT_SEP_AL_24	IDMAC Alternate Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. For channels that may have alternate flow and may use separate alpha. This bit indicates the mode of the alpha for the alternate flow 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
23 IDMAC_ALT_SEP_AL_23	IDMAC Alternate Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. For channels that may have alternate flow and may use separate alpha. This bit indicates the mode of the alpha for the alternate flow 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
-	This field is reserved. Reserved.

38.5.91 IDMAC Channel Priority 1 Register (IPU_IDMAC_CH_PRI_1)

Address: 260_0000h base + 8014h offset = 260_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		IDMAC_CH_PRI_29	IDMAC_CH_PRI_28	IDMAC_CH_PRI_27	IDMAC_CH_PRI_26	IDMAC_CH_PRI_25	IDMAC_CH_PRI_24	IDMAC_CH_PRI_23	IDMAC_CH_PRI_22	IDMAC_CH_PRI_21	IDMAC_CH_PRI_20	Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IDMAC_CH_PRI_15	IDMAC_CH_PRI_14	IDMAC_CH_PRI_13	IDMAC_CH_PRI_12	IDMAC_CH_PRI_11	IDMAC_CH_PRI_10	IDMAC_CH_PRI_9	IDMAC_CH_PRI_8	Reserved		IDMAC_CH_PRI_5	Reserved	IDMAC_CH_PRI_3	IDMAC_CH_PRI_2	IDMAC_CH_PRI_1	IDMAC_CH_PRI_0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IDMAC_CH_PRI_1 field descriptions

Field	Description
31–30 -	This field is reserved. Reserved.
29 IDMAC_CH_ PRI_29	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
28 IDMAC_CH_ PRI_28	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
27 IDMAC_CH_ PRI_27	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
26 IDMAC_CH_ PRI_26	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority

Table continues on the next page...

IPU_IDMAC_CH_PRI_1 field descriptions (continued)

Field	Description
25 IDMAC_CH_ PRI_25	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
24 IDMAC_CH_ PRI_24	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
23 IDMAC_CH_ PRI_23	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
22 IDMAC_CH_ PRI_22	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
21 IDMAC_CH_ PRI_21	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
20 IDMAC_CH_ PRI_20	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
19–16 -	This field is reserved. Reserved.
15 IDMAC_CH_ PRI_15	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
14 IDMAC_CH_ PRI_14	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
13 IDMAC_CH_ PRI_13	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
12 IDMAC_CH_ PRI_12	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
11 IDMAC_CH_ PRI_11	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
10 IDMAC_CH_ PRI_10	IDMAC Channel enable bit [i]

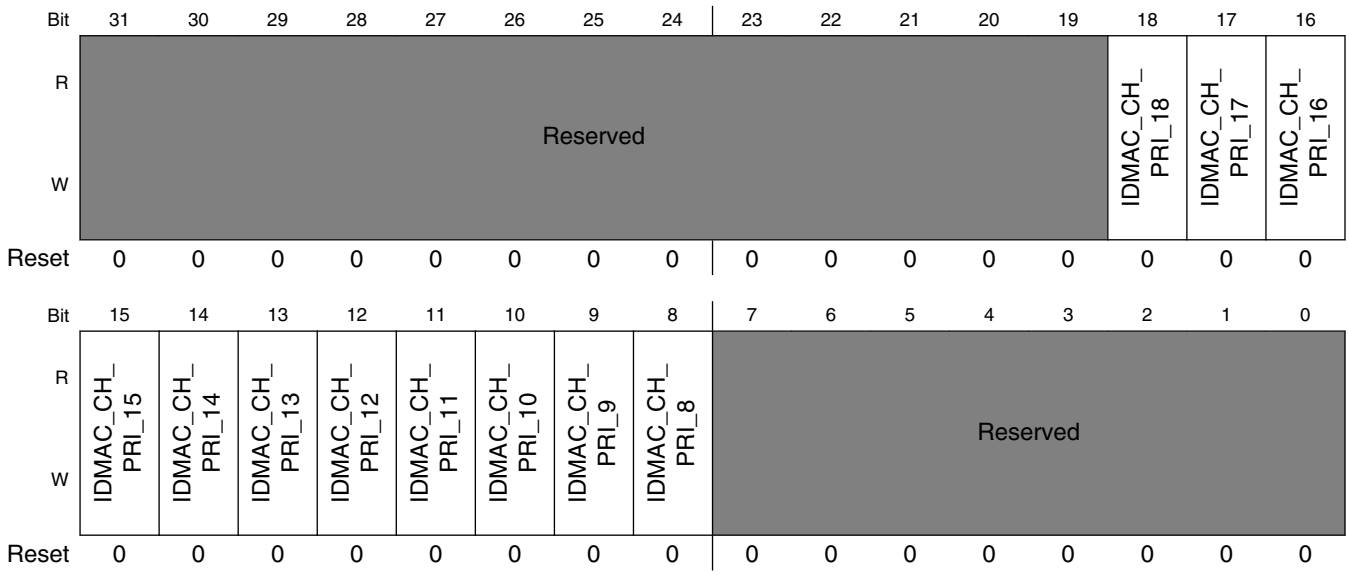
Table continues on the next page...

IPU_IDMAC_CH_PRI_1 field descriptions (continued)

Field	Description
	0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
9 IDMAC_CH_ PRI_9	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
8 IDMAC_CH_ PRI_8	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
7–6 -	This field is reserved. Reserved.
5 IDMAC_CH_ PRI_5	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
4 -	This field is reserved. Reserved.
3 IDMAC_CH_ PRI_3	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
2 IDMAC_CH_ PRI_2	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
1 IDMAC_CH_ PRI_1	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
0 IDMAC_CH_ PRI_0	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority

38.5.92 IDMAC Channel Priority 2 Register (IPU_IDMAC_CH_PRI_2)

Address: 260_0000h base + 8018h offset = 260_8018h



IPU_IDMAC_CH_PRI_2 field descriptions

Field	Description
31–19 -	This field is reserved. Reserved.
18 IDMAC_CH_PRI_18	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
17 IDMAC_CH_PRI_17	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
16 IDMAC_CH_PRI_16	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
15 IDMAC_CH_PRI_15	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
14 IDMAC_CH_PRI_14	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority

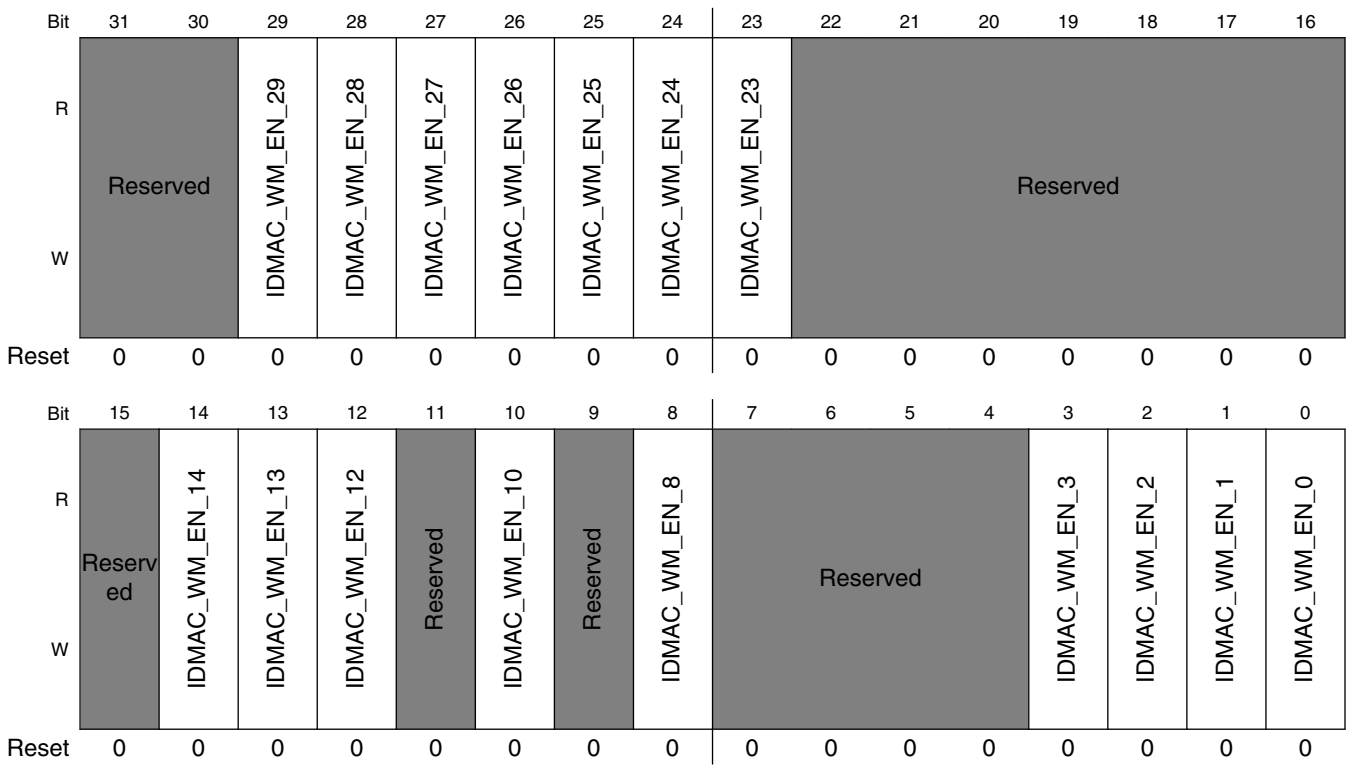
Table continues on the next page...

IPU_IDMAC_CH_PRI_2 field descriptions (continued)

Field	Description
13 IDMAC_CH_ PRI_13	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
12 IDMAC_CH_ PRI_12	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
11 IDMAC_CH_ PRI_11	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
10 IDMAC_CH_ PRI_10	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
9 IDMAC_CH_ PRI_9	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
8 IDMAC_CH_ PRI_8	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
-	This field is reserved. Reserved.

38.5.93 IDMAC Channel Watermark Enable 1 Register (IPU_IDMAC_WM_EN_1)

Address: 260_0000h base + 801Ch offset = 260_801Ch



IPU_IDMAC_WM_EN_1 field descriptions

Field	Description
31–30 -	This field is reserved. Reserved.
29 IDMAC_WM_EN_29	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
28 IDMAC_WM_EN_28	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
27 IDMAC_WM_EN_27	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
26 IDMAC_WM_EN_26	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled

Table continues on the next page...

IPU_IDMAC_WM_EN_1 field descriptions (continued)

Field	Description
25 IDMAC_WM_EN_25	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
24 IDMAC_WM_EN_24	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
23 IDMAC_WM_EN_23	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
22–15 -	This field is reserved. Reserved.
14 IDMAC_WM_EN_14	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
13 IDMAC_WM_EN_13	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
12 IDMAC_WM_EN_12	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
11 -	This field is reserved. Reserved.
10 IDMAC_WM_EN_10	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
9 -	This field is reserved. Reserved.
8 IDMAC_WM_EN_8	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
7–4 -	This field is reserved. Reserved.
3 IDMAC_WM_EN_3	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
2 IDMAC_WM_EN_2	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled

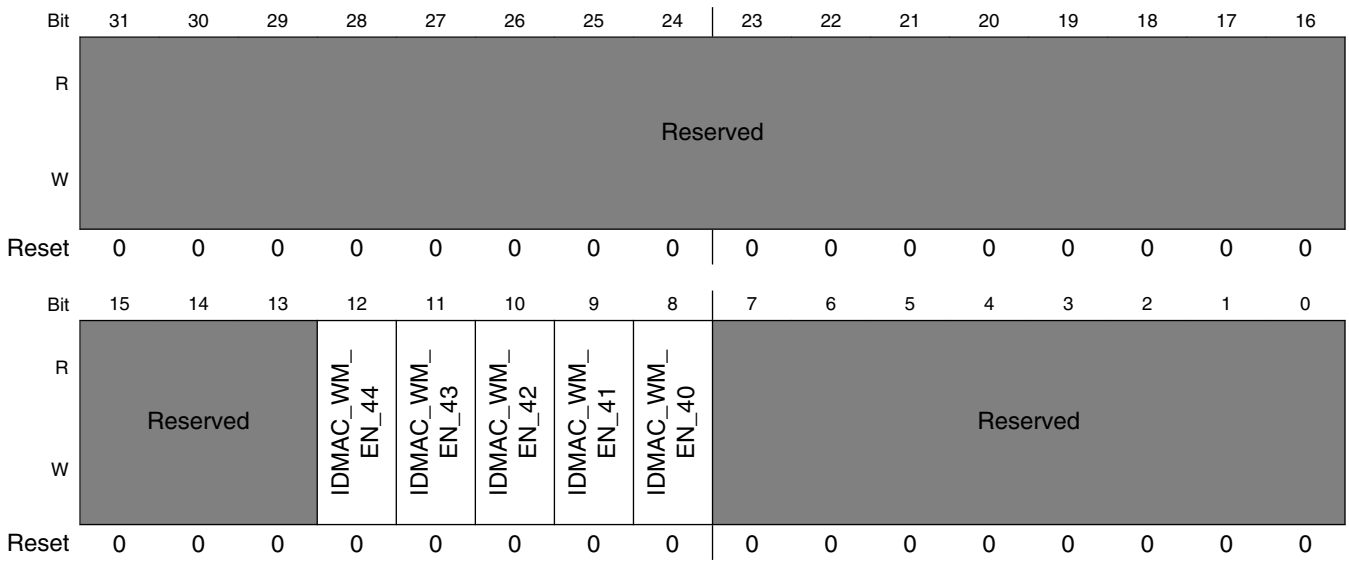
Table continues on the next page...

IPU_IDMAC_WM_EN_1 field descriptions (continued)

Field	Description
1 IDMAC_WM_ EN_1	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
0 IDMAC_WM_ EN_0	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled

38.5.94 IDMAC Channel Watermark Enable 2 Register (IPU_IDMAC_WM_EN_2)

Address: 260_0000h base + 8020h offset = 260_8020h



IPU_IDMAC_WM_EN_2 field descriptions

Field	Description
31–13 -	This field is reserved. Reserved.
12 IDMAC_WM_ EN_44	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
11 IDMAC_WM_ EN_43	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled

Table continues on the next page...

IPU_IDMAC_WM_EN_2 field descriptions (continued)

Field	Description
10 IDMAC_WM_EN_42	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
9 IDMAC_WM_EN_41	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
8 IDMAC_WM_EN_40	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
-	This field is reserved. Reserved.

38.5.95 IDMAC Channel Lock Enable 1Register (IPU_IDMAC_LOCK_EN_1)

Address: 260_0000h base + 8024h offset = 260_8024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved										IDMAC_LOCK_EN_28		IDMAC_LOCK_EN_27		IDMAC_LOCK_EN_23	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IDMAC_LOCK_EN_22		IDMAC_LOCK_EN_21		IDMAC_LOCK_EN_20		IDMAC_LOCK_EN_15		IDMAC_LOCK_EN_14		IDMAC_LOCK_EN_12		IDMAC_LOCK_EN_11		IDMAC_LOCK_EN_5	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IDMAC_LOCK_EN_1 field descriptions

Field	Description
31–22 -	This field is reserved. Reserved.
21–20 IDMAC_LOCK_EN_28	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.

Table continues on the next page...

IPU_IDMAC_LOCK_EN_1 field descriptions (continued)

Field	Description
19–18 IDMAC_LOCK_EN_27	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
17–16 IDMAC_LOCK_EN_23	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
15–14 IDMAC_LOCK_EN_22	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
13–12 IDMAC_LOCK_EN_21	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
11–10 IDMAC_LOCK_EN_20	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
9–8 IDMAC_LOCK_EN_15	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
7–6 IDMAC_LOCK_EN_14	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.

Table continues on the next page...

IPU_IDMAC_LOCK_EN_1 field descriptions (continued)

Field	Description
5–4 IDMAC_LOCK_EN_12	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
3–2 IDMAC_LOCK_EN_11	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
IDMAC_LOCK_EN_5	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.

38.5.96 IDMAC Channel Lock Enable 2Register (IPU_IDMAC_LOCK_EN_2)

Address: 260_0000h base + 8028h offset = 260_8028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				IDMAC_LOCK_50		IDMAC_LOCK_49		IDMAC_LOCK_48		IDMAC_LOCK_47		IDMAC_LOCK_46		IDMAC_LOCK_45	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IDMAC_LOCK_EN_2 field descriptions

Field	Description
31–12 -	This field is reserved. Reserved
11–10 IDMAC_LOCK_50	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request.

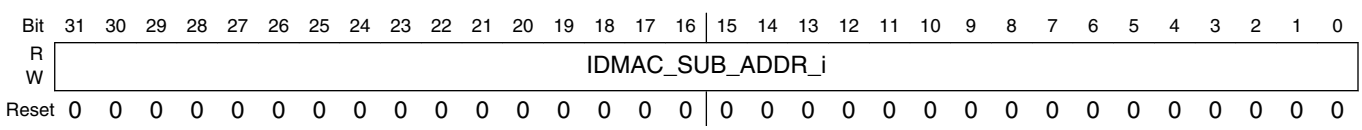
Table continues on the next page...

IPU_IDMAC_LOCK_EN_2 field descriptions (continued)

Field	Description
	10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
9–8 IDMAC_LOCK_49	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
7–6 IDMAC_LOCK_48	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
5–4 IDMAC_LOCK_47	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
3–2 IDMAC_LOCK_46	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
IDMAC_LOCK_45	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.

38.5.97 IDMAC Channel Alternate Address 0 Register (IPU_IDMAC_SUB_ADDR_0)

Address: 260_0000h base + 802Ch offset = 260_802Ch

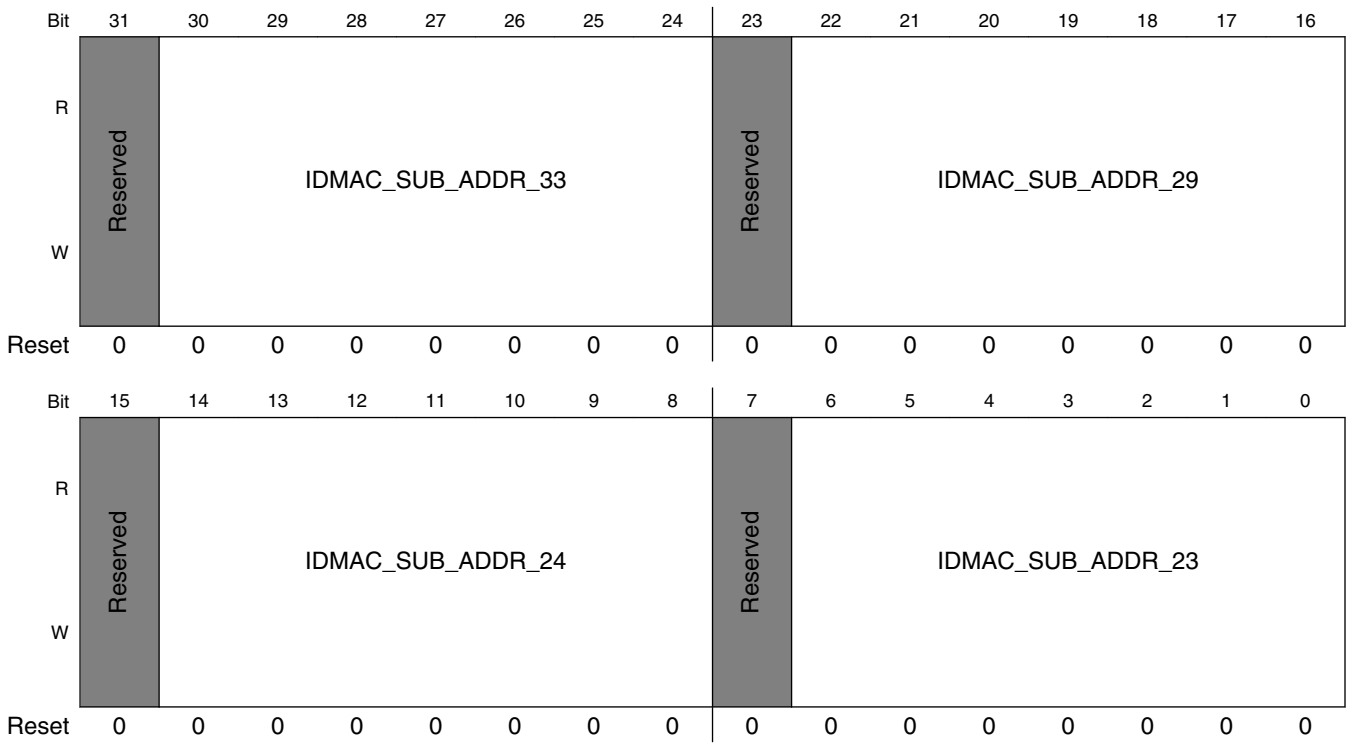


IPU_IDMAC_SUB_ADDR_0 field descriptions

Field	Description
IDMAC_SUB_ADDR_i	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR_i. The user must set each IDMAC_SUB_ADDR_i field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.

38.5.98 IDMAC Channel Alternate Address 1 Register (IPU_IDMAC_SUB_ADDR_1)

Address: 260_0000h base + 8030h offset = 260_8030h



IPU_IDMAC_SUB_ADDR_1 field descriptions

Field	Description
31 -	This field is reserved. Reserved.
30–24 IDMAC_SUB_ADDR_33	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR_i. The user must set each IDMAC_SUB_ADDR_i field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
23 -	This field is reserved. Reserved.
22–16 IDMAC_SUB_ADDR_29	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR_i. The user must set each IDMAC_SUB_ADDR_i field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.

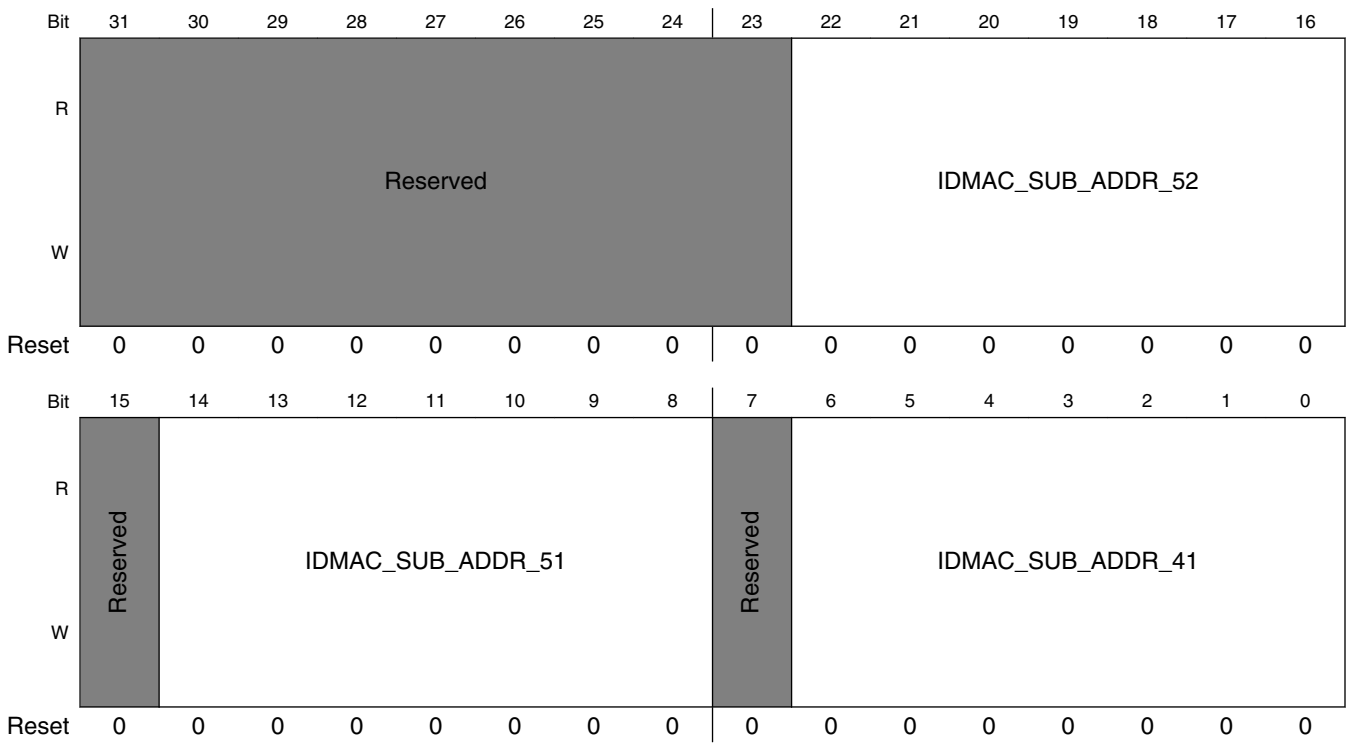
Table continues on the next page...

IPU_IDMAC_SUB_ADDR_1 field descriptions (continued)

Field	Description
15 -	This field is reserved. Reserved.
14–8 IDMAC_SUB_ADDR_24	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR_i. The user must set each IDMAC_SUB_ADDR_i field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
7 -	This field is reserved. Reserved.
IDMAC_SUB_ADDR_23	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR_i. The user must set each IDMAC_SUB_ADDR_i field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.

38.5.99 IDMAC Channel Alternate Address 2 Register (IPU_IDMAC_SUB_ADDR_2)

Address: 260_0000h base + 8034h offset = 260_8034h



IPU_IDMAC_SUB_ADDR_2 field descriptions

Field	Description
31–23 -	This field is reserved. Reserved.

Table continues on the next page...

IPU_IDMAC_SUB_ADDR_2 field descriptions (continued)

Field	Description
22–16 IDMAC_SUB_ADDR_52	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR_i. The user must set each IDMAC_SUB_ADDR_i field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
15 -	This field is reserved. Reserved.
14–8 IDMAC_SUB_ADDR_51	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR_i. The user must set each IDMAC_SUB_ADDR_i field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
7 -	This field is reserved. Reserved.
IDMAC_SUB_ADDR_41	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR_i. The user must set each IDMAC_SUB_ADDR_i field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.

38.5.100 IDMAC Channel Alternate Address 3 Register (IPU_IDMAC_SUB_ADDR_3)

Address: 260_0000h base + 8038h offset = 260_8038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	IDMAC_SUB_ADDR_27							Reserved	IDMAC_SUB_ADDR_13						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

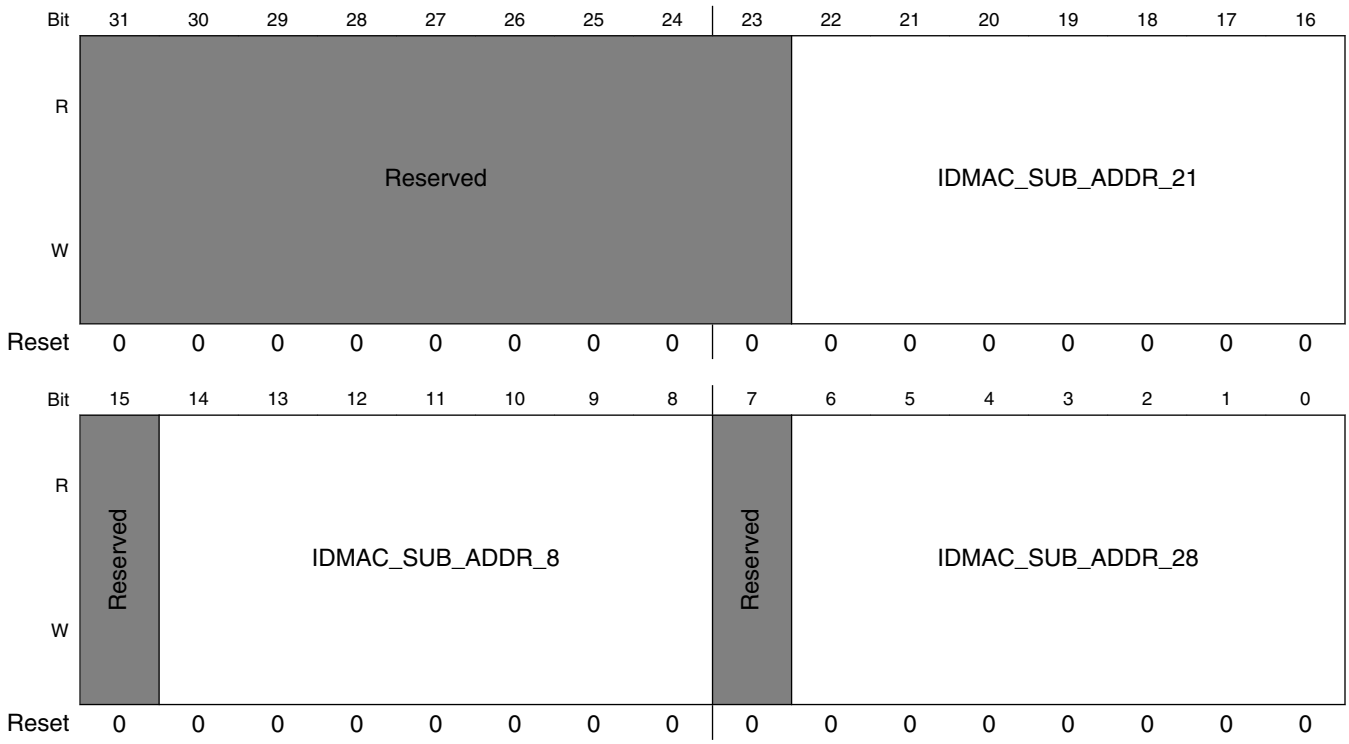
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	IDMAC_SUB_ADDR_10							Reserved	IDMAC_SUB_ADDR_9						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IDMAC_SUB_ADDR_3 field descriptions

Field	Description
31 -	This field is reserved. Reserved.
30–24 IDMAC_SUB_ADDR_27	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR_i. The user must set each IDMAC_SUB_ADDR_i field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
23 -	This field is reserved. Reserved.
22–16 IDMAC_SUB_ADDR_13	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR_i. The user must set each IDMAC_SUB_ADDR_i field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
15 -	This field is reserved. Reserved.
14–8 IDMAC_SUB_ADDR_10	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR_i. The user must set each IDMAC_SUB_ADDR_i field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
7 -	This field is reserved. Reserved.
IDMAC_SUB_ADDR_9	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR_i. The user must set each IDMAC_SUB_ADDR_i field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.

38.5.101 IDMAC Channel Alternate Address 4 Register (IPU_IDMAC_SUB_ADDR_4)

Address: 260_0000h base + 803Ch offset = 260_803Ch

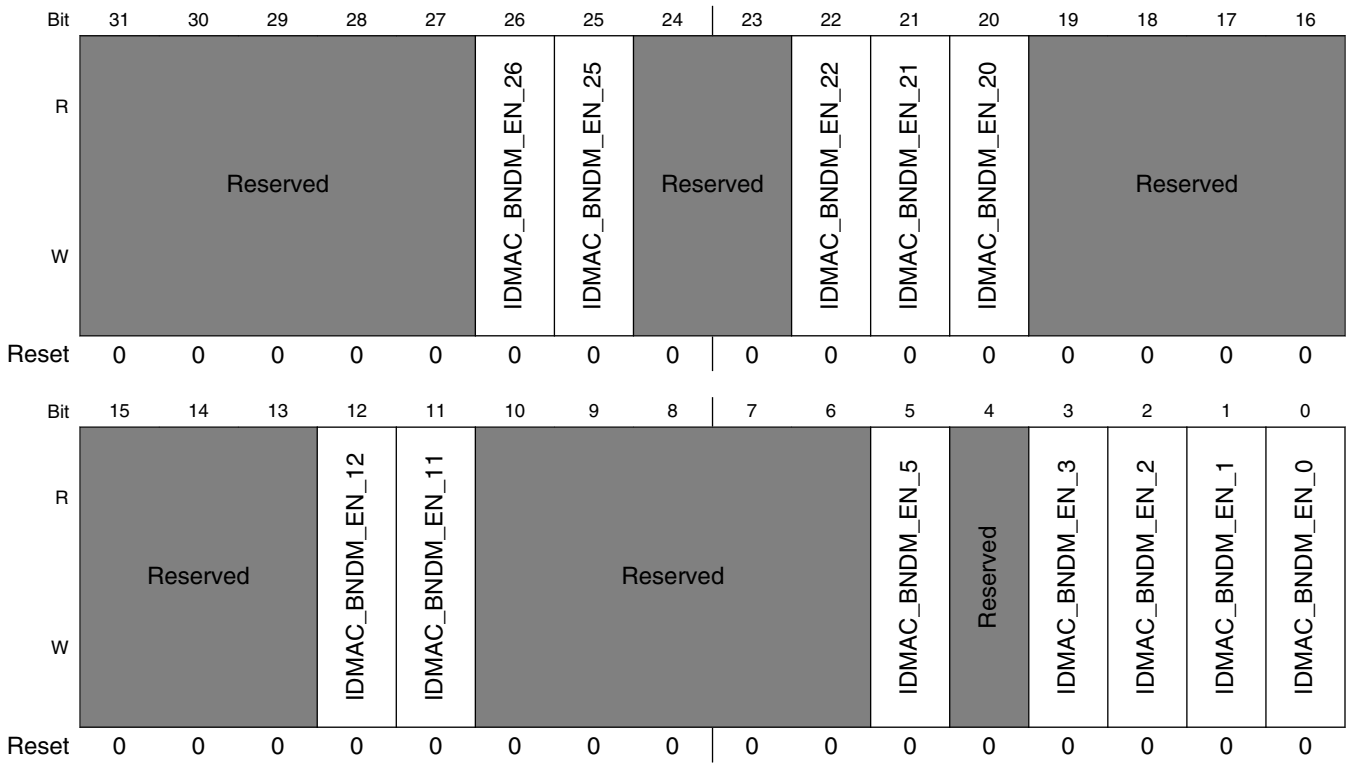


IPU_IDMAC_SUB_ADDR_4 field descriptions

Field	Description
31–23 -	This field is reserved. Reserved.
22–16 IDMAC_SUB_ADDR_21	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR_i. The user must set each IDMAC_SUB_ADDR_i field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
15 -	This field is reserved. Reserved.
14–8 IDMAC_SUB_ADDR_8	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR_i. The user must set each IDMAC_SUB_ADDR_i field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
7 -	This field is reserved. Reserved.
IDMAC_SUB_ADDR_28	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR_i. The user must set each IDMAC_SUB_ADDR_i field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.

38.5.102 IDMAC Band Mode Enable 1 Register (IPU_IDMAC_BNDM_EN_1)

Address: 260_0000h base + 8040h offset = 260_8040h



IPU_IDMAC_BNDM_EN_1 field descriptions

Field	Description
31–27 -	This field is reserved. Reserved.
26 IDMAC_BNDM_EN_26	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
25 IDMAC_BNDM_EN_25	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode

Table continues on the next page...

IPU_IDMAC_BNDM_EN_1 field descriptions (continued)

Field	Description
24–23 -	This field is reserved. Reserved.
22 IDMAC_BNDM_EN_22	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
21 IDMAC_BNDM_EN_21	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
20 IDMAC_BNDM_EN_20	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
19–13 -	This field is reserved. Reserved.
12 IDMAC_BNDM_EN_12	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
11 IDMAC_BNDM_EN_11	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
10–6 -	This field is reserved. Reserved.
5 IDMAC_BNDM_EN_5	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration.

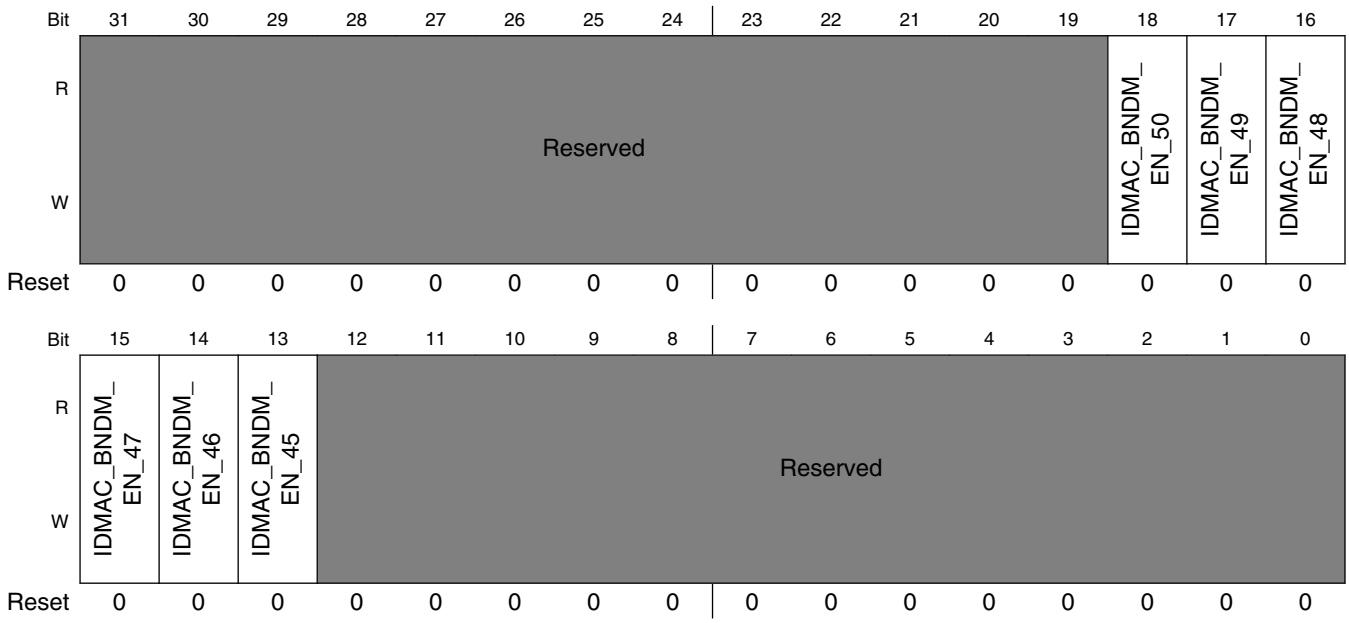
Table continues on the next page...

IPU_IDMAC_BNDM_EN_1 field descriptions (continued)

Field	Description
	0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
4 -	This field is reserved. Reserved.
3 IDMAC_BNDM_EN_3	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
2 IDMAC_BNDM_EN_2	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
1 IDMAC_BNDM_EN_1	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
0 IDMAC_BNDM_EN_0	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode

38.5.103 IDMAC Band Mode Enable 2 Register (IPU_IDMAC_BNDM_EN_2)

Address: 260_0000h base + 8044h offset = 260_8044h



IPU_IDMAC_BNDM_EN_2 field descriptions

Field	Description
31–19 -	This field is reserved. Reserved.
18 IDMAC_BNDM_EN_50	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
17 IDMAC_BNDM_EN_49	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
16 IDMAC_BNDM_EN_48	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration.

Table continues on the next page...

IPU_IDMAC_BNDM_EN_2 field descriptions (continued)

Field	Description
	0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
15 IDMAC_BNDM_EN_47	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
14 IDMAC_BNDM_EN_46	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
13 IDMAC_BNDM_EN_45	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
-	This field is reserved. Reserved.

38.5.104 IDMAC Scroll Coordinations Register (IPU_IDMAC_SC_CORD)

Address: 260_0000h base + 8048h offset = 260_8048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				SX0												Reserved					SY0											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IDMAC_SC_CORD field descriptions

Field	Description
31–28 -	This field is reserved. Reserved, should be cleared.
27–16 SX0	Scroll X coordination This field indicates the X coordinate of the scroll. This parameter has an affect on continuos scroll mode only. Units are pixels. When the alpha buffer resides in a separate buffer and the alpha is different than 8 BPP this field should be 0.

Table continues on the next page...

IPU_IDMAC_SC_CORD field descriptions (continued)

Field	Description
15–11 -	This field is reserved. Reserved, should be cleared.
SY0	Scroll Y coordination This field indicates the Y coordinate of the scroll. This parameter has an affect on continuos scroll mode only. Units are pixels. When the alpha buffer resides in a separate buffer and the alpha is different than 8 BPP this field should be 0.

38.5.105 IDMAC Scroll Coordinations Register 1 (IPU_IDMAC_SC_CORD_1)

Address: 260_0000h base + 804Ch offset = 260_804Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				SX1												Reserved					SY1											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

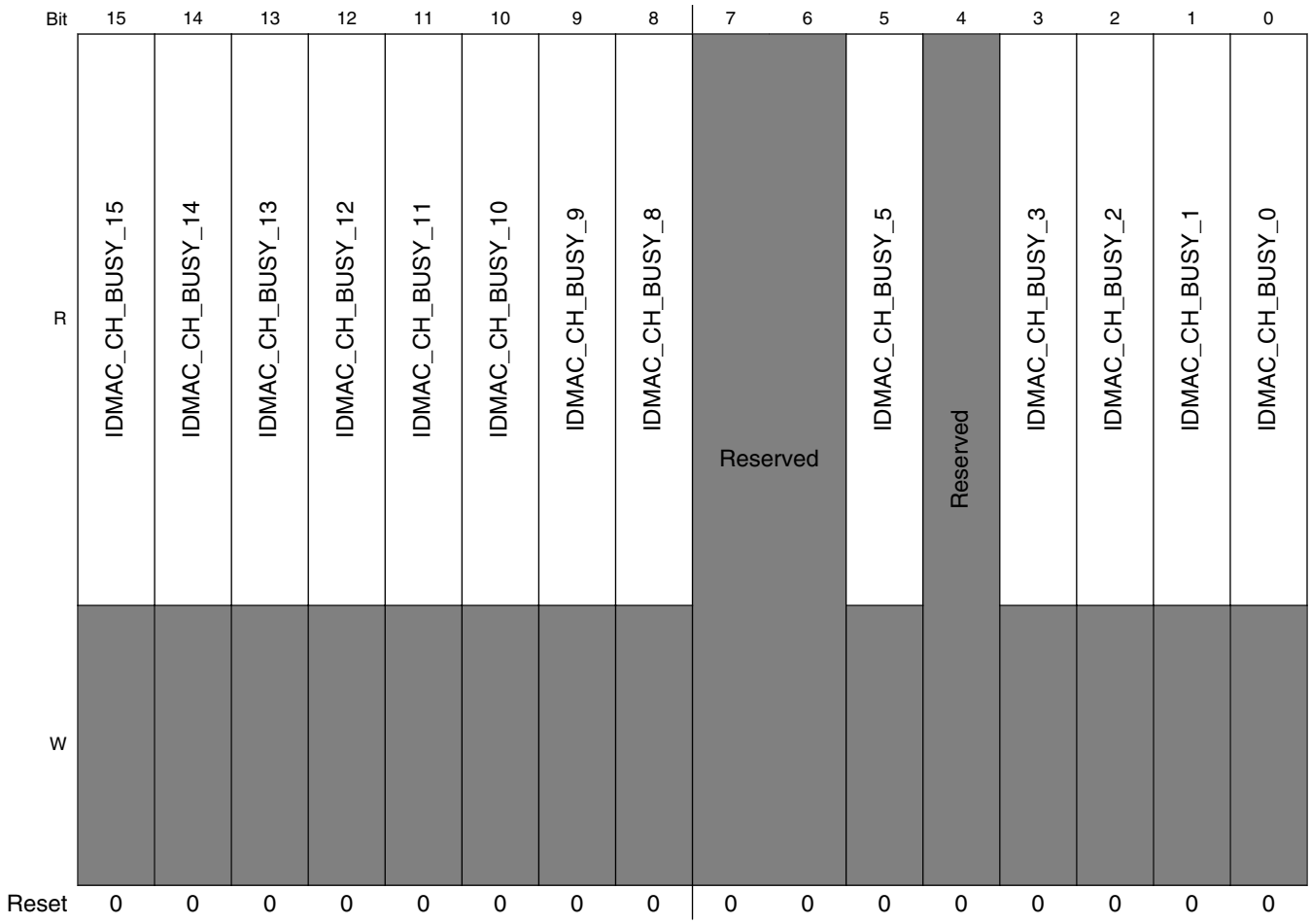
IPU_IDMAC_SC_CORD_1 field descriptions

Field	Description
31–28 -	This field is reserved. Reserved, should be cleared.
27–16 SX1	Scroll X coordination (2nd set) This field indicates the X coordinate of the scroll. This parameter has an affect on continuos scroll mode only. Units are pixels. When the alpha buffer resides in a separate buffer and the alpha is different than 8 BPP this field should be 0.
15–11 -	This field is reserved. Reserved, should be cleared.
SY1	Scroll Y coordination (2nd set) This field indicates the Y coordinate of the scroll. This parameter has an affect on continuos scroll mode only. Units are pixels. When the alpha buffer resides in a separate buffer and the alpha is different than 8 BPP this field should be 0.

38.5.106 IDMAC Channel Busy 1 Register (IPU_IDMAC_CH_BUSY_1)

Address: 260_0000h base + 8100h offset = 260_8100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_CH_BUSY_	Reserved	IDMAC_CH_BUSY_29	IDMAC_CH_BUSY_28	IDMAC_CH_BUSY_27	IDMAC_CH_BUSY_26	IDMAC_CH_BUSY_25	IDMAC_CH_BUSY_24	IDMAC_CH_BUSY_23	IDMAC_CH_BUSY_22	IDMAC_CH_BUSY_21	IDMAC_CH_BUSY_20	Reserved	IDMAC_CH_BUSY_18	IDMAC_CH_BUSY_17	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



IPU_IDMAC_CH_BUSY_1 field descriptions

Field	Description
31 IDMAC_CH_BUSY_	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
30 -	This field is reserved. Reserved.
29 IDMAC_CH_BUSY_29	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
28 IDMAC_CH_BUSY_28	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC.

Table continues on the next page...

IPU_IDMAC_CH_BUSY_1 field descriptions (continued)

Field	Description
	0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
27 IDMAC_CH_ BUSY_27	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
26 IDMAC_CH_ BUSY_26	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
25 IDMAC_CH_ BUSY_25	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
24 IDMAC_CH_ BUSY_24	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
23 IDMAC_CH_ BUSY_23	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
22 IDMAC_CH_ BUSY_22	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
21 IDMAC_CH_ BUSY_21	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy

Table continues on the next page...

IPU_IDMAC_CH_BUSY_1 field descriptions (continued)

Field	Description
20 IDMAC_CH_BUSY_20	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
19 -	This field is reserved. Reserved.
18 IDMAC_CH_BUSY_18	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
17 IDMAC_CH_BUSY_17	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
16 -	This field is reserved. Reserved.
15 IDMAC_CH_BUSY_15	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
14 IDMAC_CH_BUSY_14	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
13 IDMAC_CH_BUSY_13	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
12 IDMAC_CH_BUSY_12	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC.

Table continues on the next page...

IPU_IDMAC_CH_BUSY_1 field descriptions (continued)

Field	Description
	0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
11 IDMAC_CH_BUSY_11	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
10 IDMAC_CH_BUSY_10	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
9 IDMAC_CH_BUSY_9	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
8 IDMAC_CH_BUSY_8	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
7–6 -	This field is reserved. Reserved.
5 IDMAC_CH_BUSY_5	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
4 -	This field is reserved. Reserved.
3 IDMAC_CH_BUSY_3	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
2 IDMAC_CH_BUSY_2	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC.

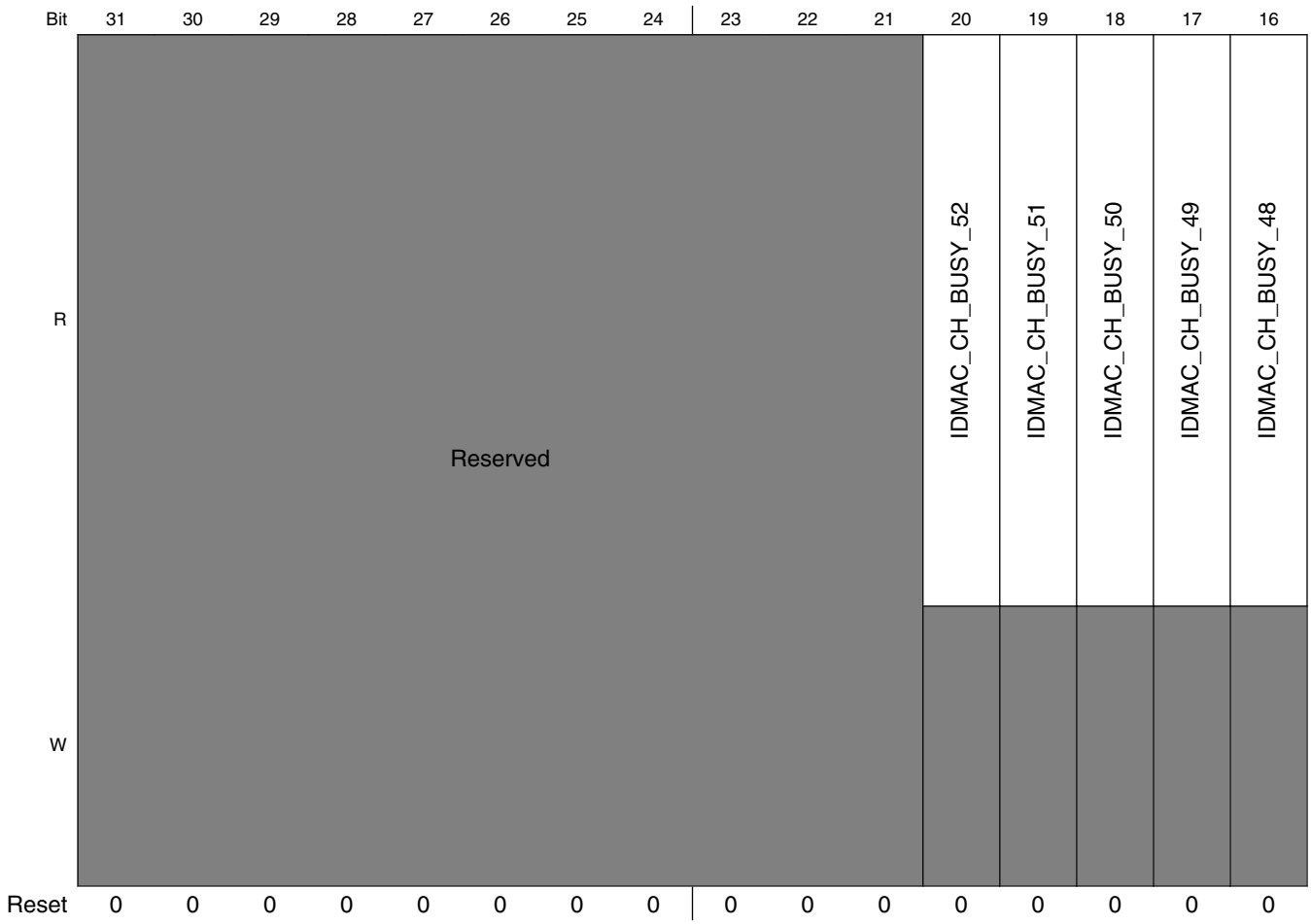
Table continues on the next page...

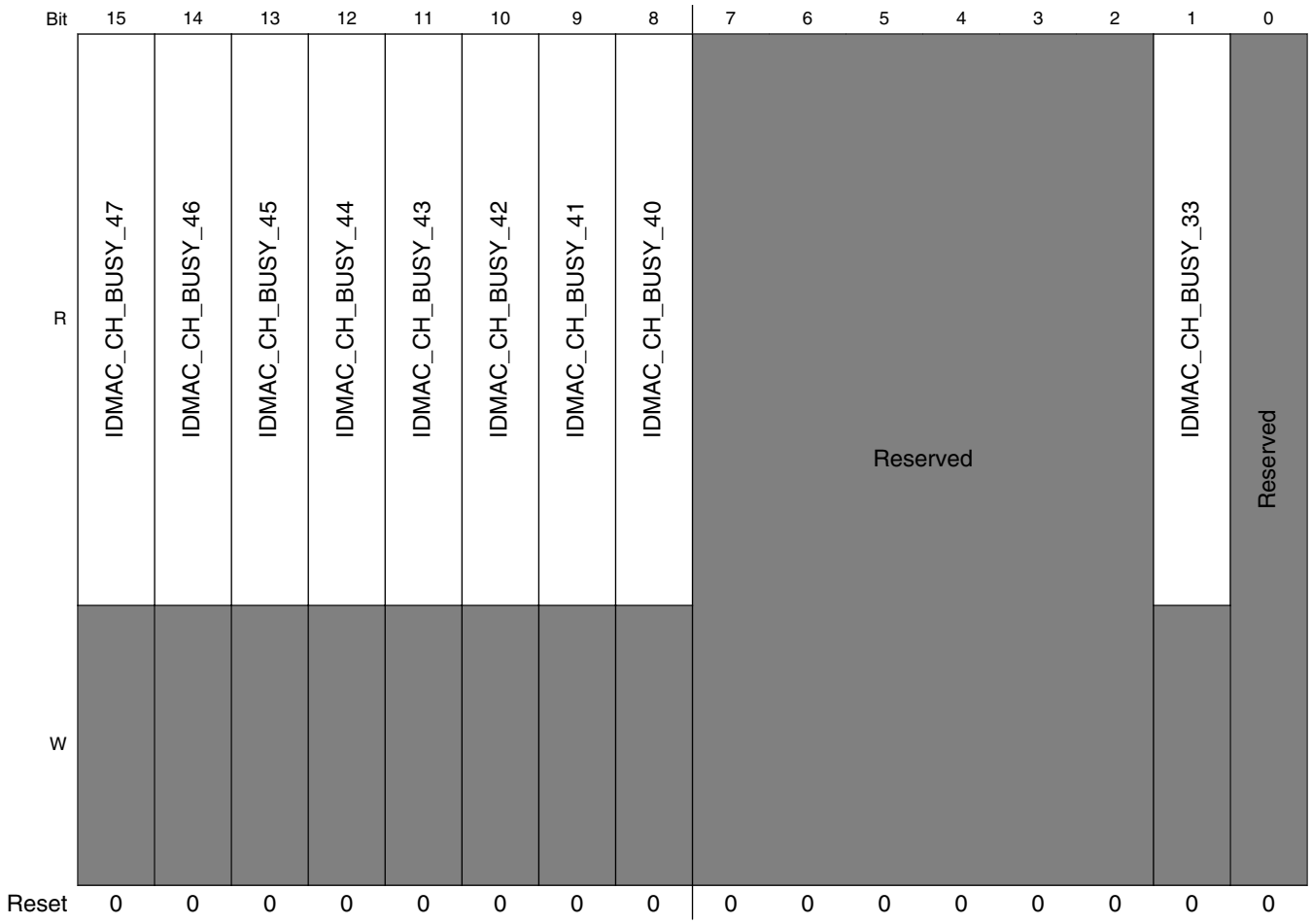
IPU_IDMAC_CH_BUSY_1 field descriptions (continued)

Field	Description
	This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
1 IDMAC_CH_ BUSY_1	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
0 IDMAC_CH_ BUSY_0	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy

38.5.107 IDMAC Channel Busy 2 Register (IPU_IDMAC_CH_BUSY_2)

Address: 260_0000h base + 8104h offset = 260_8104h





IPU_IDMAC_CH_BUSY_2 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved.
20 IDMAC_CH_ BUSY_52	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
19 IDMAC_CH_ BUSY_51	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
18 IDMAC_CH_ BUSY_50	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC.

Table continues on the next page...

IPU_IDMAC_CH_BUSY_2 field descriptions (continued)

Field	Description
	0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
17 IDMAC_CH_ BUSY_49	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
16 IDMAC_CH_ BUSY_48	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
15 IDMAC_CH_ BUSY_47	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
14 IDMAC_CH_ BUSY_46	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
13 IDMAC_CH_ BUSY_45	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
12 IDMAC_CH_ BUSY_44	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
11 IDMAC_CH_ BUSY_43	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy

Table continues on the next page...

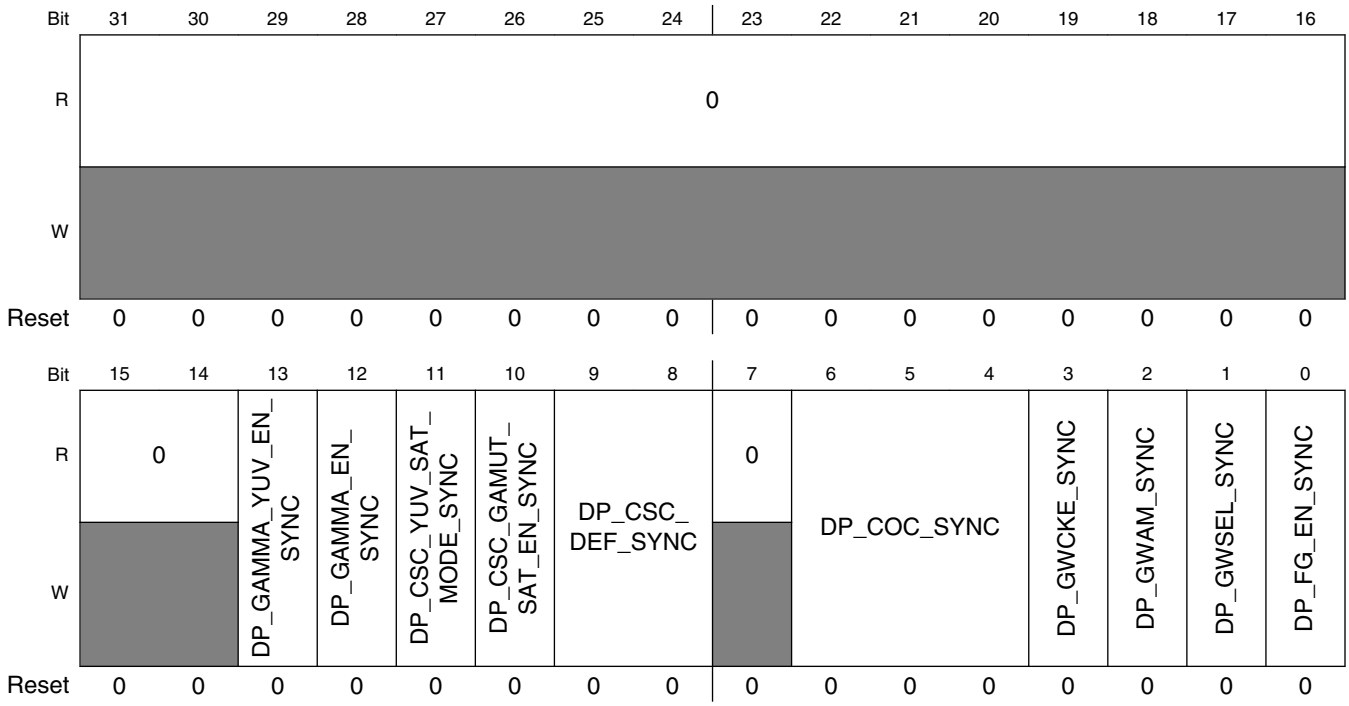
IPU_IDMAC_CH_BUSY_2 field descriptions (continued)

Field	Description
10 IDMAC_CH_ BUSY_42	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
9 IDMAC_CH_ BUSY_41	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
8 IDMAC_CH_ BUSY_40	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
7–2 -	This field is reserved. Reserved.
1 IDMAC_CH_ BUSY_33	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
0 -	This field is reserved. Reserved.

38.5.108 DP Common Configuration Sync Flow Register (IPU_DP_COM_CONF_SYNC)

This register contains common configuration parameters for the DP.

Address: 260_0000h base + 1_8000h offset = 261_8000h



IPU_DP_COM_CONF_SYNC field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
13 DP_GAMMA_YUV_EN_SYNC	GAMMA's YUV mode enable for sync flow 0 YUV mode is OFF. 1 YUV mode is ON.
12 DP_GAMMA_EN_SYNC	GAMMA_EN - Gamma correction block enable bit 0 disable 1 enable
11 DP_CSC_YUV_SAT_MODE_SYNC	CSC_YUV_SAT_MODE YUV saturation mode for color space conversion 0 Y/U/V range 0 -255 1 Y range 16-235, U/V range 16-240

Table continues on the next page...

IPU_DP_COM_CONF_SYNC field descriptions (continued)

Field	Description
10 DP_CSC_ GAMUT_SAT_ EN_SYNC	CSC_GAMUT_SAT_EN Indicate if GAMUT saturation is enabled 0 disable GAMUT mapping 1 enable GAMUT mapping
9–8 DP_CSC_DEF_ SYNC	CSC_DEF Enable or disable Color Space Conversion. 00 CSC disable 01 CSC enable after combining 10 CSC enable before combining on BG channel 11 CSC enable before combining on FG channel
7 Reserved	This read-only field is reserved and always has the value 0.
6–4 DP_COC_SYNC	COC - Cursor Operation Control Controls the format of the cursor and the type of arithmetic operations 000 Transparent, cursor is disabled. 001 Full cursor. 010 Reversed cursor. 011 AND between full plane and cursor. 100 Reserved 101 OR between full plane and cursor. 110 XOR between full plane and cursor. 111 Reserved.
3 DP_GWCKE_ SYNC	GWCKE - Graphic Window Color Keying Enable Enable or disable graphic window color keying. 1 Enable color keying of graphic window 0 Disable color keying of graphic window
2 DP_GWAM_ SYNC	GWAM - Graphic Window Alpha Mode Select the use of Alpha to be global or local. 1 Global Alpha. 0 Local Alpha.
1 DP_GWSEL_ SYNC	GWSEL - Graphic Window Select Select graphic window to be on partial plane or full plane. 1 Graphic window is partial plane. 0 Graphic window is full plane.
0 DP_FG_EN_ SYNC	FG_EN - partial plane Enable. This bit enables the partial plane channel. 1 partial plane channel is enabled. 0 partial plane channel is disabled.

38.5.109 DP Graphic Window Control Sync Flow Register (IPU_DP_Graph_Wind_CTRL_SYNC)

This register contains common configuration parameters for the DP.

Address: 260_0000h base + 1_8004h offset = 261_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DP_GWAV_SYNC								DP_GWCKR_SYNC								DP_GWCKG_SYNC								DP_GWCKB_SYNC							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_Graph_Wind_CTRL_SYNC field descriptions

Field	Description
31–24 DP_GWAV_SYNC	GWAV - Graphic Window Alpha Value Defines the alpha value of graphic window used for alpha blending between graphic window and full plane. The Value the number that writing to GWAV register. The Actual Value the number, that insert to calculation in Combining Module. If Value = < 0.5- 1/256 (01111111) then Actual Value = Value. If Value >= 0.5 (10000000), then Actual Value = Value + 1/256. 00000000 Actual value is 00000000; Graphic window totally opaque i.e. overlay on LCD screen 01111111 Actual value is 01111111; 10000000 Actual value is 10000001 10000001 Actual value is 10000010 11111110 Actual value is 11111111 11111111 Actual value is 100000000;Graphic window totally transparent i.e. not displayed on LCD screen
23–16 DP_GWCKR_SYNC	GWCKR - Graphic Window Color Keying Red Component Defines the red component of graphic window color keying. 00000000 No red 11111111 Full red
15–8 DP_GWCKG_SYNC	GWCKG - Graphic Window Color Keying Green Component Defines the green component of graphic window color keying. 00000000 No Green 11111111 Full Green
DP_GWCKB_SYNC	GWCKB - Graphic Window Color Keying Blue Component Defines the blue component of graphic window color keying. 00000000 No blue 11111111 Full blue

38.5.110 DP Partial Plane Window Position Sync Flow Register (IPU_DP_FG_POS_SYNC)

The DP partial plane Window Position Register is used to determine the starting position of the partial plane window relative to BG window position.

This Register is used for the synchronous flow only.

Address: 260_0000h base + 1_8008h offset = 261_8008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					DP_FGXP_SYNC											0					DP_FGYP_SYNC										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_FG_POS_SYNC field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26–16 DP_FGXP_ SYNC	FGXP partial plane Window X Position. partial plane Window X Position - Specifies the number of pixels between the start of full plane window X position and the beginning of the first data of new line.
15–11 Reserved	This read-only field is reserved and always has the value 0.
DP_FGYP_ SYNC	FGYP partial plane window Y position partial plane Window Y Position - Specifies the number of lines between the start of full plane windows Y position and the beginning of the first data.

38.5.111 DP Cursor Position and Size Sync Flow Register (IPU_DP_CUR_POS_SYNC)

The LCD Cursor Position Register is used to determine the starting position of the cursor relative to BG windows position.

Address: 260_0000h base + 1_800Ch offset = 261_800Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DP_CYP_SYNC					DP_CYH_SYNC											DP_CXP_SYNC					DP_CXW_SYNC										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CUR_POS_SYNC field descriptions

Field	Description
31–27 DP_CYP_SYNC	CYP - Cursor Y Position Represents the cursors vertical starting position Y in pixel count (from 0 to CYH).Live View Resolution Mode.
26–16 DP_CYH_SYNC	CYH - Cursor Height Specifies the height of the hardware cursor in pixels.
15–11 DP_CXP_SYNC	CXP - Cursor X Position Represents the cursors horizontal starting position X in pixel count (from 0 to CXW).
DP_CXW_SYNC	CXW - Cursor Width. Specifies the width of the hardware cursor in pixels.

38.5.112 DP Color Cursor Mapping Sync Flow Register (IPU_DP_CUR_MAP_SYNC)

The LCD Color Cursor Mapping Register defines the color of the cursor in passive or TFT color modes.

Address: 260_0000h base + 1_8010h offset = 261_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DP_CUR_COL_B_SYNC								DP_CUR_COL_G_SYNC								DP_CUR_COL_R_SYNC							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CUR_MAP_SYNC field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value 0.
23–16 DP_CUR_COL_B_SYNC	CUR_COL_B - Cursor Blue Field Defines the Blue component of the cursor color in color mode 00000000 No Blue. 11111111 Full Blue.
15–8 DP_CUR_COL_G_SYNC	CUR_COL_G - Cursor Green Field Defines the Green component of the cursor color in color mode 00000000 No Green. 11111111 Full Green.
DP_CUR_COL_R_SYNC	CUR_COL_B - Cursor Red Field Defines the Red component of the cursor color in color mode

Table continues on the next page...

IPU_DP_CUR_MAP_SYNC field descriptions (continued)

Field	Description
	00000000 No Red. 11111111 Full Red.

38.5.113 DP Gamma Constants Sync Flow Register i (IPU_DP_GAMMA_C_SYNC_i)

This registers contains CONSTANTi parameters used for gamma correction inside the display processor (DP).

Address: 260_0000h base + 1_8014h offset = 261_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							DP_GAMMA_C_SYNC_2i_1									0							DP_GAMMA_C_SYNC_2i								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_GAMMA_C_SYNC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 DP_GAMMA_C_SYNC_2i_1	CONSTANTi+1 parameter of Gamma Correction.
15–9 Reserved	This read-only field is reserved and always has the value 0.
DP_GAMMA_C_SYNC_2i	CONSTANTi parameter of Gamma Correction.

38.5.114 DP Gamma Correction Slope Sync Flow Register i (IPU_DP_GAMMA_S_SYNC_i)

This registers contains SLOPEi parameters used for Gamma Correction inside the display processor (DP).

Address: 260_0000h base + 1_8034h offset = 261_8034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DP_GAMMA_S_SYNC_4i_3								DP_GAMMA_S_SYNC_4i_2								DP_GAMMA_S_SYNC_4i_1								DP_GAMMA_S_SYNC_4i							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_GAMMA_S_SYNC_i field descriptions

Field	Description
31–24 DP_GAMMA_S_SYNC_4i_3	SLOPE<4*i+3> parameter of Gamma Correction.
23–16 DP_GAMMA_S_SYNC_4i_2	SLOPE<4*i+2> parameter of Gamma Correction.
15–8 DP_GAMMA_S_SYNC_4i_1	SLOPE<4*i+1> parameter of Gamma Correction.
DP_GAMMA_S_SYNC_4i	SLOPE<4*i> parameter of Gamma Correction.

38.5.115 DP Color Space Conversion Control Sync Flow Registers (IPU_DP_CSCA_SYNC_i)

Address: 260_0000h base + 1_8044h offset = 261_8044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	0						DP_CSC_A_SYNC_2i_1										0						DP_CSC_A_SYNC_2i										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CSCA_SYNC_i field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DP_CSCA_SYNC_i field descriptions (continued)

Field	Description
25–16 DP_CSC_A_ SYNC_2i_1	A<2*i+1> parameter of color conversion
15–10 Reserved	This read-only field is reserved and always has the value 0.
DP_CSC_A_ SYNC_2i	A<2*i> parameter of color conversion.

38.5.116 DP Color Conversion Control Sync Flow Register 0 (IPU_DP_SCS_SYNC_0)

Address: 260_0000h base + 1_8054h offset = 261_8054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DP_CSC_ S0_SYNC		DP_CSC_B0_SYNC													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						DP_CSC_A8_SYNC									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_SCS_SYNC_0 field descriptions

Field	Description
31–30 DP_CSC_S0_ SYNC	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
29–16 DP_CSC_B0_ SYNC	B0 parameter of color conversion.
15–10 Reserved	This read-only field is reserved and always has the value 0.
DP_CSC_A8_ SYNC	A9 parameter of color conversion.

38.5.117 DP Color Conversion Control Sync Flow Register 1 (IPU_DP_SCS_SYNC_1)

Address: 260_0000h base + 1_8058h offset = 261_8058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DP_CSC_S2_SYNC		DP_CSC_B2_SYNC													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DP_CSC_S1_SYNC		DP_CSC_B1_SYNC													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_SCS_SYNC_1 field descriptions

Field	Description
31–30 DP_CSC_S2_SYNC	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
29–16 DP_CSC_B2_SYNC	B0 parameter of color conversion.
15–14 DP_CSC_S1_SYNC	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
DP_CSC_B1_SYNC	B0 parameter of color conversion.

38.5.118 DP Cursor Position and Size Alternate Register (IPU_DP_CUR_POS_ALT)

The LCD Cursor Position Register is used to determine the starting position of the cursor relative to BG windows position for the alternative flow.

Address: 260_0000h base + 1_805Ch offset = 261_805Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DP_CYP_SYNC_ALT					DP_CYH_SYNC_ALT											DP_CXP_SYNC_ALT					DP_CXW_SYNC_ALT										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

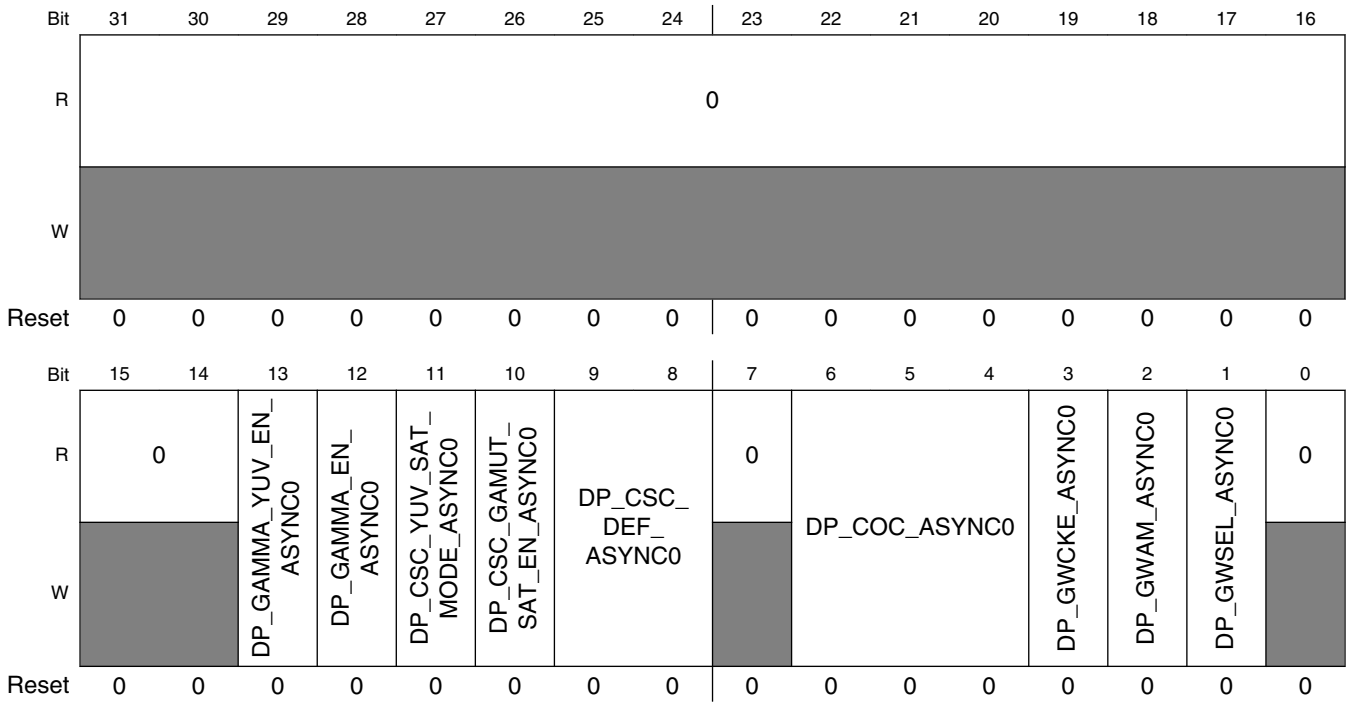
IPU_DP_CUR_POS_ALT field descriptions

Field	Description
31–27 DP_CYP_SYNC_ALT	CYP_ALT - Cursor Y Position Represents the cursors vertical starting position Y in pixel count (from 0 to CYH).Live View Resolution Mode for the alternative flow.
26–16 DP_CYH_SYNC_ALT	CYH_ALT - Cursor Height Specifies the height of the hardware cursor in pixels.
15–11 DP_CXP_SYNC_ALT	CXP_ALT - Cursor X Position Represents the cursors horizontal starting position X in pixel count (from 0 to CXW) for the alternative flow.
DP_CXW_SYNC_ALT	CXW_ALT - Cursor Width. Specifies the width of the hardware cursor in pixels for the alternative flow.

38.5.119 DP Common Configuration Async 0 Flow Register (IPU_DP_COM_CONF_ASYNC0)

This register contains common configuration parameters for the DP.

Address: 260_0000h base + 1_8060h offset = 261_8060h



IPU_DP_COM_CONF_ASYNC0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
13 DP_GAMMA_YUV_EN_ASYNC0	GAMMA's YUV mode enable for async flow 0 0 YUV mode is OFF. 1 YUV mode is ON.
12 DP_GAMMA_EN_ASYNC0	GAMMA_EN - Gamma correction block enable bit 0 disable 1 enable
11 DP_CSC_YUV_SAT_MODE_ASYNC0	CSC_YUV_SAT_MODE YUV saturation mode for color space conversion 0 Y/U/V range 0 -255 1 Y range 16-235, U/V range 16-240

Table continues on the next page...

IPU_DP_COM_CONF_ASYNC0 field descriptions (continued)

Field	Description
10 DP_CSC_ GAMUT_SAT_ EN_ASYNC0	CSC_GAMUT_SAT_EN Indicate if GAMUT saturation is enabled 0 disable GAMUT mapping 1 enable GAMUT mapping
9–8 DP_CSC_DEF_ ASYNC0	CSC_DEF Enable or disable Color Space Conversion. 00 CSC disable 01 CSC enable after combining 10 CSC enable before combining on BG channel 11 CSC enable before combining on FG channel
7 Reserved	This read-only field is reserved and always has the value 0.
6–4 DP_COC_ ASYNC0	COC - Cursor Operation Control Controls the format of the cursor and the type of arithmetic operations 000 Transparent, cursor is disabled. 001 Full cursor. 010 Reversed cursor. 011 AND between full plane and cursor. 100 Reserved 101 OR between full plane and cursor. 110 XOR between full plane and cursor. 111 Reserved
3 DP_GWCKE_ ASYNC0	GWCKE - Graphic Window Color Keying Enable Enable or disable graphic window color keying. 1 Enable color keying of graphic window 0 Disable color keying of graphic window
2 DP_GWAM_ ASYNC0	GWAM - Graphic Window Alpha Mode Select the use of Alpha to be global or local. 1 Global Alpha. 0 Local Alpha.
1 DP_GWSEL_ ASYNC0	GWSEL - Graphic Window Select Select graphic window to be on partial plane or full plane. 1 Graphic window is partial plane. 0 Graphic window is full plane.5
0 Reserved	This read-only field is reserved and always has the value 0.

38.5.120 DP Graphic Window Control Async 0 Flow Register (IPU_DP_GRAPH_WIND_CTRL_ASYNC0)

This register contains common configuration parameters for the DP.

Address: 260_0000h base + 1_8064h offset = 261_8064h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DP_GWAV_ASYNC0								DP_GWCKR_ASYNC0								DP_GWCKG_ASYNC0								DP_GWCKB_ASYNC0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_GRAPH_WIND_CTRL_ASYNC0 field descriptions

Field	Description
31–24 DP_GWAV_ASYNC0	GWAV - Graphic Window Alpha Value Defines the alpha value of graphic window used for alpha blending between graphic window and full plane. The Value the number that writing to GWAV register. The Actual Value the number, that insert to calculation in Combining Module. If Value = < 0.5- 1/256 (01111111) then Actual Value = Value. If Value >= 0.5 (10000000), then Actual Value = Value + 1/256. 00000000 Actual value is 00000000; Graphic window totally opaque i.e. overlay on LCD screen 01111111 Actual value is 01111111; 10000000 Actual value is 10000001 10000001 Actual value is 10000010 11111110 Actual value is 11111111 11111111 Actual value is 100000000;Graphic window totally transparent i.e. not displayed on LCD screen
23–16 DP_GWCKR_ASYNC0	GWCKR - Graphic Window Color Keying Red Component Defines the red component of graphic window color keying. 00000000 No red 11111111 Full red
15–8 DP_GWCKG_ASYNC0	GWCKG - Graphic Window Color Keying Green Component Defines the green component of graphic window color keying. 00000000 No Green 11111111 Full Green
DP_GWCKB_ASYNC0	GWCKB - Graphic Window Color Keying Blue Component Defines the blue component of graphic window color keying. 00000000 No blue 11111111 Full blue

38.5.121 DP Partial Plane Window Position Async 0 Flow Register (IPU_DP_FG_POS_ASYNC0)

The DP partial plane Window Position Register is used to determine the starting position of the partial plane window relative to BG window position.

This Register is used for the synchronous flow only.

Address: 260_0000h base + 1_8068h offset = 261_8068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					DP_FGXP_ASYNC0											0					DP_FGYP_ASYNC0										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_FG_POS_ASYNC0 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26–16 DP_FGXP_ASYNC0	FGXP partial plane Window X Position. partial plane Window X Position - Specifies the number of pixels between the start of full plane window X position and the beginning of the first data of new line.
15–11 Reserved	This read-only field is reserved and always has the value 0.
DP_FGYP_ASYNC0	FGYP partial plane window Y position partial plane Window Y Position - Specifies the number of lines between the start of full plane windows Y position and the beginning of the first data.

38.5.122 DP Cursor Position and Size Async 0 Flow Register (IPU_DP_CUR_POS_ASYNC0)

The LCD Cursor Position Register is used to determine the starting position of the cursor relative to BG windows position.

Address: 260_0000h base + 1_806Ch offset = 261_806Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DP_CYP_ASYNC0					DP_CYH_ASYNC0											DP_CXP_ASYNC0					DP_CXW_ASYNC0										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

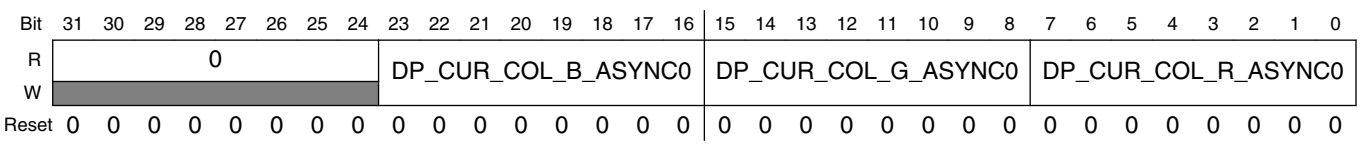
IPU_DP_CUR_POS_ASYNC0 field descriptions

Field	Description
31–27 DP_CYP_ ASYNC0	CYP - Cursor Y Position Represents the cursors vertical starting position Y in pixel count (from 0 to CYH).Live View Resolution Mode.
26–16 DP_CYH_ ASYNC0	CYH - Cursor Height Specifies the height of the hardware cursor in pixels.
15–11 DP_CXP_ ASYNC0	CXP - Cursor X Position Represents the cursors horizontal starting position X in pixel count (from 0 to CXW).
DP_CXW_ ASYNC0	CXW - Cursor Width. Specifies the width of the hardware cursor in pixels.

38.5.123 DP Color Cursor Mapping Async 0 Flow Register (IPU_DP_CUR_MAP_ASYNC0)

The LCD Color Cursor Mapping Register defines the color of the cursor in passive or TFT color modes.

Address: 260_0000h base + 1_8070h offset = 261_8070h



IPU_DP_CUR_MAP_ASYNC0 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value 0.
23–16 DP_CUR_COL_ B_ASYNC0	CUR_COL_B - Cursor Blue Field Defines the Blue component of the cursor color in color mode 00000000 No Blue. 11111111 Full Blue.
15–8 DP_CUR_COL_ G_ASYNC0	CUR_COL_G - Cursor Green Field Defines the Green component of the cursor color in color mode 00000000 No Green. 11111111 Full Green.
DP_CUR_COL_ R_ASYNC0	CUR_COL_R - Cursor Red Field Defines the Red component of the cursor color in color mode

Table continues on the next page...

IPU_DP_CUR_MAP_ASYNC0 field descriptions (continued)

Field	Description
	00000000 No Red. 11111111 Full Red.

38.5.124 DP Gamma Constant Async 0 Flow Register i (IPU_DP_GAMMA_C_ASYNC0_i)

This registers contains CONSTANTi parameters used for gamma correction inside the display processor (DP).

Address: 260_0000h base + 1_8074h offset = 261_8074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DP_GAMMA_C_ASYNC0_2i_1												0							DP_GAMMA_C_ASYNC0_2i									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_GAMMA_C_ASYNC0_i field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–16 DP_GAMMA_C_ASYNC0_2i_1	CONSTANTi+1 parameter of Gamma Correction.
15–9 Reserved	This read-only field is reserved and always has the value 0.
DP_GAMMA_C_ASYNC0_2i	CONSTANTi parameter of Gamma Correction.

38.5.125 DP Gamma Correction Slope Async 0 Flow Register i (IPU_DP_GAMMA_S_ASYNC0_i)

This registers contains SLOPEi parameters used for Gamma Correction inside the display processor (DP).

Address: 260_0000h base + 1_8094h offset = 261_8094h

Bit	31–24	23–16	15–8	7–0
R W	DP_GAMMA_S_ASYNC0_4i_3	DP_GAMMA_S_ASYNC0_4i_2	DP_GAMMA_S_ASYNC0_4i_1	DP_GAMMA_S_ASYNC0_4i
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

IPU_DP_GAMMA_S_ASYNC0_i field descriptions

Field	Description
31–24 DP_GAMMA_S_ASYNC0_4i_3	SLOPE<4*i+3> parameter of Gamma Correction.
23–16 DP_GAMMA_S_ASYNC0_4i_2	SLOPE<4*i+2> parameter of Gamma Correction.
15–8 DP_GAMMA_S_ASYNC0_4i_1	SLOPE<4*i+1> parameter of Gamma Correction.
DP_GAMMA_S_ASYNC0_4i	SLOPE<4*i> parameter of Gamma Correction.

38.5.126 DP Color Space Conversion Control Async 0 Flow Register i (IPU_DP_CSCA_ASYNC0_i)

Address: 260_0000h base + 1_80A4h offset = 261_80A4h

Bit	31–26	25–16	15–10	9–0
R W	0	DP_CSC_A_ASYNC0_2i_1	0	DP_CSC_A_ASYNC0_2i
Reset	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0

IPU_DP_CSCA_ASYNC0_i field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DP_CSCA_ASYNC0_i field descriptions (continued)

Field	Description
25–16 DP_CSC_A_ ASYNC0_2i_1	A<2*i+1> parameter of color conversion
15–10 Reserved	This read-only field is reserved and always has the value 0.
DP_CSC_A_ ASYNC0_2i	A<2*i> parameter of color conversion.

38.5.127 DP Color Conversion Control Async 0 Flow Register 0 (IPU_DP_CSC_ASYNC0_0)

Address: 260_0000h base + 1_80B4h offset = 261_80B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DP_CSC_ S0_ASYNC0		DP_CSC_B0_ASYNC0													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						DP_CSC_A8_ASYNC0									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CSC_ASYNC0_0 field descriptions

Field	Description
31–30 DP_CSC_S0_ ASYNC0	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
29–16 DP_CSC_B0_ ASYNC0	B0 parameter of color conversion.
15–10 Reserved	This read-only field is reserved and always has the value 0.
DP_CSC_A8_ ASYNC0	A9 parameter of color conversion.

38.5.128 DP Color Conversion Control Async 1 Flow Register (IPU_DP_CSC_ASYNC_1)

Address: 260_0000h base + 1_80B8h offset = 261_80B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DP_CSC_S2_ASYNC0		DP_CSC_B2_ASYNC0													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DP_CSC_S1_ASYNC0		DP_CSC_B1_ASYNC0													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

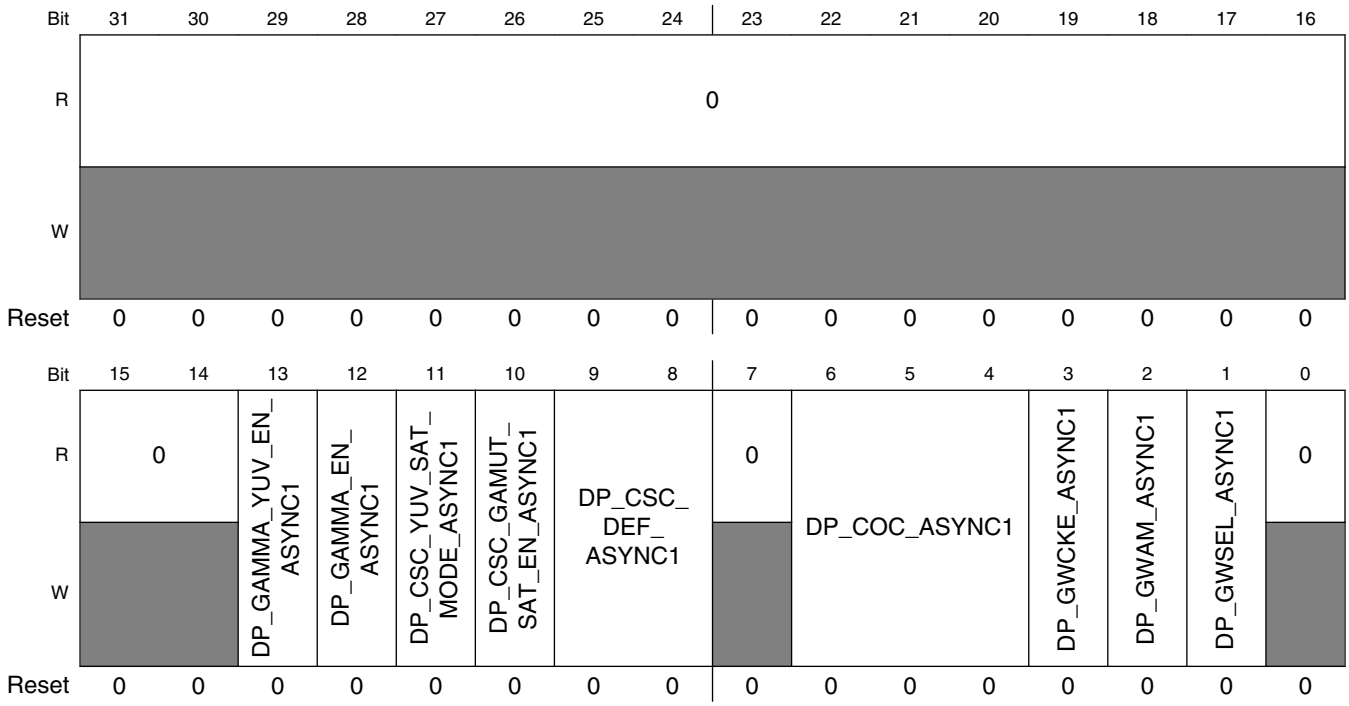
IPU_DP_CSC_ASYNC_1 field descriptions

Field	Description
31–30 DP_CSC_S2_ASYNC0	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
29–16 DP_CSC_B2_ASYNC0	B0 parameter of color conversion.
15–14 DP_CSC_S1_ASYNC0	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
DP_CSC_B1_ASYNC0	B0 parameter of color conversion.

38.5.129 DP Common Configuration Async 1 Flow Register (IPU_DP_COM_CONF_ASYNC1)

This register contains common configuration parameters for the DP.

Address: 260_0000h base + 1_80BCh offset = 261_80BCh



IPU_DP_COM_CONF_ASYNC1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
13 DP_GAMMA_YUV_EN_ASYNC1	GAMMA's YUV mode enable for async flow 1 0 YUV mode is OFF. 1 YUV mode is ON.
12 DP_GAMMA_EN_ASYNC1	GAMMA_EN - Gamma correction block enable bit 0 disable 1 enable
11 DP_CSC_YUV_SAT_MODE_ASYNC1	CSC_YUV_SAT_MODE YUV saturation mode for color space conversion 0 Y/U/V range 0 -255 1 Y range 16-235, U/V range 16-240

Table continues on the next page...

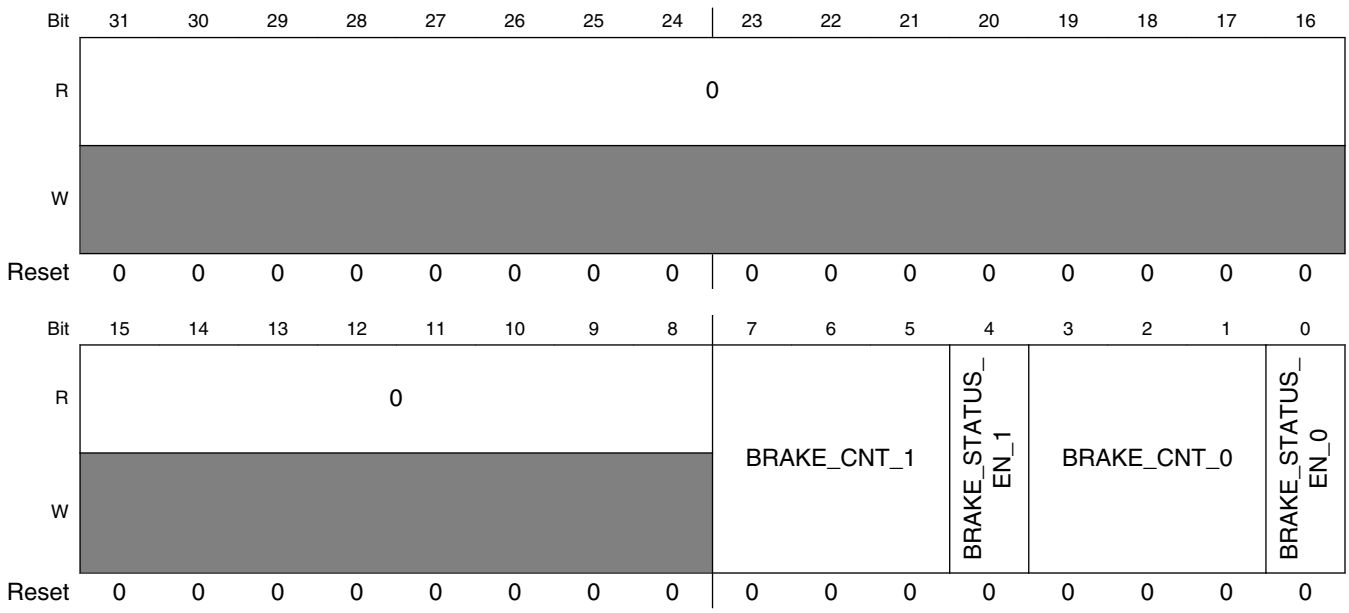
IPU_DP_COM_CONF_ASYNC1 field descriptions (continued)

Field	Description
10 DP_CSC_ GAMUT_SAT_ EN_ASYNC1	CSC_GAMUT_SAT_EN Indicate if GAMUT saturation is enabled 0 disable GAMUT mapping 1 enable GAMUT mapping
9–8 DP_CSC_DEF_ ASYNC1	CSC_DEF Enable or disable Color Space Conversion. 00 CSC disable 01 CSC enable after combining 10 CSC enable before combining on BG channel 11 CSC enable before combining on FG channel
7 Reserved	This read-only field is reserved and always has the value 0.
6–4 DP_COC_ ASYNC1	COC - Cursor Operation Control Controls the format of the cursor and the type of arithmetic operations 000 Transparent, cursor is disabled. 001 Full cursor. 010 Reversed cursor. 011 AND between full plane and cursor. 100 Reserved 101 OR between full plane and cursor. 110 XOR between full plane and cursor. 111 Reserved
3 DP_GWCKE_ ASYNC1	GWCKE - Graphic Window Color Keying Enable Enable or disable graphic window color keying. 1 Enable color keying of graphic window 0 Disable color keying of graphic window
2 DP_GWAM_ ASYNC1	GWAM - Graphic Window Alpha Mode Select the use of Alpha to be global or local. 1 Global Alpha. 0 Local Alpha.
1 DP_GWSEL_ ASYNC1	GWSEL - Graphic Window Select Select graphic window to be on partial plane or full plane. 1 Graphic window is partial plane. 0 Graphic window is full plane.
0 Reserved	This read-only field is reserved and always has the value 0.

38.5.130 DP Debug Control Register (IPU_DP_DEBUG_CNT)

This is the debug unit control register. This register is not stored in the SRM.

Address: 260_0000h base + 1_80BCh offset = 261_80BCh



IPU_DP_DEBUG_CNT field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.
7–5 BRAKE_CNT_1	The async flow can be broken multiple times. It possible to control which breaking event will cause the interrupt. This field counts the breaking events for unit #1
4 BRAKE_ STATUS_EN_1	This bit enables the break/status unit #1
3–1 BRAKE_CNT_0	The async flow can be broken multiple times. It possible to control which breaking event will cause the interrupt. This field counts the breaking events for unit #0
0 BRAKE_ STATUS_EN_0	This bit enables the break/status unit #0

38.5.131 DP Graphic Window Control Async 1 Flow Register (IPU_DP_GRAPH_WIND_CTRL_ASYNC1)

This register contains common configuration parameters for the DP.

Address: 260_0000h base + 1_80C0h offset = 261_80C0h

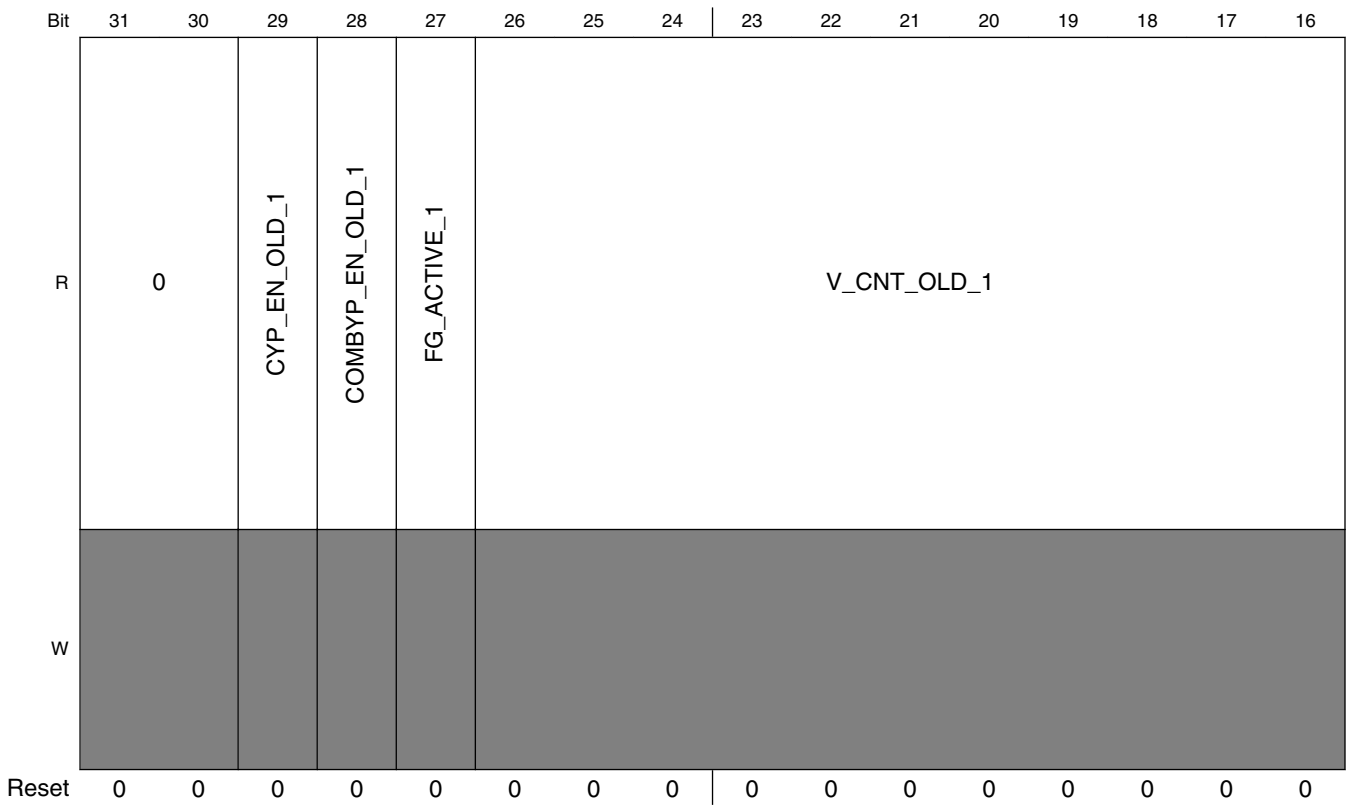
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DP_GWAV_ASYNC1								DP_GWCKR_ASYNC1								DP_GWCKG_ASYNC1								DP_GWCKB_ASYNC1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

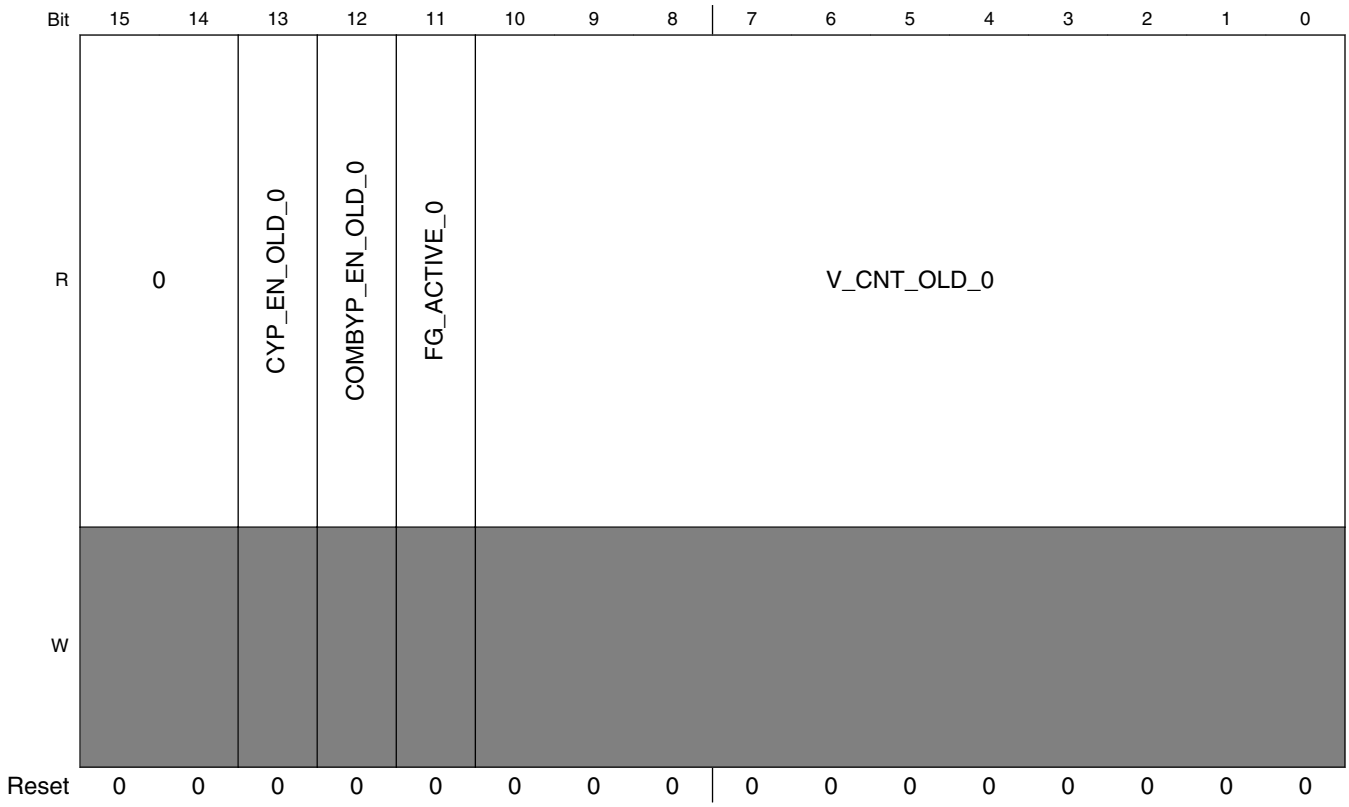
IPU_DP_GRAPH_WIND_CTRL_ASYNC1 field descriptions

Field	Description
31–24 DP_GWAV_ASYNC1	GWAV - Graphic Window Alpha Value Defines the alpha value of graphic window used for alpha blending between graphic window and full plane. The Value the number that writing to GWAV register. The Actual Value the number, that insert to calculation in Combining Module. If Value = < 0.5- 1/256 (01111111) then Actual Value = Value. If Value >= 0.5 (10000000), then Actual Value = Value + 1/256. 00000000 Actual value is 00000000; Graphic window totally opaque i.e. overlay on LCD screen 01111111 Actual value is 01111111; 10000000 Actual value is 10000001 10000001 Actual value is 10000010 11111110 Actual value is 11111111 11111111 Actual value is 100000000;Graphic window totally transparent i.e. not displayed on LCD screen
23–16 DP_GWCKR_ASYNC1	GWCKR - Graphic Window Color Keying Red Component Defines the red component of graphic window color keying. 00000000 No red 11111111 Full red
15–8 DP_GWCKG_ASYNC1	GWCKG - Graphic Window Color Keying Green Component Defines the green component of graphic window color keying. 00000000 No Green 11111111 Full Green
DP_GWCKB_ASYNC1	GWCKB - Graphic Window Color Keying Blue Component Defines the blue component of graphic window color keying. 00000000 No blue 11111111 Full blue

38.5.132 DP Debug Status Register (IPU_DP_DEBUG_STAT)

Address: 260_0000h base + 1_80C0h offset = 261_80C0h





IPU_DP_DEBUG_STAT field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value 0.
29 CYP_EN_OLD_1	The async flow has been broken in the middle of a cursor (This filed is relevant for debug unit #1)
28 COMBYP_EN_OLD_1	the async1 flow has been broken in the middle of combining (This filed is relevant for debug unit #1)
27 FG_ACTIVE_1	Displaying the partial frame has been started (This filed is relevant for debug unit #1)
26–16 V_CNT_OLD_1	The exact row where the async flow has been broken (This filed is relevant for debug unit #0)
15–14 Reserved	This read-only field is reserved and always has the value 0.
13 CYP_EN_OLD_0	The async flow has been broken in the middle of a cursor (This filed is relevant for debug unit #0)
12 COMBYP_EN_OLD_0	the async flow has been broken in the middle of combining (This filed is relevant for debug unit #0)
11 FG_ACTIVE_0	Displaying the partial frame has been started for async flow (This filed is relevant for debug unit #0)
V_CNT_OLD_0	The exact row where the async flow has been broken (This filed is relevant for debug unit #0)

38.5.133 DP Partial Plane Window Position Async 1 Flow Register (IPU_DP_FG_POS_ASYNC1)

The DP partial plane Window Position Register is used to determine the starting position of the partial plane window relative to BG window position.

This Register is used for the synchronous flow only.

Address: 260_0000h base + 1_80C4h offset = 261_80C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					DP_FGXP_ASYNC1											0					DP_FGYP_ASYNC1										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_FG_POS_ASYNC1 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26–16 DP_FGXP_ASYNC1	FGXP partial plane Window X Position. partial plane Window X Position - Specifies the number of pixels between the start of full plane window X position and the beginning of the first data of new line.
15–11 Reserved	This read-only field is reserved and always has the value 0.
DP_FGYP_ASYNC1	FGYP partial plane window Y position partial plane Window Y Position - Specifies the number of lines between the start of full plane windows Y position and the beginning of the first data.

38.5.134 DP Cursor Postion and Size Async 1 Flow Register (IPU_DP_CUR_POS_ASYNC1)

The LCD Cursor Position Register is used to determine the starting position of the cursor relative to BG windows position.

Address: 260_0000h base + 1_80C8h offset = 261_80C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DP_CYP_ASYNC1					DP_CYH_ASYNC1											DP_CXP_ASYNC1					DP_CXW_ASYNC1										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

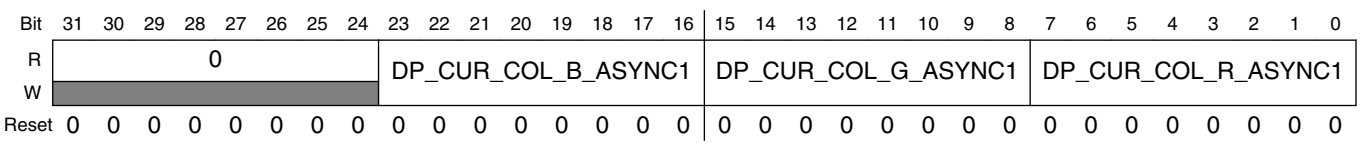
IPU_DP_CUR_POS_ASYNC1 field descriptions

Field	Description
31–27 DP_CYP_ ASYNC1	CYP - Cursor Y Position Represents the cursors vertical starting position Y in pixel count (from 0 to CYH).Live View Resolution Mode.
26–16 DP_CYH_ ASYNC1	CYH - Cursor Height Specifies the height of the hardware cursor in pixels.
15–11 DP_CXP_ ASYNC1	CXP - Cursor X Position Represents the cursors horizontal starting position X in pixel count (from 0 to CXW).
DP_CXW_ ASYNC1	CXW - Cursor Width. Specifies the width of the hardware cursor in pixels.

38.5.135 DP Color Cursor Mapping Async 1 Flow Register (IPU_DP_CUR_MAP_ASYNC1)

The LCD Color Cursor Mapping Register defines the color of the cursor in passive or TFT color modes.

Address: 260_0000h base + 1_80CCh offset = 261_80CCh



IPU_DP_CUR_MAP_ASYNC1 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value 0.
23–16 DP_CUR_COL_ B_ASYNC1	CUR_COL_B - Cursor Blue Field Defines the Blue component of the cursor color in color mode 00000000 No Blue. 11111111 Full Blue.
15–8 DP_CUR_COL_ G_ASYNC1	CUR_COL_G - Cursor Green Field Defines the Green component of the cursor color in color mode 00000000 No Green. 11111111 Full Green.
DP_CUR_COL_ R_ASYNC1	CUR_COL_R - Cursor Red Field Defines the Red component of the cursor color in color mode

Table continues on the next page...

IPU_DP_CUR_MAP_ASYNC1 field descriptions (continued)

Field	Description
	00000000 No Red. 11111111 Full Red.

38.5.136 DP Gamma Constants Async 1 Flow Register i (IPU_DP_GAMMA_C_ASYNC1_i)

This registers contains CONSTANTi parameters used for gamma correction inside the display processor (DP).

Address: 260_0000h base + 1_80D0h offset = 261_80D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							DP_GAMMA_C_ASYNC1_2i_1									0							DP_GAMMA_C_ASYNC1_2i								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_GAMMA_C_ASYNC1_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 DP_GAMMA_C_ASYNC1_2i_1	CONSTANTi+1 parameter of Gamma Correction.
15–9 Reserved	This read-only field is reserved and always has the value 0.
DP_GAMMA_C_ASYNC1_2i	CONSTANTi parameter of Gamma Correction.

38.5.137 DP Gamma Correction Slope Async 1 Flow Register i (IPU_DP_GAMMA_S_ASYN1_i)

This registers contains SLOPEi parameters used for Gamma Correction inside the display processor (DP).

Address: 260_0000h base + 1_80F0h offset = 261_80F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DP_GAMMA_S_ASYNC1_4i_3								DP_GAMMA_S_ASYNC1_4i_2								DP_GAMMA_S_ASYNC1_4i_1								DP_GAMMA_S_ASYNC1_4i							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_GAMMA_S_ASYN1_i field descriptions

Field	Description
31–24 DP_GAMMA_S_ASYNC1_4i_3	SLOPE<4*i+3> parameter of Gamma Correction.
23–16 DP_GAMMA_S_ASYNC1_4i_2	SLOPE<4*i+2> parameter of Gamma Correction.
15–8 DP_GAMMA_S_ASYNC1_4i_1	SLOPE<4*i+1> parameter of Gamma Correction.
DP_GAMMA_S_ASYNC1_4i	SLOPE<4*i> parameter of Gamma Correction.

38.5.138 DP Color Space Converstion Control Async 1 Flow Register i (IPU_DP_CSCA_ASYNC1_i)

Address: 260_0000h base + 1_8100h offset = 261_8100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	0						DP_CSC_A_ASYNC1_2i_1										0						DP_CSC_A_ASYNC1_2i									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CSCA_ASYNC1_i field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DP_CSCA_ASYNC1_i field descriptions (continued)

Field	Description
25–16 DP_CSC_A_ ASYNC1_2i_1	A<2*i+1> parameter of color conversion.
15–10 Reserved	This read-only field is reserved and always has the value 0.
DP_CSC_A_ ASYNC1_2i	A<2*i> parameter of color conversion.

38.5.139 DP Color Conversion Control Async 1 Flow Register 0 (IPU_DP_CSC_ASYNC1_0)

Address: 260_0000h base + 1_8110h offset = 261_8110h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DP_CSC_ S0_ASYNC1		DP_CSC_B0_ASYNC1													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						DP_CSC_A8_ASYNC1									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CSC_ASYNC1_0 field descriptions

Field	Description
31–30 DP_CSC_S0_ ASYNC1	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
29–16 DP_CSC_B0_ ASYNC1	B0 parameter of color conversion.
15–10 Reserved	This read-only field is reserved and always has the value 0.
DP_CSC_A8_ ASYNC1	A9 parameter of color conversion.

38.5.140 DP Color Conversion Control Async 1 Flow Register 1 (IPU_DP_CSC_ASYNC1_1)

Address: 260_0000h base + 1_8114h offset = 261_8114h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DP_CSC_S2_ASYNC1		DP_CSC_B2_ASYNC1													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DP_CSC_S1_ASYNC1		DP_CSC_B1_ASYNC1													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

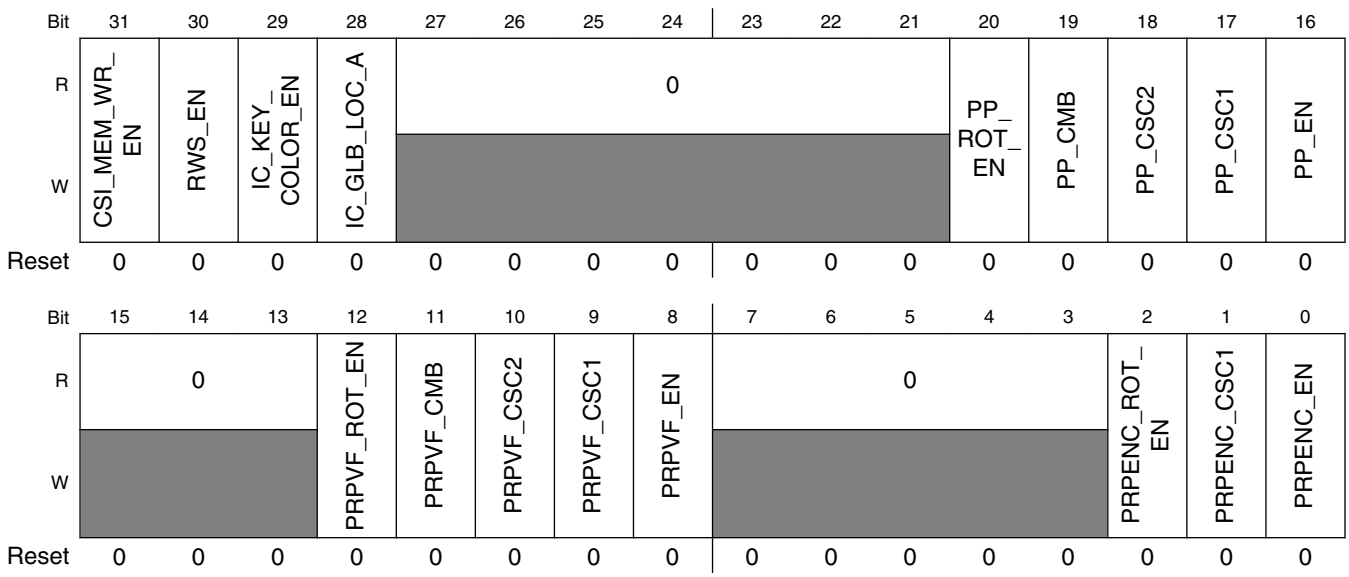
IPU_DP_CSC_ASYNC1_1 field descriptions

Field	Description
31–30 DP_CSC_S2_ASYNC1	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
29–16 DP_CSC_B2_ASYNC1	B0 parameter of color conversion.
15–14 DP_CSC_S1_ASYNC1	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
DP_CSC_B1_ASYNC1	B0 parameter of color conversion.

38.5.141 IC Configuration Register (IPU_IC_CONF)

This register contains control parameter for IC 3 tasks (pre-processing for encoding, pre-processing for view-finder and post processing).

Address: 260_0000h base + 2_0000h offset = 262_0000h



IPU_IC_CONF field descriptions

Field	Description
31 CSI_MEM_WR_EN	CSI direct memory write enable. This bit enables writing data from sensor directly to memory even when a raw sensor is not attached. 0 CSI direct writing to memory is disabled. 1 CSI direct writing to memory is enabled.
30 RWS_EN	Raw sensor enable. This bit indicate if a Raw sensor is attached (Bayer format). This bit is used together with the CSI_MEM_WR_EN bit as follows: CSI_MEM_WR_EN=0, RWS_EN=0 - data is fed from the CSI to the IC for processing; CSI_MEM_WR_EN=1, RWS_EN=0 - data is fed from the CSI to the IC for processing and also for writing to the system memory; CSI_MEM_WR_EN=0, RWS_EN=1 - data is fed from the CSI to the system memory (via the IC) and from the system memory to the IC for processing; CSI_MEM_WR_EN=1, RWS_EN=1 - non-valid configuration. 0 Raw sensor is not attached. 1 Raw sensor is attached.
29 IC_KEY_COLOR_EN	Key Color enable. This bit enables the key color feature.

Table continues on the next page...

IPU_IC_CONF field descriptions (continued)

Field	Description
	0 Key color is disabled. 1 Key color is enabled.
28 IC_GLB_LOC_A	Global Alpha. This bit select the source of Alpha parameter. 0 Alpha parameter is local. 1 Alpha parameter is global.
27–21 Reserved	This read-only field is reserved and always has the value 0.
20 PP_ROT_EN	Post-Processing Rotation Task enable. This bit enable Post-Processing Rotation Task. 0 Rotation is disabled. 1 Rotation is enabled.
19 PP_CMB	Post-Processing Task combining enable. This bit enables combining. 0 Combining is disabled. 1 Combining is enabled.
18 PP_CSC2	Post-Processing Task color conversion RGB-->YUV enable. This bit enables YUV-->RGB. Reserved 0 RGB-->YUV is disabled. 1 RGB-->YUV is enabled.
17 PP_CSC1	Post-Processing Task color conversion YUV-->RGB enable. This bit enables YUV-->RGB. 0 YUV-->RGB is disabled. 1 YUV-->RGB is enabled.
16 PP_EN	Post-Processing Task enable. This bit enables the Post-Processing task. 0 Task is disabled. 1 Task is enabled.
15–13 Reserved	This read-only field is reserved and always has the value 0.
12 PRPVF_ROT_EN	Preprocessing Rotation Task for viewfinder enable. This bit enable Preprocessing Rotation Task for viewfinder. 0 Rotation is disabled. 1 Rotation is enabled.
11 PRPVF_CMB	Preprocessing Task for View-Finder combining enable. This bit enables combining. 0 Combining is disabled. 1 Combining is enabled.
10 PRPVF_CSC2	Reserved
9 PRPVF_CSC1	Pre-processing task for view-finder first color conversion enable. This bit enables first color conversion. 0 First color conversion is disabled. 1 First color conversion is enabled.
8 PRPVF_EN	Preprocessing Task for View-Finder enable. This bit enables the View-Finder task.

Table continues on the next page...

IPU_IC_CONF field descriptions (continued)

Field	Description
	0 Task is disabled. 1 Task is enabled.
7–3 Reserved	This read-only field is reserved and always has the value 0.
2 PRPENC_ROT_EN	Preprocessing Rotation Task for encoding enable. This bit enable Preprocessing Rotation Task for encoding. 0 Rotation is disabled. 1 Rotation is enabled.
1 PRPENC_CSC1	Preprocessing Task for encoding color conversion enable. This bit enables color conversion. 0 Color conversion is disabled. 1 Color conversion is enabled.
0 PRPENC_EN	Preprocessing Task for encoding enable. This bit enables the encoding task. 0 Task is disabled. 1 Task is enabled.

38.5.142 IC Preprocessing Encoder Resizing Coefficients Register (IPU_IC_PRP_ENC_RSC)

This register contains the resizing and downsizing parameters for Preprocessing task for encoding.

Address: 260_0000h base + 2_0004h offset = 262_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PRPENC_DS_R_V		PRPENC_RS_R_V														PRPENC_DS_R_H		PRPENC_RS_R_H													
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IC_PRP_ENC_RSC field descriptions

Field	Description
31–30 PRPENC_DS_R_V	Preprocessing task for encoding Downsizing vertical Ratio. This field contains the downsizing vertical coefficient of Preprocessing for Encoding.
29–16 PRPENC_RS_R_V	Preprocessing task for encoding Resizing vertical Ratio. This field contains the resizing vertical coefficient of Preprocessing for Encoding.

Table continues on the next page...

IPU_IC_PRP_ENC_RSC field descriptions (continued)

Field	Description
	Resizing Ratio is equal to PRPENC_RS_R_V: M Where M = 2^13; SI - input size; SO - output size PRPENC_RS_R_V = floor(M*(SI-1)/(SO-1));
15–14 PRPENC_DS_R_H	Preprocessing task for encoding Downsizing horizontal Ratio. This field contains the downsizing horizontal coefficient of Preprocessing for Encoding. Values: 00 1 01 2 10 4 11 RSV
PRPENC_RS_R_H	Preprocessing task for encoding Resizing horizontal Ratio. This field contains the resizing horizontal coefficient of Preprocessing for Encoding. Resizing Ratio is equal to PRPENC_RS_R_H: M Where M = 2^13; SI - input size; SO - output size PRPENC_RS_R_H = floor(M*(SI-1)/(SO-1));

38.5.143 IC Preprocessing View-Finder Resizing Coefficients Register (IPU_IC_PRP_VF_RSC)

This register contains the resizing and downsizing parameters for preprocessing task for display.

Address: 260_0000h base + 2_0008h offset = 262_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PRPVF_DS_R_V		PRPVF_RS_R_V													
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PRPVF_DS_R_H		PRPVF_RS_R_H													
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IC_PRP_VF_RSC field descriptions

Field	Description
31–30 PRPVF_DS_R_V	Preprocessing task for encoding Downsizing vertical Ratio. This field contains the downsizing vertical coefficient of Preprocessing for View-Finder.
29–16 PRPVF_RS_R_V	Preprocessing task for encoding Resizing vertical Ratio. This field contains the resizing vertical coefficient of Preprocessing for View-Finder.

Table continues on the next page...

IPU_IC_PRP_VF_RSC field descriptions (continued)

Field	Description
	Resizing Ratio is equal to PRPVF_RS_R_V: M Where M = 2^13; SI - input size; SO - output size PRPVF_RS_R_V = floor(M*(SI-1)/(SO-1));
15–14 PRPVF_DS_R_H	Preprocessing task for encoding Downsizing horizontal Ratio. This field contains the downsizing horizontal coefficient of Preprocessing for View-Finder. Values: 00 1 01 2 10 4 11 RSV
PRPVF_RS_R_H	Preprocessing task for view-finding resizing horizontal ratio. This field contains the resizing horizontal coefficient of preprocessing Task For View-finder. Resizing Ratio is equal to PRPVF_RS_R_H: M Where M = 2^13; SI - input size; SO - output size PRPVF_RS_R_H = floor(M*(SI-1)/(SO-1));

38.5.144 IC Postprocessing Encoder Resizing Coefficients Register (IPU_IC_PP_RSC)

This register contains the resizing and downsizing parameters for Post-Processing task for display.

Address: 260_0000h base + 2_000Ch offset = 262_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PP_DS_R_V		PP_RS_R_V														PP_DS_R_H		PP_RS_R_H													
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IC_PP_RSC field descriptions

Field	Description
31–30 PP_DS_R_V	Post-Processing task Downsizing vertical Ratio. This field contains the downsizing vertical coefficient of Post-Processing.
29–16 PP_RS_R_V	Post-Processing task Resizing vertical Ratio. This field contains the resizing vertical coefficient of Post-Processing. Resizing Ratio is equal to PP_RS_R_V: M Where M = 2^13; SI - input size; SO - output size PP_RS_R_V = floor(M*(SI-1)/(SO-1));

Table continues on the next page...

IPU_IC_PP_RSC field descriptions (continued)

Field	Description
15–14 PP_DS_R_H	Post-Processing task Downsizing horizontal Ratio. This field contains the downsizing horizontal coefficient of Post-Processing. 00 1 01 2 10 4 11 RSV
PP_RS_R_H	Post-Processing task Resizing horizontal Ratio. This field contains the resizing horizontal coefficient of Post-Processing. Resizing Ratio is equal to PP_RS_R_H: M Where M = 2^13; SI - input size; SO - output size PP_RS_R_H = floor(M*(SI-1)/(SO-1));

38.5.145 IC Combining Parameters Register 1 (IPU_IC_CMBP_1)

Address: 260_0000h base + 2_0010h offset = 262_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																IC_PP_ALPHA_V								IC_PRPVF_ALPHA_V							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IC_CMBP_1 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 IC_PP_ALPHA_V	Post-Processing task Global Alpha. This field contains the Global Alpha value of Post-Processing.
IC_PRPVF_ALPHA_V	Preprocessing task for viewfinder Global Alpha. This field contains the Global Alpha value of Preprocessing for viewfinder.

38.5.146 IC Combining Parameters Register 2 (IPU_IC_CMBP_2)

Address: 260_0000h base + 2_0014h offset = 262_0014h

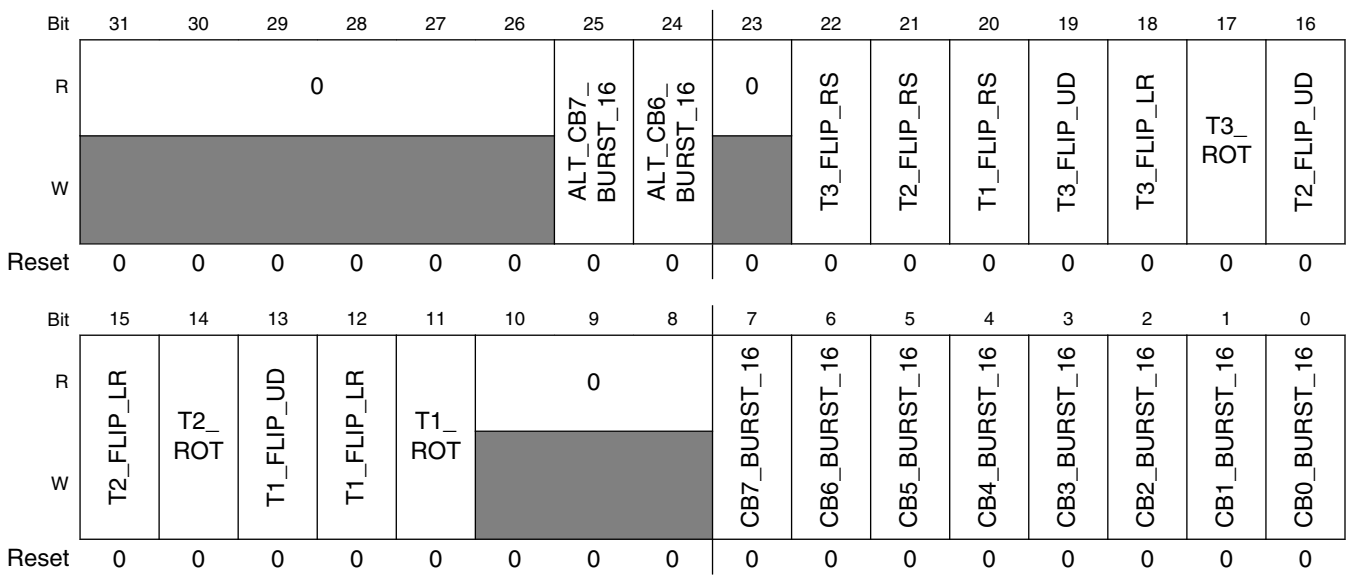
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								IC_KEY_COLOR_R								IC_KEY_COLOR_G								IC_KEY_COLOR_B							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IC_CMBP_2 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value 0.
23–16 IC_KEY_ COLOR_R	Key Color Red.
15–8 IC_KEY_ COLOR_G	Key Color Green.
IC_KEY_ COLOR_B	Key Color Blue.

38.5.147 IC IDMAC Parameters 1 Register (IPU_IC_IDMAC_1)

Address: 260_0000h base + 2_0018h offset = 262_0018h



IPU_IC_IDMAC_1 field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.
25 ALT_CB7_ BURST_16	Reserved
24 ALT_CB6_ BURST_16	Reserved
23 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_IC_IDMAC_1 field descriptions (continued)

Field	Description
22 T3_FLIP_RS	LEFT/RIGHT flip for Post Processing (PP) task; his bit affect the flipping done on the resizing unit. The value of this field must be identical to the corresponding channel's HF parameter in the IDMAC's CPMEM 1 horizontal flip enabled 0 no flip
21 T2_FLIP_RS	LEFT/RIGHT flip for View Finder (VF) task; this bit affect the flipping done on the resizing unit. The value of this field must be identical to the corresponding channel's HF parameter in the IDMAC's CPMEM 1 horizontal flip enabled 0 no flip
20 T1_FLIP_RS	LEFT/RIGHT flip for Encoding (ENC) task; this bit affect the flipping done on the resizing unit. The value of this field must be identical to the corresponding channel's HF parameter in the IDMAC's CPMEM 1 horizontal flip enabled 0 no flip
19 T3_FLIP_UD	UP/DOWN flip for Post Processing (PP) task The value of this field must be identical to the corresponding channel's VF parameters in the IDMAC's CPMEM 1 Vertical flip enable 0 no flip
18 T3_FLIP_LR	LEFT/RIGHT flip for Post Processing (PP) task; this bit affect the flipping done on the rotation unit The value of this field must be identical to the corresponding channel's HF parameter in the IDMAC's CPMEM 1 horizontal flip enabled 0 no flip
17 T3_ROT	Rotation for Post Processing (PP) task The value of this field must be identical to the corresponding channel's ROT parameters in the IDMAC's CPMEM 1 90 degree rotation clockwise 0 no rotation
16 T2_FLIP_UD	UP/DOWN flip for View Finder (VF) task The value of this field must be identical to the corresponding channel's VF parameters in the IDMAC's CPMEM 1 Vertical flip enable 0 no flip
15 T2_FLIP_LR	LEFT/RIGHT flip for View Finder (VF) task; this bit affect the flipping done on the rotation unit The value of this field must be identical to the corresponding channel's HF parameter in the IDMAC's CPMEM 1 horizontal flip enabled 0 no flip

Table continues on the next page...

IPU_IC_IDMAC_1 field descriptions (continued)

Field	Description
14 T2_ROT	Rotation for View Finder (VF) task The value of this field must be identical to the corresponding channel's ROT parameters in the IDMAC's CPMEM 1 90 degree rotation clockwise 0 no rotation
13 T1_FLIP_UD	UP/DOWN flip for Encoding (ENC) task The value of this field must be identical to the corresponding channel's VF parameters in the IDMAC's CPMEM 1 Vertical flip enable 0 no flip
12 T1_FLIP_LR	LEFT/RIGHT flip for Encoding (ENC) task; this bit affect the flipping done on the rotation unit The value of this field must be identical to the corresponding channel's HF parameter in the IDMAC's CPMEM 1 horizontal flip enabled 0 no flip
11 T1_ROT	Rotation for Encoding (ENC) task The value of this field must be identical to the corresponding channel's ROT parameters in the IDMAC's CPMEM 1 90 degree rotation clockwise 0 no rotation
10–8 Reserved	This read-only field is reserved and always has the value 0.
7 CB7_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB7 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM 0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111
6 CB6_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB6 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM 0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111
5 CB5_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB5 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM 0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111
4 CB4_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB4 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM

Table continues on the next page...

IPU_IC_IDMAC_1 field descriptions (continued)

Field	Description
	0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111
3 CB3_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB3 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM 0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111
2 CB2_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB2 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM 0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111
1 CB1_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB1 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM 0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111
0 CB0_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB0 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM 0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111

38.5.148 IC IDMAC Parameters 2 Register (IPU_IC_IDMAC_2)

Address: 260_0000h base + 2_001Ch offset = 262_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		T3_FR_HEIGHT										T2_FR_HEIGHT										T1_FR_HEIGHT									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IC_IDMAC_2 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value 0.
29–20 T3_FR_HEIGHT	Frame Height for Post Processing (PP) task The value of this field must be identical to the corresponding FH channel's parameters in the IDMAC's CPMEM. This parameter refers to the output's size -1

Table continues on the next page...

IPU_IC_IDMAC_2 field descriptions (continued)

Field	Description
19–10 T2_FR_HEIGHT	Frame Height for View Finder (VF) task The value of this field must be identical to the corresponding FH channel's parameters in the IDMAC's CPMEM. This parameter refers to the output's size -1
T1_FR_HEIGHT	Frame Height for Encoding (ENC) task The value of this field must be identical to corresponding FH channel's parameters in the IDMAC's CPMEM. This parameter refers to the output's size -1

38.5.149 IC IDMAC Parameters 3Register (IPU_IC_IDMAC_3)

Address: 260_0000h base + 2_0020h offset = 262_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		T3_FR_WIDTH														T2_FR_WIDTH														T1_FR_WIDTH			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IC_IDMAC_3 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value 0.
29–20 T3_FR_WIDTH	Frame Width for Post Processing (PP) task The value of this field must be identical to the corresponding FW channel's parameters in the IDMAC's CPMEM. This parameter refers to the output's size -1
19–10 T2_FR_WIDTH	Frame Width for View Finder (VF) task The value of this field must be identical to the corresponding FW channel's parameters in the IDMAC's CPMEM. This parameter refers to the output's size -1
T1_FR_WIDTH	Frame Width for Encoding (ENC) task The value of this field must be identical to corresponding FW channel's parameters in the IDMAC's CPMEM. This parameter refers to the output's size -1

38.5.150 IC IDMAC Parameters 4 Register (IPU_IC_IDMAC_4)

Address: 260_0000h base + 2_0024h offset = 262_0024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																rm_brdg_ max_rq				ibm_brdg_ max_rq				mpm_dmfc_ brdg_max_rq				mpm_rw_ brdg_max_rq			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IC_IDMAC_4 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–12 rm_brdg_max_rq	RM memory Bridge Max Requests 0000 Feature is disabled 0001 Max request is 1 1111 Max request are15
11–8 ibm_brdg_max_rq	IBM memory Bridge Max Requests 0000 Feature is disabled 0001 Max request is 1 1111 Max request are15
7–4 mpm_dmfc_brdg_max_rq	MPM memory Bridge Max Requests for the IC DMFC interface 0000 Feature is disabled 0001 Max request is 1 1111 Max request are15
mpm_rw_brdg_max_rq	MPM memory Bridge Max Requests between MPM's read and writes 0000 Feature is disabled 0001 Max request is 1 1111 Max request are15

38.5.151 CSI0 Sensor Configuration Register (IPU_CSI0_SENS_CONF)

Address: 260_0000h base + 3_0000h offset = 263_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CSI0_DATA_EN_POL	0	CSI0_FORCE_EOF	CSI0_JPEG_MODE	CSI0_JPEG8_EN	CSI0_DATA_DEST			CSI0_DIV_RATIO							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSI0_EXT_VSYNC	CSI0_DATA_WIDTH				CSI0_SENS_DATA_FORMAT			CSI0_PACK_TIGHT	CSI0_SENS_PRTCL			CSI0_SENS_PIX_CLK_POL	CSI0_DATA_POL	CSI0_HSYNC_POL	CSI0_VSYNC_POL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI0_SENS_CONF field descriptions

Field	Description
31 CSI0_DATA_EN_POL	Invert IPP_IND_SENSB_DATA_EN input. This bit selects the polarity of IPP_IND_SENSB_DATA_EN signal. 0 IPP_IND_SENSB_DATA_EN is directly applied to internal circuitry. 1 IPP_IND_SENSB_DATA_EN is inverted before applied to internal circuitry.
30 Reserved	This read-only field is reserved and always has the value 0.
29 CSI0_FORCE_EOF	Force End of frame This is a self clear bit allowing the user to force an End-of-frame event; This bit can be used in cases where the frame sent by the sensor was not completed. 1 force end of frame 0 no action
28 CSI0_JPEG_MODE	JPEG Mode - this bit defines the mode of the control signals when working in JPEG mode 1 The data is valid as long as HSYNC and VSYNC signals are active; HSYNC is valid for single frame 0 The frame starts withe the assertion of VSYNC. The frame ends on the next VSYNC or by setting the **CSI0_FORCE_EOF bit.**
27 CSI0_JPEG8_EN	JPEG8 enable bit 1 JPEG8 detection is enabled 0 JPEG8 is disabled
26–24 CSI0_DATA_DEST	These bits enable the destination of the data coming from the CSI. CSI0_DATA_DEST[0] - Reserved CSI0_DATA_DEST[1] - destination is IC CSI0_DATA_DEST[2] - destination is IDMAC via SMFC
23–16 CSI0_DIV_RATIO	DIV Ratio Clock division ratio minus 1. This field defines the division ratio of HSP_CLK into SENSB_MCLK: SENSB_MCLK rate = HSP_CLK rate /(DIV_RATIO+1)
15 CSI0_EXT_VSYNC	External VSYNC enable. This bits select between external and internal VSYNC. 0 Internal VSYNC mode. 1 External VSYNC mode.
14–11 CSI0_DATA_WIDTH	Data width. This field defines the number of bits per color. Values: 0000 4 bits per color 0001 8 bits per color 0010 9 bits per color 0011 10 bits per color 0100 11 bits per color 0101 12 bits per color 0110 13 bits per color 0111 14 bits per color 1000 15 bits per color 1001 16 bits per color

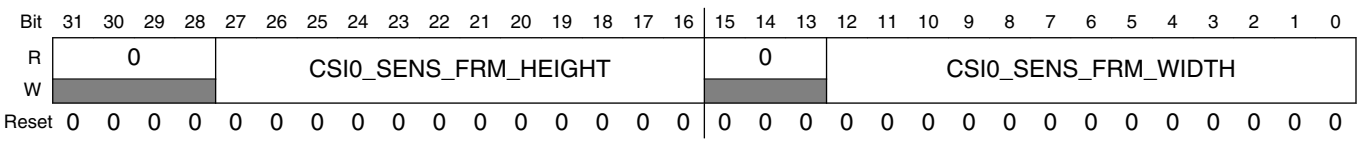
Table continues on the next page...

IPU_CSI0_SENS_CONF field descriptions (continued)

Field	Description
10–8 CSI0_SENS_ DATA_FORMAT	Data format from the sensor. This field defines the data format for the input of the CSI sensor. Values: 000 full RGB or YUV444 001 YUV422 (YUYV...) 010 YUV422 (UYVY...) 011 Bayer or Generic data 100 RGB565 101 RGB555 110 RGB444 111 JPEG
7 CSI0_PACK_ TIGHT	CSI0 Pack Tight When the data format is YUV or RGB and the component's width is 9-16 bits, it can be sent to the memory in 2 different ways. 1 Three 10 bits components are packed into a 32 bit word. Color extension/reduction is performed 0 Each component is written as a 16 bit word where the MSB is written to bit #15, color extension is done for the remaining least significant bits.
6–4 CSI0_SENS_ PRTCL	Sensor Protocol. This bit defines the Sensor timing/data mode protocol. Values: 000 Gated clock mode 001 Non-gated clock mode 010 CCIR progressive mode (BT.656) 011 CCIR interlaced mode (BT.656) 100 CCIR progressive (BT.1120 DDR mode: data arrives on every edge of the clock) 101 CCIR progressive (BT.1120 SDR mode: data arrives only on the positive edge of the clock) 110 CCIR interlaced mode (BT.1120 DDR mode: data arrives on every edge of the clock) 111 CCIR interlaced mode (BT.1120 SDR mode: data arrives only on the positive edge of the clock)
3 CSI0_SENS_ PIX_CLK_POL	Invert Pixel clock input. This bit selects the polarity of pixel clock. 0 pixel clock is directly applied to internal circuitry. 1 pixel clock is inverted before applied to internal circuitry.
2 CSI0_DATA_ POL	Invert data input. This bit selects the polarity of data input. 0 data lines are directly applied to internal circuitry. 1 data lines are inverted before applied to internal circuitry.
1 CSI0_HSYNC_ POL	Invert IPP_IND_SENSB_HSYNC input. This bit selects the polarity of IPP_IND_SENSB_HSYNC signal. 0 IPP_IND_SENSB_HSYNC is directly applied to internal circuitry. 1 IPP_IND_SENSB_HSYNC is inverted before applied to internal circuitry.
0 CSI0_VSYNC_ POL	Invert IPP_IND_SENSB_VSYNC input. This bit selects the polarity of IPP_IND_SENSB_VSYNC signal. 0 IPP_IND_SENSB_VSYNC is not inverted before applied to internal circuitry. 1 IPP_IND_SENSB_VSYNC is inverted before applied to internal circuitry.

38.5.152 CSI0 Sense Frame Size Register (IPU_CSI0_SENS_FRM_SIZE)

Address: 260_0000h base + 3_0004h offset = 263_0004h

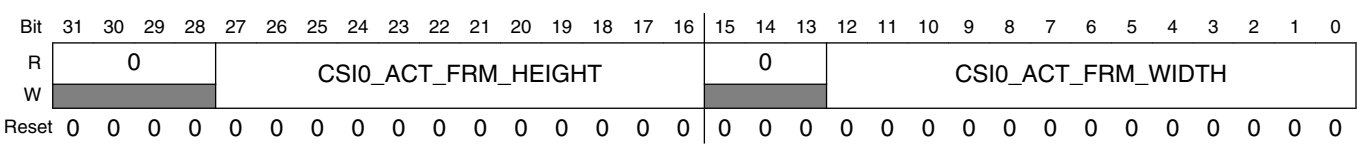


IPU_CSI0_SENS_FRM_SIZE field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–16 CSI0_SENS_ FRM_HEIGHT	Sensor frame height minus 1. This field defines the sensor frame rows number minus 1.
15–13 Reserved	This read-only field is reserved and always has the value 0.
CSI0_SENS_ FRM_WIDTH	Sensor frame width minus 1. This field defines the sensor frame column number minus 1.

38.5.153 CSI0 Actual Frame Size Register (IPU_CSI0_ACT_FRM_SIZE)

Address: 260_0000h base + 3_0008h offset = 263_0008h

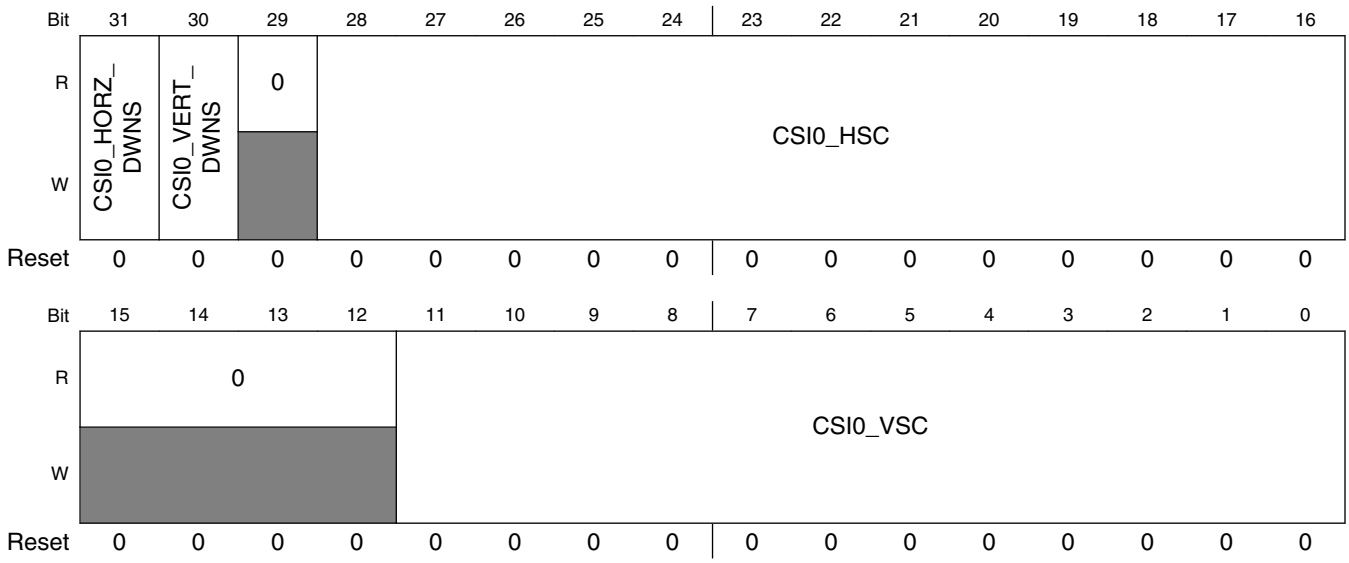


IPU_CSI0_ACT_FRM_SIZE field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–16 CSI0_ACT_ FRM_HEIGHT	Actual frame height minus 1. This field defines the CSI output frame rows number minus 1.
15–13 Reserved	This read-only field is reserved and always has the value 0.
CSI0_ACT_ FRM_WIDTH	Actual frame width minus 1. This field defines the CSI output frame columns number minus 1.

38.5.154 CSI0 Output Control Register (IPU_CSI0_OUT_FRM_CTRL)

Address: 260_0000h base + 3_000Ch offset = 263_000Ch

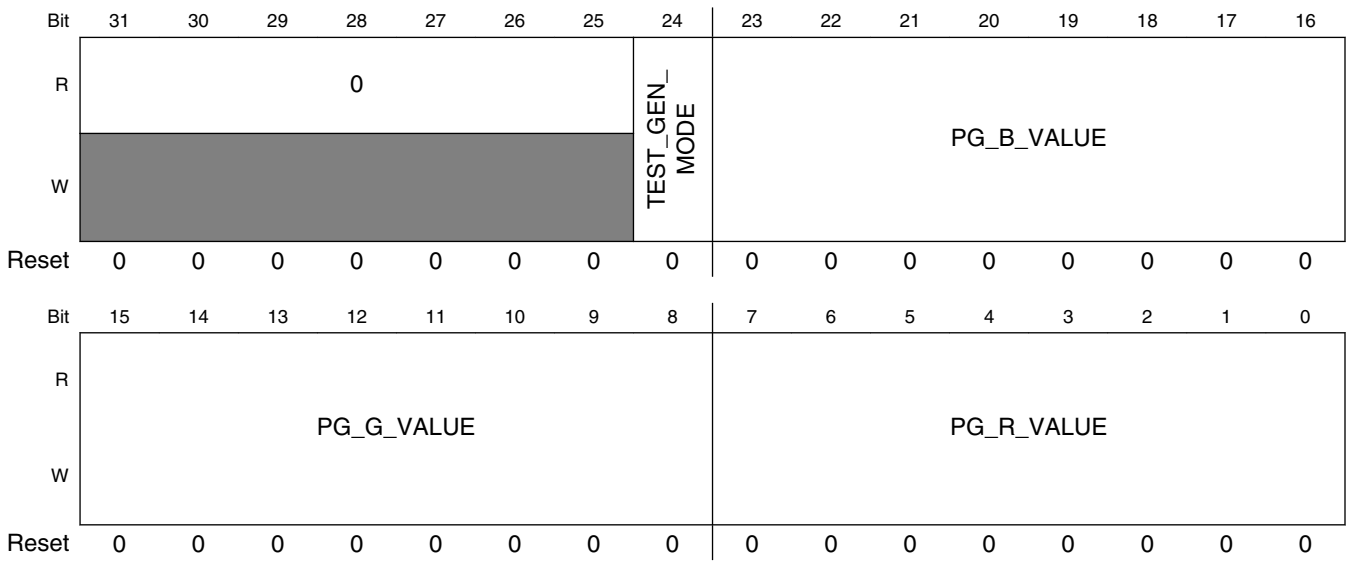


IPU_CSI0_OUT_FRM_CTRL field descriptions

Field	Description
31 CSI0_HORZ_DWNS	Enable horizontal downsizing (decimation) by 2. 0 Downsizing disabled 1 Downsizing enabled
30 CSI0_VERT_DWNS	Enable vertical downsizing (decimation) by 2. 0 Downsizing disabled 1 Downsizing enabled
29 Reserved	This read-only field is reserved and always has the value 0.
28–16 CSI0_HSC	Horizontal skip. This field defines the number of columns to skip. In Interlaced mode this number refers to the number of lines per field.
15–12 Reserved	This read-only field is reserved and always has the value 0.
CSI0_VSC	Vertical skip. This field defines the number of rows to skip.

38.5.155 CSI0 Test Control Register (IPU_CSI0_TST_CTRL)

Address: 260_0000h base + 3_0010h offset = 263_0010h

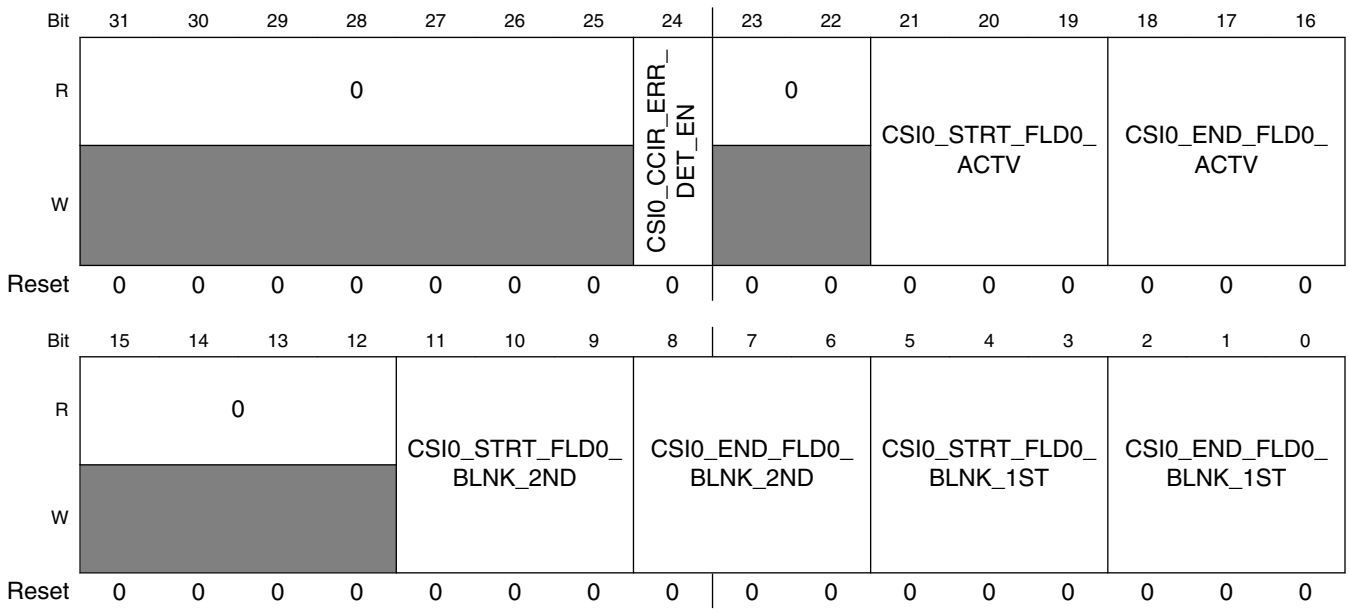


IPU_CSI0_TST_CTRL field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24 TEST_GEN_ MODE	Test generator mode. This bit activates the signal generation. 0 Test signal generator is inactive. 1 Test signal generator is active.
23–16 PG_B_VALUE	Pattern generator B value. This field selects the B value for the generated pattern of even pixel.
15–8 PG_G_VALUE	Pattern generator G value. This field selects the G value for the generated pattern of even pixel.
PG_R_VALUE	Pattern generator R value. This field selects the R value for the generated pattern of even pixel.

38.5.156 CSI0 CCIR Code Register 1 (IPU_CSI0_CCIR_CODE_1)

Address: 260_0000h base + 3_0014h offset = 263_0014h



IPU_CSI0_CCIR_CODE_1 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24 CSI0_CCIR_ERR_DET_EN	Enable error detection and correction for CCIR interlaced mode with protection bit. 0 Error detection and correction is disabled. 1 Error detection and correction is enabled.
23–22 Reserved	This read-only field is reserved and always has the value 0.
21–19 CSI0_STRT_FLD0_ACTV	Start of field 0 active line command (interlaces mode). (In progressive mode, start of active line command mode).
18–16 CSI0_END_FLD0_ACTV	End of field 0 active line command (interlaces mode). (In progressive mode, end of active line command mode).
15–12 Reserved	This read-only field is reserved and always has the value 0.
11–9 CSI0_STRT_FLD0_BLNK_2ND	Start of field 0 second blanking line command (interlaces mode). (In progressive mode this field is ignored).
8–6 CSI0_END_FLD0_BLNK_2ND	End of field 0 second blanking line command (interlaces mode). (In progressive mode this field is ignored).

Table continues on the next page...

IPU_CSI0_CCIR_CODE_1 field descriptions (continued)

Field	Description
5–3 CSI0_STRT_ FLD0_BLNK_ 1ST	Start of field 0 first blanking line command (interlaces mode). (In progressive mode this field indicates start of blanking line command).
CSI0_END_ FLD0_BLNK_ 1ST	End of field 0 first blanking line command (interlaces mode). (In progressive mode this field is ignored).

38.5.157 CSI0 CCIR Code Register 2 (IPU_CSI0_CCIR_CODE_2)

Address: 260_0000h base + 3_0018h offset = 263_0018h

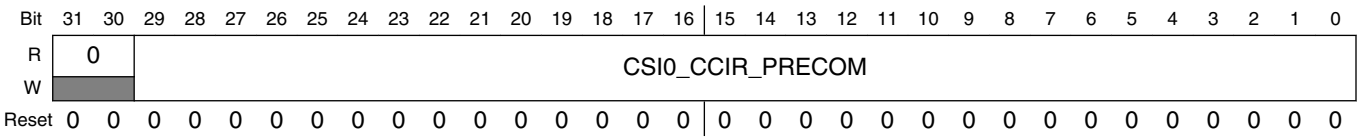
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										CSI0_STRT_FLD1_ACTV			CSI0_END_FLD1_ACTV			0				CSI0_STRT_FLD1_BLNK_2ND			CSI0_END_FLD1_BLNK_2ND			CSI0_STRT_FLD1_BLNK_1ST			CSI0_END_FLD1_BLNK_1ST		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI0_CCIR_CODE_2 field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value 0.
21–19 CSI0_STRT_ FLD1_ACTV	Start of field 1 active line command (interlaces mode). (In progressive mode this field is ignored).
18–16 CSI0_END_ FLD1_ACTV	End of field 1 active line command (interlaces mode). (In progressive mode this field is ignored).
15–12 Reserved	This read-only field is reserved and always has the value 0.
11–9 CSI0_STRT_ FLD1_BLNK_ 2ND	Start of field 1 second blanking line command (interlaces mode). (In progressive mode this field is ignored).
8–6 CSI0_END_ FLD1_BLNK_ 2ND	End of field 1 second blanking line command (interlaces mode). (In progressive mode this field is ignored).
5–3 CSI0_STRT_ FLD1_BLNK_ 1ST	Start of field 1 first blanking line command (interlaces mode). (In progressive mode this field is ignored).
CSI0_END_ FLD1_BLNK_ 1ST	End of field 1 first blanking line command (interlaces mode). (In progressive mode this field is ignored).

38.5.158 CSI0 CCIR Code Register 3 (IPU_CSI0_CCIR_CODE_3)

Address: 260_0000h base + 3_001Ch offset = 263_001Ch

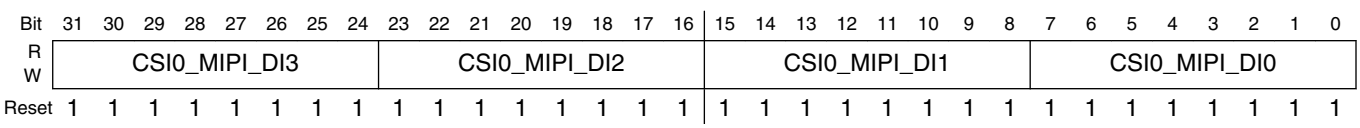


IPU_CSI0_CCIR_CODE_3 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value 0.
CSI0_CCIR_ PRECOM	CCIR pre command. This field defines the sequence which comes before the CCIR command. For BT.656 the code should be written to bits [23:0] while bits [29:24] are ignored (3X8bit) For BT.1120 the code should be written to bits [29:0] (3X10bit)

38.5.159 CSI0 Data Identifier Register (IPU_CSI0_DI)

Address: 260_0000h base + 3_0020h offset = 263_0020h



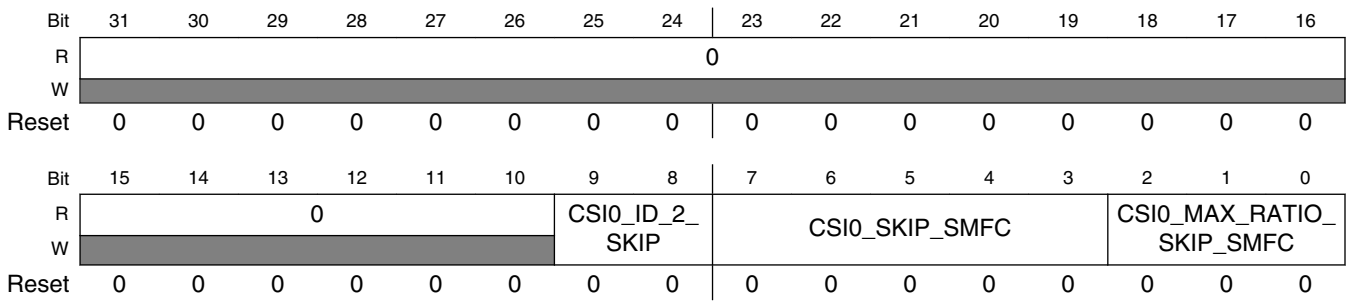
IPU_CSI0_DI field descriptions

Field	Description
31–24 CSI0_MIPI_DI3	CSI0_MIPI_DI3 This field holds the Data Identifier #3 handled by the CSI. The upper two bits should always be written as 00.
23–16 CSI0_MIPI_DI2	CSI0_MIPI_DI2 This field holds the Data Identifier #2 handled by the CSI. The upper two bits should always be written as 00.
15–8 CSI0_MIPI_DI1	CSI0_MIPI_DI1 This field holds the Data Identifier #1 handled by the CSI. The upper two bits should always be written as 00.
CSI0_MIPI_DI0	CSI0_MIPI_DI0 This field holds the Data Identifier #0 handled by the CSI. The upper two bits should always be written as 00. This is the main stream.

38.5.160 CSI0 SKIP Register (IPU_CSI0_SKIP)

This register controls the frame skipping supported between CSI0 and the SMFC.

Address: 260_0000h base + 3_0024h offset = 263_0024h



IPU_CSI0_SKIP field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value 0.
9–8 CSI0_ID_2_SKIP	CSI0 to SMFC Skipping ID. Data from the CSI0 to the SMFC has an ID associated with it. The ID is received from the MIPI interface. The skipping mechanism between the CSI0 and the SMFC can be used for only one ID. There is no skipping for data coming with ID different from the ID programed in this bits 00 - Skipping mechanism is activated on frames with ID equal to 00 01 - Skipping mechanism is activated on frames with ID equal to 01 10 - Skipping mechanism is activated on frames with ID equal to 10 11 - Skipping mechanism is activated on frames with ID equal to 11
7–3 CSI0_SKIP_SMFC	CSI0 SKIP SMFC These 5 bits define the skipping pattern of the frames send to the SMFC. Skipping is done for a set of frames. The number of frames in a set is defined at CSI0_MAX_RATIO_SKIP_SMFC. when CSI0_MAX_RATIO_SKIP_SMFC = 1 => CSI0_SKIP_SMFC[0] is used; other bits are ignored when CSI0_MAX_RATIO_SKIP_SMFC = 2 => CSI0_SKIP_SMFC[1:0] are used; other bits are ignored when CSI0_MAX_RATIO_SKIP_SMFC = 3 => CSI0_SKIP_SMFC[2:0] are used; other bits are ignored when CSI0_MAX_RATIO_SKIP_SMFC = 4 => CSI0_SKIP_SMFC[3:0] are used; other bits are ignored when CSI0_MAX_RATIO_SKIP_SMFC = 5 => CSI0_SKIP_SMFC[4:0] are used; Setting bit #n of CSI0_SKIP_SMFC means that the #n frame in the set is skipped. For example: if CSI0_MAX_RATIO_SKIP_SMFC = 4 and CSI0_SKIP_SMFC = 11010 Frames #0 & Frame #2 will not be skipped as bit0 and bit2 are cleared Frames #1 & Frame #3 will be skipped as bit1 and bit3 are set bit #4 is ignored as CSI0_MAX_RATIO_SKIP_SMFC is set to 4

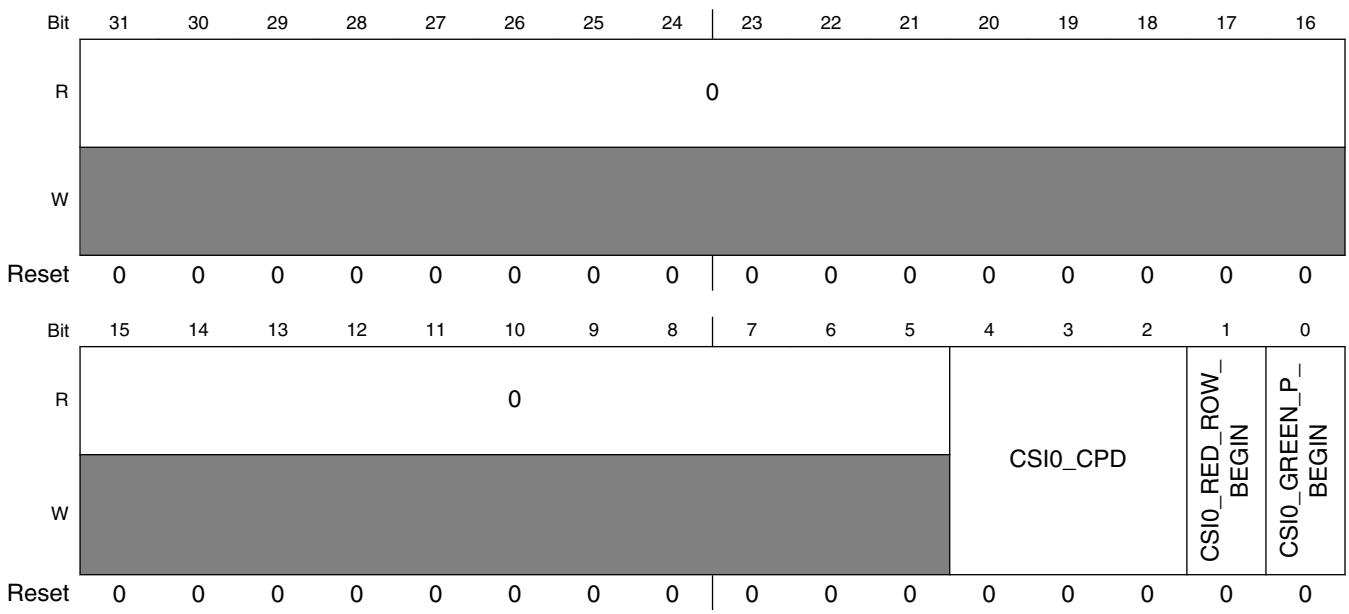
Table continues on the next page...

IPU_CSI0_SKIP field descriptions (continued)

Field	Description
CSI0_MAX_ RATIO_SKIP_ SMFC	CSI0 Maximum Ratio Skip for SMFC These bits define the number of frames in a skipping set. The skipping number is equal to CSI0_MAX_RATIO_SKIP_SMFC+1; The maximum value of this bits is 5. When set to 0 the skipping is disabled.

38.5.161 CSI0 Compander Control Register (IPU_CSI0_CPD_CTRL)

Address: 260_0000h base + 3_0028h offset = 263_0028h



IPU_CSI0_CPD_CTRL field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4–2 CSI0_CPD	CSI0_CPD These bits enable the compander in the path to different destination. CSI0_CPD[0] - CSI0_CPD[1] - Enable for the compander for data sent to the IC CSI0_CPD[2] - Enable for the compander for data sent to the IDMAC via SMFC If all the 3 bits are zero the compander is disabled Reserved

Table continues on the next page...

IPU_CSI0_CPD_CTRL field descriptions (continued)

Field	Description
1 CSI0_RED_ROW_BEGIN	Color of first row in the frame. Reserved 0 First row in the frame is GBGB. 1 First row in the frame is GRGR.
0 CSI0_GREEN_P_BEGIN	Color of first component in the frame. Reserved 0 First component in the frame is blue or red, depending from RED_ROW bit. 1 First component in the frame is green

38.5.162 CSI0 Red Component Compander Constants Register <i> (IPU_CSI0_CPD_RC_i)

These registers contain CONSTANT <i> parameters used for companding of red component.

Address: 260_0000h base + 3_002Ch offset = 263_002Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CSI0_CPD_RC_2i_1									0							CSI0_CPD_RC_2i								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI0_CPD_RC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 CSI0_CPD_RC_2i_1	CONSTANT <2*i+1> parameter of Compander, Red component. Reserved
15–9 Reserved	This read-only field is reserved and always has the value 0.
CSI0_CPD_RC_2i	CONSTANT <2*i> parameter of Compander, Red component. Reserved

38.5.163 CSI0 Red Component Compander SLOPE Register <i> (IPU_CSI0_CPD_RS_i)

These registers contain SLOPE <i> parameters used for companding of red component.

Address: 260_0000h base + 3_004Ch offset = 263_004Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSI0_CPD_RS_4i_3								CSI0_CPD_RS_4i_2								CSI0_CPD_RS_4i_1								CSI0_CPD_RS_4i							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI0_CPD_RS_i field descriptions

Field	Description
31–24 CSI0_CPD_RS_4i_3	Reserved
23–16 CSI0_CPD_RS_4i_2	Reserved
15–8 CSI0_CPD_RS_4i_1	Reserved
CSI0_CPD_RS_4i	Reserved

38.5.164 CSI0 GR Component Compander Constants Register <i> (IPU_CSI0_CPD_GRC_i)

These registers contain CONSTANTi parameters used for companding of green components in GRGR rows.

Address: 260_0000h base + 3_005Ch offset = 263_005Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	0								CSI0_CPD_GRC_2i_1								0								CSI0_CPD_GRC_2i							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI0_CPD_GRC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 CSI0_CPD_ GRC_2i_1	Reserved
15–9 Reserved	This read-only field is reserved and always has the value 0.
CSI0_CPD_ GRC_2i	Reserved

38.5.165 CSI0 GR Component Compander SLOPE Register <i> (IPU_CSI0_CPD_GRS_i)

These registers contain SLOPEi parameters used for companding of green components in GRGR rows.

Address: 260_0000h base + 3_007Ch offset = 263_007Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSI0_CPD_GRS_4i_3								CSI0_CPD_GRS_4i_2								CSI0_CPD_GRS_4i_1								CSI0_CPD_GRS_4i							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI0_CPD_GRS_i field descriptions

Field	Description
31–24 CSI0_CPD_ GRS_4i_3	Reserved
23–16 CSI0_CPD_ GRS_4i_2	Reserved
15–8 CSI0_CPD_ GRS_4i_1	Reserved
CSI0_CPD_ GRS_4i	Reserved

38.5.166 CSI0 GB Component Compander Constants Register <i> (IPU_CSI0_CPD_GBC_i)

These registers contain CONSTANTi parameters used for companding of green components in GBGB rows.

Address: 260_0000h base + 3_008Ch offset = 263_008Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CSI0_CPD_GBC_2i_1								0								CSI0_CPD_GBC_2i							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI0_CPD_GBC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 CSI0_CPD_ GBC_2i_1	Reserved
15–9 Reserved	This read-only field is reserved and always has the value 0.
CSI0_CPD_ GBC_2i	Reserved

38.5.167 CSI0 GB Component Compander SLOPE Register <i> (IPU_CSI0_CPD_GBS_i)

These registers contain SLOPEi parameters used for companding of green components in GBGB rows.

Address: 260_0000h base + 3_00ACh offset = 263_00ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSI0_CPD_GBS_4i_3								CSI0_CPD_GBS_4i_2								CSI0_CPD_GBS_4i_1								CSI0_CPD_GBS_4i							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

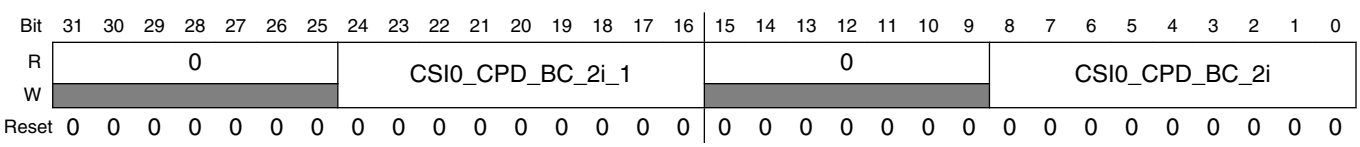
IPU_CSI0_CPD_GBS_i field descriptions

Field	Description
31–24 CSI0_CPD_ GBS_4i_3	Reserved
23–16 CSI0_CPD_ GBS_4i_2	Reserved
15–8 CSI0_CPD_ GBS_4i_1	Reserved
CSI0_CPD_ GBS_4i	Reserved

38.5.168 CSI0 Blue Component Compander Constants Register <i> (IPU_CSI0_CPD_BC_i)

These registers contain CONSTANTi parameters used for companding of blue component.

Address: 260_0000h base + 3_00BCh offset = 263_00BCh



IPU_CSI0_CPD_BC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 CSI0_CPD_BC_ 2i_1	Reserved
15–9 Reserved	This read-only field is reserved and always has the value 0.
CSI0_CPD_BC_ 2i	Reserved

38.5.169 CSI0 Blue Component Compander SLOPE Register <i> (IPU_CSI0_CPD_BS_i)

These registers contain SLOPEi parameters used for companding of red component.

Address: 260_0000h base + 3_00DCh offset = 263_00DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSI0_CPD_BS_4i_3								CSI0_CPD_BS_4i_2								CSI0_CPD_BS_4i_1								CSI0_CPD_BS_4i							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI0_CPD_BS_i field descriptions

Field	Description
31–24 CSI0_CPD_BS_4i_3	Reserved
23–16 CSI0_CPD_BS_4i_2	Reserved
15–8 CSI0_CPD_BS_4i_1	Reserved
CSI0_CPD_BS_4i	Reserved I

38.5.170 CSI0 Compander Offset Register 1 (IPU_CSI0_CPD_OFFSET1)

These registers contain Offset parameters used for companding.

Address: 260_0000h base + 3_00ECh offset = 263_00ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	0	CSI0_CPD_B_OFFSET											CSI0_GB_OFFSET										CSI0_GR_OFFSET									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

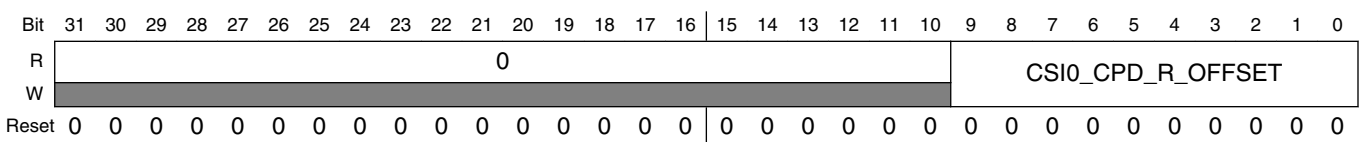
IPU_CSI0_CPD_OFFSET1 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value 0.
29–20 CSI0_CPD_B_ OFFSET	Reserved
19–10 CSI0_GB_ OFFSET	Reserved
CSI0_GR_ OFFSET	Reserved

38.5.171 CSI0 Compander Offset Register 2 (IPU_CSI0_CPD_OFFSET2)

This register contain Offset parameters used for companding.

Address: 260_0000h base + 3_00F0h offset = 263_00F0h

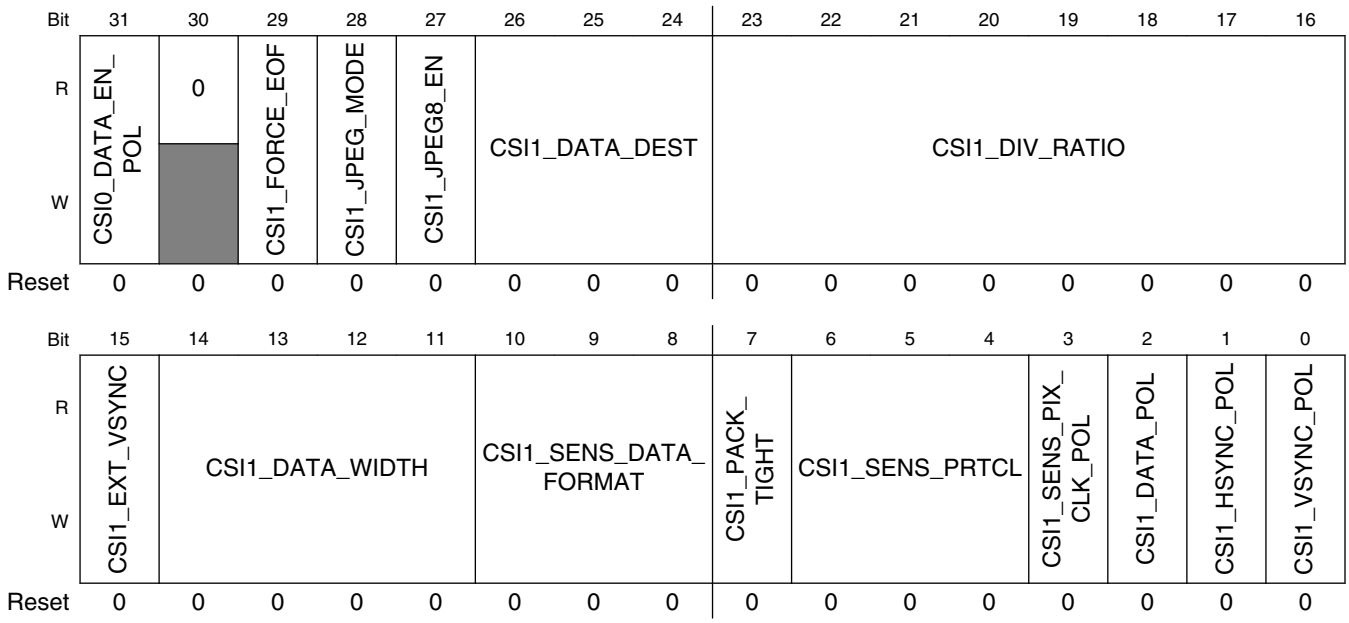


IPU_CSI0_CPD_OFFSET2 field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value 0.
CSI0_CPD_R_ OFFSET	CSI0 Red component offset The value is between -512 to 511. The value is added to the red component before companding. Clipping: If the result of the red components value + the offset is smaller than 0, the result is zero If the result of the red components value + the offset is greater than 1023, the result is 1023 Reserved

38.5.172 CSI1 Sensor Configuration Register (IPU_CSI1_SENS_CONF)

Address: 260_0000h base + 3_8000h offset = 263_8000h



IPU_CSI1_SENS_CONF field descriptions

Field	Description
31 CSI0_DATA_EN_POL	Invert IPP_IND_SENSB_DATA_EN input. This bit selects the polarity of IPP_IND_SENSB_DATA_EN signal. 0 IPP_IND_SENSB_DATA_EN is directly applied to internal circuitry. 1 IPP_IND_SENSB_DATA_EN is inverted before applied to internal circuitry.
30 Reserved	This read-only field is reserved and always has the value 0.
29 CSI1_FORCE_EOF	Force End of frame This is a self clear bit allowing the user to force an End-of-frame event; This bit can be used in cases where the frame sent by the sensor was not completed. 1 force end of frame 0 no action
28 CSI1_JPEG_MODE	JPEG Mode - this bit defines the mode of the control signals when working in JPEG mode 1 The data is valid as long as HSYNC and VSYNC signals are active; HSYNC is valid for single frame 0 The frame starts withe the assertion of VSYNC. The frame ends on the next VSYNC or by setting the **CSI0_FORCE_EOF bit**
27 CSI1_JPEG8_EN	JPEG8 enable bit 1 JPEG8 detection is enabled 0 JPEG8 is disabled

Table continues on the next page...

IPU_CSI1_SENS_CONF field descriptions (continued)

Field	Description
26–24 CSI1_DATA_DEST	These bits enable the destination of the data coming from the CSI. CSI1_DATA_DEST[0] - Reserved CSI1_DATA_DEST[1] - destination is IC CSI1_DATA_DEST[2] - destination is IDMAC via SMFC
23–16 CSI1_DIV_RATIO	DIV Ratio Clock division ratio minus 1. This field defines the division ratio of HSP_CLK into SENSB_MCLK: SENSB_MCLK rate = HSP_CLK rate /(DIV_RATIO+1)
15 CSI1_EXT_VSYNC	External VSYNC enable. This bits select between external and internal VSYNC. 0 Internal VSYNC mode. 1 External VSYNC mode.
14–11 CSI1_DATA_WIDTH	Data width. This fields defines the number of bits per color. Values: 0000 4 bits per color 0000 Reserved 0001 8 bits per color 0010 9 bits per color 0010 Reserved 0011 10 bits per color 0100 11 bits per color 0100 Reserved 0101 12 bits per color 0101 Reserved 0110 13 bits per color 0110 Reserved 0111 14 bits per color 0111 Reserved 1000 15 bits per color 1000 Reserved 1001 16 bits per color
10–8 CSI1_SENS_DATA_FORMAT	Data format from the sensor. This field defines the data format for the input of the CSI sensor. Values: 000 full RGB or YUV444 001 YUV422 (YUYV...) 010 YUV422 (UYVY...) 011 Bayer or Generic data 100 RGB565 101 RGB555 110 RGB444 111 JPEG
7 CSI1_PACK_TIGHT	**CSI1 Pack Tight** When the data format is YUV or RGB and the component's width is 9-16 bits, it can be sent to the memory in 2 different ways

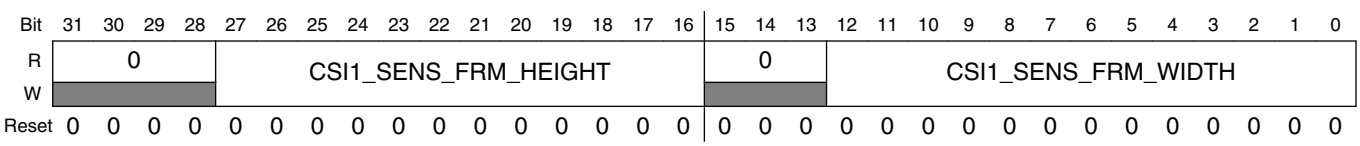
Table continues on the next page...

IPU_CSI1_SENS_CONF field descriptions (continued)

Field	Description
	1 Three 10 bits components are packed into a 32 bit word. Color extension/reduction is performed 0 Each component is written as a 16 bit word where the MSB is written to bit #15, color extension is done for the remaining least significant bits.
6–4 CSI1_SENS_PRTCL	Sensor Protocol. This bit defines the Sensor timing/data mode protocol. Values: 000 Gated clock mode 001 Non-gated clock mode 010 CCIR progressive mode (BT.656) 011 CCIR interlaced mode (BT.656) 100 CCIR progressive (BT.1120 DDR mode: data arrives on every edge of the clock) 101 CCIR progressive (BT.1120 SDR mode: data arrives only on the positive edge of the clock) 110 CCIR interlaced mode (BT.1120 DDR mode: data arrives on every edge of the clock) 111 CCIR interlaced mode (BT.1120 SDR mode: data arrives only on the positive edge of the clock)
3 CSI1_SENS_PIX_CLK_POL	Invert Pixel clock input. This bit selects the polarity of pixel clock. 0 pixel clock is directly applied to internal circuitry. 1 pixel clock is inverted before applied to internal circuitry.
2 CSI1_DATA_POL	Invert data input. This bit selects the polarity of data input. 0 data lines are directly applied to internal circuitry. 1 data lines are inverted before applied to internal circuitry.
1 CSI1_HSYNC_POL	Invert IPP_IND_SENSB_HSYNC input. This bit selects the polarity of IPP_IND_SENSB_HSYNC signal. 0 IPP_IND_SENSB_HSYNC is directly applied to internal circuitry. 1 IPP_IND_SENSB_HSYNC is inverted before applied to internal circuitry.
0 CSI1_VSYNC_POL	Invert IPP_IND_SENSB_VSYNC input. This bit selects the polarity of IPP_IND_SENSB_VSYNC signal. 0 IPP_IND_SENSB_VSYNC is not inverted before applied to internal circuitry. 1 IPP_IND_SENSB_VSYNC is inverted before applied to internal circuitry.

38.5.173 CSI1 Sense Frame Size Register (IPU_CSI1_SENS_FRM_SIZE)

Address: 260_0000h base + 3_8004h offset = 263_8004h



IPU_CSI1_SENS_FRM_SIZE field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.

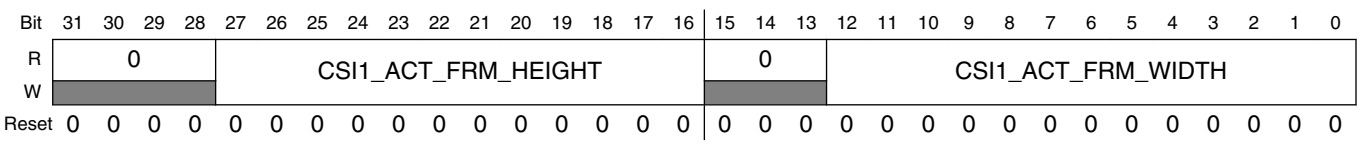
Table continues on the next page...

IPU_CSI1_SENS_FRM_SIZE field descriptions (continued)

Field	Description
27–16 CSI1_SENS_ FRM_HEIGHT	Sensor frame height minus 1. This field defines the sensor frame rows number minus 1.
15–13 Reserved	This read-only field is reserved and always has the value 0.
CSI1_SENS_ FRM_WIDTH	Sensor frame width minus 1. This field defines the sensor frame column number minus 1.

38.5.174 CSI1 Actual Frame Size Register (IPU_CSI1_ACT_FRM_SIZE)

Address: 260_0000h base + 3_8008h offset = 263_8008h

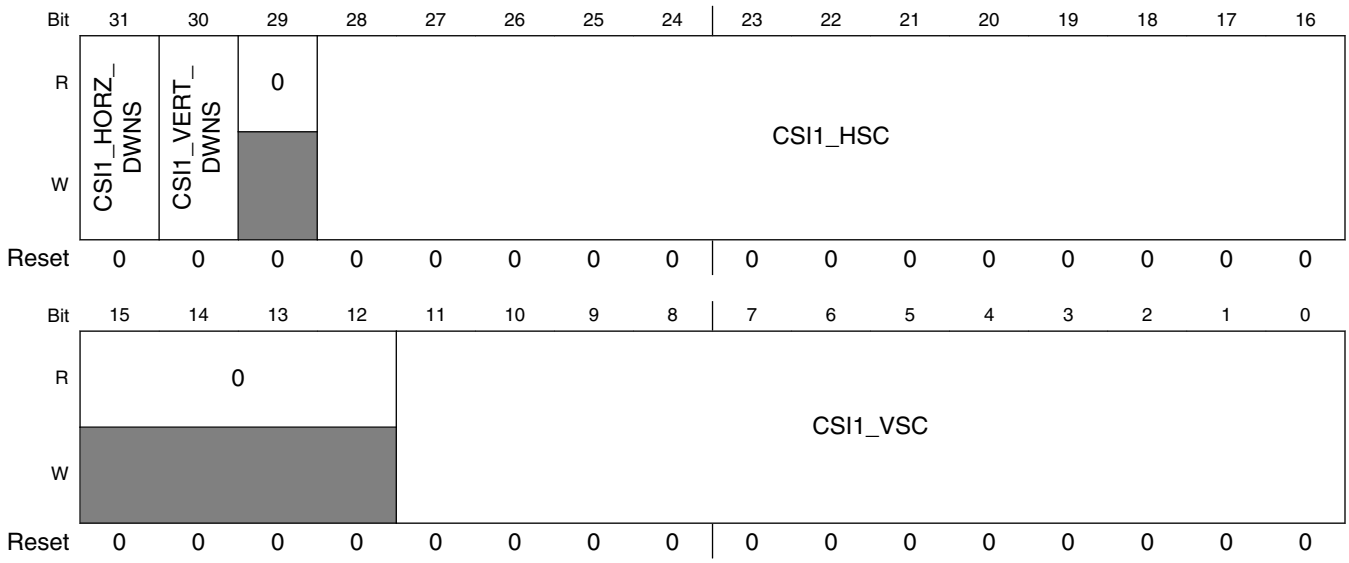


IPU_CSI1_ACT_FRM_SIZE field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–16 CSI1_ACT_ FRM_HEIGHT	Actual frame height minus 1. This field defines the CSI output frame rows number minus 1.
15–13 Reserved	This read-only field is reserved and always has the value 0.
CSI1_ACT_ FRM_WIDTH	Actual frame width minus 1. This field defines the CSI output frame columns number minus 1.

38.5.175 CSI1 Output Control Register (IPU_CSI1_OUT_FRM_CTRL)

Address: 260_0000h base + 3_800Ch offset = 263_800Ch

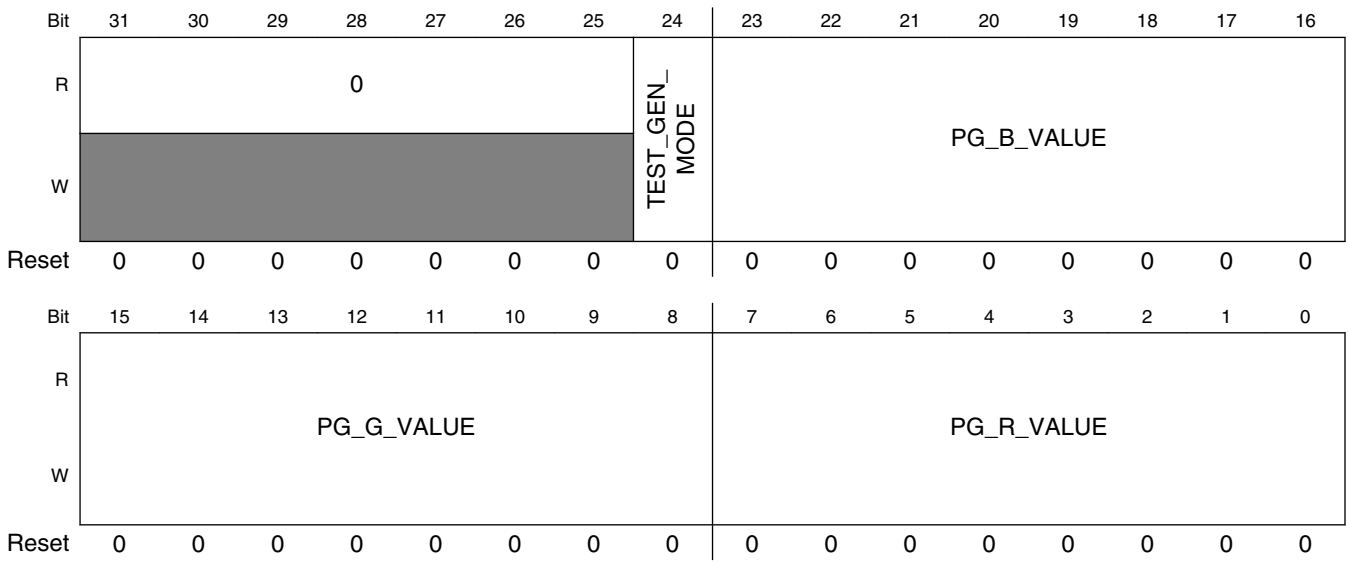


IPU_CSI1_OUT_FRM_CTRL field descriptions

Field	Description
31 CSI1_HORZ_DWNS	Enable horizontal downsizing (decimation) by 2. 0 Downsizing disabled 1 Downsizing enabled
30 CSI1_VERT_DWNS	Enable vertical downsizing (decimation) by 2. 0 Downsizing disabled 1 Downsizing enabled
29 Reserved	This read-only field is reserved and always has the value 0.
28–16 CSI1_HSC	Horizontal skip. This field defines the number of columns to skip. In Interlaced mode this number refers to the number of lines per field
15–12 Reserved	This read-only field is reserved and always has the value 0.
CSI1_VSC	Vertical skip. This field defines the number of rows to skip.

38.5.176 CSI1 Test Control Register (IPU_CSI1_TST_CTRL)

Address: 260_0000h base + 3_8010h offset = 263_8010h

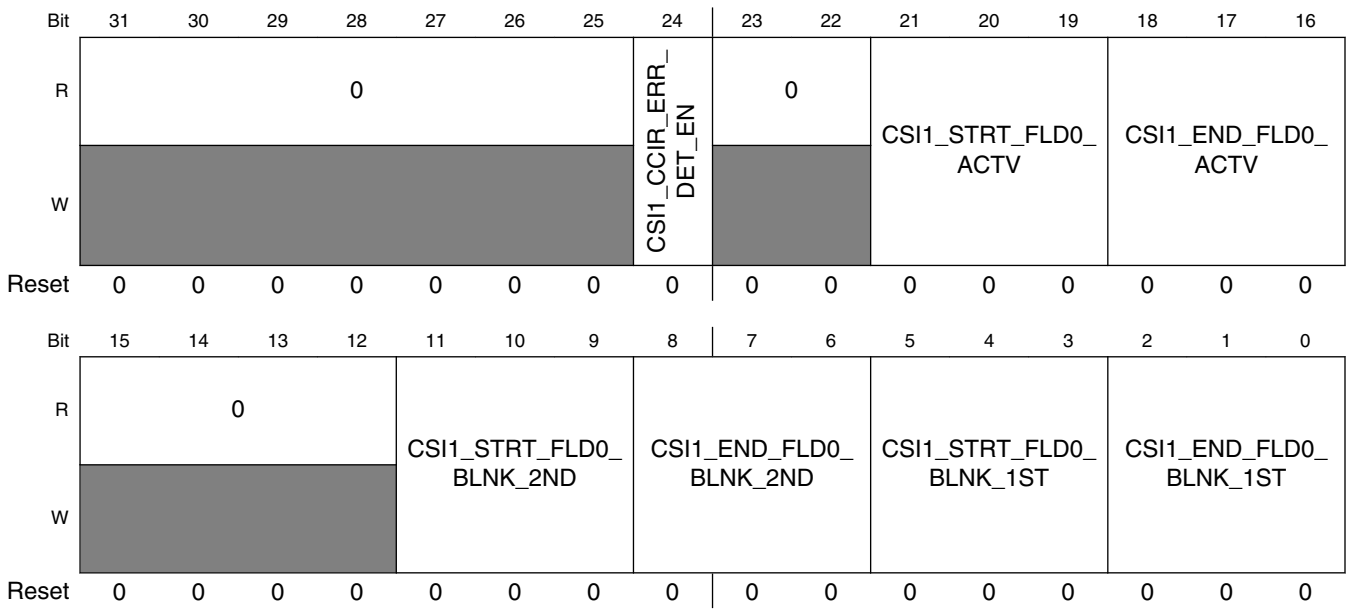


IPU_CSI1_TST_CTRL field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24 TEST_GEN_ MODE	Test generator mode. This bit activates the signal generation. 0 Test signal generator is inactive. 1 Test signal generator is active.
23–16 PG_B_VALUE	Pattern generator B value. This field selects the B value for the generated pattern of even pixel.
15–8 PG_G_VALUE	Pattern generator G value. This field selects the G value for the generated pattern of even pixel.
PG_R_VALUE	Pattern generator R value. This field selects the R value for the generated pattern of even pixel.

38.5.177 CSI1 CCIR Code Register 1 (IPU_CSI1_CCIR_CODE_1)

Address: 260_0000h base + 3_8014h offset = 263_8014h



IPU_CSI1_CCIR_CODE_1 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24 CSI1_CCIR_ERR_DET_EN	Enable error detection and correction for CCIR interlaced mode with protection bit. 0 Error detection and correction is disabled. 1 Error detection and correction is enabled.
23–22 Reserved	This read-only field is reserved and always has the value 0.
21–19 CSI1_STRT_FLD0_ACTV	Start of field 0 active line command (interlaces mode). (In progressive mode, start of active line command mode).
18–16 CSI1_END_FLD0_ACTV	End of field 0 active line command (interlaces mode). (In progressive mode, end of active line command mode).
15–12 Reserved	This read-only field is reserved and always has the value 0.
11–9 CSI1_STRT_FLD0_BLNK_2ND	Start of field 0 second blanking line command (interlaces mode). (In progressive mode this field is ignored).
8–6 CSI1_END_FLD0_BLNK_2ND	End of field 0 second blanking line command (interlaces mode). (In progressive mode this field is ignored).

Table continues on the next page...

IPU_CSI1_CCIR_CODE_1 field descriptions (continued)

Field	Description
5–3 CSI1_STRT_ FLD0_BLNK_ 1ST	Start of field 0 first blanking line command (interlaces mode). (In progressive mode this field indicates start of blanking line command).
CSI1_END_ FLD0_BLNK_ 1ST	End of field 0 first blanking line command (interlaces mode). (In progressive mode this field is ignored).

38.5.178 CSI1 CCIR Code Register 2 (IPU_CSI1_CCIR_CODE_2)

Address: 260_0000h base + 3_8018h offset = 263_8018h

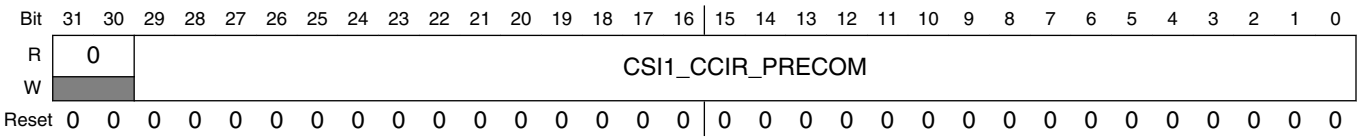
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										CSI1_STRT_FLD1_ACTV			CSI1_END_FLD1_ACTV			0				CSI1_STRT_FLD1_BLNK_2ND			CSI1_END_FLD1_BLNK_2ND			CSI1_STRT_FLD1_BLNK_1ST			CSI1_END_FLD1_BLNK_1ST		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_CCIR_CODE_2 field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value 0.
21–19 CSI1_STRT_ FLD1_ACTV	Start of field 1 active line command (interlaces mode). (In progressive mode this field is ignored).
18–16 CSI1_END_ FLD1_ACTV	End of field 1 active line command (interlaces mode). (In progressive mode this field is ignored).
15–12 Reserved	This read-only field is reserved and always has the value 0.
11–9 CSI1_STRT_ FLD1_BLNK_ 2ND	Start of field 1 second blanking line command (interlaces mode). (In progressive mode this field is ignored).
8–6 CSI1_END_ FLD1_BLNK_ 2ND	End of field 1 second blanking line command (interlaces mode). (In progressive mode this field is ignored).
5–3 CSI1_STRT_ FLD1_BLNK_ 1ST	Start of field 1 first blanking line command (interlaces mode). (In progressive mode this field is ignored).
CSI1_END_ FLD1_BLNK_ 1ST	End of field 1 first blanking line command (interlaces mode). (In progressive mode this field is ignored).

38.5.179 CSI1 CCIR Code Register 3 (IPU_CSI1_CCIR_CODE_3)

Address: 260_0000h base + 3_801Ch offset = 263_801Ch

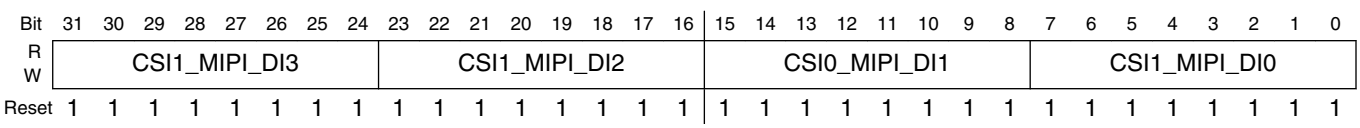


IPU_CSI1_CCIR_CODE_3 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value 0.
CSI1_CCIR_ PRECOM	CCIR pre command. This field defines the sequence which comes before the CCIR command. For BT.656 the code should be written to bits [23:0] while bits [29:24] are ignored (3X8bit) For BT.1120 the code should be written to bits [29:0] (3X10bit)

38.5.180 CSI1 Data Identifier Register (IPU_CSI1_DI)

Address: 260_0000h base + 3_8020h offset = 263_8020h



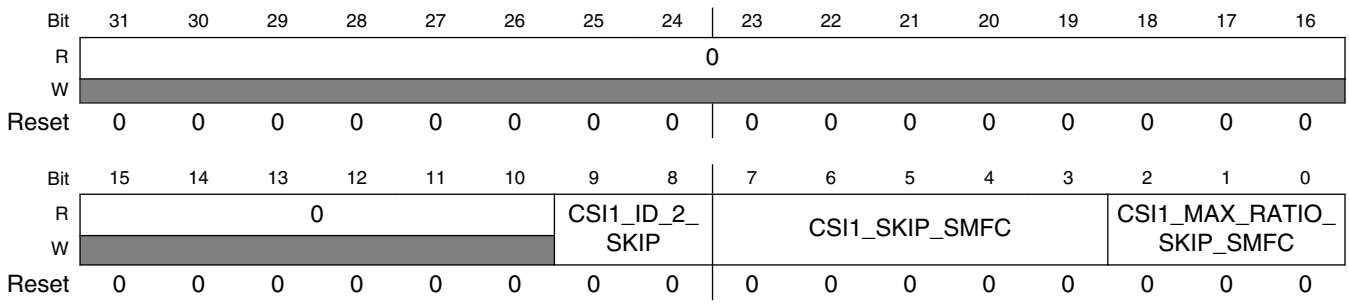
IPU_CSI1_DI field descriptions

Field	Description
31–24 CSI1_MIPI_DI3	CSI1_MIPI_DI3 This field holds the Data Identifier #3 handled by the CSI. The upper two bits should always be written as 00.
23–16 CSI1_MIPI_DI2	CSI1_MIPI_DI2 This field holds the Data Identifier #2 handled by the CSI. The upper two bits should always be written as 00.
15–8 CSI0_MIPI_DI1	CSI1_MIPI_DI1 This field holds the Data Identifier #1 handled by the CSI. The upper two bits should always be written as 00.
CSI1_MIPI_DI0	CSI1_MIPI_DI0 This field holds the Data Identifier #0 handled by the CSI. The upper two bits should always be written as 00. This is the main stream.

38.5.181 CSI1 SKIP Register (IPU_CSI1_SKIP)

This register control the frame skipping supported between CSI1 and the SMFC.

Address: 260_0000h base + 3_8024h offset = 263_8024h



IPU_CSI1_SKIP field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value 0.
9–8 CSI1_ID_2_SKIP	CSI1 to SMFC Skipping ID. Data from the CSI1 to the SMFC has an ID associated with it. The ID is received from the MIPI interface. The skipping mechanism between the CSI1 and the SMFC can be used for only one ID. There is no skipping for data coming with ID different from the ID programed in this bits 00 - Skipping mechanism is activated on frames with ID equal to 00 01 - Skipping mechanism is activated on frames with ID equal to 01 10 - Skipping mechanism is activated on frames with ID equal to 10 11 - Skipping mechanism is activated on frames with ID equal to 11
7–3 CSI1_SKIP_SMFC	CSI1 SKIP SMFC These 5 bits define the skipping pattern of the frames send to the SMFC. Skipping is done for a set of frames. The number of frames in a set is defined at CSI1_MAX_RATIO_SKIP_SMFC. when CSI1_MAX_RATIO_SKIP_SMFC = 1 => CSI1_SKIP_SMFC[0] is used; other bits are ignored when CSI1_MAX_RATIO_SKIP_SMFC = 2 => CSI1_SKIP_SMFC[1:0] are used; other bits are ignored when CSI1_MAX_RATIO_SKIP_SMFC = 3 => CSI1_SKIP_SMFC[2:0] are used; other bits are ignored when CSI1_MAX_RATIO_SKIP_SMFC = 4 => CSI1_SKIP_SMFC[3:0] are used; other bits are ignored when CSI1_MAX_RATIO_SKIP_SMFC = 5 => CSI1_SKIP_SMFC[4:0] are used; Setting bit #n of CSI1_SKIP_SMFC means that the #n frame in the set is skipped. For example: if CSI1_MAX_RATIO_SKIP_SMFC = 4 and CSI1_SKIP_SMFC = 11010 Frames #0 & Frame #2 will not be skipped as bit0 and bit2 are cleared Frames #1 & Frame #3 will be skipped as bit1 and bit3 are set bit #4 is ignored as CSI1_MAX_RATIO_SKIP_SMFC is set to 4

Table continues on the next page...

IPU_CSI1_SKIP field descriptions (continued)

Field	Description
CSI1_MAX_ RATIO_SKIP_ SMFC	CSI1 Maximum Ratio Skip for SMFC These bits define the number of frames in a skipping set. These bits define the number of frames in a skipping set. The skipping number is equal to CSI1_MAX_RATIO_SKIP_SMFC+1; The maximum value of this bits is 5. When set to 0 the skipping is disabled.

38.5.182 CSI1 Compander Control Register (IPU_CSI1_CPD_CTRL)

Address: 260_0000h base + 3_8028h offset = 263_8028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											0			0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

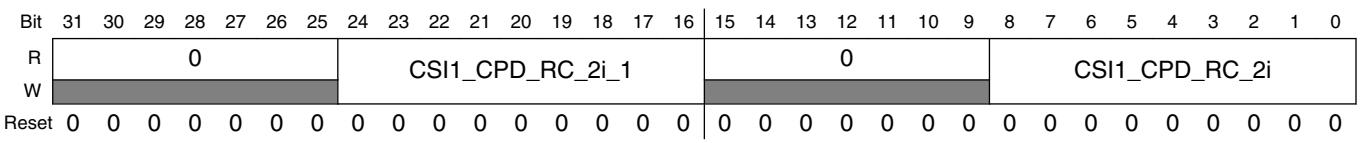
IPU_CSI1_CPD_CTRL field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4–2 Reserved	This read-only field is reserved and always has the value 0.
1 Reserved	This read-only field is reserved and always has the value 0.
0 Reserved	This read-only field is reserved and always has the value 0.

38.5.183 CSI1 Red Component Compander Constants Register <i> (IPU_CSI1_CPD_RC_i)

These registers contain CONSTANT <i> parameters used for companding of red component.

Address: 260_0000h base + 3_802Ch offset = 263_802Ch



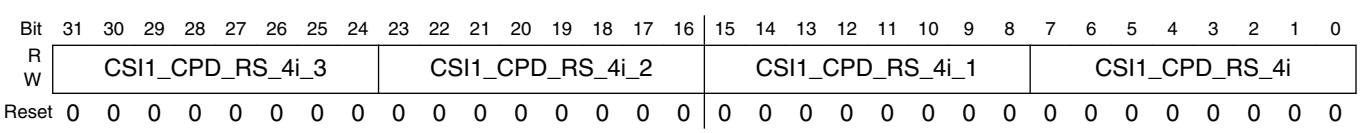
IPU_CSI1_CPD_RC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 CSI1_CPD_RC_2i_1	CONSTANT <2*i+1> parameter of Compander, Red component. Reserved
15–9 Reserved	This read-only field is reserved and always has the value 0.
CSI1_CPD_RC_2i	CONSTANT <2*i> parameter of Compander, Red component. Reserved

38.5.184 CSI1 Red Component Compander SLOPE Register <i> (IPU_CSI1_CPD_RS_i)

These registers contain SLOPE <i> parameters used for companding of red component.

Address: 260_0000h base + 3_804Ch offset = 263_804Ch



IPU_CSI1_CPD_RS_i field descriptions

Field	Description
31–24 CSI1_CPD_RS_4i_3	SLOPE<4*i+3> parameter of Compander, Red component. Reserved
23–16 CSI1_CPD_RS_4i_2	SLOPE<4*i+2> parameter of Compander, Red component. Reserved
15–8 CSI1_CPD_RS_4i_1	SLOPE<4*i+1> parameter of Compander, Red component. Reserved
CSI1_CPD_RS_4i	SLOPE<4*i> parameter of Compander, Red component. Reserved

38.5.185 CSI1 GR Component Compander Constants Register <i> (IPU_CSI1_CPD_GRC_i)

These registers contain CONSTANTi parameters used for companding of green components in GRGR rows.

Address: 260_0000h base + 3_805Ch offset = 263_805Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CSI1_CPD_GRC_2i_1									0							CSI1_CPD_GRC_2i								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_CPD_GRC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 CSI1_CPD_GRC_2i_1	CONST<2*i+1> parameter of Compander, GR component. If the input format is RGB/YUV then CSI1_CPD_GRC should be equal to CSI1_CPD_GBC Reserved
15–9 Reserved	This read-only field is reserved and always has the value 0.
CSI1_CPD_GRC_2i	CONSTANT<2*i> parameter of Compander, GR component. If the input format is RGB/YUV then CSI1_CPD_GRC should be equal to CSI1_CPD_GBC Reserved

38.5.186 CSI1 GR Component Compander SLOPE Register <i> (IPU_CSI1_CPD_GRS_i)

These registers contain SLOPEi parameters used for companding of green components in GRGR rows.

Address: 260_0000h base + 3_807Ch offset = 263_807Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSI1_CPD_GRS_4i_3								CSI1_CPD_GRS_4i_2								CSI1_CPD_GRS_4i_1								CSI1_CPD_GRS_4i							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_CPD_GRS_i field descriptions

Field	Description
31–24 CSI1_CPD_ GRS_4i_3	SLOPE<4*i+3> parameter of Compander, GR component. If the input format is RGB/YUV then CSI1_CPD_GRS should be equal to CSI1_CPD_GBS Reserved
23–16 CSI1_CPD_ GRS_4i_2	SLOPE<4*i+2> parameter of Compander, GR component. If the input format is RGB/YUV then CSI1_CPD_GRS should be equal to CSI1_CPD_GBS Reserved
15–8 CSI1_CPD_ GRS_4i_1	SLOPE<4*i+1> parameter of Compander, GR component. If the input format is RGB/YUV then CSI1_CPD_GRS should be equal to CSI1_CPD_GBS Reserved
CSI1_CPD_ GRS_4i	SLOPE<4*i> parameter of Compander, GR component. If the input format is RGB/YUV then CSI1_CPD_GRS should be equal to CSI1_CPD_GBS Reserved

38.5.187 CSI1 GB Component Compander Constants Register <i> (IPU_CSI1_CPD_GBC_i)

These registers contain CONSTANTi parameters used for companding of green components in GBGB rows.

Address: 260_0000h base + 3_808Ch offset = 263_808Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	0							CSI1_CPD_GBC_2i_1									0							CSI1_CPD_GBC_2i							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_CPD_GBC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 CSI1_CPD_GBC_2i_1	CONSTi+1 parameter of Compander, GB component. If the input format is RGB/YUV then CSI1_CPD_GBC should be equal to CSI1_CPD_GRC Reserved
15–9 Reserved	This read-only field is reserved and always has the value 0.
CSI1_CPD_GBC_2i	CONSTANTi parameter of Compander, GB component. If the input format is RGB/YUV then CSI1_CPD_GBC should be equal to CSI1_CPD_GRC Reserved

38.5.188 CSI1 GB Component Compander SLOPE Register <i> (IPU_CSI1_CPD_GBS_i)

These registers contain SLOPEi parameters used for companding of green components in GBGB rows.

Address: 260_0000h base + 3_80ACh offset = 263_80ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSI1_CPD_GBS_4i_3								CSI1_CPD_GBS_4i_2								CSI1_CPD_GBS_4i_1								CSI1_CPD_GBS_4i							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

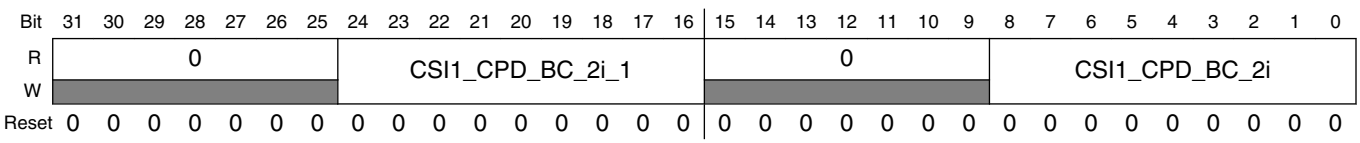
IPU_CSI1_CPD_GBS_i field descriptions

Field	Description
31–24 CSI1_CPD_GBS_4i_3	SLOPE<4*i+3> parameter of Compander, GB component. If the input format is RGB/YUV then CSI1_CPD_GBS should be equal to CSI1_CPD_GRS Reserved
23–16 CSI1_CPD_GBS_4i_2	SLOPE<4*i+2> parameter of Compander, GB component. If the input format is RGB/YUV then CSI1_CPD_GBS should be equal to CSI1_CPD_GRS Reserved
15–8 CSI1_CPD_GBS_4i_1	SLOPE<4*i+1> parameter of Compander, GB component. If the input format is RGB/YUV then CSI1_CPD_GBS should be equal to CSI1_CPD_GRS Reserved
CSI1_CPD_GBS_4i	SLOPE<4*i> parameter of Compander, GB component. If the input format is RGB/YUV then CSI1_CPD_GBS should be equal to CSI1_CPD_GRS Reserved

38.5.189 CSI1 Blue Component Compander Constants Register <i> (IPU_CSI1_CPD_BC_i)

These registers contend CONSTANTi parameters used for companding of blue component.

Address: 260_0000h base + 3_80BCh offset = 263_80BCh



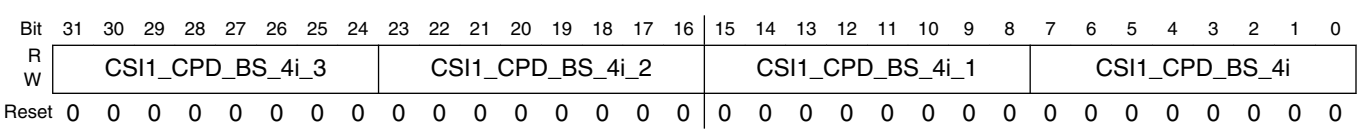
IPU_CSI1_CPD_BC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 CSI1_CPD_BC_2i_1	CONSTANT<2*i+1> parameter of Compander, Blue component. Reserved
15–9 Reserved	This read-only field is reserved and always has the value 0.
CSI1_CPD_BC_2i	CONSTANT<2*i> parameter of Compander, Blue component. Reserved

38.5.190 CSI1 Blue Component Compander SLOPE Register <i> (IPU_CSI1_CPD_BS_i)

This registers contain SLOPEi parameters used for companding of red component.

Address: 260_0000h base + 3_80DCh offset = 263_80DCh



IPU_CSI1_CPD_BS_i field descriptions

Field	Description
31–24 CSI1_CPD_BS_4i_3	SLOPE<4*i+3> parameter of Compander, Blue component. Reserved
23–16 CSI1_CPD_BS_4i_2	SLOPE<4*i+2> parameter of Compander, Blue component. Reserved
15–8 CSI1_CPD_BS_4i_1	SLOPE<4*i+1> parameter of Compander, Blue component. Reserved
CSI1_CPD_BS_4i	SLOPE<4*i> parameter of Compander, Blue component. Reserved

38.5.191 CSI1 Compander Offset Register 1 (IPU_CSI1_CPD_OFFSET1)

These registers contain Offset parameters used for companding.

Address: 260_0000h base + 3_80ECh offset = 263_80ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		CSI1_CPD_B_OFFSET										CSI1_CPD_GB_OFFSET										CSI1_CPD_GR_OFFSET									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_CPD_OFFSET1 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value 0.
29–20 CSI1_CPD_B_OFFSET	CSI1 Blue component offset The value is between -512 to 511. The value is added to the blue component before companding. Clipping: If the result of the blue components value + the offset is smaller than 0, the result is zero If the result of the blue components value + the offset is greater than 1023, the result is 1023 Reserved
19–10 CSI1_CPD_GB_OFFSET	CSI1 Green Blue component offset The value is between -512 to 511. The value is added to the blue component before companding. Clipping: If the result of the green-blue components value + the offset is smaller than 0, the result is zero If the result of the green-blue components value + the offset is greater than 1023, the result is 1023

Table continues on the next page...

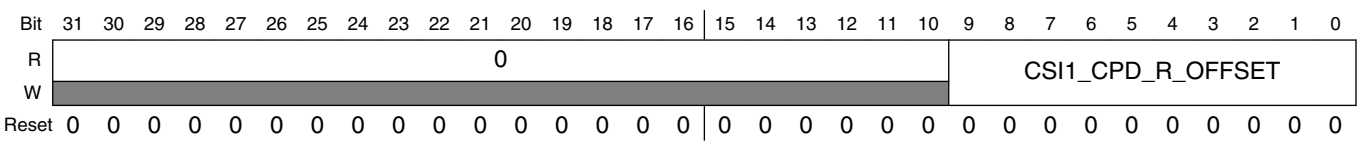
IPU_CSI1_CPD_OFFSET1 field descriptions (continued)

Field	Description
	If the input format is RGB/YUV then CSI1_GB_OFFSET must be equal to CSI1_GR_OFFSET Reserved
CSI1_CPD_GR_OFFSET	CSI1 Green Red component offset The value is between -512 to 511. The value is added to the green-red component before companding. Clipping: If the result of the green-red components value + the offset is smaller than 0, the result is zero If the result of the green-red components value + the offset is greater than 1023, the result is 1023 Reserved

38.5.192 CSI1 Compander Offset Register 2 (IPU_CSI1_CPD_OFFSET2)

These registers contain Offset parameters used for companding.

Address: 260_0000h base + 3_80F0h offset = 263_80F0h



IPU_CSI1_CPD_OFFSET2 field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value 0.
CSI1_CPD_R_OFFSET	CSI1 Red component offset The value is between -512 to 511. The value is added to the red component before companding. Clipping: If the result of the red components value + the offset is smaller than 0, the result is zero If the result of the red components value + the offset is greater than 1023, the result is 1023 Reserved

38.5.193 DI0 General Register (IPU_DI0_GENERAL)

Address: 260_0000h base + 4_0000h offset = 264_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	di0_pin8_pin15_sel	di0_disp_y_sel			DI0_CLOCK_STOP_MODE				DI0_DISP_CLOCK_INIT	di0_mask_sel	di0_vsync_ext	di0_clk_ext	DI0_WATCHDOG_MODE		di0_polarity_disp_clk	0
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	di0_sync_count_sel				di0_err_treatment	di0_erm_vsync_sel	di0_polarity_cs1	di0_polarity_cs0	di0_polarity_i_1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_GENERAL field descriptions

Field	Description
31 di0_pin8_pin15_sel	This bit routes PIN8 over PIN15 1 PIN8 is routed to PIN15, PIN8 is also routed to PIN8 0 PIN15 is routed to PIN15, PIN8 is routed to PIN8
30–28 di0_disp_y_sel	DI0 Display Vertical coordinate (Y) select. This field defines which one of the 8 counters will be used as a display's line counter. 000 counter #1 is selected 111 counter #8 is selected
27–24 DI0_CLOCK_STOP_MODE	DI clock stop mode When performing a clock change. The DI stops the clock to the display. These field defines when the clock will be stopped. Stopping at EOL/EOF is supported for the case where the data is coming from the IDMAC (DMA access). In case that only direct accesses is performed, the user should set this field to 0000 0001-1001 stop at the next event of one of the counters (counter #1 to counter #9) 0000 stop at the next edge of the display clock 1100 stop at EOL (end of a line), but if stop request is during blanking interval, stop now 1101 stop at EOF (end of a frame), but if stop request is during blanking interval, stop now

Table continues on the next page...

IPU_DI0_GENERAL field descriptions (continued)

Field	Description
	1110 stop at EOL (end of a line), but if stop request is during blanking interval, stop at the end of the next line 1111 stop at EOF (end of a frame), but if stop request is during blanking interval, stop at the end of the next frame
23 DI0_DISP_CLOCK_INIT	Display clock's initial mode For synchronization error conditions the display clock can be stopped on the next VSYNC 1 The display's clock is running after the next VSYNC (indicating new frame) 0 The display's clock is stopped after the next VSYNC (indicating new frame)
22 di0_mask_sel	DI0 Mask select. IPP_PIN_2 output of the DI that functions as MASK signal can come from 2 sources: counter #2 or extracted from the MASK data coming from the memory. 1 IPP_PIN_2 is coming from extracted MASK data coming from the memory 0 IPP_PIN_2 is coming from counter #2
21 di0_vsync_ext	DI0 External VSYNC. This bit selects the source of the VSYNC signal 1 External to the IPU 0 Internally generated by the IPU
20 di0_clk_ext	DI0 External Clock. This bit selects the source of the DI0's clock 1 The source of the clock is external to the IPU 0 The clock is internally generated by the IPU
19–18 DI0_WATCHDOG_MODE	DI0 watchdog mode In case of a display error where the DI clock is stopped (defined at di0_err_treatment). An internal watchdog counts DI clocks. If this timer reached its pre defined value the DI will skip the current frame and restart on the frame. This 2 bits define the number of DI clock cycles that the timer counts. 00 The timer counts 4 DI cycles 01 The timer counts 16 DI cycles 10 The timer counts 64 DI cycles 11 The timer counts 128 DI cycles
17 di0_polarity_disp_clk	DI0 Output Clock's polarity This bits define the polarity of the DI0's clock. 1 The output clock is active high 0 The output clock is active low
16 Reserved	This read-only field is reserved and always has the value 0.
15–12 di0_sync_count_sel	For synchronous flow error: selects synchronous flow synchronization counter in DI:
11 di0_err_treatment	In case of synchronous flow error there are 2 ways to handle the display

Table continues on the next page...

IPU_DI0_GENERAL field descriptions (continued)

Field	Description
	1 to wait (stop clock) 0 Drive the last component
10 di0_erm_vsync_sel	DI0 error recovery block's VSYNC source select The error recovery block detect a case where the DI's VSYNC is asserted before the EOF. This bit selects the source of the VSYNC signal monitored by this mechanism. 1 vsync_post - an internal VSYNC signal asserted 2 lines after the DI's VSYNC 0 vsync_pre - an internal VSYNC signal asserted 2 lines before the DI's VSYNC
9 di0_polarity_cs1	DI0 Chip Select's 1 polarity This bits define the polarity of the DI's CS1. 1 The CS1 is active high 0 The CS1 is active low
8 di0_polarity_cs0	DI0 Chip Select's 0 polarity This bits define the polarity of the DI's CS0. 1 The CS0 is active high 0 The CS0 is active low
di0_polarity_i_1	DI0 output pin's polarity This bits define the polarity of each of the DI's outputs. 1 The output pin is active high 0 The output pin is active low

38.5.194 DI0 Base Sync Clock Gen 0 Register (IPU_DI0_BS_CLKGEN0)

Address: 260_0000h base + 4_0004h offset = 264_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							di0_disp_clk_offset									0				di0_disp_clk_period											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_BS_CLKGEN0 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 di0_disp_clk_offset	DI0 Display Clock Offset The DI has the ability to delay the display's clock This field defines the amount of IPU's clock cycles added as delay on this clock.
15–12 Reserved	This read-only field is reserved and always has the value 0.

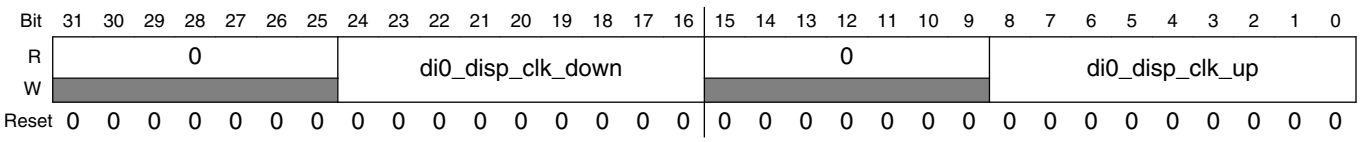
Table continues on the next page...

IPU_DI0_BS_CLKGEN0 field descriptions (continued)

Field	Description
di0_disp_clk_ period	DI0 Display Clock Period This field defines the Display interface clock period for display write access. This parameter contains an integer part (bits 11:4) and a fractional part (bits 3:0). It defines a fractional division ratio of the HSP_CLK clock for generation of the display's interface clock.

38.5.195 DI0 Base Sync Clock Gen 1 Register (IPU_DI0_BS_CLKGEN1)

Address: 260_0000h base + 4_0008h offset = 264_0008h



IPU_DI0_BS_CLKGEN1 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 di0_disp_clk_ down	DI0 display clock falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is a time interval between display's access start point and display's interface clock falling edge.
15–9 Reserved	This read-only field is reserved and always has the value 0.
di0_disp_clk_up	DI0 display clock rising edge position This parameter contains an integer part (bits 8:1) and a fractional part (bit 0). The position value is a time interval between display's access start point and display's interface clock rising edge.

38.5.196 DI0 Sync Wave Gen 1 Register 0 (IPU_DI0_SW_GEN0_1)

Address: 260_0000h base + 4_000Ch offset = 264_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_run_value_m1_1												di0_run_resolution_1		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_offset_value_1												di0_offset_ resolution_1		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN0_1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di0_run_value_ m1_1	DI0 counter #1 pre defined value This fields defines the counter #1 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_1 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_ resolution_1	DI0 counter #1 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 NA 011 NA 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSIs according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.
14–3 di0_offset_value_ 1	DI0 counter #1 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di0_offset_ resolution_1	DI0 counter #1 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock 010 NA 011 NA

Table continues on the next page...

IPU_DI0_SW_GEN0_1 field descriptions (continued)

Field	Description
	100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSIs according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

38.5.197 DI0 Sync Wave Gen 2 Register 0 (IPU_DI0_SW_GEN0_2)

Address: 260_0000h base + 4_0010h offset = 264_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_run_value_m1_2												di0_run_resolution_2		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_offset_value_2												di0_offset_ resolution_2		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN0_2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di0_run_value_ m1_2	DI0 counter #2 pre defined value This fields defines the counter #2 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_2 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_ resolution_2	DI0 counter #2 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock 010 The Counter is triggered by counter #1 011 NA 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DI0_SW_GEN0_2 field descriptions (continued)

Field	Description
14–3 di0_offset_value_2	DI0 counter #2 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di0_offset_resolution_2	DI0 counter #2 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 NA 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSIs according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

38.5.198 DI0 Sync Wave Gen 3 Register 0 (IPU_DI0_SW_GEN0_3)

Address: 260_0000h base + 4_0014h offset = 264_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_run_value_m1_3												di0_run_resolution_3		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_offset_value_3												di0_offset_resolution_3		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN0_3 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di0_run_value_m1_3	DI0 counter #3 pre defined value This fields defines the counter #3 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_3 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_resolution_3	DI0 counter #3 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock.

Table continues on the next page...

IPU_DI0_SW_GEN0_3 field descriptions (continued)

Field	Description
	010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.
14–3 di0_offset_value_3	DI0 counter #3 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di0_offset_resolution_3	DI0 counter #3 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

38.5.199 DI0 Sync Wave Gen 4 Register 0 (IPU_DI0_SW_GEN0_4)

Address: 260_0000h base + 4_0018h offset = 264_0018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_run_value_m1_4												di0_run_resolution_4		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_offset_value_4												di0_offset_resolution_4		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN0_4 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di0_run_value_m1_4	DI0 counter #4 pre defined value This fields defines the counter #4 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_4 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_resolution_4	DI0 counter #4 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.
14–3 di0_offset_value_4	DI0 counter #4 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di0_offset_resolution_4	DI0 counter #4 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

38.5.200 DI0 Sync Wave Gen 5 Register 0 (IPU_DI0_SW_GEN0_5)

Address: 260_0000h base + 4_001Ch offset = 264_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_run_value_m1_5												di0_run_resolution_5		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_offset_value_5												di0_offset_ resolution_5		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN0_5 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di0_run_value_ m1_5	DI0 counter #5 pre defined value This fields defines the counter #5 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_5 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_ resolution_5	DI0 counter #5 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.
14–3 di0_offset_value_ 5	DI0 counter #5 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di0_offset_ resolution_5	DI0 counter #5 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 -The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4

Table continues on the next page...

IPU_DI0_SW_GEN0_5 field descriptions (continued)

Field	Description
	110 External VSYNC 111 Counter is always on.

38.5.201 DI0 Sync Wave Gen 6 Register 0 (IPU_DI0_SW_GEN0_6)

Address: 260_0000h base + 4_0020h offset = 264_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_run_value_m1_6												di0_run_resolution_6		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_offset_value_6												di0_offset_ resolution_6		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN0_6 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di0_run_value_ m1_6	DI0 counter #6 pre defined value This fields defines the counter #6 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_6 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_ resolution_6	DI0 counter #6 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.
14–3 di0_offset_value_ 6	DI0 counter #6 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di0_offset_ resolution_6	DI0 counter #6 offset Resolution This field defines the trigger causing the offset counter to increment

Table continues on the next page...

IPU_DI0_SW_GEN0_6 field descriptions (continued)

Field	Description
	000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

38.5.202 DI0 Sync Wave Gen 7 Register 0 (IPU_DI0_SW_GEN0_7)

Address: 260_0000h base + 4_0024h offset = 264_0024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_run_value_m1_7												di0_run_resolution_7		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_offset_value_7												di0_offset_ resolution_1		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN0_7 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di0_run_value_ m1_7	DI0 counter #7 pre defined value This fields defines the counter #7 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_7 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_ resolution_7	DI0 counter #1 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DI0_SW_GEN0_7 field descriptions (continued)

Field	Description
14–3 di0_offset_value_7	DI0 counter #7 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di0_offset_resolution_1	DI0 counter #7 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

38.5.203 DI0 Sync Wave Gen 8 Register 0 (IPU_DI0_SW_GEN0_8)

Address: 260_0000h base + 4_0028h offset = 264_0028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_run_value_m1_8												di0_run_resolution_8		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_offset_value_8												di0_offset_resolution_8		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN0_8 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di0_run_value_m1_8	DI0 counter #8 pre defined value This fields defines the counter #8 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_8 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_resolution_8	DI0 counter #8 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

Table continues on the next page...

IPU_DI0_SW_GEN0_8 field descriptions (continued)

Field	Description
	100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.
14–3 di0_offset_value_8	DI0 counter #8 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di0_offset_resolution_8	DI0 counter #8 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

38.5.204 DI0 Sync Wave Gen 9 Register 0 (IPU_DI0_SW_GEN0_9)

Address: 260_0000h base + 4_002Ch offset = 264_002Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_run_value_m1_9												di0_run_resolution_9		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_offset_value_9												di0_offset_resolution_9		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN0_9 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di0_run_value_m1_9	DI0 counter #9 pre defined value This fields defines the counter #9 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_9 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.

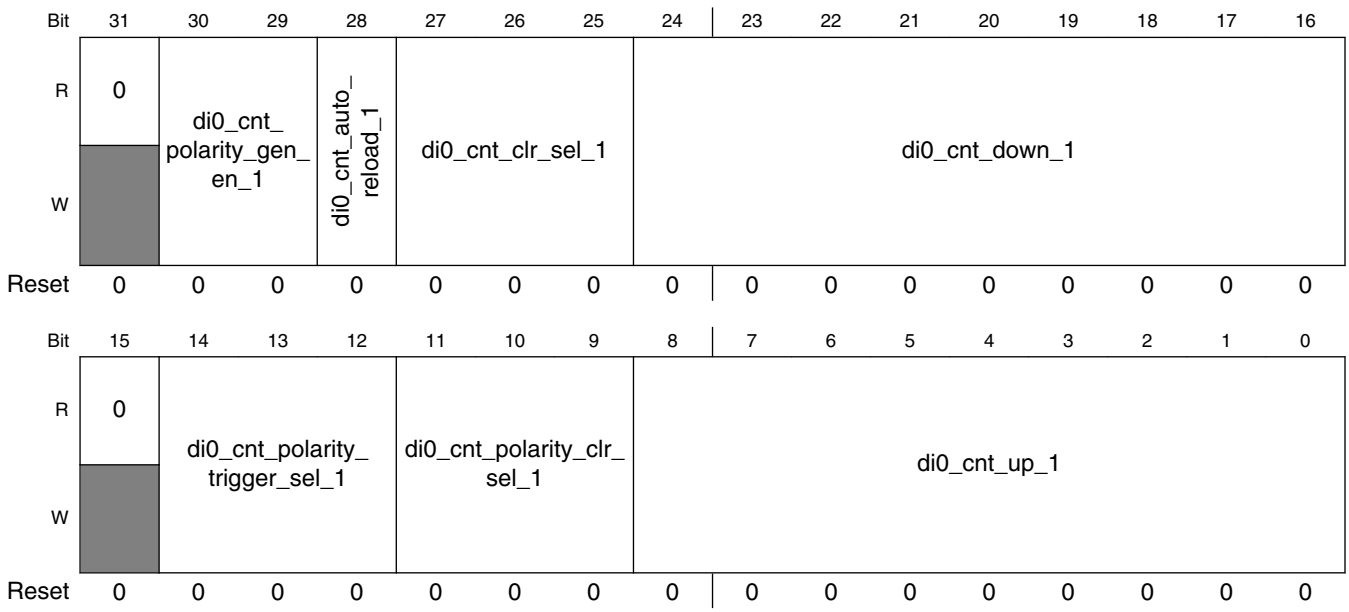
Table continues on the next page...

IPU_DI0_SW_GEN0_9 field descriptions (continued)

Field	Description
18–16 di0_run_ resolution_9	DI0 counter #9 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.
14–3 di0_offset_value_ 9	DI0 counter #9 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di0_offset_ resolution_9	DI0 counter #9 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

38.5.205 DI0 Sync Wave Gen 1 Register 1 (IPU_DI0_SW_GEN1_1)

Address: 260_0000h base + 4_0030h offset = 264_0030h



IPU_DI0_SW_GEN1_1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di0_cnt_polarity_gen_en_1	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_reload_1	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_1 field
27–25 di0_cnt_clr_sel_1	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 Reserved 011 Reserved

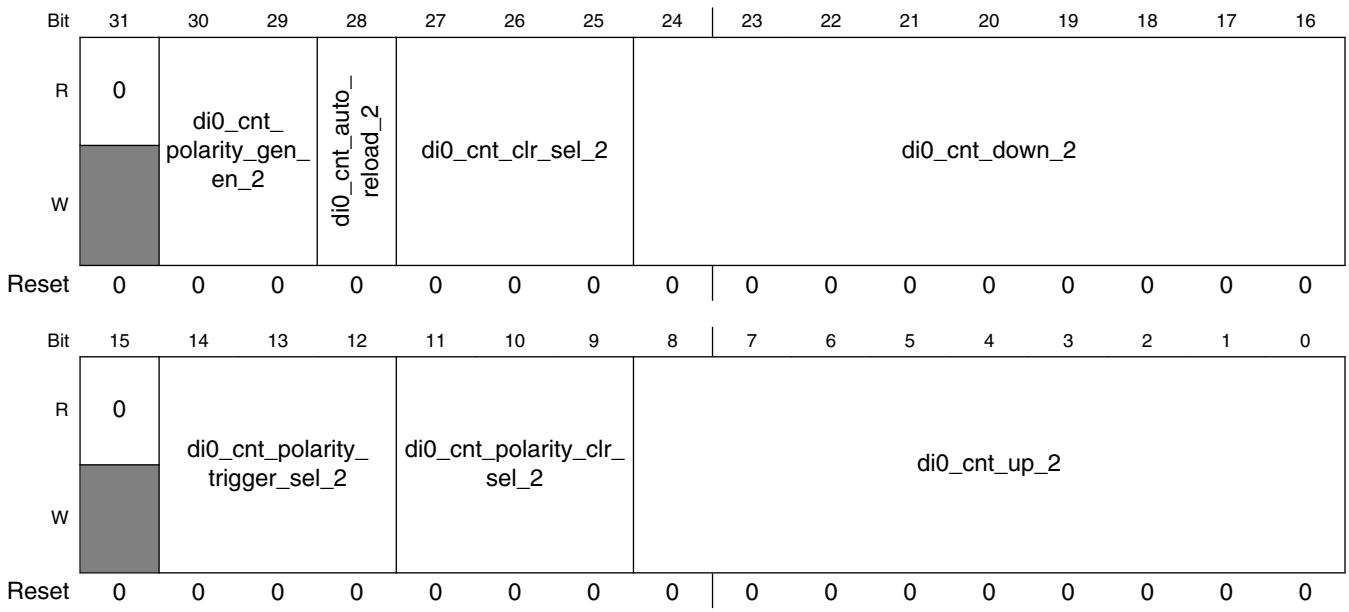
Table continues on the next page...

IPU_DI0_SW_GEN1_1 field descriptions (continued)

Field	Description
	100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
24–16 di0_cnt_down_1	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di0_cnt_polarity_trigger_sel_1	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 Reserved 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_1	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Reserved 011 Reserved 100 Reserved 101 Reserved 110 Reserved 111 Reserved
di0_cnt_up_1	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.206 DI0 Sync Wave Gen 2 Register 1 (IPU_DI0_SW_GEN1_2)

Address: 260_0000h base + 4_0034h offset = 264_0034h



IPU_DI0_SW_GEN1_2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di0_cnt_polarity_gen_en_2	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_reload_2	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_2	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 Reserved

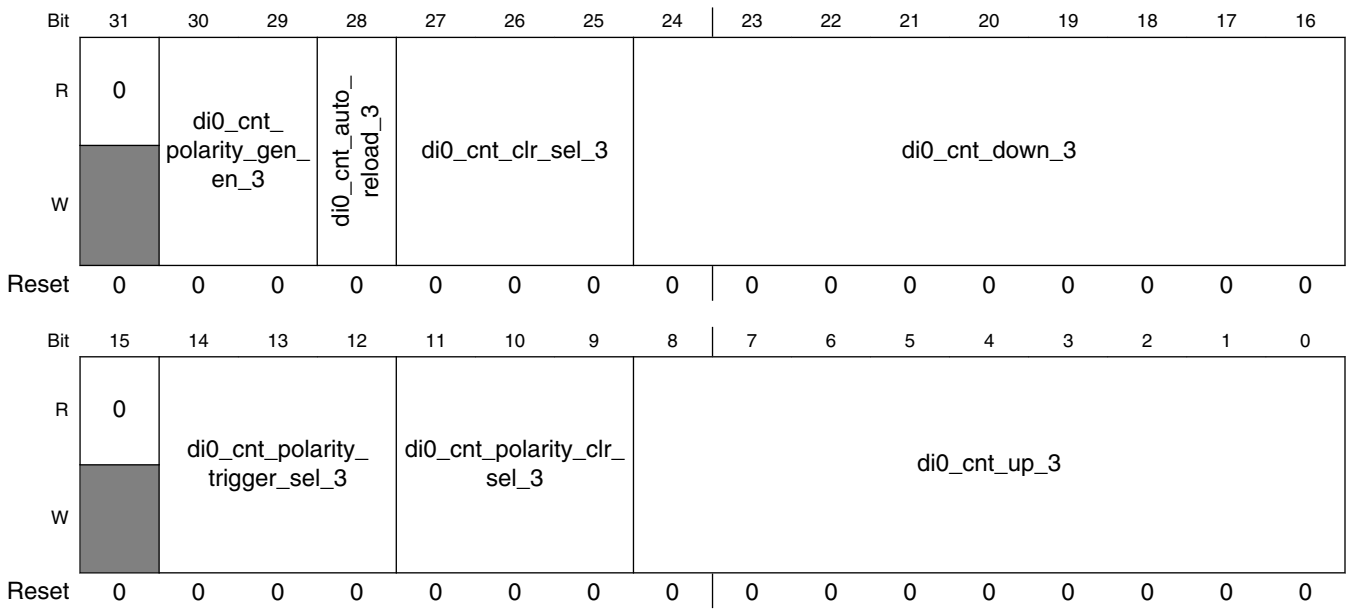
Table continues on the next page...

IPU_DI0_SW_GEN1_2 field descriptions (continued)

Field	Description
	100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
24–16 di0_cnt_down_2	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di0_cnt_polarity_trigger_sel_2	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_2	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Reserved 011 Reserved 100 Reserved 101 Reserved 110 Reserved 111 Reserved
di0_cnt_up_2	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.207 DI0 Sync Wave Gen 3 Register 1 (IPU_DI0_SW_GEN1_3)

Address: 260_0000h base + 4_0038h offset = 264_0038h



IPU_DI0_SW_GEN1_3 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di0_cnt_polarity_gen_en_3	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_reload_3	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_3	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

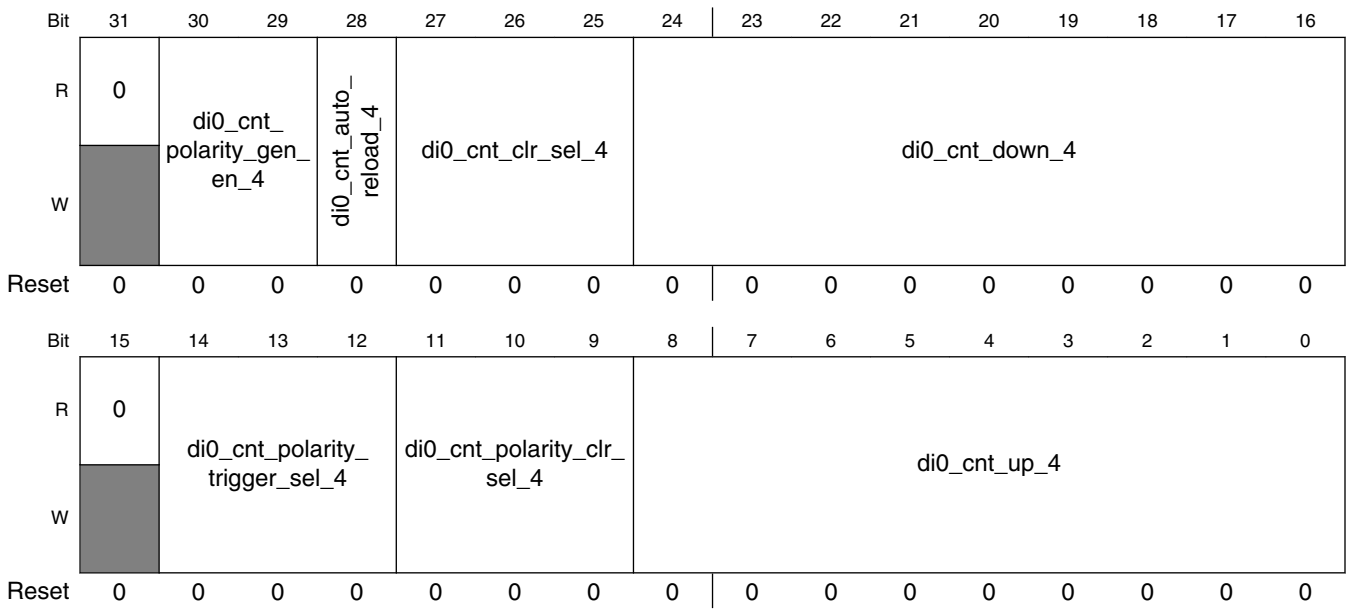
Table continues on the next page...

IPU_DI0_SW_GEN1_3 field descriptions (continued)

Field	Description
	100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
24–16 di0_cnt_down_3	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di0_cnt_polarity_trigger_sel_3	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_3	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Reserved 101 Reserved 110 Reserved 111 Reserved
di0_cnt_up_3	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.208 DI0 Sync Wave Gen 4 Register 1 (IPU_DI0_SW_GEN1_4)

Address: 260_0000h base + 4_003Ch offset = 264_003Ch



IPU_DI0_SW_GEN1_4 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di0_cnt_polarity_gen_en_4	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_reload_4	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_4	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

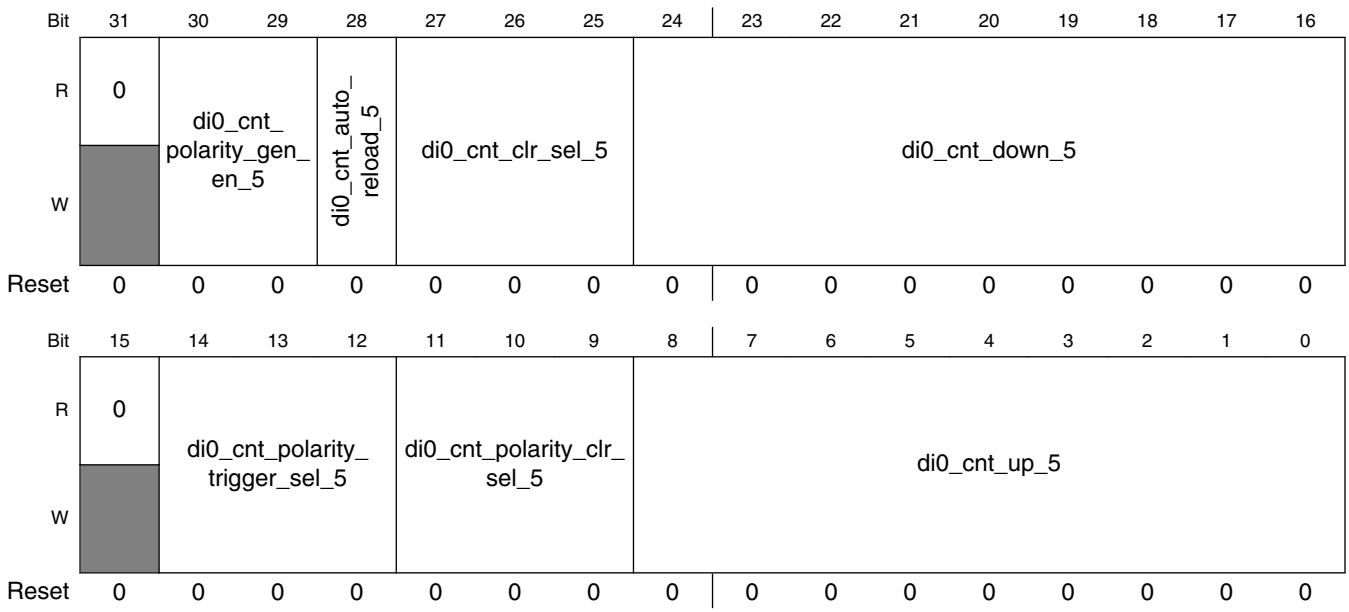
Table continues on the next page...

IPU_DI0_SW_GEN1_4 field descriptions (continued)

Field	Description
	100 The Counter is triggered by counter #3 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
24–16 di0_cnt_down_4	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di0_cnt_polarity_trigger_sel_4	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_4	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Reserved 110 Reserved 111 Reserved
di0_cnt_up_4	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.209 DI0 Sync Wave Gen 5 Register 1 (IPU_DI0_SW_GEN1_5)

Address: 260_0000h base + 4_0040h offset = 264_0040h



IPU_DI0_SW_GEN1_5 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di0_cnt_polarity_gen_en_5	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_reload_5	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_5	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

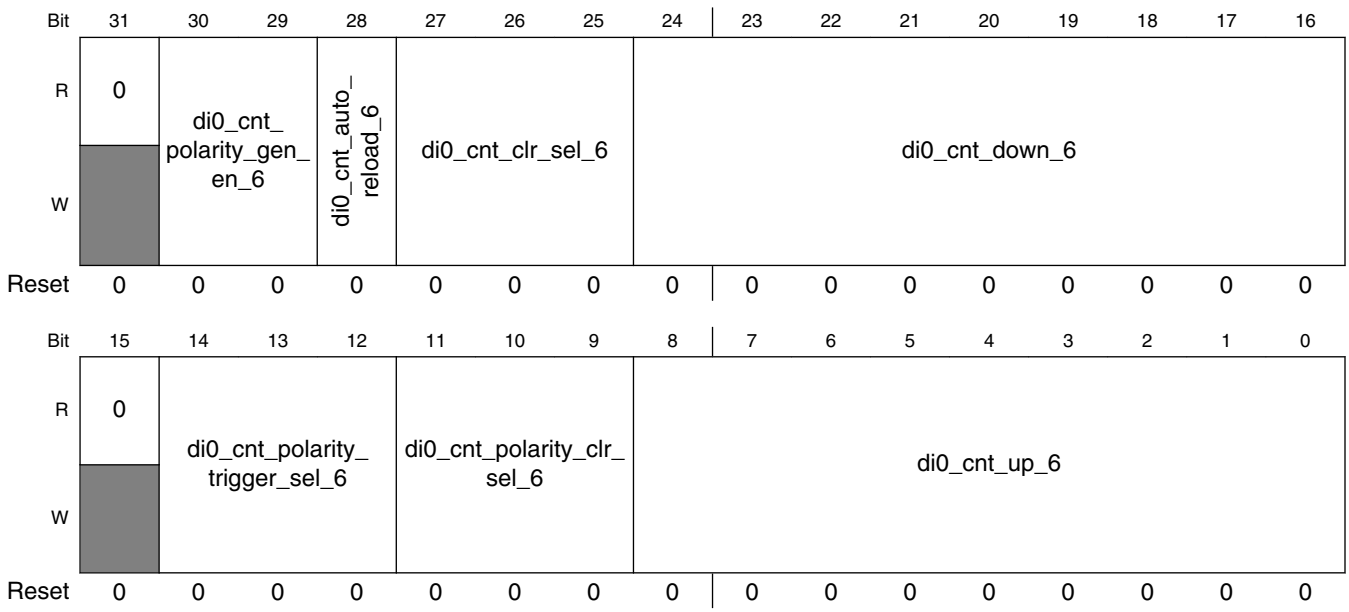
Table continues on the next page...

IPU_DI0_SW_GEN1_5 field descriptions (continued)

Field	Description
	100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 External VSYNC 111 Counter is always on.
24–16 di0_cnt_down_5	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di0_cnt_polarity_trigger_sel_5	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 External VSYNC 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_5	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Reserved 111 Reserved
di0_cnt_up_5	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.210 DI0 Sync Wave Gen 6 Register 1 (IPU_DI0_SW_GEN1_6)

Address: 260_0000h base + 4_0044h offset = 264_0044h



IPU_DI0_SW_GEN1_6 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di0_cnt_polarity_gen_en_6	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_reload_6	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_6	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

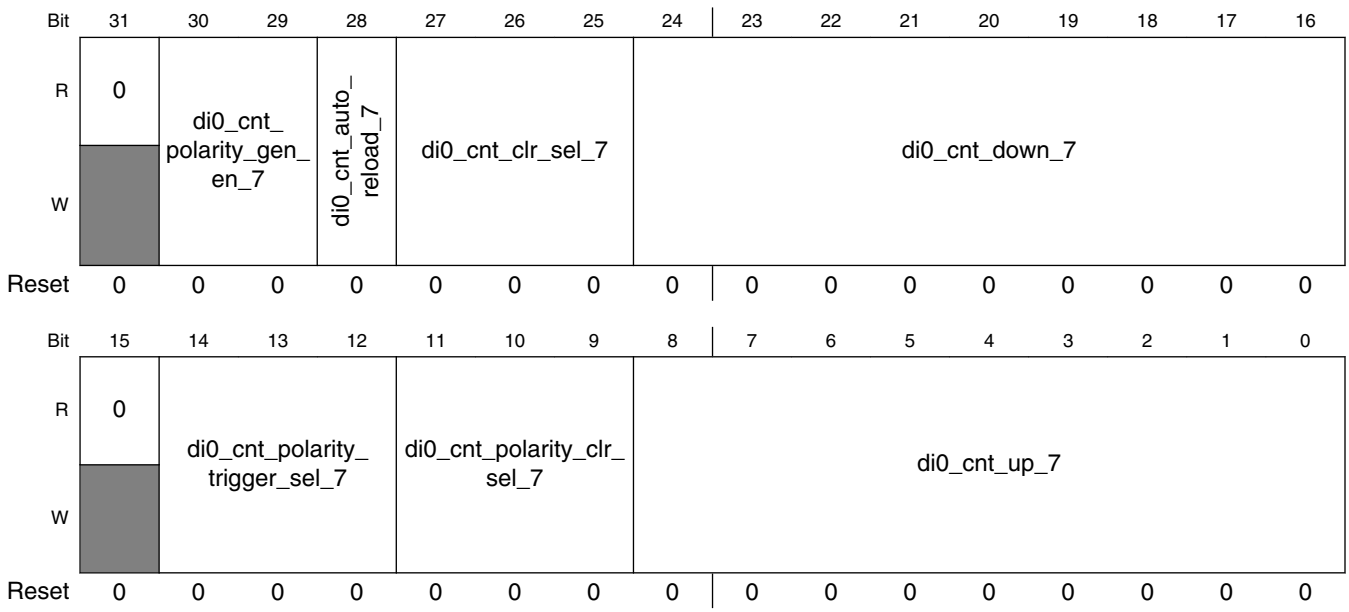
Table continues on the next page...

IPU_DI0_SW_GEN1_6 field descriptions (continued)

Field	Description
	100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
24–16 di0_cnt_down_6	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di0_cnt_polarity_trigger_sel_6	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_6	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Output is inverted if the output of counter #5 is set 111 Reserved
di0_cnt_up_6	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.211 DI0 Sync Wave Gen 7 Register 1 (IPU_DI0_SW_GEN1_7)

Address: 260_0000h base + 4_0048h offset = 264_0048h



IPU_DI0_SW_GEN1_7 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di0_cnt_polarity_gen_en_7	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_reload_7	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_7	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

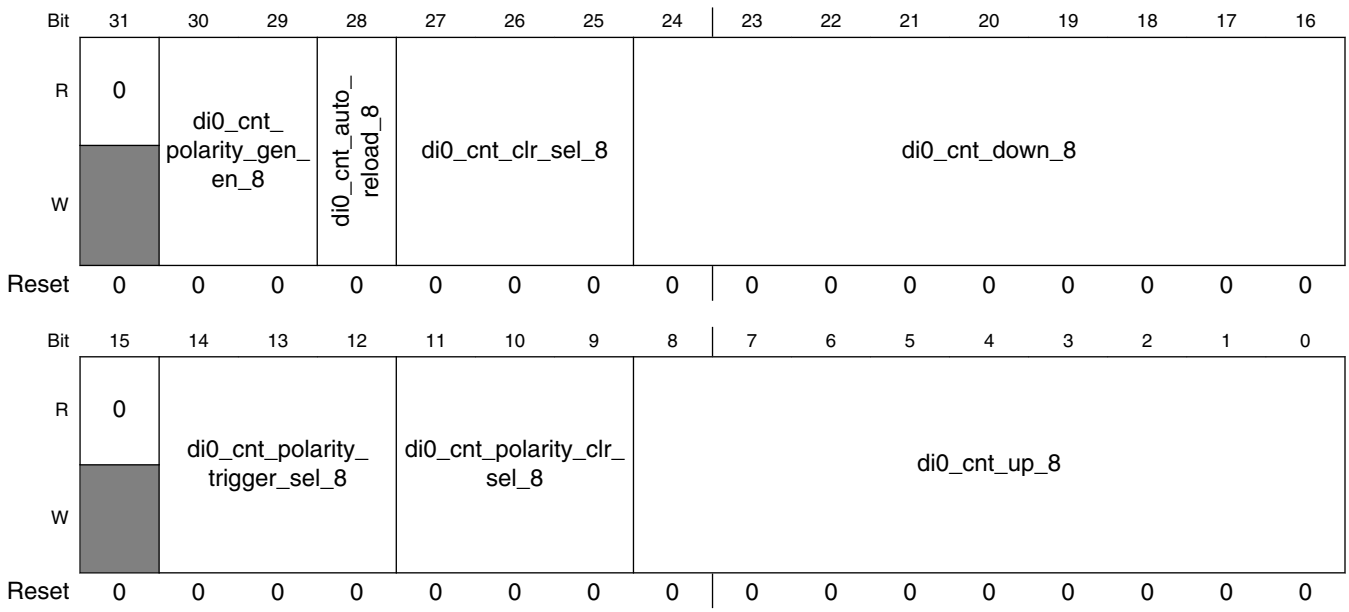
Table continues on the next page...

IPU_DI0_SW_GEN1_7 field descriptions (continued)

Field	Description
	100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
24–16 di0_cnt_down_7	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di0_cnt_polarity_trigger_sel_7	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_7	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Output is inverted if the output of counter #5 is set 111 Output is inverted if the output of counter #6 is set
di0_cnt_up_7	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.212 DI0 Sync Wave Gen 8 Register 1 (IPU_DI0_SW_GEN1_8)

Address: 260_0000h base + 4_004Ch offset = 264_004Ch



IPU_DI0_SW_GEN1_8 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di0_cnt_polarity_gen_en_8	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_reload_8	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_8	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

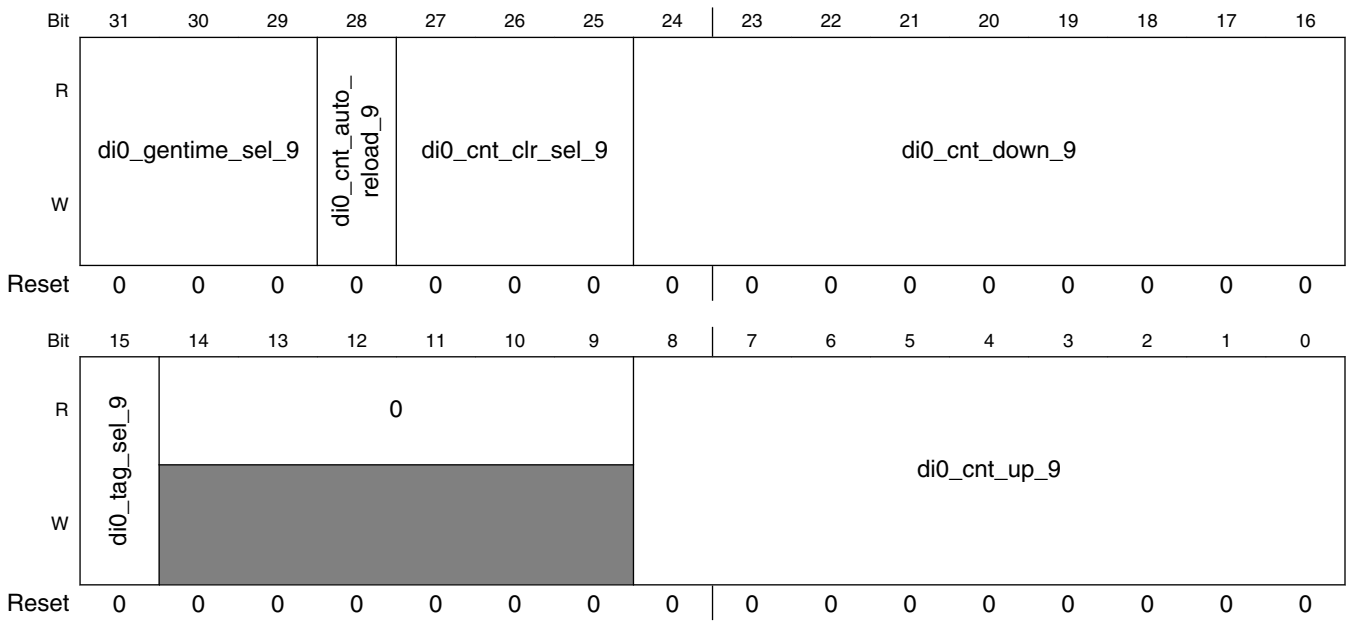
Table continues on the next page...

IPU_DI0_SW_GEN1_8 field descriptions (continued)

Field	Description
	100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
24–16 di0_cnt_down_8	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di0_cnt_polarity_trigger_sel_8	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_8	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Output is inverted if the output of counter #5 is set 111 Output is inverted if the output of counter #6 is set
di0_cnt_up_8	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.213 DI0 Sync Wave Gen 9 Register 1 (IPU_DI0_SW_GEN1_9)

Address: 260_0000h base + 4_0050h offset = 264_0050h



IPU_DI0_SW_GEN1_9 field descriptions

Field	Description
31–29 di0_gentime_sel_9	Counter #9 main waveform select This field defines the counter that counter #9's auxiliary waveform will be attached too. 000 Counter #9's waveform is attached to counter #1's waveform 001 Counter #9's waveform is attached to counter #2's waveform 010 Counter #9's waveform is attached to counter #3's waveform 011 Counter #9's waveform is attached to counter #4's waveform 100 Counter #9's waveform is attached to counter #5's waveform 101 Counter #9's waveform is attached to counter #6's waveform 110 Counter #9's waveform is attached to counter #7's waveform 111 Counter #9's waveform is attached to counter #8's waveform
28 di0_cnt_auto_reload_9	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_9	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

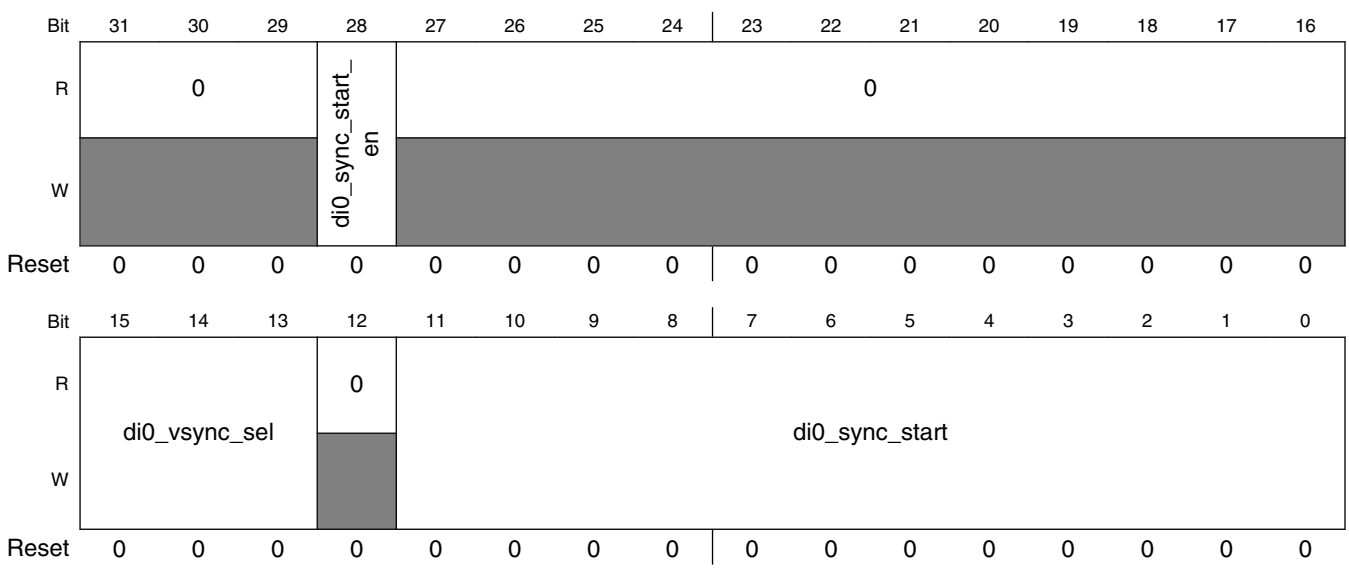
Table continues on the next page...

IPU_DI0_SW_GEN1_9 field descriptions (continued)

Field	Description
	100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
24–16 di0_cnt_down_9	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 di0_tag_sel_9	Counter #9 can send a synchronous tag when counter #9 reach its predefined value or when it's triggering counter reaches its pre defined value. 1 tag source is counter #9 0 Tag's source is the triggering counter.
14–9 Reserved	This read-only field is reserved and always has the value 0.
di0_cnt_up_9	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.214 DI0 Sync Assistance Gen Register (IPU_DI0_SYNC_AS_GEN)

Address: 260_0000h base + 4_0054h offset = 264_0054h



IPU_DI0_SYNC_AS_GEN field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28 di0_sync_start_en	di0_sync_start_en
27–16 Reserved	This read-only field is reserved and always has the value 0.
15–13 di0_vsync_sel	VSYNC select This field defines which of the counters functions as VSYNC signal 000 VSYNC is coming from counter #1 001 VSYNC is coming from counter #2 111 VSYNC is coming from counter #8
12 Reserved	This read-only field is reserved and always has the value 0.
di0_sync_start	DI0 Sync start This field defines the number of low (including blanking rows) on the which the DI0 starts preparing the data for the next frame.

38.5.215 DI0 Data Wave Gen <i> Register (IPU_DI0_DW_GEN_i)

Address: 260_0000h base + 4_0058h offset = 264_0058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	di0_access_size_i								di0_componnent_size_i								di0_cst_i		di0_pt_6_i		di0_pt_5_i		di0_pt_4_i		di0_pt_3_i		di0_pt_2_i		di0_pt_1_i		di0_pt_0_i	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_DW_GEN_i field descriptions

Field	Description
31–24 di0_access_size_i	DI0 Access Size <i> This field defines the amount of IPU cycles between any 2 accesses (an access may be a pixel or generic data that may have more one component)
23–16 di0_componnent_size_i	DI0 component Size This field defines the amount of IPU cycles between any 2 components
15–14 di0_cst_i	DI0 Chip Select pointer for waveform <i> This field points to a register that defines the waveform of the CS pin. The CS is automatically mapped to a specific display 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i>

Table continues on the next page...

IPU_DI0_DW_GEN_i field descriptions (continued)

Field	Description
	10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>
13–12 di0_pt_6_i	DI0 PIN_17 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_17 pin. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>
11–10 di0_pt_5_i	DI0 PIN_16 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_16 pin. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>
9–8 di0_pt_4_i	DI0 PIN_15 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_15 pin. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>
7–6 di0_pt_3_i	DI0 PIN_14 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_14 pin. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>
5–4 di0_pt_2_i	DI0 PIN_13 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_13 pin. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>
3–2 di0_pt_1_i	DI0 PIN_12 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_12 pin. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>
di0_pt_0_i	DI0 PIN_11 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_11 pin.

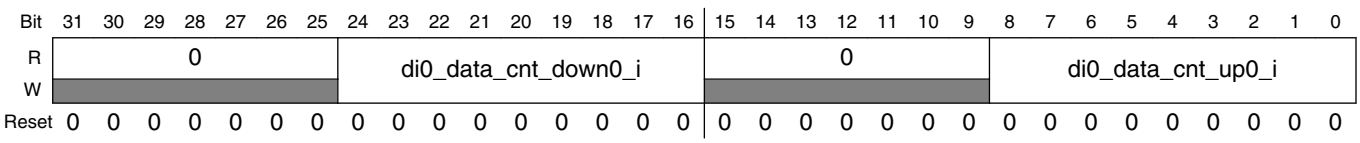
Table continues on the next page...

IPU_DI0_DW_GEN_i field descriptions (continued)

Field	Description
	00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>

38.5.216 DI0 Data Wave Set 0 <i> Register (IPU_DI0_DW_SET0_i)

Address: 260_0000h base + 4_0088h offset = 264_0088h

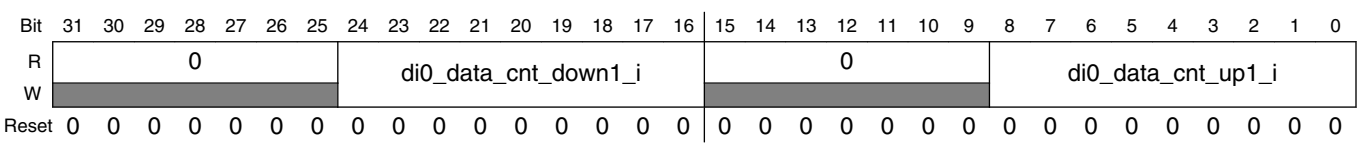


IPU_DI0_DW_SET0_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 di0_data_cnt_down0_i	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a point according to the corresponding di0_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value 0.
di0_data_cnt_up0_i	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a point according to the corresponding di0_pt_*_<i>

38.5.217 DI0 Data Wave Set 1 <i> Register (IPU_DI0_DW_SET1_i)

Address: 260_0000h base + 4_00B8h offset = 264_00B8h



IPU_DI0_DW_SET1_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DI0_DW_SET1_i field descriptions (continued)

Field	Description
24–16 di0_data_cnt_down1_i	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a pint according to the corresponding di0_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value 0.
di0_data_cnt_up1_i	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a pint according to the corresponding di0_pt_*_<i>

38.5.218 DI0 Data Wave Set 2 <i> Register (IPU_DI0_DW_SET2_i)

Address: 260_0000h base + 4_00E8h offset = 264_00E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							di0_data_cnt_down2_i									0							di0_data_cnt_up2_i								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_DW_SET2_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 di0_data_cnt_down2_i	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a pint according to the corresponding di0_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value 0.
di0_data_cnt_up2_i	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a pint according to the corresponding di0_pt_*_<i>

38.5.219 DI0 Data Wave Set 3 <i> Register (IPU_DI0_DW_SET3_i)

Address: 260_0000h base + 4_0118h offset = 264_0118h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							di0_data_cnt_down3_i									0							di0_data_cnt_up3_i								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_DW_SET3_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 di0_data_cnt_down3_i	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a pint according to the corresponding di0_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value 0.
di0_data_cnt_up3_i	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a pint according to the corresponding di0_pt_*_<i>

38.5.220 DI0 Step Repeat <i> Registers (IPU_DI0_STP_REP_i)

Address: 260_0000h base + 4_0148h offset = 264_0148h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				di0_step_repeat_2i												0				di0_step_repeat_2i_minus_1											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_STP_REP_i field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–16 di0_step_repeat_2i	Step Repeat <i> This fields defines the amount of repetitions that will be performed by the counter <i>
15–12 Reserved	This read-only field is reserved and always has the value 0.
di0_step_repeat_2i_minus_1	Step Repeat <i> This fields defines the amount of repetitions that will be performed by the counter <i>

38.5.221 DI0 Step Repeat 9 Registers (IPU_DI0_STP_REP_9)

Address: 260_0000h base + 4_0158h offset = 264_0158h

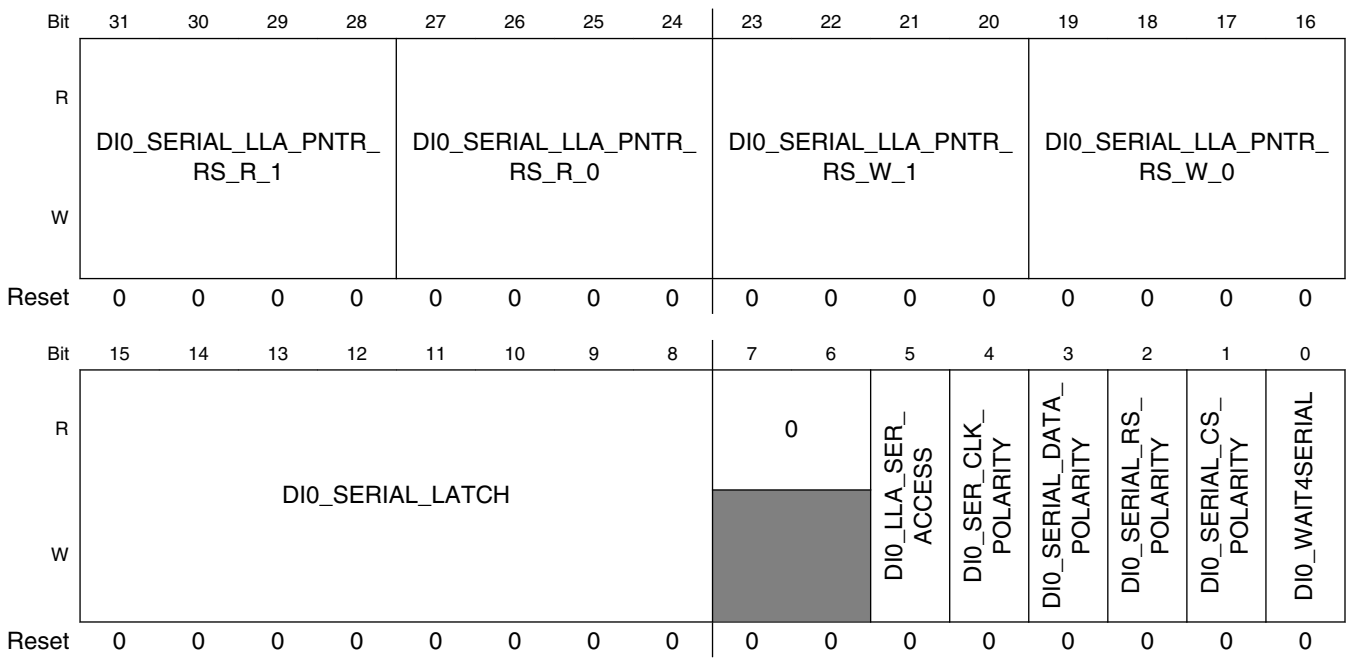
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																				di0_step_repeat_9											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_STP_REP_9 field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value 0.
di0_step_repeat_9	Step Repeat 9 This fields defines the amount of repetitions that will be performed by the counter 9

38.5.222 DI0 Serial Display Control Register (IPU_DI0_SER_CONF)

Address: 260_0000h base + 4_015Ch offset = 264_015Ch



IPU_DI0_SER_CONF field descriptions

Field	Description
31–28 DI0_SERIAL_LLA_PNTR_RS_R_1	RS 3 waveform pointer for read low level access This pointer defines which waveform set will be chosen when the read low level access is targeted to RS group 1. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved

Table continues on the next page...

IPU_DI0_SER_CONF field descriptions (continued)

Field	Description
27–24 DI0_SERIAL_LLA_PNTR_RS_R_0	RS 2 waveform pointer for read low level access This pointer defines which waveform set will be chosen when the read low level access is targeted to RS group 0. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved
23–20 DI0_SERIAL_LLA_PNTR_RS_W_1	RS 1 waveform pointer for write low level access This pointer defines which waveform set will be chosen when the low level write access is targeted to RS group 1. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved
19–16 DI0_SERIAL_LLA_PNTR_RS_W_0	RS 0 waveform pointer for write low level access This pointer defines which waveform set will be chosen when the low level write access is targeted to RS group 0. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved
15–8 DI0_SERIAL_LATCH	DI0 Serial Latch This field defines how many cycles to insert between serial read accesses start to data sampling in the
7–6 Reserved	This read-only field is reserved and always has the value 0.
5 DI0_LLA_SER_ACCESS	Direct Low Level Access to Serial display 1 ARM platform access is performed via a direct path to the serial display in LLA mode, in this mode only the ARM platform in LLA mode can access the serial port 0 ARM platform access to the serial display port is not done directly, hence other source are allowed to access the serial port. The arbitration is done automatically
4 DI0_SER_CLK_POLARITY	Serial Clock Polarity The output polarity of the SER_CLK pin 1 The clock is inverted 0 The clock is not inverted

Table continues on the next page...

IPU_DI0_SER_CONF field descriptions (continued)

Field	Description
3 DI0_SERIAL_ DATA_ POLARITY	Serial Data Polarity The output polarity of the SER_DATA pin 1 The data is inverted 0 The data is not inverted
2 DI0_SERIAL_ RS_POLARITY	Serial RS Polarity The output polarity of the SER_RS pin 1 The RS is inverted 0 The RS is not inverted
1 DI0_SERIAL_ CS_POLARITY	Serial Chip Select Polarity The output polarity of the SER_CS pin 1 The CS is inverted 0 The CS is not inverted
0 DI0_ WAIT4SERIAL	Wait for Serial When the parallel display share pins with the serial port. Accessing the parallel port is not allowed till the serial port completes its access. 1 The parallel port should wait to the serial port as the pins are shared 0 The parallel port should not wait to the serial port as the pins are not shared

38.5.223 DI0 Special Signals Control Register (IPU_DI0_SSC)

Address: 260_0000h base + 4_0160h offset = 264_0160h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								DI0_PIN17_ERM	DI0_PIN16_ERM	DI0_PIN15_ERM	DI0_PIN14_ERM	DI0_PIN13_ERM	DI0_PIN12_ERM	DI0_PIN11_ERM	DI0_CS_ERM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										DI0_WAIT_ON	0	DI0_BYTE_EN_RD_IN	DI0_BYTE_EN_PNTR		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SSC field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value 0.
23 DI0_PIN17_ERM	DI0 PIN17 error recovery mode. This bit defines the error recovery mode of the PIN17 pin. 1 The PIN17 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC 0 Nothing is done to the PIN17 pin following a display error detection
22 DI0_PIN16_ERM	DI0 PIN16 error recovery mode. This bit defines the error recovery mode of the PIN16 pin. 1 The PIN16 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC 0 Nothing is done to the PIN16 pin following a display error detection
21 DI0_PIN15_ERM	DI0 PIN15 error recovery mode. This bit defines the error recovery mode of the PIN15 pin. 1 The PIN15 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC 0 Nothing is done to the PIN15 pin following a display error detection
20 DI0_PIN14_ERM	DI0 PIN14 error recovery mode. This bit defines the error recovery mode of the PIN14 pin. 1 The PIN14 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC 0 Nothing is done to the PIN14 pin following a display error detection
19 DI0_PIN13_ERM	DI0 PIN13 error recovery mode. This bit defines the error recovery mode of the PIN13 pin. 1 The PIN13 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC 0 Nothing is done to the PIN13 pin following a display error detection
18 DI0_PIN12_ERM	DI0 PIN12 error recovery mode. This bit defines the error recovery mode of the PIN12 pin. 1 The PIN12 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC 0 Nothing is done to the PIN12 pin following a display error detection
17 DI0_PIN11_ERM	DI0 PIN11 error recovery mode. This bit defines the error recovery mode of the PIN11 pin. 1 The PIN11 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC 0 Nothing is done to the PIN11 pin following a display error detection
16 DI0_CS_ERM	DI0 GLUELOGIC error recovery mode. This bit defines the error recovery mode of the GLUELOGIC.

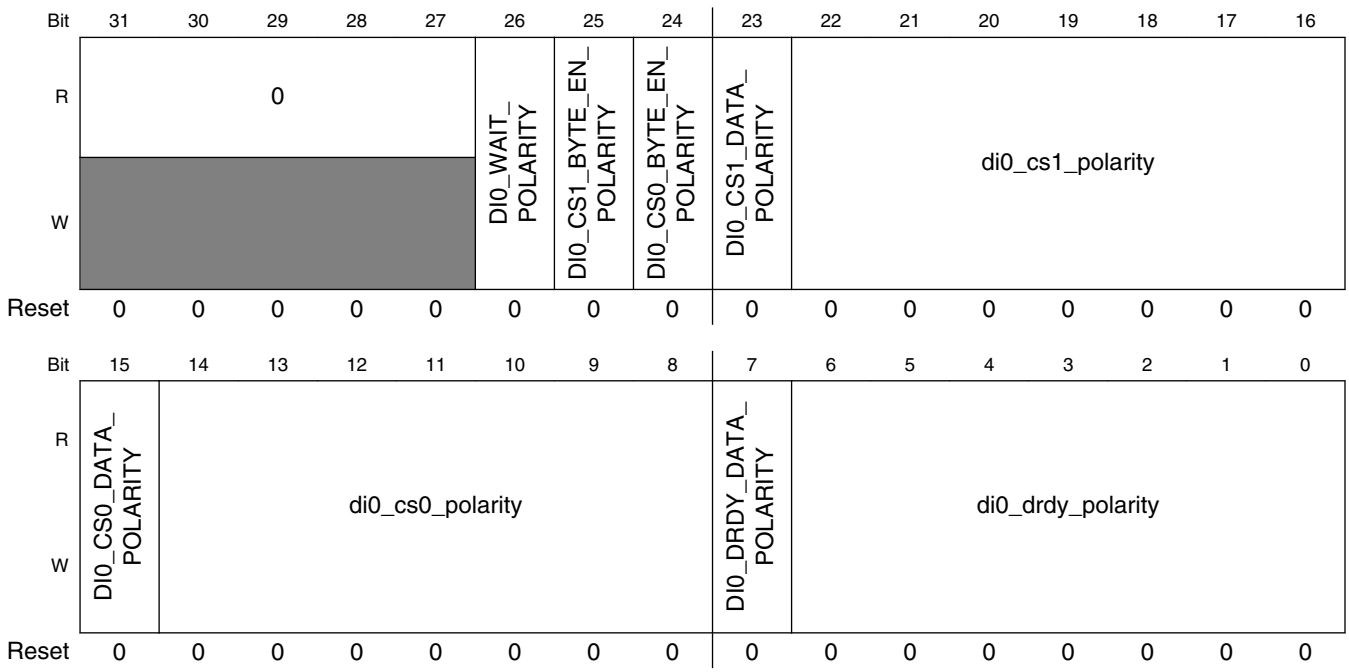
Table continues on the next page...

IPU_DI0_SSC field descriptions (continued)

Field	Description
	1 The GLUELOGIC is release in case of a synchronous display error. The release will be done on the next VSYNC 0 Nothing is done to the GLUELOGIC following a display error detection
15–6 Reserved	This read-only field is reserved and always has the value 0.
5 DI0_WAIT_ON	Wait On This field defines the DC's response to WAIT signal 1 The DC holds the flow as long as WAIT is asserted 0 The DC continues the flow regardless the WAIT signal
4 Reserved	This read-only field is reserved and always has the value 0.
3 DI0_BYTE_EN_RD_IN	Byte Enable Read In This bit selects the source of the byte enable pins 1 The write byte enable signals are routed via bits [17:16] of the display's data, The read byte enable signals are routed via bits [19:18 of the display's data 0 The byte enable signals are routed via bits [17:16] of the display's data for both read and write
DI0_BYTE_EN_PNTR	Byte Enable Pointer This pointer selects the pin asserted along with the byte enables signals 000 wave form of byte enable as pin_11 001 wave form of byte enable as pin_12 111 wave form of byte enable as suitable CS pin

38.5.224 DI0 Polarity Register (IPU_DI0_POL)

Address: 260_0000h base + 4_0164h offset = 264_0164h



IPU_DI0_POL field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26 DI0_WAIT_POLARITY	WAIT polarity This bit defines the polarity of the wait signal input coming from the displa1 1 active high 0 active low
25 DI0_CS1_BYTE_EN_POLARITY	Byte Enable associated with CS1 polarity This bit defines the polarity of the byte enable signals to the display 1 active high 0 active low
24 DI0_CS0_BYTE_EN_POLARITY	Byte Enable associated with CS0 polarity This bit defines the polarity of the byte enable signals to the display 1 active high 0 active low
23 DI0_CS1_DATA_POLARITY	Data Polarity associated with CS1

Table continues on the next page...

IPU_DI0_POL field descriptions (continued)

Field	Description
22–16 di0_cs1_polarity	DI0 output pin's polarity for CS1 This bits define the polarity of each of the DI's outputs when CS1 is asserted 1 The output pin is active high 0 The output pin is active low
15 DI0_CS0_DATA_POLARITY	Data Polarity associated with CS0
14–8 di0_cs0_polarity	DI0 output pin's polarity for CS1 This bits define the polarity of each of the DI's outputs when CS1 is asserted 1 The output pin is active high 0 The output pin is active low
7 DI0_DRDY_DATA_POLARITY	Data Polarity associated with DRDY
di0_drdy_polarity	DI0 output dynamic pin's polarity for synchronous access This bits define the polarity of each of the DI's outputs when synchronous display access is asserted The pins' default polarity is the same as defined in the di0_drdy_polarity bits 1 The output pin is active high 0 The output pin is active low

38.5.225 DI0 Active Window 0 Register (IPU_DI0_AW0)

Address: 260_0000h base + 4_0168h offset = 264_0168h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DI0_AW_TRIG_SEL				DI0_AW_HEND												DI0_AW_HCOUNT_SEL				DI0_AW_HSTART											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_AW0 field descriptions

Field	Description
31–28 DI0_AW_TRIG_SEL	This field selects the trigger for sending data during the display's active window 000 disabled 001 The trigger is the same trigger that triggers the displays clock. 010 The trigger is counter #1 011 The trigger is counter #2 100 The trigger is counter #3 101 The trigger is counter #4

Table continues on the next page...

IPU_DI0_AW0 field descriptions (continued)

Field	Description
	110 The trigger is counter #5 111 The trigger is always on.
27–16 DI0_AW_HEND	This field defines the horizontal end of the active window
15–12 DI0_AW_ HCOUNT_SEL	GM: This field selects the counter that counts the horizontal position of the display's active window 0000 disabled 0001 reserved 0010 The counter is counter #1 0011 The counter is counter #2 0100 The counter is counter #3 0101 The counter is counter #4 0110 The counter is counter #5 1001 The counter is counter #8
DI0_AW_ HSTART	This field defines the horizontal start of the active window DI0_AW_HSTART < DI0_AW_HEND

38.5.226 DI0 Active Window 1 Register (IPU_DI0_AW1)

Address: 260_0000h base + 4_016Ch offset = 264_016Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DI0_AW_VEND												DI0_AW_ VCOUNT_ SEL				DI0_AW_VSTART											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_AW1 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–16 DI0_AW_VEND	This field defines the vertical end of the active window
15–12 DI0_AW_ VCOUNT_SEL	This field selects the counter that counts the vertical position of the display's active window 0000 disabled 0001 reserved 0010 The counter is counter #1 0011 The counter is counter #2 0100 The counter is counter #3 0101 The counter is counter #4 0110 The counter is counter #5 1001 The counter is counter #8

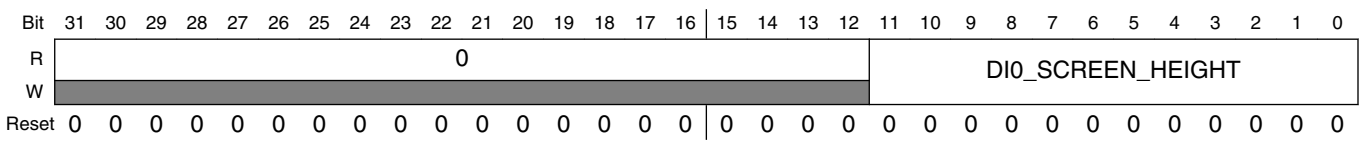
Table continues on the next page...

IPU_DI0_AW1 field descriptions (continued)

Field	Description
DI0_AW_ VSTART	This field defines the vertical start of the active window DI0_AW_VSTART < DI0_AW_VEND

38.5.227 DI0 Screen Configuration Register (IPU_DI0_SCR_CONF)

Address: 260_0000h base + 4_0170h offset = 264_0170h

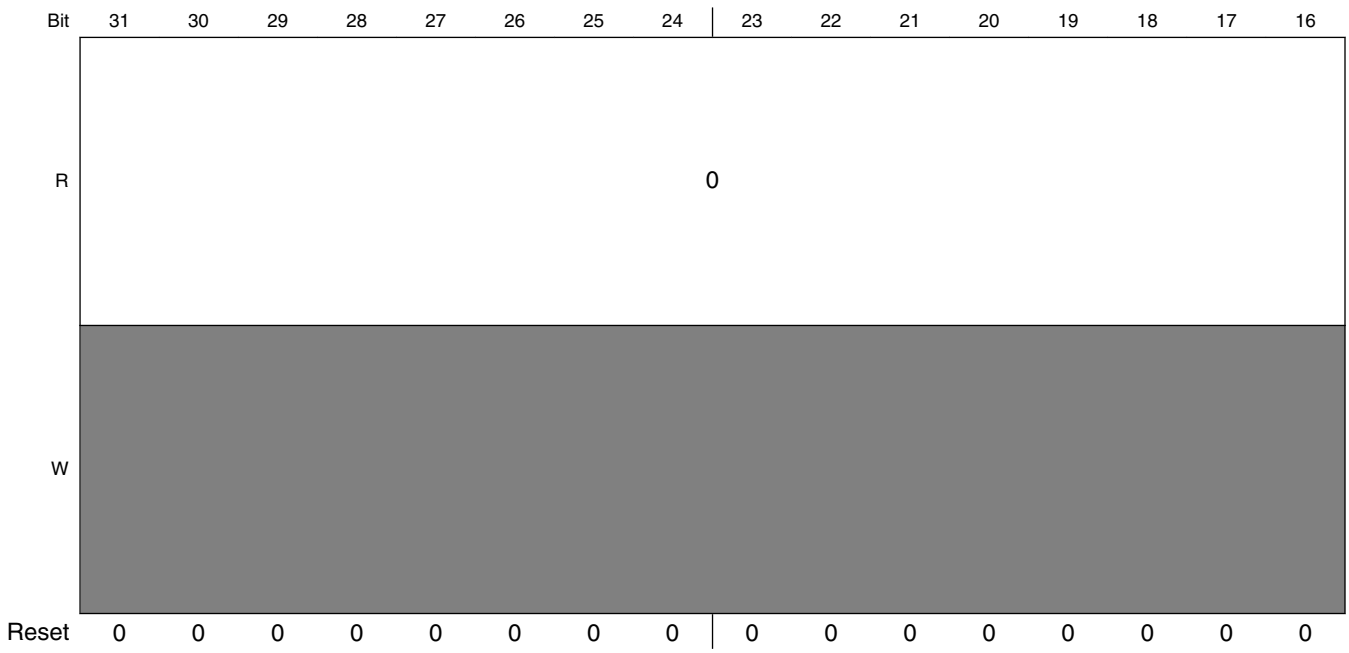


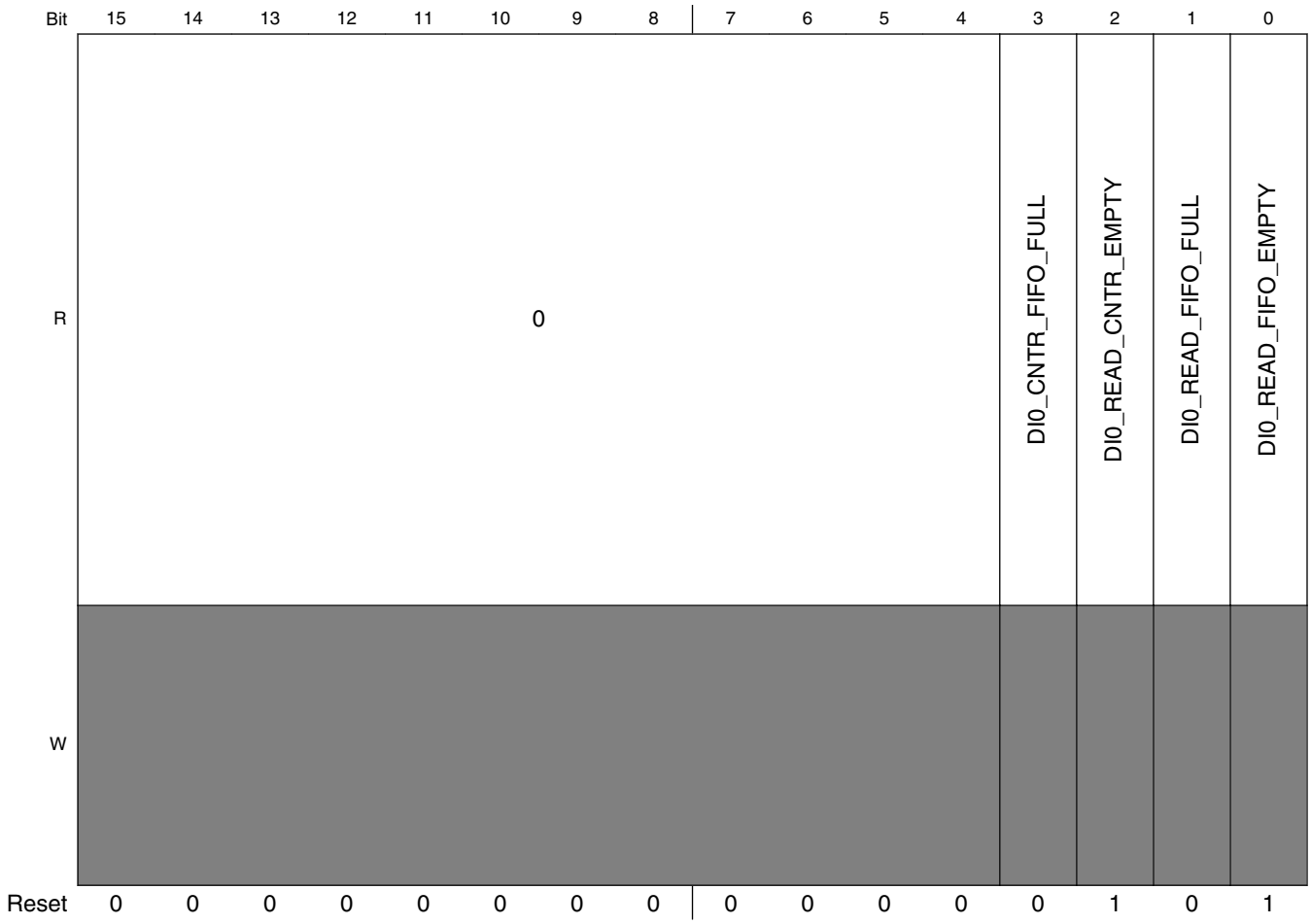
IPU_DI0_SCR_CONF field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value 0.
DI0_SCREEN_ HEIGHT	This field defines the number of display rows (Number_of_ROWS = DI0_SCREEN_HEIGHT+1) This field is used for VSYNC calculation and for anti-tearing

38.5.228 DI0 Status Register (IPU_DI0_STAT)

Address: 260_0000h base + 4_0174h offset = 264_0174h



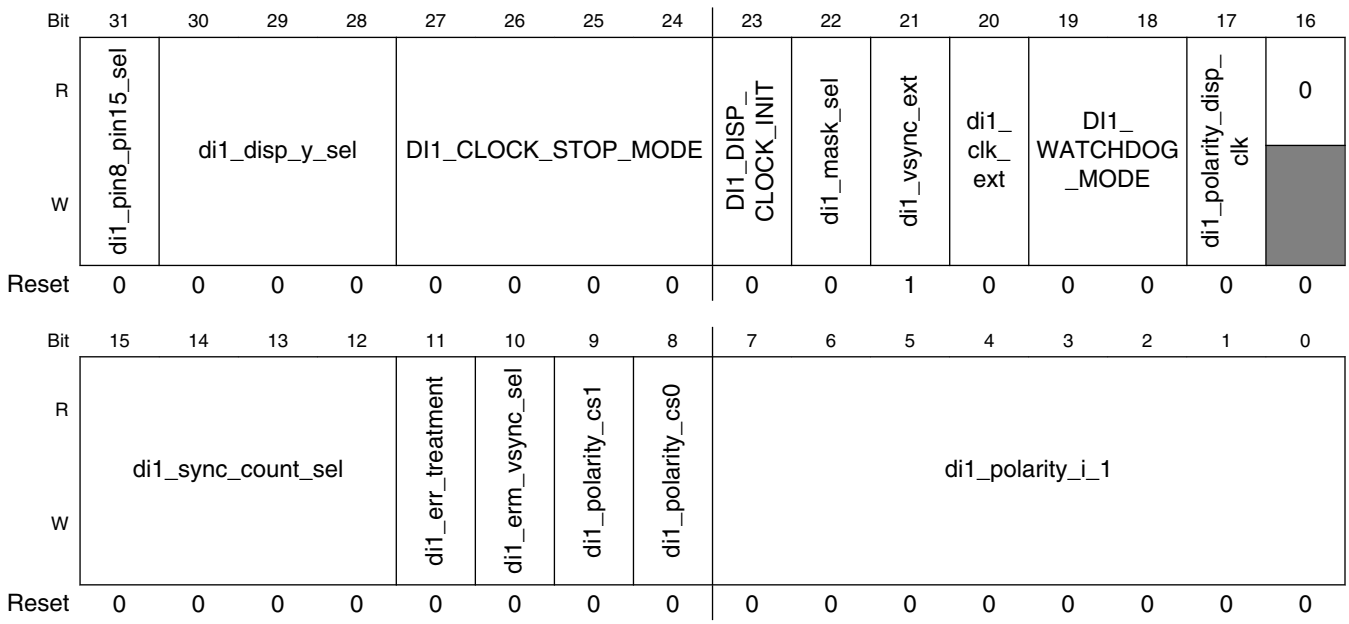


IPU_DI0_STAT field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value 0.
3 DI0_CNTR_ FIFO_FULL	This bit indicates a full state of the DI0 FIFO. This FIFO is part of the DI0 synchronizer.
2 DI0_READ_ CNTR_EMPTY	This bit indicates an empty state of the DI0 FIFO. This FIFO is part of the DI0 synchronizer.
1 DI0_READ_ FIFO_FULL	This bit indicates a full state of the DI0 FIFO when performing a read. This FIFO is part of the DI0 synchronizer.
0 DI0_READ_ FIFO_EMPTY	This bit indicates an empty state of the DI0 FIFO when performing a read. This FIFO is part of the DI0 synchronizer.

38.5.229 DI1General Register (IPU_DI1_GENERAL)

Address: 260_0000h base + 4_8000h offset = 264_8000h



IPU_DI1_GENERAL field descriptions

Field	Description
31 di1_pin8_pin15_sel	This bit routes PIN8 over PIN15 1 PIN8 is routed to PIN15, PIN8 is also routed to PIN8 0 PIN15 is routed to PIN15, PIN8 is routed to PIN8
30–28 di1_disp_y_sel	DI1 Display Vertical coordinate (Y) select. This field defines which one of the 8 counters will be used as a display's line counter. 000 counter #1 is selected 111 counter #8 is selected
27–24 DI1_CLOCK_STOP_MODE	DI clock stop mode When performing a clock change. The DI stops the clock to the display. These field defines when the clock will be stopped Stopping at EOL/EOF is supported for the case where the data is coming from the IDMAC (DMA access). In case that only direct accesses is performed, the user should set this field to 0000 0001-1001 stop at the next event of one of the counters (counter #1 to counter #9) 0000 stop at the next edge of the display clock 1100 stop at EOL (end of a line), but if stop request is during blanking interval, stop now 1101 stop at EOF (end of a frame), but if stop request is during blanking interval, stop now

Table continues on the next page...

IPU_DI1_GENERAL field descriptions (continued)

Field	Description
	1110 stop at EOL (end of a line), but if stop request is during blanking interval, stop at the end of the next line 1111 stop at EOF (end of a frame), but if stop request is during blanking interval, stop at the end of the next frame
23 DI1_DISP_CLOCK_INIT	Display clock's initial mode For synchronization error conditions the display clock can be stopped on the next VSYNC 1 The display's clock is running after the next VSYNC (indicating new frame) 0 The display's clock is stopped after the next VSYNC (indicating new frame)
22 di1_mask_sel	DI1 Mask select. IPP_PIN_2 output of the DI that functions as MASK signal can come from 2 sources: counter #2 or extracted from the MASK data coming from the memory. 1 IPP_PIN_2 is coming from extracted MASK data coming from the memory 0 IPP_PIN_2 is coming from counter #2
21 di1_vsync_ext	DI1 External VSYNC. This bit selects the source of the VSYNC signal 1 External to the IPU 0 Internally generated by the IPU
20 di1_clk_ext	DI1 External Clock. This bit selects the source of the DI's clock 1 The source of the clock is external to the IPU 0 The clock is internally generated by the IPU
19–18 DI1_WATCHDOG_MODE	DI1 watchdog mode In case of a display error where the DI clock is stopped (defined at di0_err_treatment). An internal watchdog counts DI clocks. If this timer reached its pre defined value the DI will skip the current frame and restart on the frame. This 2 bits define the number of DI clock cycles that the timer counts. 00 The timer counts 4 DI cycles 01 The timer counts 16 DI cycles 10 The timer counts 64 DI cycles 11 The timer counts 128 DI cycles
17 di1_polarity_disp_clk	DI1 Output Clock's polarity This bits define the polarity of the DI's clock. 1 The output clock is active high 0 The output clock is active low
16 Reserved	This read-only field is reserved and always has the value 0.
15–12 di1_sync_count_sel	For synchronous flow error: selects synchronous flow synchronization counter in DI:
11 di1_err_treatment	In case of synchronous flow error there are 2 ways to handle the display

Table continues on the next page...

IPU_DI1_GENERAL field descriptions (continued)

Field	Description
	1 to wait (i.e. stop clock) 0 Drive the last component
10 di1_erm_vsync_sel	DI1 error recovery module's VSYNC source select The error recovery block detect a case where the DI's VSYNC is asserted before the EOF. This bit selects the source of the VSYNC signal monitored by this mechanism. 1 vsync_post - an internal VSYNC signal asserted 2 lines after the DI's VSYNC 0 vsync_pre - an internal VSYNC signal asserted 2 lines before the DI's VSYNC
9 di1_polarity_cs1	DI1 Chip Select's 1 polarity This bits define the polarity of the DI's CS1. 1 The CS1 is active high 0 The CS1 is active low
8 di1_polarity_cs0	DI1 Chip Select's 0 polarity This bits define the polarity of the DI's CS0. 1 The CS0 is active high 0 The CS0 is active low
di1_polarity_i_1	DI1 output pin's polarity This bits define the polarity of each of the DI's outputs. 1 The output pin is active high 0 The output pin is active low

38.5.230 DI1 Base Sync Clock Gen 0 Register (IPU_DI1_BS_CLKGEN0)

Address: 260_0000h base + 4_8004h offset = 264_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							di1_disp_clk_offset									0				di1_disp_clk_period											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_BS_CLKGEN0 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 di1_disp_clk_offset	DI1 Display Clock Offset The DI has the ability to delay the display's clock This field defines the amount of IPU's clock cycles added as delay on this clock.
15–12 Reserved	This read-only field is reserved and always has the value 0.

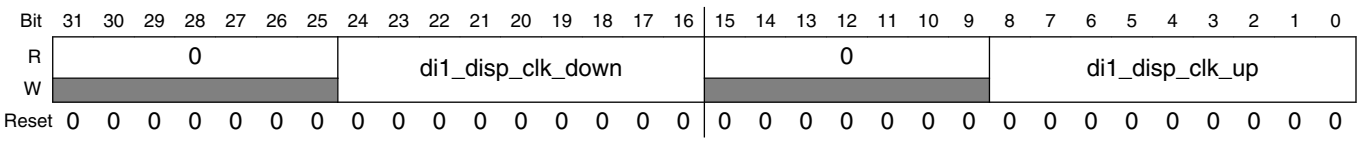
Table continues on the next page...

IPU_DI1_BS_CLKGEN0 field descriptions (continued)

Field	Description
di1_disp_clk_ period	DI1 Display Clock Period This field defines the Display interface clock period for display write access. This parameter contains an integer part (bits 11:4) and a fractional part (bits 3:0). It defines a fractional division ratio of the HSP_CLK clock for generation of the display's interface clock.

38.5.231 DI1 Base Sync Clock Gen 1 Register (IPU_DI1_BS_CLKGEN1)

Address: 260_0000h base + 4_8008h offset = 264_8008h

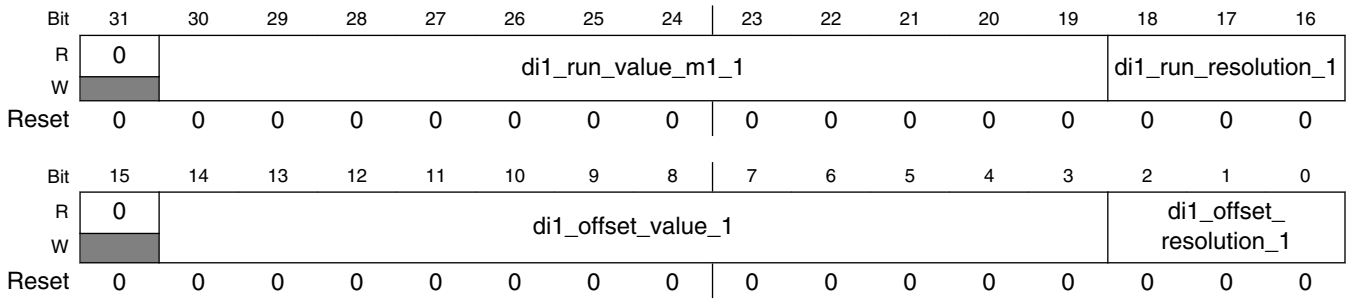


IPU_DI1_BS_CLKGEN1 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 di1_disp_clk_ down	DI1 display clock falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is a time interval between display's access start point and display 's interface clock falling edge.
15–9 Reserved	This read-only field is reserved and always has the value 0.
di1_disp_clk_up	DI1 display clock rising edge position This parameter contains an integer part (bits 8:1) and a fractional part (bit 0). The position value is a time interval between display's access start point and display 's interface clock rising edge.

38.5.232 DI1 Sync Wave Gen 1 Register 0 (IPU_DI1_SW_GEN0_1)

Address: 260_0000h base + 4_800Ch offset = 264_800Ch



IPU_DI1_SW_GEN0_1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di1_run_value_m1_1	DI1 counter #1 pre defined value This fields defines the counter #1 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_1 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_resolution_1	DI1 counter #1 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 NA 011 NA 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.
14–3 di1_offset_value_1	DI1 counter #1 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di1_offset_resolution_1	DI1 counter #1 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 NA 011 NA

Table continues on the next page...

IPU_DI1_SW_GEN0_1 field descriptions (continued)

Field	Description
	100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

38.5.233 DI1 Sync Wave Gen 2 Register 0 (IPU_DI1_SW_GEN0_2)

Address: 260_0000h base + 4_8010h offset = 264_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_run_value_m1_2												di1_run_resolution_2		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_offset_value_2												di1_offset_ resolution_2		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN0_2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di1_run_value_ m1_2	DI1 counter #2 pre defined value This fields defines the counter #2 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_2 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_ resolution_2	DI1 counter #2 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 NA 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DI1_SW_GEN0_2 field descriptions (continued)

Field	Description
14–3 di1_offset_value_2	DI1 counter #2 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di1_offset_resolution_2	DI1 counter #2 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 NA 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

38.5.234 DI1 Sync Wave Gen 3 Register 0 (IPU_DI1_SW_GEN0_3)

Address: 260_0000h base + 4_8014h offset = 264_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_run_value_m1_3												di1_run_resolution_3		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_offset_value_3												di1_offset_resolution_3		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN0_3 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di1_run_value_m1_3	DI1 counter #3 pre defined value This fields defines the counter #3 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_3 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_resolution_3	DI1 counter #3 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock.

Table continues on the next page...

IPU_DI1_SW_GEN0_3 field descriptions (continued)

Field	Description
	010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.
14–3 di1_offset_value_3	DI1 counter #3 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di1_offset_resolution_3	DI1 counter #3 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

38.5.235 DI1 Sync Wave Gen 4 Register 0 (IPU_DI1_SW_GEN0_4)

Address: 260_0000h base + 4_8018h offset = 264_8018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_run_value_m1_4												di1_run_resolution_4		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_offset_value_4												di1_offset_resolution_4		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN0_4 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di1_run_value_m1_4	DI1 counter #4 pre defined value This fields defines the counter #4 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_4 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_resolution_4	DI1 counter #4 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.
14–3 di1_offset_value_4	DI1 counter #4 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di1_offset_resolution_4	DI1 counter #4 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

38.5.236 DI1 Sync Wave Gen 5 Register 0 (IPU_DI1_SW_GEN0_5)

Address: 260_0000h base + 4_801Ch offset = 264_801Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_run_value_m1_5												di1_run_resolution_5		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_offset_value_5												di1_offset_ resolution_5		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN0_5 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di1_run_value_ m1_5	DI1 counter #5 pre defined value This fields defines the counter #5 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_5 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_ resolution_5	DI1 counter #5 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.
14–3 di1_offset_value_ 5	DI1 counter #5 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di1_offset_ resolution_5	DI1 counter #5 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 -The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4

Table continues on the next page...

IPU_DI1_SW_GEN0_5 field descriptions (continued)

Field	Description
	110 External VSYNC 111 Counter is always on.

38.5.237 DI1 Sync Wave Gen 6 Register 0 (IPU_DI1_SW_GEN0_6)

Address: 260_0000h base + 4_8020h offset = 264_8020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_run_value_m1_6												di1_run_resolution_6		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_offset_value_6												di1_offset_ resolution_6		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN0_6 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di1_run_value_ m1_6	DI1 counter #6 pre defined value This fields defines the counter #6 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_6 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_ resolution_6	DI1 counter #6 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.
14–3 di1_offset_value_ 6	DI1 counter #6 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di1_offset_ resolution_6	DI1 counter #6 offset Resolution This field defines the trigger causing the offset counter to increment

Table continues on the next page...

IPU_DI1_SW_GEN0_6 field descriptions (continued)

Field	Description
	000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

38.5.238 DI1 Sync Wave Gen 7 Register 0 (IPU_DI1_SW_GEN0_7)

Address: 260_0000h base + 4_8024h offset = 264_8024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_run_value_m1_7												di1_run_resolution_7		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_offset_value_7												di1_offset_ resolution_1		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN0_7 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di1_run_value_ m1_7	DI1 counter #7 pre defined value This fields defines the counter #7 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_7 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_ resolution_7	DI1 counter #1 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DI1_SW_GEN0_7 field descriptions (continued)

Field	Description
14–3 di1_offset_value_7	DI1 counter #7 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di1_offset_resolution_1	DI1 counter #7 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

38.5.239 DI1 Sync Wave Gen 8 Register 0 (IPU_DI1_SW_GEN0_8)

Address: 260_0000h base + 4_8028h offset = 264_8028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_run_value_m1_8												di1_run_resolution_8		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_offset_value_8												di1_offset_resolution_8		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN0_8 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di1_run_value_m1_8	DI1 counter #8 pre defined value This fields defines the counter #8 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_8 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_resolution_8	DI1 counter #8 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

Table continues on the next page...

IPU_DI1_SW_GEN0_8 field descriptions (continued)

Field	Description
	100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.
14–3 di1_offset_value_8	DI1 counter #8 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di1_offset_resolution_8	DI1 counter #8 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

38.5.240 DI1Sync Wave Gen 9 Register 0 (IPU_DI1_SW_GEN0_9)

Address: 260_0000h base + 4_802Ch offset = 264_802Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_run_value_m1_9												di1_run_resolution_9		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_offset_value_9												di1_offset_resolution_9		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN0_9 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–19 di1_run_value_m1_9	DI1 counter #9 pre defined value This fields defines the counter #9 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_9 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.

Table continues on the next page...

IPU_DI1_SW_GEN0_9 field descriptions (continued)

Field	Description
18–16 di1_run_ resolution_9	DI1 counter #9 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value 0.
14–3 di1_offset_value_ 9	DI1 counter #9 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
di1_offset_ resolution_9	DI1 counter #9 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

38.5.241 DI1 Sync Wave Gen 1 Register 1 (IPU_DI1_SW_GEN1_1)

Address: 260_0000h base + 4_8030h offset = 264_8030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_cnt_polarity_gen_en_1		di1_cnt_auto_reload_1	di1_cnt_clr_sel_1				di1_cnt_down_1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_cnt_polarity_trigger_sel_1			di1_cnt_polarity_clr_sel_1				di1_cnt_up_1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN1_1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di1_cnt_polarity_gen_en_1	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_reload_1	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_1 field
27–25 di1_cnt_clr_sel_1	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 Reserved 011 Reserved

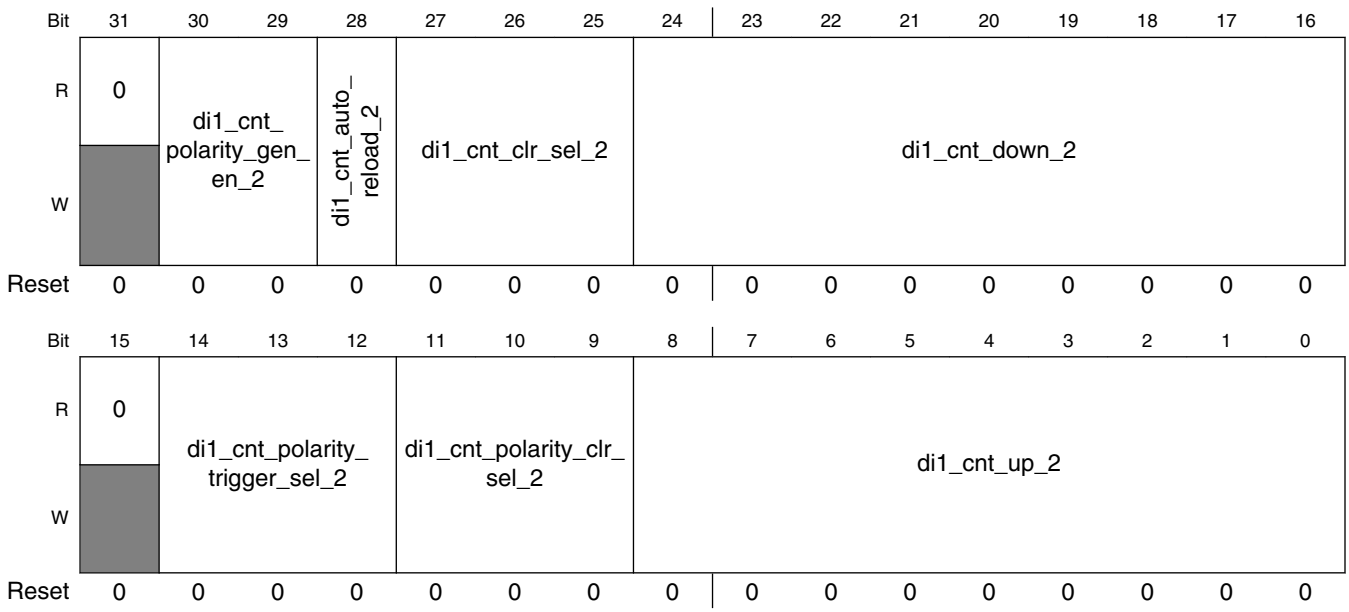
Table continues on the next page...

IPU_DI1_SW_GEN1_1 field descriptions (continued)

Field	Description
	100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
24–16 di1_cnt_down_1	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di1_cnt_polarity_trigger_sel_1	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 Reserved 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_1	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Reserved 011 Reserved 100 Reserved 101 Reserved 110 Reserved 111 Reserved
di1_cnt_up_1	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.242 DI1 Sync Wave Gen 2 Register 1 (IPU_DI1_SW_GEN1_2)

Address: 260_0000h base + 4_8034h offset = 264_8034h



IPU_DI1_SW_GEN1_2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di1_cnt_polarity_gen_en_2	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_reload_2	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_2	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 Reserved

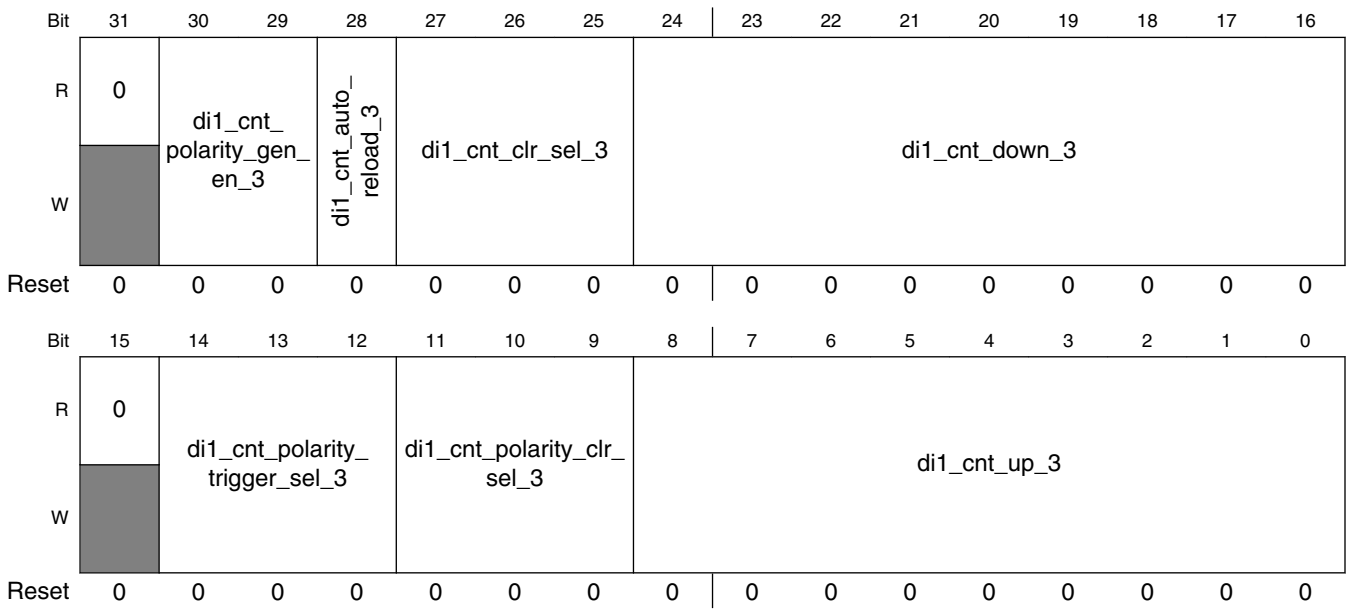
Table continues on the next page...

IPU_DI1_SW_GEN1_2 field descriptions (continued)

Field	Description
	100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
24–16 di1_cnt_down_2	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di1_cnt_polarity_trigger_sel_2	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_2	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Reserved 011 Reserved 100 Reserved 101 Reserved 110 Reserved 111 Reserved
di1_cnt_up_2	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.243 DI1 Sync Wave Gen 3 Register 1 (IPU_DI1_SW_GEN1_3)

Address: 260_0000h base + 4_8038h offset = 264_8038h



IPU_DI1_SW_GEN1_3 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di1_cnt_polarity_gen_en_3	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_reload_3	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_3	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

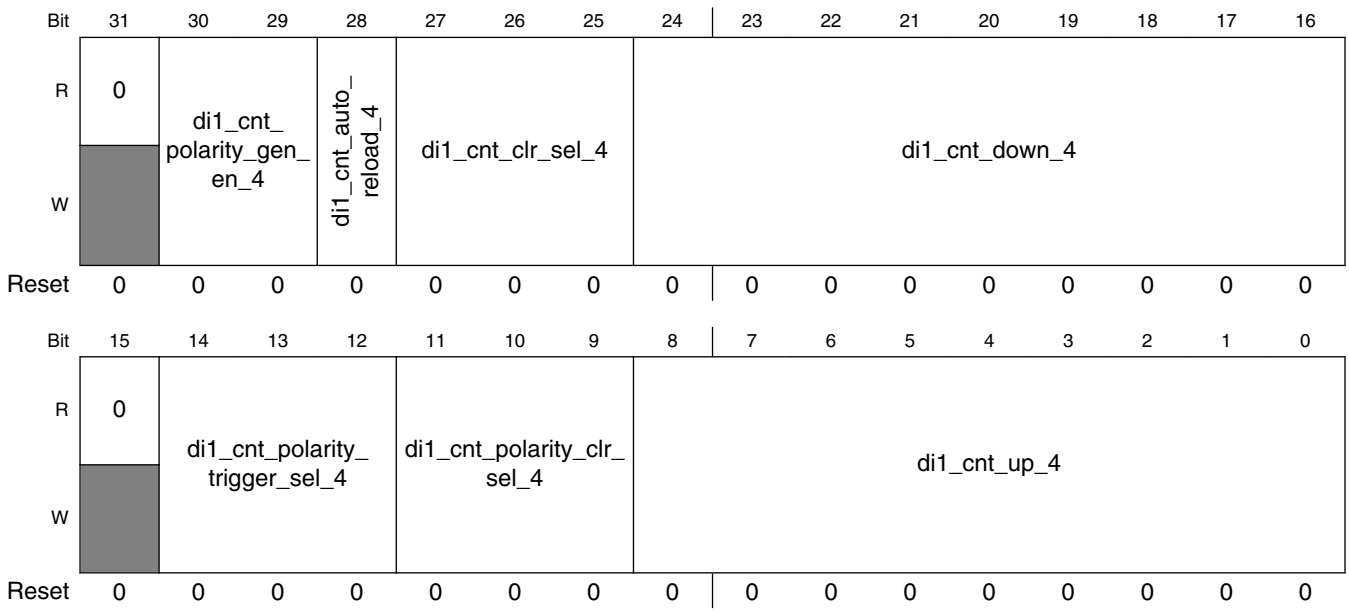
Table continues on the next page...

IPU_DI1_SW_GEN1_3 field descriptions (continued)

Field	Description
	100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
24–16 di1_cnt_down_3	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di1_cnt_polarity_trigger_sel_3	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_3	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Reserved 101 Reserved 110 Reserved 111 Reserved
di1_cnt_up_3	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.244 DI1 Sync Wave Gen 4 Register 1 (IPU_DI1_SW_GEN1_4)

Address: 260_0000h base + 4_803Ch offset = 264_803Ch



IPU_DI1_SW_GEN1_4 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di1_cnt_polarity_gen_en_4	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_reload_4	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_4	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

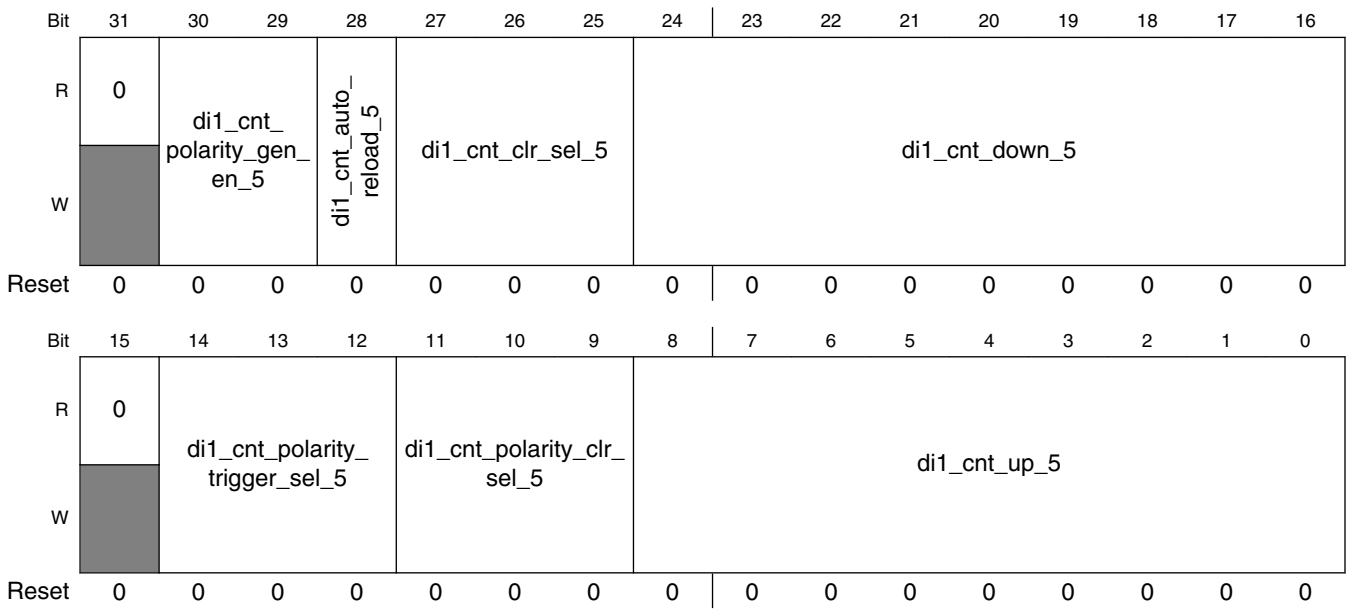
Table continues on the next page...

IPU_DI1_SW_GEN1_4 field descriptions (continued)

Field	Description
	100 The Counter is triggered by counter #3 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
24–16 di1_cnt_down_4	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di1_cnt_polarity_trigger_sel_4	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 - Counter is disabled 001 - The counter is triggered by the same trigger that triggers the displays clock. 010 - The Counter is triggered by counter #1 011 - The Counter is triggered by counter #2 100 - The Counter is triggered by counter #3 101 - CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_4	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Reserved 110 Reserved 111 Reserved
di1_cnt_up_4	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.245 DI1 Sync Wave Gen 5 Register 1 (IPU_DI1_SW_GEN1_5)

Address: 260_0000h base + 4_8040h offset = 264_8040h



IPU_DI1_SW_GEN1_5 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di1_cnt_polarity_gen_en_5	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_reload_5	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_5	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

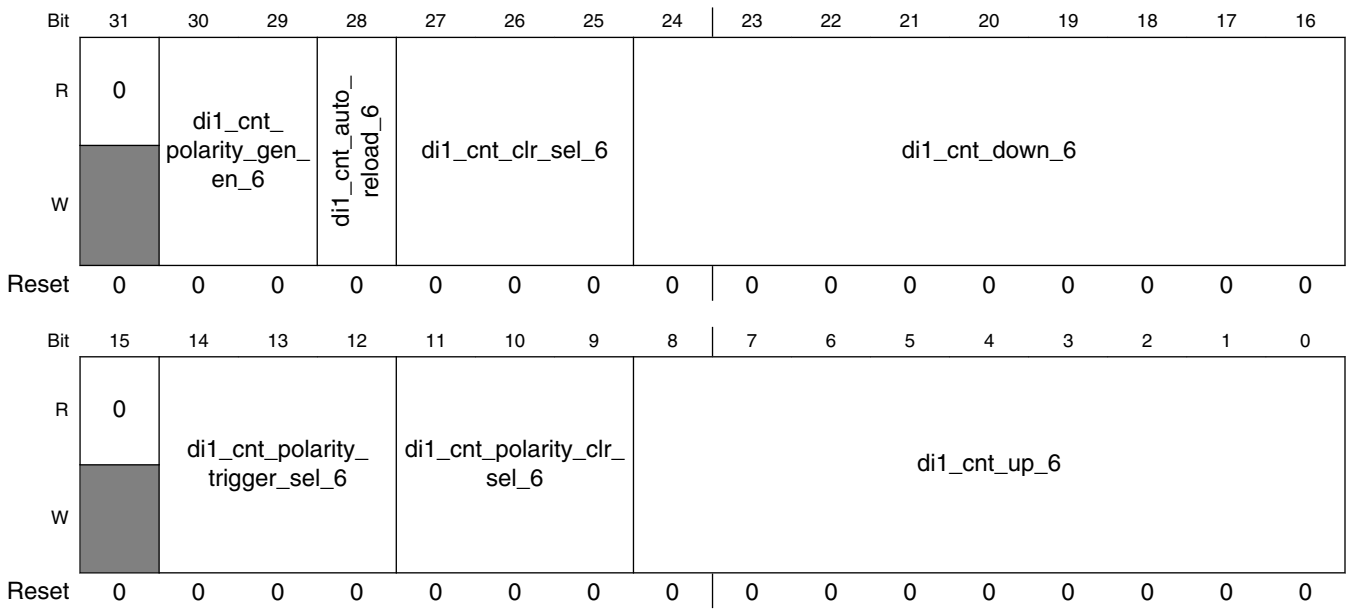
Table continues on the next page...

IPU_DI1_SW_GEN1_5 field descriptions (continued)

Field	Description
	100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 External VSYNC 111 Counter is always on.
24–16 di1_cnt_down_5	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di1_cnt_polarity_trigger_sel_5	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 External VSYNC 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_5	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Reserved 111 Reserved
di1_cnt_up_5	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.246 DI1 Sync Wave Gen 6 Register 1 (IPU_DI1_SW_GEN1_6)

Address: 260_0000h base + 4_8044h offset = 264_8044h



IPU_DI1_SW_GEN1_6 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di1_cnt_polarity_gen_en_6	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_reload_6	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_6	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

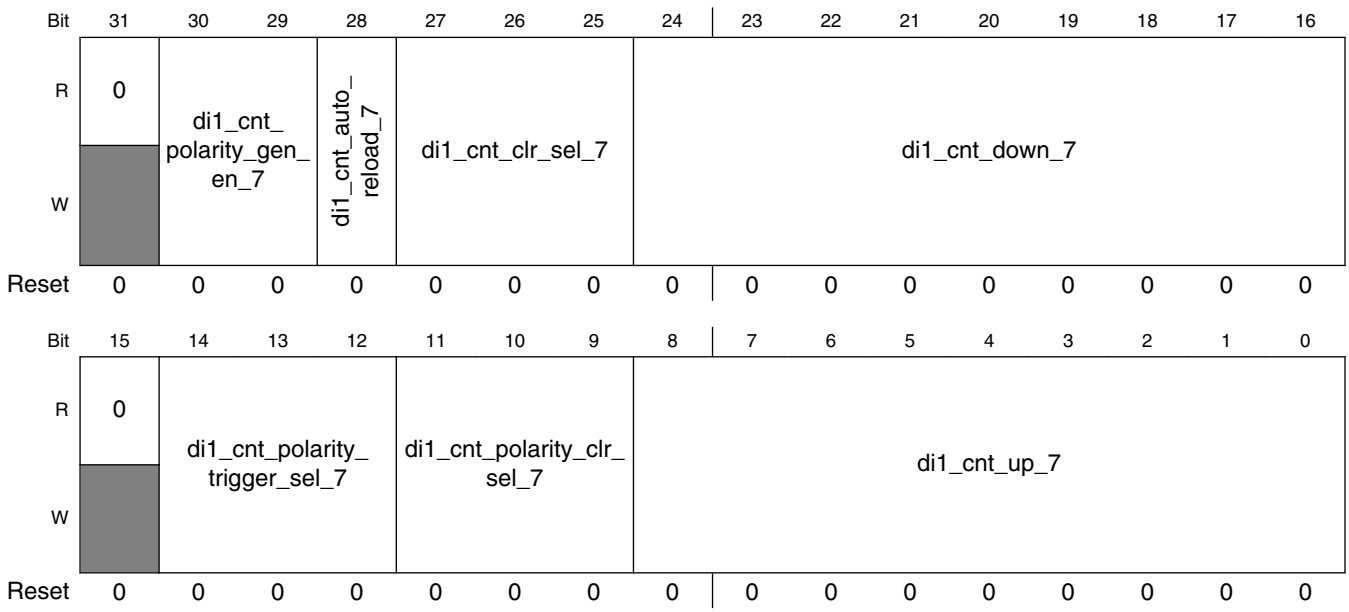
Table continues on the next page...

IPU_DI1_SW_GEN1_6 field descriptions (continued)

Field	Description
	100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
24–16 di1_cnt_down_6	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di1_cnt_polarity_trigger_sel_6	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_6	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Output is inverted if the output of counter #5 is set 111 Reserved
di1_cnt_up_6	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.247 DI1Sync Wave Gen 7 Register 1 (IPU_DI1_SW_GEN1_7)

Address: 260_0000h base + 4_8048h offset = 264_8048h



IPU_DI1_SW_GEN1_7 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di1_cnt_polarity_gen_en_7	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_reload_7	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_7	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

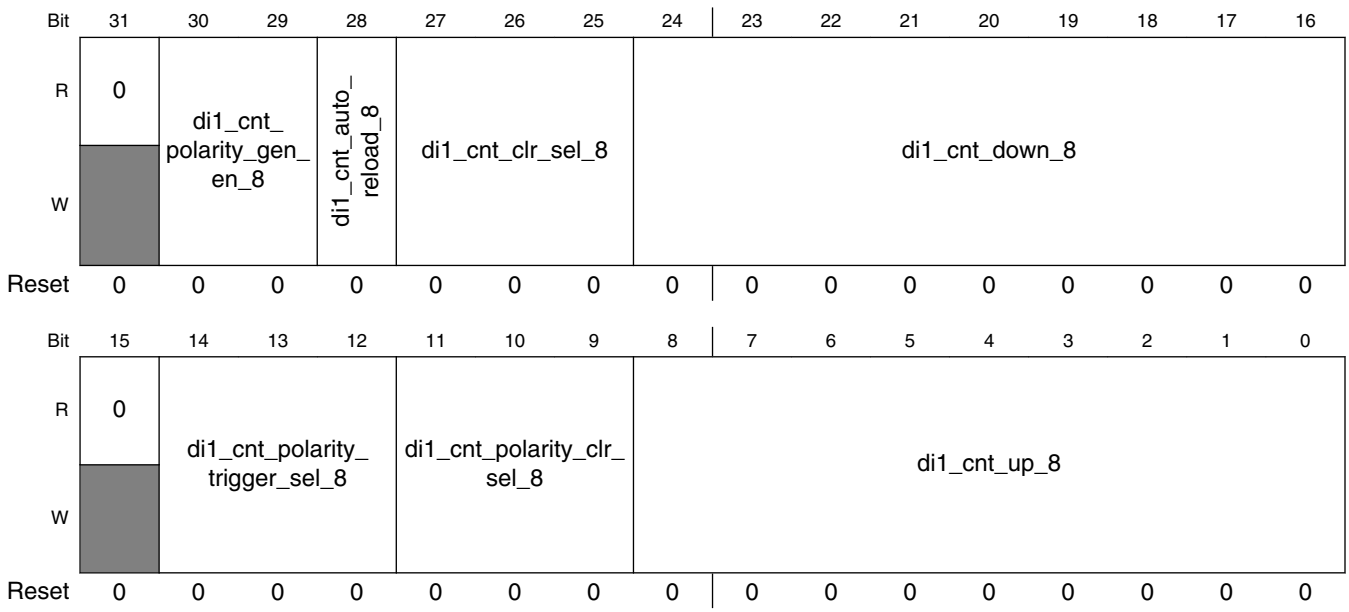
Table continues on the next page...

IPU_DI1_SW_GEN1_7 field descriptions (continued)

Field	Description
	100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
24–16 di1_cnt_down_7	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di1_cnt_polarity_trigger_sel_7	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_7	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Output is inverted if the output of counter #5 is set 111 Output is inverted if the output of counter #6 is set
di1_cnt_up_7	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.248 DI1 Sync Wave Gen 8 Register 1 (IPU_DI1_SW_GEN1_8)

Address: 260_0000h base + 4_804Ch offset = 264_804Ch



IPU_DI1_SW_GEN1_8 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 di1_cnt_polarity_gen_en_8	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_reload_8	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_8	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

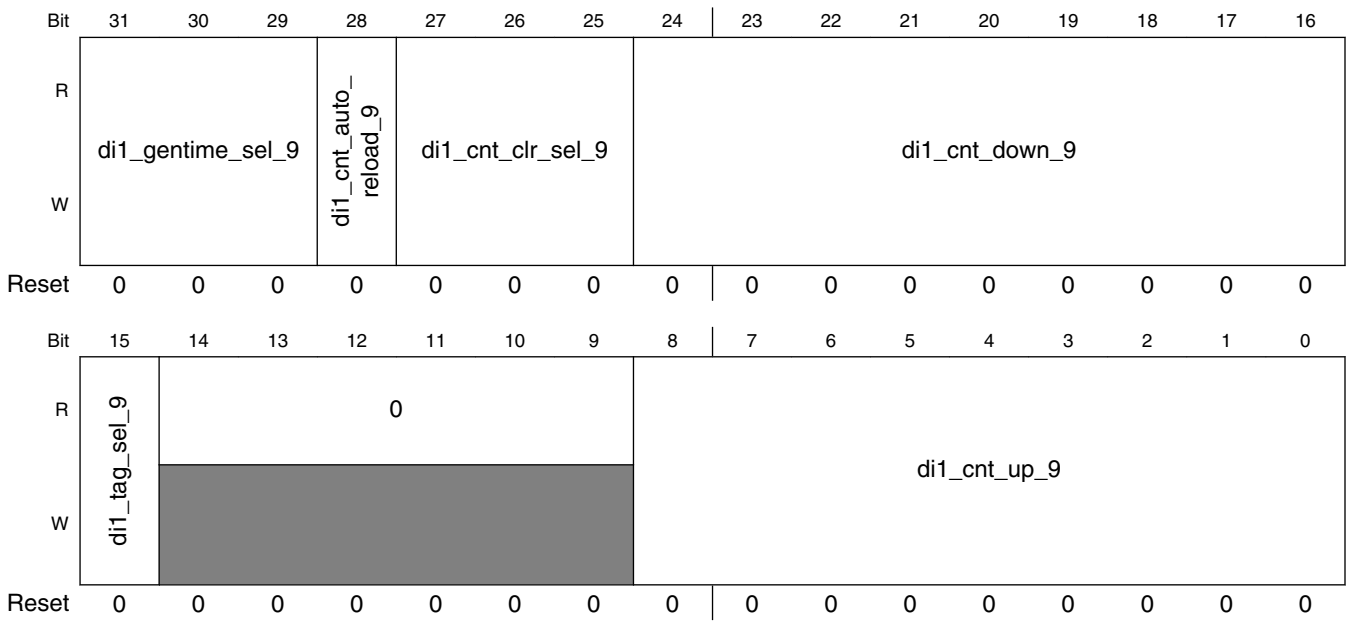
Table continues on the next page...

IPU_DI1_SW_GEN1_8 field descriptions (continued)

Field	Description
	100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
24–16 di1_cnt_down_8	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 di1_cnt_polarity_trigger_sel_8	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_8	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Output is inverted if the output of counter #5 is set 111 Output is inverted if the output of counter #6 is set
di1_cnt_up_8	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.249 DI1 Sync Wave Gen 9 Register 1 (IPU_DI1_SW_GEN1_9)

Address: 260_0000h base + 4_8050h offset = 264_8050h



IPU_DI1_SW_GEN1_9 field descriptions

Field	Description
31–29 di1_gentime_sel_9	Counter #9 main waveform select This field defines the counter that counter #9's auxiliary waveform will be attached too. 000 Counter #9's waveform is attached to counter #1's waveform 001 Counter #9's waveform is attached to counter #2's waveform 010 Counter #9's waveform is attached to counter #3's waveform 011 Counter #9's waveform is attached to counter #4's waveform 100 Counter #9's waveform is attached to counter #5's waveform 101 Counter #9's waveform is attached to counter #6's waveform 110 Counter #9's waveform is attached to counter #7's waveform 111 Counter #9's waveform is attached to counter #8's waveform
28 di1_cnt_auto_reload_9	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_9	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2

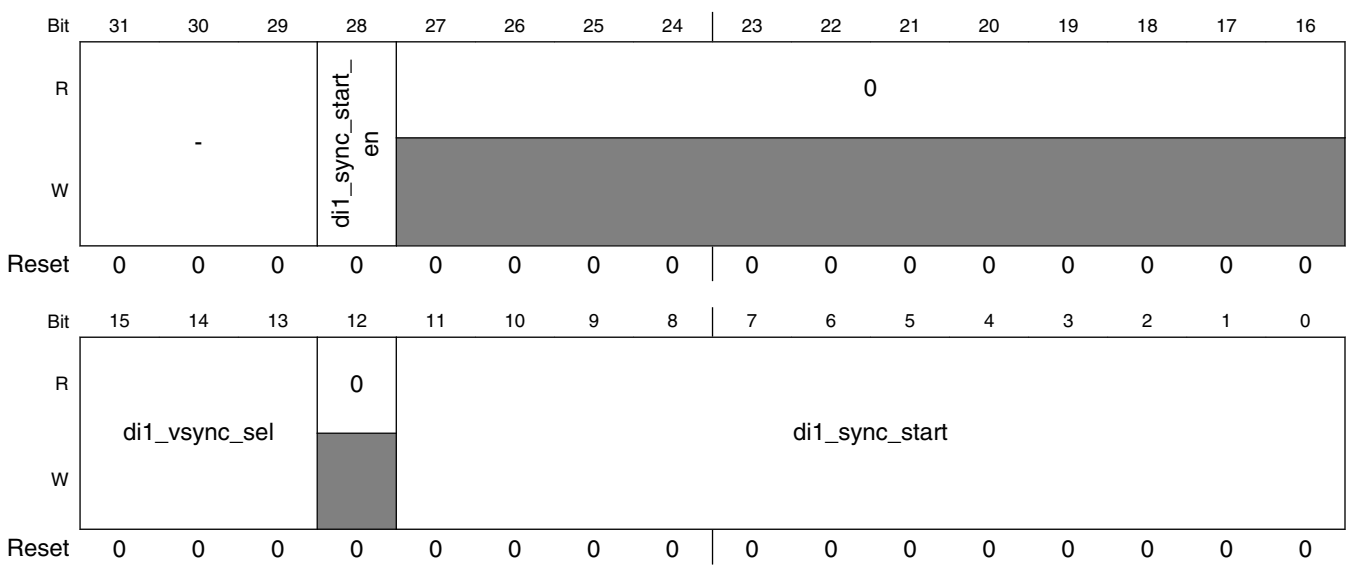
Table continues on the next page...

IPU_DI1_SW_GEN1_9 field descriptions (continued)

Field	Description
	100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
24–16 di1_cnt_down_9	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 di1_tag_sel_9	Counter #9 can send a synchronous tag when counter #9 reach its predefined value or when it's triggering counter reaches its pre defined value. 1 tag source is counter #9 0 Tag's source is the triggering counter.
14–9 Reserved	This read-only field is reserved and always has the value 0.
di1_cnt_up_9	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

38.5.250 DI1 Sync Assistance Gen Register (IPU_DI1_SYNC_AS_GEN)

Address: 260_0000h base + 4_8054h offset = 264_8054h



IPU_DI1_SYNC_AS_GEN field descriptions

Field	Description
31–29 -	Reserve
28 di1_sync_start_en	di1_sync_start_en
27–16 Reserved	This read-only field is reserved and always has the value 0.
15–13 di1_vsync_sel	VSYNC select This field defines which of the counters functions as VSYNC signal 000 VSYNC is coming from counter #1 001 VSYNC is coming from counter #2 111 VSYNC is coming from counter #8
12 Reserved	This read-only field is reserved and always has the value 0.
di1_sync_start	DI1 Sync start This field defines the number of low (including blanking rows) on the which the DI1 starts preparing the data for the next frame.

38.5.251 DI1 Data Wave Gen <i> Register (IPU_DI1_DW_GEN_i)

The DI1_DW_GEN_<i> register holds pointers for the waveform generators.

These registers have different bit arrangements for parallel and serial display. When using a parallel display VDI Plane Size Register 4 is applicable. When using a serial interface VDI Plane Size Register 4 is applicable.

Address: 260_0000h base + 4_8058h offset = 264_8058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	di1_access_size_i								di1_componnent_size_i								di1_cst_i		di1_pt_6_i		di1_pt_5_i		di1_pt_4_i		di1_pt_3_i		di1_pt_2_i		di1_pt_1_i		di1_pt_0_i	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_DW_GEN_i field descriptions

Field	Description
31–24 di1_access_size_i	DI1 Access Size <i> This field defines the amount of IPU cycles between any 2 accesses (an access may be a pixel or generic data that may have more one component)
23–16 di1_componnent_size_i	DI1 component Size This field defines the amount of IPU cycles between any 2 components

Table continues on the next page...

IPU_DI1_DW_GEN_i field descriptions (continued)

Field	Description
15–14 di1_cst_i	DI1 Chip Select pointer for waveform <i> This field points to a register that defines the waveform of the CS pin. The CS is automatically mapped to a specific display 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>
13–12 di1_pt_6_i	DI1 PIN_17 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_17 pin. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>
11–10 di1_pt_5_i	DI1 PIN_16 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_16 pin. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>
9–8 di1_pt_4_i	DI1 PIN_15 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_15 pin. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>
7–6 di1_pt_3_i	DI1 PIN_14 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_14 pin. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>
5–4 di1_pt_2_i	DI1 PIN_13 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_13 pin. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>
3–2 di1_pt_1_i	DI1 PIN_12 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_12 pin. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i>

Table continues on the next page...

IPU_DI1_DW_GEN_i field descriptions (continued)

Field	Description
	01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>
di1_pt_0_i	DI1 PIN_11 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_11 pin. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 - The waveform is defined according to the settings on DI1_DW_SET3_<i>

38.5.252 DI1 Data Wave Set 0 <i> Register (IPU_DI1_DW_SET0_i)

Address: 260_0000h base + 4_8088h offset = 264_8088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							di1_data_cnt_down0_i									0							di1_data_cnt_up0_i								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_DW_SET0_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 di1_data_cnt_down0_i	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value 0.
di1_data_cnt_up0_i	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>

38.5.253 DI1 Data Wave Set 1 <i> Register (IPU_DI1_DW_SET1_i)

Address: 260_0000h base + 4_80B8h offset = 264_80B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							di1_data_cnt_down1_i									0							di1_data_cnt_up1_i								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_DW_SET1_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 di1_data_cnt_down1_i	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value 0.
di1_data_cnt_up1_i	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>

38.5.254 DI1 Data Wave Set 2 <i> Register (IPU_DI1_DW_SET2_i)

Address: 260_0000h base + 4_80E8h offset = 264_80E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							di1_data_cnt_down2_i									0							di1_data_cnt_up2_i								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_DW_SET2_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 di1_data_cnt_down2_i	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value 0.
di1_data_cnt_up2_i	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>

38.5.255 DI1 Data Wave Set 3 <i> Register (IPU_DI1_DW_SET3_i)

Address: 260_0000h base + 4_8118h offset = 264_8118h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							di1_data_cnt_down3_i									0							di1_data_cnt_up3_i								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_DW_SET3_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 di1_data_cnt_down3_i	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value 0.
di1_data_cnt_up3_i	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>

38.5.256 DI1 Step Repeat <i> Registers (IPU_D1_STP_REP_i)

Address: 260_0000h base + 4_8148h offset = 264_8148h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				di1_step_repeat_2i												0				di1_step_repeat_2i_minus_1											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_D1_STP_REP_i field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–16 di1_step_repeat_2i	Step Repeat <i> This fields defines the amount of repetitions that will be performed by the counter <i>
15–12 Reserved	This read-only field is reserved and always has the value 0.
di1_step_repeat_2i_minus_1	Step Repeat <i> This fields defines the amount of repetitions that will be performed by the counter <i>

38.5.257 DI1Step Repeat 9 Registers (IPU_DI1_STP_REP_9)

Address: 260_0000h base + 4_8158h offset = 264_8158h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																				di1_step_repeat_9											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_STP_REP_9 field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value 0.
di1_step_repeat_9	Step Repeat 9 This fields defines the amount of repetitions that will be performed by the counter 9.

38.5.258 DI1 Serial Display Control Register (IPU_DI1_SER_CONF)

Address: 260_0000h base + 4_815Ch offset = 264_815Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DI1_SERIAL_LLA_PNTR_RS_R_1				DI1_SERIAL_LLA_PNTR_RS_R_0				DI1_SERIAL_LLA_PNTR_RS_W_1				DI1_SERIAL_LLA_PNTR_RS_W_0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DI1_SERIAL_LATCH								0		DI1_LLA_SER_ACCESS	DI1_SER_CLK_POLARITY	DI1_SERIAL_DATA_POLARITY	DI1_SERIAL_RS_POLARITY	DI1_SERIAL_CS_POLARITY	DI1_WAIT4SERIAL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SER_CONF field descriptions

Field	Description
31–28 DI1_SERIAL_LLA_PNTR_RS_R_1	RS 3 waveform pointer for read low level access This pointer defines which waveform set will be chosen when the read low level access is targeted to RS group 1. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved

Table continues on the next page...

IPU_DI1_SER_CONF field descriptions (continued)

Field	Description
27–24 DI1_SERIAL_LLA_PNTR_RS_R_0	RS 2 waveform pointer for low level access This pointer defines which waveform set will be chosen when the read low level access is targeted to RS group 0. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved
23–20 DI1_SERIAL_LLA_PNTR_RS_W_1	RS 1 waveform pointer for write low level access This pointer defines which waveform set will be chosen when the write low level access is targeted to RS group 1. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved
19–16 DI1_SERIAL_LLA_PNTR_RS_W_0	RS 0 waveform pointer for write low level access This pointer defines which waveform set will be chosen when the write low level access is targeted to RS group 0. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved
15–8 DI1_SERIAL_LATCH	DI1 Serial Latch This field defines how many cycles to insert between serial read accesses start to data sampling in the
7–6 Reserved	This read-only field is reserved and always has the value 0.
5 DI1_LLA_SER_ACCESS	Direct Low Level Access to Serial display 1 ARM platform access is performed via a direct path to the serial display in LLA mode, in this mode only the ARM platform in LLA mode can access the serial port 0 ARM platform access to the serial display port is not done directly, hence other source are allowed to access the serial port. The arbitration is done automatically
4 DI1_SER_CLK_POLARITY	Serial Clock Polarity The output polarity of the SER_CLK pin 1 The clock is inverted 0 The clock is not inverted

Table continues on the next page...

IPU_DI1_SER_CONF field descriptions (continued)

Field	Description
3 DI1_SERIAL_ DATA_ POLARITY	Serial Data Polarity The output polarity of the SER_DATA pin 1 The data is inverted 0 The data is not inverted
2 DI1_SERIAL_ RS_POLARITY	Serial RS Polarity The output polarity of the SER_RS pin 1 The RS is inverted 0 The RS is not inverted
1 DI1_SERIAL_ CS_POLARITY	Serial Chip Select Polarity The output polarity of the SER_CS pin 1 The CS is inverted 0 The CS is not inverted
0 DI1_ WAIT4SERIAL	Wait for Serial When the parallel display share pins with the serial port. Accessing the parallel port is not allowed till the serial port completes its access. 1 The parallel port should wait to the serial port as the pins are shared 0 The parallel port should not wait to the serial port as the pins are not shared

38.5.259 DI1 Special Signals Control Register (IPU_DI1_SSC)

Address: 260_0000h base + 4_8160h offset = 264_8160h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								DI1_PIN17_ERM	DI1_PIN16_ERM	DI1_PIN15_ERM	DI1_PIN14_ERM	DI1_PIN13_ERM	DI1_PIN12_ERM	DI1_PIN11_ERM	DI1_CS_ERM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										DI1_WAIT_ON	DI1_BYTE_EN_POLARITY	DI1_BYTE_EN_RD_IN	DI1_BYTE_EN_PNTR		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SSC field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value 0.
23 DI1_PIN17_ERM	DI1 PIN17 error recovery mode. This bit defines the error recovery mode of the PIN17 pin. 1 The PIN17 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC. 0 Nothing is done to the PIN17 pin following a display error detection.
22 DI1_PIN16_ERM	DI1 PIN16 error recovery mode. This bit defines the error recovery mode of the PIN16 pin. 1 The PIN16 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC. 0 Nothing is done to the PIN16 pin following a display error detection.
21 DI1_PIN15_ERM	DI1 PIN15 error recovery mode. This bit defines the error recovery mode of the PIN15 pin. 1 The PIN15 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC. 0 Nothing is done to the PIN15 pin following a display error detection.
20 DI1_PIN14_ERM	DI1 PIN14 error recovery mode. This bit defines the error recovery mode of the PIN14 pin. 1 The PIN14 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC. 0 Nothing is done to the PIN14 pin following a display error detection.
19 DI1_PIN13_ERM	DI1 PIN13 error recovery mode. This bit defines the error recovery mode of the PIN13 pin. 1 The PIN13 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC. 0 Nothing is done to the PIN13 pin following a display error detection.
18 DI1_PIN12_ERM	DI1 PIN12 error recovery mode. This bit defines the error recovery mode of the PIN12 pin. 1 The PIN12 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC. 0 Nothing is done to the PIN12 pin following a display error detection.
17 DI1_PIN11_ERM	DI1 PIN11 error recovery mode. This bit defines the error recovery mode of the PIN11 pin. 1 The PIN11 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC. 0 Nothing is done to the PIN11 pin following a display error detection.
16 DI1_CS_ERM	DI1 GLUELOGIC error recovery mode. This bit defines the error recovery mode of the GLUELOGIC.

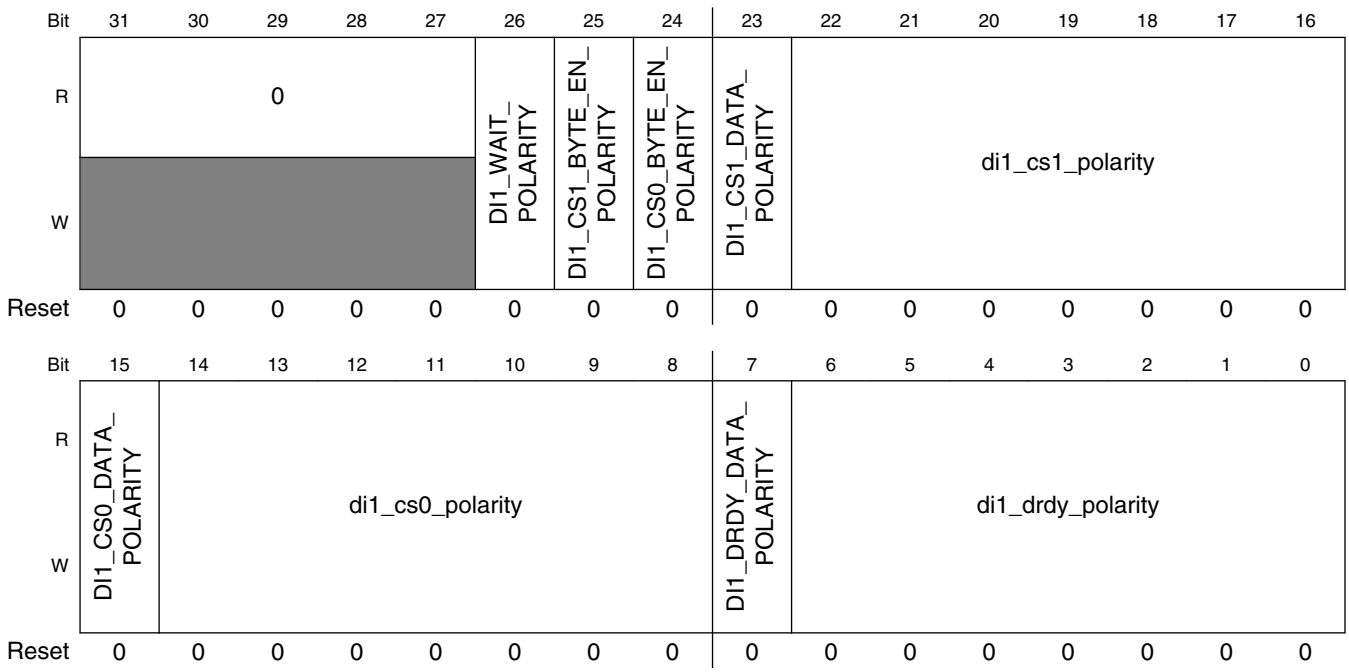
Table continues on the next page...

IPU_DI1_SSC field descriptions (continued)

Field	Description
	1 The GLUELOGIC is release in case of a synchronous display error. The release will be done on the next VSYNC. 0 Nothing is done to the GLUELOGIC following a display error detection.
15–6 Reserved	This read-only field is reserved and always has the value 0.
5 DI1_WAIT_ON	Wait On This field defines the DC's response to WAIT signal 1 The DC holds the flow as long as WAIT is asserted. 0 The DC continues the flow regardless the WAIT signal.
4 DI1_BYTE_EN_POLARITY	Byte Enable polarity This bit defines the polarity of the byte enable signals to the display. 1 active high. 0 active low.
3 DI1_BYTE_EN_RD_IN	Byte Enable Read In This bit selects the source of the byte enable pins 1 The write byte enable signals are routed via bits [17:16] of the display's data, The read byte enable signals are routed via bits [19:18 of the display's data 0 The byte enable signals are routed via bits [17:16] of the display's data for both read and write
DI1_BYTE_EN_PNTR	Byte Enable Pointer This pointer selects the pin asserted along with the byte enables signals 000 wave form of byte enable as pin_11 001 wave form of byte enable as pin_12 111 wave form of byte enable as suitable CS pin

38.5.260 DI1 Polarity Register (IPU_DI1_POL)

Address: 260_0000h base + 4_8164h offset = 264_8164h



IPU_DI1_POL field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26 DI1_WAIT_POLARITY	WAIT polarity This bit defines the polarity of the wait signal input coming from the displa1 1 active high 0 active low
25 DI1_CS1_BYTE_EN_POLARITY	Byte Enable associated with CS1 polarity This bit defines the polarity of the byte enable signals to the display 1 active high 0 active low
24 DI1_CS0_BYTE_EN_POLARITY	Byte Enable associated with CS0 polarity This bit defines the polarity of the byte enable signals to the display 1 active high 0 active low
23 DI1_CS1_DATA_POLARITY	Data Polarity associated with CS1

Table continues on the next page...

IPU_DI1_POL field descriptions (continued)

Field	Description
22–16 di1_cs1_polarity	DI1 output pin's polarity for CS1 This bits define the polarity of each of the DI's outputs when CS1 is asserted 1 The output pin is active high 0 The output pin is active low
15 DI1_CS0_DATA_POLARITY	Data Polarity associated with CS0
14–8 di1_cs0_polarity	DI1 output pin's polarity for CS1 This bits define the polarity of each of the DI's outputs when CS1 is asserted 1 The output pin is active high 0 The output pin is active low
7 DI1_DRDY_DATA_POLARITY	Data Polarity associated with DRDY
di1_drdy_polarity	DI1 output dynamic pin's polarity for synchronous access This bits define the polarity of each of the DI's outputs when synchronous display access is asserted The pins' default polarity is the same as defined in the di0_drdy_polarity bits 1 The output pin is active high 0 The output pin is active low

38.5.261 DI1Active Window 0 Register (IPU_DI1_AW0)

Address: 260_0000h base + 4_8168h offset = 264_8168h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DI1_AW_TRIG_SEL				DI1_AW_HEND												DI1_AW_HCOUNT_SEL				DI1_AW_HSTART											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_AW0 field descriptions

Field	Description
31–28 DI1_AW_TRIG_SEL	This field selects the trigger for sending data during the display's active window 000 disabled 001 The trigger is the same trigger that triggers the displays clock. 010 The trigger is counter #1 011 The trigger is counter #2 100 The trigger is counter #3 101 The trigger is counter #4

Table continues on the next page...

IPU_DI1_AW0 field descriptions (continued)

Field	Description
	110 The trigger is counter #5 111 The trigger is always on.
27–16 DI1_AW_HEND	This field defines the horizontal end of the active window
15–12 DI1_AW_ HCOUNT_SEL	This field selects the counter that counts the horizontal position of the display's active window 0000 disabled 0001 reserved 0010 The counter is counter #1 0011 The counter is counter #2 0100 The counter is counter #3 0101 The counter is counter #4 0110 The counter is counter #5 1001 The counter is counter #8
DI1_AW_ HSTART	This field defines the horizontal start of the active window DI1_AW_HSTART < DI1_AW_HEND

38.5.262 DI1 Active Window 1 Register (IPU_DI1_AW1)

Address: 260_0000h base + 4_816Ch offset = 264_816Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DI1_AW_VEND												DI1_AW_ VCOUNT_ SEL				DI1_AW_VSTART											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_AW1 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–16 DI1_AW_VEND	This field defines the vertical end of the active window
15–12 DI1_AW_ VCOUNT_SEL	This field selects the counter that counts the vertical position of the display's active window 0000 disabled 0001 reserved 0010 The counter is counter #1 0011 The counter is counter #2 0100 The counter is counter #3 0101 The counter is counter #4 0110 The counter is counter #5 1001 The counter is counter #8

Table continues on the next page...

IPU_DI1_AW1 field descriptions (continued)

Field	Description
DI1_AW_ VSTART	This field defines the vertical start of the active window DI1_AW_VSTART < DI1_AW_VEND

38.5.263 DI1 Screen Configuration Register (IPU_DI1_SCR_CONF)

Address: 260_0000h base + 4_8170h offset = 264_8170h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																DI1_SCREEN_HEIGHT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SCR_CONF field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value 0.
DI1_SCREEN_ HEIGHT	This field defines the number of display rows (Number_of_ROWS = DI1_SCREEN_HEIGHT+1) This field is used for VSYNC calculation and for anti-tearing

38.5.264 DI1 Status Register (IPU_DI1_STAT)

Address: 260_0000h base + 4_8174h offset = 264_8174h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												DI1_CNTR_FIFO_FULL	DI1_CNTR_FIFO_EMPTY	DI1_READ_FIFO_FULL	DI1_READ_FIFO_EMPTY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IPU_DI1_STAT field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value 0.
3 DI1_CNTR_ FIFO_FULL	This bit indicates a full state of the DI1 FIFO. This FIFO is part of the DI1 synchronizer.
2 DI1_CNTR_ FIFO_EMPTY	This bit indicates an empty state of the DI1 FIFO. This FIFO is part of the DI1 synchronizer.
1 DI1_READ_ FIFO_FULL	This bit indicates a full state of the DI1 FIFO when performing a read. This FIFO is part of the DI1 synchronizer.
0 DI1_READ_ FIFO_EMPTY	This bit indicates an empty state of the DI1 FIFO when performing a read. This FIFO is part of the DI1 synchronizer.

38.5.265 SMFC Mapping Register (IPU_SMFC_MAP)

The purpose of this register is to map CSI frames to IDMAC channels.

Address: 260_0000h base + 5_0000h offset = 265_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																				MAP_ CH3			MAP_ CH2			MAP_ CH1			MAP_ CH0		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_SMFC_MAP field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value 0.
11–9 MAP_CH3	DMASMFC channel 3mapping bits. 000 CSI0, ID=0 mapped to DMASMFC channel 3. 001 CSI0, ID=1 mapped to DMASMFC channel 3. 010 CSI0, ID=2 mapped to DMASMFC channel 3. 011 CSI0, ID=3 mapped to DMASMFC channel 3. 100 CSI1, ID=0 mapped to DMASMFC channel 3. 101 CSI1, ID=1 mapped to DMASMFC channel 3. 110 CSI1, ID=2 mapped to DMASMFC channel 3. 111 CSI1, ID=3 mapped to DMASMFC channel 3.
8–6 MAP_CH2	DMASMFC channel 2 mapping bits. 000 CSI0, ID=0 mapped to DMASMFC channel 2.

Table continues on the next page...

IPU_SMFC_MAP field descriptions (continued)

Field	Description
	001 CSI0, ID=1 mapped to DMASMFC channel 2. 010 CSI0, ID=2 mapped to DMASMFC channel 2. 011 CSI0, ID=3 mapped to DMASMFC channel 2. 100 CSI1, ID=0 mapped to DMASMFC channel 2. 101 CSI1, ID=1 mapped to DMASMFC channel 2. 110 CSI1, ID=2 mapped to DMASMFC channel 2. 111 CSI1, ID=3 mapped to DMASMFC channel 2.
5–3 MAP_CH1	DMASMFC channel 1 mapping bits. 000 CSI0, ID=0 mapped to DMASMFC channel 1. 001 CSI0, ID=1 mapped to DMASMFC channel 1. 010 CSI0, ID=2 mapped to DMASMFC channel 1. 011 CSI0, ID=3 mapped to DMASMFC channel 1. 100 CSI1, ID=0 mapped to DMASMFC channel 1. 101 CSI1, ID=1 mapped to DMASMFC channel 1. 110 CSI1, ID=2 mapped to DMASMFC channel 1. 111 CSI1, ID=3 mapped to DMASMFC channel 1.
MAP_CH0	DMASMFC channel 0 mapping bits. 000 CSI0, ID=0 mapped to DMASMFC channel 0. 001 CSI0, ID=1 mapped to DMASMFC channel 0. 010 CSI0, ID=2 mapped to DMASMFC channel 0. 011 CSI0, ID=3 mapped to DMASMFC channel 0. 100 CSI1, ID=0 mapped to DMASMFC channel 0. 101 CSI1, ID=1 mapped to DMASMFC channel 0. 110 CSI1, ID=2 mapped to DMASMFC channel 0. 111 CSI1, ID=3 mapped to DMASMFC channel 0.

38.5.266 SMFC Watermark Control Register (IPU_SMFC_WMC)

The purpose of this register is to control watermarks levels of DMA channels. The bit setting given relative to FIFO size and not in number of words since FIFO size depend from number of enabled DMA channels.

Address: 260_0000h base + 5_0004h offset = 265_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				WM3_			WM3_			WM2_			WM2_			0				WM1_			WM1_			WM0_			WM0_		
W					CLR			SET			CLR			SET							CLR			SET			CLR			SET		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	1	0

IPU_SMFC_WMC field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–25 WM3_CLR	Watermark "clear" level of DMASMFC channel 3. 000 clear watermark level when FIFO is full on 1/8 of their size. 001 clear watermark level when FIFO is full on 2/8 of their size. 110 clear watermark level when FIFO is full on 7/8 of their size. 111 clear watermark level when FIFO is full.
24–22 WM3_SET	Watermark "set" level of DMASMFC channel 3 000 set watermark level when FIFO is full on 1/8 of their size. 001 set watermark level when FIFO is full on 2/8 of their size. 110 set watermark level when FIFO is full on 7/8 of their size. 111 set watermark level when FIFO is full.
21–19 WM2_CLR	Watermark "clear" level of DMASMFC channel 2. 000 clear watermark level when FIFO is full on 1/8 of their size. 001 clear watermark level when FIFO is full on 2/8 of their size. 110 clear watermark level when FIFO is full on 7/8 of their size. 111 clear watermark level when FIFO is full.
18–16 WM2_SET	Watermark "set" level of DMASMFC channel 2. 000 set watermark level when FIFO is full on 1/8 of their size. 001 set watermark level when FIFO is full on 2/8 of their size. 110 set watermark level when FIFO is full on 7/8 of their size. 111 set watermark level when FIFO is full.
15–12 Reserved	This read-only field is reserved and always has the value 0.
11–9 WM1_CLR	Watermark "clear" level of DMASMFC channel 1. 000 clear watermark level when FIFO is full on 1/8 of their size. 001 clear watermark level when FIFO is full on 2/8 of their size. 110 clear watermark level when FIFO is full on 7/8 of their size. 111 clear watermark level when FIFO is full.
8–6 WM1_SET	Watermark "set" level of DMASMFC channel 1. 000 set watermark level when FIFO is full on 1/8 of their size. 001 set watermark level when FIFO is full on 2/8 of their size. 110 set watermark level when FIFO is full on 7/8 of their size. 111 set watermark level when FIFO is full.
5–3 WM0_CLR	Watermark "clear" level of DMASMFC channel 0. 000 clear watermark level when FIFO is full on 1/8 of their size. 001 clear watermark level when FIFO is full on 2/8 of their size. 110 clear watermark level when FIFO is full on 7/8 of their size. 111 clear watermark level when FIFO is full.
WM0_SET	Watermark "set" level of DMASMFC channel 0.

Table continues on the next page...

IPU_SMFC_WMC field descriptions (continued)

Field	Description
	000 set watermark level when FIFO is full on 1/8 of their size. 001 set watermark level when FIFO is full on 2/8 of their size. 110 set watermark level when FIFO is full on 7/8 of their size. 111 set watermark level when FIFO is full.

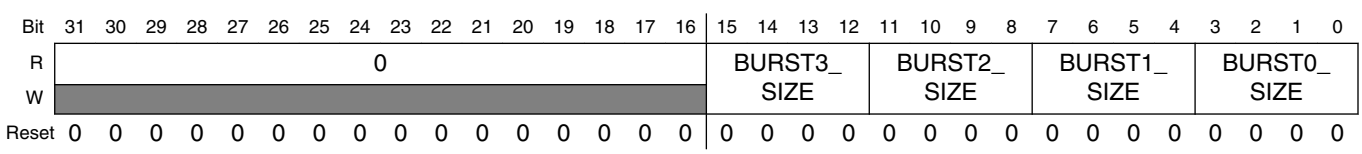
38.5.267 SMFC Burst Size Register (IPU_SMFC_BS)

This register holds the burst size value for each DMASMFC channel. The burst size is the number of IDMAC's active accesses that will done for each IDMAC's burst. This number is a function of PFS, BPP & NPB parameters in the IDMAC's CPMEM. These are the parameters corresponding to the IDMAC's channel used. The table below describes what should be the burst size according to PFS, BPP & NPB settings

Table 38-39. SMFC Burst Size

BPP	PFS	BURST_SIZE
8	6	NPB[6:4]
16	6	NPB[6:3]
All other	All other	NPB[6:2]

Address: 260_0000h base + 5_0008h offset = 265_0008h



IPU_SMFC_BS field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–12 BURST3_SIZE	Burst Size of SMFCDMA channel 3. The value programed here should be BURST_SIZE-1 (for example: for a burst size of 1, write 0 to these bits)
11–8 BURST2_SIZE	Burst Size of SMFCDMA channel 2. The value programed here should be BURST_SIZE-1 (for example: for a burst size of 1, write 0 to these bits)
7–4 BURST1_SIZE	Burst Size of SMFCDMA channel 1. The value programed here should be BURST_SIZE-1 (for example: for a burst size of 1, write 0 to these bits)
BURST0_SIZE	Burst Size of SMFCDMA channel 0.

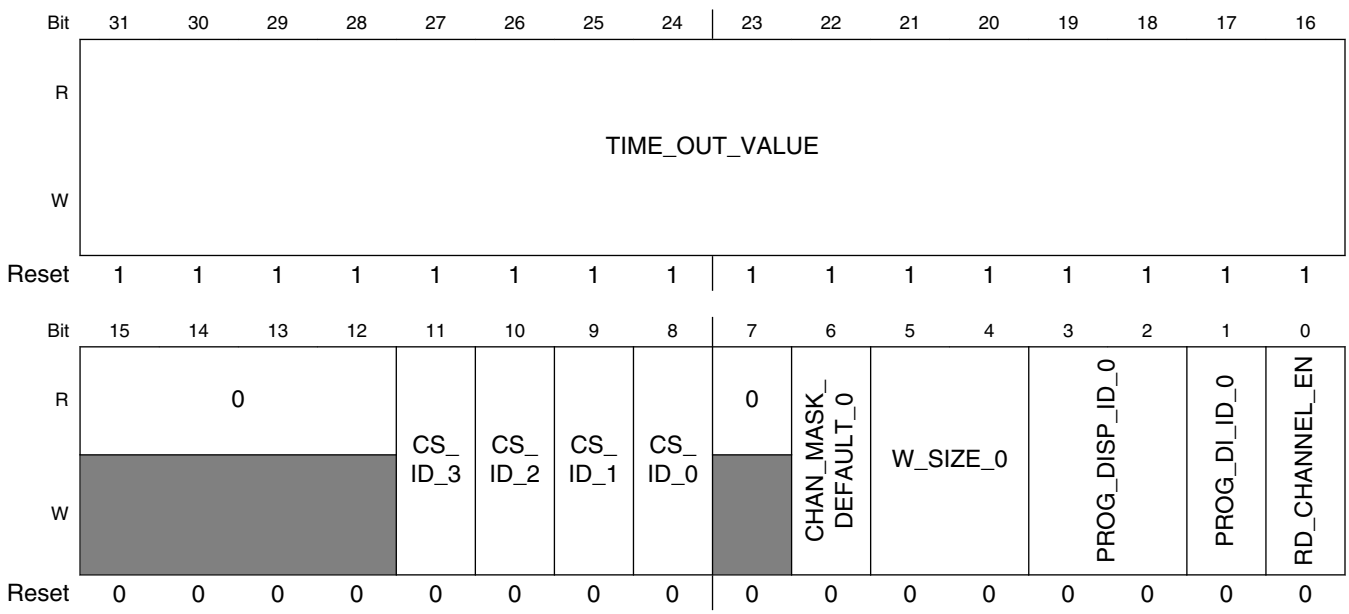
Table continues on the next page...

IPU_SMFC_BS field descriptions (continued)

Field	Description
	The value programed here should be BURST_SIZE-1 (for example: for a burst size of 1, write 0 to these bits)

38.5.268 DC Read Channel Configuration Register (IPU_DC_READ_CH_CONF)

Address: 260_0000h base + 5_8000h offset = 265_8000h



IPU_DC_READ_CH_CONF field descriptions

Field	Description
31–16 TIME_OUT_ VALUE	Time out value. In case of a error during read accesses to the display, where no response from the display was received. A time-out counter will terminate the current access and perform the next commend defined in the microcode. This field defines the amount of the hsp_clk cycles counted before the time-out event is issued. This event is tied to the interrupt controller and can generate an error interrupt.
15–12 Reserved	This read-only field is reserved and always has the value 0.
11 CS_ID_3	This bit maps an asynchronous display to a chip select 1 display #3 is connected to CS1 0 display #3 is connected to CS0
10 CS_ID_2	This bit maps an asynchronous display to a chip select

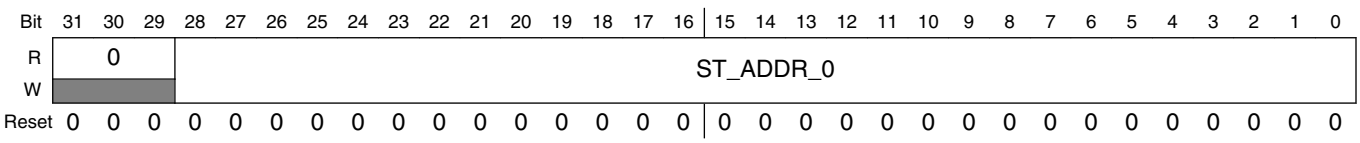
Table continues on the next page...

IPU_DC_READ_CH_CONF field descriptions (continued)

Field	Description
	1 display #2 is connected to CS1 0 display #2 is connected to CS0
9 CS_ID_1	This bit maps an asynchronous display to a chip select 1 display #1 is connected to CS1 0 display #1 is connected to CS0
8 CS_ID_0	This bit maps an asynchronous display to a chip select 1 display #0 is connected to CS1 0 display #0 is connected to CS0
7 Reserved	This read-only field is reserved and always has the value 0.
6 CHAN_MASK_DEFAULT_0	Event mask bit for the read channel When more then one event is used during a flow (EOF, EOL, NL, NF, EOFIELD, etc.) masks all the event besides the event that is defined as the highest priority event 1 All the events are used - no mask 0 Only the highest priority event is used, the rest are masked
5–4 W_SIZE_0	Word Size The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used
3–2 PROG_DISP_ID_0	The field defines which one of the 4 displays can be read. 00 display #0 01 display #1 10 display #2 11 display #3
1 PROG_DI_ID_0	This bit select the DI which a read transaction can be performed through 1 DI #1 0 DI #0
0 RD_CHANNEL_EN	This bit enables the read channel. 1 The Read channel is enabled 0 The Read channel is disabled

38.5.269 DC Read Channel Start Address Register (IPU_DC_READ_SH_ADDR)

Address: 260_0000h base + 5_8004h offset = 265_8004h

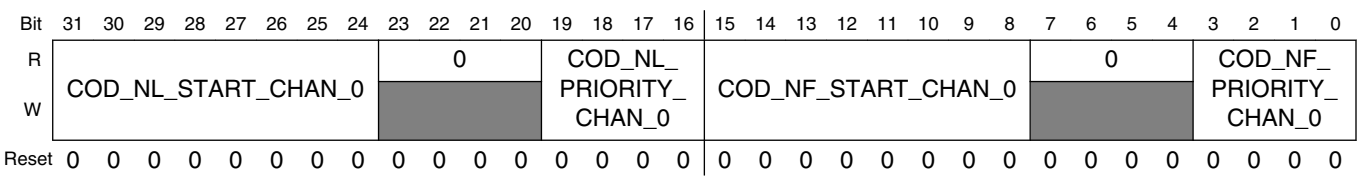


IPU_DC_READ_SH_ADDR field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
ST_ADDR_0	This field defines the start address within the display's memory space where the read transactions will be done from.

38.5.270 DC Routine Link Register 0 Channel 0 (IPU_DC_RL0_CH_0)

Address: 260_0000h base + 5_8008h offset = 265_8008h



IPU_DC_RL0_CH_0 field descriptions

Field	Description
31–24 COD_NL_START_CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the new line event (NL) resides
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_NL_PRIORITY_CHAN_0	This field defines the priority of the new line (NL) event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest)

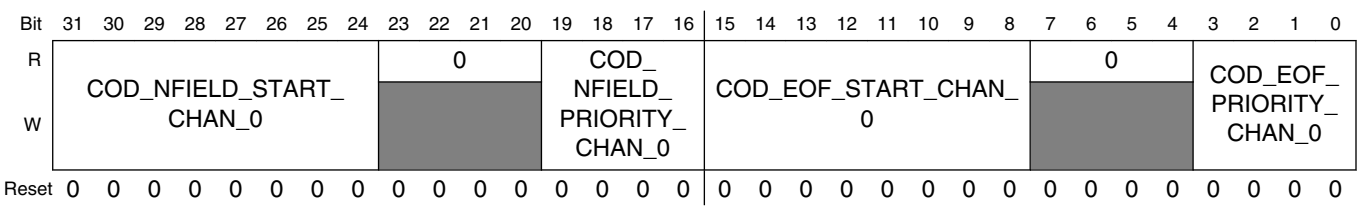
Table continues on the next page...

IPU_DC_RL0_CH_0 field descriptions (continued)

Field	Description
	1110 Reserved 1111 Reserved
15–8 COD_NF_START_CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the new Frame event (NF) resides
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NF_PRIORITY_CHAN_0	This field defines the priority of the new frame (NF) event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.271 DC Routine Link Register 1 Channel 0 (IPU_DC_RL1_CH_0)

Address: 260_0000h base + 5_800Ch offset = 265_800Ch



IPU_DC_RL1_CH_0 field descriptions

Field	Description
31–24 COD_NFIELD_START_CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the new field event resides
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_NFIELD_PRIORITY_CHAN_0	This field defines the priority of the new field event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2

Table continues on the next page...

IPU_DC_RL1_CH_0 field descriptions (continued)

Field	Description
	1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOF_ START_CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the end-of-frame event (EOF) resides
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_EOF_ PRIORITY_ CHAN_0	This field defines the priority of the end-of-frame event (EOF) event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.272 DC Routine Link Register2 Channel 0 (IPU_DC_RL2_CH_0)

Address: 260_0000h base + 5_8010h offset = 265_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_EOFIELD_START_CHAN_0								0				COD_EOFIELD_PRIORITY_CHAN_0				COD_EOL_START_CHAN_0								0				COD_EOL_PRIORITY_CHAN_0			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL2_CH_0 field descriptions

Field	Description
31–24 COD_EOFIELD_ START_CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the end-of-field event resides
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_EOFIELD_ PRIORITY_ CHAN_0	This field defines the priority of the end-of-field event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest)

Table continues on the next page...

IPU_DC_RL2_CH_0 field descriptions (continued)

Field	Description
	0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOL_ START_CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the end-of-line event (EOL) resides
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_EOL_ PRIORITY_ CHAN_0	This field defines the priority of the end-of-line event (EOL) event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.273 DC Routine Link Registe3 Channel 0 (IPU_DC_RL3_CH_0)

Address: 260_0000h base + 5_8014h offset = 265_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_CHAN_START_CHAN_0								0				COD_NEW_CHAN_PRIORITY_CHAN_0				COD_NEW_ADDR_START_CHAN_0								0				COD_NEW_ADDR_PRIORITY_CHAN_0			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL3_CH_0 field descriptions

Field	Description
31–24 COD_NEW_ CHAN_START_ CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_NEW_ CHAN_ PRIORITY_ CHAN_0	This field defines the priority of the new channel event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable)

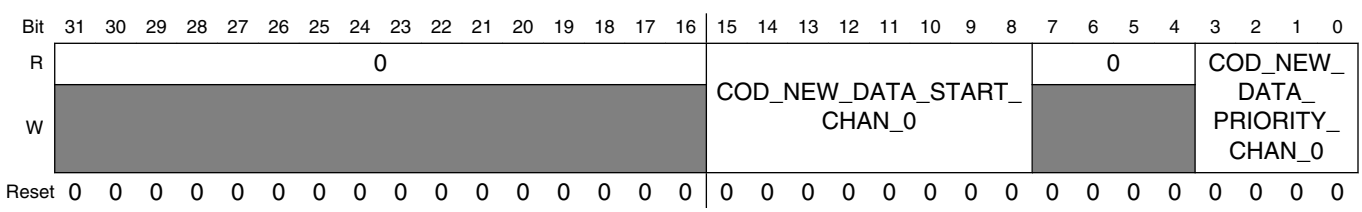
Table continues on the next page...

IPU_DC_RL3_CH_0 field descriptions (continued)

Field	Description
	0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NEW_ADDR_START_CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NEW_ADDR_PRIORITY_CHAN_0	This field defines the priority of the new address event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.274 DC Routine Link Register 4 Channel 0 (IPU_DC_RL4_CH_0)

Address: 260_0000h base + 5_8018h offset = 265_8018h



IPU_DC_RL4_CH_0 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_DATA_START_CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides
7–4 Reserved	This read-only field is reserved and always has the value 0.

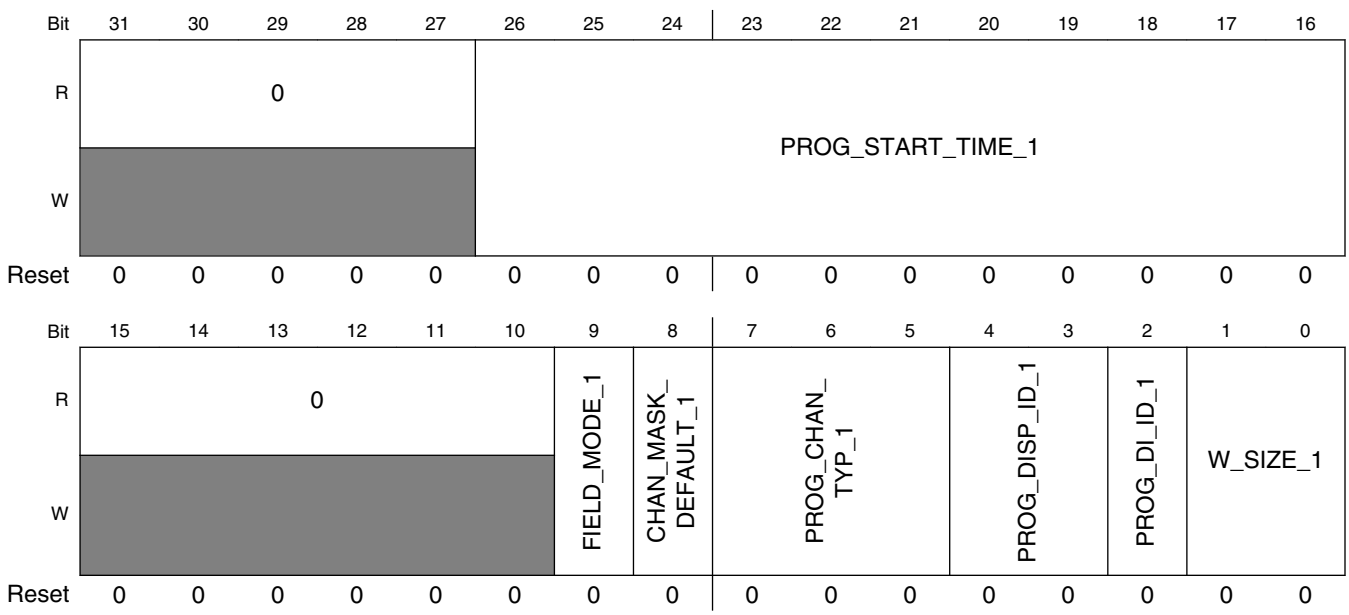
Table continues on the next page...

IPU_DC_RL4_CH_0 field descriptions (continued)

Field	Description
COD_NEW_DATA_PRIORITY_CHAN_0	This field defines the priority of the new data event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.275 DC Write Channel 1 Configuration Register (IPU_DC_WR_CH_CONF_1)

Address: 260_0000h base + 5_801Ch offset = 265_801Ch



IPU_DC_WR_CH_CONF_1 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26–16 PROG_START_TIME_1	This field defines the delay between display 's vertical synchronization pulse and the start time point of DC's channel 1 window. The delay is defined in pairs of rows. It is used for tearing elimination
15–10 Reserved	This read-only field is reserved and always has the value 0.

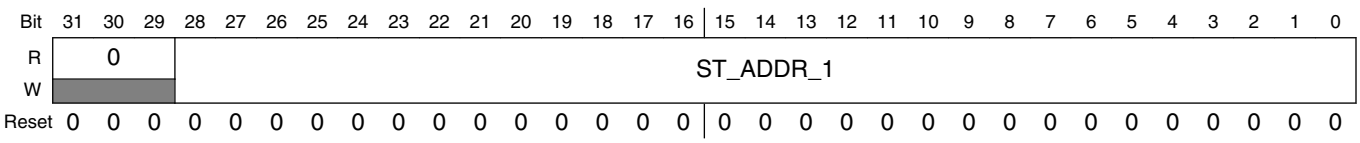
Table continues on the next page...

IPU_DC_WR_CH_CONF_1 field descriptions (continued)

Field	Description
9 FIELD_MODE_1	Field mode bit for channel #1 This bit defines if the channel works in field mode or frame mode; This bit is relevant if the flow is sync flow 1 Field mode 0 Frame mode
8 CHAN_MASK_DEFAULT_1	Event mask bit for channel #1 When more then one event is used during a flow (EOF, EOL, NL, NF, EOFIELD, etc.) masks all the event besides the event that is defined as the highest priority event 1 All the events are used - no mask 0 Only the highest priority event is used, the rest are masked
7–5 PROG_CHAN_TYP_1	This field define the mode of operation of channel #1 000 Disable 001 Reserved 010 Reserved 100 Normal mode without anti-tearing. For sync display this is the only mode allowed 101 Normal mode with anti-tearing 110 Reserved 111 Additional command channel is added to the flow handled by DC channel #1
4–3 PROG_DISP_ID_1	The field defines which one of the 4 displays is associated with channel #1. 00 display #0 01 display #1 10 display #2 11 display #3
2 PROG_DI_ID_1	This bit select the DI which a transaction associated with channel #1 can be performed to 1 DI #1 0 DI #0
W_SIZE_1	Word Size associated with channel #1 The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

38.5.276 DC Write Channel 1 Address Configuration Register (IPU_DC_WR_CH_ADDR_1)

Address: 260_0000h base + 5_8020h offset = 265_8020h

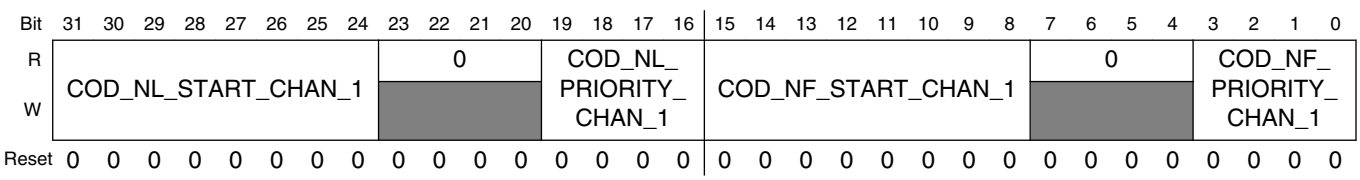


IPU_DC_WR_CH_ADDR_1 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
ST_ADDR_1	This field defines the start address within the display's memory space where the write transactions will be done to for channel #1.

38.5.277 DC Routine Link Register 0 Channel 1 (IPU_DC_RL0_CH_1)

Address: 260_0000h base + 5_8024h offset = 265_8024h



IPU_DC_RL0_CH_1 field descriptions

Field	Description
31–24 COD_NL_START_CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the new line event resides (associated with channel #1)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_NL_PRIORITY_CHAN_1	This field defines the priority of the new line event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest)

Table continues on the next page...

IPU_DC_RL0_CH_1 field descriptions (continued)

Field	Description
	1110 Reserved 1111 Reserved
15–8 COD_NF_START_CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the new frame event resides (associated with channel #1)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NF_PRIORITY_CHAN_1	This field defines the priority of the new frame event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.278 DC Routine Link Register 1 Channel 1 (IPU_DC_RL1_CH_1)

Address: 260_0000h base + 5_8028h offset = 265_8028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NFIELD_START_CHAN_1								0				COD_NFIELD_PRIORITY_CHAN_1				COD_EOF_START_CHAN_1								0				COD_EOF_PRIORITY_CHAN_1			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL1_CH_1 field descriptions

Field	Description
31–24 COD_NFIELD_START_CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the new field event resides (associated with channel #1)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_NFIELD_PRIORITY_CHAN_1	This field defines the priority of the new field event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2

Table continues on the next page...

IPU_DC_RL1_CH_1 field descriptions (continued)

Field	Description
	1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOF_ START_CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the end of frame event resides (associated with channel #1)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_EOF_ PRIORITY_ CHAN_1	This field defines the priority of the end of frame event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.279 DC Routine Link Register 2 Channel 1 (IPU_DC_RL2_CH_1)

Address: 260_0000h base + 5_8030h offset = 265_8030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_EOFIELD_START_CHAN_1								0				COD_EOFIELD_PRIORITY_CHAN_1				COD_EOL_START_CHAN_1								0				COD_EOL_PRIORITY_CHAN_1			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL2_CH_1 field descriptions

Field	Description
31–24 COD_EOFIELD_ START_CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the end of field event resides (associated with channel #1)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_EOFIELD_ PRIORITY_ CHAN_1	This field defines the priority of the end of field event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest)

Table continues on the next page...

IPU_DC_RL2_CH_1 field descriptions (continued)

Field	Description
	0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOL_ START_CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the end of line event resides (associated with channel #1)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_EOL_ PRIORITY_ CHAN_1	This field defines the priority of the end of line event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.280 DC Routine Link Register 3 Channel 1 (IPU_DC_RL3_CH_1)

Address: 260_0000h base + 5_8032h offset = 265_8032h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_CHAN_START_CHAN_1								0				COD_NEW_CHAN_PRIORITY_CHAN_1				COD_NEW_ADDR_START_CHAN_1								0				COD_NEW_ADDR_PRIORITY_CHAN_1			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL3_CH_1 field descriptions

Field	Description
31–24 COD_NEW_ CHAN_START_ CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #1)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_NEW_ CHAN_ PRIORITY_ CHAN_1	This field defines the priority of the new channel event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable)

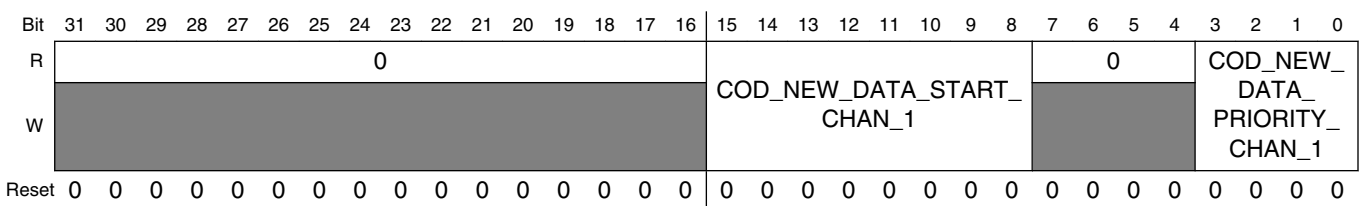
Table continues on the next page...

IPU_DC_RL3_CH_1 field descriptions (continued)

Field	Description
	0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NEW_ADDR_START_CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #1)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NEW_ADDR_PRIORITY_CHAN_1	This field defines the priority of the new address event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.281 DC Routine Link Register 4 Channel 1 (IPU_DC_RL4_CH_1)

Address: 260_0000h base + 5_8034h offset = 265_8034h



IPU_DC_RL4_CH_1 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_DATA_START_CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #1)
7–4 Reserved	This read-only field is reserved and always has the value 0.

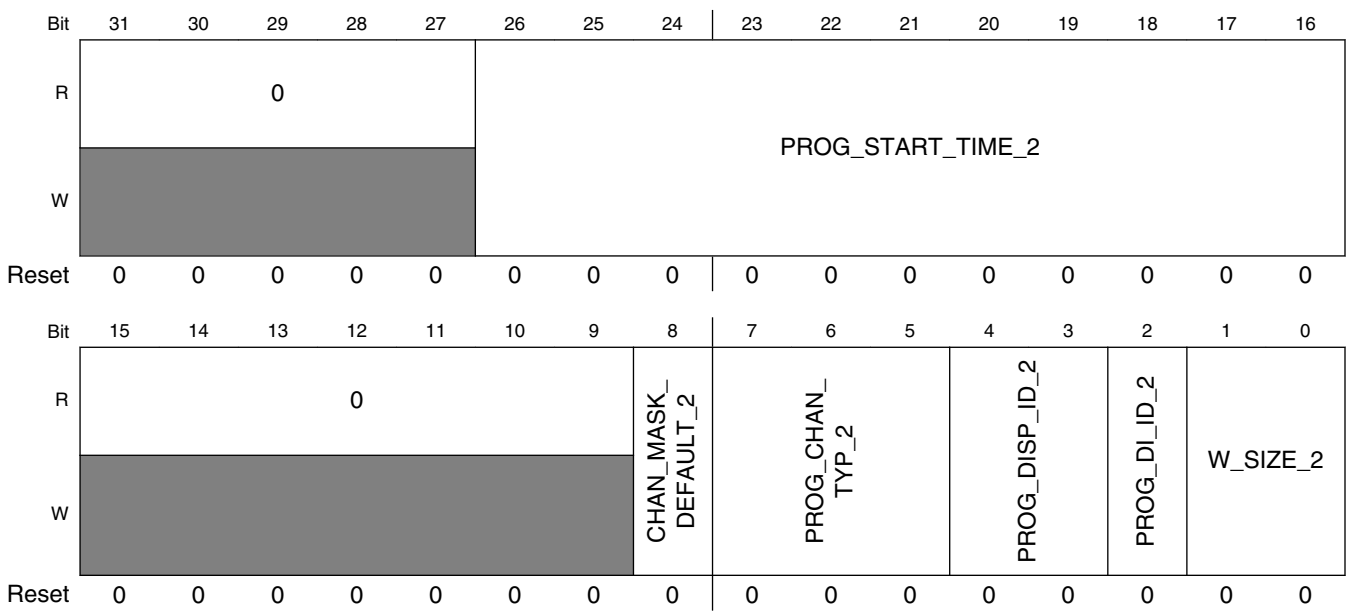
Table continues on the next page...

IPU_DC_RL4_CH_1 field descriptions (continued)

Field	Description
COD_NEW_DATA_PRIORITY_CHAN_1	This field defines the priority of the new data event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.282 DC Write Channel 2 Configuration Register (IPU_DC_WR_CH_CONF_2)

Address: 260_0000h base + 5_8038h offset = 265_8038h



IPU_DC_WR_CH_CONF_2 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26–16 PROG_START_TIME_2	This field defines the delay between display 's vertical synchronization pulse and the start time point of DC's channel 2 window. The delay is defined in pairs of rows. It is used for tearing elimination
15–9 Reserved	This read-only field is reserved and always has the value 0.

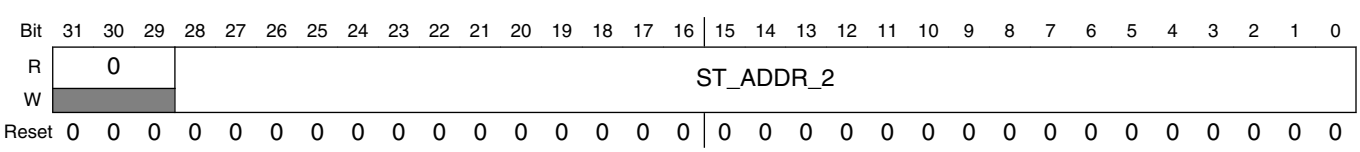
Table continues on the next page...

IPU_DC_WR_CH_CONF_2 field descriptions (continued)

Field	Description
8 CHAN_MASK_DEFAULT_2	Event mask bit for channel #2 When more then one event is used during a flow (EOF, EOL, NL, NF, EOFIELD, etc.) masks all the event besides the event that is defined as the highest priority event 1 All the events are used - no mask 0 Only the highest priority event is used, the rest are masked
7–5 PROG_CHAN_TYP_2	This field define the mode of operation of channel #2 000 Disable 001 Reserved 010 Reserved 100 Normal mode without anti-tearing. For sync display this is the only mode allowed 101 Normal mode with anti-tearing 110 Reserved 111 Additional command channel is added to the flow handled by DC channel #2
4–3 PROG_DISP_ID_2	The field defines which one of the 4 displays is associated with channel #2. 00 display #0 01 display #1 10 display #2 11 display #3
2 PROG_DI_ID_2	This bit select the DI which a transaction associated with channel #2 can be performed to 1 DI #1 0 DI #0
W_SIZE_2	Word Size The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

38.5.283 DC Write Channel 2 Address Configuration Register (IPU_DC_WR_CH_ADDR_2)

Address: 260_0000h base + 5_803Ch offset = 265_803Ch



IPU_DC_WR_CH_ADDR_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
ST_ADDR_2	This field defines the start address within the display's memory space where the write transactions will be done to for channel #2.

38.5.284 DC Routine Link Register 0 Channel 2 (IPU_DC_RL0_CH_2)

Address: 260_0000h base + 5_8040h offset = 265_8040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NL_START_CHAN_2								0				COD_NL_PRIORITY_CHAN_2				COD_NF_START_CHAN_2								0				COD_NF_PRIORITY_CHAN_2			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL0_CH_2 field descriptions

Field	Description
31–24 COD_NL_START_CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the new line event resides (associated with channel #2)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_NL_PRIORITY_CHAN_2	This field defines the priority of the new line event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NF_START_CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the new frame event resides (associated with channel #2)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NF_PRIORITY_CHAN_2	This field defines the priority of the new frame event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest)

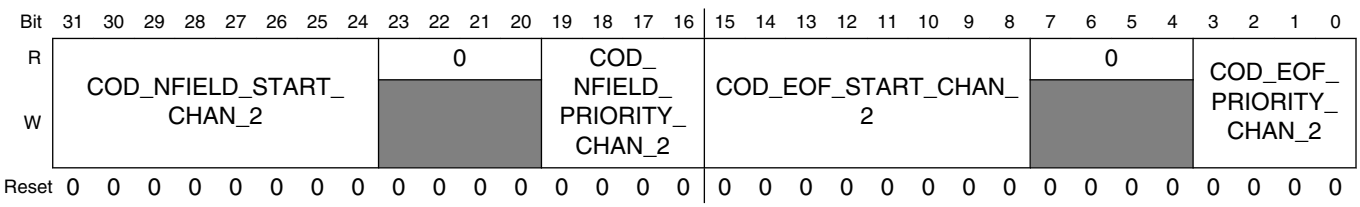
Table continues on the next page...

IPU_DC_RL0_CH_2 field descriptions (continued)

Field	Description
	0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.285 DC Routine Link Register 1 Channel 2 (IPU_DC_RL1_CH_2)

Address: 260_0000h base + 5_8044h offset = 265_8044h



IPU_DC_RL1_CH_2 field descriptions

Field	Description
31–24 COD_NFIELD_START_CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the new field event resides (associated with channel #2)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_NFIELD_PRIORITY_CHAN_2	This field defines the priority of the new field event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOF_START_CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the end of frame event resides (associated with channel #2)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_EOF_PRIORITY_CHAN_2	This field defines the priority of the end of frame event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable

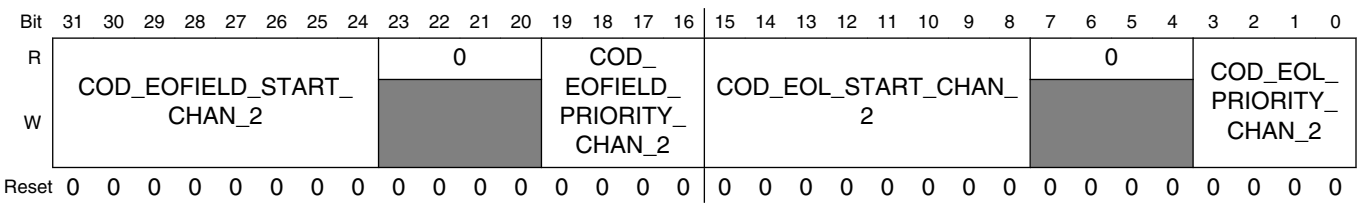
Table continues on the next page...

IPU_DC_RL1_CH_2 field descriptions (continued)

Field	Description
	0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.286 DC Routine Link Register 2 Channel 2 (IPU_DC_RL2_CH_2)

Address: 260_0000h base + 5_8048h offset = 265_8048h



IPU_DC_RL2_CH_2 field descriptions

Field	Description
31–24 COD_EOFIELD_START_CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the end of field event resides (associated with channel #2)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_EOFIELD_PRIORITY_CHAN_2	This field defines the priority of the end of field event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOL_START_CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the end of line event resides (associated with channel #2)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_EOL_PRIORITY_CHAN_2	This field defines the priority of the end of line event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable)

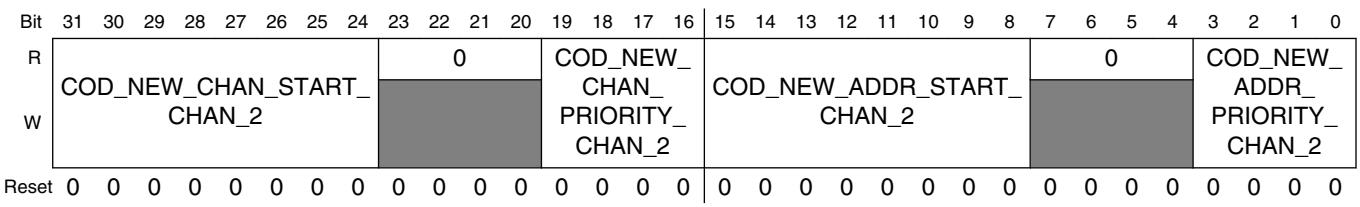
Table continues on the next page...

IPU_DC_RL2_CH_2 field descriptions (continued)

Field	Description
	0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.287 DC Routine Link Register 3 Channel 2 (IPU_DC_RL3_CH_2)

Address: 260_0000h base + 5_804Ch offset = 265_804Ch



IPU_DC_RL3_CH_2 field descriptions

Field	Description
31–24 COD_NEW_CHAN_START_CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #2)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_NEW_CHAN_PRIORITY_CHAN_2	This field defines the priority of the end of line event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NEW_ADDR_START_CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #2)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NEW_ADDR_	This field defines the priority of the end of line event (associated with channel #2)

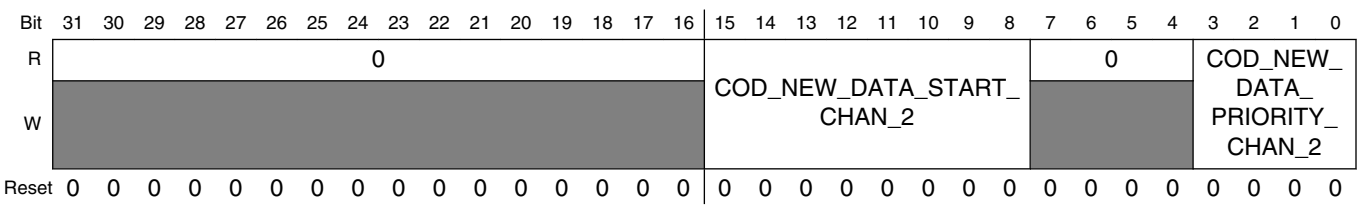
Table continues on the next page...

IPU_DC_RL3_CH_2 field descriptions (continued)

Field	Description
PRIORITY_ CHAN_2	The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.288 DC Routine Link Register 4 Channel 2 (IPU_DC_RL4_CH_2)

Address: 260_0000h base + 5_8050h offset = 265_8050h

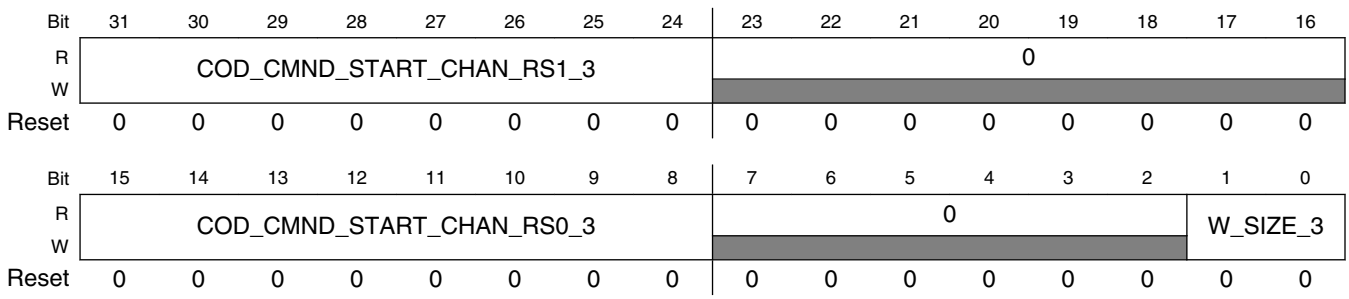


IPU_DC_RL4_CH_2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_ DATA_START_ CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #2)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NEW_ DATA_ PRIORITY_ CHAN_2	This field defines the priority of the end of line event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.289 DC Command Channel 3 Configuration Register (IPU_DC_CMD_CH_CONF_3)

Address: 260_0000h base + 5_8054h offset = 265_8054h

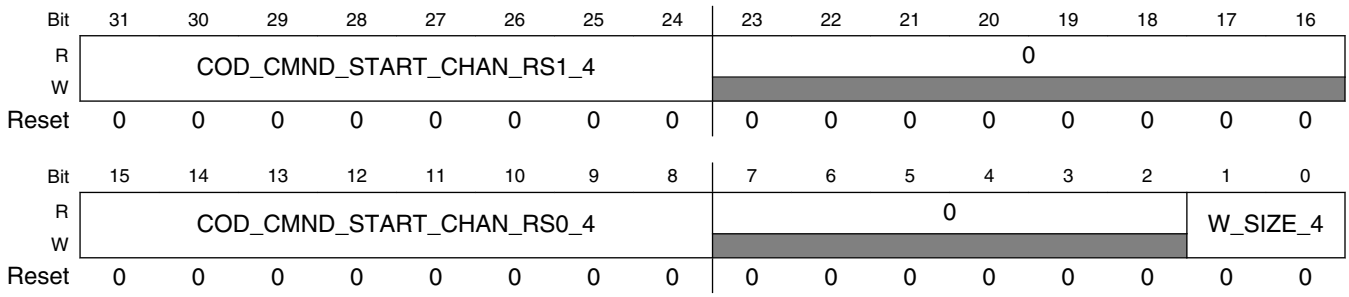


IPU_DC_CMD_CH_CONF_3 field descriptions

Field	Description
31–24 COD_CMND_START_CHAN_RS1_3	This field is a pointer to the address within the microcode memory where the routine that handles the command start event resides (associated with channel #3); This field is relevant when RS is equal to 1
23–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_CMND_START_CHAN_RS0_3	This field is a pointer to the address within the microcode memory where the routine that handles the command start event resides (associated with channel #3); This field is relevant when RS is equal to 0
7–2 Reserved	This read-only field is reserved and always has the value 0.
W_SIZE_3	Word Size associated with channel #3 The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

38.5.290 DC Command Channel 4 Configuration Register (IPU_DC_CMD_CH_CONF_4)

Address: 260_0000h base + 5_8058h offset = 265_8058h

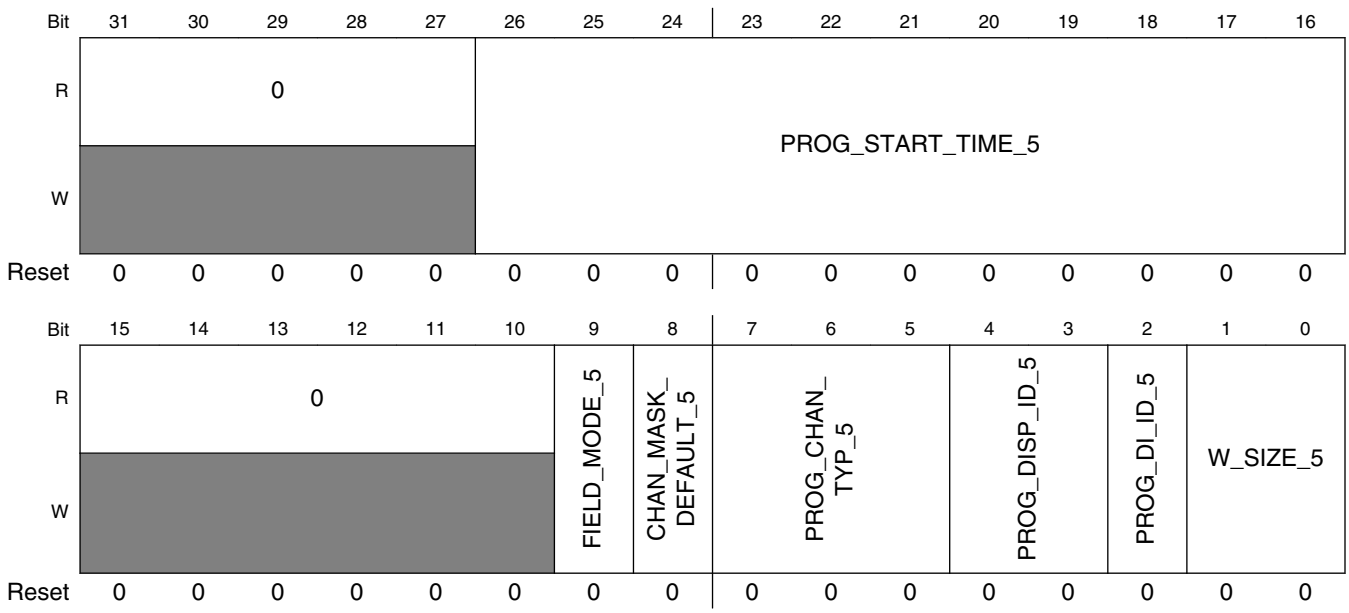


IPU_DC_CMD_CH_CONF_4 field descriptions

Field	Description
31–24 COD_CMND_START_CHAN_RS1_4	This field is a pointer to the address within the microcode memory where the routine that handles the command start event resides (associated with channel #4); This field is relevant when RS is equal to 1
23–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_CMND_START_CHAN_RS0_4	This field is a pointer to the address within the microcode memory where the routine that handles the command start event resides (associated with channel #4); This field is relevant when RS is equal to 0
7–2 Reserved	This read-only field is reserved and always has the value 0.
W_SIZE_4	Word Size associated with channel #4 The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

38.5.291 DC Write Channel 5Configuration Register (IPU_DC_WR_CH_CONF_5)

Address: 260_0000h base + 5_805Ch offset = 265_805Ch



IPU_DC_WR_CH_CONF_5 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26–16 PROG_START_TIME_5	This field defines the delay between display 's vertical synchronization pulse and the start time point of DC's channel 5 window. The delay is defined in pairs of rows. It is used for tearing elimination
15–10 Reserved	This read-only field is reserved and always has the value 0.
9 FIELD_MODE_5	Field mode bit for channel #5 This bit defines if the channel works in field mode or frame mode; This bit is relevant if the flow is sync flow 1 Field mode 0 Frame mode
8 CHAN_MASK_DEFAULT_5	Event mask bit for channel #5 When more then one event is used during a flow (EOF, EOL, NL, NF, EOFIELD, etc.) masks all the event besides the event that is defined as the highest priority event 1 All the events are used - no mask 0 Only the highest priority event is used, the rest are masked
7–5 PROG_CHAN_TYP_5	This field define the mode of operation of channel #5 000 Disable 001 Reserved

Table continues on the next page...

IPU_DC_WR_CH_CONF_5 field descriptions (continued)

Field	Description
	010 Reserved 100 Normal mode without anti-tearing. For sync display this is the only mode allowed 101 Normal mode with anti-tearing 110 Reserved 111 Additional command channel is added to the flow handled by DC channel #5
4–3 PROG_DISP_ID_5	The field defines which one of the 4 displays is associated with channel #5. 00 display #0 01 display #1 10 display #2 11 display #3
2 PROG_DI_ID_5	This bit select the DI which a transaction associated with channel #5 can be performed to. When channel 28 is connected to DI0, channel 23 must be connected to DI1 even if ch23 is not used. This is done by writing 1 to this bit. 1 DI #1 0 DI #0
W_SIZE_5	Word Size associated with channel #5 The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

38.5.292 DC Write Channel 5Address Configuration Register (IPU_DC_WR_CH_ADDR_5)

Address: 260_0000h base + 5_8060h offset = 265_8060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																ST_ADDR_5															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_WR_CH_ADDR_5 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
ST_ADDR_5	This field defines the start address within the display's memory space where the write transactions will be done to for channel #5.

38.5.293 DC Routine Link Register 0 Channel 5 (IPU_DC_RL0_CH_5)

Address: 260_0000h base + 5_8064h offset = 265_8064h

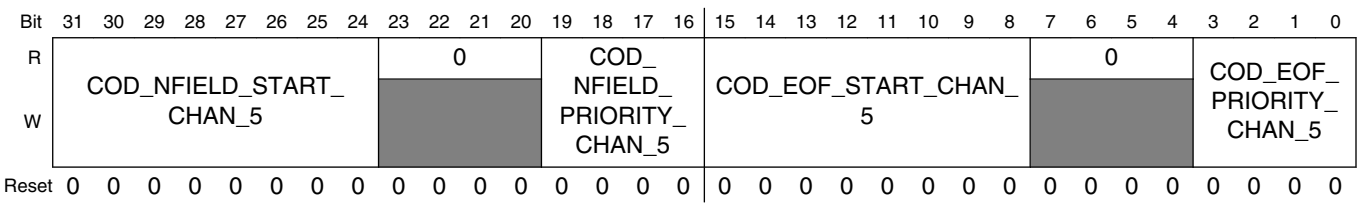
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NL_START_CHAN_5								0				COD_NL_PRIORITY_CHAN_5				COD_NF_START_CHAN_5								0				COD_NF_PRIORITY_CHAN_5			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL0_CH_5 field descriptions

Field	Description
31–24 COD_NL_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the new line event resides (associated with channel #5)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_NL_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NF_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the new frame event resides (associated with channel #5)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NF_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.294 DC Routine Link Register 1 Channel 5 (IPU_DC_RL1_CH_5)

Address: 260_0000h base + 5_8068h offset = 265_8068h

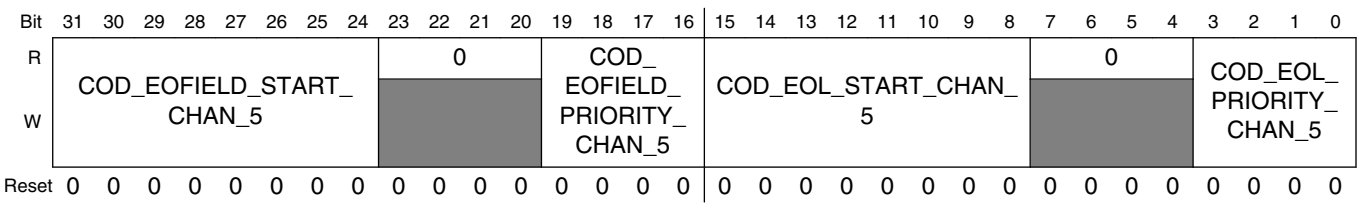


IPU_DC_RL1_CH_5 field descriptions

Field	Description
31–24 COD_NFIELD_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the new field event resides (associated with channel #5)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_NFIELD_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOF_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the end of frame event resides (associated with channel #5)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_EOF_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.295 DC Routine Link Register 2 Channel 5 (IPU_DC_RL2_CH_5)

Address: 260_0000h base + 5_806Ch offset = 265_806Ch

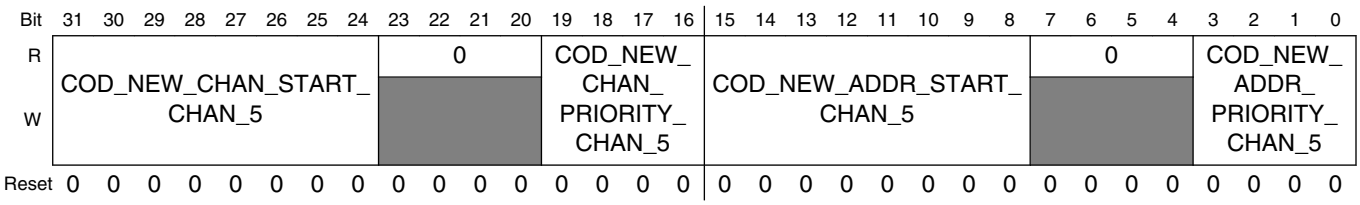


IPU_DC_RL2_CH_5 field descriptions

Field	Description
31–24 COD_EOFIELD_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the end of field event resides (associated with channel #5)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_EOFIELD_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOL_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the end of line event resides (associated with channel #5)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_EOL_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.296 DC Routine Link Register3 Channel 5 (IPU_DC_RL3_CH_5)

Address: 260_0000h base + 5_8070h offset = 265_8070h

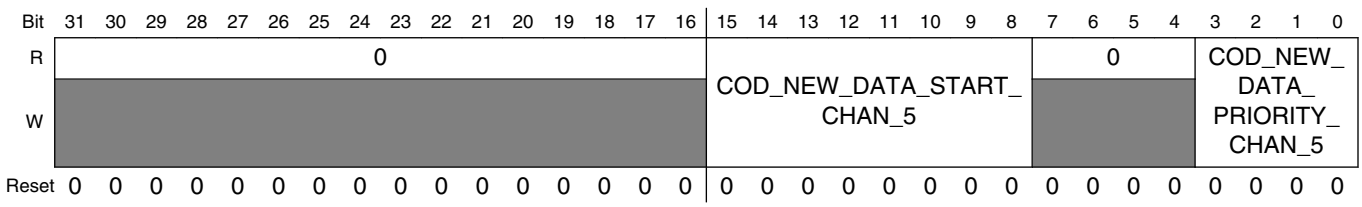


IPU_DC_RL3_CH_5 field descriptions

Field	Description
31–24 COD_NEW_CHAN_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #5)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_NEW_CHAN_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NEW_ADDR_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #5)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NEW_ADDR_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.297 DC Routine Link Register 4 Channel 5 (IPU_DC_RL4_CH_5)

Address: 260_0000h base + 5_8074h offset = 265_8074h

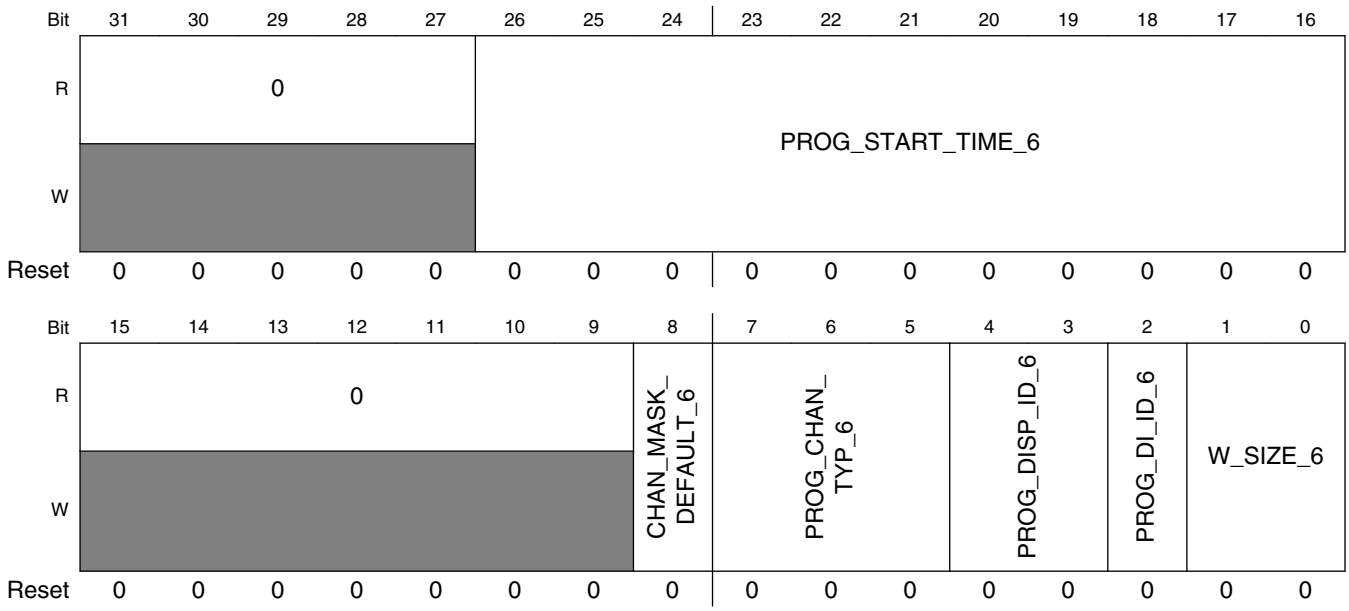


IPU_DC_RL4_CH_5 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_DATA_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #5)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NEW_DATA_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.298 DC Write Channel 6 Configuration Register (IPU_DC_WR_CH_CONF_6)

Address: 260_0000h base + 5_8078h offset = 265_8078h



IPU_DC_WR_CH_CONF_6 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26–16 PROG_START_TIME_6	This field defines the delay between display 's vertical synchronization pulse and the start time point of DC's channel 6 window. The delay is defined in pairs of rows. It is used for tearing elimination
15–9 Reserved	This read-only field is reserved and always has the value 0.
8 CHAN_MASK_DEFAULT_6	Event mask bit for channel #6 When more then one event is used during a flow (EOF, EOL, NL, NF, EOFIELD, etc.) masks all the event besides the event that is defined as the highest priority event 1 All the events are used - no mask 0 Only the highest priority event is used, the rest are masked
7–5 PROG_CHAN_TYP_6	This field define the mode of operation of channel #6 000 Disable 001 Reserved 010 Reserved 100 Normal mode without anti-tearing. For sync display this is the only mode allowed 101 Normal mode with anti-tearing 110 Reserved 111 Additional command channel is added to the flow handled by DC channel #6

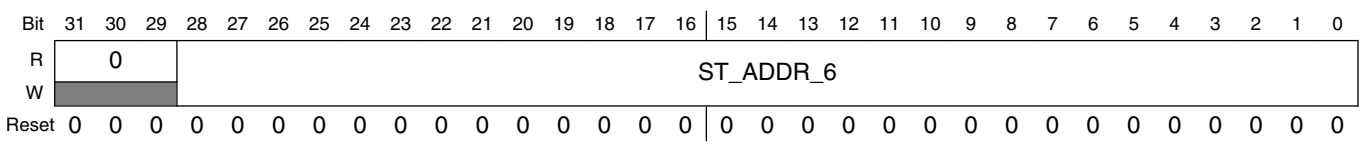
Table continues on the next page...

IPU_DC_WR_CH_CONF_6 field descriptions (continued)

Field	Description
4–3 PROG_DISP_ID_6	The field defines which one of the 4 displays is associated with channel #6. 00 display #0 01 display #1 10 display #2 11 display #3
2 PROG_DI_ID_6	This bit select the DI which a transaction associated with channel #6 can be performed to 1 DI #1 0 DI #0
W_SIZE_6	Word Size associated with channel #6 The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

38.5.299 DC Write Channel 6 Address Configuration Register (IPU_DC_WR_CH_ADDR_6)

Address: 260_0000h base + 5_807Ch offset = 265_807Ch



IPU_DC_WR_CH_ADDR_6 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
ST_ADDR_6	This field defines the start address within the display's memory space where the write transactions will be done to for channel #6.

38.5.300 DC Routine Link Register 0Channel 6 (IPU_DC_RL0_CH_6)

Address: 260_0000h base + 5_8080h offset = 265_8080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NL_START_CHAN_6								0				COD_NL_PRIORITY_CHAN_6				COD_NF_START_CHAN_6								0				COD_NF_PRIORITY_CHAN_6			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL0_CH_6 field descriptions

Field	Description
31–24 COD_NL_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the new line event resides (associated with channel #6)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_NL_PRIORITY_CHAN_6	This field defines the priority of the new line event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NF_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the new frame event resides (associated with channel #6)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NF_PRIORITY_CHAN_6	This field defines the priority of the new frame event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.301 DC Routine Link Register 1 Channel 6 (IPU_DC_RL1_CH_6)

Address: 260_0000h base + 5_8084h offset = 265_8084h

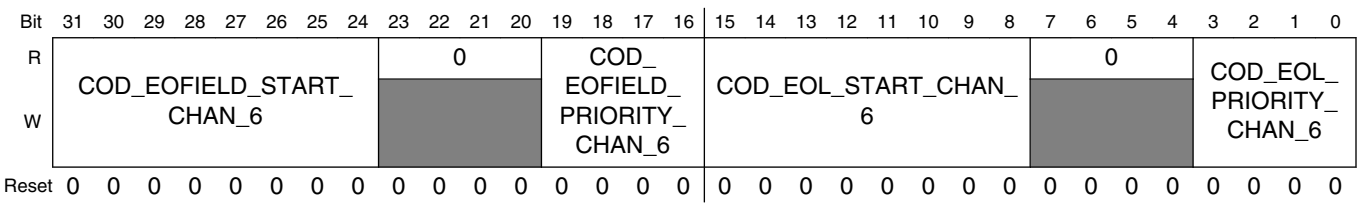
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NFIELD_START_CHAN_6								0				COD_NFIELD_PRIORITY_CHAN_6				COD_EOF_START_CHAN_6								0				COD_EOF_PRIORITY_CHAN_6			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL1_CH_6 field descriptions

Field	Description
31–24 COD_NFIELD_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the new field event resides (associated with channel #6)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_NFIELD_PRIORITY_CHAN_6	This field defines the priority of the new field event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOF_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the end of frame event resides (associated with channel #6)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_EOF_PRIORITY_CHAN_6	This field defines the priority of the new field event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.302 DC Routine Link Register 2 Channel 6 (IPU_DC_RL2_CH_6)

Address: 260_0000h base + 5_8088h offset = 265_8088h



IPU_DC_RL2_CH_6 field descriptions

Field	Description
31–24 COD_EOFIELD_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the end of field event resides (associated with channel #6)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_EOFIELD_PRIORITY_CHAN_6	This field defines the priority of the new field event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOL_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the end of line event resides (associated with channel #6)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_EOL_PRIORITY_CHAN_6	This field defines the priority of the new field event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.303 DC Routine Link Register 3 Channel 6 (IPU_DC_RL3_CH_6)

Address: 260_0000h base + 5_808Ch offset = 265_808Ch

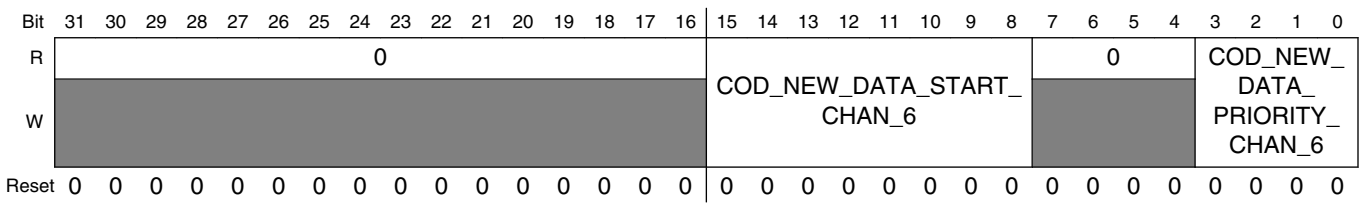
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_CHAN_START_CHAN_6								0				COD_NEW_CHAN_PRIORITY_CHAN_6				COD_NEW_ADDR_START_CHAN_6								0				COD_NEW_ADDR_PRIORITY_CHAN_6			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL3_CH_6 field descriptions

Field	Description
31–24 COD_NEW_CHAN_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #6)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 COD_NEW_CHAN_PRIORITY_CHAN_6	This field defines the priority of the new field event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NEW_ADDR_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #6)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NEW_ADDR_PRIORITY_CHAN_6	This field defines the priority of the new field event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.304 DC Routine Link Register 4 Channel 6 (IPU_DC_RL4_CH_6)

Address: 260_0000h base + 5_8090h offset = 265_8090h

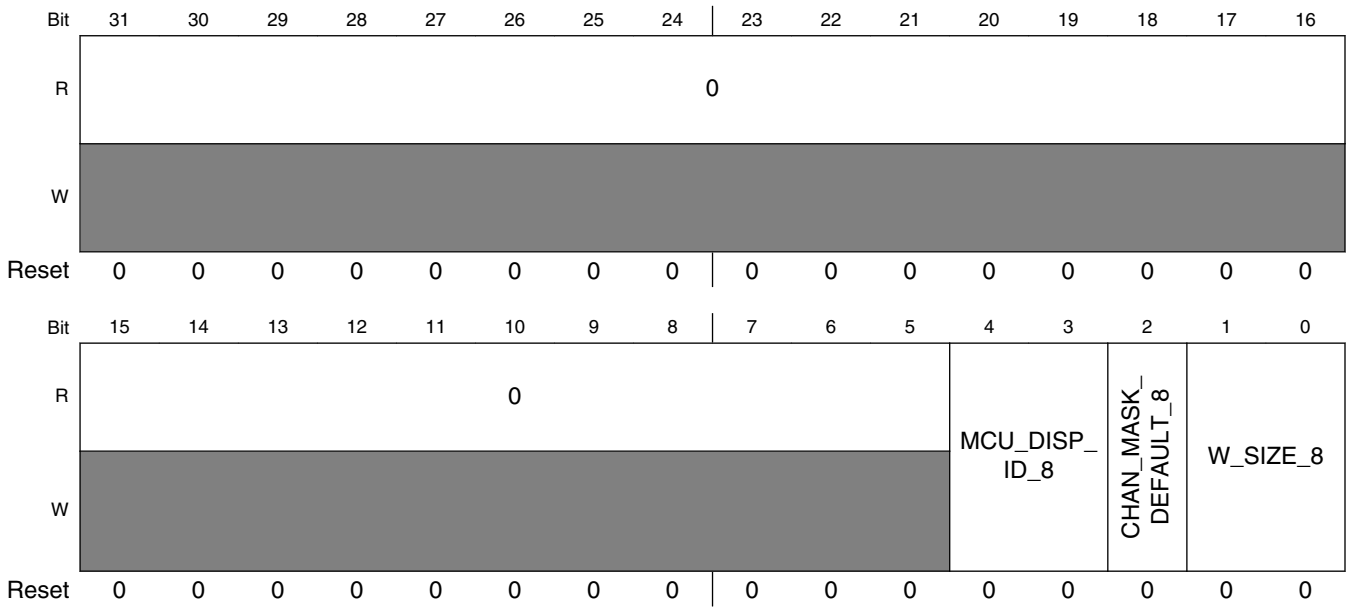


IPU_DC_RL4_CH_6 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_DATA_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #6)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NEW_DATA_PRIORITY_CHAN_6	This field defines the priority of the new field event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.305 DC Write Channel 8 Configuration 1Register (IPU_DC_WR_CH_CONF1_8)

Address: 260_0000h base + 5_8094h offset = 265_8094h

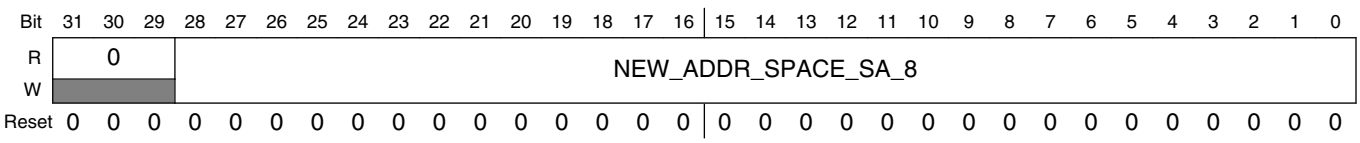


IPU_DC_WR_CH_CONF1_8 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4–3 MCU_DISP_ID_8	The field defines which one of the 4 displays is associated with channel #8. 00 display #0 01 display #1 10 display #2 11 display #3
2 CHAN_MASK_DEFAULT_8	Event mask bit for channel #8 When more then one event is used during a flow (EOF, EOL, NL, NF, EOFIELD, etc.) masks all the event besides the event that is defined as the highest priority event 1 All the events are used - no mask 0 Only the highest priority event is used, the rest are masked
W_SIZE_8	Word Size associated with channel #8 The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

38.5.306 DC Write Channel 8 Configuration 2 Register (IPU_DC_WR_CH_CONF2_8)

Address: 260_0000h base + 5_8098h offset = 265_8098h

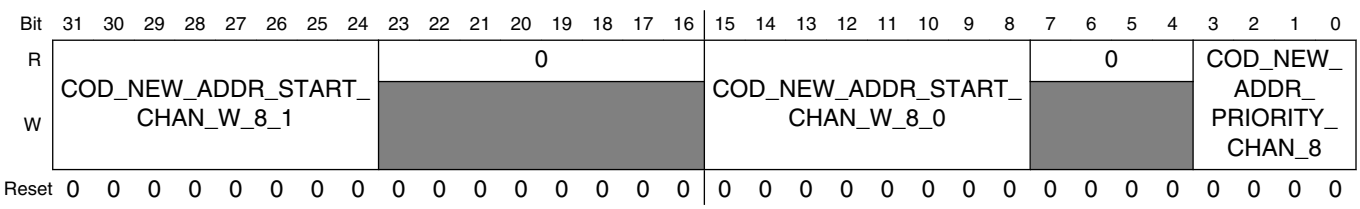


IPU_DC_WR_CH_CONF2_8 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
NEW_ADDR_ SPACE_SA_8	Channel #8 is used for ARM platform direct access to the display. This field defines the base address of the second region accessible on the display

38.5.307 DC Routine Link Register 1 Channel 8 (IPU_DC_RL1_CH_8)

Address: 260_0000h base + 5_809Ch offset = 265_809Ch



IPU_DC_RL1_CH_8 field descriptions

Field	Description
31–24 COD_NEW_ ADDR_START_ CHAN_W_8_1	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #8, second region)
23–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_ ADDR_START_ CHAN_W_8_0	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #8, first region)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NEW_ ADDR_	This field defines the priority of the new address event (associated with channel #8, both regions)

Table continues on the next page...

IPU_DC_RL1_CH_8 field descriptions (continued)

Field	Description
PRIORITY_ CHAN_8	The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.308 DC Routine Link Register 2 Channel 8 (IPU_DC_RL2_CH_8)

Address: 260_0000h base + 5_80A0h offset = 265_80A0h

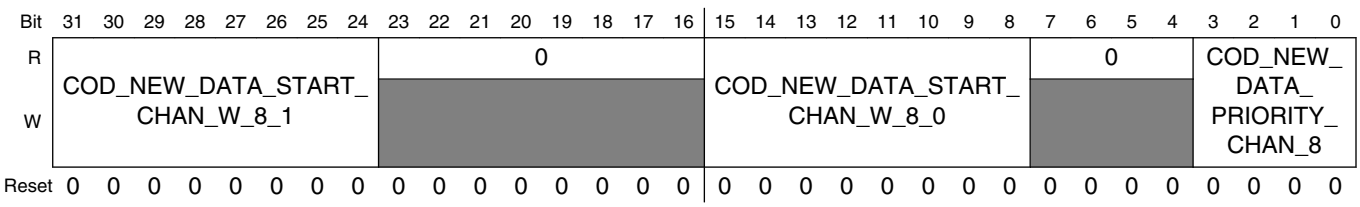
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_CHAN_START_ CHAN_W_8_1								0								COD_NEW_CHAN_START_ CHAN_W_8_0								0				COD_NEW_ CHAN_ PRIORITY_ CHAN_8			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL2_CH_8 field descriptions

Field	Description
31–24 COD_NEW_ CHAN_START_ CHAN_W_8_1	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #8, second region)
23–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_ CHAN_START_ CHAN_W_8_0	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #8, first region)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NEW_ CHAN_ PRIORITY_ CHAN_8	This field defines the priority of the new address event (associated with channel #8, both regions) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.309 DC Routine Link Register 3 Channel 8 (IPU_DC_RL3_CH_8)

Address: 260_0000h base + 5_80A4h offset = 265_80A4h

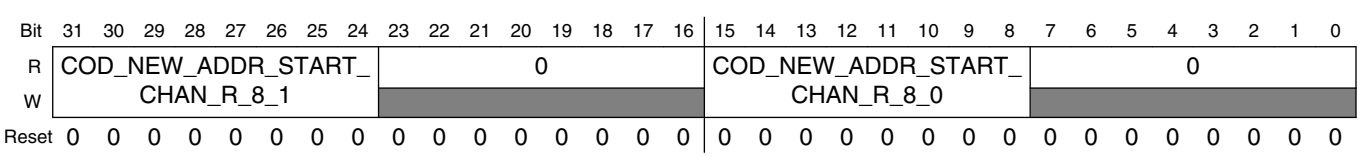


IPU_DC_RL3_CH_8 field descriptions

Field	Description
31–24 COD_NEW_DATA_START_CHAN_W_8_1	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #8,second region)
23–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_DATA_START_CHAN_W_8_0	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #8, first region)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NEW_DATA_PRIORITY_CHAN_8	This field defines the priority of the new address event (associated with channel #8, both regions) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.310 DC Routine Link Register 4 Channel 8 (IPU_DC_RL4_CH_8)

Address: 260_0000h base + 5_80A8h offset = 265_80A8h



IPU_DC_RL4_CH_8 field descriptions

Field	Description
31–24 COD_NEW_ ADDR_START_ CHAN_R_8_1	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #8, second region)
23–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_ ADDR_START_ CHAN_R_8_0	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #8, first region)
Reserved	This read-only field is reserved and always has the value 0.

38.5.311 DC Routine Link Register 5 Channel 8 (IPU_DC_RL5_CH_8)

Address: 260_0000h base + 5_80ACh offset = 265_80ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_CHAN_START_CHAN_R_8_1								0								COD_NEW_CHAN_START_CHAN_R_8_0								0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL5_CH_8 field descriptions

Field	Description
31–24 COD_NEW_ CHAN_START_ CHAN_R_8_1	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #8, second region)
23–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_ CHAN_START_ CHAN_R_8_0	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #8, first region)
Reserved	This read-only field is reserved and always has the value 0.

38.5.312 DC Routine Link Register 6 Channel 8 (IPU_DC_RL6_CH_8)

Address: 260_0000h base + 5_80B0h offset = 265_80B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_DATA_START_CHAN_R_8_1								0								COD_NEW_DATA_START_CHAN_R_8_0								0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL6_CH_8 field descriptions

Field	Description
31–24 COD_NEW_DATA_START_CHAN_R_8_1	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #8, second region)
23–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_DATA_START_CHAN_R_8_0	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #8, first region)
Reserved	This read-only field is reserved and always has the value 0.

38.5.313 DC Write Channel 9 Configuration 1Register (IPU_DC_WR_CH_CONF1_9)

Address: 260_0000h base + 5_80B4h offset = 265_80B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											MCU_DISP_ID_9		CHAN_MASK_DEFAULT_9	W_SIZE_9	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_WR_CH_CONF1_9 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4–3 MCU_DISP_ID_9	The field defines which one of the 4 displays is associated with channel #9. 00 display #0 01 display #1 10 display #2 11 display #3
2 CHAN_MASK_DEFAULT_9	Event mask bit for channel #9 When more then one event is used during a flow (EOF, EOL, NL, NF, EOFIELD, etc.) masks all the event besides the event that is defined as the highest priority event 1 All the events are used - no mask 0 Only the highest priority event is used, the rest are masked
W_SIZE_9	Word Size associated with channel #9 The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

38.5.314 DC Write Channel 9Configuration 2Register (IPU_DC_WR_CH_CONF2_9)

Address: 260_0000h base + 5_80B8h offset = 265_80B8h

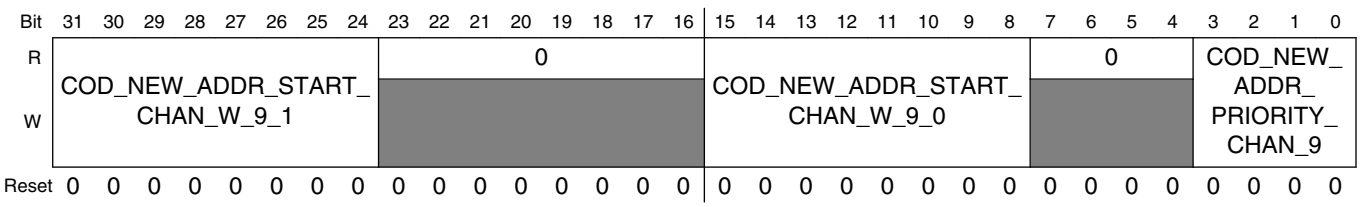
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																NEW_ADDR_SPACE_SA_9															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_WR_CH_CONF2_9 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
NEW_ADDR_SPACE_SA_9	Channel #8 is used for ARM platform direct access to the display. This field defines the base address of the second region accessible on the display

38.5.315 DC Routine Link Register 1 Channel 9 (IPU_DC_RL1_CH_9)

Address: 260_0000h base + 5_80BCh offset = 265_80BCh

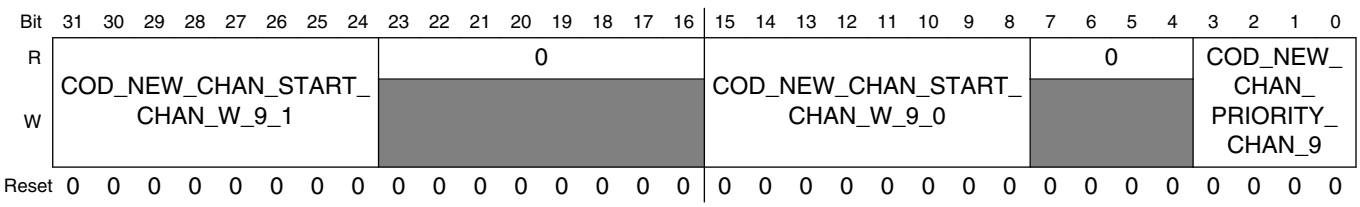


IPU_DC_RL1_CH_9 field descriptions

Field	Description
31–24 COD_NEW_ADDR_START_CHAN_W_9_1	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #9, second region)
23–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_ADDR_START_CHAN_W_9_0	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #9, first region)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NEW_ADDR_PRIORITY_CHAN_9	This field defines the priority of the new address event (associated with channel #9, both regions) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.316 DC Routine Link Register 2 Channel 9 (IPU_DC_RL2_CH_9)

Address: 260_0000h base + 5_80C0h offset = 265_80C0h



IPU_DC_RL2_CH_9 field descriptions

Field	Description
31–24 COD_NEW_CHAN_START_CHAN_W_9_1	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #9, second region)
23–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_CHAN_START_CHAN_W_9_0	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #9, first region)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NEW_CHAN_PRIORITY_CHAN_9	This field defines the priority of the new address event (associated with channel #9, both regions) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.317 DC Routine Link Register 3Channel 9 (IPU_DC_RL3_CH_9)

Address: 260_0000h base + 5_80C4h offset = 265_80C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_DATA_START_CHAN_W_9_1								0								COD_NEW_DATA_START_CHAN_W_9_0								0				COD_NEW_DATA_PRIORITY_CHAN_9			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL3_CH_9 field descriptions

Field	Description
31–24 COD_NEW_DATA_START_CHAN_W_9_1	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #9, second region)
23–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_DATA_START_CHAN_W_9_0	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #9, first region)
7–4 Reserved	This read-only field is reserved and always has the value 0.
COD_NEW_DATA_PRIORITY_CHAN_9	This field defines the priority of the new address event (associated with channel #9, both regions) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

38.5.318 DC Routine Link Register 4 Channel 9 (IPU_DC_RL4_CH_9)

Address: 260_0000h base + 5_80C8h offset = 265_80C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_ADDR_START_CHAN_R_9_1								0								COD_NEW_ADDR_START_CHAN_R_9_0								0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL4_CH_9 field descriptions

Field	Description
31–24 COD_NEW_ ADDR_START_ CHAN_R_9_1	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #9, second region)
23–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_ ADDR_START_ CHAN_R_9_0	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #9, first region)
Reserved	This read-only field is reserved and always has the value 0.

38.5.319 DC Routine Link Register 5 Channel 9 (IPU_DC_RL5_CH_9)

Address: 260_0000h base + 5_80CCh offset = 265_80CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_CHAN_START_CHAN_R_9_1								0								COD_NEW_CHAN_START_CHAN_R_9_0								0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL5_CH_9 field descriptions

Field	Description
31–24 COD_NEW_ CHAN_START_ CHAN_R_9_1	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #9, second region)
23–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_ CHAN_START_ CHAN_R_9_0	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #9, first region)
Reserved	This read-only field is reserved and always has the value 0.

38.5.320 DC Routine Link Register 6 Channel 9 (IPU_DC_RL6_CH_9)

Address: 260_0000h base + 5_80D0h offset = 265_80D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_DATA_START_CHAN_R_9_1								0								COD_NEW_DATA_START_CHAN_R_9_0								0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL6_CH_9 field descriptions

Field	Description
31–24 COD_NEW_DATA_START_CHAN_R_9_1	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #9, second region)
23–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 COD_NEW_DATA_START_CHAN_R_9_0	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #9, first region)
Reserved	This read-only field is reserved and always has the value 0.

38.5.321 DC General Register (IPU_DC_GEN)

Address: 260_0000h base + 5_80D4h offset = 265_80D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							DC_BK_EN	DC_BKDIV							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							DC_CH5_TYPE	SYNC_PRIORITY_1	SYNC_PRIORITY_5	MASK4CHAN_5	MASK_EN	0	SYNC_1_6		0
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IPU_DC_GEN field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24 DC_BK_EN	Cursor blinking enable 1 blinking is enabled 0 blinking is disabled
23–16 DC_BKDIV	Blinking Rate This field defines the blinking rate. The blinking occurs every N-th frame While N is defined by DC_BKDIV
15–9 Reserved	This read-only field is reserved and always has the value 0.
8 DC_CH5_TYPE	Channel 5 is used for synchronous flow. When this channel is used for accessing asynchronous display that is activated in a synchronous way 1 Enable the asynchronous interface via channel 5 0 normal mode, synchronous flow via channel 5
7 SYNC_ PRIORITY_1	When 2 sync flows are running, this bit sets the priority of channel #1. both SYNC_PRIORITY_5 and SYNC_PRIORITY_1 should not have the value of 0 This bit should be 1 high Priority 0 low Priority
6 SYNC_ PRIORITY_5	When 2 sync flows are running, this bit sets the priority of channel #5. both SYNC_PRIORITY_5 and SYNC_PRIORITY_1 should not have the value of 0 This bit should be 1 high priority 0 low Priority
5 MASK4CHAN_5	Sync flow can be associated with a mask channel. Only one sync flow can have a mask. This bit is ignored if MASK_EN is clear 1 mask channel is associated to the sync flow via DP 0 mask channel is associated to the sync flow via DC (without DP)
4 MASK_EN	Enable of the mask channel 1 mask channel is enabled 0 mask channel is disabled
3 Reserved	This read-only field is reserved and always has the value 0.
2–1 SYNC_1_6	This field 00 Channel 1 of the DC handles async flow 01 Illegal 10 Channel 1 of the DC handles sync flow 11 illegal

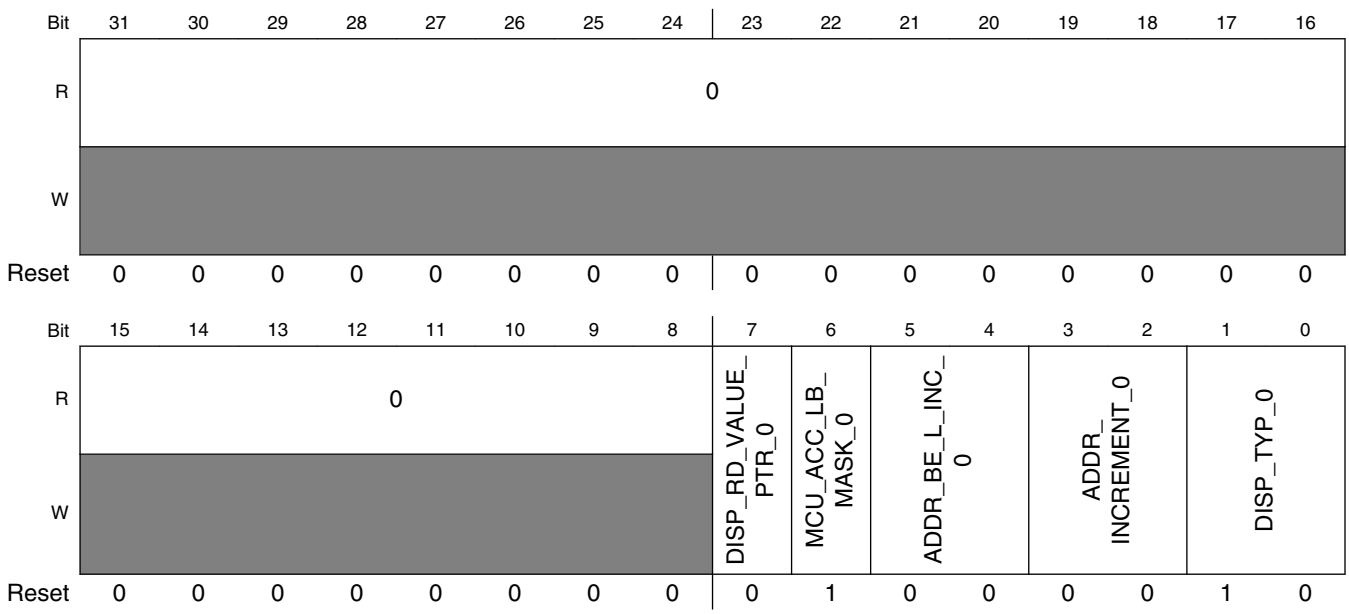
Table continues on the next page...

IPU_DC_GEN field descriptions (continued)

Field	Description
0 Reserved	This read-only field is reserved and always has the value 0.

38.5.322 DC Display Configuration 1 Register 0 (IPU_DC_DISP_CONF1_0)

Address: 260_0000h base + 5_80D8h offset = 265_80D8h



IPU_DC_DISP_CONF1_0 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.
7 DISP_RD_ VALUE_PTR_0	When the display works in wait for status mode. The IPU polls the display and compare the value to the value stored on the status DI_READ_DATA_ACK_VALUE and mask it with DI_READ_DATA_MASK. The DC holds 2 sets of mask and value. This field holds a pointer to the corresponding set of mask and value. 1 DI_READ_DATA_ACK_VALUE_1 & DI_READ_DATA_MASK_1 are used for display 0 0 DI_READ_DATA_ACK_VALUE_0 & DI_READ_DATA_MASK_0 are used for display 0
6 MCU_ACC_LB_ MASK_0	The DC compares between the current access to the a calculated address. The calculated address is the next consecutive address following the last address. This bit defines the comparing mode 1 The 2 addresses are fully compared 0 The 2 addresses are compared, but the ADDR[0] bit of the new address is ignored
5–4 ADDR_BE_L_ INC_0	This bits define the increment mode when the latest access was done with some of the byte enable signals are low, in that case different increment should be done instead of the normal auto increment of the address

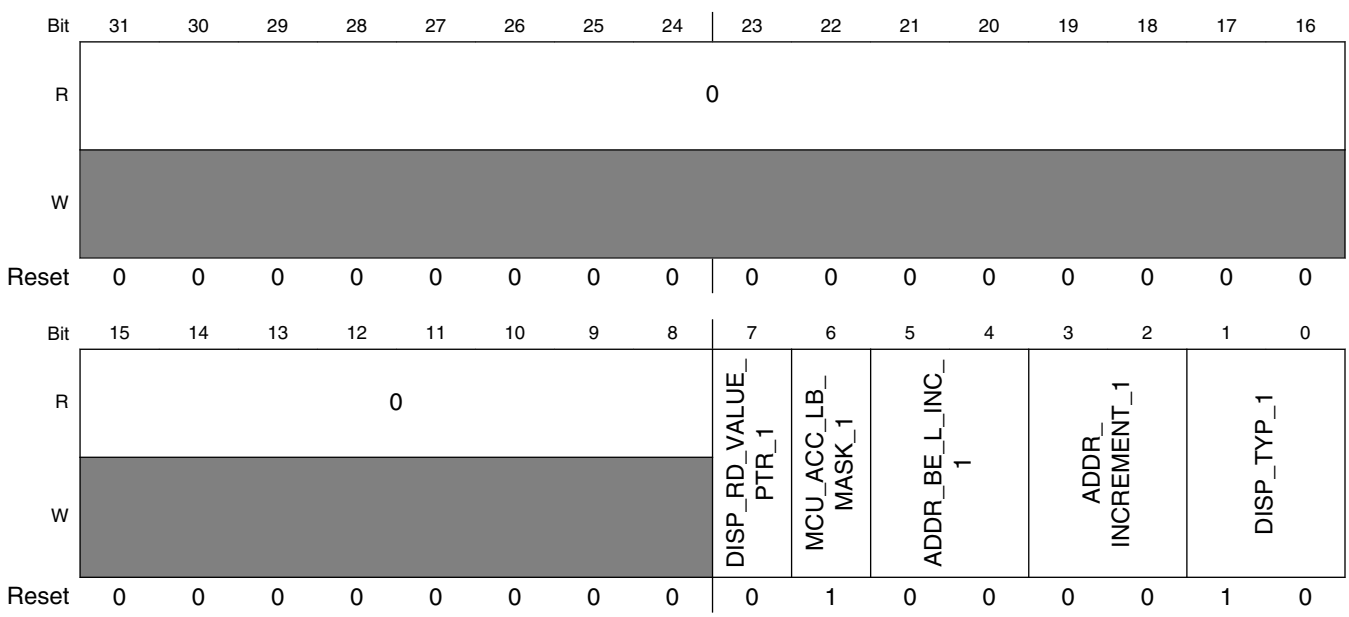
Table continues on the next page...

IPU_DC_DISP_CONF1_0 field descriptions (continued)

Field	Description
	IF MCU_ACC_LB_MASK_0 is 0 then only 00 and 10 values are allowed. 00 No increment 01 Increment by 1 10 Increment by 2 11 Increment by 3
3–2 ADDR_ INCREMENT_0	This field is the increment step for auto increment mode 00 Increment the address by 1 01 Increment the address by 2 10 Increment the address by 3 11 Increment the address by 4
DISP_TYP_0	This field defines the type of the display 00 Serial accesses display 01 Reserved 10 parallel display, without byte_enable support 11 parallel display, with byte_enable support

38.5.323 DC Display Configuration 1 Register 1 (IPU_DC_DISP_CONF1_1)

Address: 260_0000h base + 5_80DCh offset = 265_80DCh

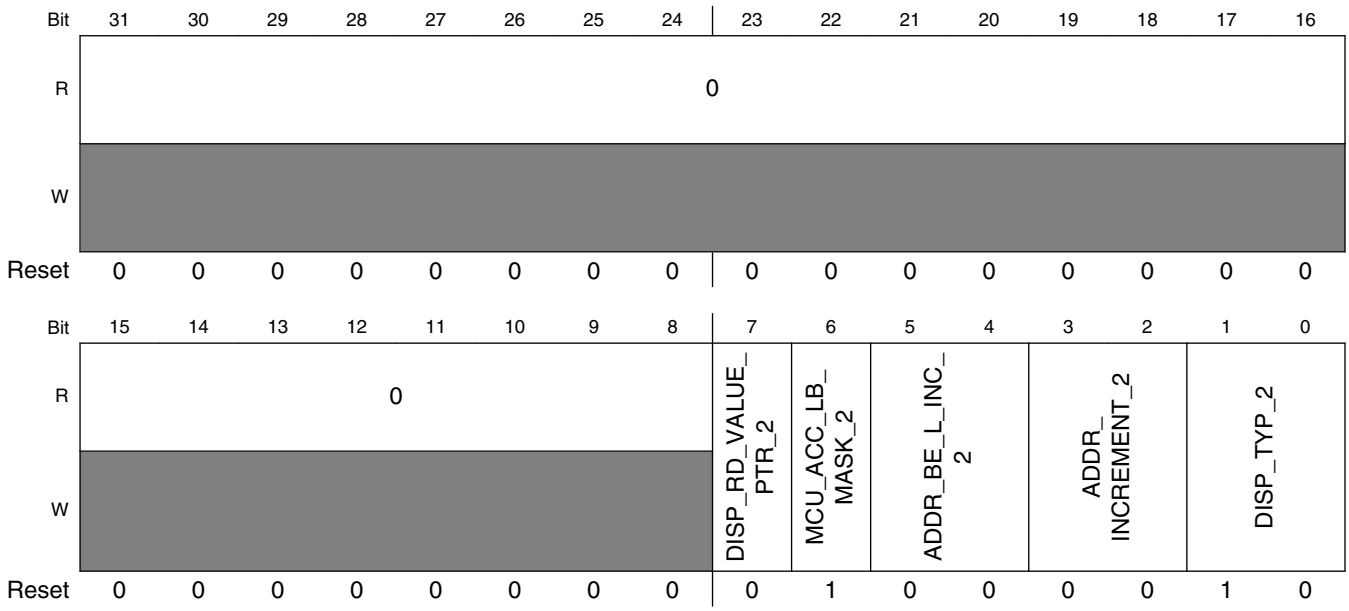


IPU_DC_DISP_CONF1_1 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.
7 DISP_RD_ VALUE_PTR_1	When the display works in wait for status mode. The IPU polls the display and compare the value to the value stored on the status DI_READ_DATA_ACK_VALUE and mask it with DI_READ_DATA_MASK. The DC holds 2 sets of mask and value. This field holds a pointer to the corresponding set of mask and value. 1 DI_READ_DATA_ACK_VALUE_1 & DI_READ_DATA_MASK_1 are used for display 1 0 DI_READ_DATA_ACK_VALUE_0 & DI_READ_DATA_MASK_0 are used for display 1
6 MCU_ACC_LB_ MASK_1	The DC compares between the current access to the a calculated address. The calculated address is the next consecutive address following the last address. This bit defines the comparing mode 1 The 2 addresses are fully compared 0 The 2 addresses are compared, but the ADDR[0] bit of the new address is ignored
5–4 ADDR_BE_L_ INC_1	This bits define the increment mode when the latest access was done with some of the byte enable signals are low, in that case different increment should be done instead of the normal auto increment of the address IF MCU_ACC_LB_MASK_1 is 0 then only 00 and 10 values are allowed. 00 No increment 01 Increment by 1 10 Increment by 2 11 Increment by 3
3–2 ADDR_ INCREMENT_1	This field is the increment step for auto increment mode 00 Increment the address by 1 01 Increment the address by 2 10 Increment the address by 3 11 Increment the address by 4
DISP_TYP_1	This field defines the type of the display 00 Serial accesses display 01 Reserved 10 parallel display, without byte_enable support 11 parallel display, with byte_enable support

38.5.324 DC Display Configuration 1 Register 2 (IPU_DC_DISP_CONF1_2)

Address: 260_0000h base + 5_80E0h offset = 265_80E0h



IPU_DC_DISP_CONF1_2 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.
7 DISP_RD_VALUE_PTR_2	When the display works in wait for status mode. The IPU polls the display and compare the value to the value stored on the status DI_READ_DATA_ACK_VALUE and mask it with DI_READ_DATA_MASK. The DC holds 2 sets of mask and value. This field holds a pointer to the corresponding set of mask and value. 1 DI_READ_DATA_ACK_VALUE_1 & DI_READ_DATA_MASK_1 are used for display 2 0 DI_READ_DATA_ACK_VALUE_0 & DI_READ_DATA_MASK_0 are used for display 2
6 MCU_ACC_LB_MASK_2	The DC compares between the current access to the a calculated address. The calculated address is the next consecutive address following the last address. This bit defines the comparing mode 1 The 2 addresses are fully compared 0 The 2 addresses are compared, but the ADDR[0] bit of the new address is ignored
5–4 ADDR_BE_L_INC_2	This bits define the increment mode when the latest access was done with some of the byte enable signals are low, in that case different increment should be done instead of the normal auto increment of the address IF MCU_ACC_LB_MASK_2 is 0 then only 00 and 10 values are allowed. 00 No increment 01 Increment by 1 10 Increment by 2 11 Increment by 3

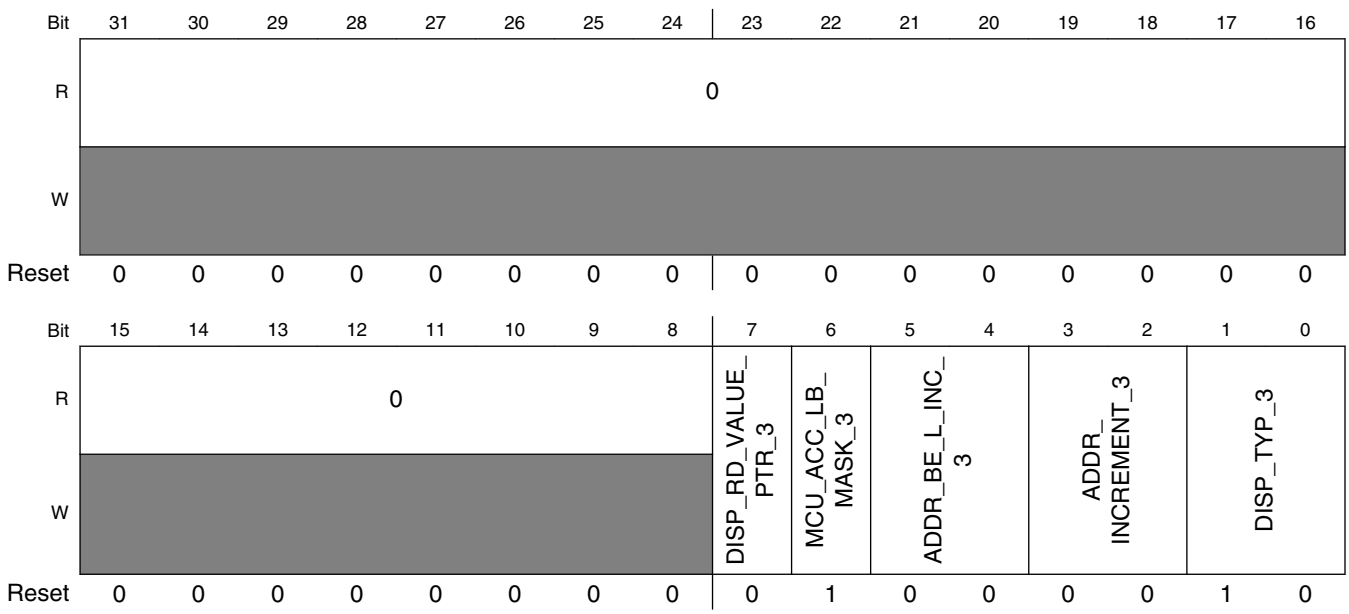
Table continues on the next page...

IPU_DC_DISP_CONF1_2 field descriptions (continued)

Field	Description
3–2 ADDR_ INCREMENT_2	This field is the increment step for auto increment mode 00 Increment the address by 1 01 Increment the address by 2 10 Increment the address by 3 11 Increment the address by 4
DISP_TYP_2	This field defines the type of the display 00 Serial accesses display 01 Reserved 10 parallel display, without byte_enable support 11 parallel display, with byte_enable support

38.5.325 DC Display Configuration 1 Register 3 (IPU_DC_DISP_CONF1_3)

Address: 260_0000h base + 5_80E4h offset = 265_80E4h



IPU_DC_DISP_CONF1_3 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.
7 DISP_RD_ VALUE_PTR_3	When the display works in wait for status mode. The IPU polls the display and compare the value to the value stored on the status DI_READ_DATA_ACK_VALUE and mask it with DI_READ_DATA_MASK. The DC holds 2 sets of mask and value. This field holds a pointer to the corresponding set of mask and value.

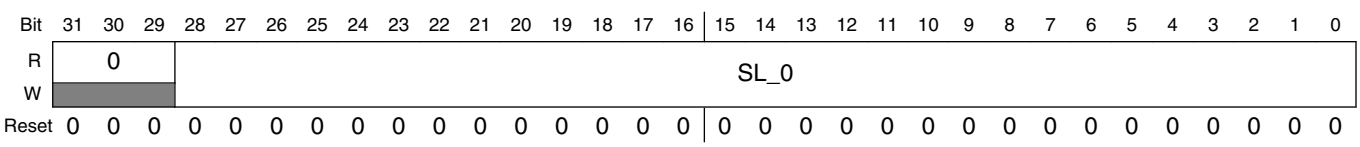
Table continues on the next page...

IPU_DC_DISP_CONF1_3 field descriptions (continued)

Field	Description
	1 DI_READ_DATA_ACK_VALUE_1 & DI_READ_DATA_MASK_1 are used for display 3 0 DI_READ_DATA_ACK_VALUE_0 & DI_READ_DATA_MASK_0 are used for display 3
6 MCU_ACC_LB_MASK_3	The DC compares between the current access to the a calculated address. The calculated address is the next consecutive address following the last address. This bit defines the comparing mode 1 The 2 addresses are fully compared 0 The 2 addresses are compared, but the ADDR[0] bit of the new address is ignored
5–4 ADDR_BE_L_INC_3	This bits define the increment mode when the latest access was done with some of the byte enable signals are low, in that case different increment should be done instead of the normal auto increment of the address IF MCU_ACC_LB_MASK_3 is 0 then only 00 and 10 values are allowed. 00 No increment 01 Increment by 1 10 Increment by 2 11 Increment by 3
3–2 ADDR_INCREMENT_3	This field is the increment step for auto increment mode 00 Increment the address by 1 01 Increment the address by 2 10 Increment the address by 3 11 Increment the address by 4
DISP_TYP_3	This field defines the type of the display 00 Serial accesses display 01 Reserved 10 parallel display, without byte_enable support 11 parallel display, with byte_enable support

38.5.326 DC Display Configuration 2 Register 0 (IPU_DC_DISP_CONF2_0)

Address: 260_0000h base + 5_80E8h offset = 265_80E8h



IPU_DC_DISP_CONF2_0 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
SL_0	Stride line of display 0

38.5.327 DC Display Configuration 2 Register 1 (IPU_DC_DISP_CONF2_1)

Address: 260_0000h base + 5_80ECh offset = 265_80ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																SL_1															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_DISP_CONF2_1 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
SL_1	Stride line of display 1

38.5.328 DC Display Configuration 2 Register 2 (IPU_DC_DISP_CONF2_2)

Address: 260_0000h base + 5_80F0h offset = 265_80F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																SL_2															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_DISP_CONF2_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
SL_2	Stride line of display 2

38.5.329 DC Display Configuration 2 Register 3 (IPU_DC_DISP_CONF2_3)

Address: 260_0000h base + 5_80F4h offset = 265_80F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																SL_3															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_DISP_CONF2_3 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
SL_3	Stride line of display 3

38.5.330 DC DI0Configuration Register 1 (IPU_DC_DI0_CONF_1)

Address: 260_0000h base + 5_80F8h offset = 265_80F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DI_READ_DATA_MASK_0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_DI0_CONF_1 field descriptions

Field	Description
DI_READ_ DATA_MASK_0	This field defines the mask value of the data read from the display.

38.5.331 DC DI0Configuration Register 2 (IPU_DC_DI0_CONF_2)

Address: 260_0000h base + 5_80FCh offset = 265_80FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DI_READ_DATA_ACK_VALUE_0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_DI0_CONF_2 field descriptions

Field	Description
DI_READ_ DATA_ACK_ VALUE_0	This is the expected data to be read from the display. The value reads from the display is anded with the DI_READ_DATA_MASK_0 and compared with the DI_READ_DATA_ACK_VALUE_0. This field is used for the READ_STATUS task of the DC

38.5.332 DC DI1Configuration Register 1 (IPU_DC_DI1_CONF_1)

Address: 260_0000h base + 5_8100h offset = 265_8100h

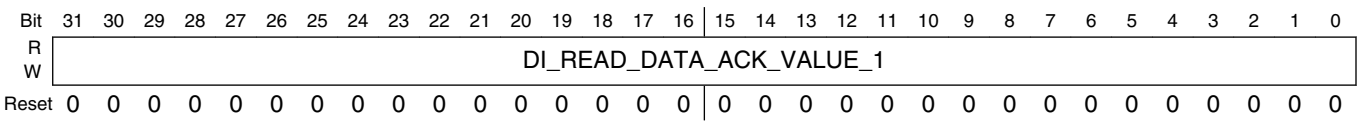
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DI_READ_DATA_MASK_1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_DI1_CONF_1 field descriptions

Field	Description
DI_READ_ DATA_MASK_1	This field defines the mask value of the data read from the display.

38.5.333 DC DI1Configuration Register 2 (IPU_DC_DI1_CONF_2)

Address: 260_0000h base + 5_8104h offset = 265_8104h

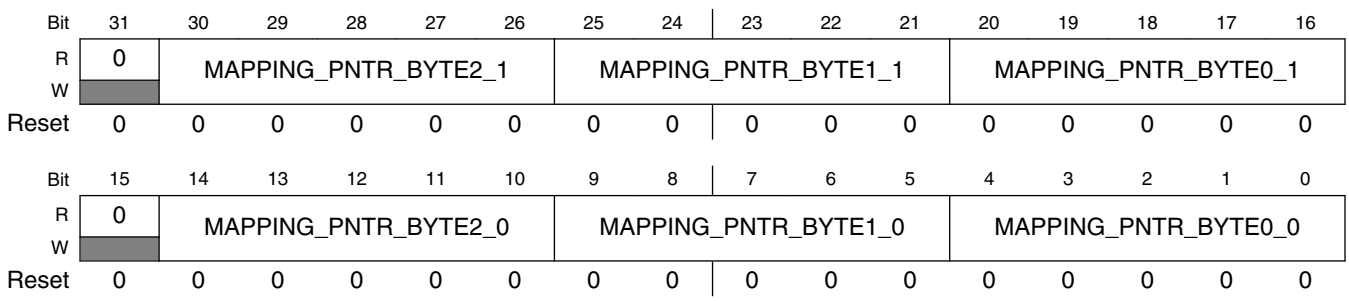


IPU_DC_DI1_CONF_2 field descriptions

Field	Description
DI_READ_ DATA_ACK_ VALUE_1	This is the expected data to be read from the display. The value reads from the display is anded with the DI_READ_DATA_MASK_1 and compared with the DI_READ_DATA_ACK_VALUE_1 This field is used for the READ_STATUS task of the DC

38.5.334 DC Mapping Configuration Register 0 (IPU_DC_MAP_CONF_0)

Address: 260_0000h base + 5_8108h offset = 265_8108h



IPU_DC_MAP_CONF_0 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–26 MAPPING_ PNTR_BYTE2_1	Mapping pointer #1 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_1	Mapping pointer #1 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1

Table continues on the next page...

IPU_DC_MAP_CONF_0 field descriptions (continued)

Field	Description
20–16 MAPPING_ PNTR_BYTE0_1	Mapping pointer #1 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value 0.
14–10 MAPPING_ PNTR_BYTE2_0	Mapping pointer #0 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_0	Mapping pointer #0 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
MAPPING_ PNTR_BYTE0_0	Mapping pointer #0 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

38.5.335 DC Mapping Configuration Register 1 (IPU_DC_MAP_CONF_1)

Address: 260_0000h base + 5_810Ch offset = 265_810Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAPPING_PNTR_BYTE2_3							MAPPING_PNTR_BYTE1_3			MAPPING_PNTR_BYTE0_3				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_PNTR_BYTE2_2							MAPPING_PNTR_BYTE1_2			MAPPING_PNTR_BYTE0_2				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–26 MAPPING_ PNTR_BYTE2_3	Mapping pointer #3 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_3	Mapping pointer #3 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_3	Mapping pointer #3 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DC_MAP_CONF_1 field descriptions (continued)

Field	Description
14–10 MAPPING_ PNTR_BYTE2_2	Mapping pointer #1 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_2	Mapping pointer #1 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
MAPPING_ PNTR_BYTE0_2	Mapping pointer #1 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

38.5.336 DC Mapping Configuration Register 2 (IPU_DC_MAP_CONF_2)

Address: 260_0000h base + 5_8110h offset = 265_8110h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAPPING_PNTR_BYTE2_5							MAPPING_PNTR_BYTE1_5			MAPPING_PNTR_BYTE0_5				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_PNTR_BYTE2_4							MAPPING_PNTR_BYTE1_4			MAPPING_PNTR_BYTE0_4				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–26 MAPPING_ PNTR_BYTE2_5	Mapping pointer #5 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_5	Mapping pointer #5 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_5	Mapping pointer #5 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value 0.
14–10 MAPPING_ PNTR_BYTE2_4	Mapping pointer #4 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_4	Mapping pointer #4 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1

Table continues on the next page...

IPU_DC_MAP_CONF_2 field descriptions (continued)

Field	Description
MAPPING_ PNTR_BYTE0_4	Mapping pointer #4 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

38.5.337 DC Mapping Configuration Register 3 (IPU_DC_MAP_CONF_3)

Address: 260_0000h base + 5_8114h offset = 265_8114h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAPPING_PNTR_BYTE2_7							MAPPING_PNTR_BYTE1_7			MAPPING_PNTR_BYTE0_7				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_PNTR_BYTE2_6							MAPPING_PNTR_BYTE1_6			MAPPING_PNTR_BYTE0_6				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_3 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–26 MAPPING_ PNTR_BYTE2_7	Mapping pointer #7 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_7	Mapping pointer #7 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_7	Mapping pointer #7 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value 0.
14–10 MAPPING_ PNTR_BYTE2_6	Mapping pointer #6 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_6	Mapping pointer #6 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
MAPPING_ PNTR_BYTE0_6	Mapping pointer #6 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

38.5.338 DC Mapping Configuration Register 4 (IPU_DC_MAP_CONF_4)

Address: 260_0000h base + 5_8118h offset = 265_8118h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAPPING_PNTR_BYTE2_9							MAPPING_PNTR_BYTE1_1			MAPPING_PNTR_BYTE0_9				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_PNTR_BYTE2_8							MAPPING_PNTR_BYTE1_8			MAPPING_PNTR_BYTE0_8				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_4 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–26 MAPPING_ PNTR_BYTE2_9	Mapping pointer #9 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_1	Mapping pointer #9 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_9	Mapping pointer #9 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value 0.
14–10 MAPPING_ PNTR_BYTE2_8	Mapping pointer #8 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_8	Mapping pointer #8 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
MAPPING_ PNTR_BYTE0_8	Mapping pointer #8 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

38.5.339 DC Mapping Configuration Register 5 (IPU_DC_MAP_CONF_5)

Address: 260_0000h base + 5_811Ch offset = 265_811Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAPPING_PNTR_BYTE2_11							MAPPING_PNTR_BYTE1_11				MAPPING_PNTR_BYTE0_11			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_PNTR_BYTE2_10							MAPPING_PNTR_BYTE1_10				MAPPING_PNTR_BYTE0_10			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_5 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–26 MAPPING_ PNTR_BYTE2_ 11	Mapping pointer #11 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 11	Mapping pointer #11 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 11	Mapping pointer #11 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value 0.
14–10 MAPPING_ PNTR_BYTE2_ 10	Mapping pointer #10 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 10	Mapping pointer #10 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
MAPPING_ PNTR_BYTE0_ 10	Mapping pointer #10 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

38.5.340 DC Mapping Configuration Register 6 (IPU_DC_MAP_CONF_6)

Address: 260_0000h base + 5_8120h offset = 265_8120h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAPPING_PNTR_BYTE2_13					MAPPING_PNTR_BYTE1_13					MAPPING_PNTR_BYTE0_13				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_PNTR_BYTE2_12					MAPPING_PNTR_BYTE1_12					MAPPING_PNTR_BYTE0_12				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_6 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–26 MAPPING_ PNTR_BYTE2_ 13	Mapping pointer #13 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 13	Mapping pointer #13 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 13	Mapping pointer #13 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value 0.
14–10 MAPPING_ PNTR_BYTE2_ 12	Mapping pointer #12 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 12	Mapping pointer #12 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
MAPPING_ PNTR_BYTE0_ 12	Mapping pointer #12 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

38.5.341 DC Mapping Configuration Register 7 (IPU_DC_MAP_CONF_7)

Address: 260_0000h base + 5_8124h offset = 265_8124h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAPPING_PNTR_BYTE2_15					MAPPING_PNTR_BYTE1_15					MAPPING_PNTR_BYTE0_15				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_PNTR_BYTE2_14					MAPPING_PNTR_BYTE1_14					MAPPING_PNTR_BYTE0_14				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_7 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–26 MAPPING_ PNTR_BYTE2_ 15	Mapping pointer #15 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 15	Mapping pointer #15 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 15	Mapping pointer #15 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value 0.
14–10 MAPPING_ PNTR_BYTE2_ 14	Mapping pointer #14 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 14	Mapping pointer #14 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
MAPPING_ PNTR_BYTE0_ 14	Mapping pointer #14 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

38.5.342 DC Mapping Configuration Register 8 (IPU_DC_MAP_CONF_8)

Address: 260_0000h base + 5_8128h offset = 265_8128h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAPPING_PNTR_BYTE2_17					MAPPING_PNTR_BYTE1_17					MAPPING_PNTR_BYTE0_17				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_PNTR_BYTE2_16					MAPPING_PNTR_BYTE1_16					MAPPING_PNTR_BYTE0_16				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_8 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–26 MAPPING_ PNTR_BYTE2_ 17	Mapping pointer #17 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 17	Mapping pointer #17 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 17	Mapping pointer #17 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value 0.
14–10 MAPPING_ PNTR_BYTE2_ 16	Mapping pointer #16 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 16	Mapping pointer #16 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
MAPPING_ PNTR_BYTE0_ 16	Mapping pointer #16 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

38.5.343 DC Mapping Configuration Register 9 (IPU_DC_MAP_CONF_9)

Address: 260_0000h base + 5_812Ch offset = 265_812Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAPPING_PNTR_BYTE2_19							MAPPING_PNTR_BYTE1_19			MAPPING_PNTR_BYTE0_19				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_PNTR_BYTE2_18							MAPPING_PNTR_BYTE1_18			MAPPING_PNTR_BYTE0_18				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_9 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–26 MAPPING_ PNTR_BYTE2_ 19	Mapping pointer #19 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 19	Mapping pointer #19 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 19	Mapping pointer #19 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value 0.
14–10 MAPPING_ PNTR_BYTE2_ 18	Mapping pointer #18 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 18	Mapping pointer #18 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
MAPPING_ PNTR_BYTE0_ 18	Mapping pointer #18 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

38.5.344 DC Mapping Configuration Register 10 (IPU_DC_MAP_CONF_10)

Address: 260_0000h base + 5_8130h offset = 265_8130h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAPPING_PNTR_BYTE2_21					MAPPING_PNTR_BYTE1_21					MAPPING_PNTR_BYTE0_21				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_PNTR_BYTE2_20					MAPPING_PNTR_BYTE1_20					MAPPING_PNTR_BYTE0_20				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_10 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–26 MAPPING_ PNTR_BYTE2_ 21	Mapping pointer #21 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 21	Mapping pointer #21 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 21	Mapping pointer #21 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value 0.
14–10 MAPPING_ PNTR_BYTE2_ 20	Mapping pointer #20 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 20	Mapping pointer #20 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
MAPPING_ PNTR_BYTE0_ 20	Mapping pointer #20 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

38.5.345 DC Mapping Configuration Register 11 (IPU_DC_MAP_CONF_11)

Address: 260_0000h base + 5_8134h offset = 265_8134h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAPPING_PNTR_BYTE2_23							MAPPING_PNTR_BYTE1_23			MAPPING_PNTR_BYTE0_23				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_PNTR_BYTE2_22							MAPPING_PNTR_BYTE1_22			MAPPING_PNTR_BYTE0_22				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_11 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–26 MAPPING_ PNTR_BYTE2_ 23	Mapping pointer #23 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 23	Mapping pointer #23 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 23	Mapping pointer #23 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value 0.
14–10 MAPPING_ PNTR_BYTE2_ 22	Mapping pointer #22 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 22	Mapping pointer #22 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
MAPPING_ PNTR_BYTE0_ 22	Mapping pointer #22 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

38.5.346 DC Mapping Configuration Register 12 (IPU_DC_MAP_CONF_12)

Address: 260_0000h base + 5_8138h offset = 265_8138h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAPPING_PNTR_BYTE2_25					MAPPING_PNTR_BYTE1_25					MAPPING_PNTR_BYTE0_25				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_PNTR_BYTE2_24					MAPPING_PNTR_BYTE1_24					MAPPING_PNTR_BYTE0_24				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_12 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–26 MAPPING_ PNTR_BYTE2_ 25	Mapping pointer #25 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 25	Mapping pointer #25 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 25	Mapping pointer #25 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value 0.
14–10 MAPPING_ PNTR_BYTE2_ 24	Mapping pointer #24 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 24	Mapping pointer #24 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
MAPPING_ PNTR_BYTE0_ 24	Mapping pointer #24 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

38.5.347 DC Mapping Configuration Register 13 (IPU_DC_MAP_CONF_13)

Address: 260_0000h base + 5_813Ch offset = 265_813Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAPPING_PNTR_BYTE2_27							MAPPING_PNTR_BYTE1_27				MAPPING_PNTR_BYTE0_27			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_PNTR_BYTE2_26							MAPPING_PNTR_BYTE1_26				MAPPING_PNTR_BYTE0_26			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_13 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–26 MAPPING_ PNTR_BYTE2_ 27	Mapping pointer #27 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 27	Mapping pointer #27 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 27	Mapping pointer #27 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value 0.
14–10 MAPPING_ PNTR_BYTE2_ 26	Mapping pointer #26 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 26	Mapping pointer #26 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
MAPPING_ PNTR_BYTE0_ 26	Mapping pointer #26 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

38.5.348 DC Mapping Configuration Register 14 (IPU_DC_MAP_CONF_14)

Address: 260_0000h base + 5_8140h offset = 265_8140h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAPPING_PNTR_BYTE2_29							MAPPING_PNTR_BYTE1_29			MAPPING_PNTR_BYTE0_29				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

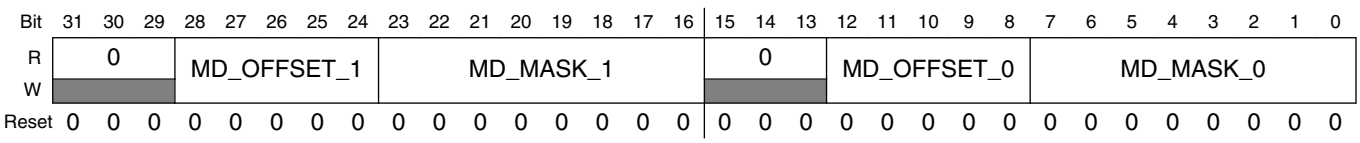
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_PNTR_BYTE2_28f							MAPPING_PNTR_BYTE1_28			MAPPING_PNTR_BYTE2_28				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_14 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–26 MAPPING_ PNTR_BYTE2_ 29	Mapping pointer #29 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 29	Mapping pointer #29 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 29	Mapping pointer #29 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value 0.
14–10 MAPPING_ PNTR_BYTE2_ 28f	Mapping pointer #28 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 28	Mapping pointer #28 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
MAPPING_ PNTR_BYTE2_ 28	Mapping pointer #28 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

38.5.349 DC Mapping Configuration Register 15 (IPU_DC_MAP_CONF_15)

Address: 260_0000h base + 5_8144h offset = 265_8144h

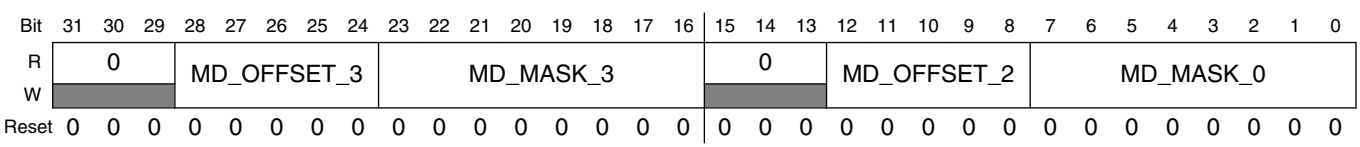


IPU_DC_MAP_CONF_15 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–24 MD_OFFSET_1	Mapping unit's offset parameter #1 This field defines the offset parameter #1 within the 24bit word coming from the DC.
23–16 MD_MASK_1	Mapping unit's mask value #1 This field defines the mask value #1 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value 0.
12–8 MD_OFFSET_0	Mapping unit's offset parameter #0 This field defines the offset parameter #0 within the 24bit word coming from the DC.
MD_MASK_0	Mapping unit's mask value #0 This field defines the mask value #0 within the 8bit word coming from the DC.

38.5.350 DC Mapping Configuration Register 16 (IPU_DC_MAP_CONF_16)

Address: 260_0000h base + 5_8148h offset = 265_8148h



IPU_DC_MAP_CONF_16 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–24 MD_OFFSET_3	Mapping unit's offset parameter #3 This field defines the offset parameter #3 within the 24bit word coming from the DC.

Table continues on the next page...

IPU_DC_MAP_CONF_16 field descriptions (continued)

Field	Description
23–16 MD_MASK_3	Mapping unit's mask value #3 This field defines the mask value #3 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value 0.
12–8 MD_OFFSET_2	Mapping unit's offset parameter #2 This field defines the offset parameter #2 within the 24bit word coming from the DC.
MD_MASK_0	Mapping unit's mask value #2 This field defines the mask value #2 within the 8bit word coming from the DC.

38.5.351 DC Mapping Configuration Register 17 (IPU_DC_MAP_CONF_17)

Address: 260_0000h base + 5_814Ch offset = 265_814Ch

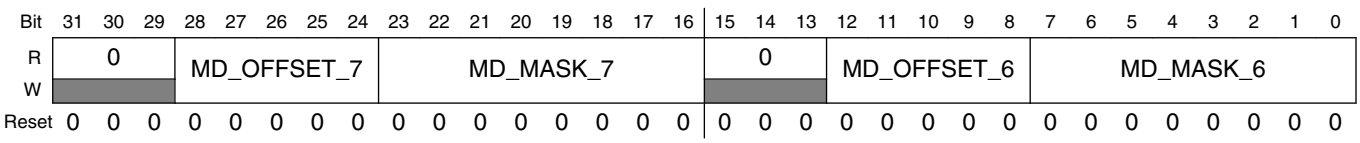
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			MD_OFFSET_5					MD_MASK_5								0			MD_OFFSET_4					MD_MASK_4							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_17 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–24 MD_OFFSET_5	Mapping unit's offset parameter #5 This field defines the offset parameter #5 within the 24bit word coming from the DC.
23–16 MD_MASK_5	Mapping unit's mask value #5 This field defines the mask value #5 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value 0.
12–8 MD_OFFSET_4	Mapping unit's offset parameter #4 This field defines the offset parameter #4 within the 24bit word coming from the DC.
MD_MASK_4	Mapping unit's mask value #4 This field defines the mask value #4 within the 8bit word coming from the DC.

38.5.352 DC Mapping Configuration Register 18 (IPU_DC_MAP_CONF_18)

Address: 260_0000h base + 5_8150h offset = 265_8150h

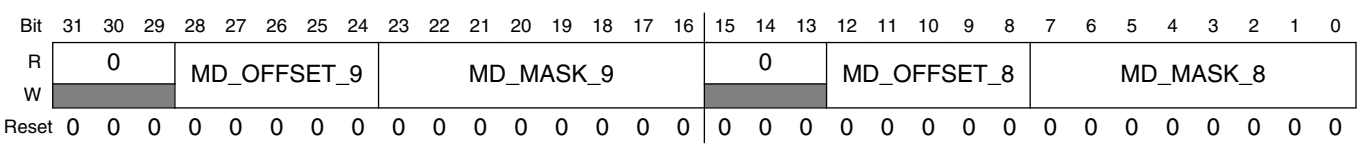


IPU_DC_MAP_CONF_18 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–24 MD_OFFSET_7	Mapping unit's offset parameter #7 This field defines the offset parameter #7 within the 24bit word coming from the DC.
23–16 MD_MASK_7	Mapping unit's mask value #7 This field defines the mask value #7 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value 0.
12–8 MD_OFFSET_6	Mapping unit's offset parameter #6 This field defines the offset parameter #6 within the 24bit word coming from the DC.
MD_MASK_6	Mapping unit's mask value #6 This field defines the mask value #6 within the 8bit word coming from the DC.

38.5.353 DC Mapping Configuration Register 19 (IPU_DC_MAP_CONF_19)

Address: 260_0000h base + 5_8154h offset = 265_8154h



IPU_DC_MAP_CONF_19 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–24 MD_OFFSET_9	Mapping unit's offset parameter #9 This field defines the offset parameter #9 within the 24bit word coming from the DC.

Table continues on the next page...

IPU_DC_MAP_CONF_19 field descriptions (continued)

Field	Description
23–16 MD_MASK_9	Mapping unit's mask value #9 This field defines the mask value #9 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value 0.
12–8 MD_OFFSET_8	Mapping unit's offset parameter #8 This field defines the offset parameter #8 within the 24bit word coming from the DC.
MD_MASK_8	Mapping unit's mask value #8 This field defines the mask value #8 within the 8bit word coming from the DC.

38.5.354 DC Mapping Configuration Register 20 (IPU_DC_MAP_CONF_20)

Address: 260_0000h base + 5_8158h offset = 265_8158h

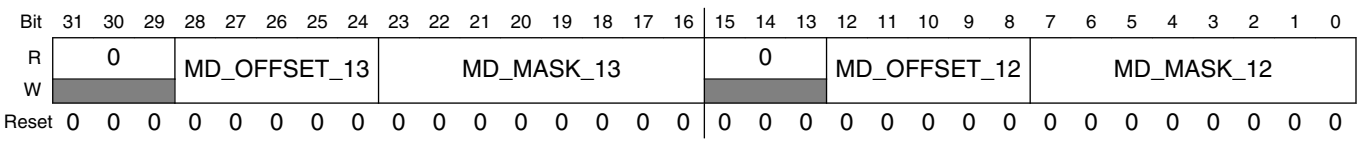
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			MD_OFFSET_11					MD_MASK_11								0			MD_OFFSET_10					MD_MASK_10							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_20 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–24 MD_OFFSET_11	Mapping unit's offset parameter #11 This field defines the offset parameter #11 within the 24bit word coming from the DC.
23–16 MD_MASK_11	Mapping unit's mask value #11 This field defines the mask value #11 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value 0.
12–8 MD_OFFSET_10	Mapping unit's offset parameter #10 This field defines the offset parameter #10 within the 24bit word coming from the DC.
MD_MASK_10	Mapping unit's mask value #10 This field defines the mask value #10 within the 8bit word coming from the DC.

38.5.355 DC Mapping Configuration Register 21 (IPU_DC_MAP_CONF_21)

Address: 260_0000h base + 5_815Ch offset = 265_815Ch

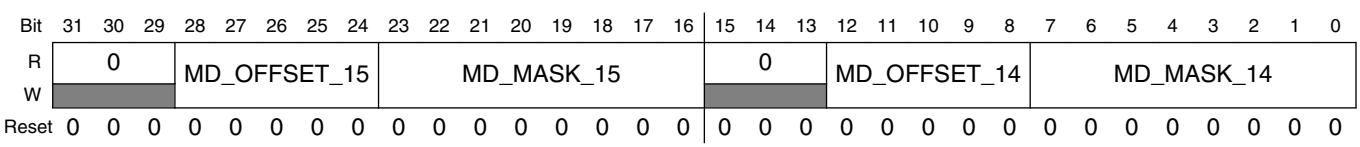


IPU_DC_MAP_CONF_21 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–24 MD_OFFSET_13	Mapping unit's offset parameter #13 This field defines the offset parameter #13 within the 24bit word coming from the DC.
23–16 MD_MASK_13	Mapping unit's mask value #13 This field defines the mask value #13 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value 0.
12–8 MD_OFFSET_12	Mapping unit's offset parameter #12 This field defines the offset parameter #12 within the 24bit word coming from the DC.
MD_MASK_12	Mapping unit's mask value #12 This field defines the mask value #12 within the 8bit word coming from the DC.

38.5.356 DC Mapping Configuration Register 22 (IPU_DC_MAP_CONF_22)

Address: 260_0000h base + 5_8160h offset = 265_8160h



IPU_DC_MAP_CONF_22 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–24 MD_OFFSET_15	Mapping unit's offset parameter #15 This field defines the offset parameter #15 within the 24bit word coming from the DC.

Table continues on the next page...

IPU_DC_MAP_CONF_22 field descriptions (continued)

Field	Description
23–16 MD_MASK_15	Mapping unit's mask value #15 This field defines the mask value #15 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value 0.
12–8 MD_OFFSET_14	Mapping unit's offset parameter #14 This field defines the offset parameter #14 within the 24bit word coming from the DC.
MD_MASK_14	Mapping unit's mask value #14 This field defines the mask value #14 within the 8bit word coming from the DC.

38.5.357 DC Mapping Configuration Register 23 (IPU_DC_MAP_CONF_23)

Address: 260_0000h base + 5_8164h offset = 265_8164h

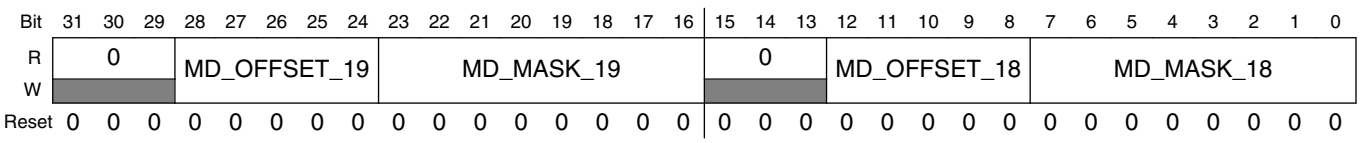
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			MD_OFFSET_17					MD_MASK_17								0			MD_OFFSET_16					MD_MASK_16							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_23 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–24 MD_OFFSET_17	Mapping unit's offset parameter #17 This field defines the offset parameter #17 within the 24bit word coming from the DC.
23–16 MD_MASK_17	Mapping unit's mask value #17 This field defines the mask value #17 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value 0.
12–8 MD_OFFSET_16	Mapping unit's offset parameter #16 This field defines the offset parameter #16 within the 24bit word coming from the DC.
MD_MASK_16	Mapping unit's mask value #16 This field defines the mask value #16 within the 8bit word coming from the DC.

38.5.358 DC Mapping Configuration Register 24 (IPU_DC_MAP_CONF_24)

Address: 260_0000h base + 5_8168h offset = 265_8168h

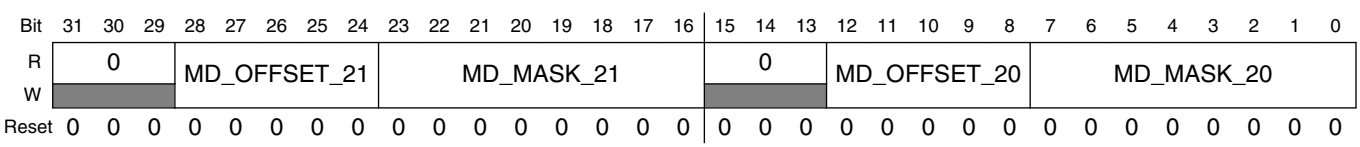


IPU_DC_MAP_CONF_24 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–24 MD_OFFSET_19	Mapping unit's offset parameter #19 This field defines the offset parameter #19 within the 24bit word coming from the DC.
23–16 MD_MASK_19	Mapping unit's mask value #19 This field defines the mask value #19 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value 0.
12–8 MD_OFFSET_18	Mapping unit's offset parameter #18 This field defines the offset parameter #18 within the 24bit word coming from the DC.
MD_MASK_18	Mapping unit's mask value #18 This field defines the mask value #18 within the 8bit word coming from the DC.

38.5.359 DC Mapping Configuration Register 25 (IPU_DC_MAP_CONF_25)

Address: 260_0000h base + 5_816Ch offset = 265_816Ch



IPU_DC_MAP_CONF_25 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–24 MD_OFFSET_21	Mapping unit's offset parameter #21 This field defines the offset parameter #21 within the 24bit word coming from the DC.

Table continues on the next page...

IPU_DC_MAP_CONF_25 field descriptions (continued)

Field	Description
23–16 MD_MASK_21	Mapping unit's mask value #21 This field defines the mask value #21 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value 0.
12–8 MD_OFFSET_20	Mapping unit's offset parameter #20 This field defines the offset parameter #20 within the 24bit word coming from the DC.
MD_MASK_20	Mapping unit's mask value #20 This field defines the mask value #20 within the 8bit word coming from the DC.

38.5.360 DC Mapping Configuration Register 26 (IPU_DC_MAP_CONF_26)

Address: 260_0000h base + 5_8170h offset = 265_8170h

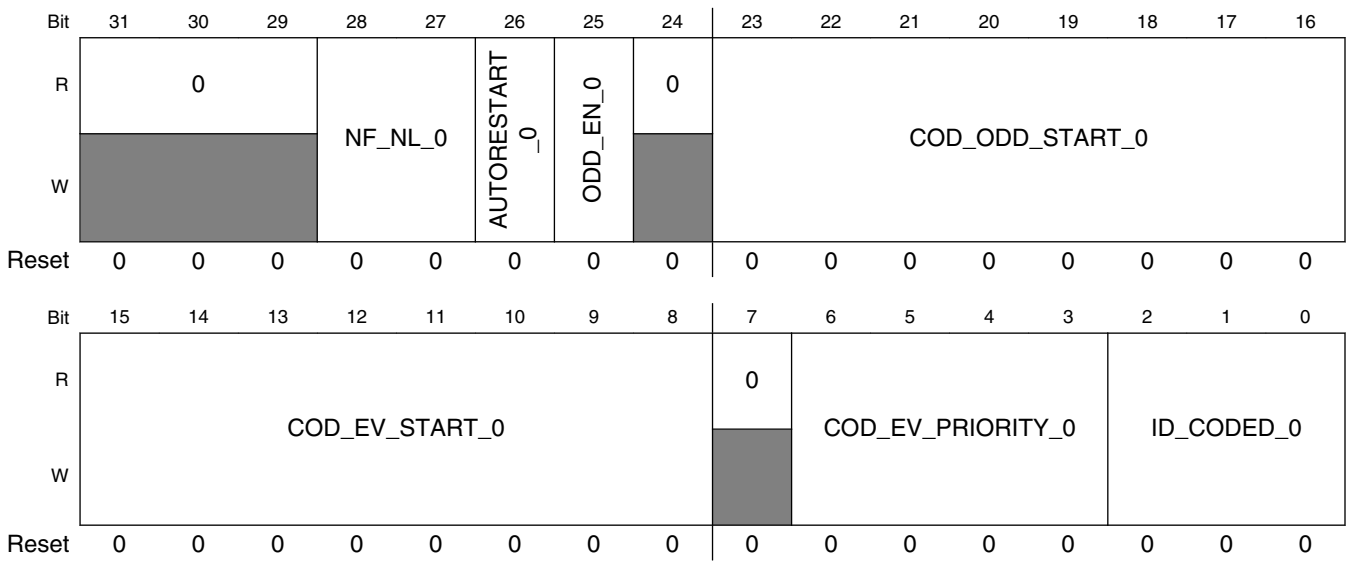
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			MD_OFFSET_23					MD_MASK_23								0			MD_OFFSET_22					MD_MASK_22							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_26 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–24 MD_OFFSET_23	Mapping unit's offset parameter #23 This field defines the offset parameter #23 within the 24bit word coming from the DC.
23–16 MD_MASK_23	Mapping unit's mask value #23 This field defines the mask value #23 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value 0.
12–8 MD_OFFSET_22	Mapping unit's offset parameter #22 This field defines the offset parameter #22 within the 24bit word coming from the DC.
MD_MASK_22	Mapping unit's mask value #22 This field defines the mask value #22 within the 8bit word coming from the DC.

38.5.361 DC User General Data Event 0 Register 0 (IPU_DC_UGDE0_0)

Address: 260_0000h base + 5_8174h offset = 265_8174h



IPU_DC_UGDE0_0 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–27 NF_NL_0	the user may attach his general event #0 to New-line New-Frame and New-field events. One of these event triggers the user's general event #0's counter. The actual internal trigger is the pixel following the occurrence of the selected event. 00 New Line 01 New Frame 10 New Field 11 Reserved
26 AUTORESTART_0	User's general event #0 auto restart mode 0 disable 1 User's general event #0's counter is automatically restarted.
25 ODD_EN_0	The user's general event #0 may be split into 2 internal signals. One one mode all the events are sent on one signal, on the second mode odd events are sent over one signal while even events are sent over the other signal (ODD_MODE) 1 ODD_MODE is enabled 0 ODD_MODE is disabled
24 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DC_UGDE0_0 field descriptions (continued)

Field	Description
23–16 COD_ODD_ START_0	This field holds a pointer in the microcode holding the routine to be performed following the user general event #0. When ODD_MODE is enabled, only the odd events will use this pointer When ODD_MODE is disabled this field is ignored
15–8 COD_EV_ START_0	This field holds a pointer in the microcode holding the routine to be performed following the user general event #0. When ODD_MODE is enabled, only the even events will use this pointer When ODD_MODE is disabled, all the events will use this pointer
7 Reserved	This read-only field is reserved and always has the value 0.
6–3 COD_EV_ PRIORITY_0	This field defines the priority of the user general event #0 The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
ID_CODED_0	This field defines the number of DC channel number that user's general event #0 will be associated to. 000 DC channel_0 001 DC channel_1 010 DC channel_2 011 DC channel_5 (DP_SYNC) 100 DC channel_6 (DP ASYNC) 101 Reserved 110 Reserved. 111 Reserved.

38.5.362 DC User General Data Event 0 Register 1 (IPU_DC_UGDE0_1)

Address: 260_0000h base + 5_8178h offset = 265_8178h

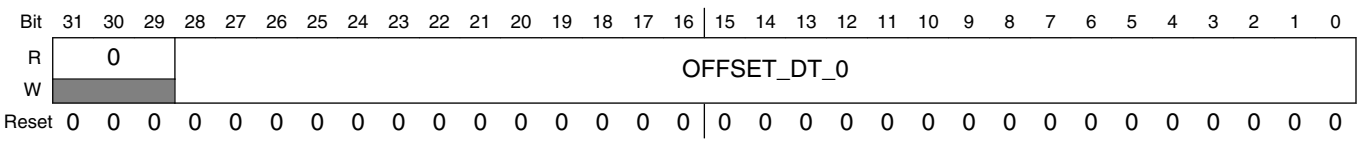
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																STEP_0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE0_1 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
STEP_0	This field holds the pre defined value that the counter counts too.

38.5.363 DC User General Data Event 0 Register2 (IPU_DC_UGDE0_2)

Address: 260_0000h base + 5_817Ch offset = 265_817Ch

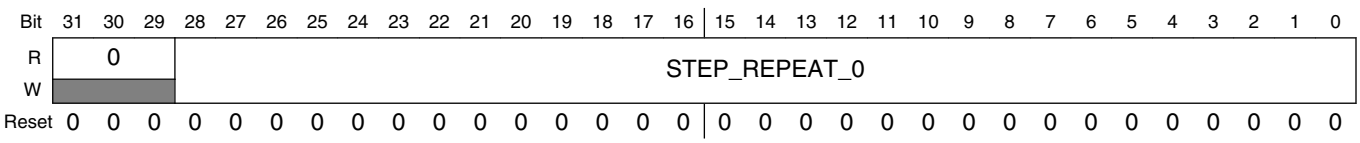


IPU_DC_UGDE0_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
OFFSET_DT_0	This field defines the offset value from which the counter of user general event #0 will start counting from

38.5.364 DC User General Data Event 0 Register 3 (IPU_DC_UGDE0_3)

Address: 260_0000h base + 5_8180h offset = 265_8180h

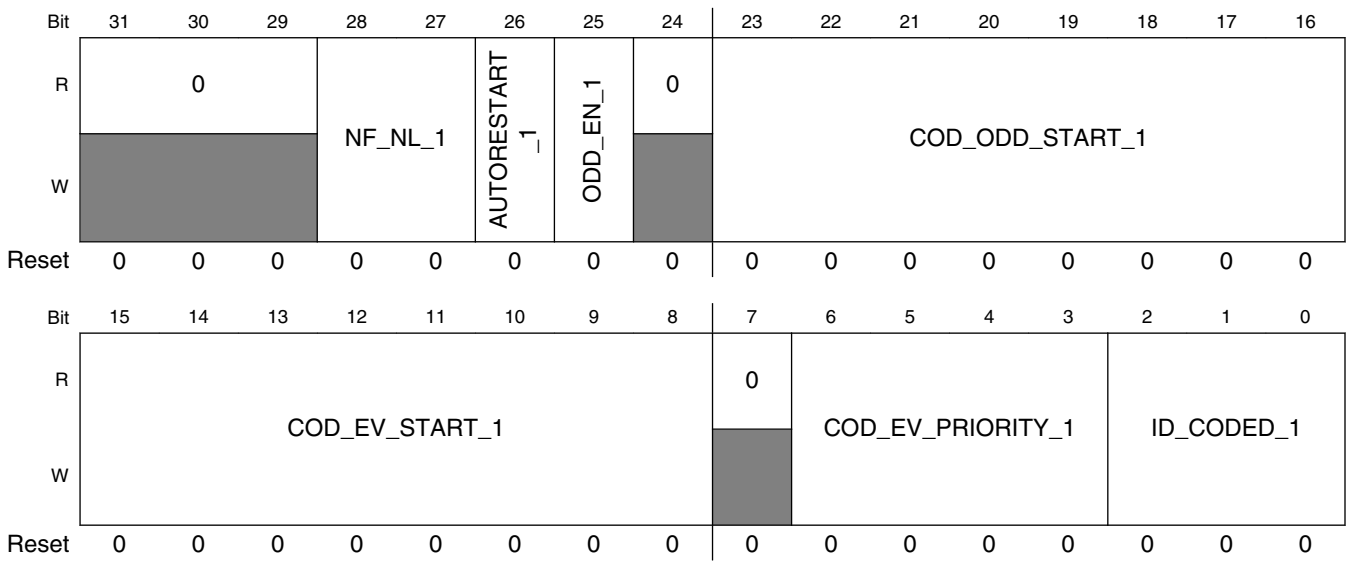


IPU_DC_UGDE0_3 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
STEP_REPEAT_0	When auto reload mode is disabled this field defines the number of events that will be generated by the user general event #0 mechanism

38.5.365 DC User General Data Event 1Register0 (IPU_DC_UGDE1_0)

Address: 260_0000h base + 5_8184h offset = 265_8184h



IPU_DC_UGDE1_0 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–27 NF_NL_1	the user may attach his general event #1 to New-line New-Frame and New-field events. One of these event triggers the user's general event #1's counter. The actual internal trigger is the pixel following the occurrence of the selected event. 00 New Line 01 New Frame 10 New Field 11 Reserved
26 AUTORESTART_1	User's general event #1 auto restart mode 0 disable 1 User's general event #1's counter is automatically restarted.
25 ODD_EN_1	The user's general event #1 may be split into 2 internal signals. One one mode all the events are sent on one signal, on the second mode odd events are sent over one signal while even events are sent over the other signal (ODD_MODE) 1 ODD_MODE is enabled 0 ODD_MODE is disabled
24 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DC_UGDE1_0 field descriptions (continued)

Field	Description
23–16 COD_ODD_ START_1	This field holds a pointer in the microcode holding the routine to be performed following the user general event #1. When ODD_MODE is enabled, only the odd events will use this pointer When ODD_MODE is disabled this field is ignored
15–8 COD_EV_ START_1	This field holds a pointer in the microcode holding the routine to be performed following the user general event #1. When ODD_MODE is enabled, only the even events will use this pointer When ODD_MODE is disabled, all the events will use this pointer
7 Reserved	This read-only field is reserved and always has the value 0.
6–3 COD_EV_ PRIORITY_1	This field defines the priority of the user general event #1 The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
ID_CODED_1	This field defines the number of DC channel number that user's general event #1 will be associated to 000 DC channel_0 001 DC channel_1 010 DC channel_2 011 DC channel_5 (DP_SYNC) 100 DC channel_6 (DP ASYNC) 101 Reserved 110 Reserved 111 Reserved

38.5.366 DC User General Data Event 1 Register 1 (IPU_DC_UGDE1_1)

Address: 260_0000h base + 5_8188h offset = 265_8188h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																STEP_1															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE1_1 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
STEP_1	This field hold the pre defined value that the counter counts too

38.5.367 DC User General Data Event 1Register 2 (IPU_DC_UGDE1_2)

Address: 260_0000h base + 5_818Ch offset = 265_818Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																OFFSET_DT_1																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE1_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
OFFSET_DT_1	This field defines the offset value from which the counter of user general event #1 will start counting from

38.5.368 DC User General Data Event 1Register 3 (IPU_DC_UGDE1_3)

Address: 260_0000h base + 5_8190h offset = 265_8190h

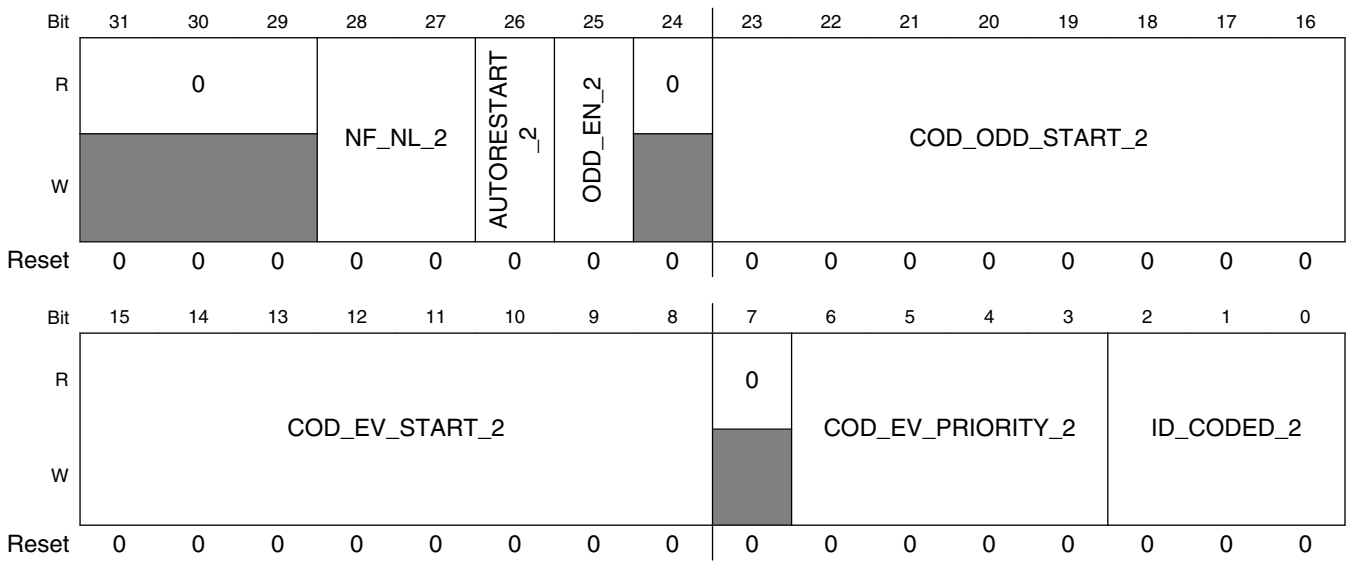
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																STEP_REPEAT_1																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE1_3 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
STEP_REPEAT_1	When auto reload mode is disabled this field defines the number of events that will be generated by the user general event #1 mechanism

38.5.369 DC User General Data Event 2 Register 0 (IPU_DC_UGDE2_0)

Address: 260_0000h base + 5_8194h offset = 265_8194h



IPU_DC_UGDE2_0 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–27 NF_NL_2	the user may attach his general event #2 to New-line New-Frame and New-field events. One of these event triggers the user's general event #2's counter. The actual internal trigger is the pixel following the occurrence of the selected event. 00 New Line 01 New Frame 10 New Field 11 Reserved
26 AUTORESTART_2	User's general event #2 auto restart mode 0 disable 1 User's general event #2's counter is automatically restarted.
25 ODD_EN_2	The user's general event #2 may be split into 2 internal signals. One one mode all the events are sent on one signal, on the second mode odd events are sent over one signal while even events are sent over the other signal (ODD_MODE) 1 ODD_MODE is enabled 0 ODD_MODE is disabled
24 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DC_UGDE2_0 field descriptions (continued)

Field	Description
23–16 COD_ODD_ START_2	This field holds a pointer in the microcode holding the routine to be performed following the user general event #2 When ODD_MODE is enabled, only the odd events will use this pointer When ODD_MODE is disabled this field is ignored
15–8 COD_EV_ START_2	This field holds a pointer in the microcode holding the routine to be performed following the user general event #2. When ODD_MODE is enabled, only the even events will use this pointer When ODD_MODE is disabled, all the events will use this pointer
7 Reserved	This read-only field is reserved and always has the value 0.
6–3 COD_EV_ PRIORITY_2	This field defines the priority of the user general event #2 The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
ID_CODED_2	This field defines the number of DC channel number that user's general event #2 will be associated to 000 DC channel_0 001 DC channel_1 010 DC channel_2 011 DC channel_5 (DP_SYNC) 100 DC channel_6 (DP ASYNC) 101 Reserved 110 Reserved 111 Reserved

38.5.370 DC User General Data Event 2 Register 1 (IPU_DC_UGDE2_1)

Address: 260_0000h base + 5_8198h offset = 265_8198h

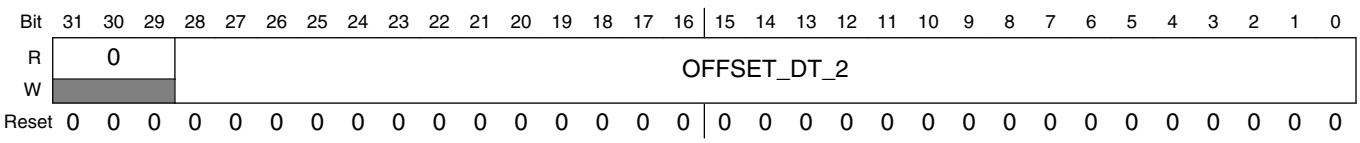
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																STEP_2															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE2_1 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
STEP_2	This field hold the pre defined value that the counter counts too

38.5.371 DC User General Data Event 2Register 2 (IPU_DC_UGDE2_2)

Address: 260_0000h base + 5_819Ch offset = 265_819Ch

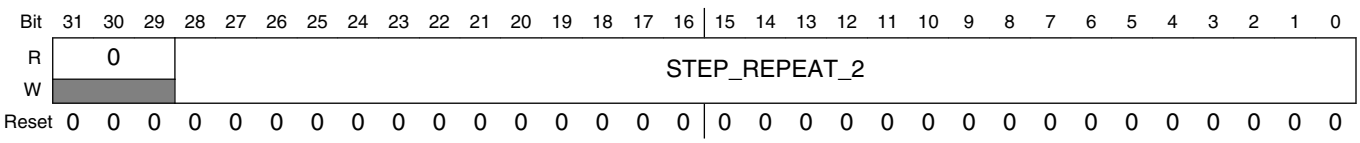


IPU_DC_UGDE2_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
OFFSET_DT_2	This field defines the offset value from which the counter of user general event #2 will start counting from

38.5.372 DC User General Data Event 2Register 3 (IPU_DC_UGDE2_3)

Address: 260_0000h base + 5_81A0h offset = 265_81A0h

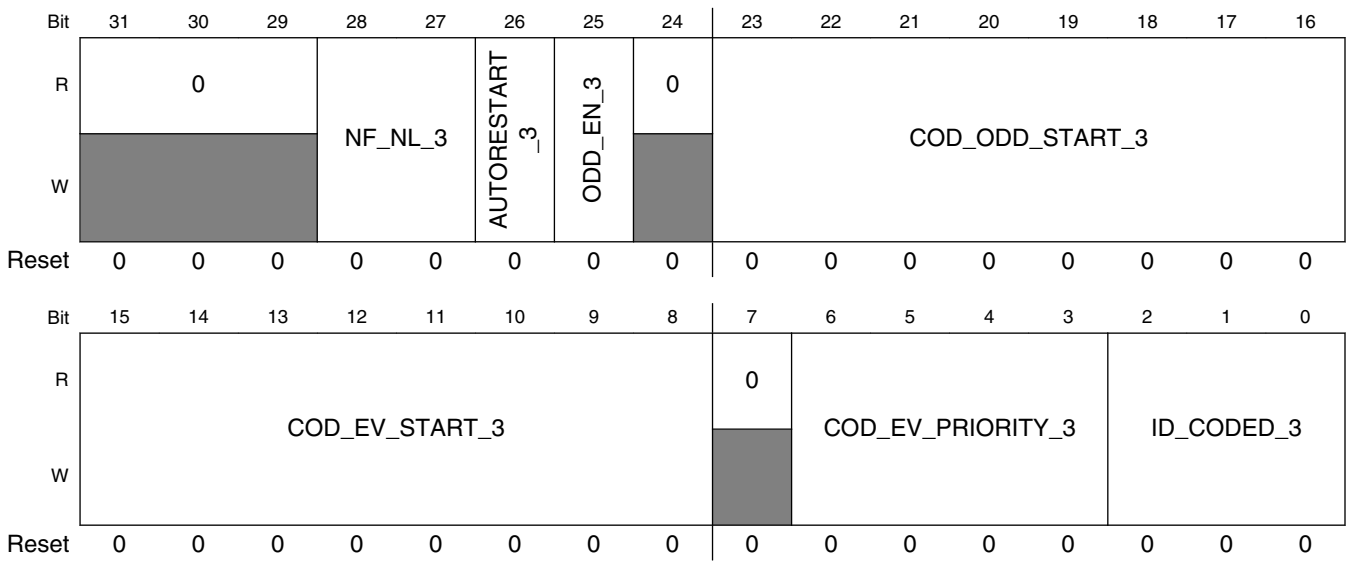


IPU_DC_UGDE2_3 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
STEP_REPEAT_2	When auto reload mode is disabled this field defines the number of events that will be generated by the user general event #2 mechanism

38.5.373 DC User General Data Event 3Register 0 (IPU_DC_UGDE3_0)

Address: 260_0000h base + 5_81A4h offset = 265_81A4h



IPU_DC_UGDE3_0 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–27 NF_NL_3	the user may attach his general event #3 to New-line New-Frame and New-field events. One of these event triggers the user's general event #3's counter. The actual internal trigger is the pixel following the occurrence of the selected event. 00 New Line 01 New Frame 10 New Field 11 Reserved
26 AUTORESTART_3	User's general event #3 auto restart mode 0 disable 1 User's general event #3's counter is automatically restarted.
25 ODD_EN_3	The user's general event #3 may be split into 2 internal signals. One one mode all the events are sent on one signal, on the second mode odd events are sent over one signal while even events are sent over the other signal (ODD_MODE) 1 ODD_MODE is enabled 0 ODD_MODE is disabled
24 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_DC_UGDE3_0 field descriptions (continued)

Field	Description
23–16 COD_ODD_ START_3	This field holds a pointer in the microcode holding the routine to be performed following the user general event #3. When ODD_MODE is enabled, only the odd events will use this pointer When ODD_MODE is disabled this field is ignored
15–8 COD_EV_ START_3	This field holds a pointer in the microcode holding the routine to be performed following the user general event #3. When ODD_MODE is enabled, only the even events will use this pointer When ODD_MODE is disabled, all the events will use this pointer
7 Reserved	This read-only field is reserved and always has the value 0.
6–3 COD_EV_ PRIORITY_3	This field defines the priority of the user general event #3 0000 disable The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
ID_CODED_3	This field defines the number of DC channel number that user's general event #3 will be associated to 000 DC channel_0 001 DC channel_1 010 DC channel_2 011 DC channel_5 (DP_SYNC) 100 DC channel_6 (DP ASYNC) 101 Reserved 110 Reserved 111 Reserved.

38.5.374 DC User General Data Event 3Register 1 (IPU_DC_UGDE3_1)

Address: 260_0000h base + 5_81A8h offset = 265_81A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																STEP_3															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE3_1 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
STEP_3	This field hold the pre defined value that the counter counts too

38.5.375 DC User General Data Event 3Register 2 (IPU_DC_UGDE3_2)

Address: 260_0000h base + 5_81ACh offset = 265_81ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																OFFSET_DT_3															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE3_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
OFFSET_DT_3	This field defines the offset value from which the counter of user general event #3 will start counting from

38.5.376 DC User General Data Event 3Register 2 (IPU_DC_UGDE3_3)

Address: 260_0000h base + 5_81B0h offset = 265_81B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																STEP_REPEAT_3															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE3_3 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
STEP_REPEAT_3	When auto reload mode is disabled this field defines the number of events that will be generated by the user general event #3 mechanism

38.5.377 DC Low Level Access Control Register 0 (IPU_DC_LLA0)

Address: 260_0000h base + 5_81B4h offset = 265_81B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MCU_RS_3_0								MCU_RS_2_0								MCU_RS_1_0								MCU_RS_0_0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_LLA0 field descriptions

Field	Description
31–24 MCU_RS_3_0	This field holds a pointer in the microcode handling the RS_3 routine for the display defined at DISP_ID_8, when in Low level access mode,
23–16 MCU_RS_2_0	This field holds a pointer in the microcode handling the RS_2 routine for the display defined at DISP_ID_8, when in Low level access mode,
15–8 MCU_RS_1_0	This field holds a pointer in the microcode handling the RS_1 routine for the display defined at DISP_ID_8, when in Low level access mode,
MCU_RS_0_0	This field holds a pointer in the microcode handling the RS_0 routine for the display defined at DISP_ID_8, when in Low level access mode,

38.5.378 DC Low Level Access Control Register 1 (IPU_DC_LLA1)

Address: 260_0000h base + 5_81B8h offset = 265_81B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MCU_RS_3_1								MCU_RS_2_1								MCU_RS_1_1								MCU_RS_0_1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_LLA1 field descriptions

Field	Description
31–24 MCU_RS_3_1	This field holds a pointer in the microcode handling the RS_3 routine for the display defined at DISP_ID_9, when in Low level access mode,
23–16 MCU_RS_2_1	This field holds a pointer in the microcode handling the RS_2 routine for the display defined at DISP_ID_9, when in Low level access mode,
15–8 MCU_RS_1_1	This field holds a pointer in the microcode handling the RS_1 routine for the display defined at DISP_ID_9, when in Low level access mode,
MCU_RS_0_1	This field holds a pointer in the microcode handling the RS_0 routine for the display defined at DISP_ID_9, when in Low level access mode,

38.5.379 DC Read Low Level Read Access Control Register 0 (IPU_DC_R_LLA0)

Address: 260_0000h base + 5_81BCh offset = 265_81BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MCU_RS_3_0								MCU_RS_2_0								MCU_RS_R_1_0								MCU_RS_R_0_0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_R_LLA0 field descriptions

Field	Description
31–24 MCU_RS_3_0	This field holds a pointer in the microcode handling the RS_3 routine for the display defined at DISP_ID_8, when in Read Low level access mode,
23–16 MCU_RS_2_0	This field holds a pointer in the microcode handling the RS_2 routine for the display defined at DISP_ID_8, when in Read Low level access mode,
15–8 MCU_RS_R_1_0	This field holds a pointer in the microcode handling the RS_1 routine for the display defined at DISP_ID_8, when in Read Low level access mode,
MCU_RS_R_0_0	This field holds a pointer in the microcode handling the RS_0 routine for the display defined at DISP_ID_8, when in Read Low level access mode,

38.5.380 DC Read Low Level Read Access Control Register1 (IPU_DC_R_LLA1)

Address: 260_0000h base + 5_81C0h offset = 265_81C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MCU_RS_R_3_1								MCU_RS_R_2_1								MCU_RS_R_1_1								MCU_RS_R_0_1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_R_LLA1 field descriptions

Field	Description
31–24 MCU_RS_R_3_1	This field holds a pointer in the microcode handling the RS_3 routine for the display defined at DISP_ID_9, when in Read Low level access mode,
23–16 MCU_RS_R_2_1	This field holds a pointer in the microcode handling the RS_2 routine for the display defined at DISP_ID_9, when in Read Low level access mode,
15–8 MCU_RS_R_1_1	This field holds a pointer in the microcode handling the RS_1 routine for the display defined at DISP_ID_9, when in Read Low level access mode,
MCU_RS_R_0_1	This field holds a pointer in the microcode handling the RS_0 routine for the display defined at DISP_ID_9, when in Read Low level access mode,

38.5.381 DC Write Channel 5 Configuration Register (IPU_DC_WR_CH_ADDR_5_ALT)

Address: 260_0000h base + 5_81C4h offset = 265_81C4h

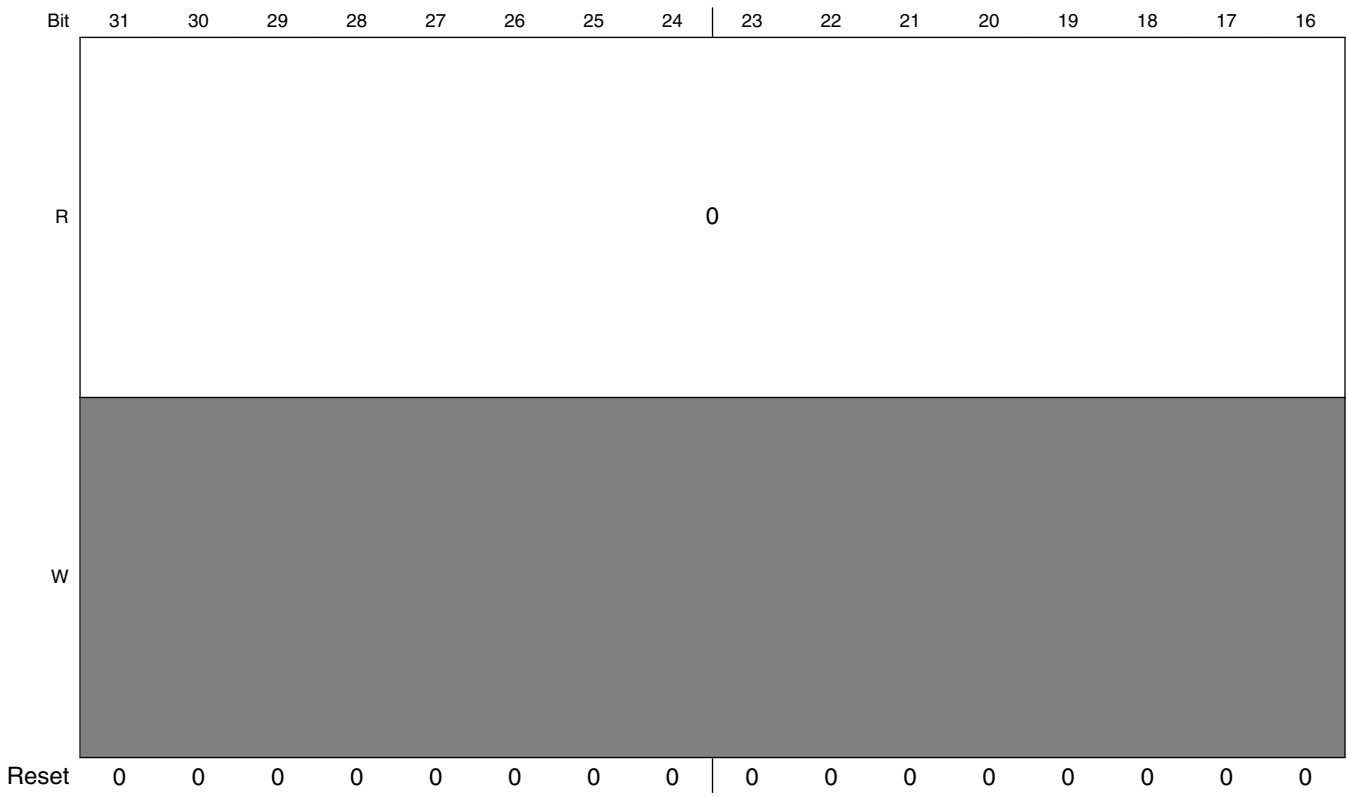
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	0			ST_ADDR_5_ALT																												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

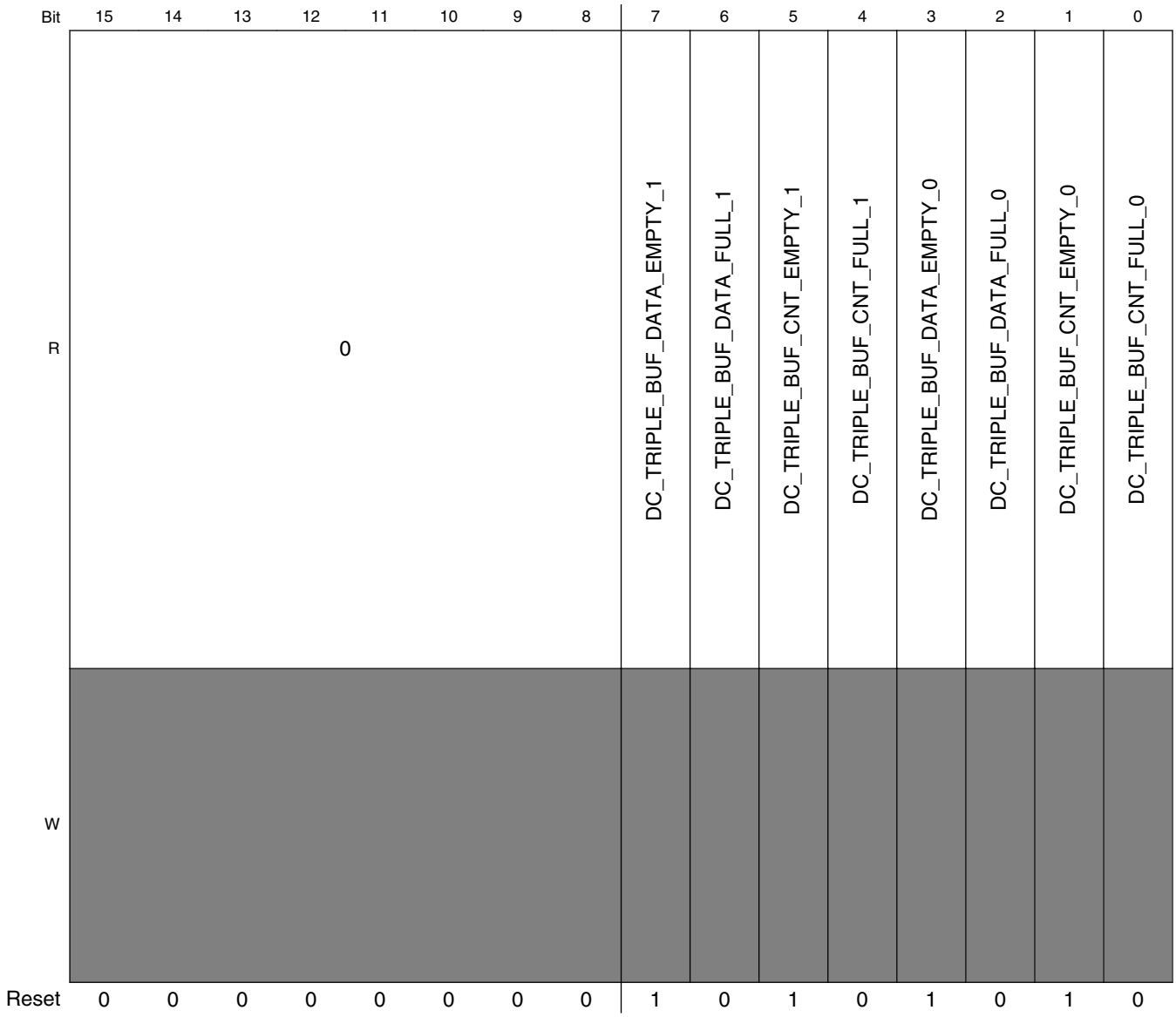
IPU_DC_WR_CH_ADDR_5_ALT field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
ST_ADDR_5_ ALT	This field defines the start address within the display's memory space where the write transactions will be done to for channel #5, when alternate flow is performed via channel #5

38.5.382 DC Status Register (IPU_DC_STAT)

Address: 260_0000h base + 5_81C8h offset = 265_81C8h





IPU_DC_STAT field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.
7 DC_TRIPLE_BUF_DATA_EMPTY_1	This bit indicates a FIFO full state on the DC FIFO accessing DI1 when write to the display flow is used
6 DC_TRIPLE_BUF_DATA_FULL_1	This bit indicates a FIFO empty state on the DC FIFO accessing DI1 when read from the display flow is used

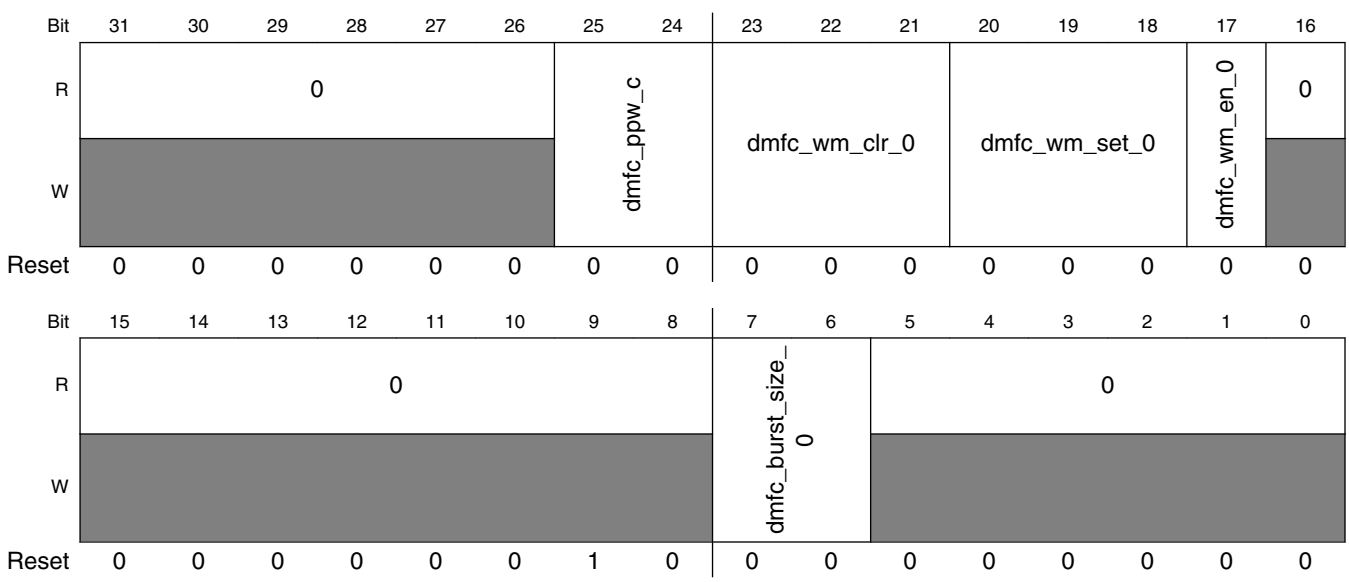
Table continues on the next page...

IPU_DC_STAT field descriptions (continued)

Field	Description
5 DC_TRIPLE_BUF_CNT_EMPTY_1	This bit indicates a FIFO empty state on the DC FIFO accessing DI1 when write to the display flow is used
4 DC_TRIPLE_BUF_CNT_FULL_1	This bit indicates a FIFO full state on the DC FIFO accessing DI1 when write to the display flow is used
3 DC_TRIPLE_BUF_DATA_EMPTY_0	This bit indicates a FIFO empty state on the DC FIFO accessing DI0 when read from the display flow is used
2 DC_TRIPLE_BUF_DATA_FULL_0	This bit indicates a FIFO full state on the DC FIFO accessing DI0 when read from the display flow is used
1 DC_TRIPLE_BUF_CNT_EMPTY_0	This bit indicates a FIFO empty state on the DC FIFO accessing DI0 when write to the display flow is used
0 DC_TRIPLE_BUF_CNT_FULL_0	This bit indicates a FIFO full state on the DC FIFO accessing DI0 when write to the display flow is used

38.5.383 DMFC Read Channel Register (IPU_DMFC_RD_CHAN)

Address: 260_0000h base + 6_0000h offset = 266_0000h



IPU_DMFC_RD_CHAN field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.
25–24 dmfc_ppw_c	Pixel Per Word coded. This field defines the size of the read data from the display. 00 8 bit per pixel 01 16 bit per pixel 10 24 (rgb) bit per pixel or 32 bit per pixel 11 Reserved
23–21 dmfc_wm_clr_0	Watermark Clear This field defines the watermark's level of the DMFC read FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of free bursts at the FIFO (dmfc_wm_clr_0 > dmfc_wm_set_0)
20–18 dmfc_wm_set_0	Watermark Set This field defines the watermark's level of the DMFC read FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of free bursts at the FIFO (dmfc_wm_clr_0 > dmfc_wm_set_0)
17 dmfc_wm_en_0	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
16–8 Reserved	This read-only field is reserved and always has the value 0.
7–6 dmfc_burst_size_0	Read burst Size This field defines the burst size of the DMFC's read accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, going to the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
Reserved	This read-only field is reserved and always has the value 0.

38.5.384 DMFC Write Channel Register (IPU_DMFC_WR_CHAN)

Address: 260_0000h base + 6_0004h offset = 266_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	dmfc_burst_size_2c		dmfc_fifo_size_2c			dmfc_st_addr_2c			dmfc_burst_size_1c		dmfc_fifo_size_1c			dmfc_st_addr_1c		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	dmfc_burst_size_2		dmfc_fifo_size_2			dmfc_st_addr_2			dmfc_burst_size_1		dmfc_fifo_size_1			dmfc_st_addr_1		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DMFC_WR_CHAN field descriptions

Field	Description
31–30 dmfc_burst_size_2c	Burst size of IDMAC's channel 43 This field defines the burst size of the IDMAC's channel 43 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
29–27 dmfc_fifo_size_2c	DMFC FIFO size for IDMAC's channel 43 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 43 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
26–24 dmfc_st_addr_2c	DMFC Start Address for IDMAC's channel 43 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 43. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
23–22 dmfc_burst_size_1c	Burst size of IDMAC's channel 42 This field defines the burst size of the IDMAC's channel 42 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings.

Table continues on the next page...

IPU_DMFC_WR_CHAN field descriptions (continued)

Field	Description
	00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
21–19 dmfc_fifo_size_1c	DMFC FIFO size for IDMAC's channel 42 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 42 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
18–16 dmfc_st_addr_1c	DMFC Start Address for IDMAC's channel 42 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 42. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
15–14 dmfc_burst_size_2	Burst size of IDMAC's channel 41 This field defines the burst size of the IDMAC's channel 41 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
13–11 dmfc_fifo_size_2	DMFC FIFO size for IDMAC's channel 41 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 41 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
10–8 dmfc_st_addr_2	DMFC Start Address for IDMAC's channel 41 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 41. The FIFO is partitioned to 8 equal segments.

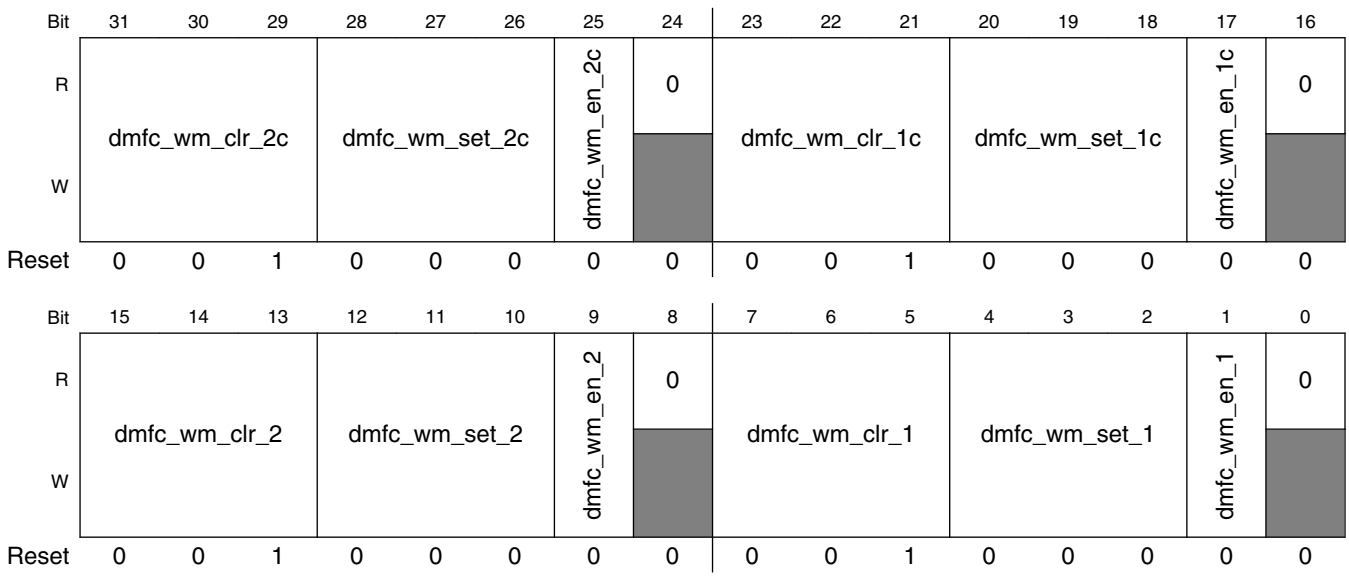
Table continues on the next page...

IPU_DMFC_WR_CHAN field descriptions (continued)

Field	Description
	Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
7–6 dmfc_burst_size_1	Burst size of IDMAC's channel 28 This field defines the burst size of the IDMAC's channel 28 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 1hbit
5–3 dmfc_fifo_size_1	DMFC FIFO size for IDMAC's channel 28 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 28 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
dmfc_st_addr_1	DMFC Start Address for IDMAC's channel 28 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 28. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7

38.5.385 DMFC Write Channel Definition Register (IPU_DMFC_WR_CHAN_DEF)

Address: 260_0000h base + 6_0008h offset = 266_0008h



IPU_DMFC_WR_CHAN_DEF field descriptions

Field	Description
31–29 dmfc_wm_clr_2c	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
28–26 dmfc_wm_set_2c	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
25 dmfc_wm_en_2c	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
24 Reserved	This read-only field is reserved and always has the value 0.
23–21 dmfc_wm_clr_1c	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
20–18 dmfc_wm_set_1c	Watermark Set

Table continues on the next page...

IPU_DMFC_WR_CHAN_DEF field descriptions (continued)

Field	Description
	This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
17 dmfc_wm_en_1c	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
16 Reserved	This read-only field is reserved and always has the value 0.
15–13 dmfc_wm_clr_2	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
12–10 dmfc_wm_set_2	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
9 dmfc_wm_en_2	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
8 Reserved	This read-only field is reserved and always has the value 0.
7–5 dmfc_wm_clr_1	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
4–2 dmfc_wm_set_1	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
1 dmfc_wm_en_1	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
0 Reserved	This read-only field is reserved and always has the value 0.

38.5.386 DMFC Display Processor Channel Register (IPU_DMFC_DP_CHAN)

Address: 260_0000h base + 6_000Ch offset = 266_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	dmfc_burst_size_6f		dmfc_fifo_size_6f			dmfc_st_addr_6f			dmfc_burst_size_6b		dmfc_fifo_size_6b			dmfc_st_addr_6b		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	dmfc_burst_size_5f		dmfc_fifo_size_5f			dmfc_st_addr_5f			dmfc_burst_size_5b		dmfc_fifo_size_5b			dmfc_st_addr_5b		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DMFC_DP_CHAN field descriptions

Field	Description
31–30 dmfc_burst_size_6f	Burst size of IDMAC's channel 29 This field defines the burst size of the IDMAC's channel 29 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
29–27 dmfc_fifo_size_6f	DMFC FIFO size for IDMAC's channel 29 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 29 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
26–24 dmfc_st_addr_6f	DMFC Start Address for IDMAC's channel 29 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 29. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7

Table continues on the next page...

IPU_DMFC_DP_CHAN field descriptions (continued)

Field	Description
23–22 dmfc_burst_size_6b	Burst size of IDMAC's channel 24 This field defines the burst size of the IDMAC's channel 24 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
21–19 dmfc_fifo_size_6b	DMFC FIFO size for IDMAC's channel 24 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 24 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
18–16 dmfc_st_addr_6b	DMFC Start Address for IDMAC's channel 24 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 24. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
15–14 dmfc_burst_size_5f	Burst size of IDMAC's channel 27 This field defines the burst size of the IDMAC's channel 27 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
13–11 dmfc_fifo_size_5f	DMFC FIFO size for IDMAC's channel 27 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 27 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel

Table continues on the next page...

IPU_DMFC_DP_CHAN field descriptions (continued)

Field	Description
10–8 dmfc_st_addr_5f	DMFC Start Address for IDMAC's channel 27 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 27. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
7–6 dmfc_burst_size_5b	Burst size of IDMAC's channel 23 This field defines the burst size of the IDMAC's channel 23 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
5–3 dmfc_fifo_size_5b	DMFC FIFO size for IDMAC's channel 23 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 23 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
dmfc_st_addr_5b	DMFC Start Address for IDMAC's channel 23 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 23. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7

38.5.387 DMFC Display Processor Channel Definition Register (IPU_DMFC_DP_CHAN_DEF)

Address: 260_0000h base + 6_0010h offset = 266_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	dmfc_wm_clr_6f			dmfc_wm_set_6f			dmfc_wm_en_6f	0	dmfc_wm_clr_6b			dmfc_wm_set_6b			dmfc_wm_en_6b	0
W																
Reset	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	dmfc_wm_clr_5f			dmfc_wm_set_5f			dmfc_wm_en_5f	0	dmfc_wm_clr_5b			dmfc_wm_set_5b			dmfc_wm_en_5b	0
W																
Reset	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

IPU_DMFC_DP_CHAN_DEF field descriptions

Field	Description
31–29 dmfc_wm_clr_6f	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
28–26 dmfc_wm_set_6f	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
25 dmfc_wm_en_6f	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
24 Reserved	This read-only field is reserved and always has the value 0.
23–21 dmfc_wm_clr_6b	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
20–18 dmfc_wm_set_6b	Watermark Set

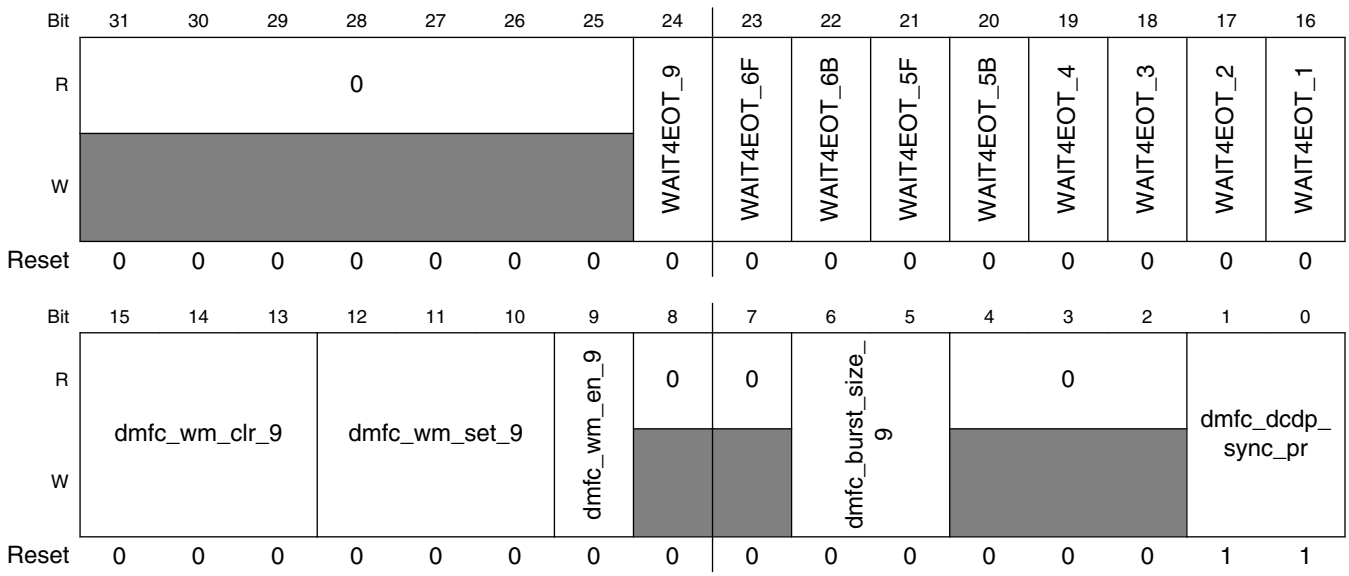
Table continues on the next page...

IPU_DMFC_DP_CHAN_DEF field descriptions (continued)

Field	Description
	This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
17 dmfc_wm_en_6b	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
16 Reserved	This read-only field is reserved and always has the value 0.
15–13 dmfc_wm_clr_5f	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
12–10 dmfc_wm_set_5f	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
9 dmfc_wm_en_5f	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
8 Reserved	This read-only field is reserved and always has the value 0.
7–5 dmfc_wm_clr_5b	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
4–2 dmfc_wm_set_5b	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
1 dmfc_wm_en_5b	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
0 Reserved	This read-only field is reserved and always has the value 0.

38.5.388 DMFC General 1 Register (IPU_DMFC_GENERAL_1)

Address: 260_0000h base + 6_0014h offset = 266_0014h



IPU_DMFC_GENERAL_1 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24 WAIT4EOT_9	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #9 is in wait4eot mode 0 FIFO #9 is in normal mode
23 WAIT4EOT_6F	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #6F is in wait4eot mode 0 FIFO #6F is in normal mode
22 WAIT4EOT_6B	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #6B is in wait4eot mode 0 FIFO #6B is in normal mode

Table continues on the next page...

IPU_DMFC_GENERAL_1 field descriptions (continued)

Field	Description
21 WAIT4EOT_5F	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #5F is in wait4eot mode 0 FIFO #5F is in normal mode
20 WAIT4EOT_5B	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #5B is in wait4eot mode 0 FIFO #5B is in normal mode
19 WAIT4EOT_4	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #4 is in wait4eot mode 0 FIFO #4 is in normal mode
18 WAIT4EOT_3	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #3 is in wait4eot mode 0 FIFO #3 is in normal mode
17 WAIT4EOT_2	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #2 is in wait4eot mode 0 FIFO #2 is in normal mode
16 WAIT4EOT_1	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #1 is in wait4eot mode 0 FIFO #1 is in normal mode
15–13 dmfc_wm_clr_9	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)

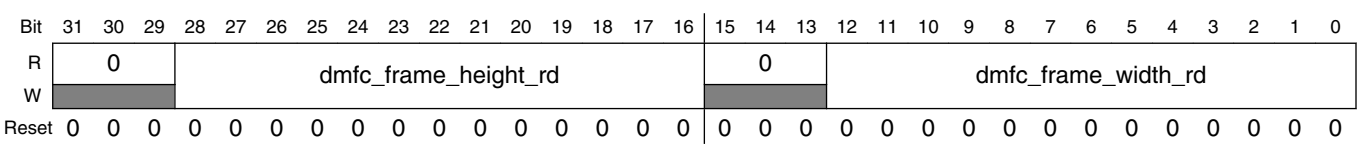
Table continues on the next page...

IPU_DMFC_GENERAL_1 field descriptions (continued)

Field	Description
12–10 dmfc_wm_set_9	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
9 dmfc_wm_en_9	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
8 Reserved	This read-only field is reserved and always has the value 0.
7 Reserved	This read-only field is reserved and always has the value 0.
6–5 dmfc_burst_size_9	Burst size of IDMAC's channel 44 This field defines the burst size of the IDMAC's channel 44 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. This channel is targeted for MASK - the FIFO size is always 32X128; The base address is always the upper half of the 8th segment 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
4–2 Reserved	This read-only field is reserved and always has the value 0.
dmfc_dcdp_sync_pr	DMFC's memory access priority settings for simultaneous synchronous flows from DC & DP 00 Forbidden - should not be used. 01 DC has higher priority over DP 10 DP has higher priority over DC 11 Round Robin

38.5.389 DMFC General 2 Register (IPU_DMFC_GENERAL_2)

Address: 260_0000h base + 6_0018h offset = 266_0018h



IPU_DMFC_GENERAL_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–16 dmfc_frame_ height_rd	Frame height for read channel from the display to the IDMAC; Units are pixels
15–13 Reserved	This read-only field is reserved and always has the value 0.
dmfc_frame_ width_rd	Frame width for read channel from the display to the IDMAC; Units are pixels

38.5.390 DMFC IC Interface Control Register (IPU_DMFC_IC_CTRL)

Address: 260_0000h base + 6_001Ch offset = 266_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	dmfc_ic_frame_height_rd													dmfc_ic_frame_ width_rd		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	dmfc_ic_frame_width_rd										dmfc_ic_ ppw_c		0	dmfc_ic_in_port		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

IPU_DMFC_IC_CTRL field descriptions

Field	Description
31–19 dmfc_ic_frame_ height_rd	Frame's height for the channel coming from IC. Units are lines
18–6 dmfc_ic_frame_ width_rd	Frame's width for the channel coming from IC. Units are pixels
5–4 dmfc_ic_ppw_c	Pixel Per Word coded from IC. This field defines the size of the data coming from the IC. 00 8 bit per pixel 01 16 bit per pixel 10 24 bit per pixel 11 Reserved
3 Reserved	This read-only field is reserved and always has the value 0.
dmfc_ic_in_port	DMFC input port

Table continues on the next page...

IPU_DMFC_IC_CTRL field descriptions (continued)

Field	Description
	When data is coming from the IC, the IC channel replaces one of the IDMAC's channels connected to the DMFC. This field defines which IDMAC's channel is replaced by the IC channel. 000 CH28 001 CH41 010 Reserved, IC channel is disabled 011 Reserved, IC channel is disabled 100 CH23 101 CH27 110 CH24 111 CH29

38.5.391 DMFC Write Channel Alternate Register (IPU_DMFC_WR_CHAN_ALT)

Address: 260_0000h base + 6_0020h offset = 266_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	dmfc_burst_size_2_alt		dmfc_fifo_size_2_alt			dmfc_st_addr_2_alt			0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DMFC_WR_CHAN_ALT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–14 dmfc_burst_size_2_alt	Burst size of IDMAC's channel 41 (for alternate flow) This field defines the burst size of the IDMAC's channel 41 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
13–11 dmfc_fifo_size_2_alt	DMFC FIFO size for IDMAC's channel 41 (for alternate flow) This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 41 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel

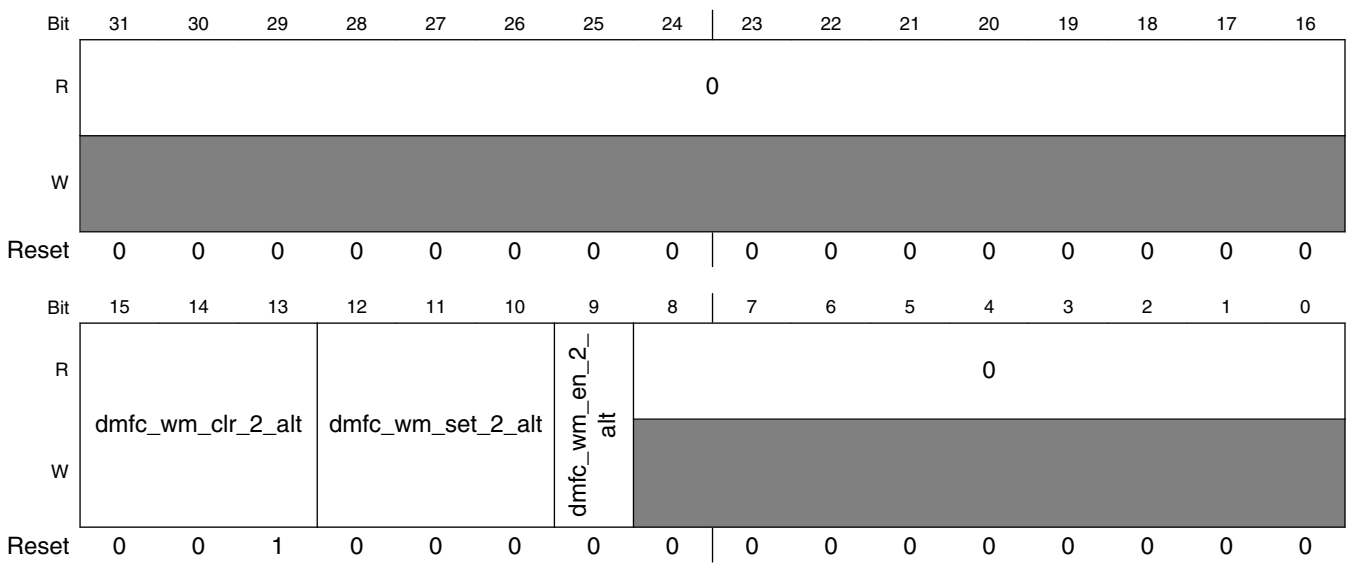
Table continues on the next page...

IPU_DMFC_WR_CHAN_ALT field descriptions (continued)

Field	Description
	011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
10–8 dmfc_st_addr_2_alt	DMFC Start Address for IDMAC's channel 41 (for alternate flow) This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 41. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
Reserved	This read-only field is reserved and always has the value 0.

38.5.392 DMFC Write Channel Definition Alternate Register (IPU_DMFC_WR_CHAN_DEF_ALT)

Address: 260_0000h base + 6_0024h offset = 266_0024h



IPU_DMFC_WR_CHAN_DEF_ALT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.

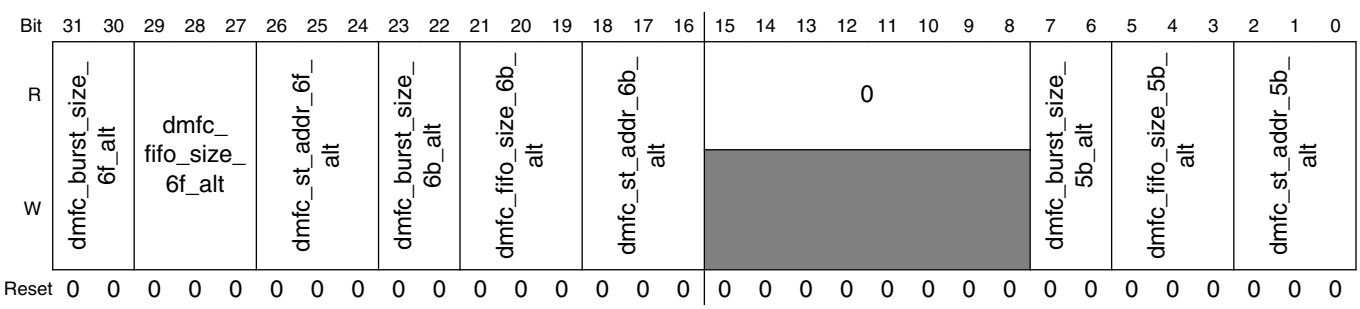
Table continues on the next page...

IPU_DMFC_WR_CHAN_DEF_ALT field descriptions (continued)

Field	Description
15–13 dmfc_wm_clr_2_alt	Watermark Clear (for alternate flow) This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
12–10 dmfc_wm_set_2_alt	Watermark Set (for alternate flow) This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
9 dmfc_wm_en_2_alt	Watermark enable. (for alternate flow) This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
Reserved	This read-only field is reserved and always has the value 0.

38.5.393 DMFC MFC Display Processor Channel Alternate Register (IPU_DMFC_DP_CHAN_ALT)

Address: 260_0000h base + 6_0028h offset = 266_0028h



IPU_DMFC_DP_CHAN_ALT field descriptions

Field	Description
31–30 dmfc_burst_size_6f_alt	Burst size of IDMAC's channel 29 (for alternate flow) This field defines the burst size of the IDMAC's channel 29 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
29–27 dmfc_fifo_size_6f_alt	DMFC FIFO size for IDMAC's channel 29 (for alternate flow) This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 29 000 All (512X128 words) the DMFC's FIFO is allocated to this channel

Table continues on the next page...

IPU_DMFC_DP_CHAN_ALT field descriptions (continued)

Field	Description
	001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
26–24 dmfc_st_addr_6f_alt	DMFC Start Address for IDMAC's channel 29 (for alternate flow) This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 29. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
23–22 dmfc_burst_size_6b_alt	Burst size of IDMAC's channel 24 (for alternate flow) This field defines the burst size of the IDMAC's channel 24 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
21–19 dmfc_fifo_size_6b_alt	DMFC FIFO size for IDMAC's channel 24 (for alternate flow) This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 24 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
18–16 dmfc_st_addr_6b_alt	DMFC Start Address for IDMAC's channel 24 (for alternate flow) This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 24. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
15–8 Reserved	This read-only field is reserved and always has the value 0.

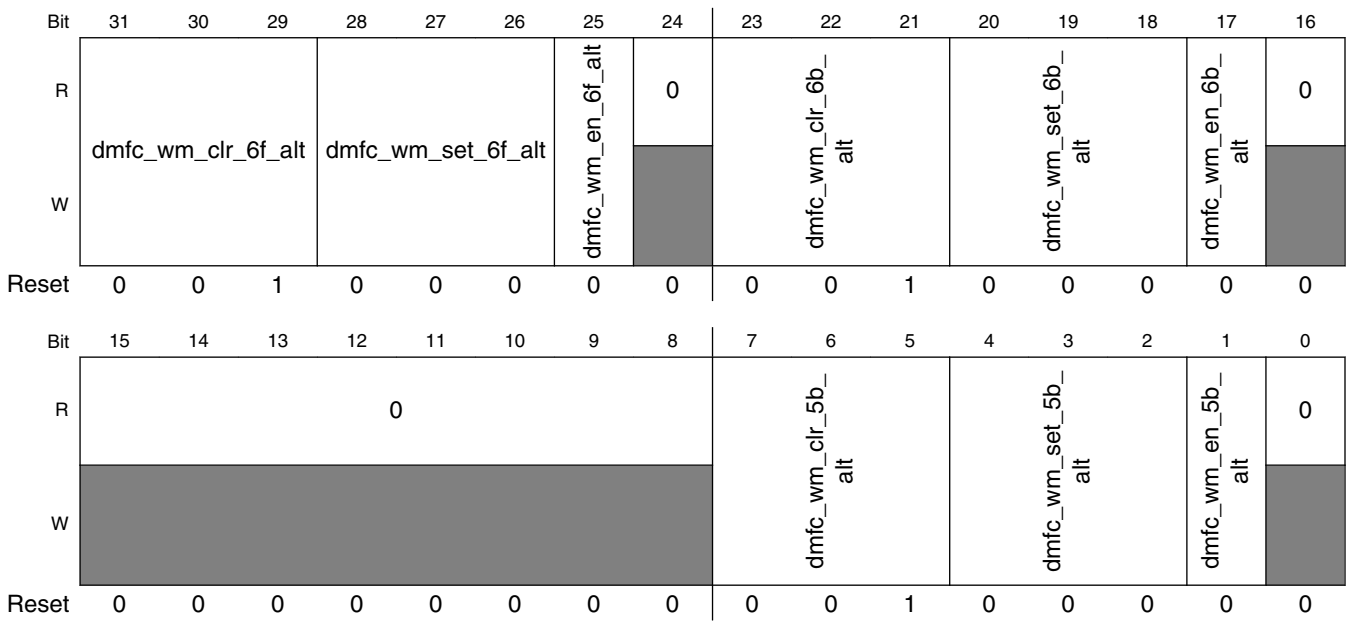
Table continues on the next page...

IPU_DMFC_DP_CHAN_ALT field descriptions (continued)

Field	Description
7–6 dmfc_burst_size_5b_alt	Burst size of IDMAC's channel 23 (for alternate flow) This field defines the burst size of the IDMAC's channel 23 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
5–3 dmfc_fifo_size_5b_alt	DMFC FIFO size for IDMAC's channel 23 (for alternate flow) This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 23 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
dmfc_st_addr_5b_alt	DMFC Start Address for IDMAC's channel 23 (for alternate flow) This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 23. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7

38.5.394 DMFC Display Channel Definition Alternate Register (IPU_DMFC_DP_CHAN_DEF_ALT)

Address: 260_0000h base + 6_002Ch offset = 266_002Ch



IPU_DMFC_DP_CHAN_DEF_ALT field descriptions

Field	Description
31–29 dmfc_wm_clr_6f_alt	Watermark Clear (for alternate flow) This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
28–26 dmfc_wm_set_6f_alt	Watermark Set (for alternate flow) This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
25 dmfc_wm_en_6f_alt	Watermark enable. (for alternate flow) This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
24 Reserved	This read-only field is reserved and always has the value 0.
23–21 dmfc_wm_clr_6b_alt	Watermark Clear (for alternate flow) This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)

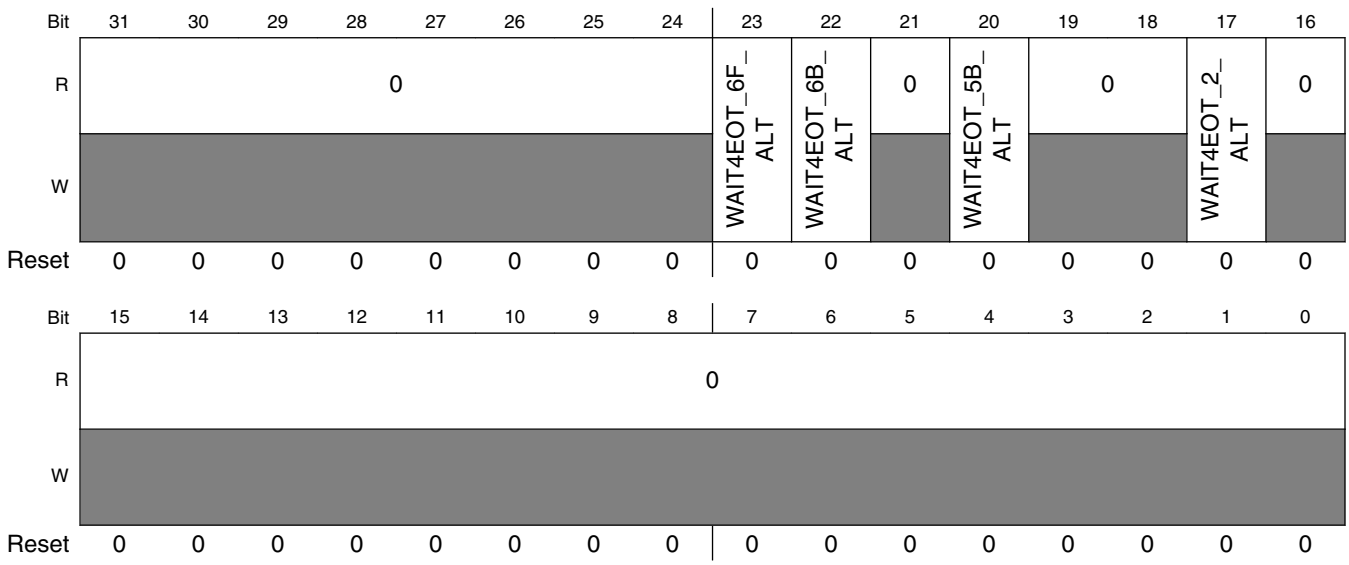
Table continues on the next page...

IPU_DMFC_DP_CHAN_DEF_ALT field descriptions (continued)

Field	Description
20–18 dmfc_wm_set_ 6b_alt	Watermark Set (for alternate flow) This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
17 dmfc_wm_en_ 6b_alt	Watermark enable. (for alternate flow) This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
16–8 Reserved	This read-only field is reserved and always has the value 0.
7–5 dmfc_wm_clr_ 5b_alt	Watermark Clear (for alternate flow) This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
4–2 dmfc_wm_set_ 5b_alt	Watermark Set (for alternate flow) This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
1 dmfc_wm_en_ 5b_alt	Watermark enable. (for alternate flow) This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
0 Reserved	This read-only field is reserved and always has the value 0.

38.5.395 DMFC General 1 Alternate Register (IPU_DMFC_GENERAL1_ALT)

Address: 260_0000h base + 6_0030h offset = 266_0030h



IPU_DMFC_GENERAL1_ALT field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value 0.
23 WAIT4EOT_6F_ALT	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #6F is in wait4eot mode (for alternate flow) 0 FIFO #6F is in normal mode (for alternate flow)
22 WAIT4EOT_6B_ALT	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #6B is in wait4eot mode (for alternate flow) 0 FIFO #6B is in normal mode (for alternate flow)
21 Reserved	This read-only field is reserved and always has the value 0.
20 WAIT4EOT_5B_ALT	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode.

Table continues on the next page...

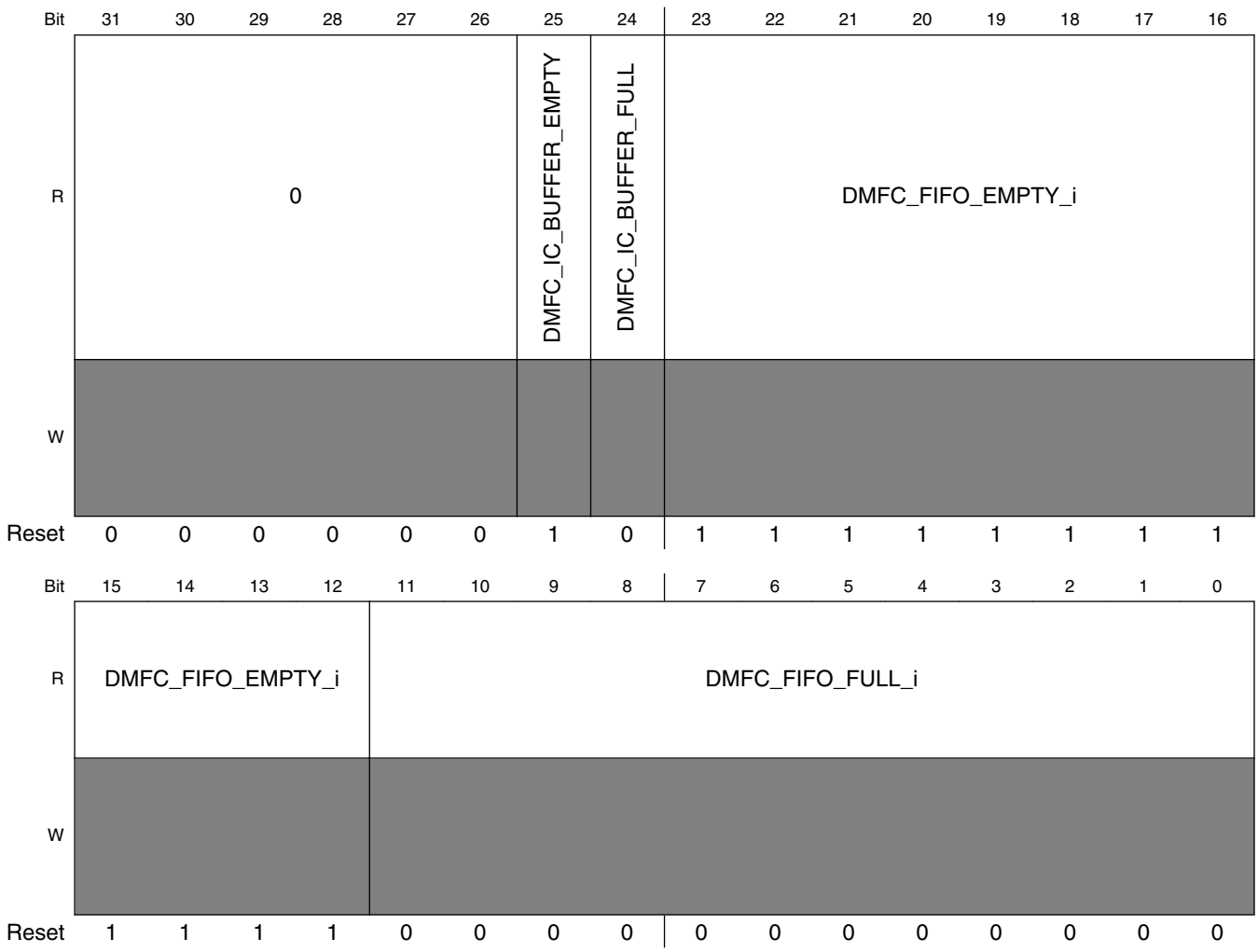
IPU_DMFC_GENERAL1_ALT field descriptions (continued)

Field	Description
	1 FIFO #5B is in wait4eot mode (for alternate flow) 0 FIFO #5B is in normal mode (for alternate flow)
19–18 Reserved	This read-only field is reserved and always has the value 0.
17 WAIT4EOT_2_ALT	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #2 is in wait4eot mode (for alternate flow) 0 FIFO #2 is in normal mode (for alternate flow)
Reserved	This read-only field is reserved and always has the value 0.

38.5.396 DMFC Status Register (IPU_DMFC_STAT)

This register contains DMFC's status bits. All the bits in this register are read-only.

Address: 260_0000h base + 6_0034h offset = 266_0034h



IPU_DMFC_STAT field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.
25 DMFC_IC_ BUFFER_ EMPTY	This bit indicates on a IC FIFO, inside the DMFC, empty condition. 0 IC FIFO not empty 1 IC FIFO is empty

Table continues on the next page...

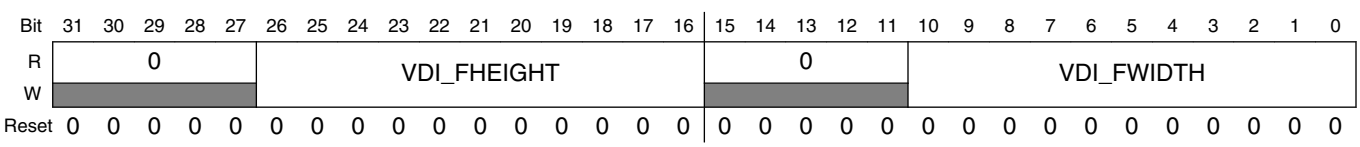
IPU_DMFC_STAT field descriptions (continued)

Field	Description
24 DMFC_IC_ BUFFER_FULL	This bit indicates on a IC FIFO, inside the DMFC, full condition. 0 IC FIFO not full 1 IC FIFO is full
23–12 DMFC_FIFO_ EMPTY_i	This bit indicates on a DMFC FIFO#<i> empty condition. Mapping of these bit to an actual FIFO number is as follows: bit 0 => 0 bit 1=> 1 bit 2 => 2 bit 3 =>1c bit 4 => 2c bit 5 => 5b bit 6 => 5f bit 7 => 6b bit 8 => 6f bit 9 => 9 bit 10 => 10 (ARM platform access) bit 11 => 11 (ARM platform access) 0 FIFO #<i> is not empty 1 FIFO #<i> is empty
DMFC_FIFO_ FULL_i	This bit indicates on a DMFC FIFO#<i> full condition. Mapping of these bit to an actual FIFO number is as follows: bit 0 => 0 bit 1=> 1 bit 2 => 2 bit 3 =>1c bit 4 => 2c bit 5 => 5b bit 6 => 5f bit 7 => 6b bit 8 => 6f bit 9 => 9 bit 10 => 10 (ARM platform access) bit 11 => 11 (ARM platform access) 0 FIFO #<i> is not full 1 FIFO #<i> is full

38.5.397 VDI Field Size Register (IPU_VDI_FSIZE)

The register used to control size of VDIC input fields.

Address: 260_0000h base + 6_8000h offset = 266_8000h



IPU_VDI_FSIZE field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26–16 VDI_FHEIGHT	Frame height The value to be written to this register is the frame's height minus 1. The frame height should not be smaller than 16. When VDI_CMB_EN bit is clear: • The frame height should not be greater than 1080. • The frame's height must be even (which means that both fields have the same height) • The frame's height in 4:2:0 format, must be multiple of 4 (which means that both chroma fields have the same height)

Table continues on the next page...

IPU_VDI_FSIZE field descriptions (continued)

Field	Description
	When VDI_CMB_EN bit is set: • The frame height should not be greater than 1200.
15–11 Reserved	This read-only field is reserved and always has the value 0.
VDI_FWIDTH	Frame width. The value to be written to this register is the frame's width minus 1. The Frame width should not be smaller than 16. The width must be even. When VDI_CMB_EN bit is clear • The Frame width should not be greater than 720968. When VDI_CMB_EN bit is set: • The Frame width should not be greater than 1920.

38.5.398 VDI Control Register (IPU_VDI_C)

The register used to control modes of operations of VDIC module.

Address: 260_0000h base + 6_8004h offset = 266_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-	-	0		VDI_VWM3_CLR				VDI_VWM3_SET		VDI_VWM1_CLR			VDI_VWM1_SET		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VDI_BURST_SIZE3				VDI_BURST_SIZE2				VDI_BURST_SIZE1				VDI_MOT_SEL		VDI_CH_422	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_VDI_C field descriptions

Field	Description
31 -	VDIC top filed (automatic) This defines what would be the top field to be processed when the data is coming from the CSI 0 top field is field 0 1 top field is field 1
30 -	VDIC top filed (manual) This defines what would be the next top field to be processed when the data is coming from the memory

Table continues on the next page...

IPU_VDI_C field descriptions (continued)

Field	Description
	0 top field is field 0 1 top field is field 1
29–28 Reserved	This read-only field is reserved and always has the value 0.
27–25 VDI_VWM3_CLR	VDIC WaterMark "clear" level for channel 3. 0 clear watermark level when FIFO3 is full on 1/8 of their size. 1 clear watermark level when FIFO3 is full on 2/8 of their size. 7 clear watermark level when FIFO3 is full.
24–22 VDI_VWM3_SET	VDIC WaterMark "set" level for channel 3. 0 set watermark level when FIFO3 is full on 1/8 of their size. 1 set watermark level when FIFO3 is full on 2/8 of their size. 7 set watermark level when FIFO3 is full.
21–19 VDI_VWM1_CLR	VDIC WaterMark "clear" level for channel 1 or channel 4 (channels 1 and 4 are not working simultaneously). 0 clear watermark level when FIFO1 is full on 1/8 of their size. 1 clear watermark level when FIFO1 is full on 2/8 of their size. 7 clear watermark level when FIFO1 is full.
18–16 VDI_VWM1_SET	VDIC WaterMark "set" level for channel 1 or channel 2 (channels 1 and 4 are not working simultaneously). 0 set watermark level when FIFO1 is full on 1/8 of their size. 1 set watermark level when FIFO1 is full on 2/8 of their size. 7 set watermark level when FIFO1 is full.
15–12 VDI_BURST_ SIZE3	Burst Size for channel 3. The VDIC's burst size is restrict by the IDMAC's restriction - This value must match the IDMAC settings and follow the IDMAC's restrictions 0 Burst size is 1 access. 1 Burst size is 2 accesses. 15 Burst size is 16 accesses.
11–8 VDI_BURST_ SIZE2	Burst Size for channel 2. The VDIC's burst size is restrict by the IDMAC's restriction - This value must match the IDMAC settings and follow the IDMAC's restrictions 0 Burst size is 1 access. 1 Burst size is 2 accesses. 15 Burst size is 16 accesses.
7–4 VDI_BURST_ SIZE1	Burst Size for channels 1 or 4 (channels 1 and 4 are not working simultaneously). The VDIC's burst size is restrict by the IDMAC's restriction - This value must match the IDMAC settings and follow the IDMAC's restrictions 0 Burst size is 1 access. 1 Burst size is 2 accesses. 15 Burst size is 16 accesses.
3–2 VDI_MOT_SEL	Motion select.

Table continues on the next page...

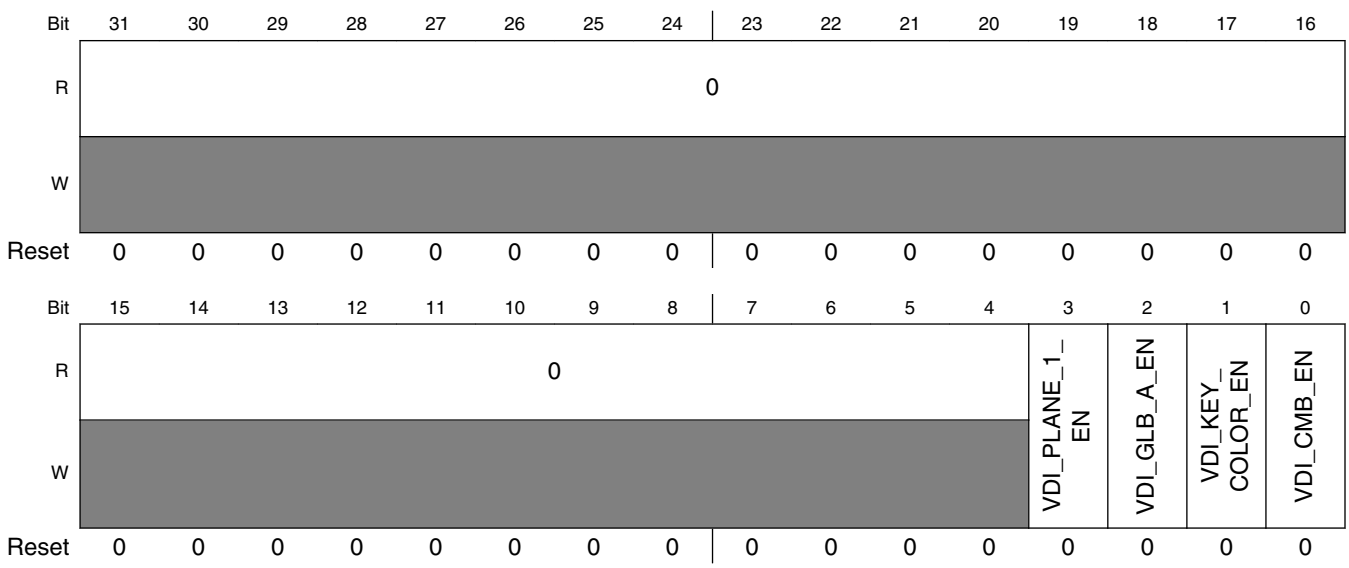
IPU_VDI_C field descriptions (continued)

Field	Description
	0 Motion determined by ROM "1" (shared toward medium/high motion). 1 Motion determined by ROM "2" (This option will not work well for high motion). 2 Full motion, only vertical filter is used 3 Forbidden.
1 VDI_CH_422	Chroma format at input and output of VDIC. 0 Chroma format is 420. 1 Chroma format is 422.
0 Reserved	This read-only field is reserved and always has the value 0.

38.5.399 VDI Control Register 2 (IPU_VDI_C2_)

The register used to control modes of operations of VDIC module.

Address: 260_0000h base + 6_8008h offset = 266_8008h



IPU_VDI_C2_ field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value 0.
3 VDI_PLANE_1_EN	Plane 1 enable 0 plane #1 is disabled 1 plane #1 is enabled

Table continues on the next page...

IPU_VDI_C2_ field descriptions (continued)

Field	Description
2 VDI_GLB_A_EN	Global alpha enable 0 Alpha is local 1 Alpha is global
1 VDI_KEY_ COLOR_EN	Key Color Enable 0 Key Color disabled. 1 Key color enabled
0 VDI_CMB_EN	Combining enable 0 Combining disabled. The VDIC works in de-interlacing mode 1 Combining enabled. The de-interlacing mode is not functional

38.5.400 VDI Combining Parameters Register 1 (IPU_VDI_CMDP_1)

The register holds combining paramemters.

Address: 260_0000h base + 6_800Ch offset = 266_800Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VDI_ALPHA								VDI_KEY_COLOR_R								VDI_KEY_COLOR_G								VDI_KEY_COLOR_B							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

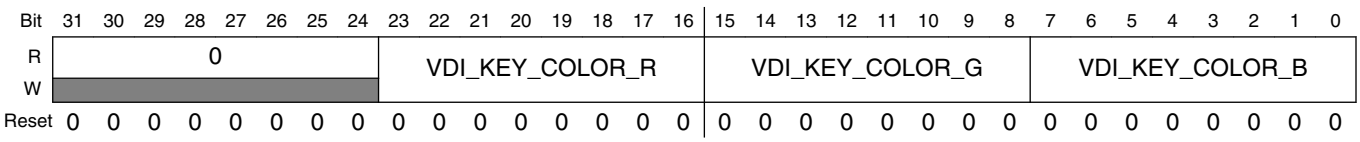
IPU_VDI_CMDP_1 field descriptions

Field	Description
31–24 VDI_ALPHA	Global Alpha Actual value of the alpha is VDI_ALPHA + VDI_ALPHA[7]
23–16 VDI_KEY_ COLOR_R	Red component of Key Color
15–8 VDI_KEY_ COLOR_G	Green component of Key Color
VDI_KEY_ COLOR_B	Blue component of Key Color

38.5.401 VDI Combining Parameters Register 2 (IPU_VDI_CMDP_2)

The register holds combining paramemters.

Address: 260_0000h base + 6_8010h offset = 266_8010h



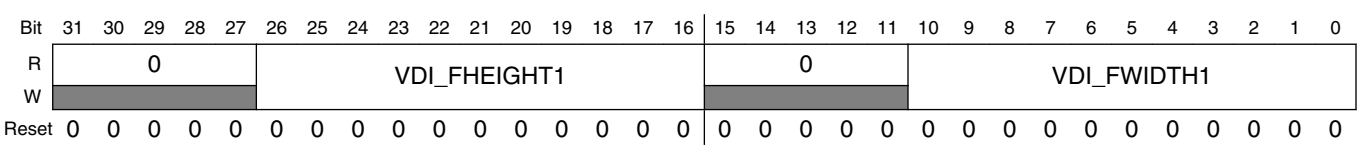
IPU_VDI_CMDP_2 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value 0.
23–16 VDI_KEY_ COLOR_R	Red component of background Color
15–8 VDI_KEY_ COLOR_G	Green component of background Color
VDI_KEY_ COLOR_B	Blue component of background Color

38.5.402 VDI Plane Size Register 1 (IPU_VDI_PS_1)

The register holds the plane size's paramemters.

Address: 260_0000h base + 6_8014h offset = 266_8014h



IPU_VDI_PS_1 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

IPU_VDI_PS_1 field descriptions (continued)

Field	Description
26–16 VDI_FHEIGHT1	Plane 1 height The value to be written to this register is the plane's height minus 1. The Plane height should not be smaller than 16. The Plane height should not be greater than 1200. The Plane's height must be even
15–11 Reserved	This read-only field is reserved and always has the value 0.
VDI_FWIDTH1	Plane 1 width. The value to be written to this register is the plane's width minus 1. The Plane width should not be smaller than 16. The Plane width should not be greater than 1920. The width must be even.

38.5.403 VDI Plane Size Register 2 (IPU_VDI_PS_2)

The register holds the plane's offset paramemters.

Address: 260_0000h base + 6_8018h offset = 266_8018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					VDI_OFFSET_VER1											0					VDI_OFFSET_HOR1										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

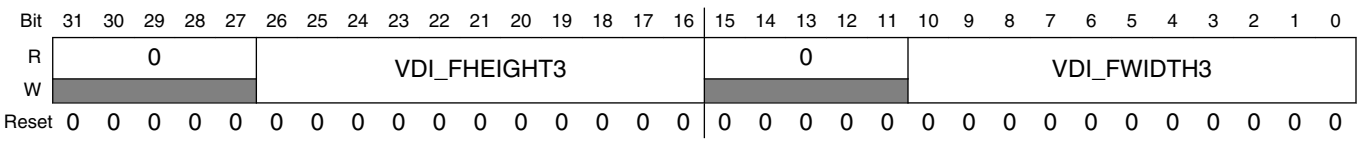
IPU_VDI_PS_2 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26–16 VDI_OFFSET_ VER1	Vertical offset of plane 1
15–11 Reserved	This read-only field is reserved and always has the value 0.
VDI_OFFSET_ HOR1	Horizontal offset of plane 1

38.5.404 VDI Plane Size Register 3 (IPU_VDI_PS_3)

The register holds the plane size's paramemters.

Address: 260_0000h base + 6_801Ch offset = 266_801Ch



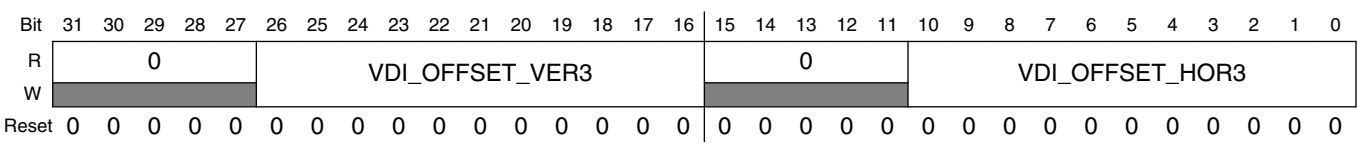
IPU_VDI_PS_3 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26–16 VDI_FHEIGHT3	Plane 3 height The value to be written to this register is the plane's height minus 1. The Plane height should not be smaller than 16. The Plane height should not be greater than 1200. The Plane's height must be even
15–11 Reserved	This read-only field is reserved and always has the value 0.
VDI_FWIDTH3	Plane 3 width. The value to be written to this register is the plane's width minus 1. The Plane width should not be smaller than 16. The Plane width should not be greater than 1920. The width must be even.

38.5.405 VDI Plane Size Register 4 (IPU_VDI_PS_4)

The register holds the plane's offset paramemters.

Address: 260_0000h base + 6_8020h offset = 266_8020h



IPU_VDI_PS_4 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26–16 VDI_OFFSET_VER3	Vertical offset of plane 3
15–11 Reserved	This read-only field is reserved and always has the value 0.
VDI_OFFSET_HOR3	Horizontal offset of plane 3

Chapter 39
Keypad Port (KPP)

39.1 Overview

The Keypad Port (KPP) is a 16-bit peripheral that can be used as a keypad matrix interface or as general purpose input/output (I/O).

The figure below shows the KPP block diagram. The KPP provides interface for the keypad matrix with 2-point contact or 3-point contact keys. The KPP is designed to simplify the software task of scanning a keypad matrix. With appropriate software support, the KPP is capable of detecting, debouncing, and decoding one or multiple keys pressed simultaneously on the keypad.

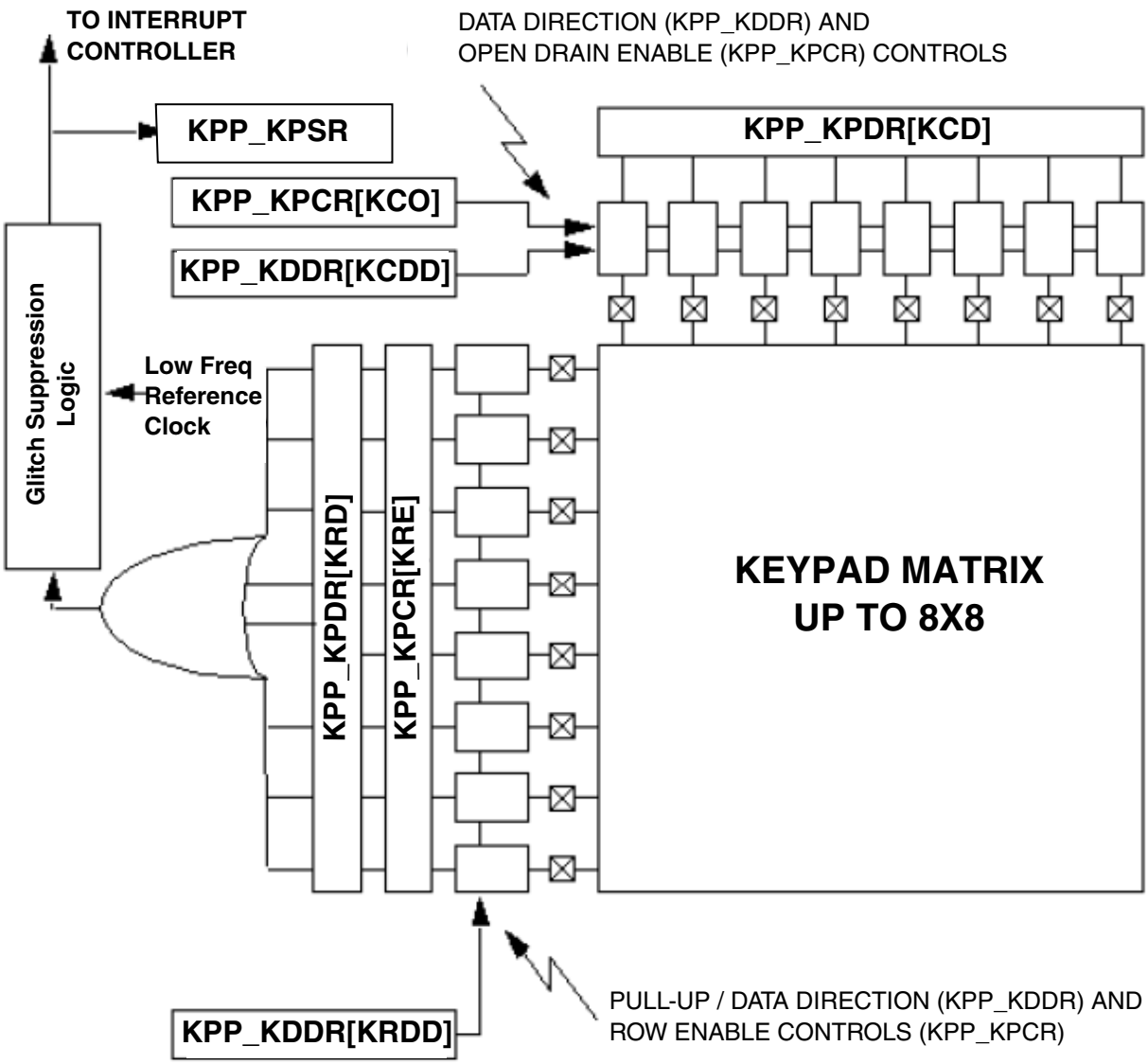


Figure 39-1. KPP Peripheral Block Diagram

39.1.1 Features

The KPP includes these distinctive features:

- Supports up to an 8 x 8 external key pad matrix
- Port pins can be used as general purpose I/O
- Open drain design
- Glitch suppression circuit design
- Multiple-key detection
- Long key-press detection
- Standby key-press detection
- Synchronizer chain clear
- Supports a 2-point and 3-point contact key matrix

39.1.2 Modes and Operations

This block supports the following modes:

- Run Mode-This is the normal functional mode in which the KPP can detect any key press event.
- Low Power Mode-The keypad can detect any key press even in low power modes (when there is no MCU clock).

39.2 Clocks

The table found here describes the clock sources for KPP.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 39-1. KPP Clocks

Clock name	Clock Root	Description
ipg_clk_32k	ckil_sync_clk_root	Low-frequency reference clock (32 kHz)
ipg_clk_s	ipg_clk_root	Peripheral access clock

39.3 External Signals

There are several pins dedicated to the KPP. Keypads of any configuration up to eight rows and eight columns are supported through the software configuration of the peripheral pins. Any pins not used for the keypad are available as general purpose I/O. The registers are configured such that the pins can be treated as an I/O port up to 16 bits wide.

See the table below for the list of external signals.

Table 39-2. KPP External Signals

Signal	Description	Pad	Mode	Direction
KEY_COL0	Column input or output pin, from chip	KEY_COL0	ALT3	I/O
KEY_COL1	Column input or output pin, from chip	KEY_COL1	ALT3	I/O
KEY_COL2	Column input or output pin, from chip	KEY_COL2	ALT3	I/O
KEY_COL3	Column input or output pin, from chip	KEY_COL3	ALT3	I/O
KEY_COL4	Column input or output pin, from chip	KEY_COL4	ALT3	I/O
KEY_COL5	Column input or output pin, from chip	CSI0_DAT4	ALT3	I/O
		GPIO_0	ALT2	
		GPIO_19	ALT0	
		SD2_CLK	ALT2	
KEY_COL6	Column input or output pin, from chip	CSI0_DAT6	ALT3	I/O
		GPIO_9	ALT2	
		SD2_DAT3	ALT2	
KEY_COL7	Column input or output pin, from chip	CSI0_DAT8	ALT3	I/O
		GPIO_4	ALT2	
		SD2_DAT1	ALT4	
KEY_ROW0	Row input or output pin, from chip	KEY_ROW0	ALT3	I/O
KEY_ROW1	Row input or output pin, from chip	KEY_ROW1	ALT3	I/O
KEY_ROW2	Row input or output pin, from chip	KEY_ROW2	ALT3	I/O
KEY_ROW3	Row input or output pin, from chip	KEY_ROW3	ALT3	I/O
KEY_ROW4	Row input or output pin, from chip	KEY_ROW4	ALT3	I/O
KEY_ROW5	Row input or output pin, from chip	CSI0_DAT5	ALT3	I/O
		GPIO_1	ALT2	
		SD2_CMD	ALT2	
KEY_ROW6	Row input or output pin, from chip	CSI0_DAT7	ALT3	I/O
		GPIO_2	ALT2	
		SD2_DAT2	ALT4	
KEY_ROW7	Row input or output pin, from chip	CSI0_DAT9	ALT3	I/O

Table continues on the next page...

Table 39-2. KPP External Signals (continued)

Signal	Description	Pad	Mode	Direction
		GPIO_5	ALT2	
		SD2_DAT0	ALT4	

39.3.1 Input Pins

Any of the 16 pins associated with the KPP can be configured as inputs by writing a "0" to the appropriate bits in the KPP_KDDR. Additionally, the least significant 8 bits (ROW inputs) corresponding to KPP_KDDR[KRDD] have internal pull-ups, which are enabled when the pin is used as an input.

39.3.2 Output Pins

Any of the 16 pins associated with the KPP can be configured as outputs by writing the appropriate bits in the KPP_KDDR to a "1". Additionally, the 8 most significant bits (15-8) can be designated as open drain outputs by writing a "1" to the appropriate bits in the KPP_KPCR. The lower 8 bits (7-0) are always in "totem pole" style, driven when configured as outputs.

See the table below.

Table 39-3. Keypad Port Column Modes

KPP_KDDR (15:8)	KPP_KPCR (15:8)	Pin Function
0	x	Input
1	0	Totem-Pole Output
1	1	Open-Drain Output

NOTE

Totem pole capability should be provided for column pins. Totem pole configuration helps for a faster discharge of keypad capacitance when all columns need to be quickly brought to a "1" during the scan routine. With this configuration, delay between the scanning of two subsequent columns is reduced.

39.3.3 Generation of Transfer Error Signal on Peripheral Bus

If there is an access to an address which is not implemented, then the KPP asserts a transfer error signal on Peripheral Bus.

39.4 Functional Description

The Keypad Port (KPP) is designed to simplify the software task of scanning a keypad matrix. With appropriate software support and matrix organization, the KPP is capable of detecting, debouncing, and decoding one or more keys pressed simultaneously on the keypad.

Logic in the KPP is capable of detecting a key press even while the processor is in one of the low power standby modes provided that a low frequency reference clock (ipg_clk_32k) is on. The KPP may generate an Arm platform interrupt any time a key press or key release is detected. This interrupt is capable of forcing the processor out of a low power mode.

39.4.1 Keypad Matrix Construction

The KPP is designed to interface to a keypad matrix, which shorts the intersecting row and column lines together whenever a key is depressed. The interface is not optimized for any other switch configuration.

39.4.2 Keypad Port Configuration

The software must initialize the KPP for the size of the keypad matrix. Pins connected to the keypad columns should be configured as open-drain outputs. Pins connected to the keypad rows should be configured as inputs. On-chip, pull-up resistors should be implemented for active keypad rows.

In addition to enabled row inputs in the Keypad Control register, corresponding interrupt (depress or/and release) must also be enabled to generate an interrupt.

Discrete switches that are not part of the matrix may be connected to any unused row inputs. The second terminal of the discrete switch is connected to ground. The hardware detects closures of these switches without the need for software polling.

39.4.3 Keypad Matrix Scanning

Keypad scanning is performed by a software loop that walks a zero across each of the keypad columns, reading the value on the rows at each step. The process is repeated several times in succession, with the results of each pass optionally compared to those from the previous pass. When several (3 or 4) consecutive scans yield the same key closures, a valid key press has been detected. Software then can decode exactly which switch was depressed and pass the value up to the next higher software layer.

The basic debouncing period, which must be defined in the software routine, may be controlled with an internal timer. The basic period is the period between the scan of two consecutive columns, so the debouncing time between two consecutive scans of the whole matrix shall be the number of columns multiplied by the basic period.

39.4.4 Keypad Standby

There is no need for the Arm platform to continually scan the keypad. Between key presses, the keypad can be left in a state that requires no software intervention until the next key press is detected. To place the keypad in a standby state, software should write all column outputs low. Row inputs are left enabled. At this point, the Arm platform can attend to other tasks or revert to a low power standby mode. The KPP will interrupt the Arm platform if any key is pressed.

Upon receiving a keypad interrupt, the Arm platform should set all the column strobes high, and begin a normal keypad scanning routine to determine which key was pressed. It is important that open-drain drivers be used when scanning to prevent a possible DC path between power and ground through two or more switches.

39.4.5 Glitch Suppression on Keypad Inputs

A glitch suppression circuit qualifies the keypad inputs to prevent noise from inadvertently interrupting the Arm platform. The circuit is a 4-state synchronizer clocked from a low frequency reference clock (ipg_clk_32k) source. This clock must continue to run in any low power mode where the keypad is a wake-up source, as the Arm platform interrupt is generated from the synchronized input. An interrupt is not generated until all four synchronizer stages have latched a valid key assertion. This guarantees the filtering out of any noise less than three clock periods in duration of a low frequency reference clock. Noise filtering of the duration between three to four clock periods cannot be guaranteed. The interrupt output is latched in an S-R latch and remains asserted until cleared by the software. The Set input of the latch is rising-edge clocked. See the figure below.

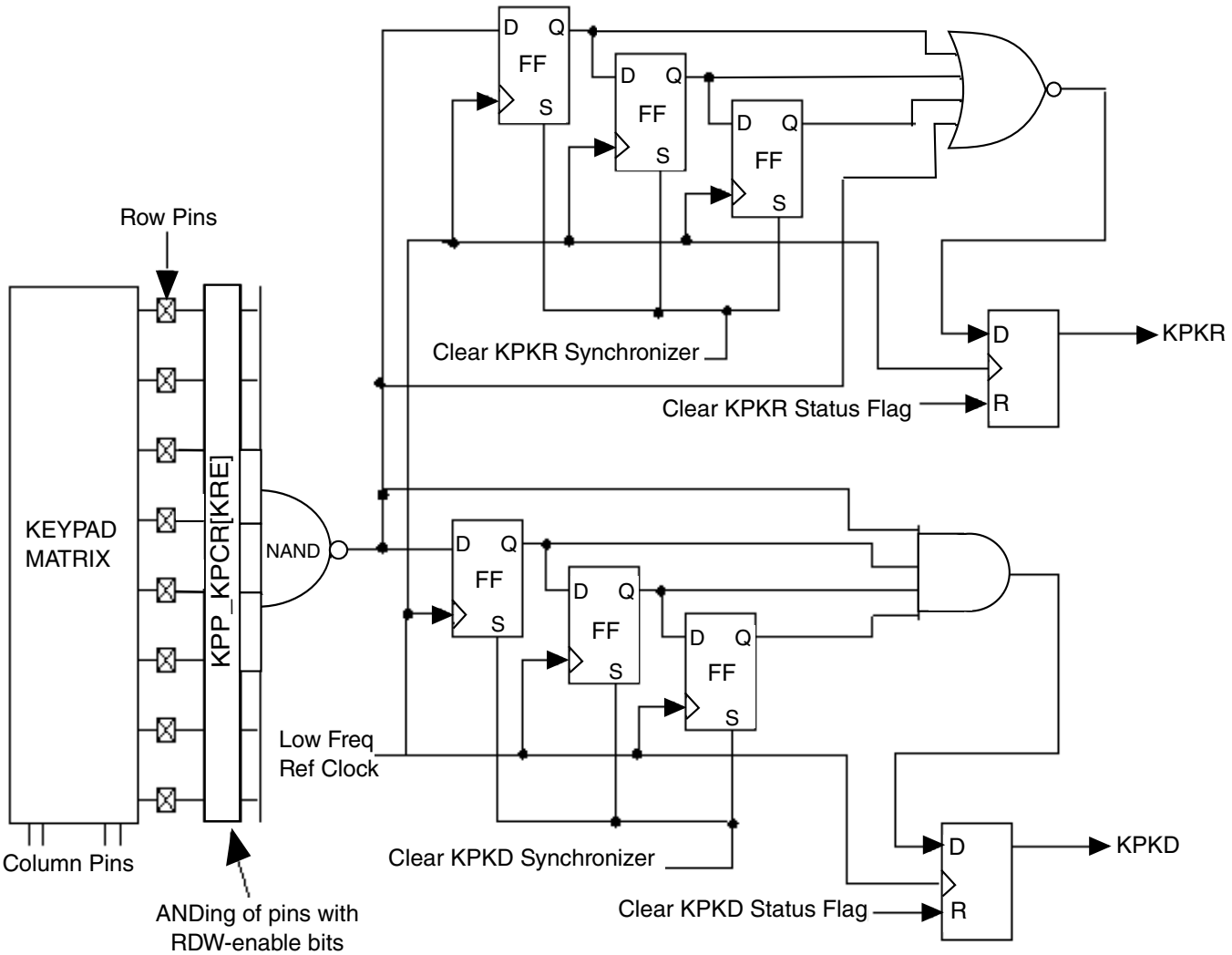


Figure 39-2. Keypad Synchronizer Functional Diagram

39.4.6 Multiple Key Closures

Using the key press and key release interrupts, the software can detect multiple keys or achieve n key rollover. The key scanning routine can be programmed accordingly (See Initialization/Application Information for more information).

The following figures illustrate the interface of a 2-contact keypad matrix with the KPP controller. With proper enabling of row lines and the performing scan-routine, multiple key presses can be detected. When keys present on the same row are pressed, corresponding row lines (multiple lines) become low when the column is driven low during a scan-routine. By reading the data-register, pressed keys can be detected.

Similarly, when keys present on same row line are pressed, the corresponding row line (only one line) becomes low when logic "0" is driven on the column line during a scan-routine.

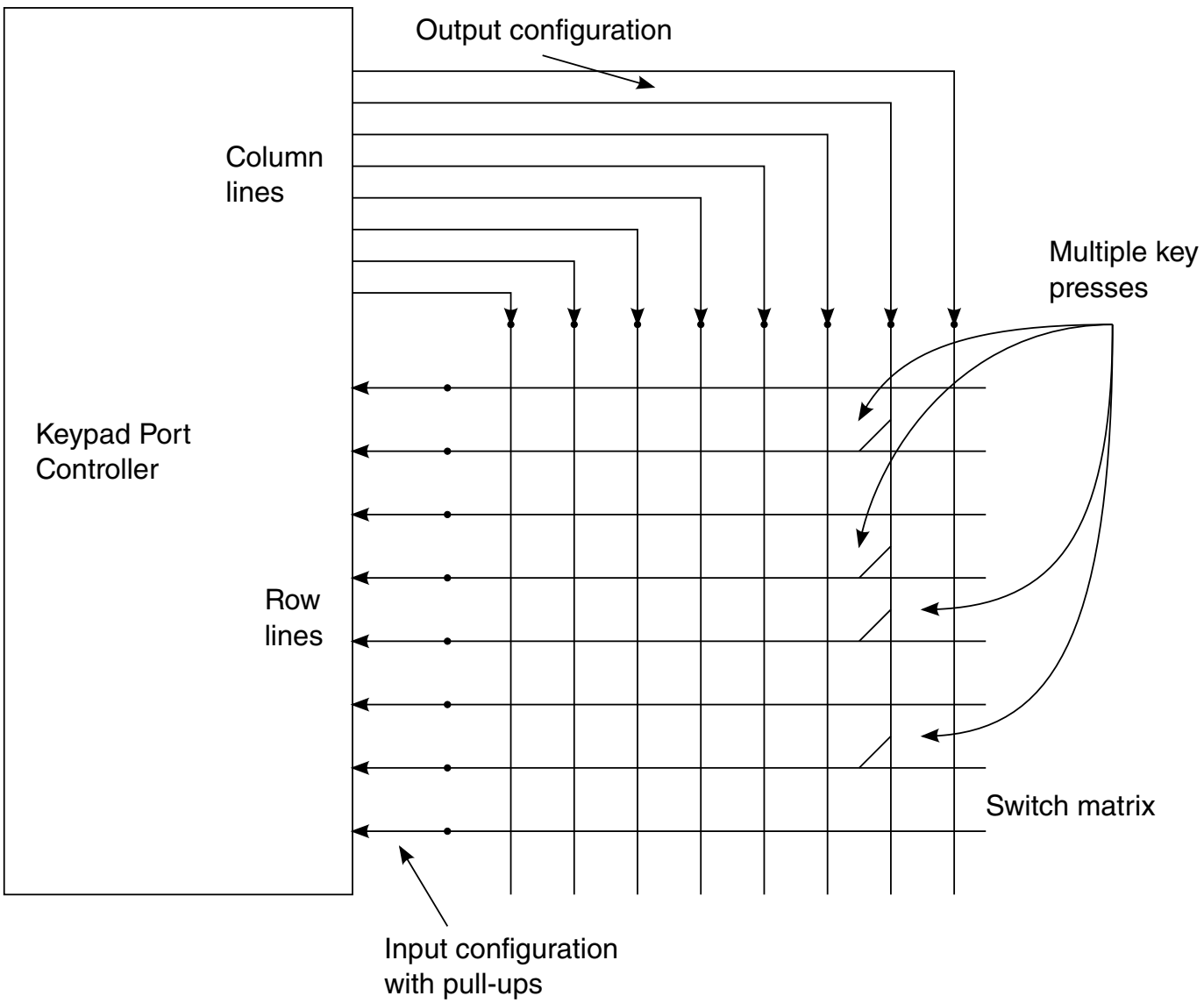


Figure 39-3. Multiple Key Presses on Same Column Line (Simplified View)

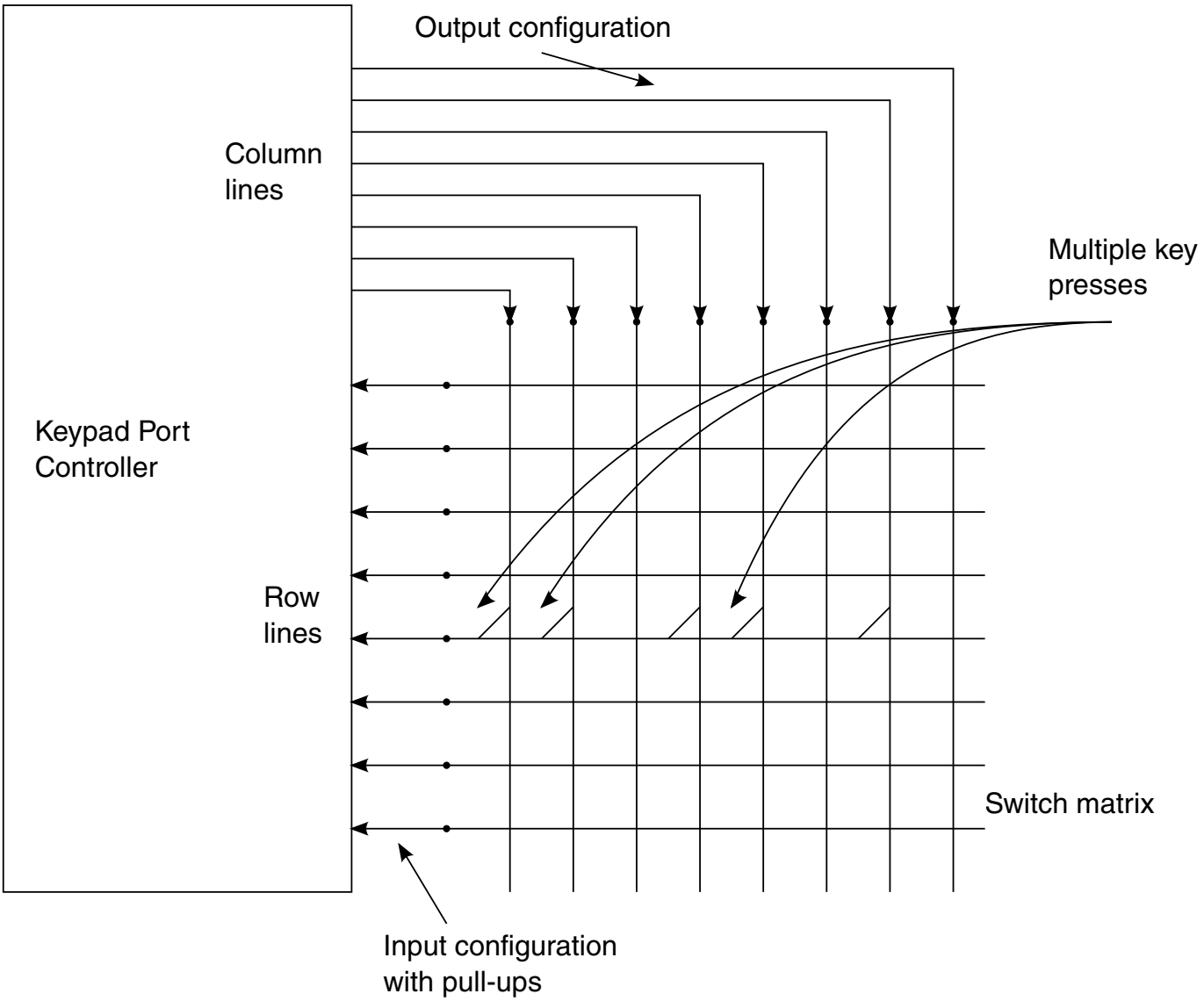


Figure 39-4. Multiple Key Presses on Same Row Line (Simplified View)

NOTE

An n key rollover is a technique with which the system can recognize the order in which keys are pressed.

39.4.6.1 Ghost Key Problem and Correction

The KPP detects if one or multiple keys are pressed or released. In the case where a simple keypad matrix with two-contact switches is used, there is a chance of "ghost" key detection when three or more keys are pressed. This is a limitation imposed by such a keypad matrix.

As seen in Figure 39-5, three keys pressed simultaneously can cause a short between the column currently "scanned" by the software and another column. Depending on the location of the third key pressed, a "ghost" key press may be detected.

However, this can be corrected by using a keypad matrix that provides "ghost" key protection. Such a matrix implements a one-way "diode" at all keypad points between rows and columns. This way, the multiple pressing of three keys will not cause a short at a fourth key (see Figure 39-6).

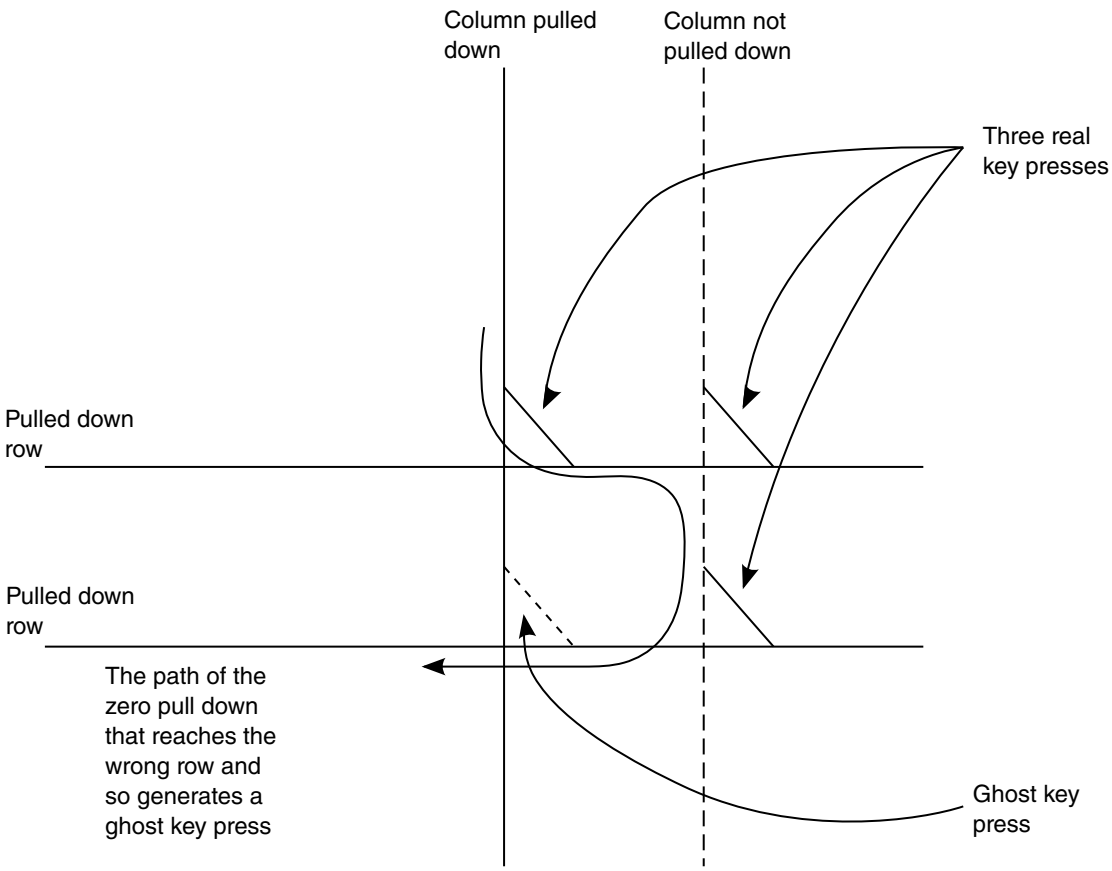


Figure 39-5. Decoding Wrong Three- Key-Presses

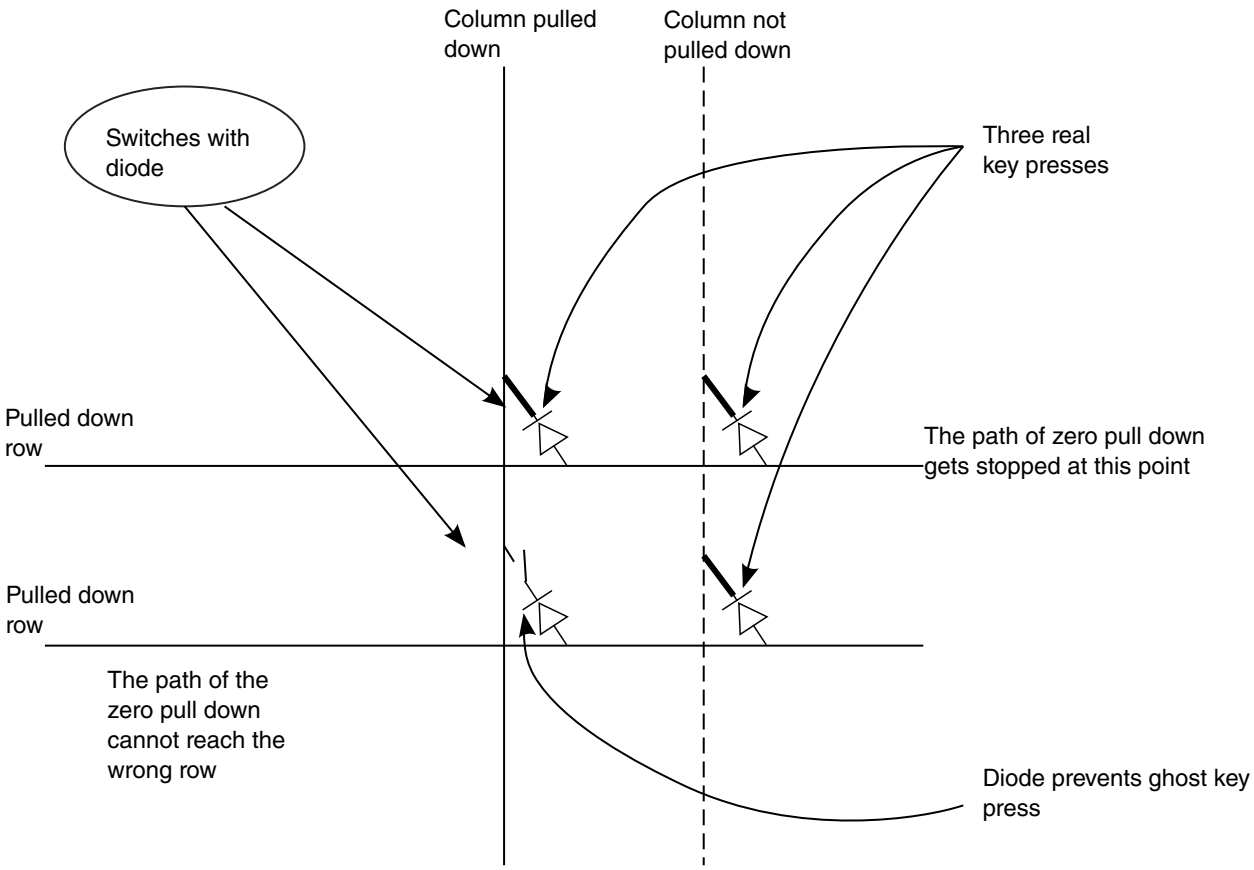


Figure 39-6. Matrix with "Ghost" Key Protections

39.4.7 3-Point Contact Keys Support

The KPP supports interfacing to a matrix consisting of 3-point contact keys. As shown in Figure 39-7, two points of such a key are connected to keypad lines, while a third point is connected to ground (low logic).

The keypad lines should be configured as input and a pull-up should to be present on these lines. When such a key is pressed, corresponding keypad lines go low and an interrupt is generated. There is no need to perform a scanning routine for identification of pressed key as it can be done by reading the keypad data-register. A limitation with such a matrix is that for every key at least one keypad row line should be used.

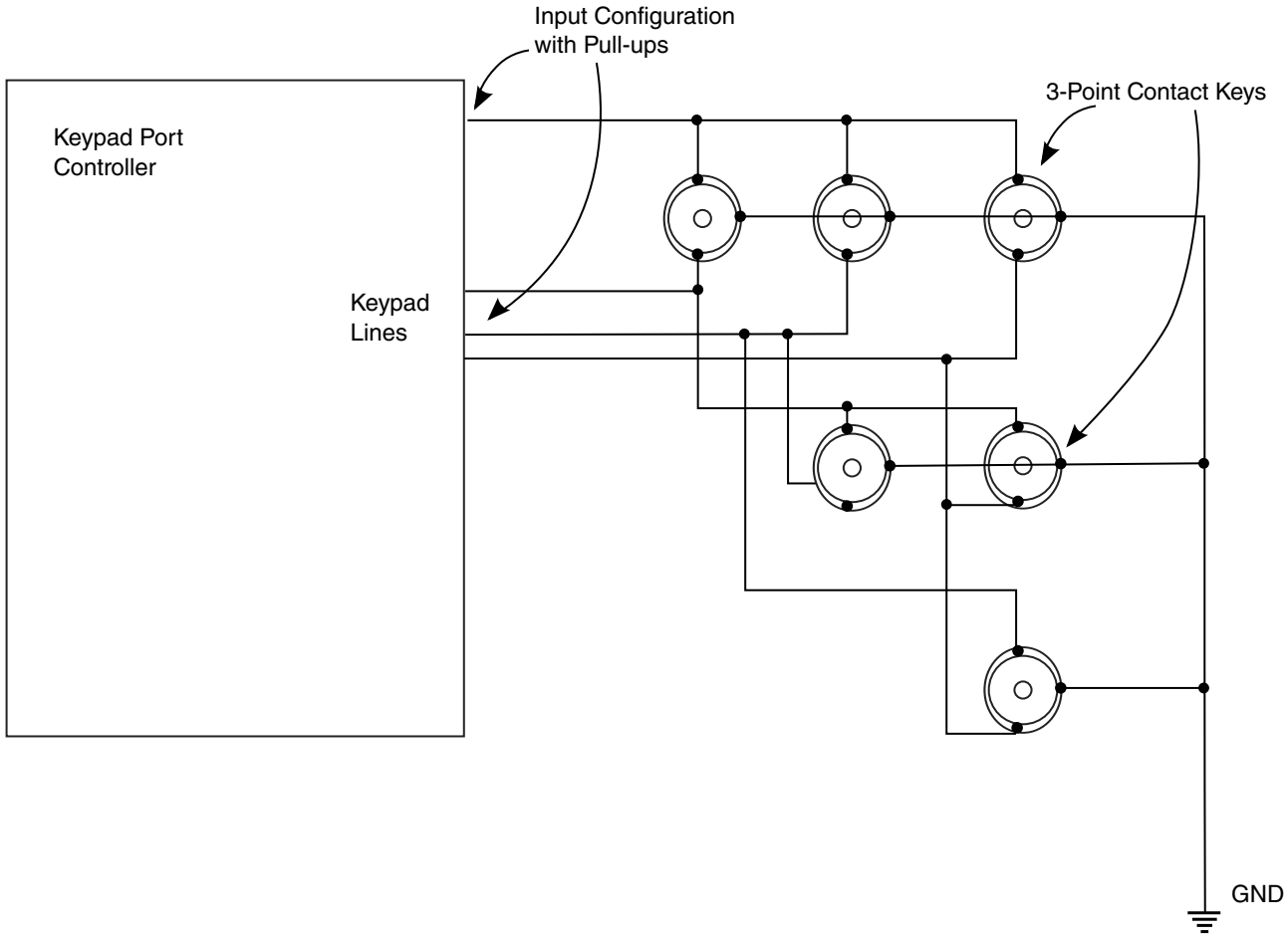


Figure 39-7. KPP Interface with 3-point Contact Key Matrix (Simplified View)

39.5 Initialization/Application Information

39.5.1 Typical Keypad Configuration and Scanning Sequence

Perform the following steps to configure the keypad:

1. Enable the number of rows in the keypad (KPP_KPCR[KRE]).
2. Write 0s to KPP_KPDR[KCD].
3. Configure the keypad columns as open-drain (KPP_KPCR[KCO]).
4. Configure columns as output (KPP_KDDR[KCDD]) and rows as input (KPP_KDDR[KRDD]).
5. Clear the KPKD Status Flag and Synchronizer chain.
6. Set the KDIE control bit, and clear the KRIE control bit (avoid false release events).

7. (The system is now in standby mode, and awaiting a key press.)

39.5.2 Key Press Interrupt Scanning Sequence

Perform the following steps to perform a keypad scanning routine:

1. Disable both (depress and release) keypad interrupts.
2. Write 1s to KPP_KPDR[KCD], setting column data to 1s.
3. Configure columns as totem pole outputs (for quick discharging of keypad capacitance).
4. Configure columns as open-drain.
5. Write a single column to 0, and other columns to 1.
6. Sample row inputs and save data. Multiple key presses can be detected on a single column.
7. Repeat Steps 2-6 for remaining columns.
8. Return all columns to 0 in preparation for standby mode.
9. Clear KPKD and KPKR status bit(s) by writing to a "1"; set the KPKR synchronizer chain by writing a "1" to the KPP_KRSS register; and clear the KPKD synchronizer chain by writing a "1" to the KDSC register.
10. Re-enable the appropriate keypad interrupt(s) so that the KDIE detects a key hold condition, or the KRIE detects a key-release event.

39.5.3 Additional Comments

The order of key press detection can be done in software only. Therefore, the software may need to run the scan routines at very short intervals of time per the application's demands. The reason that such functionality cannot be put in the KPP is that the block is limited by the number of external pins.

For the keys that require a very precise order (such as game keys), individual GPIO pins may be more useful.

39.6 KPP Memory Map/Register Definition

The KPP contains four registers.

KPP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20B_8000	Keypad Control Register (KPP_KPCR)	16	R/W	0000h	39.6.1/3611
20B_8002	Keypad Status Register (KPP_KPSR)	16	R/W	0400h	39.6.2/3612
20B_8004	Keypad Data Direction Register (KPP_KDDR)	16	R/W	0000h	39.6.3/3613
20B_8006	Keypad Data Register (KPP_KPDR)	16	R/W	0000h	39.6.4/3614

39.6.1 Keypad Control Register (KPP_KPCR)

The Keypad Control Register determines which of the eight possible column strobes are to be open drain when configured as outputs, and which of the eight row sense lines are considered in generating an interrupt to the core.

It is up to the programmer to ensure that pins being used for functions other than the keypad are properly disabled. The KPP_KPCR register is byte- or half-word-addressable.

Address: 20B_8000h base + 0h offset = 20B_8000h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	KCO								KRE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

KPP_KPCR field descriptions

Field	Description
15–8 KCO	Keypad Column Strobe Open-Drain Enable. Setting a column open-drain enable bit (KCO7-KCO0) disables the pull-up driver on that pin. Clearing the bit allows the pin to drive to the high state. This bit has no effect when the pin is configured as an input. **NOTE:** Configuration of external port control logic (for example, IOMUX) should be done properly so that the KPP controls an open-drain enable of the pin. 0 **TOTEM_POLE** — Column strobe output is totem pole drive. 1 **OPEN_DRAIN** — Column strobe output is open drain.
KRE	Keypad Row Enable. Setting a row enable control bit in this register enables the corresponding row line to participate in interrupt generation. Likewise, clearing a bit disables that row from being used to generate an interrupt. This register is cleared by a reset, disabling all rows. The row-enable logic is independent of the programmed direction of the pin. Writing a "0" to the data register of the pins configured as outputs will cause a keypad interrupt to be generated if the row enable associated with that bit is set. 0 Row is not included in the keypad key press detect. 1 Row is included in the keypad key press detect.

39.6.2 Keypad Status Register (KPP_KPSR)

The Keypad Status Register reflects the state of the key press detect circuit. The KPP_KPSR register is byte- or half-word-addressable.

Address: 20B_8000h base + 2h offset = 20B_8002h

Bit	15	14	13	12	11	10	9	8
Read	0						KRIE	KDIE
Write								
Reset	0	0	0	0	0	1	0	0

Bit	7	6	5	4	3	2	1	0
Read	0				0	0	KPKR	KPKD
Write					KRSS	KDSC	w1c	w1c
Reset	0	0	0	0	0	0	0	0

KPP_KPSR field descriptions

Field	Description
15–10 Reserved	This read-only field is reserved and always has the value 0.
9 KRIE	Keypad Release Interrupt Enable. The software should ensure that the interrupt for a Key Release event is masked until it has entered the key pressed state, and vice versa, unless this activity is desired (as might be the case when a repeated interrupt is to be generated). The synchronizer chains are capable of being initialized to detect repeated key presses or releases. If they are not initialized when the corresponding event flag is cleared, false interrupts may be generated for depress (or release) events shorter than the length of the corresponding chain. 0 No interrupt request is generated when KPKR is set. 1 An interrupt request is generated when KPKR is set.
8 KDIE	Keypad Key Depress Interrupt Enable. Software should ensure that the interrupt for a Key Release event is masked until it has entered the key pressed state, and vice-versa, unless this activity is desired (as might be the case when a repeated interrupt is to be generated). The synchronizer chains are capable of being initialized to detect repeated key presses or releases. If they are not initialized when the corresponding event flag is cleared, false interrupts may be generated for depress (or release) events shorter than the length of the corresponding chain. 0 No interrupt request is generated when KPKD is set. 1 An interrupt request is generated when KPKD is set.
7–4 Reserved	This read-only field is reserved and always has the value 0.
3 KRSS	Key Release Synchronizer Set. Self-clear bit. The Key release synchronizer is set by writing a logic one into this bit. Reads return a value of "0". 0 No effect 1 Set bits which sets keypad release synchronizer chain

Table continues on the next page...

KPP_KPSR field descriptions (continued)

Field	Description
2 KDSC	Key Depress Synchronizer Clear. Self-clear bit. The Key depress synchronizer is cleared by writing a logic "1" into this bit. Reads return a value of "0". 0 No effect 1 Set bits that clear the keypad depress synchronizer chain
1 KPKR	Keypad Key Release. The keypad key release (KPKR) status bit is set when all enabled rows are detected high after synchronization (the KPKR status bit will be set when cleared by a reset). The KPKR bit may be used to generate a maskable key release interrupt. The key release synchronizer may be set high by software after scanning the keypad to ensure a known state. Due to the logic function of the release and depress synchronizer chains, it is possible to see the re-assertion of a status flag (KPKD or KPKR) if it is cleared by software prior to the system exiting the state it represents. Reset value of register is "0" as long as reset is asserted. However when reset is de-asserted, the value of the register depends upon the external row pins and can become "1". 0 No key release detected 1 All keys have been released
0 KPKD	Keypad Key Depress. The keypad key depress (KPKD) status bit is set when one or more enabled rows are detected low after synchronization. The KPKD status bit remains set until cleared by the software. The KPKD bit may be used to generate a maskable key depress interrupt. If desired, the software may clear the key press synchronizer chain to allow a repeated interrupt to be generated while a key remains pressed. In this case, a new interrupt will be generated after the synchronizer delay (4 cycles of the low frequency reference clock (ipg_clk_32k) elapses if a key remains pressed. This functionality can be used to detect a long key press. This allows detection of additional key presses of the same key or other keys. Due to the logic function of the release and depress synchronizer chains, it is possible to see the re-assertion of a status flag (KPKD or KPKR) if it is cleared by the software prior to the system exiting the state it represents. 0 No key presses detected 1 A key has been depressed

39.6.3 Keypad Data Direction Register (KPP_KDDR)

The bits in the KPP_KDDR control the direction of the keypad port pins. The upper eight bits in the register affect the pins designated as column strobes, while the lower eight bits affect the row sense pins. Setting any bit in this register configures the corresponding pin as an output. Clearing any bit in this register configures the corresponding port pin as an input. For the Keypad Row DDR, an internal pull-up is enabled if the corresponding bit is clear. This register is cleared by a reset, configuring all pins as inputs. The KPP_KDDR register is byte- or half-word addressable.

NOTE

When a pin is used as row pin for keypad purposes, all corresponding pull-ups should be enabled at the upper level (for example, IOMUX) when the bit in KRDD is cleared.

Address: 20B_8000h base + 4h offset = 20B_8004h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	KCDD								KRDD							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

KPP_KDDR field descriptions

Field	Description
15–8 KCDD	Keypad Column Data Direction Register. Setting a bit configures the corresponding COL*n* pin as an output (where *n* = 7 through 0). 0 **INPUT** — COLn pin is configured as an input. 1 **OUTPUT** — COLn pin is configured as an output.
KRDD	Keypad Row Data Direction. Setting a bit configures the corresponding ROW*n* pin as an output (where *n* = 7 through 0). 0 **INPUT** — ROWn pin configured as an input. 1 **OUTPUT** — ROWn pin configured as an output.

39.6.4 Keypad Data Register (KPP_KPDR)

This 16-bit register is used to access the column and row data. Data written to this register is stored in an internal latch, and for each pin configured as an output, the stored data is driven onto the pin. A read of this register returns the value on the pin for those bits configured as inputs. Otherwise, the value read is the value stored in the register.

The KPP_KPDR register is byte- or half-word addressable. This register is not initialized by a reset. Valid data should be written to this register before any bits are configured as outputs.

Address: 20B_8000h base + 6h offset = 20B_8006h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	KCD								KRD							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

KPP_KPDR field descriptions

Field	Description
15–8 KCD	Keypad Column Data. A read of these bits returns the value on the pin for those bits configured as inputs. Otherwise, the value read is the value stored in the register. 0 Read/Write "0" from/to column ports 1 Read/Write "1" from/to column ports
KRD	Keypad Row Data. A read of these bits returns the value on the pin for those bits configured as inputs. Otherwise, the value read is the value stored in the register. 0 Read/Write "0" from/to row ports 1 Read/Write "1" from/to row ports

KPP_KPDR field descriptions (continued)

Field	Description

Chapter 40
LVDS Display Bridge (LDB)

40.1 Overview

The LVDS Display Bridge (LDB) connects the IPU (Image Processing Unit) to an External LVDS Display Interface.

The purpose of the LDB is to support flow of synchronous RGB data from the IPU to external display devices through the LVDS interface. This support covers all aspects of these activities:

- Connectivity to relevant devices - Displays with LVDS receivers.
- Arranging the data as required by the external display receiver and by LVDS display standards.
- Synchronization and control capabilities.

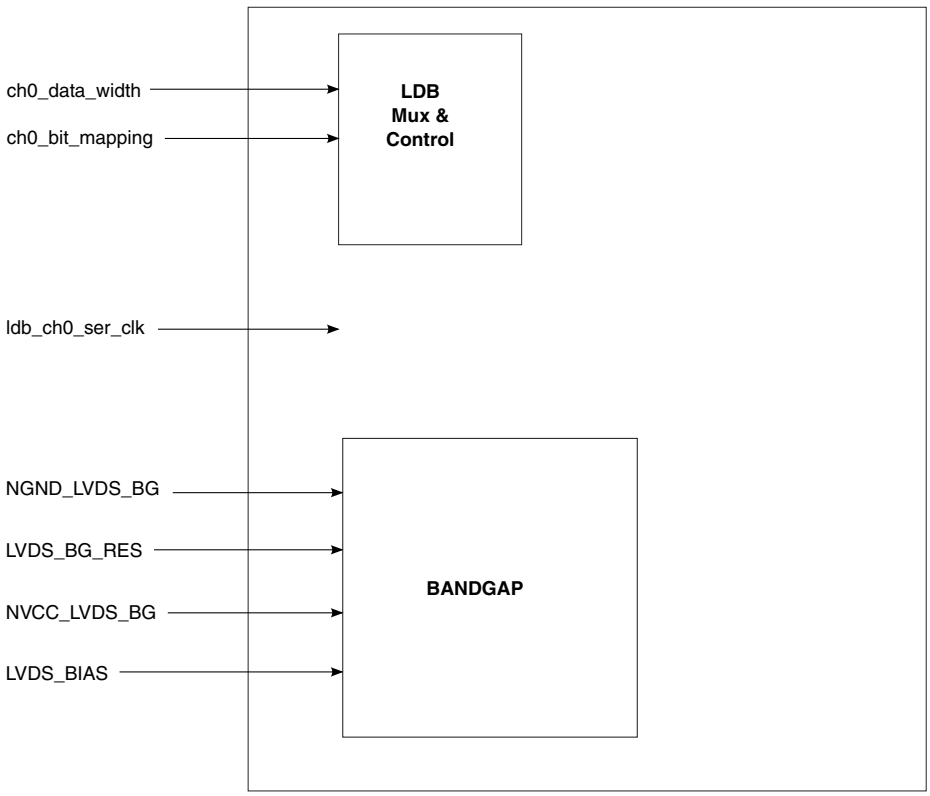


Figure 40-1. LDB Block Diagram

Table 40-1. LDB - Block Description

Block	Description
LDB Mux and Control	Gets control signals from SoC and determines parameters of LDB
Channel Serializers	LDB has 2 channel serializers. The serializer does parallel to serial conversion 7:1 to 3 or 4 data lines
Bandgap	Provides reference current to the LVDS I/O pads.

Table 40-2. LDB IP Parametric Table

Name	IPU
Function	Connectivity to displays with LVDS interface
External I/O Pins Those are LVDS I/O pads	LVDS Display port: 2 channels, consists of: • 1 clock pair • 4 data pairs Each pair contains - LVDS special differential pad (PadP, PadM). total of 20 I/O pads.
SoC Buses	None. Only configuration signals.
Interrupts	None

Table continues on the next page...

Table 40-2. LDB IP Parametric Table (continued)

Name	IPU
DMA Requests	None
Number of instantiations	1
Clock sources and range	IPU_DI0_CLK, IPU_DI1_CLK- Display interface clock: 20-170 MHz DI0_SERIAL_CLK, DI1_SERIAL_CLK - Serializer clock: 140-595 MHz

40.1.1 Relevant Standards

Below are the relevant standards to LDB:

1. ANSI EIA-644-A. Electrical Characteristics of Low Voltage Differential Signaling (LVDS) Interface Circuits.
2. SPWG Notebook Panel Specification (V3.8 from 03/2007) .
3. PSWG standards (Panel Standardization Working Group) - set of standards for panels using LVDS. All are available from http://www.vesa.org.
4. DISM Standard JEIDA-59-1999

40.2 External Signals

The following table describes the external signals of LDB:

Table 40-3. LDB External Signals

Signal	Description	Pad	Mode	Direction
LVDS0_CLK_N	LVDS0 Negative Clock Signal	LVDS0_CLK_N	No Muxing	I/O
LVDS0_CLK_P	LVDS0 Positive Clock Signal	LVDS0_CLK_P	No Muxing	I/O
LVDS0_DATA[3:0]_N	LVDS0 Negative Data Signals	LVDS0_TX[3:0]_N	No Muxing	I/O
LVDS0_DATA[3:0]_P	LVDS0 Positive Data Signals	LVDS0_TX[3:0]_P	No Muxing	I/O
LVDS1_CLK_N	LVDS1 Negative Clock Signal	LVDS1_CLK_N	No Muxing	I/O
LVDS1_CLK_P	LVDS1 Positive Clock Signal	LVDS1_CLK_P	No Muxing	I/O
LVDS1_DATA[3:0]_N	LVDS1 Negative Data Signals	LVDS1_TX[3:0]_N	No Muxing	I/O
LVDS1_DATA[3:0]_P	LVDS1 Positive Data Signals	LVDS1_TX[3:0]_P	No Muxing	I/O

40.3 Clocks

A table with the LDB Clock Sources can be found here.

Table 40-4. LDB Clock Sources

Name	Symbol	Source	Rate	Comments
IPU DI0 interface pixel clock	IPU_DI0_CLK	Clock control Module	Up to 170 MHz	See note below [1]
IPU DI1 interface pixel clock	IPU_DI1_CLK	Clock control Module	Up to 170 MHz	This input also goes to IPU DI1 as input. See note below
CH0 interface serializer clock	DI0_SERIAL_CLK	Clock control Module	Up to 595 MHz	This is x7 the rate of the DI0 interface pixel clock. See note below.
CH1 interface serializer clock	DI1_SERIAL_CLK	Clock control Module	Up to 595 MHz	This is x7 the rate of the DI1 interface pixel clock. See note below

1. In case of single-channel or separate-channels use-case, the IPU DI_CLK is identical to the LVDS DI_CLK. In case of split-channel use-case, the IPU DI_CLK has x2 higher frequency than that of the LVDS DI_CLK. Still both need to be synchronized

40.4 Input and Output Ports

40.4.1 Input Parallel Display Ports

One or two parallel RGB input ports are supported (configurable). Only synchronous access mode is supported.Each RGB data interface contains the following:

- RGB Data of 18 or 24 bits

NOTE

In 18 bits mode, if the RGB bus width is 24 bits, the RGB data are {6'b00000, R[5:0], G[5:0], b[5:0]}.

- Pixel clock
- Control signals: HSYNC, VSYNC, DE, and 1 additional optional general purpose control.

Total of up to 28 bits per data interface are transferred per pixel clock cycle.

Rates supported:

- Overall: LDB supports rates needed by WUXGA 16:10 aspect ratio (1920 x 1200 @ 60 frames per second, data rate supported up to 170 MHz)

- For single input data interface case: Up to 170 MHz pixel clock (WUXGA 1920x1200)
- For dual input data interface case: Up to 85 MHz per interface. (WXGA 1366x768 @ 60 frames per second, 35% blanking).

40.4.2 Output LVDS Ports

There is 2 LVDS channels. The output of each is used to communicate RGB data and controls to external LCD displays.

The LVDS ports may be used as follows:

- Single channel output
- Dual channel output (one input source, two channels outputs for two displays)
- Split channel output (one input source, splitted to 2 channels on output)
- Separate 2 channel output (2 input sources from IPU).

The output LVDS port complies to the EIA-644-A standard.

40.5 Processing

LDB data processing stages are as follows:

- Receive input data from 1 or 2 (configurable) parallel input interfaces. 18/24 RGB data + up to 4 controls and map them to LVDS channels.
  - If needed (dual-channel) split the input bus to two half-rate busses.
- Re-arrange the input data according to channel configuration, and muxing scheme.
- Serialize the 22/28 bit input bus (per channel) on 3-4 output serial data lines (7:1)

40.5.1 Mapping of Input Data Busses

Mapping of Parallel input interfaces (DI0, DI1) to output LVDS channels (Channel 0, Channel 1). See Table 40-5.

Table 40-5. Channel Mapping

Use Case	LVDS Channel 0	LVDS Channel 1
Single Channel DI0	DI0	Disabled
Single Channel DI1 on Channel 1	Disabled	DI1
Separate Channels	DI0	DI1

Table continues on the next page...

Table 40-5. Channel Mapping (continued)

Use Case	LVDS Channel 0	LVDS Channel 1
Dual Channels DI0	DI0	DI0
Dual Channels DI1	DI1	DI1
Split Channel DI0	DI0 (first pixel)	DI0 (second pixel)
Split Channel DI1	DI1 (first pixel)	DI1 (second pixel)

40.5.2 Bit Mapping

LDB supports two mapping standards:

- SPWG mapping
- JEIDA mapping

Table 40-6. SPWG/PSWG/VESA 18/24 bpp Data Mapping

Serializer input	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
LVDSn_DATA0	G0	R5	R4	R3	R2	R1	R0
LVDSn_DATA1	B1	B0	G5	G4	G3	G2	G1
LVDSn_DATA2	DE	VS	HS	B5	B4	B3	B2
LVDSn_DATA3 (for 24 bpp only)	CTL	B7	B6	G7	G6	R7	R6

Table 40-7. JEIDA 24bpp Data Mapping

Serializer input	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
LVDSn_DATA0	G2	R7	R6	R5	R4	R3	R2
LVDSn_DATA1	B3	B2	G7	G6	G5	G4	G3
LVDSn_DATA2	DE	VS	HS	B7	B6	B5	B4
LVDSn_DATA3	CTL	B1	B0	G1	G0	R1	R0

NOTE

Several options of control usage can be available. some display devices use only DE, some others use all 3 controls, some use only HS, VS. "CTL" is an optional general purpose control which is usually unused by display.

40.6 LDB Memory Map/Register Definition

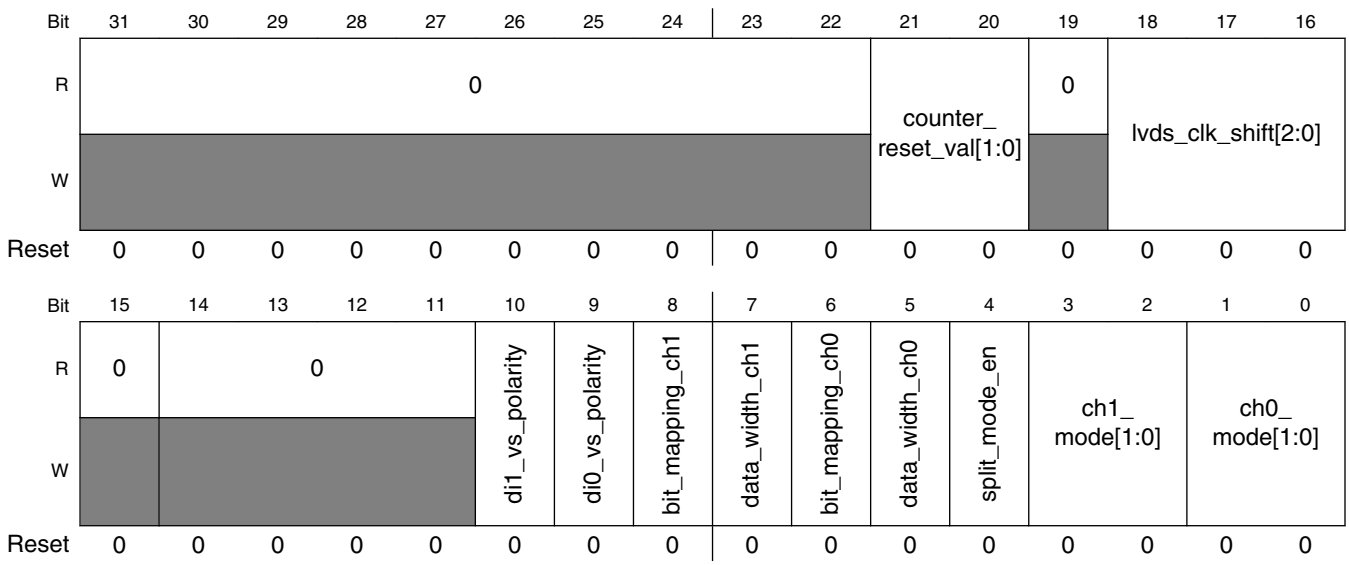
LDB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_0008	LDB Control Register (LDB_CTRL)	32	R/W	0000_0000h	40.6.1/3623

40.6.1 LDB Control Register (LDB_CTRL)

The register is implemented in the IOMUX Controller block (IOMUXC), as the register IOMUXC_GPR2.

Address: 20E_0008h base + 0h offset = 20E_0008h



LDB_CTRL field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value 0.
21–20 counter_reset_ val[1:0]	Reset value for the LDB counter which determines when the shift registers are loaded with data. **NOTE:** Used for debug purposes only. In normal functional operation must be '00' 00 Reset value is 5 01 Reset value is 3 10 Reset value is 4 11 Reset value is 6
19 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

LDB_CTRL field descriptions (continued)

Field	Description
18–16 lvds_clk_shift[2:0]	Shifts the LVDS output clock in relation to the data. **NOTE:** Used for debug purposes only. In normal functional operation must be '000' 000 Output clock is '1100011' (normal operation) 001 Output clock is '1110001' 010 Output clock is '1111000' 011 Output clock is '1000111' 100 Output clock is '0001111' 101 Output clock is '0011111' 110 Output clock is '0111100' 111 Output clock is '1100011'
15 Reserved	This read-only field is reserved and always has the value 0.
14–11 Reserved	This read-only field is reserved and always has the value 0.
10 di1_vs_polarity	Vsync polarity for IPU's DI1 interface. 0 ipu_di1_vsync is active high. 1 ipu_di1_vsync is active low.
9 di0_vs_polarity	Vsync polarity for IPU's DI0 interface. 0 ipu_di0_vsync is active high. 1 ipu_di0_vsync is active low.
8 bit_mapping_ch1	Data mapping for LVDS channel 1. 0 Use SPWG standard. 1 Use JEIDA standard.
7 data_width_ch1	Data width for LVDS channel 1. **NOTE:** This bit must be set when using JEIDA standard (bit_mapping_ch1 is set) 0 Data width is 18 bits wide (lvds1_tx3 is not used) 1 Data width is 24 bits wide.
6 bit_mapping_ch0	Data mapping for LVDS channel 0. 0 Use SPWG standard. 1 Use JEIDA standard.
5 data_width_ch0	Data width for LVDS channel 0. **NOTE:** This bit must be set when using JEIDA standard (bit_mapping_ch0 is set) 0 Data width is 18 bits wide (lvds0_tx3 is not used) 1 Data width is 24 bits wide.
4 split_mode_en	Enable split mode. **NOTE:** In this mode both channels should be enabled and working with the same DI (ch0_mode and ch1_mode should both be either '01' or '11') 0 Split mode is disabled. 1 Split mode is enabled.

Table continues on the next page...

LDB_CTRL field descriptions (continued)

Field	Description
3–2 ch1_mode[1:0]	LVDS channel 1 operation mode 00 Channel disabled. 01 Channel enabled, routed to DI0 10 Channel disabled. 11 Channel enabled, routed to DI1.
ch0_mode[1:0]	LVDS channel 0 operation mode 00 Channel disabled. 01 Channel enabled, routed to DI0 10 Channel disabled. 11 Channel enabled, routed to DI1.

Chapter 41
MIPI - Camera Serial Interface Host Controller (MIPI_CSI)

41.1 Overview

The CSI-2 Host Controller is a digital core that implements all protocol functions defined in the MIPI CSI-2 Specification, providing an interface between the System and the MIPI D-PHY, allowing the communication with a MIPI CSI-2 compliant Camera Sensor.

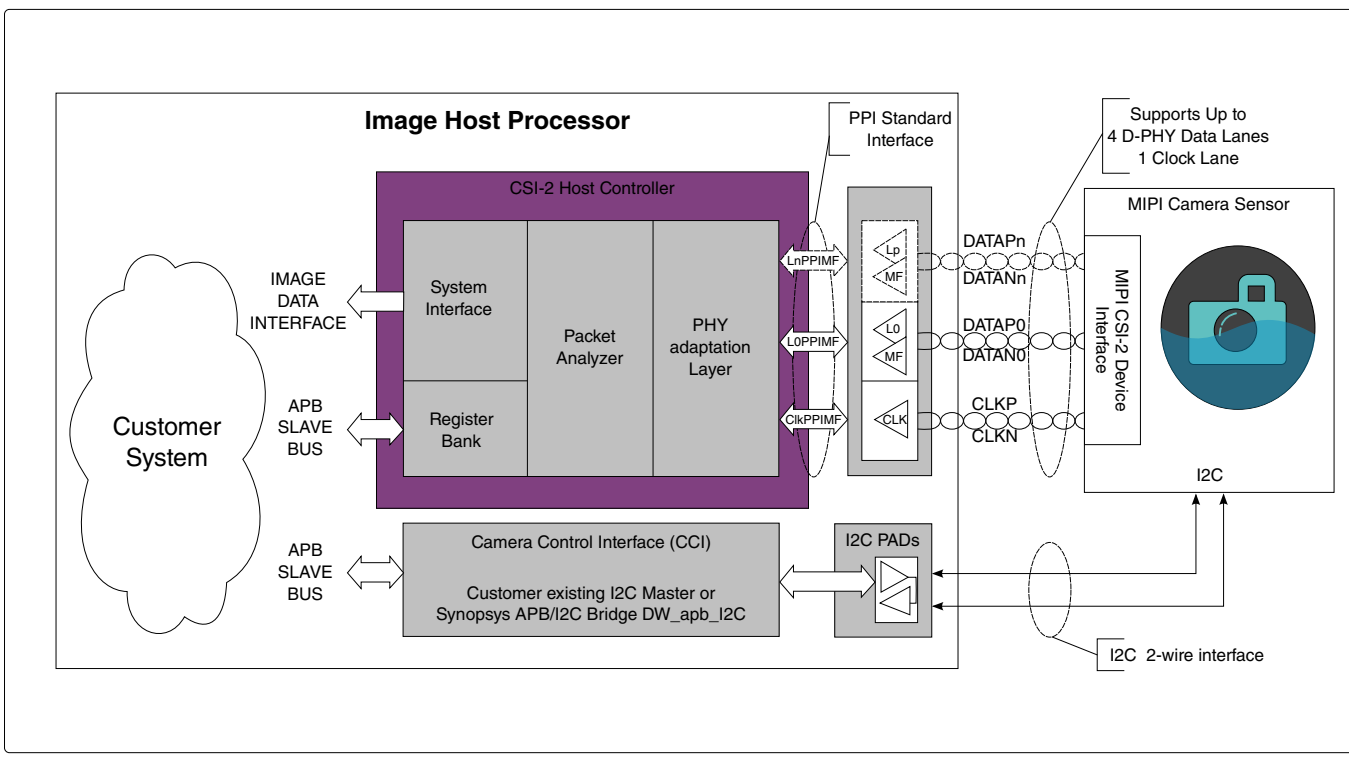


Figure 41-1. Block Diagram presenting the CSI-2 Host Controller function.

41.2 Features

The MIPI CSI-2 Host Controller supports the following features:

- Compliant with MIPI Alliance Standard for Camera Serial Interface 2 (CSI-2), Version 1.00 - 29 November 2005
- Interface with MIPI D-PHY following PHY Protocol Interface (PPI), as defined in MIPI Alliance Specification for D-PHY, Version 1.00.00 - 14 May 2009
- Supports up to 2 D-PHY Rx Data Lanes
- Dynamically configurable multi-lane merging
- Long and Short packet decoding
- Timing accurate signaling of Frame and Line synchronization packets
- Support for several frame formats such as:
  - General Frame or Digital Interlaced Video with or without accurate sync timing
  - Data type (Packet or Frame level) and Virtual Channel interleaving
- 32-bit Image Data Interface delivering data formatted as recommended in CSI-2 Specification;
- Supports all primary and secondary data formats:
  - RGB, YUV and RAW color space definitions
  - From 24-bit down to 6-bit per pixel
  - Generic or user-defined byte-based data types
- Error detection and correction:
  - PHY level
  - Packet level
  - Line level
  - Frame level

41.3 Architecture

The following figure presents the overall architecture of the CSI-2 Host Controller. The main blocks are the following:

- PHY Adaptation Layer, which is responsible for managing the D-PHY interface, including PHY error handling;
- Packet Analyzer, where data lane merging is implemented if required, together with header decoding, error detection and correction, frame size error detection and CRC error detection;
- Image Data Interface. This block separates CSI-2 packet header information and reorders data according to memory storage format, and it also generates timing

accurate video synchronization signals. Several error detections are also performed at frame-level and line-level;

- Register Bank, accessible through a standard AMBA-APB slave interface, providing access to the CSI-2 Host Controller registers for configuration and control. There is also a fully programmable interrupt generator to inform the system upon certain events;

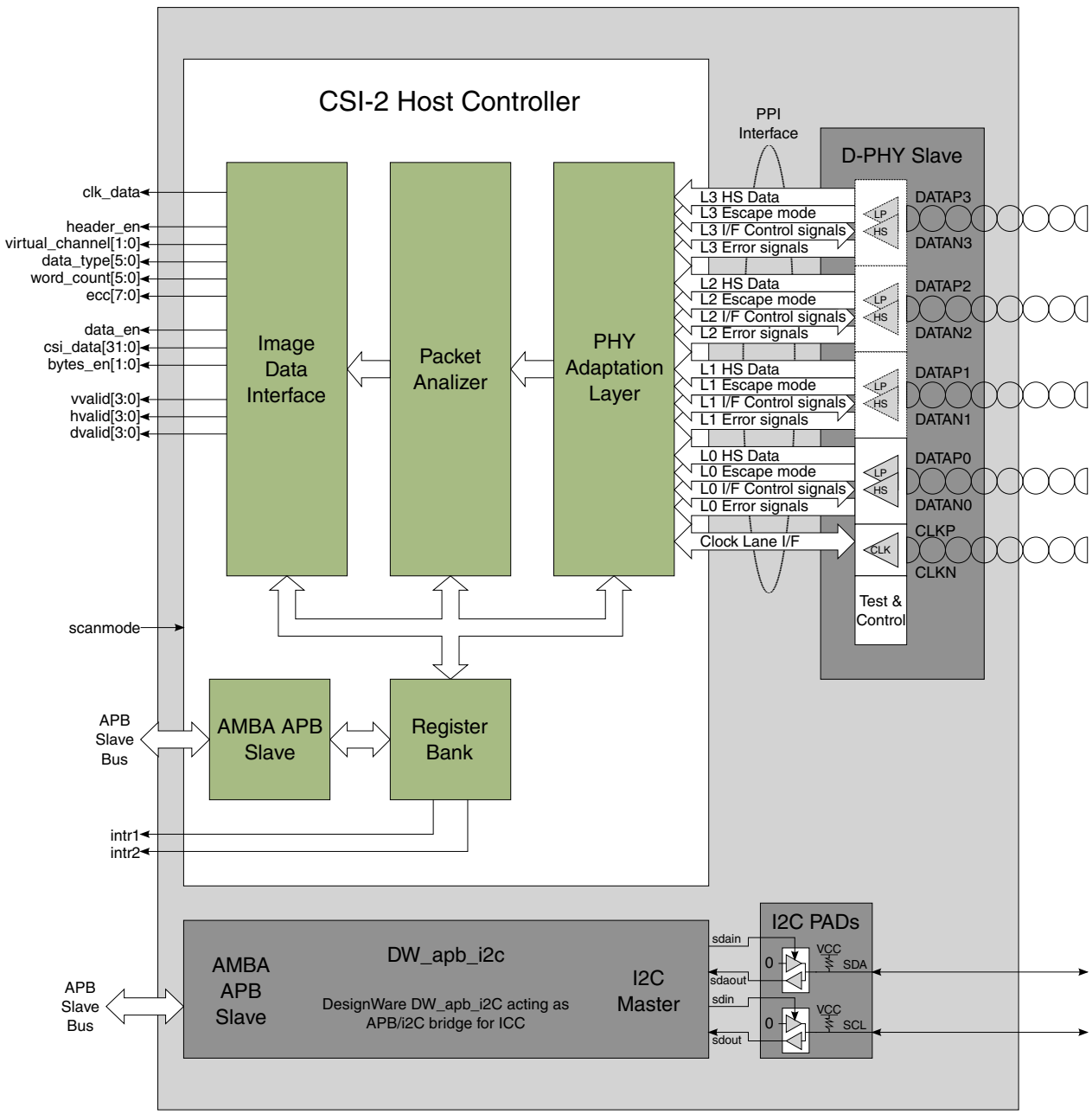


Figure 41-2. Architecture of the CSI-2 Host Controller.

41.3.1 Startup Sequence

The following information is provided as a guideline to allow a safe startup of the system operation.

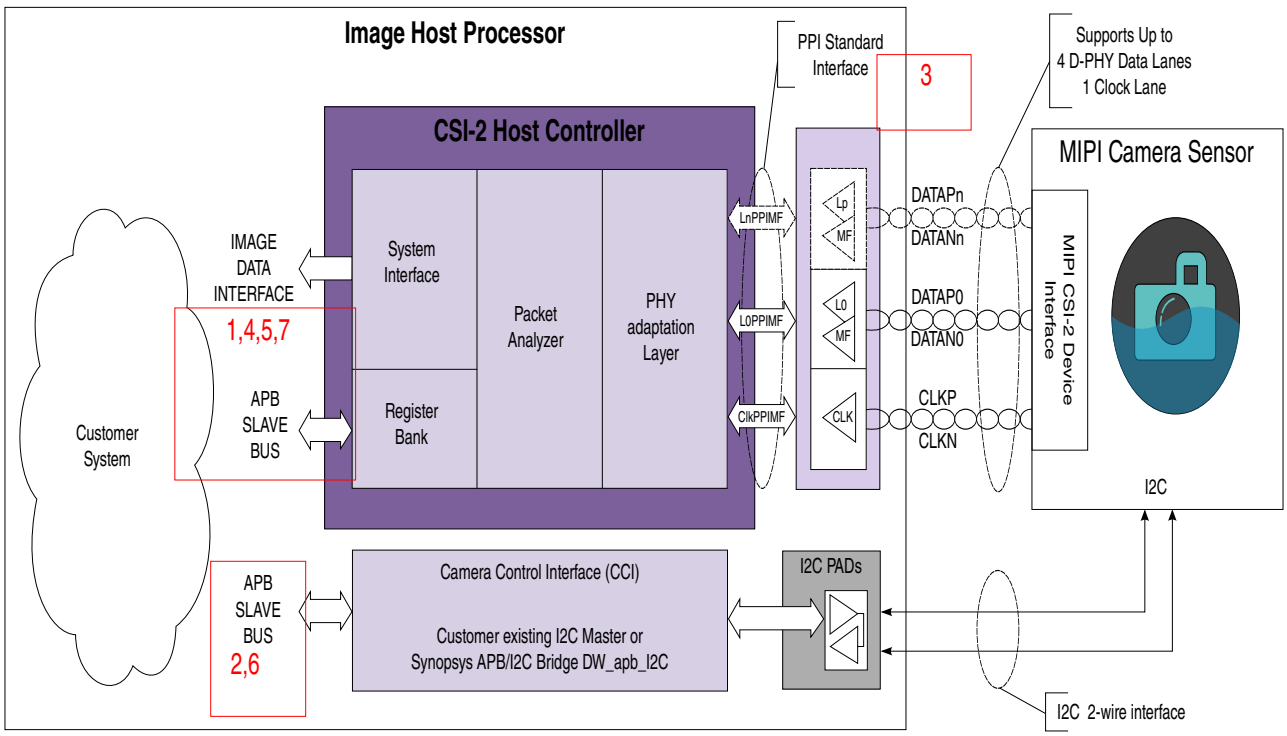


Figure 41-3. Block diagram presenting the recommended Startup Sequence.

1. Deassert CSI2 presetn signal (global reset);
2. **Configure MIPI Camera Sensor to have all Tx lanes in PL-11 state (STOPSTATE) if required** - According to D-PHY Specification the D-PHY master should be initialized at LP-11 state (STOPSTATE), nevertheless a CCI command may be required to switch-on the MIPI interface;
3. **D-PHY initialization** - Access the D-PHY programming interface to initialize and program the D-PHY according to the selected operating mode of the D-PHY. This is D-PHY dependent, and this programming should be carried out according to the D-PHY databook;
4. **CSI2 Controller programming** - Program the CSI2 Host Controller registers according to the operating mode required:
   - **Number of Lanes (register N_LANES);**
   - **Deassert PHY shutdown (register PHY_SHUTDOWNZ);**
   - **Deassert PHY reset (register PHY_RSTZ);**
   - **Deassert CSI reset (register CSI2_RESETN);**

- **(Optional) Program Data IDs for matching error reporting (registers DATA_IDS_1 and DATA_IDS_2);**
- **(Optional) Program the interrupt masks (registers MASK1 and MASK2);**

5. **CSI2 Controller programming** - Read the PHY status register (PHY_STATE) to confirm that all data and clock lanes of the D-PHY are in Stop State (i.e. ready to receive data);
6. **Configure the MIPI Camera Sensor** - Access Camera Sensor using CCI interface to initialize and configure the Camera Sensor to start transmitting a clock on the D-PHY clock lane;
7. **CSI2 Controller programming** - Read the PHY status register (PHY_STATE) to confirm that the D-PHY is receiving a clock on the D-PHY clock lane;

NOTE

Additional steps may be required to correctly configure the specific D-PHY and MIPI Camera Sensor(s) that are part of the system, as well as any other requirements that is integral part of the relevant MIPI Specifications.

41.3.2 Interrupt mechanism

The CSI-2 Host Controller provides an interrupt mechanism that is usable mostly for monitoring errors and debugging.

There are two interrupt signals, *intr1* and *intr2*, which are synchronous with the AMBA-APB clock signal. Registers MASK1 and MASK2 are used to select which bits of registers ERR1 and ERR2 are able to generate interrupts by asserting signals *intr1* and *intr2*, respectively. Both ERR1 and ERR2 will always contain the information of events, irrespective of the state of MASK1 and MASK2. Registers ERR1 and ERR2 will self-clear after a read access. Interrupt signals *intr1* and *intr2* will be de-asserted upon read access of register ERR1 and ERR2, respectively.

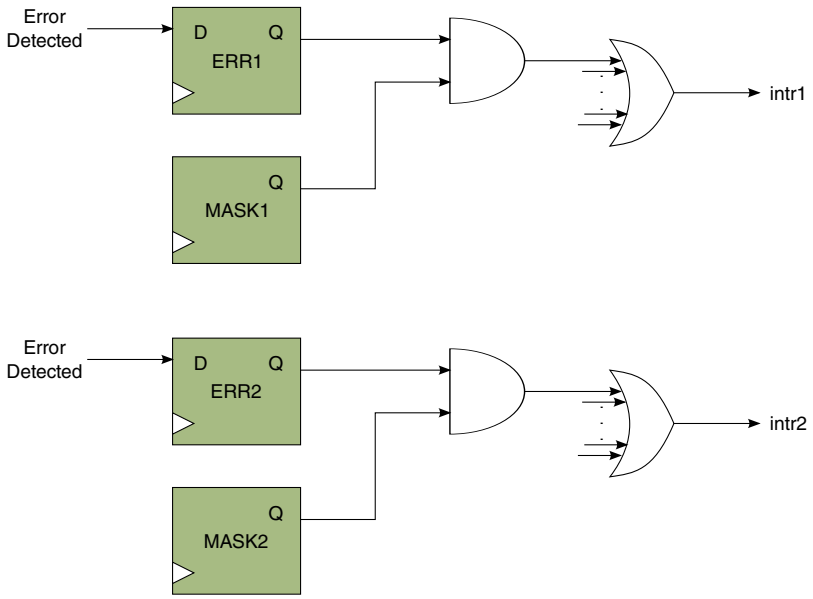


Figure 41-4. Block Diagram presenting the Interrupt mechanism.

41.3.3 Signals

Table 41-1. D-PHY External interface signals

Pin Name	Width	Direction	Description
AVDD	1	Input	D-PHY Analog power supply.
VDD	1	Input	D-PHY Digital power supply.
AVDDREF	1	Input	D-PHY Analog supply for reference generator.
AGND	1	Input	D-PHY Analog supply ground return.
VSS	1	Input	D-PHY Digital supply ground return.
AGNDREF	1	Input	D-PHY Analog supply ground return for reference generator.
REXT	1	Input	D-PHY External resistor connection.
CSI2_CLKP	1	Input	D-PHY Positive D-Phy differential clock line Receiver input
CSI2_CLKN	1	Input	D-PHY Negative D-Phy differential clock line Receiver input
CSI2_DATAP0	1	Input	D-PHY Positive D-Phy differential data line Receiver input , Lane 0
CSI2_DATAN0	1	Input	D-PHY Negative D-Phy differential data line Receiver input , Lane 0
CSI2_DATAP1	1	Input	D-PHY Positive D-Phy differential data line Receiver input , Lane 1
CSI2_DATAN1	1	Input	D-PHY Negative D-Phy differential data line Receiver input , Lane 1

Table continues on the next page...

Table 41-1. D-PHY External interface signals (continued)

CSI2_DATAP2	1	Input	D-PHY Positive D-Phy differential data line Receiver input , Lane 2
CSI2_DATAN2	1	Input	D-PHY Negative D-Phy differential data line Receiver input , Lane 2
CSI2_DATAP3	1	Input	D-PHY Positive D-Phy differential data line Receiver input , Lane 3
CSI2_DATAN3	1	Input	D-PHY Negative D-Phy differential data line Receiver input , Lane 3

41.4 Timing Interfaces

41.4.1 Image Data Interface

At the Image Data Interface, signal *header_en* is used to indicate that new data is being transferred. It rises when a new packet becomes available at the interface and falls as soon as a packet finishes. Between two consecutive packets, there is always a fall and a rise of *header_en*, since the CSI-2 transmitter must enter Low-Power State between an End of Transmission and the following Start of Transmission.

An example of transferring two short packets is presented in the following figure. The fields of the header packet become available simultaneously to the rise of *header_en*, which falls after one clock cycle, as no new data is to be transferred. This is the behavior of the circuit, independently of the number of active lanes.

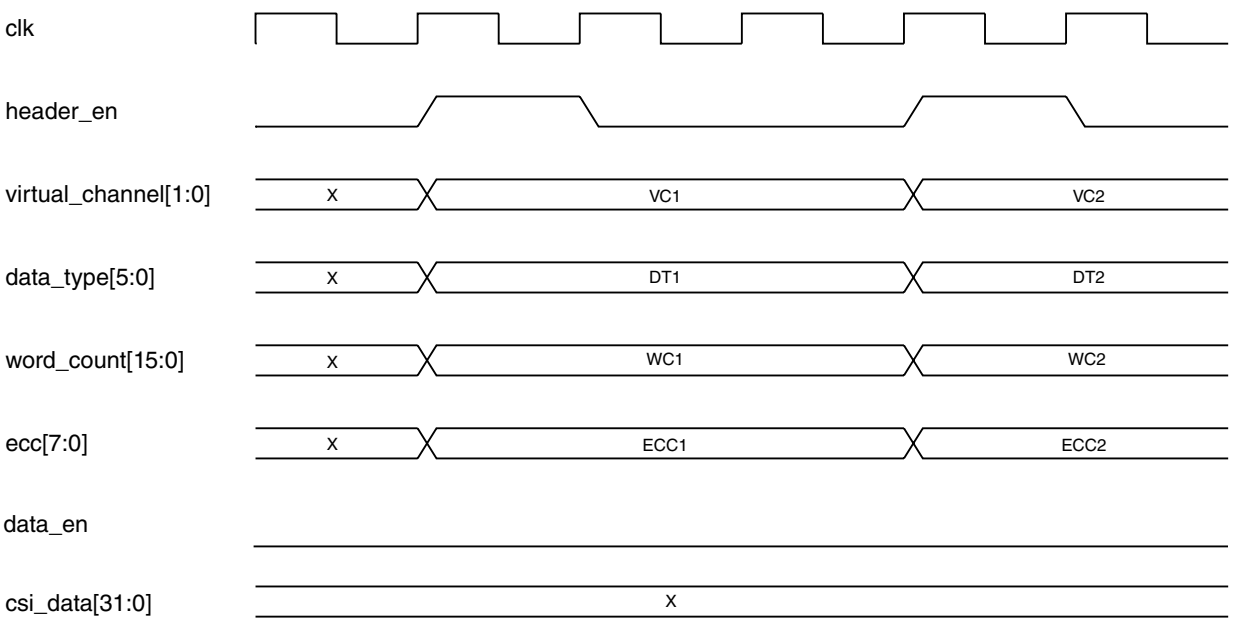


Figure 41-5. Image Data Interface example with two short packets.

When transferring long packets, the data provided by the payload on all lanes is transferred separately on *csi_data* bus, while the header fields remain stable until the transfer of the packet has completed.

Signal *data_en* is used to indicate that a new 32-bit word is available in *csi_data*, and it can only be set if *header_en* is also set. The following Figure shows an example of transferring a long packet received from 4 data lanes. Since a new 32-bit word is transferred at each clock cycle, signal *data_en* remains set until all data is transferred.

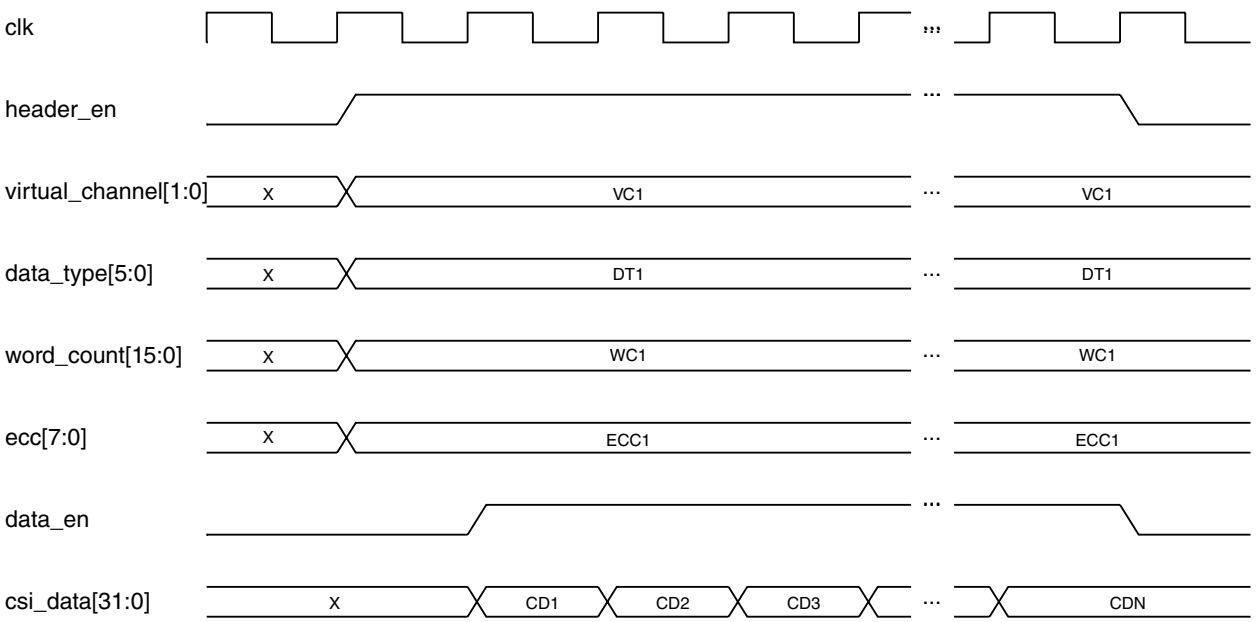


Figure 41-6. Timing interface for a long packet, receiving data from 4 lanes.

If less than 4 lanes are used, a 32-bit word might take more than one clock cycle to be received. In this case, *data_en* is set only when a new word has been fully received and becomes available at the interface. As shown in the following figure, when data is received from two lanes a new word is released every two clock cycles. Both *header_en* and *data_en* return to 0 as soon as all the data has been transferred.

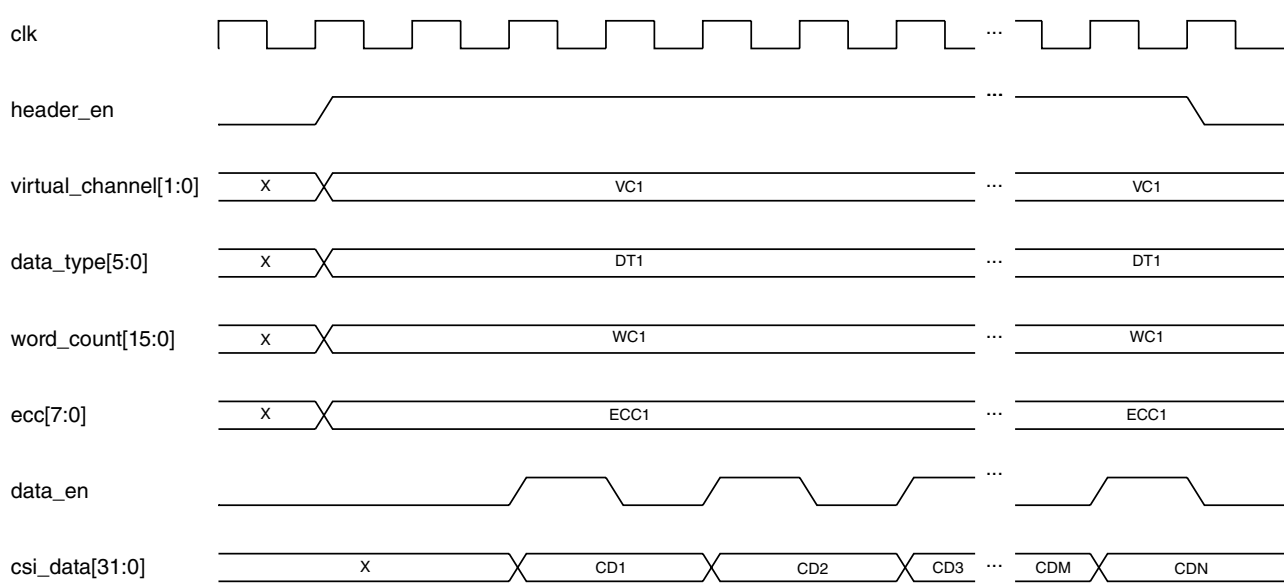


Figure 41-7. Timing interface for a long packet, receiving data from 2 lanes.

If data is transferred using only 1 lane, a new 32-bit word becomes available every four clock cycles. In the case of 3 lanes, a new word is available at clock cycle except every fourth.

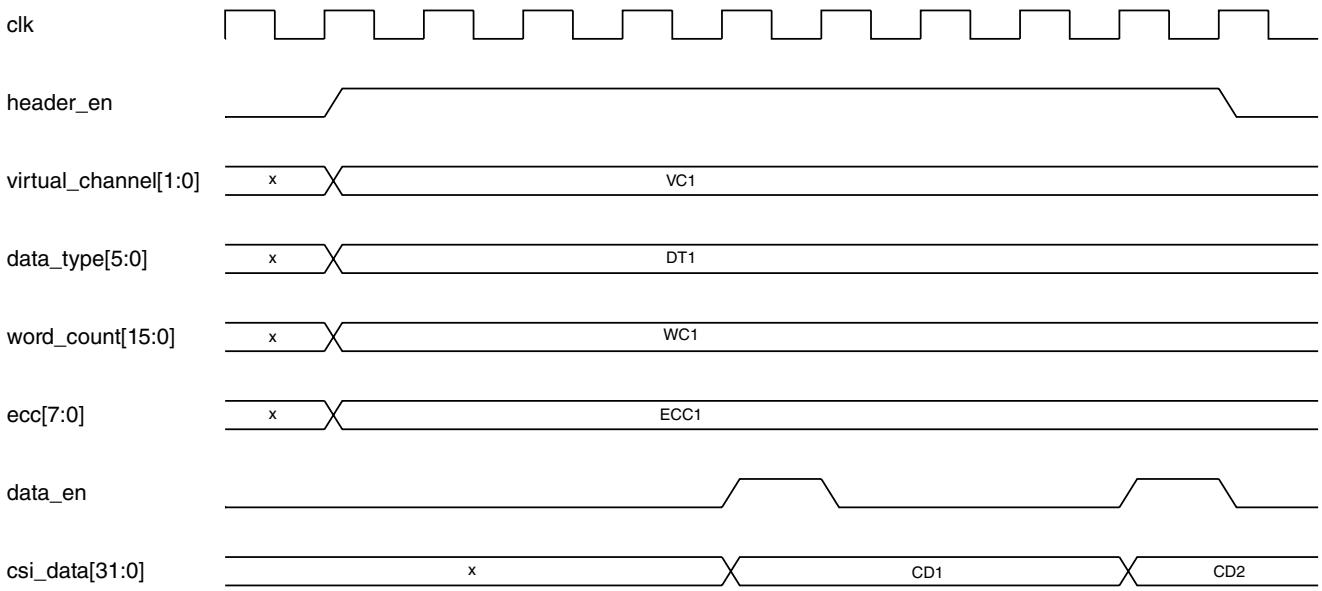


Figure 41-8. Timing interface for a long packet, receiving data from 1 lane.

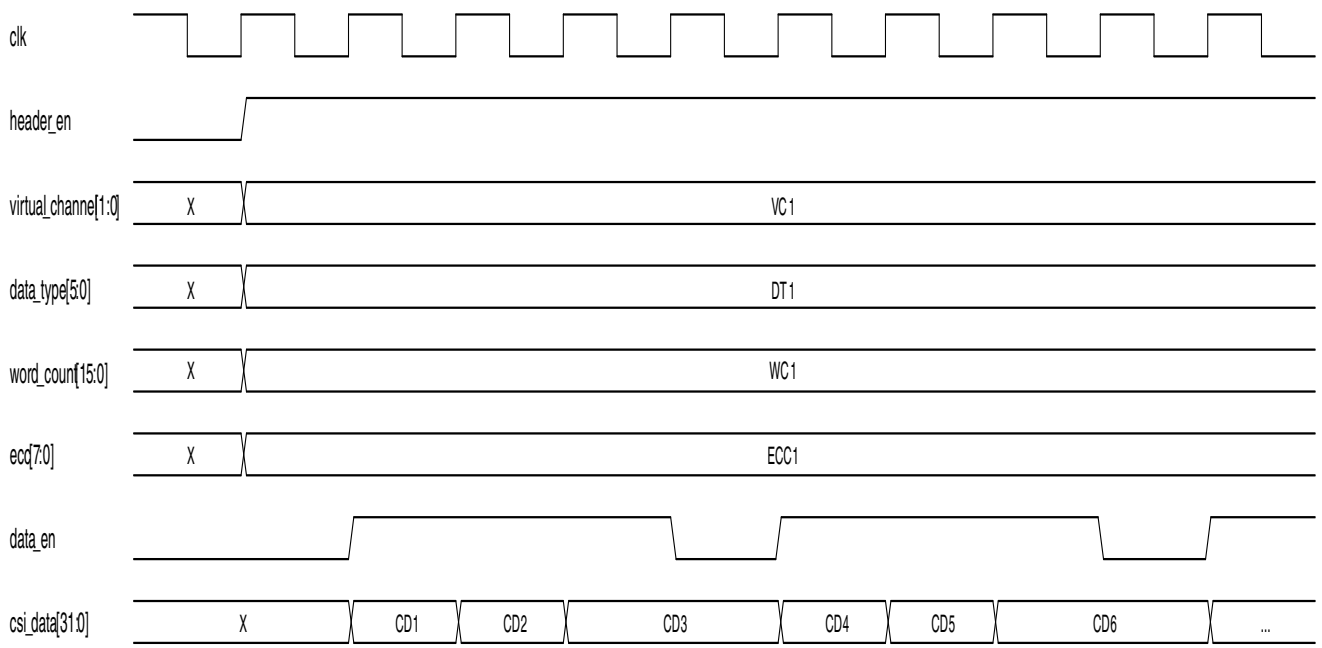


Figure 41-9. Timing interface for a long packet, receiving data from 3 lanes.

Signals *dvalid*, *hvalid* and *vvalid*, are used provide information about Frame Start, Frame End, Line Start and Line End packets. These signals are included in the interface for synchronization purposes.

dvalid is used to indicate when data is being transferred, excluding header information. Data sent through Blank or Null packets do not activate *dvalid*.

hvalid is set on detection of Line Start packets and unset by Line End ones. These packets can be used as a reference for synchronization, even though relevant data can be more or less delayed between them, as it might be surrounded by Blanking Periods, before and after the data. Because Line Start and Line End packets are optional, in case they are not available, *hvalid* adopts the same behavior of *dvalid*, and is not activated by Blank or Null packets. In the case a new Line Start packet is received without a Line End packet being received, a pulse will be generated in *hvalid* to signal that a new Line Start packet was received.

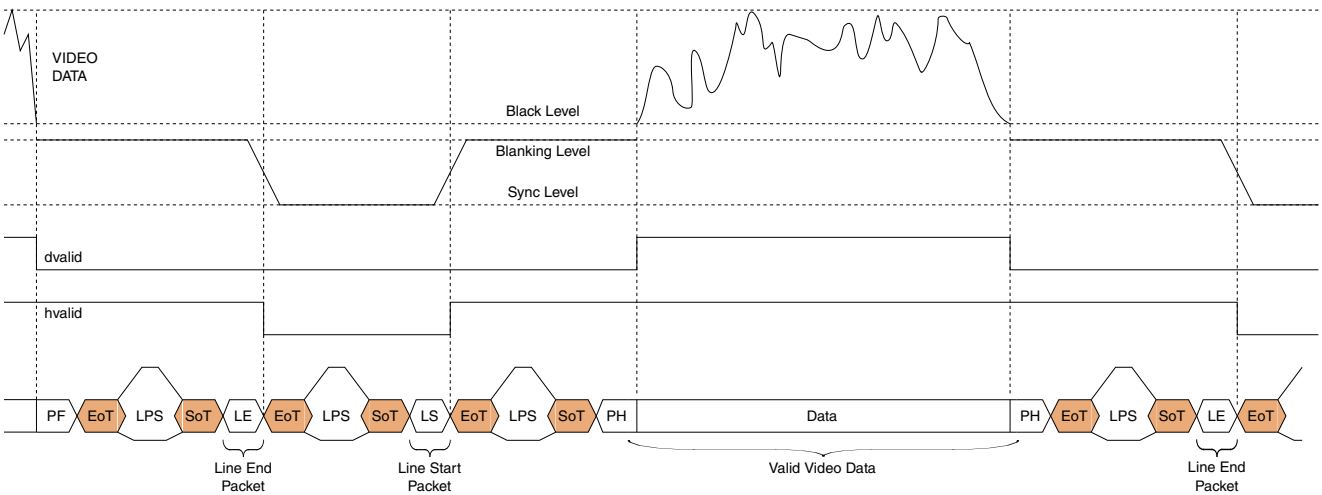


Figure 41-10. HVALID synchronization signal.

vvalid is set on detection of Frame Start packets and unset by Frame End ones. As these packets are mandatory, *vvalid* is always reliable for video applications to synchronize frame updating. In the case a new Frame Start packet is received without a Frame End packet being received, a pulse will be generated in *hvalid* to signal that a new Frame Start packet was received.

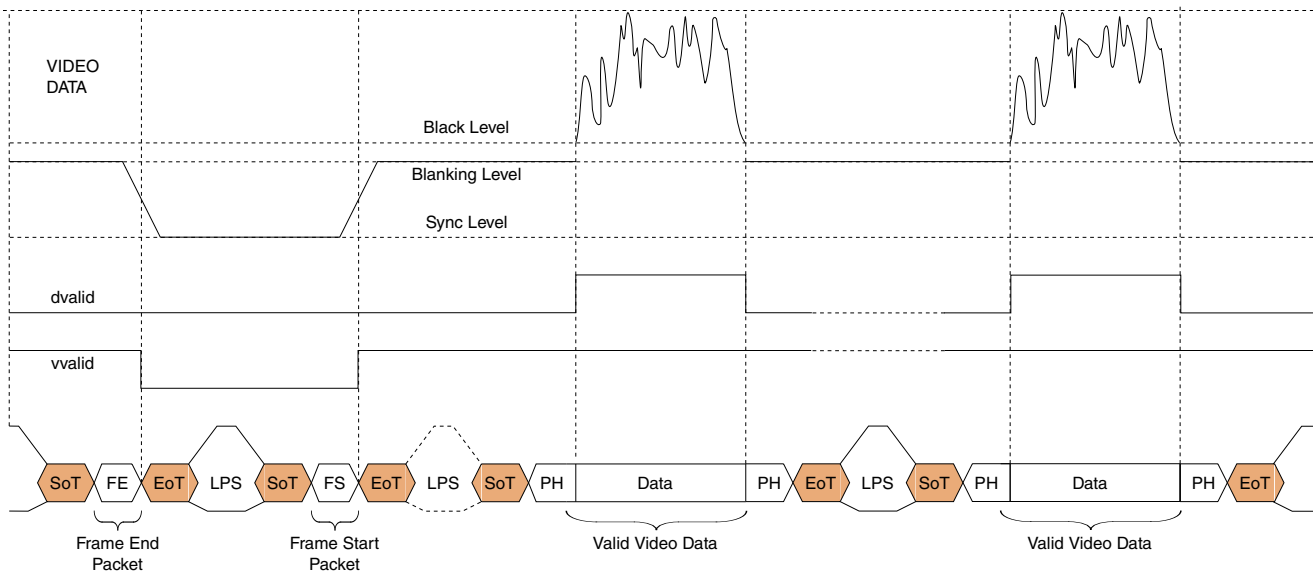


Figure 41-11. VVALID synchronization signal.

41.5 Payload Data Output Format

The Image Data Interface delivers payload data in a common data storage format, as suggested in CSI-2 Specification and described below.

The following sections describe how different data formats are transferred in output bus *csi_data*.

41.5.1 General/Arbitrary Data Reception

In the generic case and for arbitrary data, the first byte of payload data transmitted maps the least significant byte of the 32-bit memory word and the fourth byte of payload data transmitted maps to the most significant byte of the 32-bit memory word.

The following Figure illustrates the generic CSI-2 byte to 32-bit memory word mapping rule.

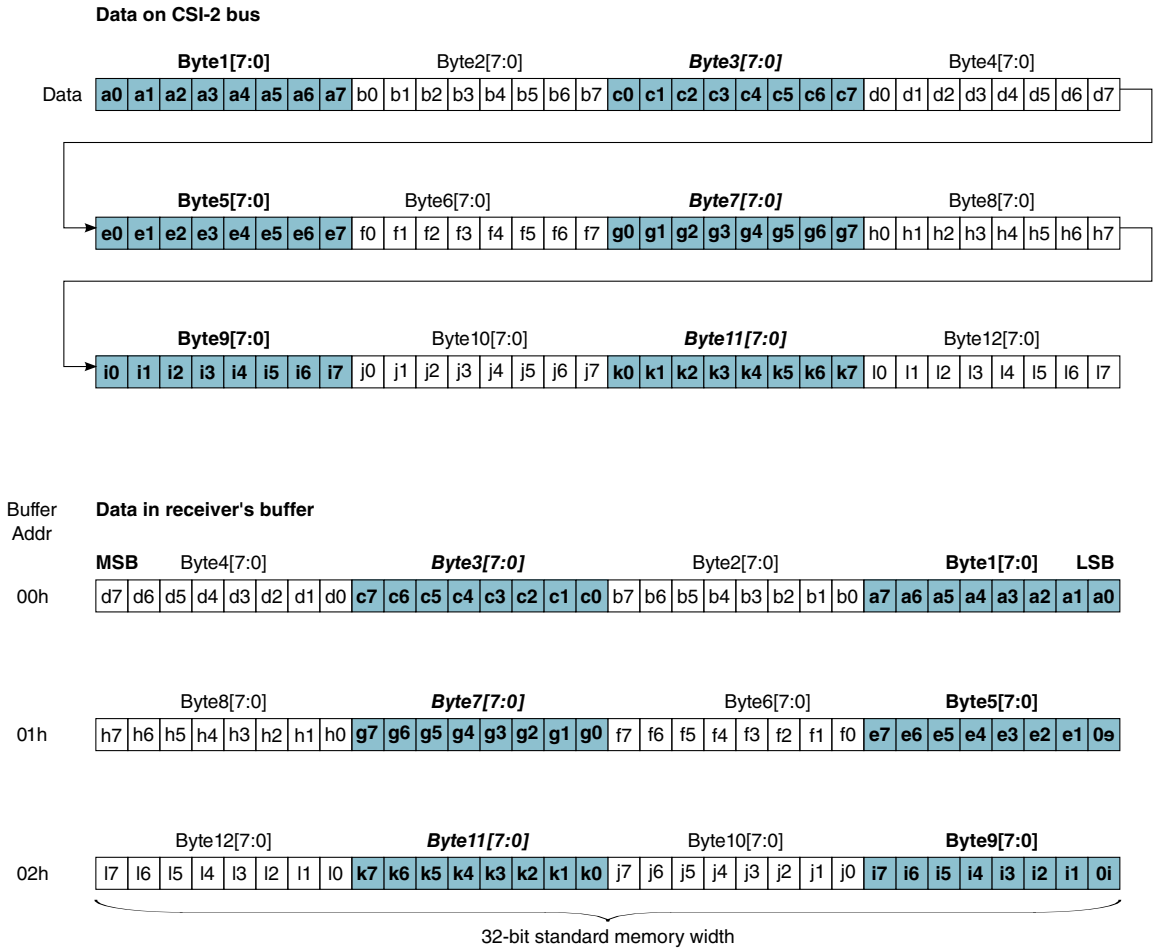


Figure 41-12. General/Arbitrary Data Reception.

41.5.2 RGB888 Data Reception

The RGB888 data format byte to 32-bit memory word mapping follows the generic CSI-2 rule.

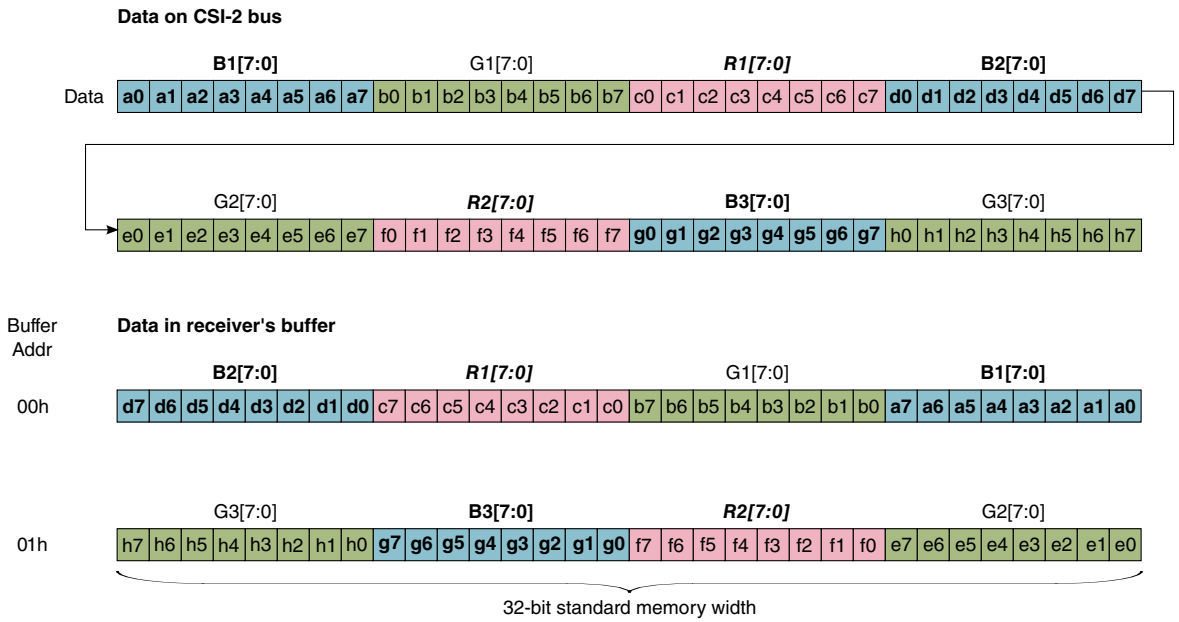


Figure 41-13. RGB888 Data Format Reception.

41.5.3 RGB666 Data Reception

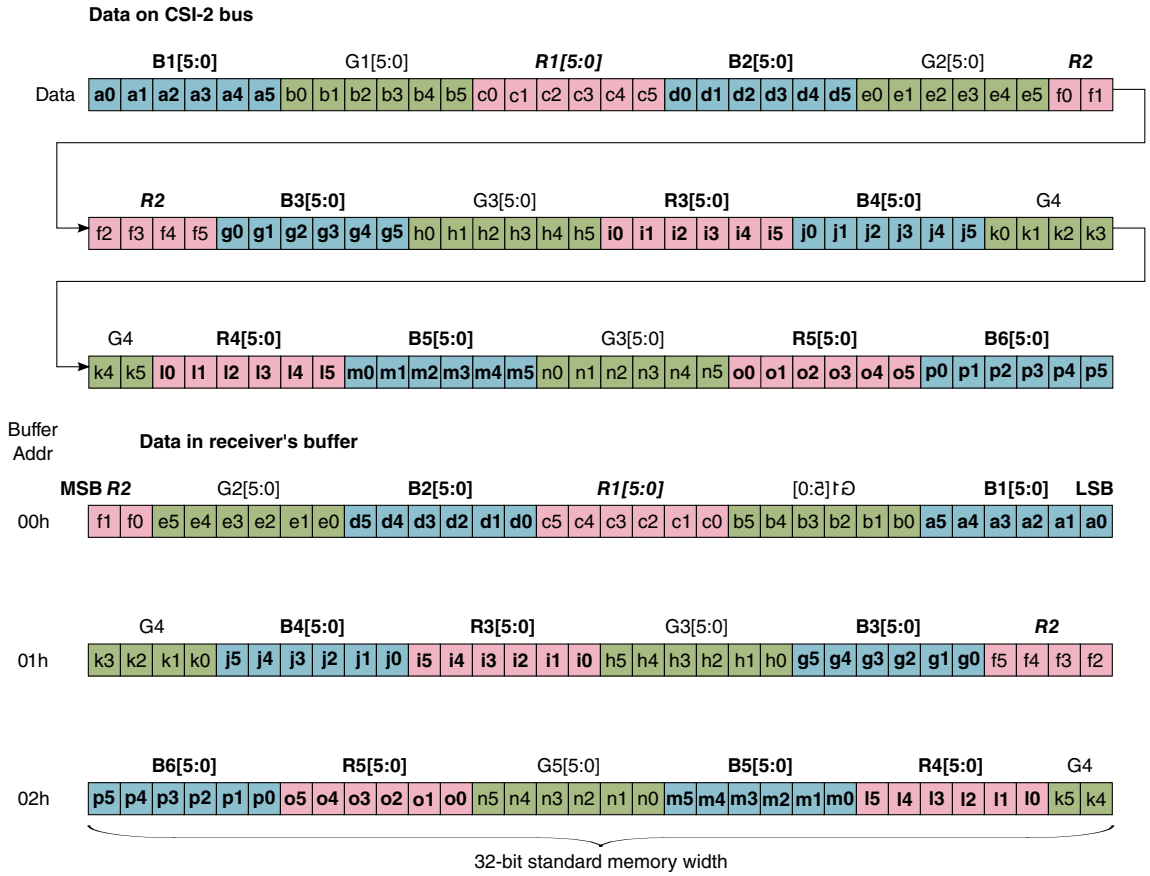


Figure 41-14. RGB666 Data Format Reception.

41.5.4 RGB565 Data Reception

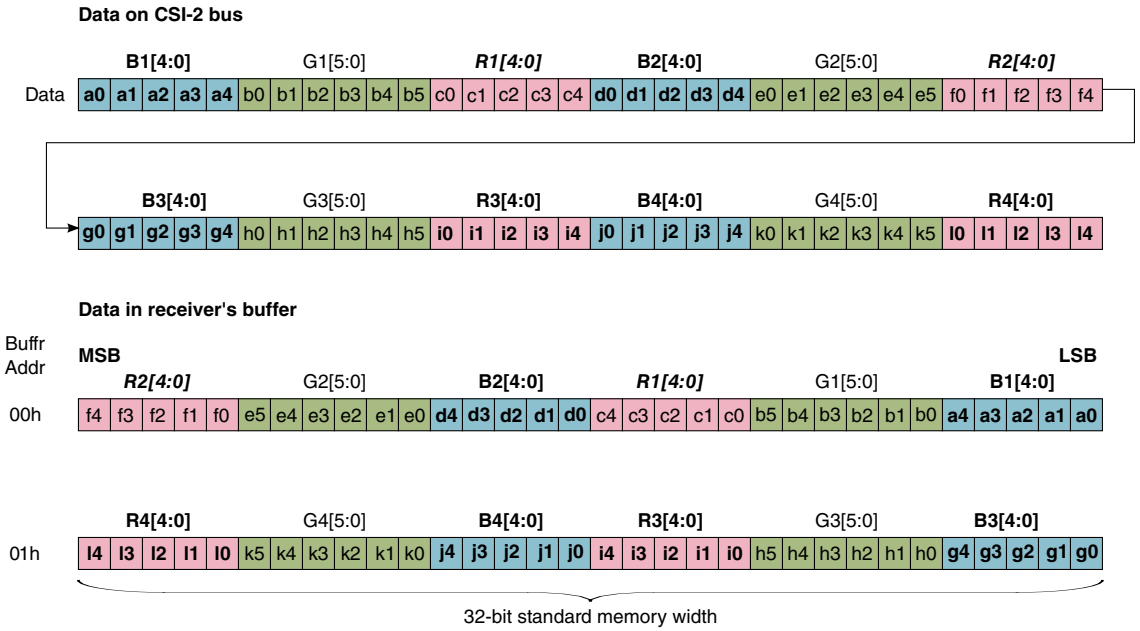


Figure 41-15. RGB565 Data Format Reception.

41.5.5 RGB555 Data Reception

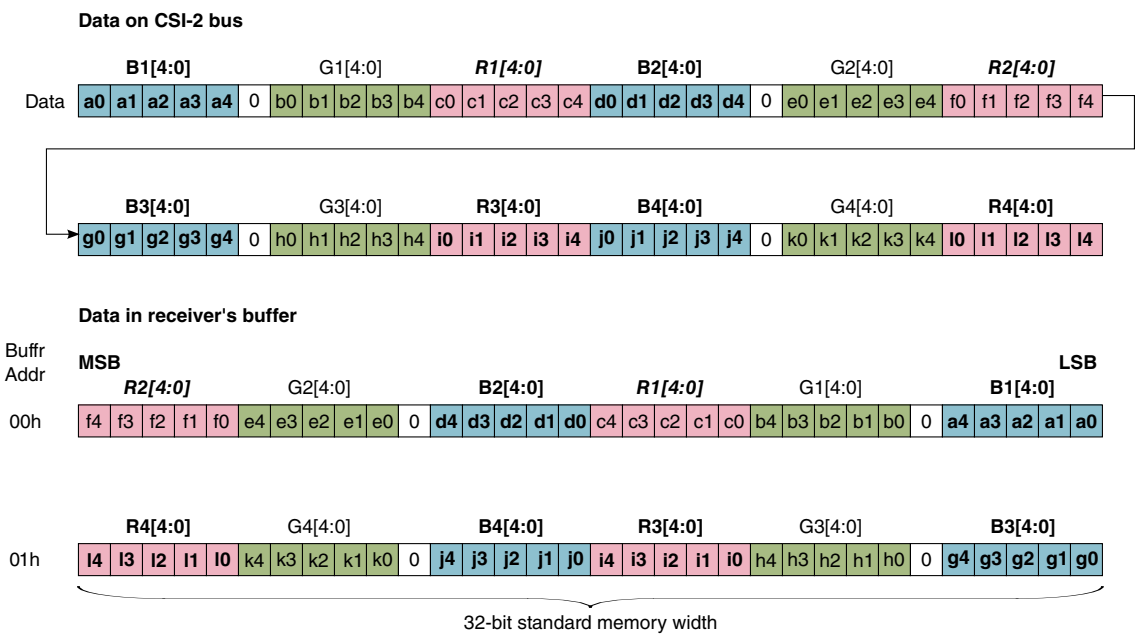


Figure 41-16. RGB555 Data Format Reception.

41.5.6 RGB444 Data Reception

The RGB444 data format byte to 32-bit memory word mapping has a special transform as shown in the following Figure:

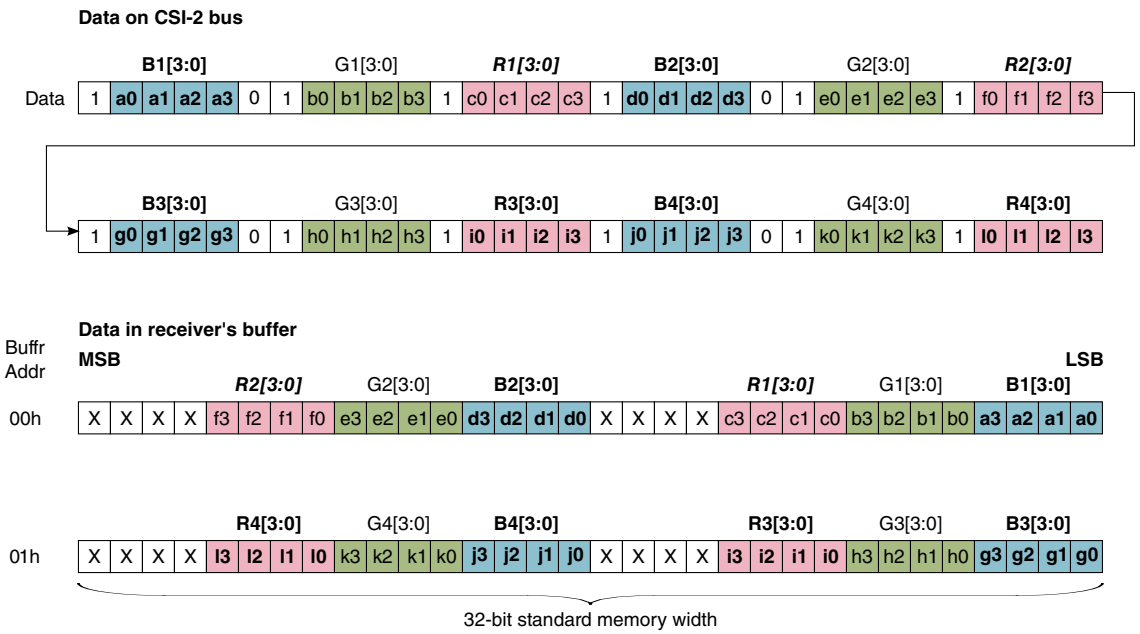


Figure 41-17. RGB444 Data Format Reception.

41.5.7 YUV422 8-bit Data Reception

The YUV422 8-bit data format the byte to 32-bit memory word mapping does not follow the generic CSI-2 rule.

For YUV422 8-bit data format the first byte of payload data transmitted maps the most significant byte of the 32-bit memory word and the fourth byte of payload data transmitted maps to the least significant byte of the 32-bit memory word.

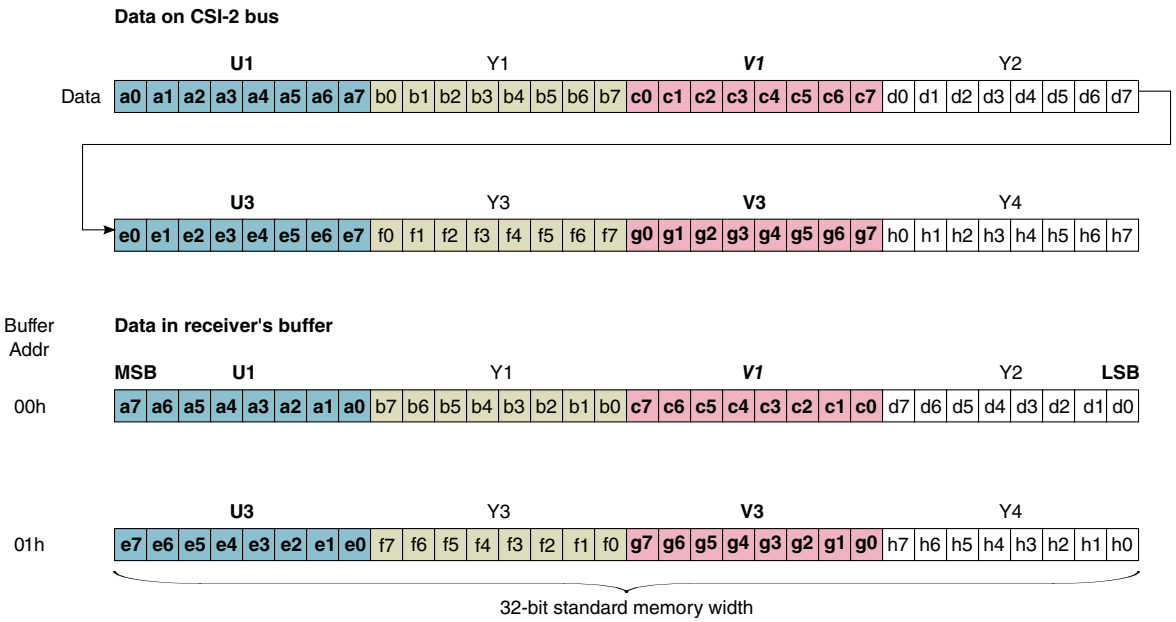


Figure 41-18. YUV422 8-bit Data Format Reception.

41.5.8 YUV422 10-bit Data Reception

The YUV422 10-bit data format the byte to 32-bit memory word mapping follows the generic CSI-2 rule.

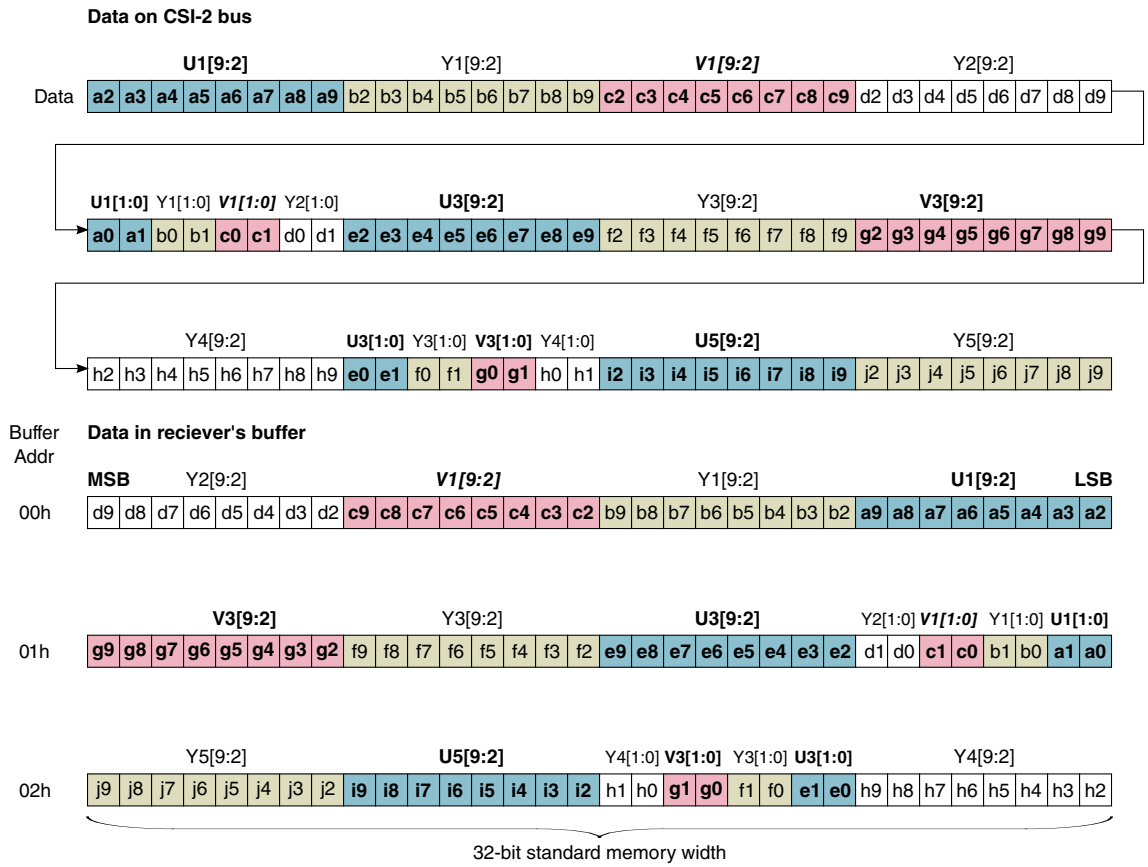


Figure 41-19. YUV422 10-bit Data Format Reception.

41.5.9 YUV420 8-bit (Legacy) Data Reception

The YUV420 8-bit (legacy) data format the byte to 32-bit memory word mapping does not follow the generic CSI-2 rule.

For YUV422 8-bit (legacy) data format the first byte of payload data transmitted maps the MS byte of the 32-bit memory word and the fourth byte of payload data transmitted maps to the LS byte of the 32-bit memory word.

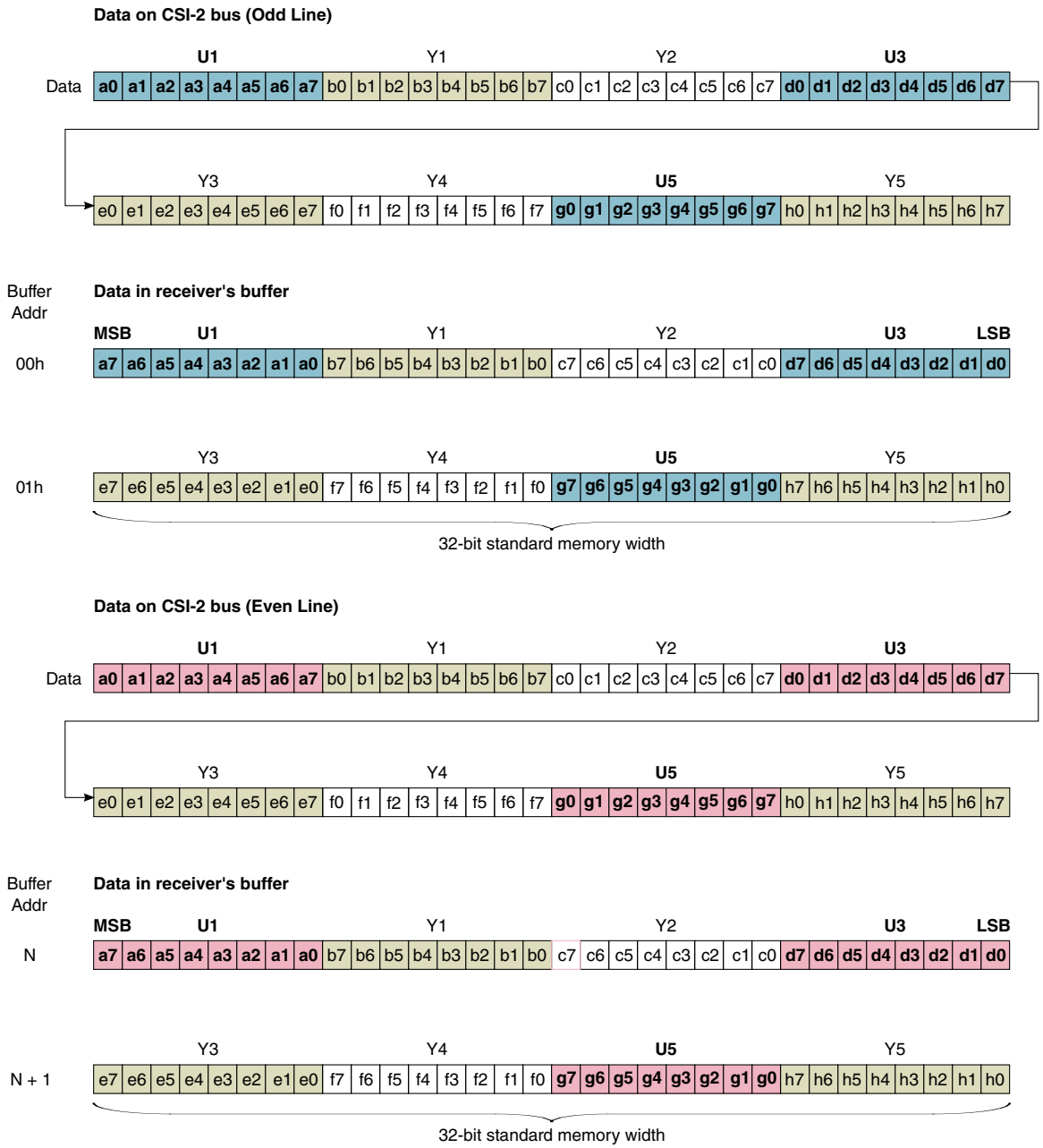


Figure 41-20. YUV420 8-bit Legacy Data Format Reception.

41.5.10 YUV420 8-bit Data Reception

The YUV420 8-bit data format the byte to 32-bit memory word mapping follows the generic CSI-2 rule.

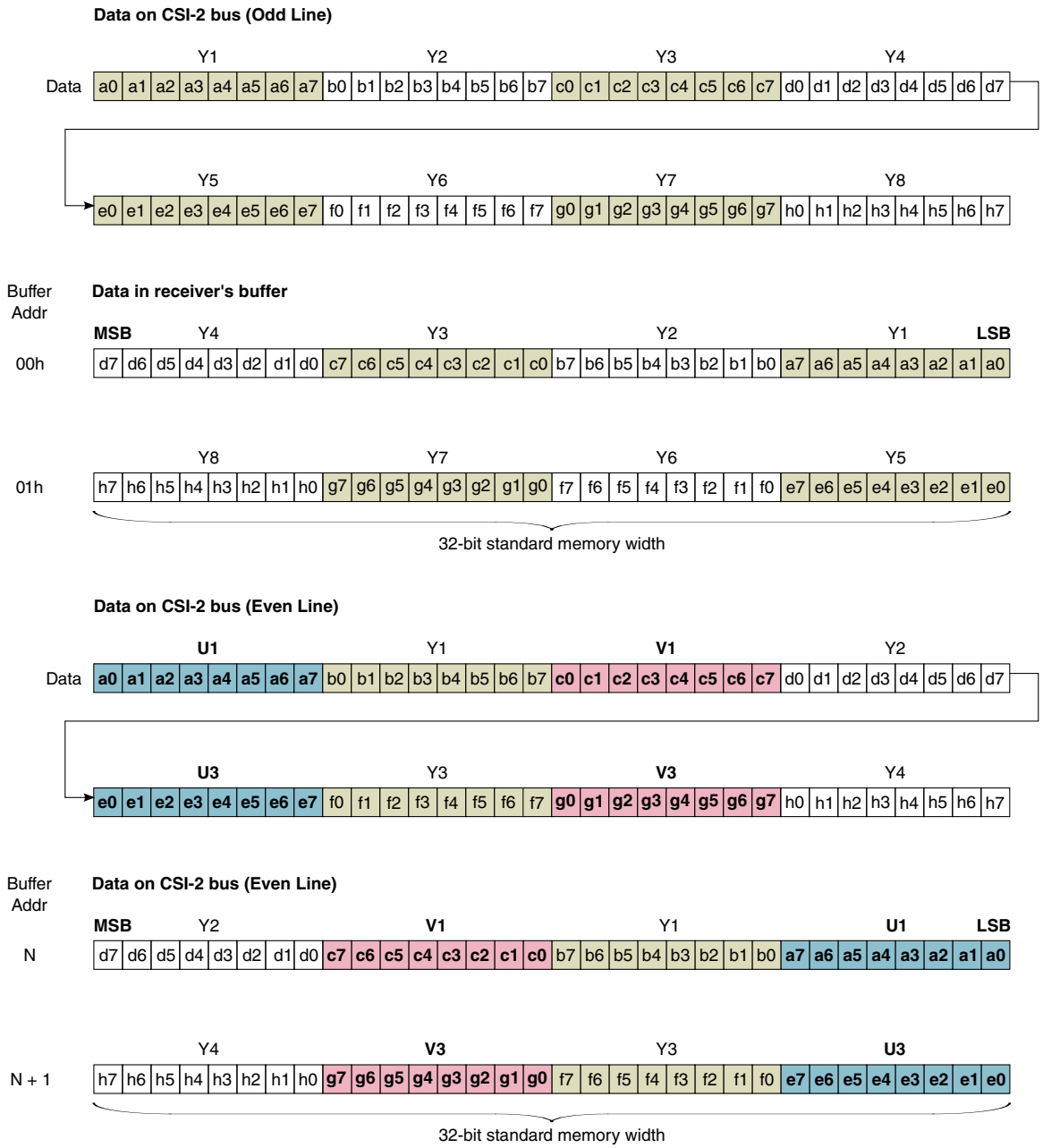


Figure 41-21. YUV420 8-bit Data Format Reception.

41.5.11 YUV420 10-bit Data Reception

The YUV420 10-bit data format the byte to 32-bit memory word mapping follows the generic CSI-2 rule.

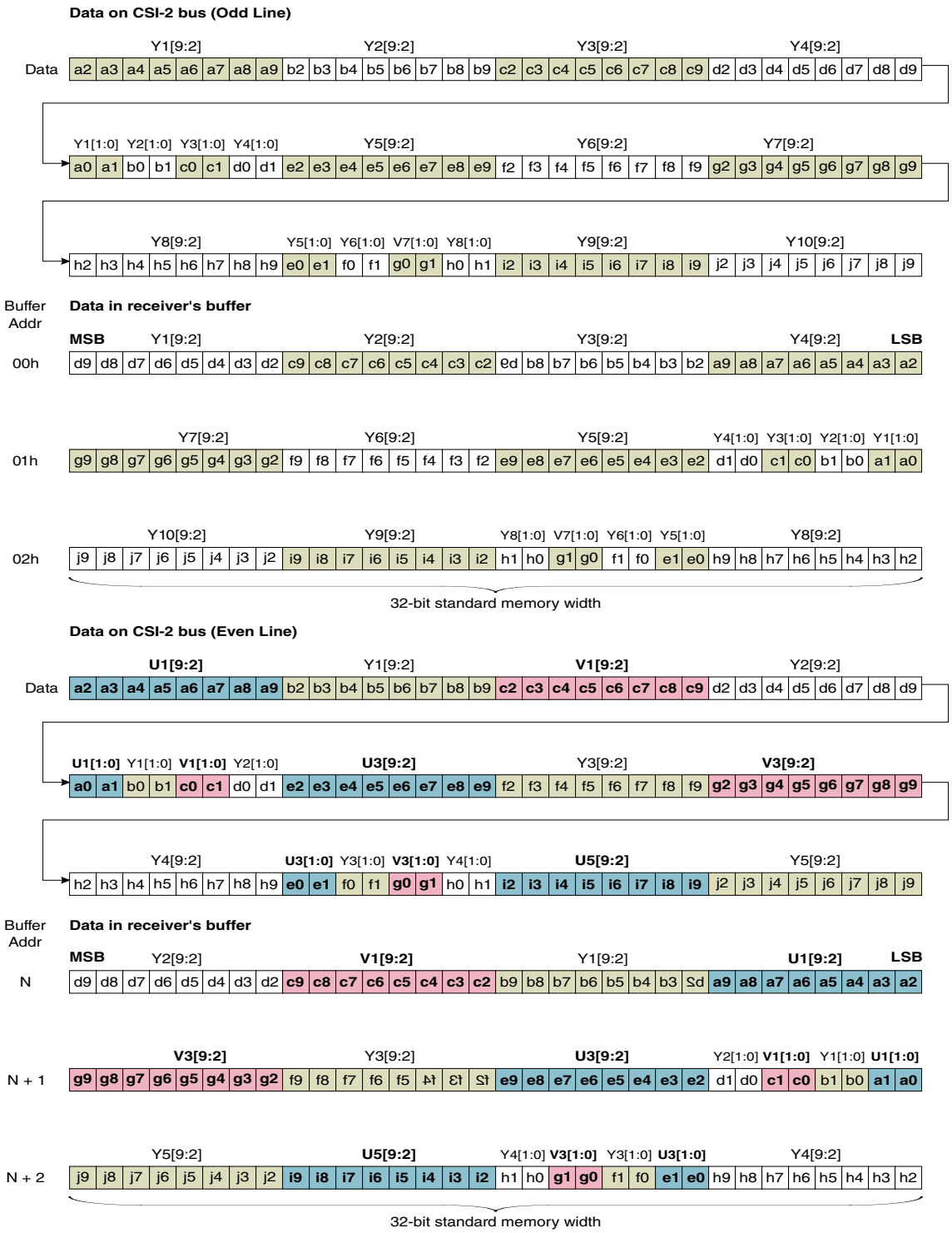


Figure 41-22. YUV420 10-bit Data Format Reception.

41.5.12 RAW6 Data Reception

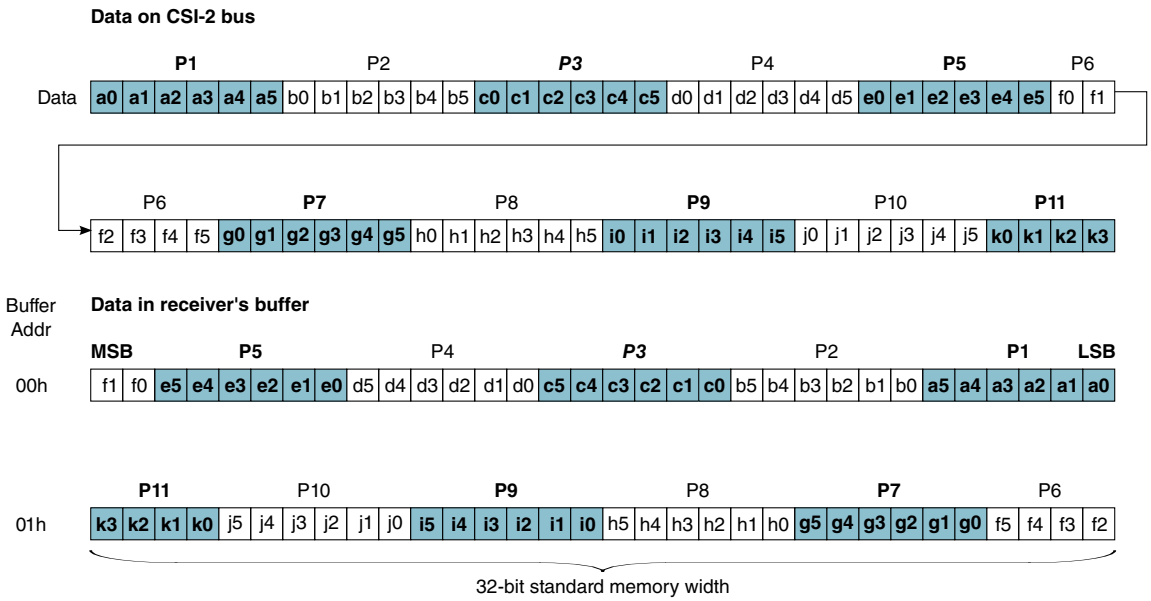


Figure 41-23. RAW6 Data Format Reception.

41.5.13 RAW7 Data Reception

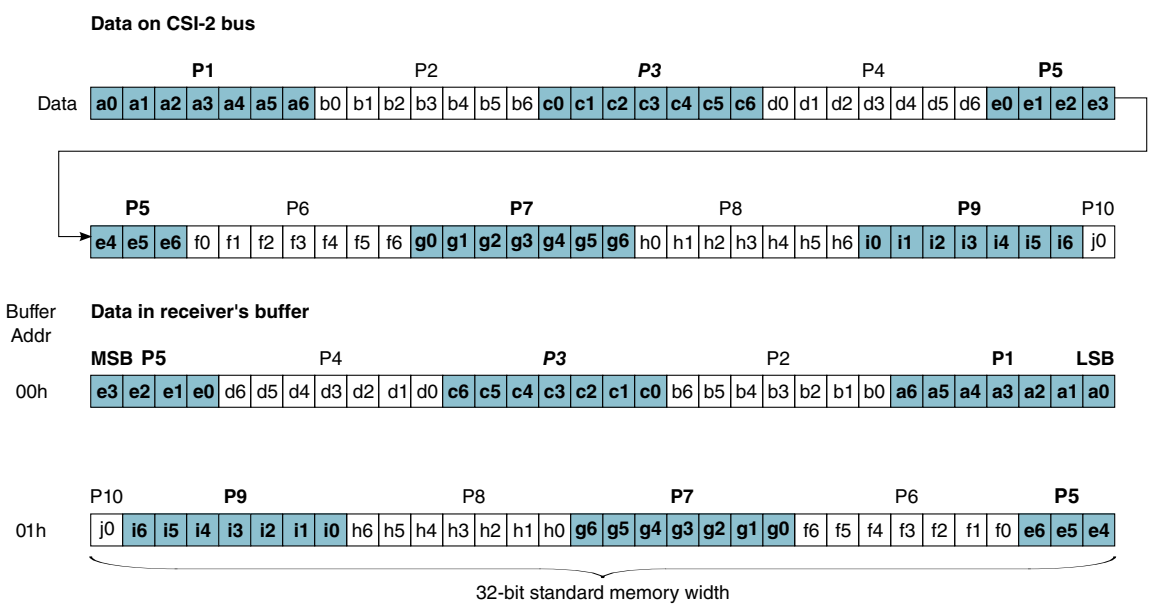


Figure 41-24. RAW7 Data Format Reception.

41.5.14 RAW8 Data Reception

The RAW8 data format the byte to 32-bit memory word mapping follows the generic CSI-2 rule.

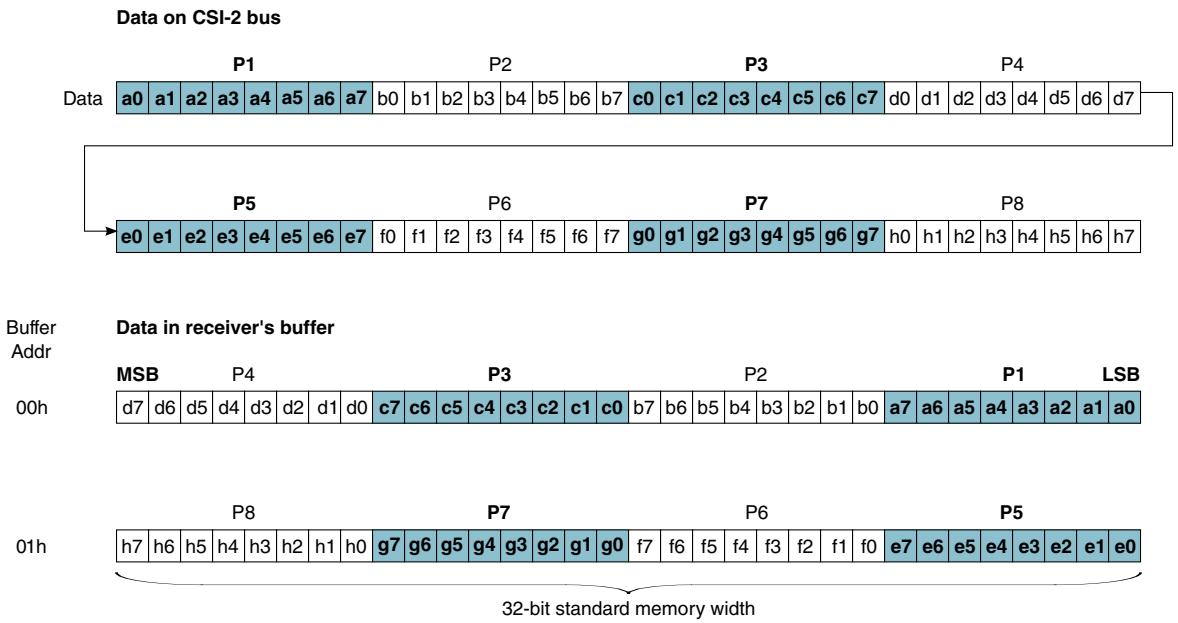


Figure 41-25. RAW8 Data Format Reception.

41.5.15 RAW10 Data Reception

The RAW10 data format the byte to 32-bit memory word mapping follows the generic CSI-2 rule.

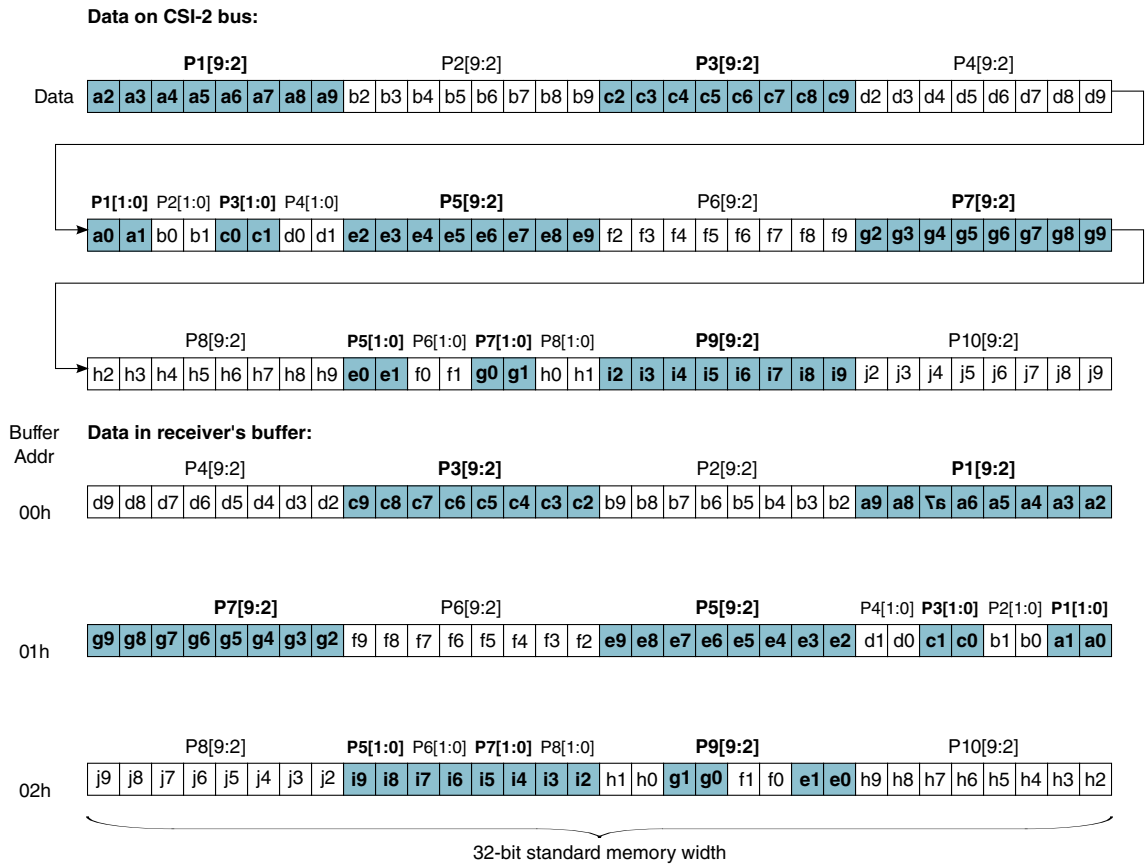


Figure 41-26. RAW10 Data Format Reception.

41.5.16 RAW12 Data Reception

The RAW12 data format the byte to 32-bit memory word mapping follows the generic CSI-2 rule.

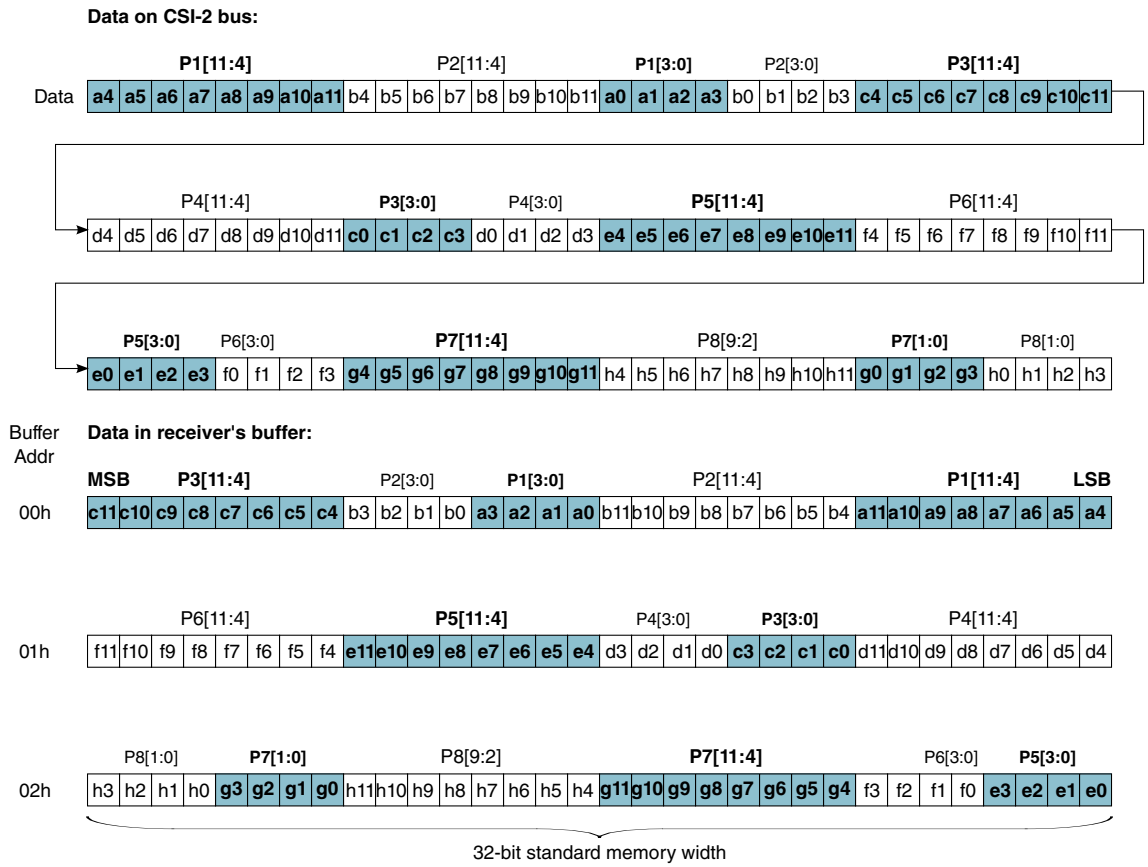


Figure 41-27. RAW12 Data Format Reception.

41.5.17 RAW14 Data Reception

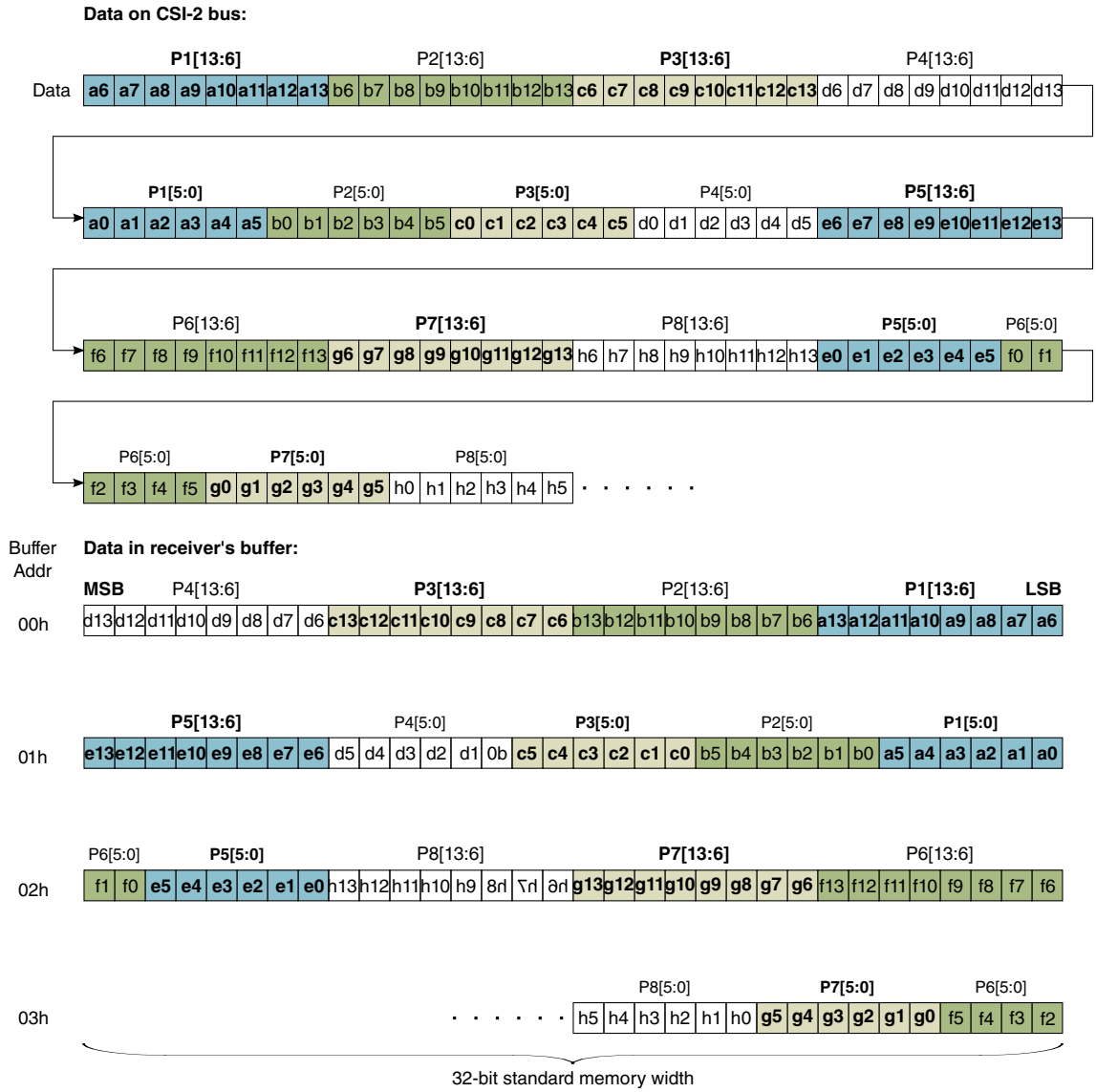


Figure 41-28. RAW 14 Data Format Reception.

41.6 MIPI_CSI Memory Map/Register Definition

MIPI_CSI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21D_C000	Controller Version Identification Register (MIPI_CSI_VERSION)	32	R	0000_0000h	41.6.1/3654

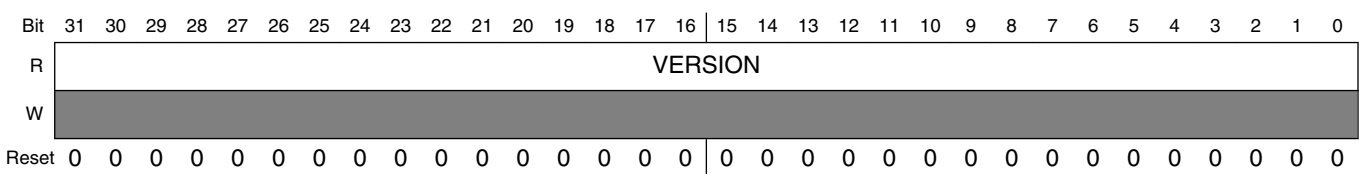
Table continues on the next page...

MIPI_CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21D_C004	Number of Active Data Lanes (MIPI_CSI_N_LANES)	32	R/W	0000_0000h	41.6.2/3655
21D_C008	Phy shutdown control (MIPI_CSI_PHY_SHUTDOWNZ)	32	R/W	0000_0000h	41.6.3/3656
21D_C00C	Phy reset control (MIPI_CSI_DPHY_RSTZ)	32	R/W	0000_0000h	41.6.4/3657
21D_C010	CSI2 controller reset (MIPI_CSI_CSI2_RESETN)	32	R/W	0000_0000h	41.6.5/3657
21D_C014	General settings for all blocks (MIPI_CSI_PHY_STATE)	32	R	0000_0000h	41.6.6/3659
21D_C018	Data IDs for which IDI reports line boundary matching errors (MIPI_CSI_DATA_IDS_1)	32	R/W	0000_0000h	41.6.7/3660
21D_C020	Error state register 1 (MIPI_CSI_ERR1)	32	R	0000_0000h	41.6.8/3662
21D_C024	Error state register 2 (MIPI_CSI_ERR2)	32	R	0000_0000h	41.6.9/3666
21D_C02C	Masks for errors 2 (MIPI_CSI_MASK2)	32	R/W	0000_0000h	41.6.10/ 3668
21D_C030	D-PHY Test interface control 0 (MIPI_CSI_PHY_TST_CRTL0)	32	R/W	0000_0000h	41.6.11/ 3670
21D_C034	D-PHY Test interface control 1 (MIPI_CSI_PHY_TST_CTRL1)	32	R/W	0000_0000h	41.6.12/ 3671

41.6.1 Controller Version Identification Register (MIPI_CSI_VERSION)

Address: 21D_C000h base + 0h offset = 21D_C000h

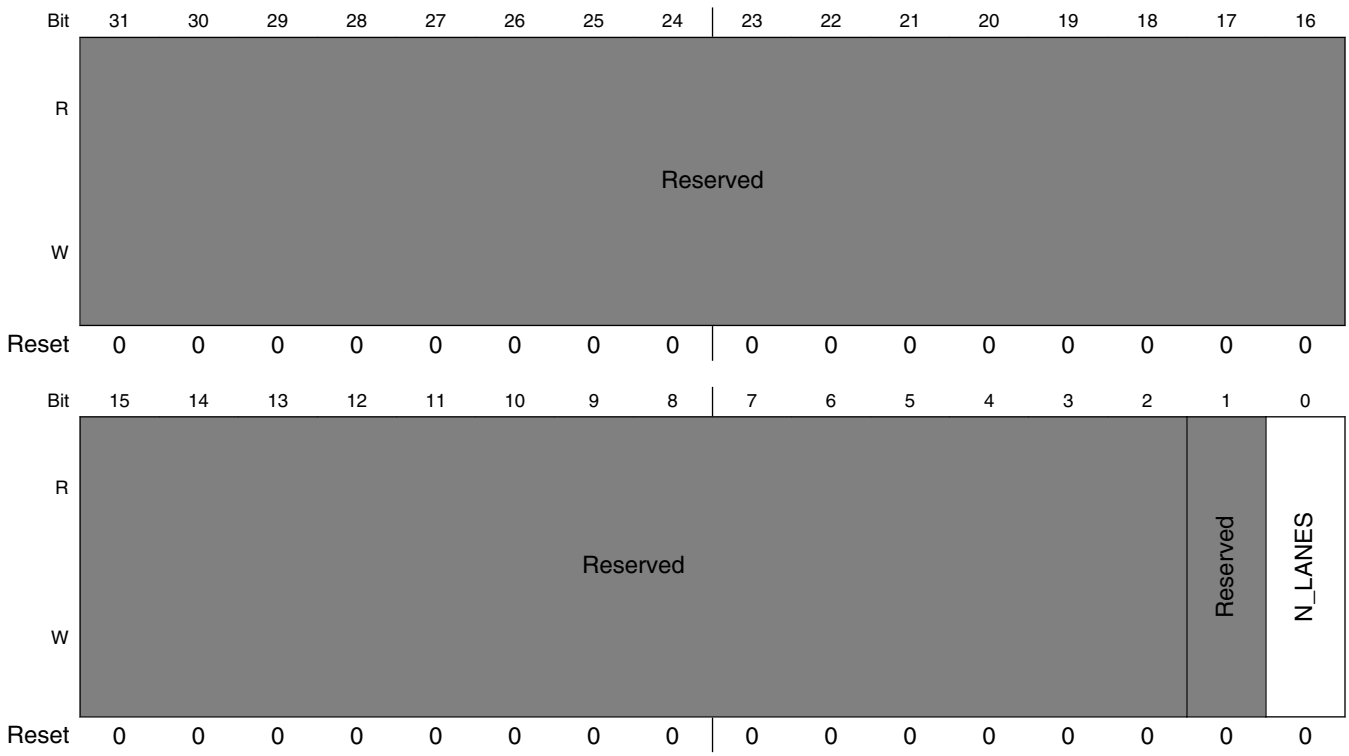


MIPI_CSI_VERSION field descriptions

Field	Description
VERSION	Version of the CSI-2 Host Controller **Default Value:** CSI_VERSION_ID

41.6.2 Number of Active Data Lanes (MIPI_CSI_N_LANES)

Address: 21D_C000h base + 4h offset = 21D_C004h

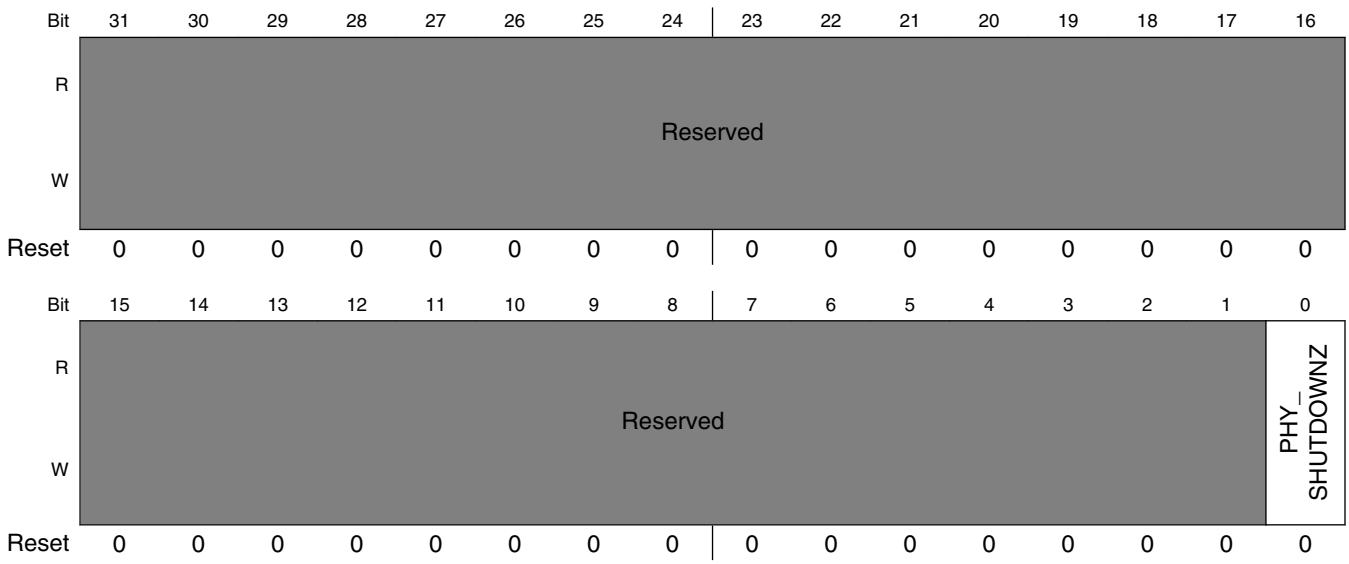


MIPI_CSI_N_LANES field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
1 -	This field is reserved.
0 N_LANES	Number of Active Data Lanes Can only be updated when the PHY lane is in stop state. **Default Value:** CSI_N_LANES 0 1 Data Lane (Lane 0) 1 2 Data Lanes (Lane 0, and 1)

41.6.3 Phy shutdown control (MIPI_CSI_PHY_SHUTDOWNZ)

Address: 21D_C000h base + 8h offset = 21D_C008h

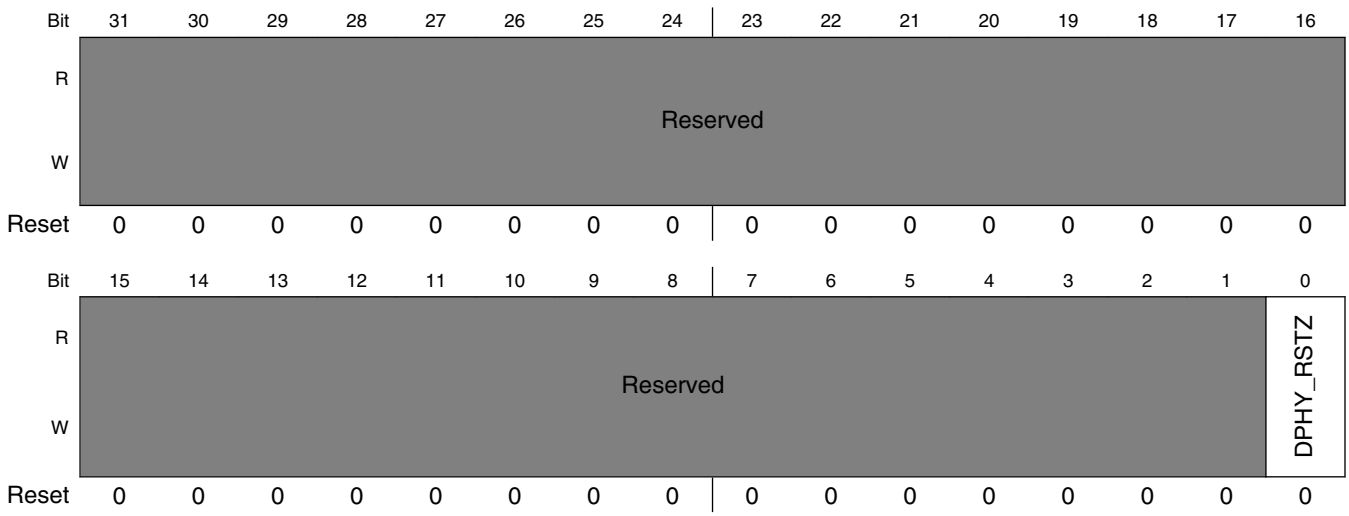


MIPI_CSI_PHY_SHUTDOWNZ field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 PHY_ SHUTDOWNZ	Shutdown input. This line is used to place the complete module in power down. All analog blocks are in power down mode and digital logic is cleared. Active Low **Default Value:** 0

41.6.4 Phy reset control (MIPI_CSI_DPHY_RSTZ)

Address: 21D_C000h base + Ch offset = 21D_C00Ch

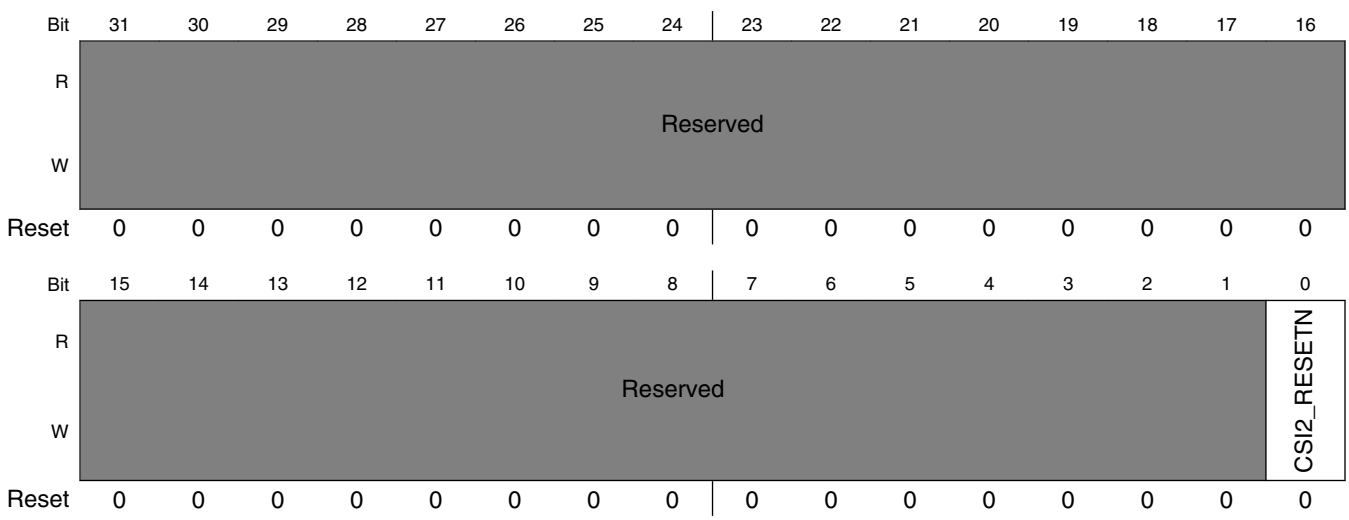


MIPI_CSI_DPHY_RSTZ field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DPHY_RSTZ	DPHY reset output. Active Low **Default Value:** 0

41.6.5 CSI2 controller reset (MIPI_CSI_CSI2_RESETN)

Address: 21D_C000h base + 10h offset = 21D_C010h



MIPI_CSI_CSI2_RESETN field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 CSI2_RESETN	CSI-2 controller reset output. Active Low **Default Value:** 0

41.6.6 General settings for all blocks (MIPI_CSI_PHY_STATE)

Address: 21D_C000h base + 14h offset = 21D_C014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				bypass_2ecc_tst	phy_stopstateclk	phy_rxulpsclknot	phy_rxclkactivehs	Reserved		phy_stopstatedata_1	phy_stopstatedata_0	Reserved		phy_rxulpsesc_1	phy_rxulpsesc_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_CSI_PHY_STATE field descriptions

Field	Description
31–12 -	This field is reserved. Reserved
11 bypass_2ecc_tst	Payload Bypass test mode for double ECC errors **Default Value:** 0
10 phy_stopstateclk	Clock Lane in Stop state **Default Value:** 0
9 phy_rxulpsclknot	Active Low. This signal indicates that the Clock Lane module has entered the Ultra Low Power state **Default Value:** 0
8 phy_rxclkactivehs	Indicates that the clock lane is actively receiving a DDR clock **Default Value:** 0
7–6 -	This field is reserved.
5 phy_ stopstatedata_1	Data Lane 1 in Stop state **Default Value:** 0
4 phy_ stopstatedata_0	Data Lane 0 in Stop state **Default Value:** 0
3–2 -	This field is reserved.
1 phy_rxulpsesc_1	Lane module 1 has entered the Ultra Low Power mode **Default Value:** 0
0 phy_rxulpsesc_0	Lane module 0 has entered the Ultra Low Power mode **Default Value:** 0

41.6.7 Data IDs for which IDI reports line boundary matching errors (MIPI_CSI_DATA_IDS_1)

Address: 21D_C000h base + 18h offset = 21D_C018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	di3_vc		di3_dt						di2_vc		di2_dt						di1_vc		di1_dt						di0_vc		di0_dt					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_CSI_DATA_IDS_1 field descriptions

Field	Description
31–30 di3_vc	Data ID 3 Virtual channel **Default Value:** 0
29–24 di3_dt	Data ID 3 Data Type **Default Value:** 0

Table continues on the next page...

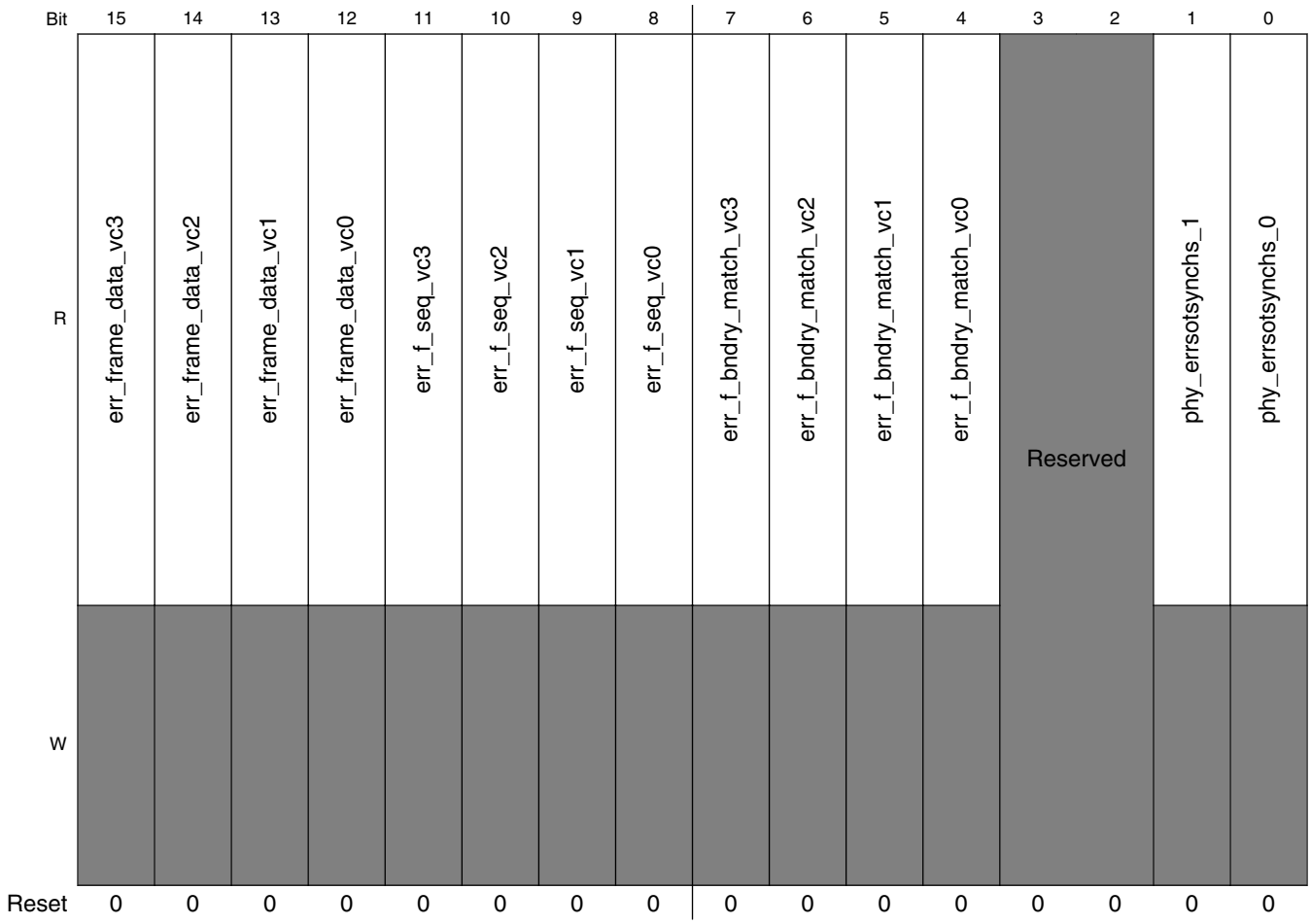
MIPI_CSI_DATA_IDS_1 field descriptions (continued)

Field	Description
23–22 di2_vc	DATA ID 2 Virtual channel **Default Value:** 0
21–16 di2_dt	DATA ID 2 Data Type **Default Value:** 0
15–14 di1_vc	Data ID 1 Virtual channel **Default Value:** 0
13–8 di1_dt	Data ID 1 Data Type **Default Value:** 0
7–6 di0_vc	Data ID 0 Virtual channel **Default Value:** 0
di0_dt	Data ID 0 Data Type **Default Value:** 0

41.6.8 Error state register 1 (MIPI_CSI_ERR1)

Address: 21D_C000h base + 20h offset = 21D_C020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			err_ecc_double	vc3_err_crc	vc2_err_crc	vc1_err_crc	vc0_err_crc	err_l_seq_di3	err_l_seq_di2	err_l_seq_di1	err_l_seq_di0	err_l_bndry_match_di3	err_l_bndry_match_di2	err_l_bndry_match_di1	err_l_bndry_match_di0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



MIPI_CSI_ERR1 field descriptions

Field	Description
31–29 -	This field is reserved. Reserved
28 err_ecc_double	Header ECC contains 2 errors. Unrecoverable. **Default Value:** 0
27 vc3_err_crc	Checksum Error detected on Virtual Channel 3 **Default Value:** 0
26 vc2_err_crc	Checksum Error detected on Virtual Channel 2 **Default Value:** 0
25 vc1_err_crc	Checksum Error detected on Virtual Channel 1 **Default Value:** 0
24 vc0_err_crc	Checksum Error detected on Virtual Channel 0 **Default Value:** 0
23 err_l_seq_di3	Error in the sequence of lines for vc3 and dt3 **Default Value:** 0

Table continues on the next page...

MIPI_CSI_ERR1 field descriptions (continued)

Field	Description
22 err_l_seq_di2	Error in the sequence of lines for vc2 and dt2 **Default Value:** 0
21 err_l_seq_di1	Error in the sequence of lines for vc1 and dt1 **Default Value:** 0
20 err_l_seq_di0	Error in the sequence of lines for vc0 and dt0 **Default Value:** 0
19 err_l_bndry_match_di3	Error matching Line Start with Line End for vc3 and dt3 **Default Value:** 0
18 err_l_bndry_match_di2	Error matching Line Start with Line End for vc2 and dt2 **Default Value:** 0
17 err_l_bndry_match_di1	Error matching Line Start with Line End for vc1 and dt1 **Default Value:** 0
16 err_l_bndry_match_di0	Error matching Line Start with Line End for vc0 and dt0 **Default Value:** 0
15 err_frame_data_vc3	Last received frame, in Virtual Channel 3, had at least one CRC error **Default Value:** 0
14 err_frame_data_vc2	Last received frame, in Virtual Channel 2, had at least one CRC error **Default Value:** 0
13 err_frame_data_vc1	Last received frame, in Virtual Channel 1, had at least one CRC error **Default Value:** 0
12 err_frame_data_vc0	Last received frame, in Virtual Channel 0, had at least one CRC error **Default Value:** 0
11 err_f_seq_vc3	Incorrect Frame Sequence detected in Virtual Channel 3 **Default Value:** 0
10 err_f_seq_vc2	Incorrect Frame Sequence detected in Virtual Channel 2 **Default Value:** 0
9 err_f_seq_vc1	Incorrect Frame Sequence detected in Virtual Channel 1 **Default Value:** 0
8 err_f_seq_vc0	Incorrect Frame Sequence detected in Virtual Channel 0 **Default Value:** 0
7 err_f_bndry_match_vc3	Error matching Frame Start with Frame End for Virtual Channel 3 **Default Value:** 0
6 err_f_bndry_match_vc2	Error matching Frame Start with Frame End for Virtual Channel 2 **Default Value:** 0

Table continues on the next page...

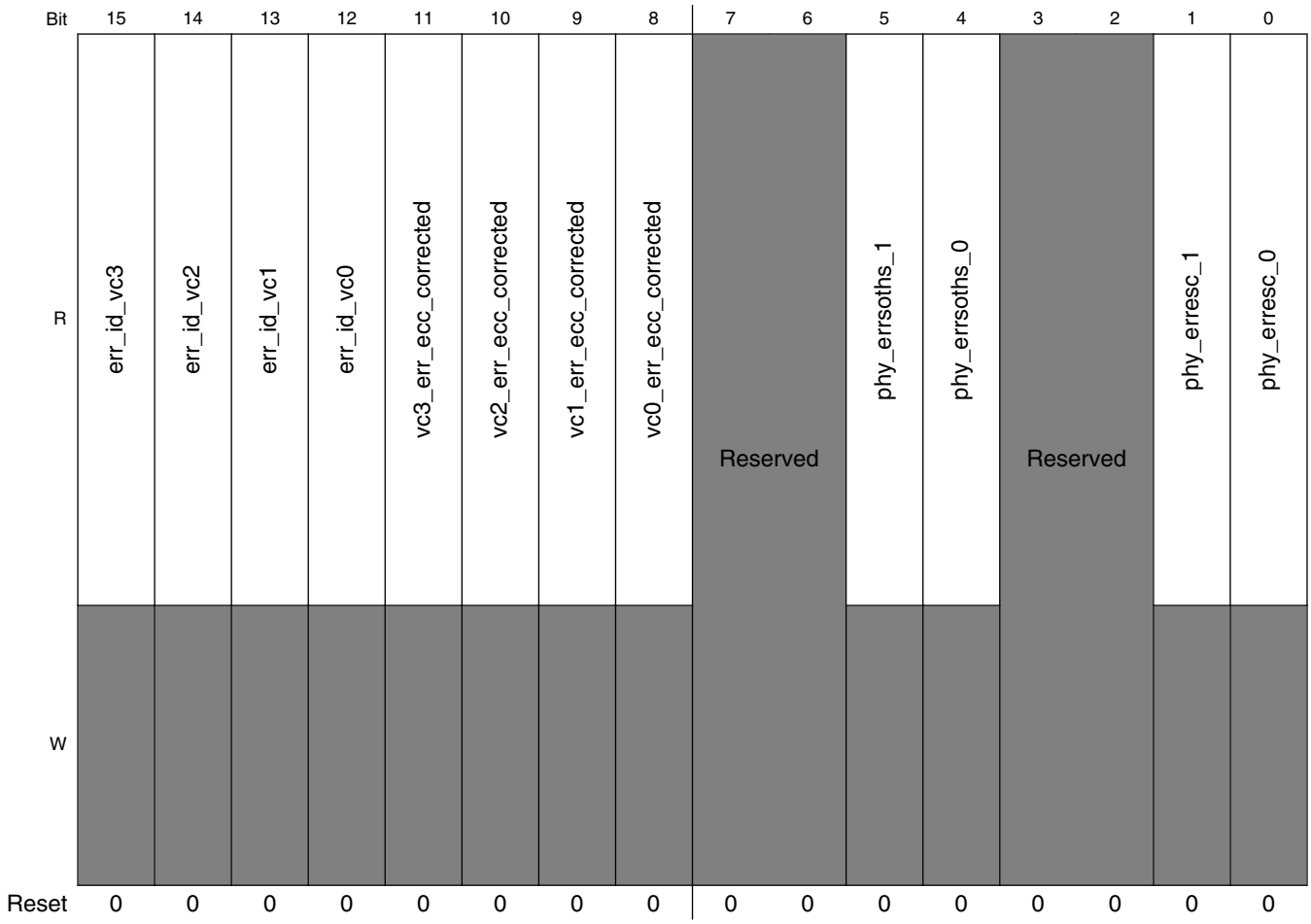
MIPI_CSI_ERR1 field descriptions (continued)

Field	Description
5 err_f_bndry_ match_vc1	Error matching Frame Start with Frame End for Virtual Channel 1 **Default Value:** 0
4 err_f_bndry_ match_vc0	Error matching Frame Start with Frame End for Virtual Channel 0 **Default Value:** 0
3–2 -	This field is reserved.
1 phy_ errsotsynchs_1	Start of Transmission Error on data lane 1 (no synchronization achieved) **Default Value:** 0
0 phy_ errsotsynchs_0	Start of Transmission Error on data lane 0 (no synchronization achieved) **Default Value:** 0

41.6.9 Error state register 2 (MIPI_CSI_ERR2)

Address: 21D_C000h base + 24h offset = 21D_C024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



MIPI_CSI_ERR2 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 -	This field is reserved.
15 err_id_vc3	Unrecognized or unimplemented data type detected in Virtual Channel 3 **Default Value:** 0
14 err_id_vc2	Unrecognized or unimplemented data type detected in Virtual Channel 2 **Default Value:** 0
13 err_id_vc1	Unrecognized or unimplemented data type detected in Virtual Channel 1 **Default Value:** 0
12 err_id_vc0	Unrecognized or unimplemented data type detected in Virtual Channel 0 **Default Value:** 0
11 vc3_err_ecc_ corrected	Header error detected and corrected on Virtual Channel 3 **Default Value:** 0

Table continues on the next page...

MIPI_CSI_ERR2 field descriptions (continued)

Field	Description
10 vc2_err_ecc_ corrected	Header error detected and corrected on Virtual Channel 2 **Default Value:** 0
9 vc1_err_ecc_ corrected	Header error detected and corrected on Virtual Channel 1 **Default Value:** 0
8 vc0_err_ecc_ corrected	Header error detected and corrected on Virtual Channel 0 **Default Value:** 0
7–6 -	This field is reserved.
5 phy_errsoths_1	Start of Transmission Error on data lane 1 (synchronization can still be achieved) **Default Value:** 0
4 phy_errsoths_0	Start of Transmission Error on data lane 0 (synchronization can still be achieved) **Default Value:** 0
3–2 -	This field is reserved.
1 phy_erresc_1	Escape Entry Error (ULPM) on data lane 1 **Default Value:** 0
0 phy_erresc_0	Escape Entry Error (ULPM) on data lane 0 **Default Value:** 0

41.6.10 Masks for errors 2 (MIPI_CSI_MASK2)

Address: 21D_C000h base + 2Ch offset = 21D_C02Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

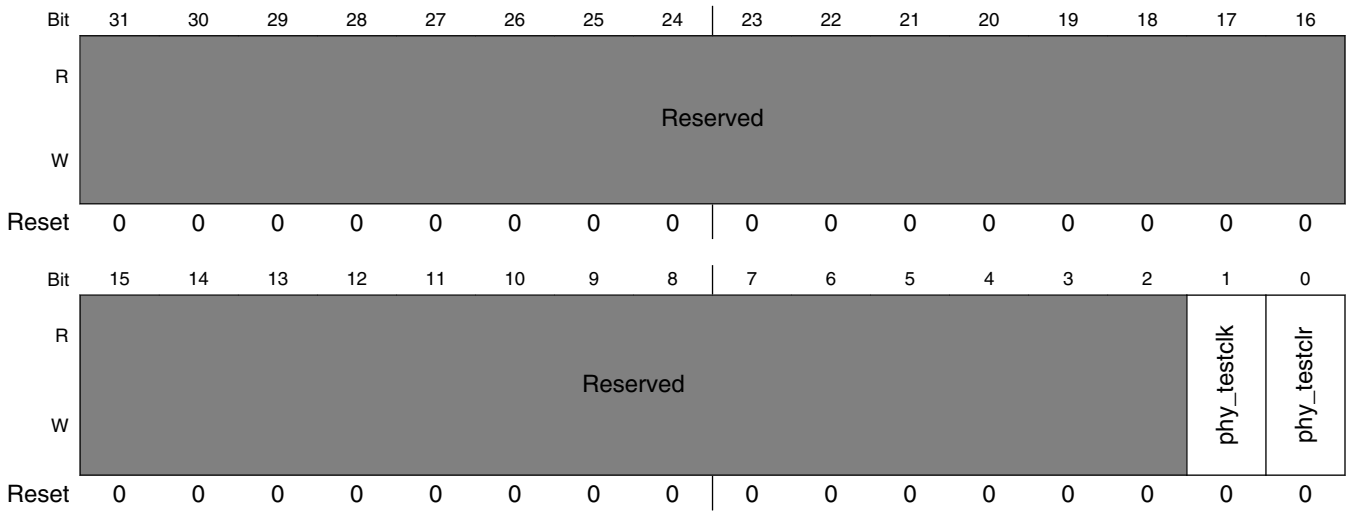
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	mask_err_id_vc3	mask_err_id_vc2	mask_err_id_vc1	mask_err_id_vc0	mask_vc3_err_ecc_corrected	mask_vc2_err_ecc_corrected	mask_vc1_err_ecc_corrected	mask_vc0_err_ecc_corrected	Reserved		mask_phy_errsoths_1	mask_phy_errsoths_0	Reserved		mask_phy_erresc_1	mask_phy_erresc_0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_CSI_MASK2 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 -	This field is reserved.
15 mask_err_id_vc3	Mask for err_id_vc3. **Default Value:** 0
14 mask_err_id_vc2	Mask for err_id_vc2. **Default Value:** 0
13 mask_err_id_vc1	Mask for err_id_vc1. **Default Value:** 0
12 mask_err_id_vc0	Mask for err_id_vc0. **Default Value:** 0
11 mask_vc3_err_ecc_corrected	Mask for vc3_err_ecc_corrected. **Default Value:** 0
10 mask_vc2_err_ecc_corrected	Mask for vc2_err_ecc_corrected. **Default Value:** 0
9 mask_vc1_err_ecc_corrected	Mask for vc1_err_ecc_corrected. **Default Value:** 0
8 mask_vc0_err_ecc_corrected	Mask for vc0_err_ecc_corrected. **Default Value:** 0
7–6 -	This field is reserved.
5 mask_phy_errsoths_1	Mask for phy_errsoths_1. **Default Value:** 0
4 mask_phy_errsoths_0	Mask for phy_errsoths_0. **Default Value:** 0
3–2 -	This field is reserved.
1 mask_phy_erresc_1	Mask for phy_erresc_1. **Default Value:** 0
0 mask_phy_erresc_0	Mask for phy_erresc_0. **Default Value:** 0

41.6.11 D-PHY Test interface control 0 (MIPI_CSI_PHY_TST_CRTL0)

Address: 21D_C000h base + 30h offset = 21D_C030h

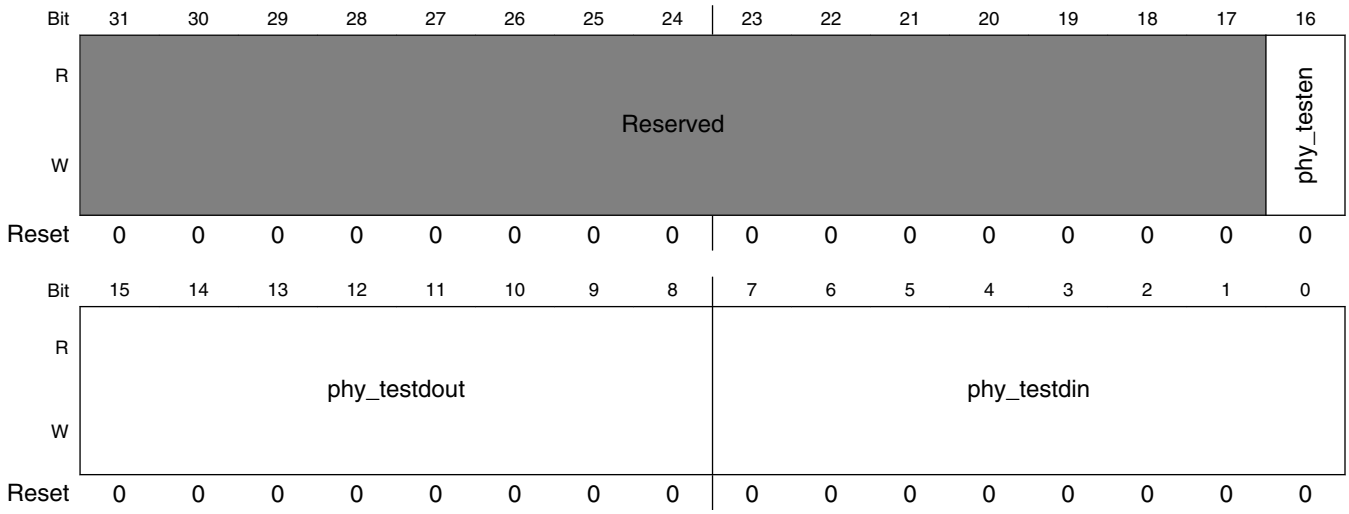


MIPI_CSI_PHY_TST_CRTL0 field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
1 phy_testclk	PHY test interface strobe signal. Used to clock TESTDIN bus into the D-PHY. In conjunction with TESTEN signal controls the operation selection. **Default Value:** 0
0 phy_testclr	PHY test interface clear. Used when active performs vendor specific interface initialization(Active High). **Default Value:** 0

41.6.12 D-PHY Test interface control 1 (MIPI_CSI_PHY_TST_CTRL1)

Address: 21D_C000h base + 34h offset = 21D_C034h



MIPI_CSI_PHY_TST_CTRL1 field descriptions

Field	Description
31–17 -	This field is reserved. Reserved
16 phy_testen	PHY test interface operation selector: 1 configures address write operation on the falling edge of TESTCLK 0 configures a data write operation on the rising edge of TESTCLK
15–8 phy_testdout	PHY output 8-bit data bus for read-back and internal probing functionalities. **Default Value:** 0
phy_testdin	PHY test interface input 8-bit data bus for internal register programming and test functionalities access **Default Value:** 0

Chapter 42
MIPI DSI Host Controller (MIPI_DSI)

42.1 Overview

The DSI Host Controller is a digital core that implements all protocol functions defined in the MIPI DSI Specification, providing an interface between the System and the MIPI D-PHY, and allowing communication with a MIPI DSI-compliant Display.

42.2 Features

The MIPI DSI Host Controller supports the following features:

Compliant with MIPI Alliance Specification for Display Serial Interface (DSI), Version 1.01.00 - 21 February 2008;

- Fully Compliant with MIPI Alliance Standard for Display Pixel Interface (DPI-2), Version 2.00 15 September 2005 with Pixel Data bus width up to 24bits
- Compliant with MIPI Alliance Standard for Display Bus Interface (DBI-2) Version 2.00 - 29 November 2005. Supported DBI types are:
  - Type B
  - 16bit, 9bit and 8bit Data bus width
- DBI and DPI interface can coexist but only one is operational
- Support all commands defined in MIPI Alliance Specification for Display Command Set (DCS), Version 1.02.00 - 23 July 2009

Interface with MIPI D-PHY following PHY Protocol Interface (PPI), as defined in MIPI Alliance Specification for D-PHY, Version 1.00.00 - 14 May 2009;

Supports up to 2 D-PHY Data Lanes:

- Bidirectional Communication and Escape Mode Support through Data Lane 0.
- Programmable display resolutions, from 160x120(QQVGA) to 1280x720(XVGA).
- Multiple Peripheral Support capability, configurable Virtual Channels.

- Video Mode Pixel Formats, 16bpp(RGB565), 18 bpp(RGB666) packed, 18 bpp(RGB666) loosely, 24 bpp(RGB888).
- Supports the transmission of all generic commands;
- ECC and Checksum capabilities;
- End of Transmission Packet (EoTp) support;
- Supports ultra low power mode
- Schemes for fault recovery.

42.2.1 System Overview

The DSI Host Controller is a digital core that implements all protocol functions defined in the MIPI DSI Specification, providing an interface between the System and the MIPI D-PHY, allowing the communication with a MIPI DSI compliant Display.

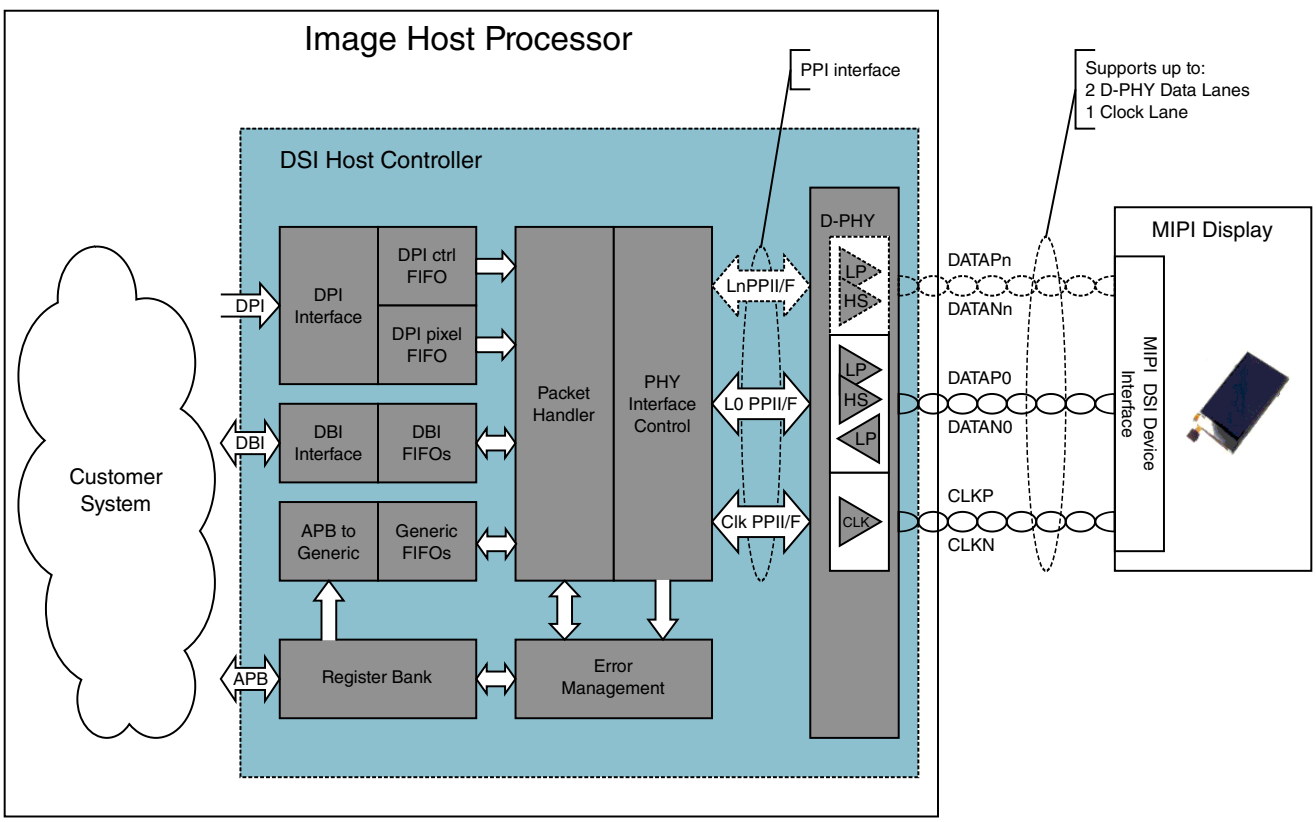


Figure 42-1. Block Diagram.

42.3 Clocks

The table found here describes the clock sources for MIPI.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 42-1. MIPI Clocks

Clock name	Clock Root	Description
ac_clk_125m	ahb_clk_root	Bus clock
pixel_clk	ipu1_ipu_hsp_clk_root	Pixel clock
cfg_clk	video_27m_clk_root	Configuration clock
ips_clk	ipg_clk_root	Peripheral clock
ips_clk_s	ipg_clk_root	Peripheral access clock
pll_refclk	video_27m_clk_root	Video reference clock (27MHz)

42.4 Architecture

This section describes the DSI Host Controller block interfaces, protocols, functionality, and implementation.

42.4.1 Architecture Overview

The following figure presents the overall architecture of the DSI Host Controller. The main blocks are the following:

DPI interface captures data and control signals from the DPI interface and conveys them in a FIFO for video control signals and another one for the pixel data.

DBI interface encapsulates DCS commands in DSI packets that are then conveyed into command and payload FIFOs. For commands that require a response from the device, the block uses an incoming data FIFO to acquire data from peripheral.

Register Bank is accessible through a standard AMBA-APB slave interface, providing access to the DSI Host Controller registers for configuration and control. There is also a fully programmable interrupt generator to inform the system upon certain events.

PHY Interface Control is responsible for managing the D-PHY PPI interface. It acknowledges current operation and enables LP transmission/reception or a High Speed transmission. It also performs data splitting between available D-PHY Lanes for High Speed transmission.

Packet Handler schedules activity inside the link. It performs several functions based on the interfaces that are currently operational (DPI and/or DBI) and the video transmission mode that is used (burst mode or non-burst mode with sync pulses or sync events). It

builds long or short packet generating correspondent ECC and CRC codes. This block also handles packet reception: Validates packet header by checking ECC, performs header correction for single bit erros and notifies and abort reception for multiple header errors. Depending on the packet type, generic read response or DCS command, route the output data to the respective port (generic or DBI)

APB to Generic block bridges APB operations into FIFOs holding the Generic commands. The block interfaces with 3 FIFOS, a command FIFO, a write payload FIFO and a read payload FIFO.

Error Management notifies and monitors error conditions on the DSI link. It controls timers used to determine if a time out condition occurred and triggers interruption for errors.

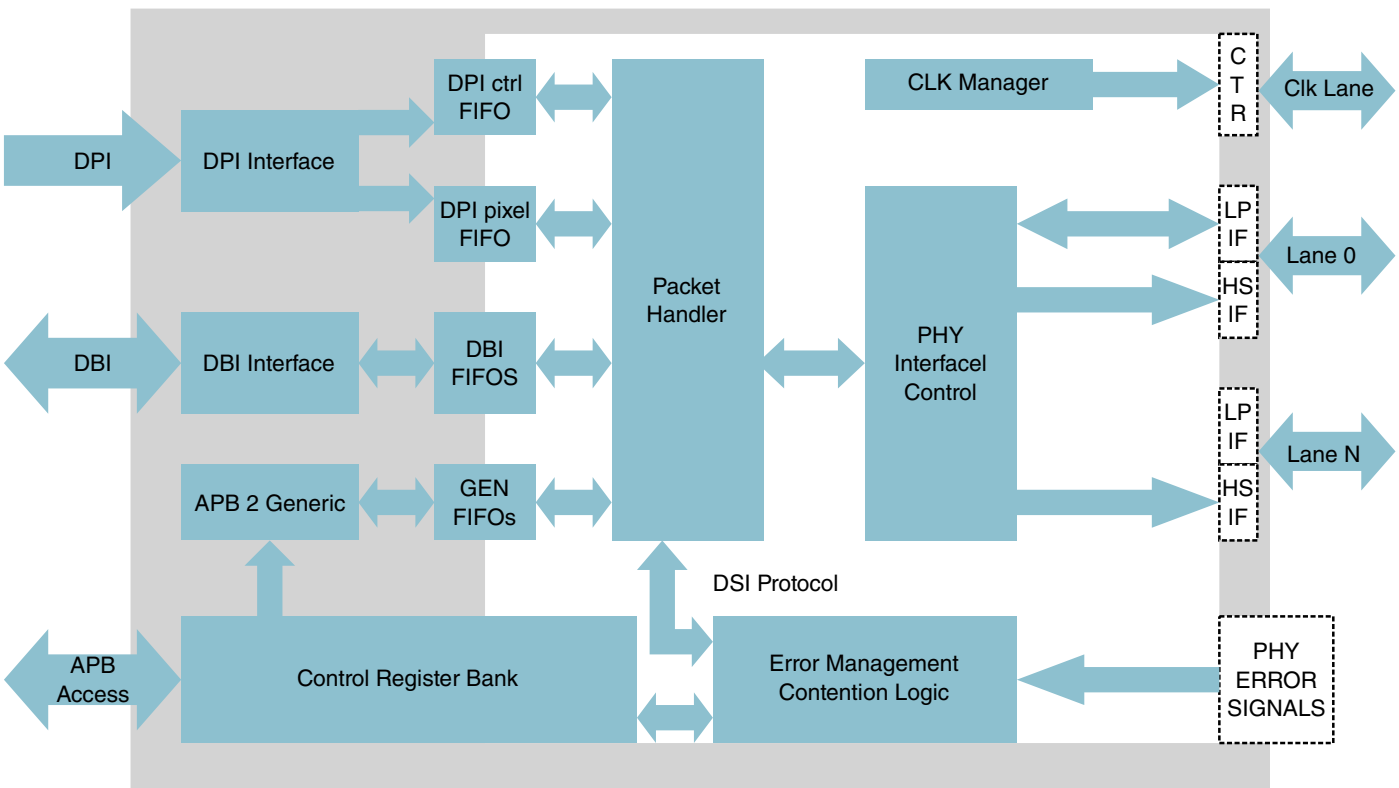


Figure 42-2. DSI Host Controller Architecture

42.4.2 DBI-2 interface

The DBI-2 interface encapsulates DCS commands in DSI packets to be transmitted through the D-PHY link. Some commands require a response from the device and the interface provides read data from the device/peripheral.

The support DBI types is Type B interface

The selection type maps the related pins on the DSI core pinout. DBI interface used RGB additive color mixing method and the pixel data format is selected on register DBI_CFG register. The field **in_dbi_conf** defines the input pixel data format and field **out_dbi_conf** defines the output pixel data that is determined by the display device. This interface needs to be associated with a particular DSI Virtual Channel that is programmed on field **dbi_vid**.

The following figures show how the pixel to byte conversion is done for each mode. All the diagrams are representing the case were the interface is working as type B.

The DBI interface type B signals list .

DBICSX: Chip select, it is low active .

DBIRDX: Read signal, the read data should be captured at the DBIRDX rising edge .

DBIWRX: Write signal, the write data will be switch at DBIWRX falling edge.

DBIDATIN[15:0]: Write data

DBIDCX: Data commnad select, Data is indicated when high and command is indicated when low .

DBITE: Tearing effect.

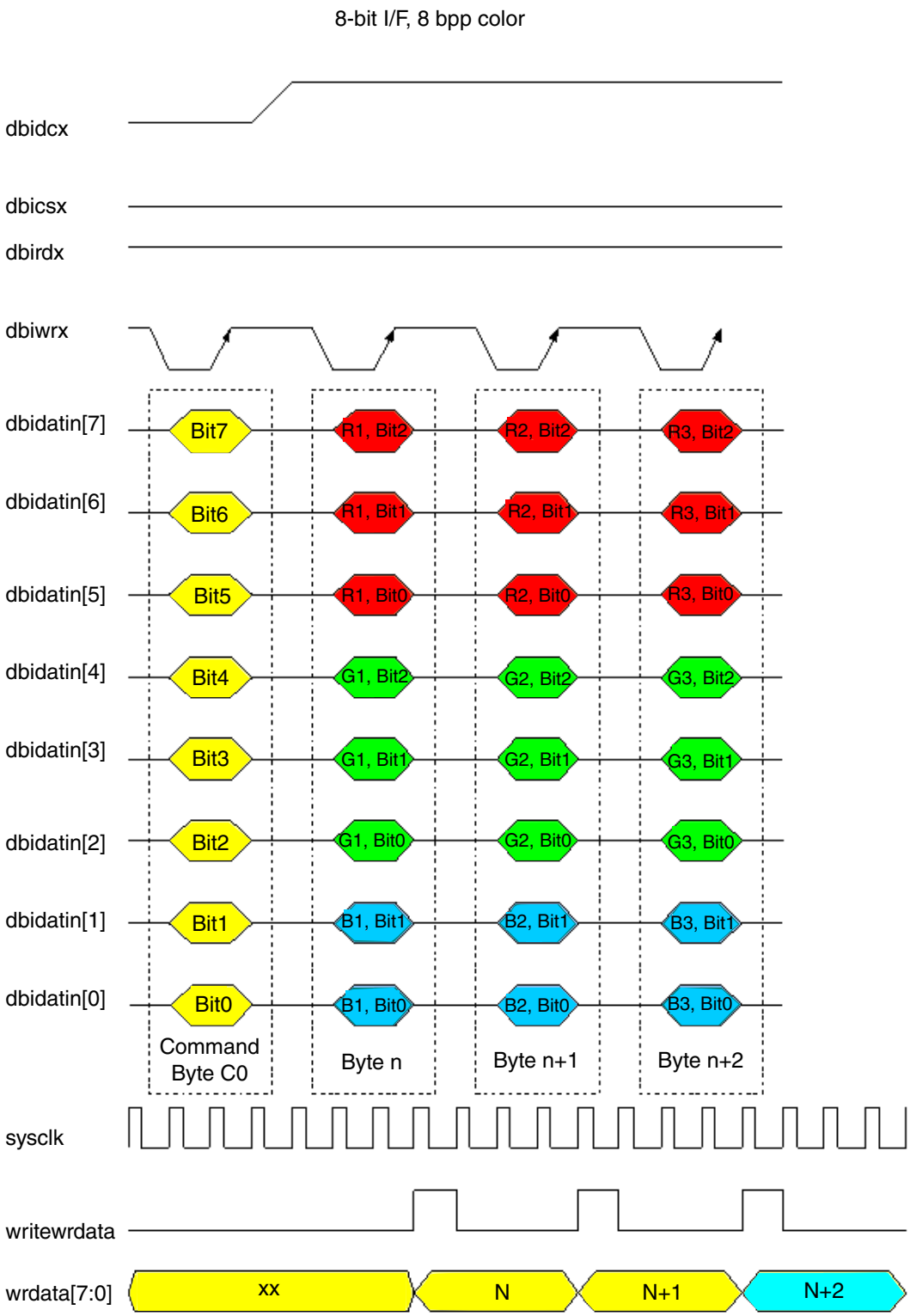


Figure 42-3. DSI 8 bit /8bpp byte write

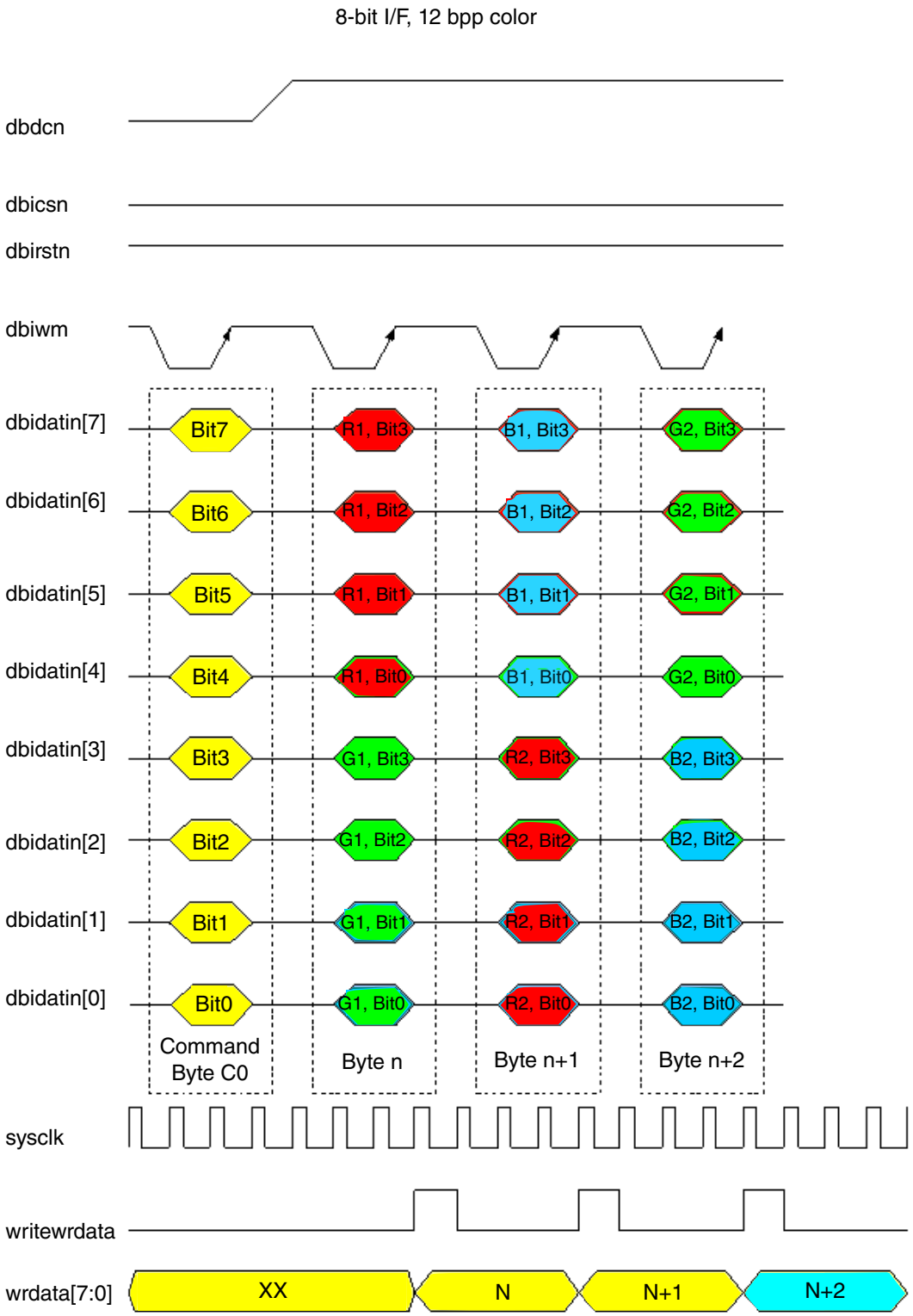


Figure 42-4. DSI 8 bit/12bpp byte write

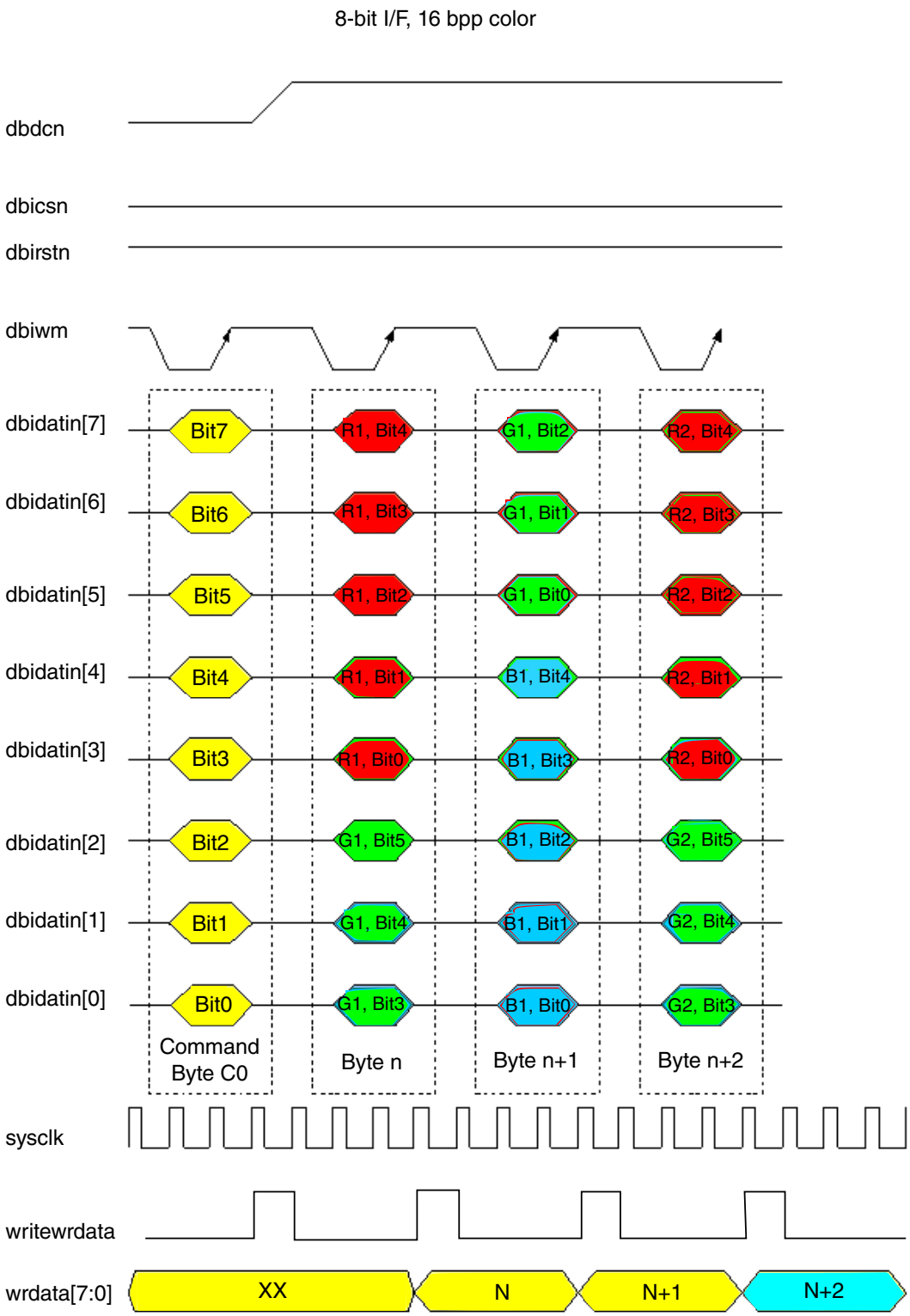


Figure 42-5. DSI 8 bit/16bpp byte write

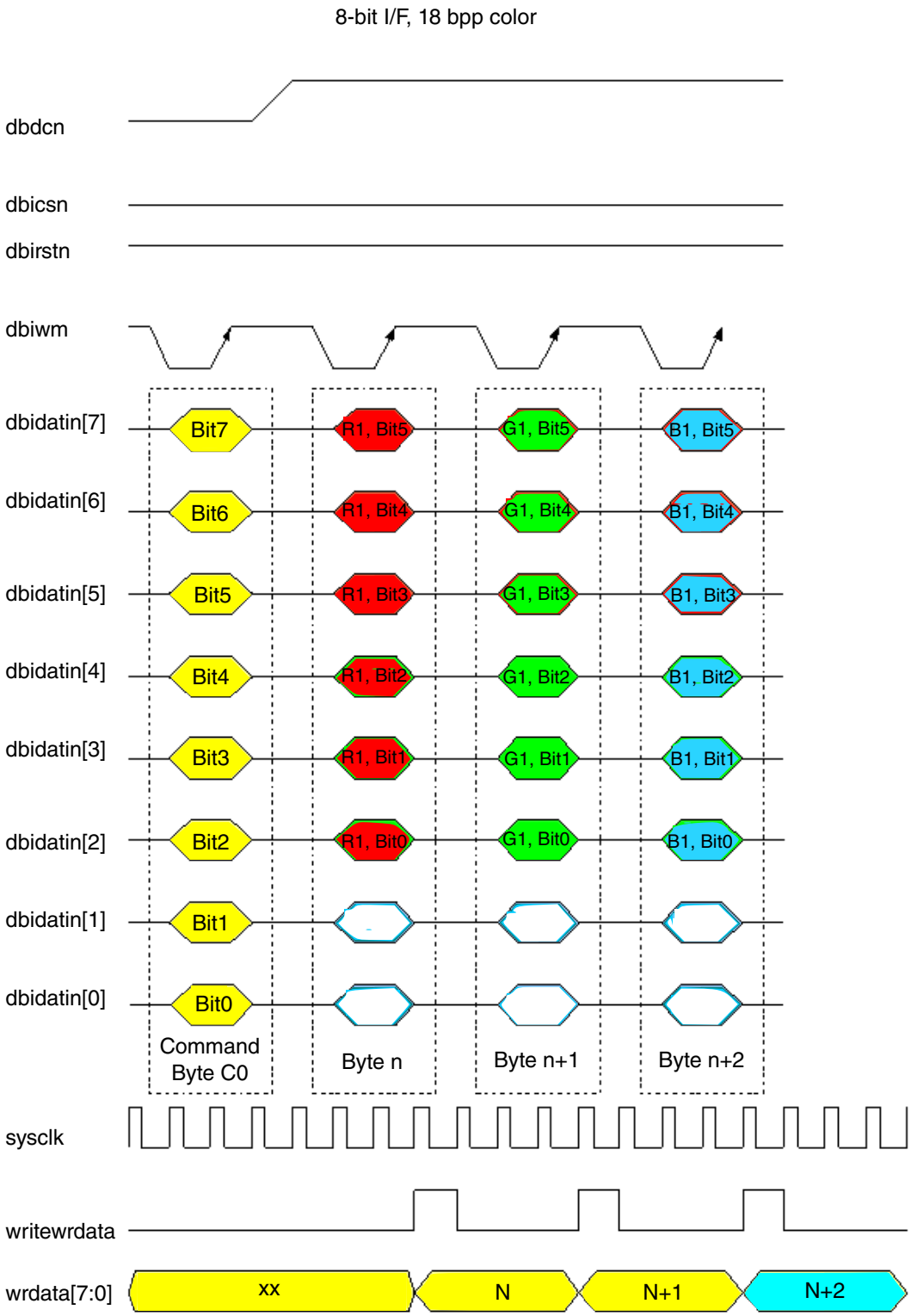


Figure 42-6. DSI 8 bit/18bpp byte write

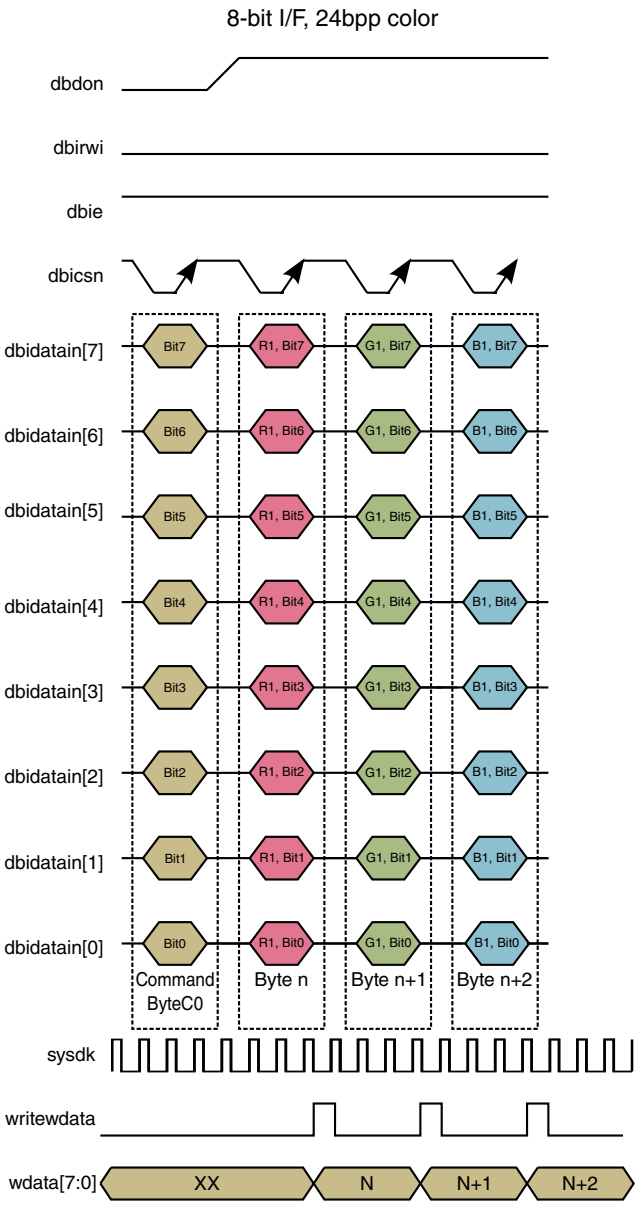


Figure 42-7. DSI 8 bit/24bpp byte write

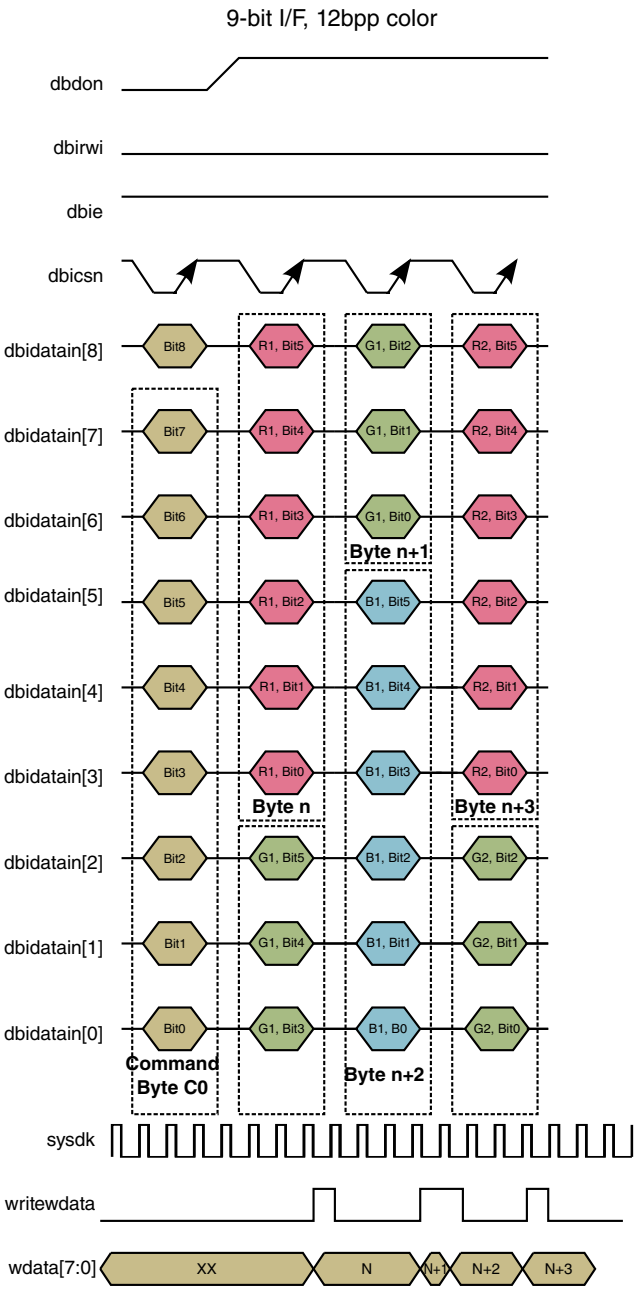


Figure 42-8. DSI 9 bit/12 byte write

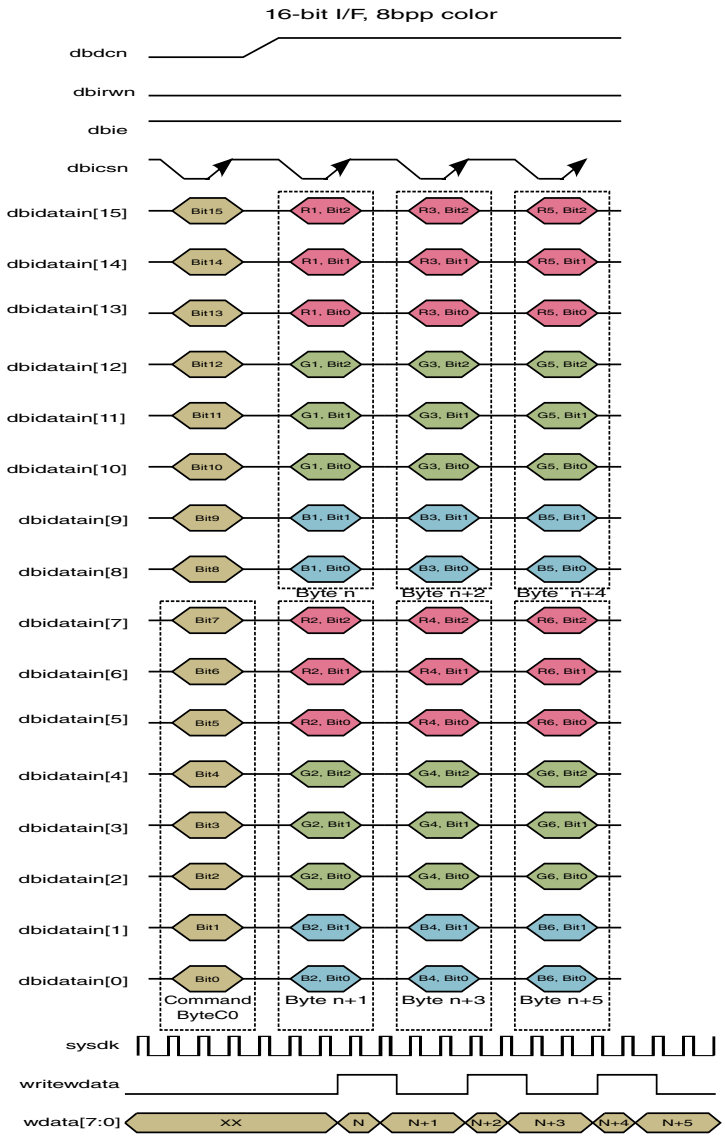


Figure 42-9. DSI 16 bit/8bpp byte write

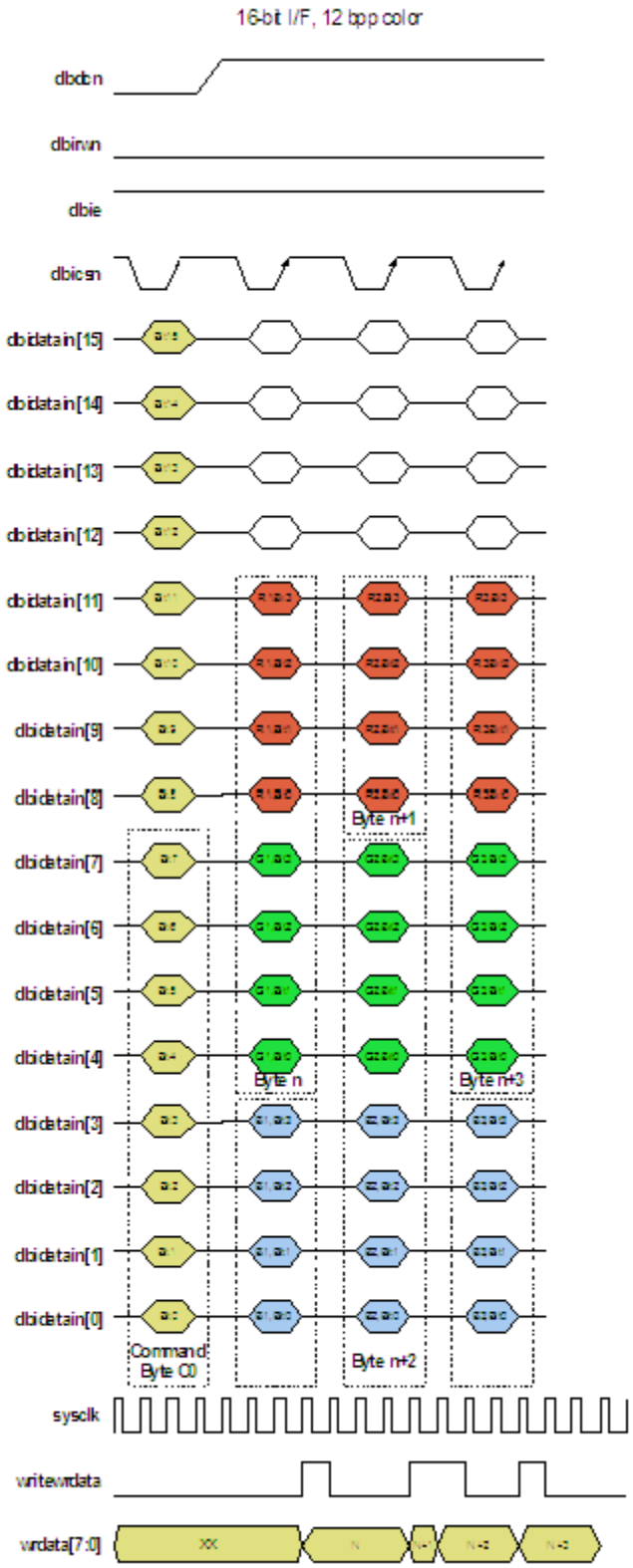


Figure 42-10. DSI 16 bit/12bpp byte write

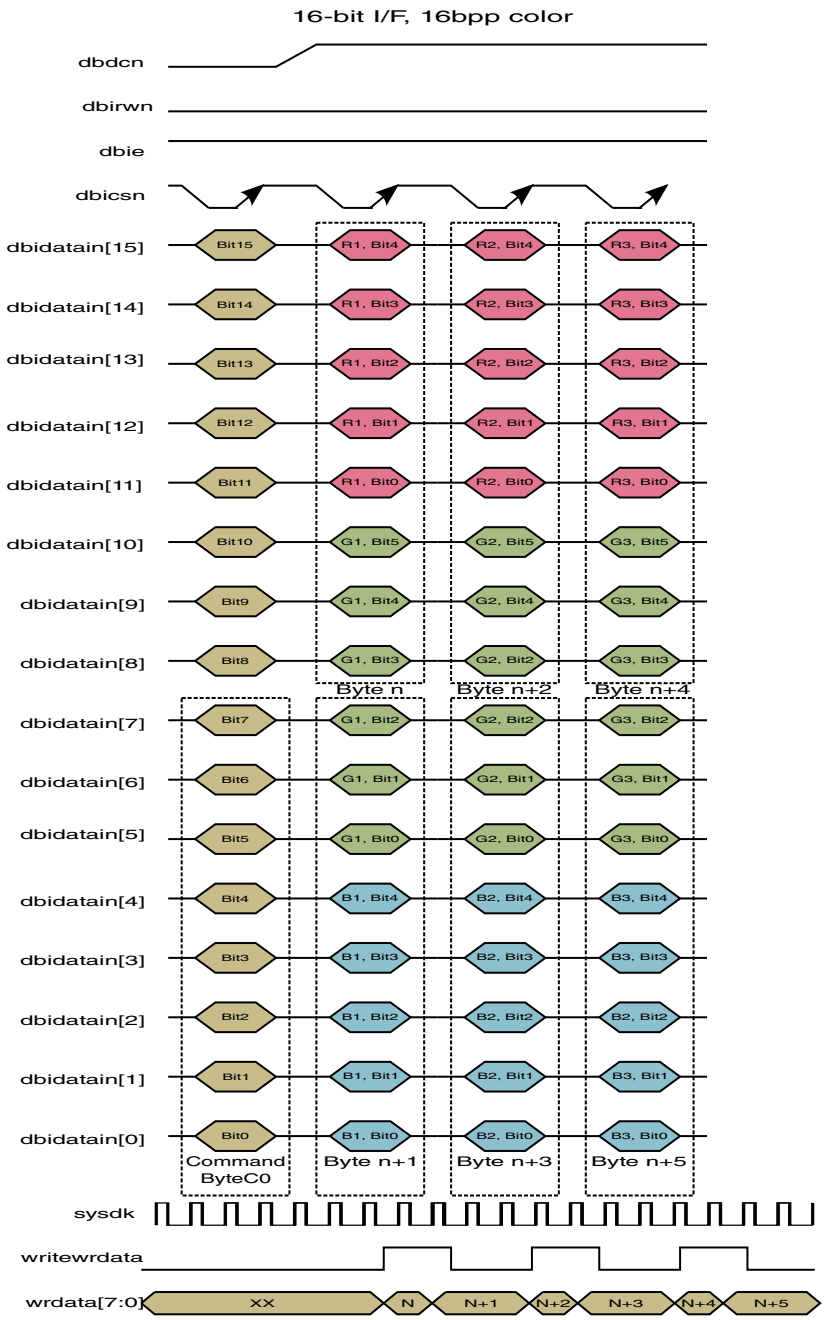


Figure 42-11. DSI 16 bit/16bpp byte write

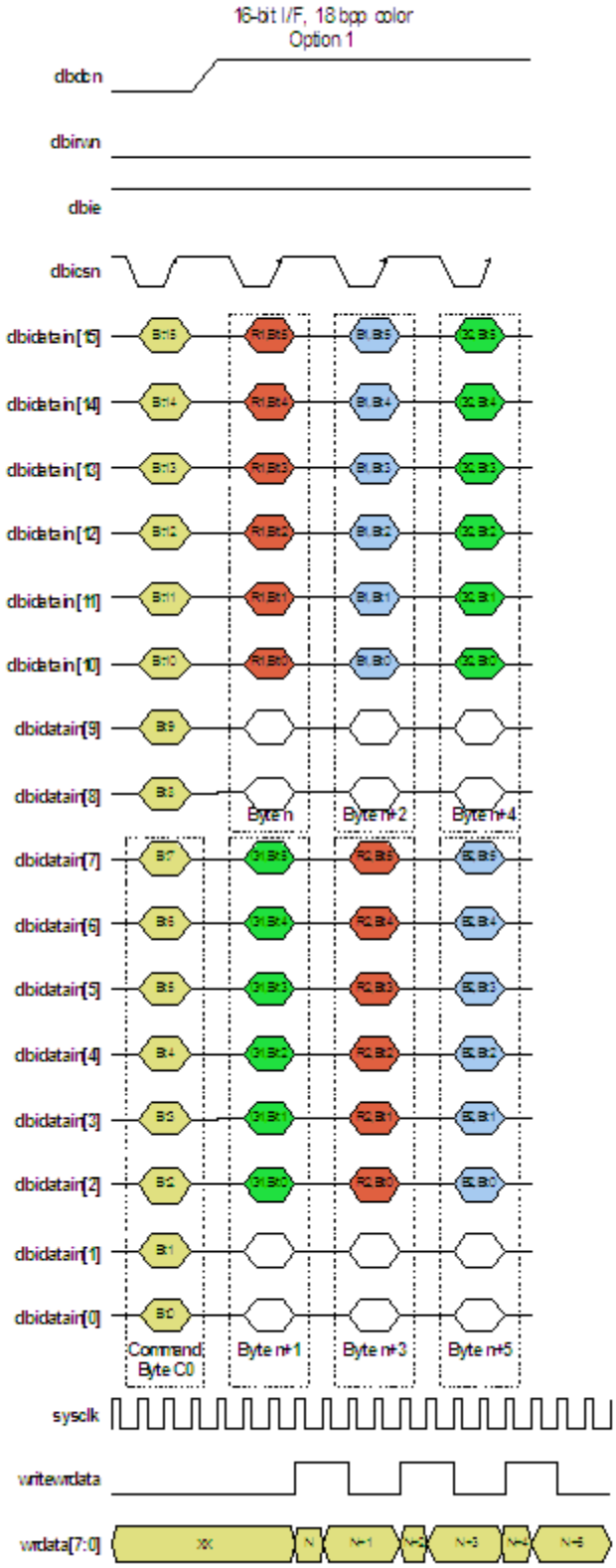


Figure 42-12. DSI 16 bit/18bpp option 1 byte write

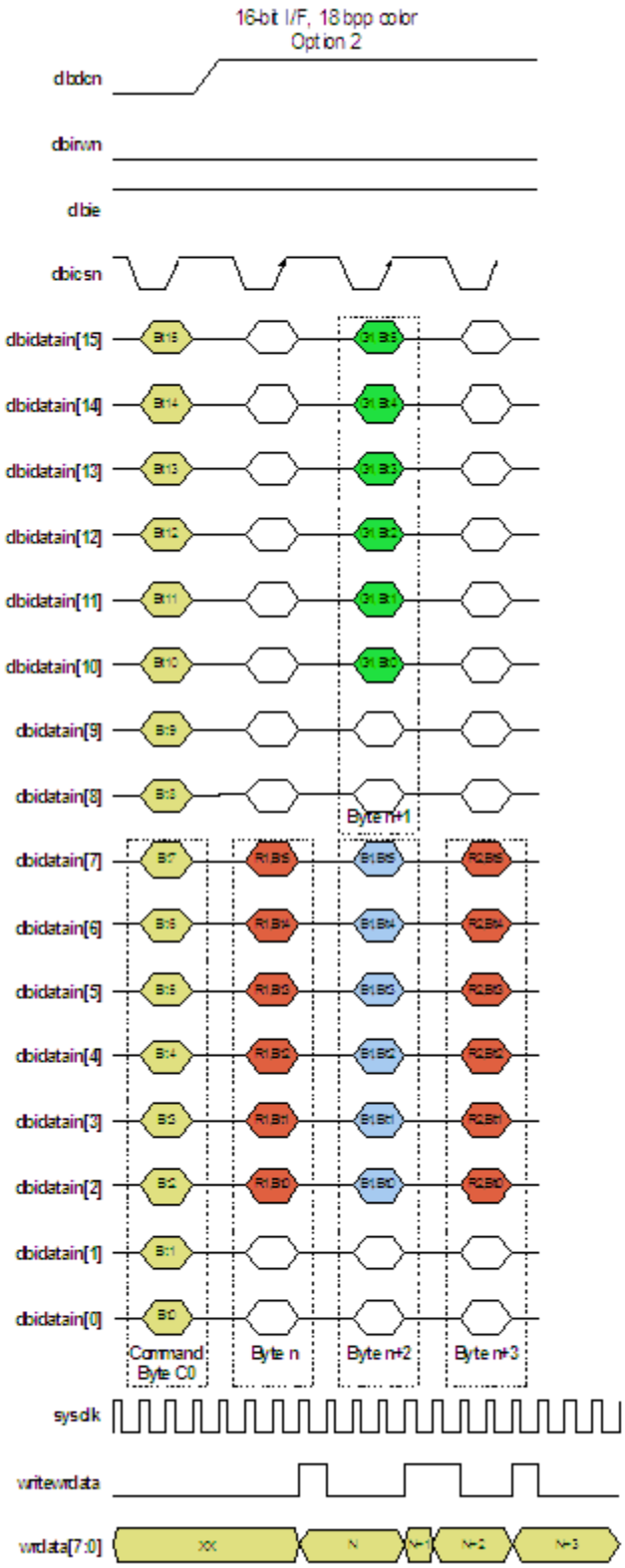


Figure 42-13. DSI 16 bit/18bpp option 2 byte write

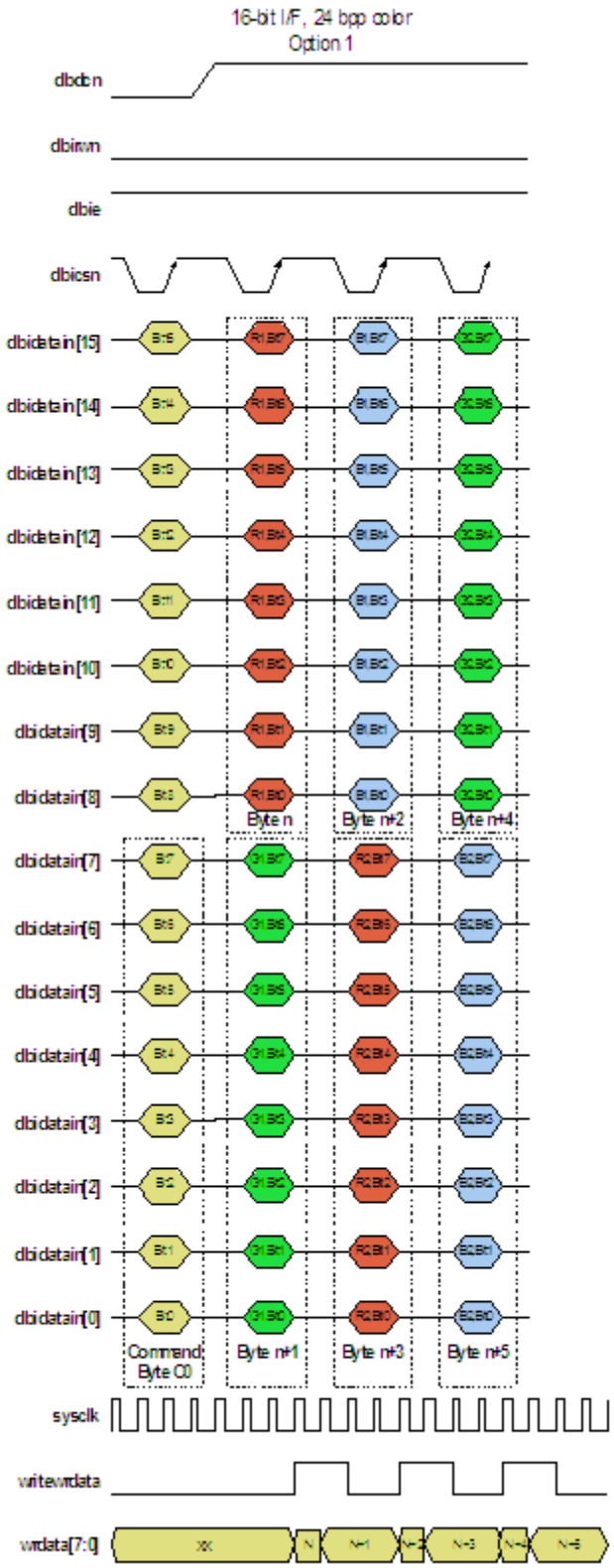


Figure 42-14. DSI 16 bit/24bpp option 1 byte write

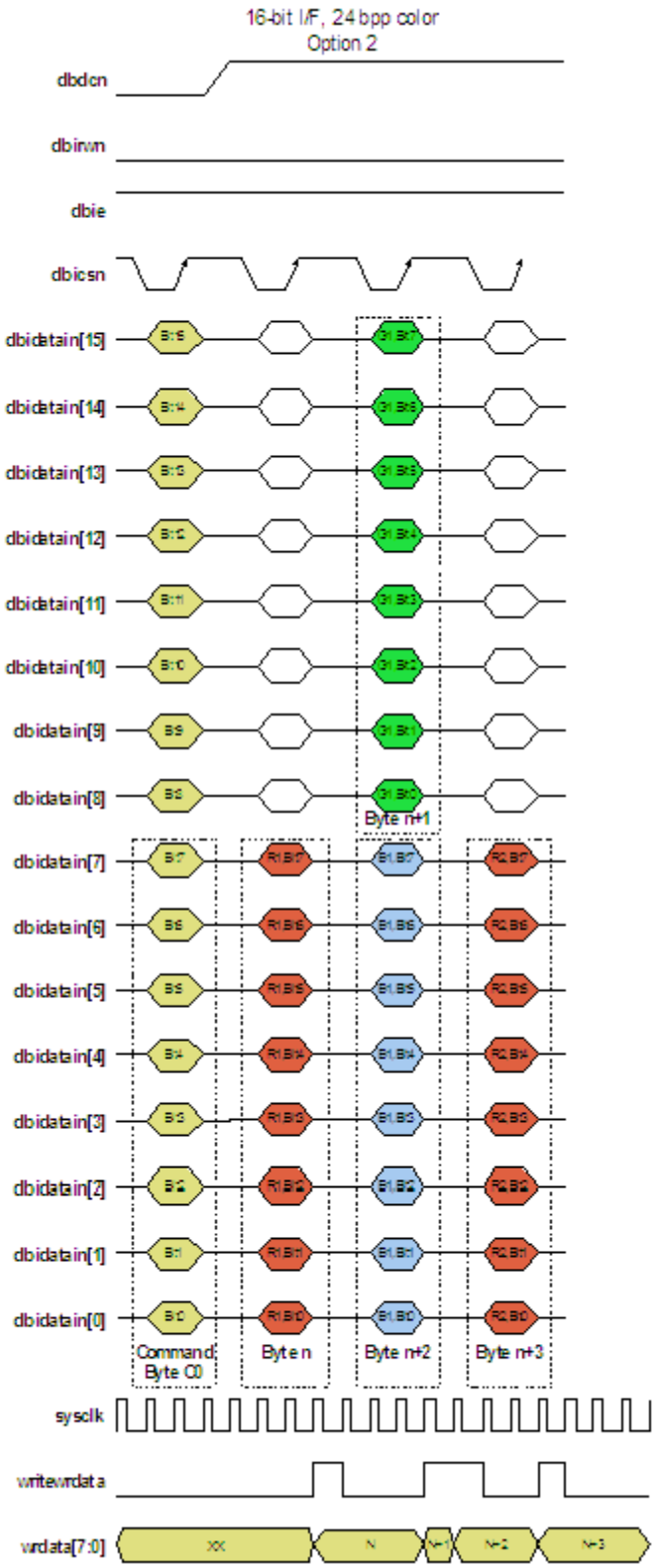


Figure 42-15. DSI 16 bit/24bpp option 2 byte write

DBI interface receives commands and decodes information for the command packetizing format. Depending on the command type, DSI core output the related DSI packet type (DCS short write packet; DCS read packet or DCS long write). If the DCS commands have variable size this needs to be configured on the DBI_CMDSIZE register. The following table presents DCS packet types:

Table 42-2. DSI packet types for DCS commands

DCS command	Command ID	DCS parameter	DSI packet type
enter_idle_mode	0x39h	0	6'h05
enter_invert_mode	0x21h	0	6'h05
enter_normal_mode	0x13h	0	6'h05
enter_partial_mode	0x12h	0	6'h05
enter_sleep_mode	0x10h	0	6'h05
exit_idle_mode	0x38h	0	6'h05
exit_invert_mode	0x20h	0	6'h05
exit_sleep_mode	0x11h	0	6'h05
nop	0x00h	0	6'h05
set_display_off	0x28h	0	6'h05
set_display_on	0x29h	0	6'h05
set_tear_off	0x34h	0	6'h05
soft_reset	0x01h	0	6'h05
set_address_mode	0x36h	1	6'h15
set_gamma_curve	0x26h	1	6'h15
set_pixel_format	0x3Ah	1	6'h15
set_tear_on	0x35h	1	6'h15
set_scroll_start	0x37h	2	6'h39
set_tear_scan_line	0x44h	2	6'h39
set_column_address	0x2Ah	4	6'h39
set_page_address	0x2Bh	4	6'h39
set_partial_area	0x30h	4	6'h39
set_scroll_area	0x33h	6	6'h39
write_LUT	0x2Dh	N	6'h39
write_memory_continue	0x3Ch	N	6'h39
write_memory_start	0x2Ch	N	6'h39
get_address_mode	0x0Bh	1(read)	6'h06
get_blue_channel	0x08h	1(read)	6'h06
get_diagnostic_result	0x0Fh	1(read)	6'h06
get_display_mode	0x0Dh	1(read)	6'h06
get_green_channel	0x07h	1(read)	6'h06
get_pixel_format	0x0Ch	1(read)	6'h06
get_power_mode	0x0Ah	1(read)	6'h06
get_red_channel	0x06h	1(read)	6'h06

Table continues on the next page...

Table 42-2. DSI packet types for DCS commands (continued)

DCS command	Command ID	DCS parameter	DSI packet type
get_scan_line	0x45h	2(read)	6'h06
get_signal_mode	0x0Eh	1(read)	6'h06
read_DDB_continue	0xA8h	n(read)	6'h06
read_DDB_start	0xA1h	n(read)	6'h06
read_memory_continue	0x3Eh	n(read)	6'h06
read_memory_start	0x2Eh	n(read)	6'h06

All packets that have 1 or 0 parameters index to the command are considered short packet and all other shall be considered long packets.

There are several register fields to configure DBI input pixel data and DBI pixel data retrieval. It is required to have prior knowledge of display supported feature to avoid incorrectly core configuration. Some guidelines are presented below to clarify how core uses the configuration for its operation.

Field **in_dbi_conf** in register DBI_CFG configures the color depth and the pixel format used by the input interface. Field **out_dbi_conf** configures the colour depth and the pixels format used by the output interface. Here the colour depth is directly related with formats supported by the peripheral. If the peripheral returns pixel data with a specific colour depth this colour depth should be consistent with the register configuration. If this is not respected, the byte to pixel converter will use an incorrect pixel mapping and it will corrupt output data.

Field **out_dbi_conf** should not be programmed for a higher number of interface pins than **in_dbi_conf**. As an example an incorrect pin configuration happens if **in_dbi_conf** defines 9 bit interface and **out_dbi_conf** defines 16 bit since pins 16 to 10 will not be available.

Field **lut_size_conf** configures the size of the DCS write_LUT command; this command will program the peripheral's colour space conversion table. This register is directly related with the capabilities of the peripheral. If the peripheral supports a colour depth of 16/18/24 bits per pixel it is expected that this register is configured for 48/128/192 bytes respectively. These coefficients should be loaded to the peripheral before normal operation of pixel data transmission is started. The DCS specification only refers to the conversion of formats equal or above 12bpp. This means that any format outside this scope might not be convertible to a higher colour depth mode by the peripheral. Taking this in consideration, when **in_dbi_conf** selects a colour depth below 12bpp, the **out_dbi_conf** should be configured such that colour depth has not a higher value. As an example, if **in_dbi_conf** is configured for 8bits/8bpp interface and **out_dbi_conf** is configured for 8bits/12bpp, this results in an incorrect configuration because the DCS

specification does not contemplates conversions from 8bpp to 12bpp. This configuration is possible inside the core but might lead into incorrect results. Nevertheless it is left for the system to configure these registers according with the desired operation.

For Read back commands the DBI specification states that the reading process does not use any acknowledge mechanism to know that the requested data is ready. There are mechanisms for the system to get this information by polling the Command Status Register CMD_PKT_STATUS. This register, besides other information, contains the status on the DBI read back payload FIFO and also contains a special busy flag **dbi_rd_cmd_busy** that can be used to understand if the read command is complete. By using the DBI read back FIFO status bits that report if the FIFO is full or empty the processor can read these bits to understand if it can read another byte/sample from the interface (probing the empty flag) or when it should stop reading bytes/samples from the interface. In addition, the Read Command Busy flag is asserted when a read command is issued by the interface and cleared once the reading process ends and the read data is fully stored inside the read back FIFO. The system needs to issue the read command and then periodically probe this status bit to acknowledge if the read back operation is complete. Once this information is available the system can assume that the entire data is stored inside the read back FIFO and the interface can deliver all the desired data.

42.4.3 DPI-2 Interface

The DPI-2 interface captures data and control signals and conveys them to FIFO interfaces that will transmit them to the DSI link. Two different streams of data are presented at the interface: video control signals and pixel data. Depending on the interface color coding, the pixel data is handled differently throughout the dpipixdata bus.

Interface pixel color coding is presented in the table below.

Table 42-3. DPI interface pixel color coding mapping.

Signal Line	16-bit			18-bit		24-bit
	Config1	Config2	Config3	Config1	Config2	
D23	Not used	Not used	Not used	Not used	Not used	R7
D22	Not used	Not used	Not used	Not used	Not used	R6
D21	Not used	Not used	R4	Not used	R5	R5
D20	Not used	R4	R3	Not used	R4	R4
D19	Not used	R3	R2	Not used	R3	R3
D18	Not used	R2	R1	Not used	R2	R2
D17	Not used	R1	R0	R5	R1	R1
D16	Not used	R0	Not used	R4	R0	R0
D15	R4	Not used	Not used	R3	Not used	G7

Table continues on the next page...

Table 42-3. DPI interface pixel color coding mapping. (continued)

Signal Line	16-bit			18-bit		24-bit
	Config1	Config2	Config3	Config1	Config2	
D14	R3	Not used	Not used	R2	Not used	G6
D13	R2	G5	G5	R1	G5	G5
D12	R1	G4	G4	R0	G4	G4
D11	R0	G3	G3	G5	G3	G3
D10	G5	G2	G2	G4	G2	G2
D9	G4	G1	G1	G3	G1	G1
D8	G3	G0	G0	G2	G0	G0
D7	G2	Not used	Not used	G1	Not used	B7
D6	G1	Not used	Not used	G0	Not used	B6
D5	G0	Not used	B4	B5	B5	B5
D4	B4	B4	B3	B4	B4	B4
D3	B3	B3	B2	B3	B3	B3
D2	B2	B2	B1	B2	B2	B2
D1	B1	B1	B0	B1	B1	B1
D0	B0	B0	Not used	B0	B0	B0

The DPI interface can be configured to increase flexibility and promote correct usage of this interface for several systems. These configuration options are described below:

- Polarity control. All the control signals are programmable to change polarity according to system requirements.
- After core reset, DPI waits for the first VSYNC active transition to start signals sampling, including pixel data and preventing image transmission in the middle of a frame.
- If interface pixel color coding is 18 bits and the 18 bit loosely packet stream is enabled, the number of lines programmed in the pixels per lines configuration is a multiple of four. This means that in this mode the 2 LSB in the configuration will be always inferred as zero. Specification states that in this mode the pixel line size should be a multiple of 4.
- To avoid pixel underflows and overflows the configured number of pixel will originate the respective memory spaces. This will happen even if the dpidataen pin is active during more or less time than necessary.
- In order to keep the memory organized in respect to the packet scheduling the number of pixels per packet parameter will be used to separate the memory space of different video packets.

For the **dpishutd** and **dpicolorm** sampling and transmission the DPI video signalling must be active. This means that if video signals like VSYNC and HSYNC are not being actively generated the DPI commands these are not transmitted through the DSI link. The

DSI commands packets have the highest priority in the class of Non Video packets. Because of such constraint and in order for commands to be correctly transmitted, the first VSYNC active pulse needs to happen for the commands sampling and transmission. When shutting down the display, it is necessary for DPI video to keep active for one frame after the command being issued. This guarantees that the commands are correctly transmitted before actually disabling the Video generation in the DPI interface.

42.4.4 Error control and timeout timers

The DSI host controller implements a set of timers and conditions to notify errors.

The core has a set of registers to control the timers that will be used to determine if a time out has occurred. It also contains a set of interruption status registers that are cleared upon read operation. These registers also trigger an interruption pin that can be used by the system to act upon an error within the DSI connection.

42.4.5 Timeout timers

Bit **to_hs_tx** on register ERROR_ST1 is set when a High Speed Transmission Time Out occurs rising an interrupt pin.

The time out is configured in **hstx_to_cnt** field of register TO_CNT_CFG. A 16 bit counter measures the time when High Speed is active. If that counter reaches the value defined by register **hstx_to_cnt**, bit **to_hs_tx** is asserted.

Bit to_lp_rx on register ERROR_ST1 is set signaling contention detection when a Low Power Reception Time Out occurs rising an interrupt pin. The time out is configured in **lprx_to_cnt** field of register TO_CNT_CFG. A 16 bit counter measures the time when Low Power reception is active. If that counter reaches the value defined by register **lprx_to_cnt**,bit **to_lp_rx** is asserted.

Time units for this 16bit counters are configured in cycles defined in **TO_CLK_DIVISION** field in register CLKMGR_CFG. The value written to **TO_CLK_DIVISION** defines the time unit for the Time out limits using Lane byte clock as input. This mechanism increases the range to define these limits.

42.4.6 Error control

Two registers ERROR_ST0 and ERROR_ST1 are associated with error condition reporting. This resister can trigger interrupt pins to inform the system about the occurrence of errors.

The coreConsultant GUI allows different interrupt pin configuration that can support:

- No interrupt pin on DSI core for notification of error although system can check for error by reading error status registers ERROR_ST0 and ERROR_ST0.
- DSI core has one interrupt pin interrupt that is set High when an error occurs either in register ERROR_ST1 or ERROR_ST2.
- DSI core has 2 interrupts associated with each error status register. Pins interrupt0 and interrupt1 are respectively associated with ERROR_ST1 and ERROR_ST2. This mechanism allows faster association of interrupts with the status register that generated the error.

The triggering of the interrupt pins, when defined, can be masked by programming the mask registers ERROR_MSK0 and ERROR_MSK1. When any bit of this registers is set to 1 it will inhibit the interrupt for that specific error. Nevertheless the error bit will always be set on the respective ERROR_ST register. Registers ERROR_ST1 and ERROR_ST2 are always cleared when read.

The figure below illustrates the location of some of the errors.

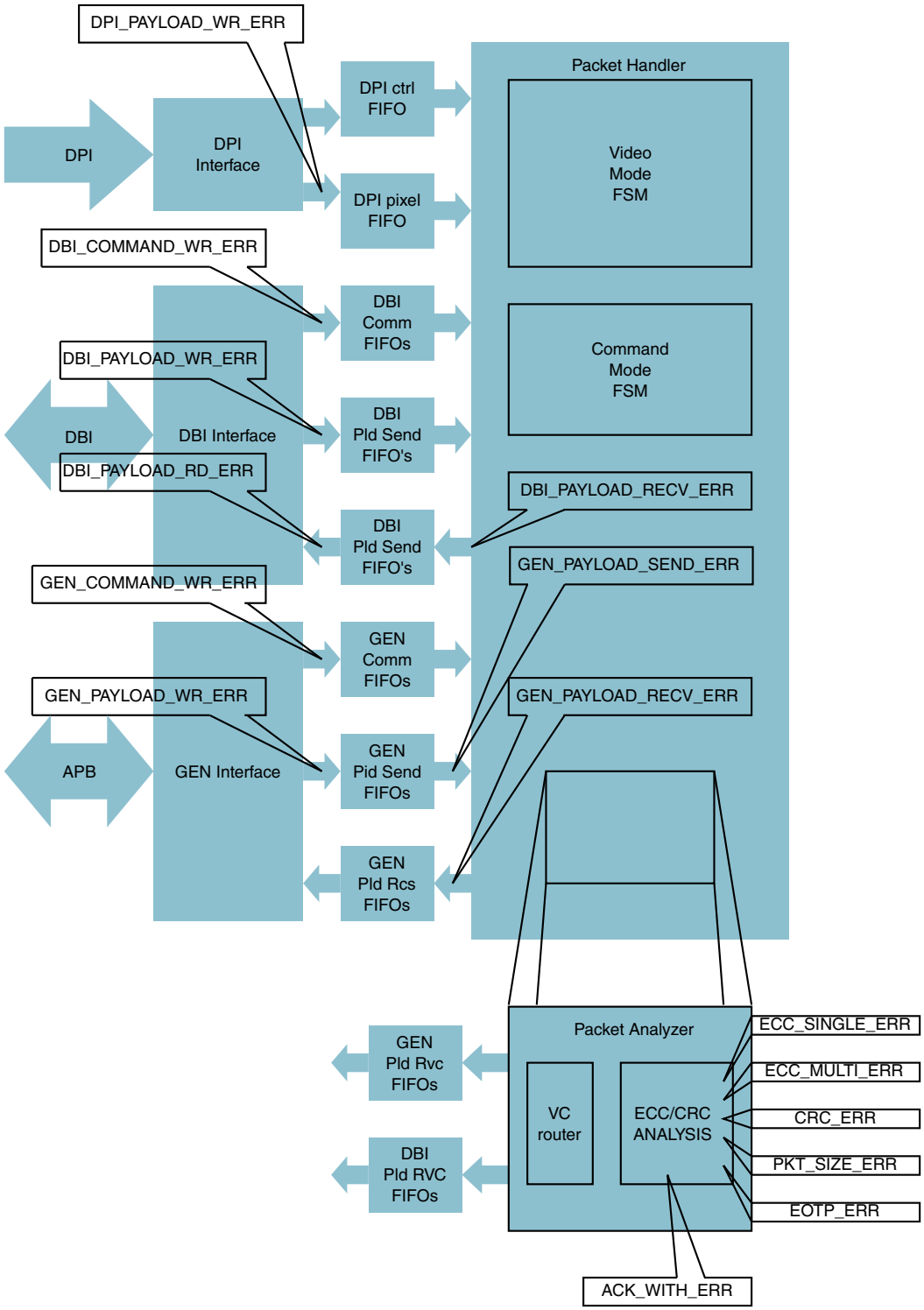


Figure 42-16. Error sources diagram

42.4.7 Generic packets

DSI host Core support generic packets transmission and reception as described in the DSI-2 specification. These packets are channeled through the APB register bank.

Register GEN_PLD_DATA is used to write payload of generic packets and when read it returns the payload of a read back operation. Register GEN_HDR contains the generic packet header type and header data. Writing to this register triggers the transmission of the packet implying that for long generic packets data needs to be written in advance on register GEN_PLD_DATA.

The valid packets available to be transmitted through the Generic interface are:

Table 42-4. Generic interface available packets

Generic Write Short Packet 0 Parameters	Generic Write Short Packet 1 Parameters	Generic Write Short Packet 2 Parameters	Generic Read Short Packet 0 Parameters
Generic Read Short Packet 1 Parameters	Generic Read Short Packet 2 Parameters	Maximum Read Packet Configuration	Generic Long Write Packet
DCS Write Short Packet 0 Parameters	DCS Write Short Packet 1 Parameters	DCS Read Short Packet 0 Parameters	DCS Write Long Packet

A set of bits in register CMD_PKT_STATUS report the status of the FIFOs associated with generic packet support.

Generic packets are always transported using one of the DSI transmission modes: Video Mode or Command Mode. If none of these modes are selected the packets will not be transmitted through the link and the related FIFOs will eventually get overflowed.

If system does not requires Generic packets support this functionality can be disabled in coreConsultant GUI and the interfaces with the 2-port RAMs that buffer command, payload and read data are removed.

42.4.8 Signals

A Clock Lane and a Data Lane (Lane0) with TX HS features are always required. Data Lanes can be up to a maximum of 2.

Table 42-5. D-PHY external interface

Pin Name	Width	Direction	Description
DPHY Supply			
AVDD	1	Input	D-PHY Analog power supply.
VDD	1	Input	D-PHY Digital power supply.
AVDDREF	1	Input	D-PHY Analog supply for reference generator.

Table continues on the next page...

Table 42-5. D-PHY external interface (continued)

Pin Name	Width	Direction	Description
AGND	1	Input	D-PHY Analog supply ground return.
VSS	1	Input	D-PHY Digital supply ground return
AGNDREF	1	Input	D-PHY Analog supply ground return for reference generator.
DPHY external pins			
REFCLK	1	Input	D-PHY Reference clock used for Master-side serial clock generation in Clock Multiplying Unit (PLL)
CFG_CLK	1	Input	D-PHY Configuration clock used for the initialization of the PHY. It is also used for exiting ULPS state.
REXT	1	Input	D-PHY External resistor connection
CLKP	1	Input	D-PHY Positive D-Phy differential clock line transceiver output.
CLKN	1	Input	D-PHY Negative D-Phy differential clock line transceiver output.
DATAP0	1	Input	D-PHY Positive D-Phy differential data line transceiver output, Lane 0
DATAN0	1	Input	D-PHY Negative D-Phy differential data line transceiver output, Lane 0
DATAP1	1	Input	D-PHY Positive D-Phy differential data line transceiver output, Lane 1
DATAN1	1	Input	D-PHY Negative D-Phy differential data line transceiver output, Lane 1

42.5 Programming

42.5.1 DSI and D-PHY initialization sequence

This chapter describes the procedure for DSI and D-PHY initialization. This process is based on APB register interface access.

- By default register PHY_RSTZ is activating the PHY resets **physhutdownz**, **phyrstz** and disabling **enableclk** and register PHY_TEST_CTRL0 is by default asserting the **testclr** pin. All the PHY reset pins are being activated by default.
- Configure Register PHY_IF_CFG with correct the number of lanes to be used by the controller.
- Configure the TX_ESC clock frequency to a frequency lower than 20MHz that is the maximum allowed frequency for D-PHY ESCAPE mode. This is done by writing in Register CLKMGR_CFG, field TX_ESC_CLK_DIVISION.
  TX_ESC_CLK_DIVISION divides Byte Clock and generates a TX_ESC clock for the D-PHY. (Note: Byte clock is limited to 125MHz (1GHz/8bits) and by writing TX_ESC_CLK_DIVISION=0x07 TX_ESC clock will always be lower than 20MHz)
- Configure the DPHY PLL clock frequency through the TEST Interface to operate at 1GHz, assuming that the REF_CLK is provided with a frequency of 27MHz

- Write @ PHY_TST_CTRL0 - 32'h00000000 this disables the **testclr** pin enabling the interface to write new values to the DPHY internal registers.
- Write @ PHY_TST_CTRL1 - 32'h00010044 this enables the **testen** pin bit 17 of this Core register and configures the **testdatain** to 8'h44. This operation initiate the configuration process of the test code number 0x44.
- Write @ PHY_TEST_CTRL0 - 32'h0000002 followed by a new write to PHY_TEST_CTRL0 - 32'h00000000. This operation toggles the **testclk** (bit 2) and the **testdin** will be sampled on the falling edge of **testclk** latching a new test code.
- Write @ PHY_TEST_CTRL1 - 32'h00000074 disabling the testen pin and configuring **testdatain** to 8'h74. This operation prepares the interface to load in test code 0x44 the 0x74 value.
- Write @ PHY_TEST_CTRL0 - 32'h00000002 followed by a new write to PHY_TEST_CTRL0 - 32'h00000000. This operation toggles the **testclk** and the **testdin** will be sampled on the rising edge of **testclk** latching a new content data to the configured test code.

- Write @ PHY_RSTZ - 32'h00000007. This operation asserts **physhutdownz**, **phyrstz** and **enableclk** releasing the PHY from power down. The PHY will startup the PLL locking procedure to 1GHz operation.
- Read @ PHY_STATUS - 32'hxxxxxxx1, until bit 0 **phylock** is detected at 1 signaling that PLL is locked and that a stable byte clock is being provided to the DSI host controller.
- Read @ PHY_STATUS - 32'hxxxxx1x1, until bit 2 **phystopstateclklane** is read '1' identifying that Clock Lane is in Stop State. Clock lane need to be in Stop state so that the D-PHY can switch to other operational states such as the High Speed mode.
- Write register PHY_IF_CTRL bit 0 to generate High Speed clock (**txrequestHSclk**).
- Only after: 1) PLL locked and 2) Clock lane in Stop-State; the PHY will drive the correct LP sequence to configure the receiver end for HS.
- D-PHY starts transmitting HS clock on the Clock Lane.

42.6 MIPI_DSI Memory Map/Register Definition

MIPI_DSI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21E_0000	Version of the DSI host ctrl (MIPI_DSI_VERSION)	32	R	0000_0000h	42.6.1/3702
21E_0004	Core power up (MIPI_DSI_PWR_UP)	32	R/W	0000_0000h	42.6.2/3702
21E_0008	Number of active data lanes (MIPI_DSI_CLKMGR_CFG)	32	R/W	0000_0000h	42.6.3/3703

Table continues on the next page...

MIPI_DSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21E_000C	DPI interface configuration (MIPI_DSI_DPI_CFG)	32	R/W	0000_0000h	42.6.4/3703
21E_0010	DBI interface configuration (MIPI_DSI_DBI_CFG)	32	R/W	0000_0000h	42.6.5/3705
21E_0014	DBI command size configuration (MIPI_DSI_DBIS_CMDSIZE)	32	R/W	0000_0000h	42.6.6/3706
21E_0018	Packet handler configuration (MIPI_DSI_PCKHDL_CFG)	32	R/W	0000_0000h	42.6.7/3707
21E_001C	Video Mode Configuration (MIPI_DSI_VID_MODE_CFG)	32	R/W	0000_0000h	42.6.8/3708
21E_0020	Video packet configuration (MIPI_DSI_VID_PKT_CFG)	32	R/W	0000_0000h	42.6.9/3709
21E_0024	Command mode configuration (MIPI_DSI_CMD_MODE_CFG)	32	R/W	0000_0000h	42.6.10/ 3710
21E_0028	Line timer configuration (MIPI_DSI_TMR_LINE_CFG)	32	R/W	0000_0000h	42.6.11/ 3712
21E_002C	Vertical timing configuration (MIPI_DSI_VTIMING_CFG)	32	R/W	0000_0000h	42.6.12/ 3712
21E_0030	D-PHY timing configuration (MIPI_DSI_PHY_TMR_CFG)	32	R/W	0000_0000h	42.6.13/ 3713
21E_0034	Generic packet Header configuration (MIPI_DSI_GEN_HDR)	32	R/W	0000_0000h	42.6.14/ 3713
21E_0038	Generic payload data in/out (MIPI_DSI_GEN_PLD_DATA)	32	R/W	0000_0000h	42.6.15/ 3714
21E_003C	Command packet status (MIPI_DSI_CMD_PKT_STATUS)	32	R/W	0000_0000h	42.6.16/ 3714
21E_0040	Time Out timers configuration (MIPI_DSI_TO_CNT_CFG0)	32	R/W	0000_0000h	42.6.17/ 3716
21E_0044	Interrupt status register 0 (MIPI_DSI_ERROR_ST0)	32	R/W	0000_0000h	42.6.18/ 3716
21E_0048	Interrupt status register 1 (MIPI_DSI_ERROR_ST1)	32	R/W	0000_0000h	42.6.19/ 3718
21E_004C	Masks Interrupt generation trigged by ERROR_ST0 register (MIPI_DSI_ERROR_MSK0)	32	R/W	0000_0000h	42.6.20/ 3719
21E_0050	Masks Interrupt generation trigged by ERROR_ST1 register (MIPI_DSI_ERROR_MSK1)	32	R/W	0000_0000h	42.6.21/ 3721
21E_0054	D-PHY reset control (MIPI_DSI_PHY_RSTZ)	32	R/W	0000_0000h	42.6.22/ 3723
21E_0058	D-PHY interface configuration (MIPI_DSI_PHY_IF_CFG_)	32	R/W	0000_0000h	42.6.23/ 3724
21E_005C	D-PHY PPI interface control (MIPI_DSI_PHY_IF_CTRL)	32	R/W	0000_0000h	42.6.24/ 3724
21E_0060	D-PHY PPI status interface (MIPI_DSI_PHY_STATUS)	32	R/W	0000_0000h	42.6.25/ 3725
21E_0064	D-PHY Test interface control 0 (MIPI_DSI_PHY_TST_CTRL0)	32	R/W	0000_0001h	42.6.26/ 3727
21E_0068	D-PHY Test interface control 1 (MIPI_DSI_PHY_TST_CTRL1)	32	R/W	0000_0000h	42.6.27/ 3727

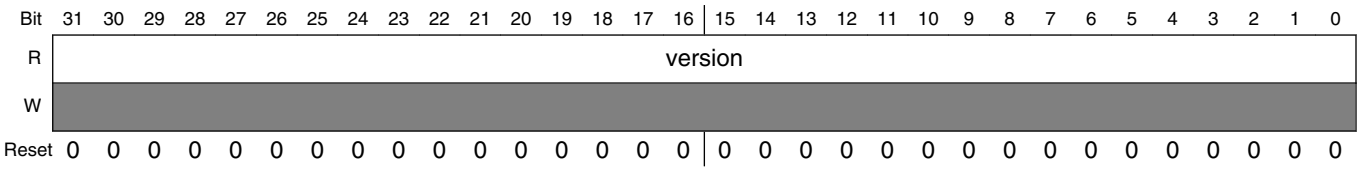
42.6.1 Version of the DSI host ctrl (MIPI_DSI_VERSION)

Size: 32 bits

Offset: 0x0

Memory Access: R

Address: 21E_0000h base + 0h offset = 21E_0000h



MIPI_DSI_VERSION field descriptions

Field	Description
version	Version of the DSI host controller

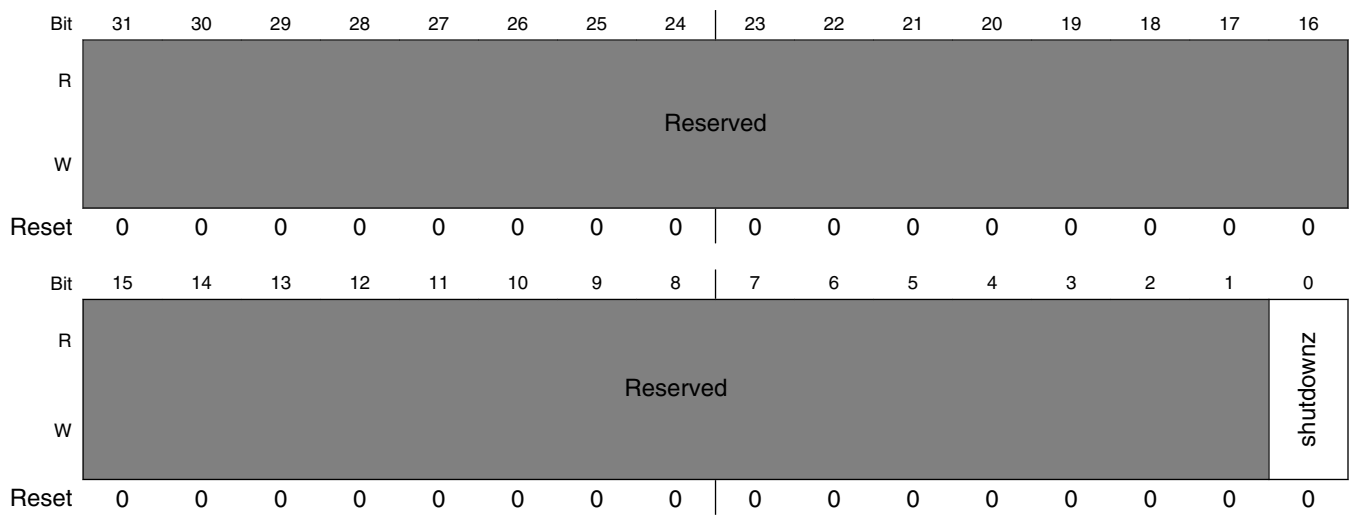
42.6.2 Core power up (MIPI_DSI_PWR_UP)

Size: 32 bits

Offset: 0x4

Memory Access: R/W

Address: 21E_0000h base + 4h offset = 21E_0004h



MIPI_DSI_PWR_UP field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 shutdownz	Core power up 0 reset; 1 power up)

42.6.3 Number of active data lanes (MIPI_DSI_CLKMGR_CFG)

Size: 32 bits

Offset: 0x8

Memory Access: R/W

Address: 21E_0000h base + 8h offset = 21E_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																TO_CLK_DIVIDSION								TX_ESC_CLK_DIVIDSION							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_CLKMGR_CFG field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15–8 TO_CLK_ DIVIDSION	Division factor for Time Out clock used as timing unit in the configuration of HS to LP and LP to HS transition error.
TX_ESC_CLK_ DIVIDSION	Division factor for TX ESCAPE clock source (lanebyteclk pin), values 0 and 1 stop TX_ESC clock generation.

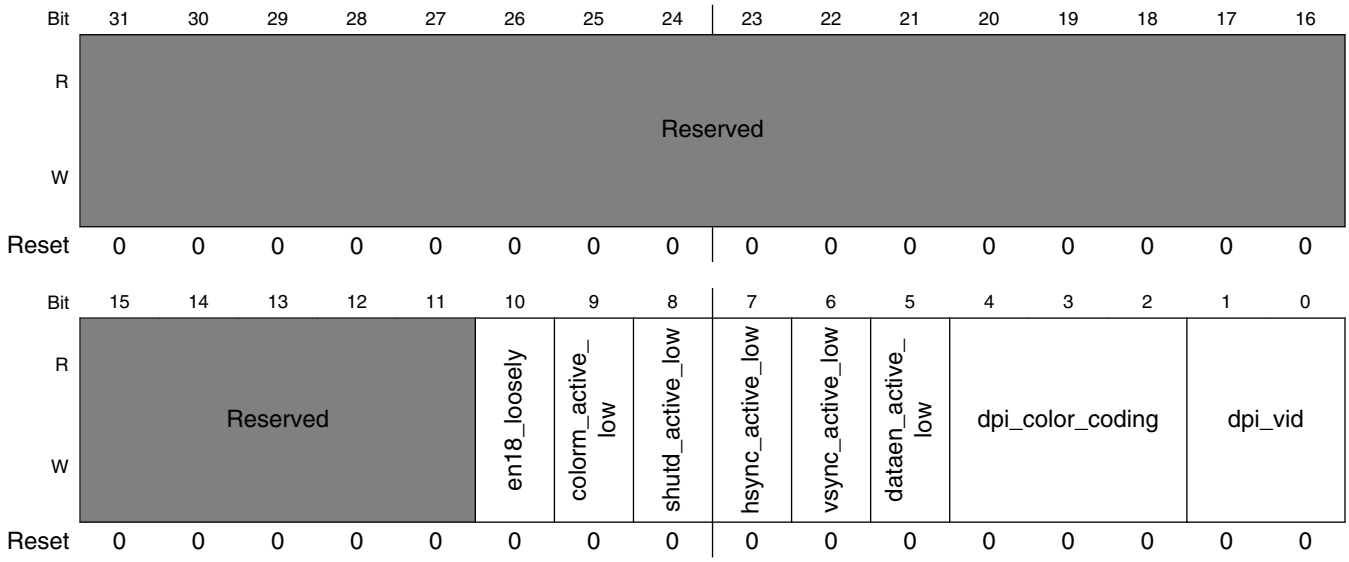
42.6.4 DPI interface configuration (MIPI_DSI_DPI_CFG)

Size: 32 bits

Offset: 0xc

Memory Access: R/W

Address: 21E_0000h base + Ch offset = 21E_000Ch



MIPI_DSI_DPI_CFG field descriptions

Field	Description
31–11 -	This field is reserved. Reserved
10 en18_loosely	Enable 18 loosely packet pixel stream.
9 colorm_active_low	Set to configure Color Mode pin (dpicolcorm) as Active low
8 shutd_active_low	Set to configure Shut Down pin (dpishutdn) as Active low
7 hsync_active_low	Set to configure Horizontal Synchronism pin (dpihsync) as Active low
6 vsync_active_low	Set to configure Vertical Synchronism pin (dpivsync) as Active low
5 dataen_active_low	Set to configure Data enable pin (dpidaten) as Active low
4–2 dpi_color_coding	DPI color coding. 0 16bit config1 1 16bit config2; 2 16bit config3; 3 18bit config1; 4 18bit config2; 5 to 7 24 bit.
dpi_vid	Configures the DPI Virtual Channel ID that will be indexed to the Video mode packets.

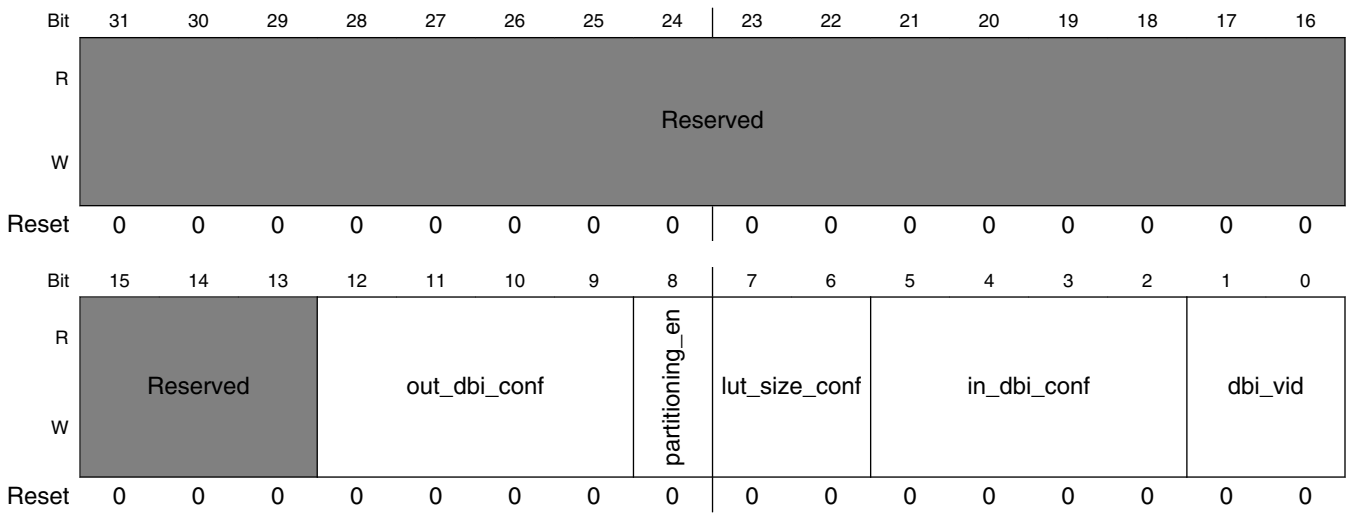
42.6.5 DBI interface configuration (MIPI_DSI_DBI_CFG)

Size: 32 bits

Offset: 0x10

Memory Access: R/W

Address: 21E_0000h base + 10h offset = 21E_0010h



MIPI_DSI_DBI_CFG field descriptions

Field	Description
31–13 -	This field is reserved. Reserved
12–9 out_dbi_conf	Configures the DBI output pixel data configuration; 0 8bit 8bpp; 1 8bit 12bpp; 2 8bit 16bpp; 3 8bit 18bpp; 4 8bit 24bpp; 5 9bit 18bpp; 6 16bit 8bpp; 7 16bit 12bpp; 8 16bit 16bpp; 9 16bit 18bpp, option1; 10 16bit 18bpp, option2; 11 16bit 24bpp, option1; 12 16bit 24bpp, option2
8 partitioning_en	Enables write memory continue through input command (system needs to ensure correct partitioning of Long Write commands)

Table continues on the next page...

MIPI_DSI_DBI_CFG field descriptions (continued)

Field	Description
7–6 lut_size_conf	Configures the size used to transport Write Lut commands; 0 16-bit color display; 1 18-bit color display; 2 24-bit color display; 3 16-bit color display
5–2 in_dbi_conf	Configures DBI input pixel data configuration; 0 8bit 8bpp; 1 8bit 12bpp; 2 8bit 16bpp; 3 8bit 18bpp; 4 8bit 24bpp; 5 9bit 18bpp; 6 16bit 8bpp; 7 16bit 12bpp; 8 16bit 16bpp; 9 16bit 18bpp, option1; 10 16bit 18bpp, option2; 11 16bit 24bpp, option1; 12 16bit 24bpp, option2
dbi_vid	Configures the DBI Virtual Channel ID that will be indexed to the DCS packets.

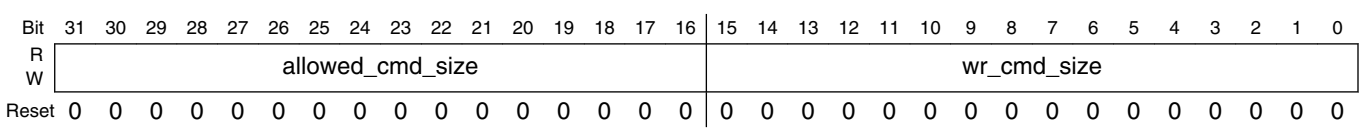
42.6.6 DBI command size configuration (MIPI_DSI_DBIS_CMDSIZE)

Size: 32 bits

Offset: 0x14

Memory Access: R/W

Address: 21E_0000h base + 14h offset = 21E_0014h



MIPI_DSI_DBIS_CMDSIZE field descriptions

Field	Description
31–16 allowed_cmd_size	Configures the maximum allowed size of a DCS write memory command. This register is used to partition a write memory command into several write memory continues. It is only used if bit 'partitioning_en' is disabled. Size of DSI packet payload is the actual payload size minus 1 since the DCS command is in the DSI packet payload.

Table continues on the next page...

MIPI_DSI_DBIS_CMDSIZE field descriptions (continued)

Field	Description
wr_cmd_size	Configures the size of the DCS write memory commands. Size of DSI packet payload is the actual payload size minus 1 since the DCS command is in the DSI packet payload.

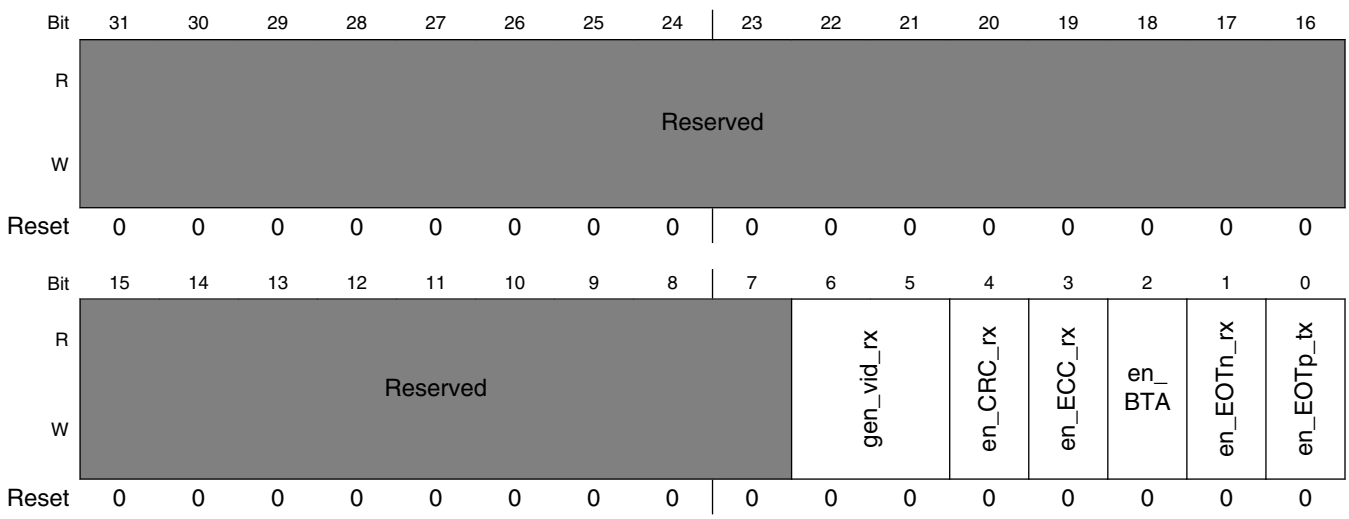
42.6.7 Packet handler configuration (MIPI_DSI_PCKHDL_CFG)

Size: 32 bits

Offset: 0x18

Memory Access: R/W

Address: 21E_0000h base + 18h offset = 21E_0018h



MIPI_DSI_PCKHDL_CFG field descriptions

Field	Description
31–7 -	This field is reserved. Reserved
6–5 gen_vid_rx	Generic interface read-back Virtual Channel identification
4 en_CRC_rx	Enables CRC reception and error reporting
3 en_ECC_rx	Enables ECC reception, error correction and reporting
2 en_BTA	Enables Bus Turn-Around request
1 en_EOTn_rx	Enables EOTp reception
0 en_EOTp_tx	Enables EOTp transmission

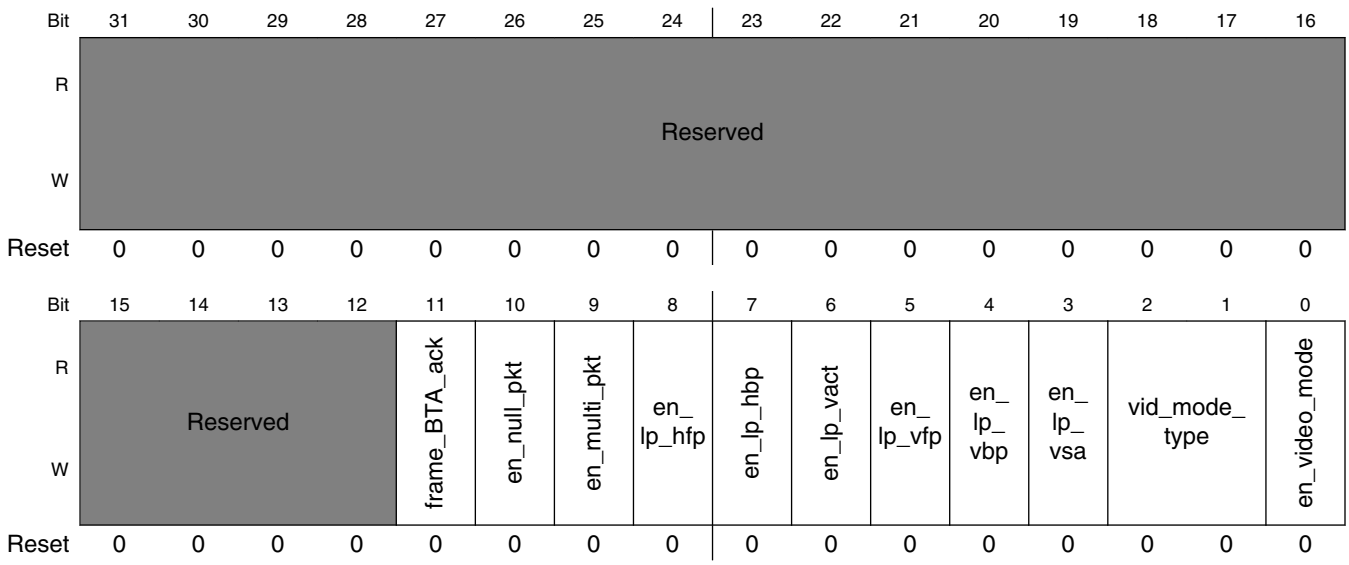
42.6.8 Video Mode Configuration (MIPI_DSI_VID_MODE_CFG)

Size: 32 bits

Offset: 0x1c

Memory Access: R/W

Address: 21E_0000h base + 1Ch offset = 21E_001Ch



MIPI_DSI_VID_MODE_CFG field descriptions

Field	Description
31–12 -	This field is reserved. Reserved
11 frame_BTA_ack	Enables the request for an acknowledge response at the end of a frame
10 en_null_pkt	Enables the tranmission of null packets in the HACT period
9 en_multi_pkt	Enables the tranmission of multi video packets in the HACT period
8 en_lp_hfp	Enables return to Low Power inside HFP period when timing allows
7 en_lp_hbp	Enables return to Low Power inside HBP period when timing allows
6 en_lp_vact	Enables return to Low Power inside VACT period when timing allows
5 en_lp_vfp	Enables return to Low Power inside VFP period when timing allows
4 en_lp_vbp	Enables return to Low Power inside VBP period when timing allows

Table continues on the next page...

MIPI_DSI_VID_MODE_CFG field descriptions (continued)

Field	Description
3 en_lp_vsa	Enables return to Low Power inside VSA period when timing allows
2–1 vid_mode_type	Selects video mode transmission type. 0: Non-burst with Sync pulses; 1: Non-burst with Sync events; 2-3: Burst with Sync pulses.
0 en_video_mode	Enables DPI Video mode transmission

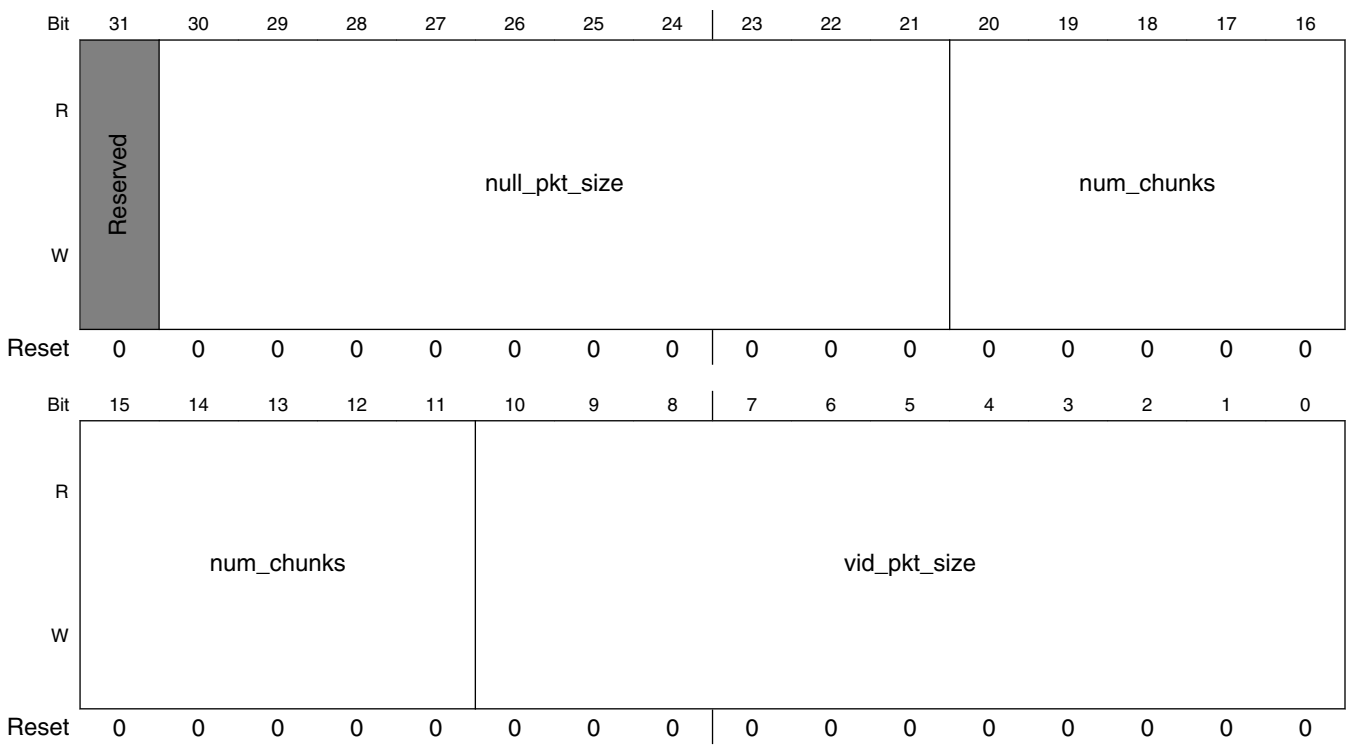
42.6.9 Video packet configuration (MIPI_DSI_VID_PKT_CFG)

Size: 32 bits

Offset: 0x20

Memory Access: R/W

Address: 21E_0000h base + 20h offset = 21E_0020h



MIPI_DSI_VID_PKT_CFG field descriptions

Field	Description
31 -	This field is reserved. Reserved

Table continues on the next page...

MIPI_DSI_VID_PKT_CFG field descriptions (continued)

Field	Description
30–21 null_pkt_size	Configures the number of bytes in a null packet
20–11 num_chunks	Configures the number of chunks to be transmitted during a Line period. (A chunk is a video packet or a null packet)
vid_pkt_size	Configures the number of pixel on a single video packet. (If using 18 bit mode and not enabling loosely packet stream this value must be a multiple of 4)

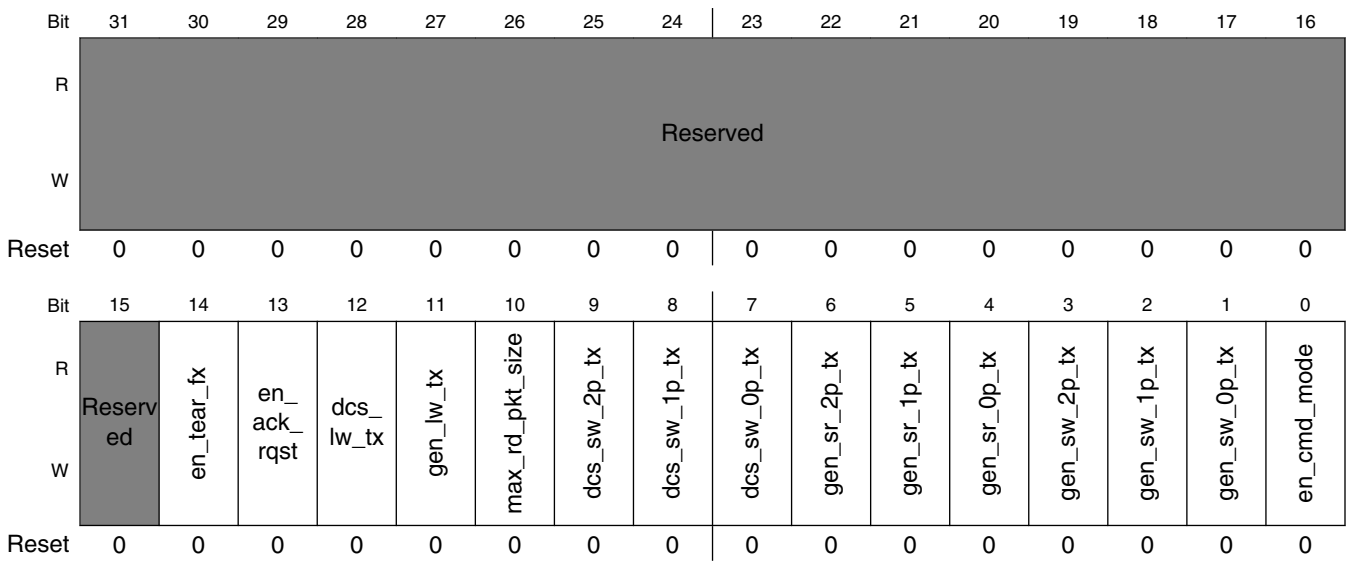
42.6.10 Command mode configuration (MIPI_DSI_CMD_MODE_CFG)

Size: 32 bits

Offset: 0x24

Memory Access: R/W

Address: 21E_0000h base + 24h offset = 21E_0024h



MIPI_DSI_CMD_MODE_CFG field descriptions

Field	Description
31–15 -	This field is reserved. Reserved
14 en_tear_fx	Enables the tearing effect acknowledge request
13 en_ack_rqst	Enables the acknowledge request after each packet transmission

Table continues on the next page...

MIPI_DSI_CMD_MODE_CFG field descriptions (continued)

Field	Description
12 dcs_lw_tx	Configures the DCS Long Write Packet command transmission type. 0 High Speed; 1 Low Power
11 gen_lw_tx	Configures the Generic Long Write Packet command transmission type. 0 High Speed; 1 Low Power
10 max_rd_pkt_size	Configures the Maximum Read Packet Size command transmission type. 0 High Speed; 1 Low Power
9 dcs_sw_2p_tx	Configures the DCS Short Write Packet with 2 Parameters command transmission type. 0 High Speed; 1 Low Power
8 dcs_sw_1p_tx	Configures the DCS Short Write Packet with 1 Parameters command transmission type. 0 High Speed; 1 Low Power
7 dcs_sw_0p_tx	Configures the DCS Short Write Packet with 0 Parameters command transmission type. 0 High Speed; 1 Low Power
6 gen_sr_2p_tx	Configures the Generic Short Read Packet with 2 Parameters command transmission type. 0 High Speed; 1 Low Power
5 gen_sr_1p_tx	Configures the Generic Short Read Packet with 1 Parameters command transmission type. 0 High Speed; 1 Low Power
4 gen_sr_0p_tx	Configures the Generic Short Read Packet with 0 Parameters command transmission type. 0 High Speed; 1 Low Power
3 gen_sw_2p_tx	Configures the Generic Short Write Packet with 2 Parameters command transmission type. 0 High Speed; 1 Low Power
2 gen_sw_1p_tx	Configures the Generic Short Write Packet with 1 Parameters command transmission type. 0 High Speed; 1 Low Power
1 gen_sw_0p_tx	Configures the Generic Short Write Packet with 0 Parameters command transmission type. 0 High Speed; 1 Low Power

Table continues on the next page...

MIPI_DSI_CMD_MODE_CFG field descriptions (continued)

Field	Description
0 en_cmd_mode	Enables the Command Mode Protocol for transmissions.

42.6.11 Line timer configuration (MIPI_DSI_TMR_LINE_CFG)

Size: 32 bits

Offset: 0x28

Memory Access: R/W

Address: 21E_0000h base + 28h offset = 21E_0028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	hline_time																hbp_time								hsa_time							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_TMR_LINE_CFG field descriptions

Field	Description
31–18 hline_time	Configures the size of the total line counted in lane byte cycles
17–9 hbp_time	Configures the Horizontal Back Porch period in lane byte clock cycles
hsa_time	Configures the Horizontal Synchronism Active period in lane byte clock cycles

42.6.12 Vertical timing configuration (MIPI_DSI_VTIMING_CFG)

Size: 32 bits

Offset: 0x2c

Memory Access: R/W

Address: 21E_0000h base + 2Ch offset = 21E_002Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					v_active_lines											vfp_lines						vbp_lines						vsa_lines			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_VTIMING_CFG field descriptions

Field	Description
31–27 -	This field is reserved. Reserved
26–16 v_active_lines	Configures the Vertical Active period measured in horizontal lines
15–10 vfp_lines	Configures the Vertical Front Porch period measured in horizontal lines
9–4 vbp_lines	Configures the Vertical Back Porch period measured in horizontal lines
vsa_lines	Configures the Vertical Synchronism Active period measured in horizontal lines

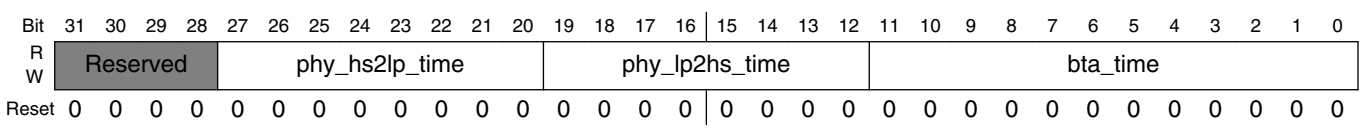
42.6.13 D-PHY timing configuration (MIPI_DSI_PHY_TMR_CFG)

Size: 32 bits

Offset: 0x30

Memory Access: R/W

Address: 21E_0000h base + 30h offset = 21E_0030h



MIPI_DSI_PHY_TMR_CFG field descriptions

Field	Description
31–28 -	This field is reserved. Reserved
27–20 phy_hs2lp_time	Configures the maximum time that the PHY takes to go from High Speed to Low Power transmission measured in lane byte clock cycles
19–12 phy_lp2hs_time	Configures the maximum time that the PHY takes to go from Low Power to High Speed transmission measured in lane byte clock cycles
bta_time	Configures the maximum time required to perform the Bus Turn Around operation measured in lane byte clock cycles

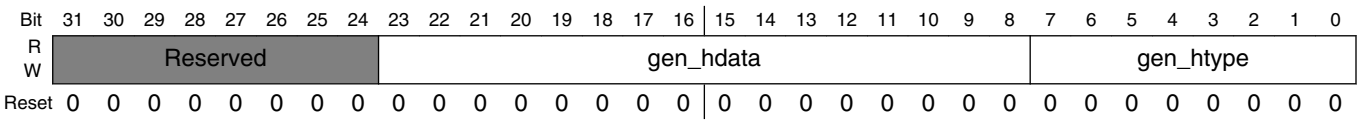
42.6.14 Generic packet Header configuration (MIPI_DSI_GEN_HDR)

Size: 32 bits

Offset: 0x34

Memory Access: R/W

Address: 21E_0000h base + 34h offset = 21E_0034h



MIPI_DSI_GEN_HDR field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–8 gen_hdata	Configures the packet data to be transmitted through the generic interface
gen_htype	Configures the packet type to be transmitted through the generic interface. Writing to this register triggers packet transmission (Payload must be written in advance)

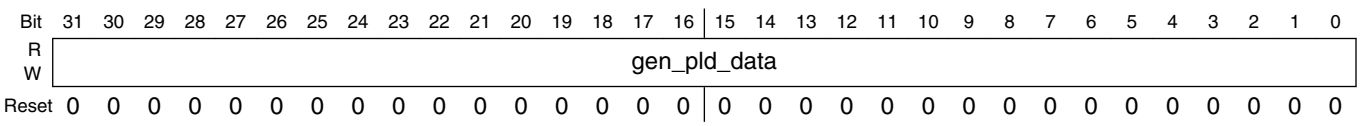
42.6.15 Generic payload data in/out (MIPI_DSI_GEN_PLD_DATA)

Size: 32 bits

Offset: 0x38

Memory Access: R/W

Address: 21E_0000h base + 38h offset = 21E_0038h



MIPI_DSI_GEN_PLD_DATA field descriptions

Field	Description
gen_pld_data	This register contains the input/output generic packet data. Write access to it writes the content of the packet payload. Read access reads the incoming generic read data

42.6.16 Command packet status (MIPI_DSI_CMD_PKT_STATUS)

Size: 32 bits

Offset: 0x3c

Memory Access: R/W

Address: 21E_0000h base + 3Ch offset = 21E_003Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	reserved2	dbi_rd_cmd_busy	dbi_pld_r_full	dbi_pld_r_empty	dbi_pld_w_full	dbi_pld_w_empty	dbi_cmd_full	dbi_cmd_empty	reserved1	gen_rd_cmd_busy	gen_pld_r_full	gen_pld_r_empty	gen_pld_w_full	gen_pld_w_empty	gen_cmd_full	gen_cmd_empty
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_CMD_PKT_STATUS field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 reserved2	This field is reserved. Reserved
14 dbi_rd_cmd_busy	Set when a read command is issued and cleared once the entire response is stored in the FIFO
13 dbi_pld_r_full	Reports the full status of the DBI read payload FIFO
12 dbi_pld_r_empty	Reports the empty status of the DBI read payload FIFO
11 dbi_pld_w_full	Reports the full status of the DBI write payload FIFO
10 dbi_pld_w_empty	Reports the empty status of the DBI write payload FIFO
9 dbi_cmd_full	Reports the full status of the DBI command FIFO
8 dbi_cmd_empty	Reports the empty status of the DBI command FIFO
7 reserved1	This field is reserved. Reserved
6 gen_rd_cmd_busy	Set when a read command is issued and cleared once the entire response is stored in the FIFO
5 gen_pld_r_full	Reports the full status of the generic read payload FIFO
4 gen_pld_r_empty	Reports the empty status of the generic read payload FIFO

Table continues on the next page...

MIPI_DSI_CMD_PKT_STATUS field descriptions (continued)

Field	Description
3 gen_pld_w_full	Reports the full status of the generic write payload FIFO
2 gen_pld_w_ empty	Reports the empty status of the generic write payload FIFO
1 gen_cmd_full	Reports the full status of the generic command FIFO
0 gen_cmd_empty	Reports the empty status of the generic command FIFO

42.6.17 Time Out timers configuration (MIPI_DSI_TO_CNT_CFG0)

Size: 32 bits

Offset: 0x40

Memory Access: R/W

Address: 21E_0000h base + 40h offset = 21E_0040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	lprx_to_cnt																hstx_to_cnt															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_TO_CNT_CFG0 field descriptions

Field	Description
31–16 lprx_to_cnt	Configures the time out counter that will trigger a Low Power Reception Time Out Contention Detection. (Measured in TO_CLK_DIVISION cycles)
hstx_to_cnt	Configures the time out counter that will trigger a High Speed Transmission Time Out Contention Detection (Measured in TO_CLK_DIVISION cycles)

42.6.18 Interrupt status register 0 (MIPI_DSI_ERROR_ST0)

Size: 32 bits

Offset: 0x44

Memory Access: R/W

Address: 21E_0000h base + 44h offset = 21E_0044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											dphy_errors_4	dphy_errors_3	dphy_errors_2	dphy_errors_1	dphy_errors_0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ack_with_err_15	ack_with_err_14	ack_with_err_13	ack_with_err_12	ack_with_err_11	ack_with_err_10	ack_with_err_9	ack_with_err_8	ack_with_err_7	ack_with_err_6	ack_with_err_5	ack_with_err_4	ack_with_err_3	ack_with_err_2	ack_with_err_1	ack_with_err_0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_ERROR_ST0 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 dphy_errors_4	ErrContentionLP1 LP1 Contention Error from Lane 0
19 dphy_errors_3	ErrContentionLP0 LP0 Contention Error from Lane 0
18 dphy_errors_2	ErrControl Control Error from Lane 0
17 dphy_errors_1	ErrSyncEsc Low-Power Data Transmission Synchronization Error from Lane 0
16 dphy_errors_0	ErrEsc Escape Entry Error from Lane 0
15 ack_with_err_15	Retrieves DSI Protocol Violation from Display Acknowledge Error Report
14 ack_with_err_14	Retrieves Reserved (specific to device) from Display Acknowledge Error Report
13 ack_with_err_13	Retrieves Invalid Transmission Length from Display Acknowledge Error Report
12 ack_with_err_12	Retrieves DSI VC ID Invalid from Display Acknowledge Error Report
11 ack_with_err_11	Retrieves DSI Data Type Not Recognized from Display Acknowledge Error Report
10 ack_with_err_10	Retrieves Checksum Error (Long packet only) from Display Acknowledge Error Report
9 ack_with_err_9	Retrieves ECC Error, multi-bit (detected, not corrected) from Display Acknowledge Error Report
8 ack_with_err_8	Retrieves ECC Error, single-bit (detected and corrected) from Display Acknowledge Error Report

Table continues on the next page...

MIPI_DSI_ERROR_ST0 field descriptions (continued)

Field	Description
7 ack_with_err_7	Retrieves Reserved (specific to device) from Display Acknowledge Error Report
6 ack_with_err_6	Retrieves False Control Error from Display Acknowledge Error Report
5 ack_with_err_5	Retrieves HS Receive Timeout Error from Display Acknowledge Error Report
4 ack_with_err_4	Retrieves Low-Power Transmit Sync Error from Display Acknowledge Error Report
3 ack_with_err_3	Retrieves Escape Mode Entry Command Error from Display Acknowledge Error Report
2 ack_with_err_2	Retrieves EoT Sync Error from Display Acknowledge Error Report
1 ack_with_err_1	Retrieves SoT Sync Error from Display Acknowledge Error Report
0 ack_with_err_0	Retrieves SoT Error from Display Acknowledge Error Report

42.6.19 Interrupt status register 1 (MIPI_DSI_ERROR_ST1)

Size: 32 bits

Offset: 0x48

Memory Access: R/W

Address: 21E_0000h base + 48h offset = 21E_0048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														dbi_illegal_comm_err	dbi_pld_recv_err
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	dbi_pld_rd_err	dbi_pld_wr_err	dbi_cmd_wr_err	gen_pld_recv_err	gen_pld_rd_err	gen_pld_send_err	gen_pld_wr_err	gen_cmd_wr_err	dpi_pld_wr_err	eopt_err	pkt_size_err	crc_err	ecc_multi_err	ecc_sinlge_err	to_lp_rx	to_hs_tx
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_ERROR_ST1 field descriptions

Field	Description
31–18 -	This field is reserved. Reserved
17 dbi_illegal_comm_err	Attempt to write an illegal command on the DPI interface and core blocked by transmission
16 dbi_pld_recv_err	During a DBI read back packet, the payload FIFO went full and received data was corrupted
15 dbi_pld_rd_err	During a DCS read data, the payload FIFO went empty and data was send to the interface corrupted
14 dbi_pld_wr_err	System tried to write payload data through the DBI interface and the FIFO was full, therefore the Command was not written
13 dbi_cmd_wr_err	System tried to write a command through the DBI but the command FIFO was full, therefore the command was not written
12 gen_pld_recv_err	During a generic interface packet read back, the payload FIFO went full and received data was corrupted
11 gen_pld_rd_err	During a DCS read data, the payload FIFO went empty and data was send to the interface corrupted
10 gen_pld_send_err	During a generic interface packet build, the payload FIFO went empty and data was sent corrupted
9 gen_pld_wr_err	System tried to write a payload data through the generic interface and FIFO was full, therefore the payload was not written
8 gen_cmd_wr_err	System tried to write a command through the generic interface and FIFO was full, therefore the command was not written
7 dpi_pld_wr_err	During a DPI pixel line storage the payload FIFO went full and data stored is corrupted
6 eopt_err	EOTp Packet not received at the end of the incoming peripheral transmission
5 pkt_size_err	Packet size error was detected during packet reception
4 crc_err	CRC error was detected in the received packet payload
3 ecc_multi_err	ECC multiple error was detected in a received packet
2 ecc_sinlge_err	ECC single error was detected and corrected in a received packet
1 to_lp_rx	Low Power Reception Time Out Counter reached the end and Contention Detection as been detected
0 to_hs_tx	High Speed Transmission Time Out Counter reached the end and Contention Detection as been detected

42.6.20 Masks Interrupt generation trigged by ERROR_ST0 register (MIPI_DSI_ERROR_MSK0)

Size: 32 bits

Offset: 0x4c

Memory Access: R/W

Address: 21E_0000h base + 4Ch offset = 21E_004Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved											dphy_errors_4	dphy_errors_3	dphy_errors_2	dphy_errors_1	dphy_errors_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ack_with_err_15	ack_with_err_14	ack_with_err_13	ack_with_err_12	ack_with_err_11	ack_with_err_10	ack_with_err_9	ack_with_err_8	ack_with_err_7	ack_with_err_6	ack_with_err_5	ack_with_err_4	ack_with_err_3	ack_with_err_2	ack_with_err_1	ack_with_err_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_ERROR_MSK0 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 dphy_errors_4	ErrContentionLP1 LP1 Contention Error from Lane 0
19 dphy_errors_3	ErrContentionLP0 LP0 Contention Error from Lane 0
18 dphy_errors_2	ErrControl Control Error from Lane 0
17 dphy_errors_1	ErrSyncEsc Low-Power Data Transmission Synchronization Error from Lane 0
16 dphy_errors_0	ErrEsc Escape Entry Error from Lane 0
15 ack_with_err_15	Masks DSI Protocol Violation from Display Acknowledge Error Report
14 ack_with_err_14	Masks Reserved (specific to device) from Display Acknowledge Error Report
13 ack_with_err_13	Masks Invalid Transmission Length from Display Acknowledge Error Report
12 ack_with_err_12	Masks DSI VC ID Invalid from Display Acknowledge Error Report
11 ack_with_err_11	Masks DSI Data Type Not Recognized from Display Acknowledge Error Report
10 ack_with_err_10	Masks Checksum Error (Long packet only) from Display Acknowledge Error Report

Table continues on the next page...

MIPI_DSI_ERROR_MSK0 field descriptions (continued)

Field	Description
9 ack_with_err_9	Masks ECC Error, multi-bit (detected, not corrected) from Display Acknowledge Error Report
8 ack_with_err_8	Masks ECC Error, single-bit (detected and corrected) from Display Acknowledge Error Report
7 ack_with_err_7	Masks Reserved (specific to device) from Display Acknowledge Error Report
6 ack_with_err_6	Masks False Control Error from Display Acknowledge Error Report
5 ack_with_err_5	Masks HS Receive Timeout Error from Display Acknowledge Error Report
4 ack_with_err_4	Masks Low-Power Transmit Sync Error from Display Acknowledge Error Report
3 ack_with_err_3	Masks Escape Mode Entry Command Error from Display Acknowledge Error Report
2 ack_with_err_2	Masks EoT Sync Error from Display Acknowledge Error Report
1 ack_with_err_1	Masks SoT Sync Error from Display Acknowledge Error Report
0 ack_with_err_0	Masks SoT Error from Display Acknowledge Error Report

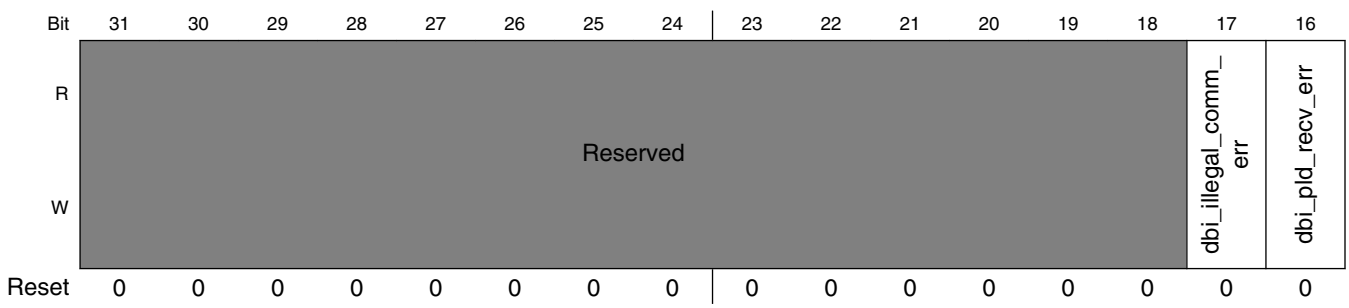
42.6.21 Masks Interrupt generation trigged by ERROR_ST1 register (MIPI_DSI_ERROR_MSK1)

Size: 32 bits

Offset: 0x50

Memory Access: R/W

Address: 21E_0000h base + 50h offset = 21E_0050h



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	dbi_pld_rd_err	dbi_pld_wr_err	dbi_cmd_wr_err	gen_pld_recv_err	gen_pld_rd_err	gen_pld_send_err	gen_pld_wr_err	gen_cmd_wr_err	dpi_pld_wr_err	eopt_err	pkt_size_err	crc_err	ecc_multi_err	ecc_sinlge_err	to_lp_rx	to_hs_tx
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_ERROR_MSK1 field descriptions

Field	Description
31–18 -	This field is reserved. Reserved
17 dbi_illegal_comm_err	Masks error attempt to write an illegal command on DPI
16 dbi_pld_recv_err	Masks DBI read back packet payload FIFO full error
15 dbi_pld_rd_err	Masks payload DBI FIFO empty error
14 dbi_pld_wr_err	Masks write payload data DBI FIFO full error
13 dbi_cmd_wr_err	Masks DBI command FIFO full error
12 gen_pld_recv_err	Masks generic interface packet read back FIFO full error
11 gen_pld_rd_err	Masks DCS read data payload FIFO empty error
10 gen_pld_send_err	Masks generic interface packet build FIFO empty error
9 gen_pld_wr_err	Masks payload data FIFO of generic interface full error
8 gen_cmd_wr_err	Masks command FIFO of generic interface full error
7 dpi_pld_wr_err	Masks DPI pixel line payload FIFO full error
6 eopt_err	Masks EOTp Packet not received error
5 pkt_size_err	Masks Packet size error
4 crc_err	Masks CRC error
3 ecc_multi_err	Masks ECC multiple error
2 ecc_sinlge_err	Masks ECC single error

Table continues on the next page...

MIPI_DSI_ERROR_MSK1 field descriptions (continued)

Field	Description
1 to_lp_rx	Masks Low Power Reception Time Out Counter error
0 to_hs_tx	Masks High Speed Transmission Time Out Counter error

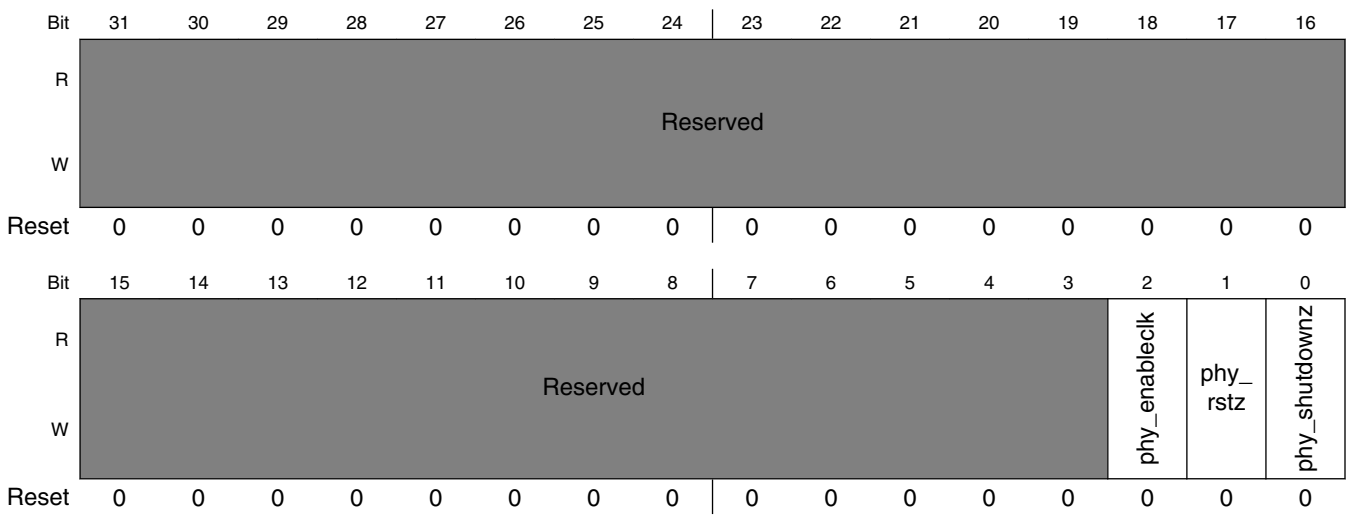
42.6.22 D-PHY reset control (MIPI_DSI_PHY_RSTZ)

Size: 32 bits

Offset: 0x54

Memory Access: R/W

Address: 21E_0000h base + 54h offset = 21E_0054h



MIPI_DSI_PHY_RSTZ field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
2 phy_enableclk	Enables D-PHY Clock Lane Module when 1
1 phy_rstz	D-PHY Reset disable when 1, used to place the digital section of D-PHY in reset state
0 phy_shutdownz	D-PHY Shutdown disable when 1, used to place the complete D-PHY macro in power down

42.6.23 D-PHY interface configuration (MIPI_DSI_PHY_IF_CFG_)

Size: 32 bits

Offset: 0x58

Memory Access: R/W

Address: 21E_0000h base + 58h offset = 21E_0058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						phy_stop_wait_time								n_lanes	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_PHY_IF_CFG_ field descriptions

Field	Description
31–10 -	This field is reserved. Reserved
9–2 phy_stop_wait_time	Configures minimum wait period to request an HS transmission after the stop state accounted in clock lane cycles
n_lanes	Number of active data lanes. 00 1 Data Lane (Lane 0) 01 2 Data Lanes (Lane 0, and Lane 1) 10 Reserved 11 Reserved

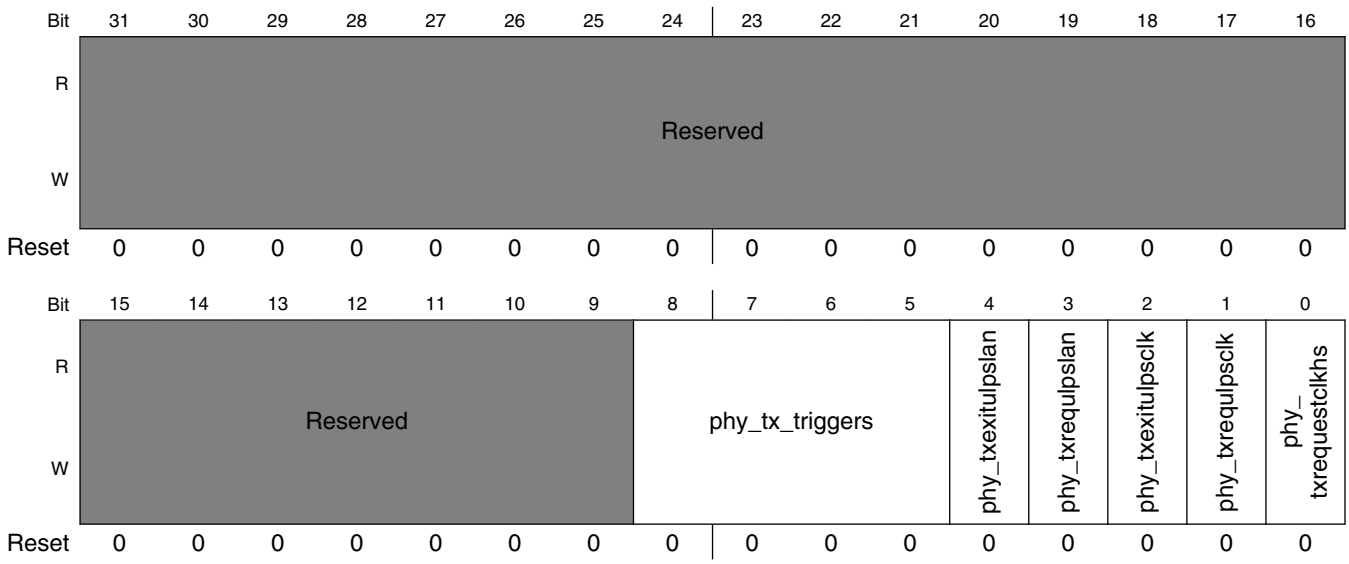
42.6.24 D-PHY PPI interface control (MIPI_DSI_PHY_IF_CTRL)

Size: 32 bits

Offset: 0x5c

Memory Access: R/W

Address: 21E_0000h base + 5Ch offset = 21E_005Ch



MIPI_DSI_PHY_IF_CTRL field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8–5 phy_tx_triggers	Controls the trigger transmissions
4 phy_txexitulpslan	ULPS mode Exit on on all active data lanes
3 phy_txrequlpslan	ULPS mode Request on all active data lanes
2 phy_txexitulpsclk	ULPS mode Exit on Clock Lane
1 phy_txrequlpsclk	ULPS mode Request on Clock Lane
0 phy_ txrequestclkhs	Controls D-PHY PPI txrequestclkHS

42.6.25 D-PHY PPI status interface (MIPI_DSI_PHY_STATUS)

Size: 32 bits

Offset: 0x60

Memory Access: R/W

Address: 21E_0000h base + 60h offset = 21E_0060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			ulpsactivenot3lane	phystopstate3lane	ulpsactivenot2lane	phystopstate2lane	ulpsactivenot1lane	phystopstate1lane	rxulpsesc0lane	ulpsactivenot0lane	phystopstate0lane	phyrxulpsclknot	phystopstateclklane	phydirection	phylock
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_PHY_STATUS field descriptions

Field	Description
31–13 -	This field is reserved. Reserved
12 ulpsactivenot3lane	Reports status of ulpsactivenot3lane D-PHY pin
11 phystopstate3lane	Reports status of phystopstate3lane D-PHY pin
10 ulpsactivenot2lane	Reports status of ulpsactivenot2lane D-PHY pin
9 phystopstate2lane	Reports status of phystopstate2lane D-PHY pin
8 ulpsactivenot1lane	Reports status of ulpsactivenot1lane D-PHY pin
7 phystopstate1lane	Reports status of phystopstate1lane D-PHY pin
6 rxulpsesc0lane	Reports status of rxulpsEsc0lane D-PHY pin
5 ulpsactivenot0lane	Reports status of ulpsactivenot0lane D-PHY pin
4 phystopstate0lane	Reports status of phystopstate0lane D-PHY pin
3 phyrxulpsclknot	Reports status of phyrxulpsclknot D-PHY pin
2 phystopstateclklane	Reports status of phystopstateclklane D-PHY pin
1 phydirection	Reports status of phydirection D-PHY pin
0 phylock	Reports status of phylock D-PHY pin

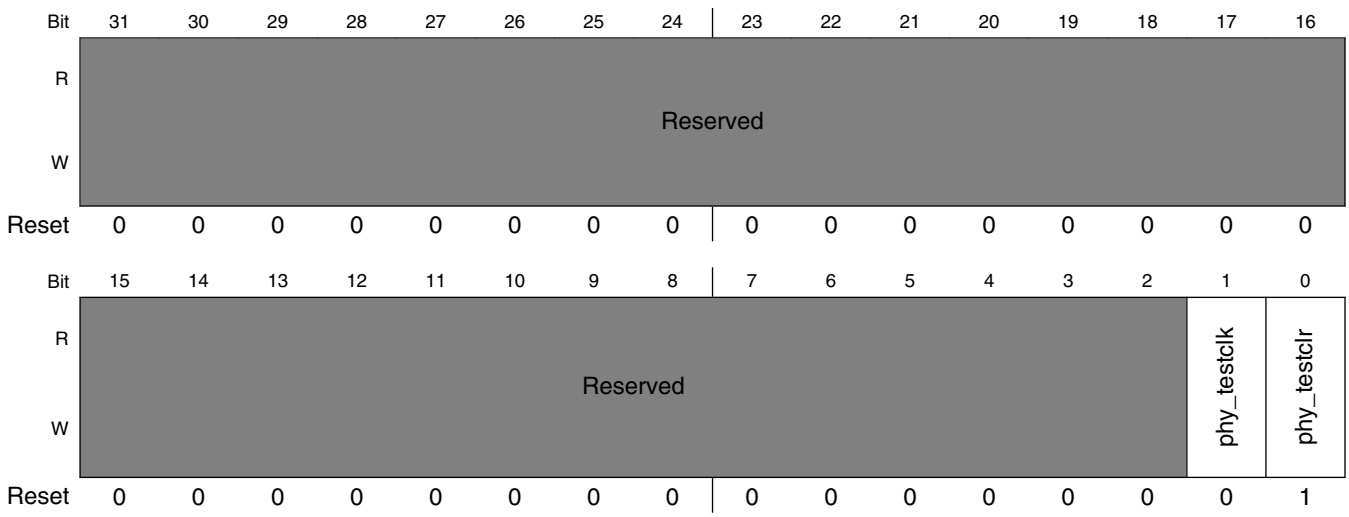
42.6.26 D-PHY Test interface control 0 (MIPI_DSI_PHY_TST_CTRL0)

Size: 32 bits

Offset: 0x64

Memory Access: R/W

Address: 21E_0000h base + 64h offset = 21E_0064h



MIPI_DSI_PHY_TST_CTRL0 field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
1 phy_testclk	PHY test interface strobe signal. Used to clock TESTDIN bus into the D-PHY. In conjunction with TESTEN signal controls the operation selection
0 phy_testclr	PHY test interface clear. When active performs vendor specific interface initialization (Active High)

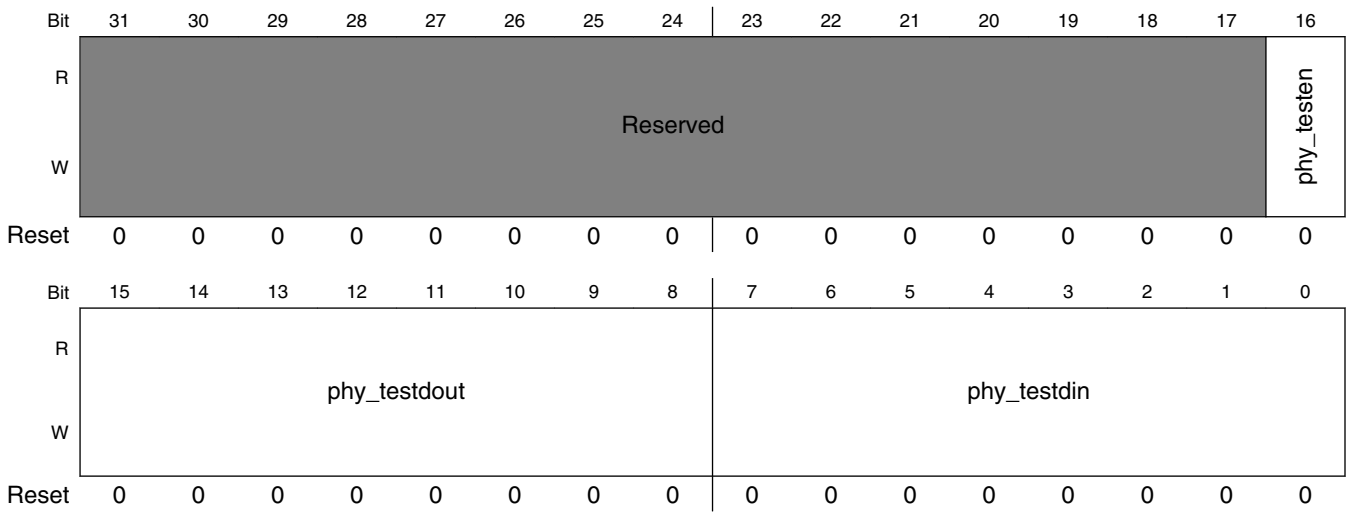
42.6.27 D-PHY Test interface control 1 (MIPI_DSI_PHY_TST_CTRL1)

Size: 32 bits

Offset: 0x68

Memory Access: R/W

Address: 21E_0000h base + 68h offset = 21E_0068h



MIPI_DSI_PHY_TST_CTRL1 field descriptions

Field	Description
31–17 -	This field is reserved. Reserved
16 phy_testen	PHY test interface operation selector: when 1 configures address write operation on the falling edge of TESTCLK; when 0 configures a data write operation on the rising edge of TESTCLK
15–8 phy_testdout	PHY output 8-bit data bus for read-back and internal probing functionalities
phy_testdin	PHY test interface input 8-bit data bus for internal register programming and test functionalities access

Chapter 43
MIPI HSI Host Controller (MIPI_HSI)

43.1 Overview

MIPI_HSI is an interface intended to connect an application processor to a cellular modem controller in cellular handsets, but it can be used in other applications.

This document describes the micro-architecture for the MIPI_HSI host controller and the block level functionality and implementation in detail.

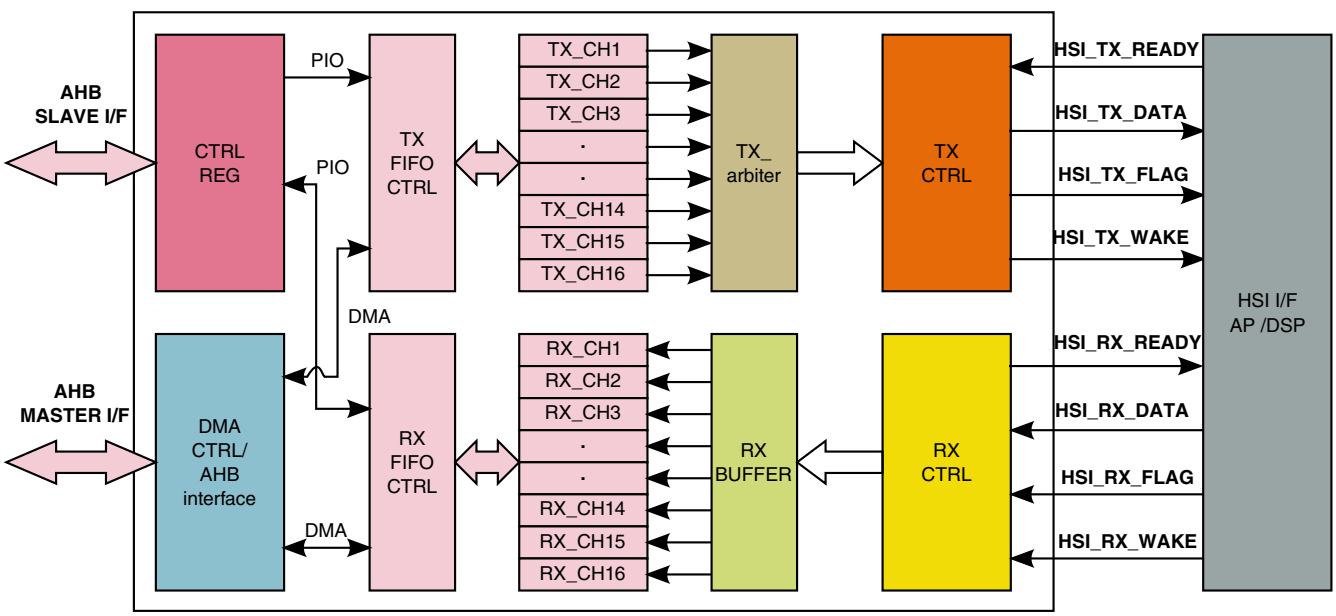


Figure 43-1. Top-Level Architecture

43.1.1 Features

The features of MIPI_HSI include the following:

- AHB slave interface support for PIO and configure.
- Supports PIO and DMA access mode
- Supports AHB Master interface to SOC interconnect
- Compatible with the MIPI_HSI specification version1.0 & MIPI_HSI Physical Layer v1.01.00 specification
- Full-Duplex High Speed Serial interface
- Supports 16 logical channels for both transmit and receive operations
- Each channel is configurable and can be enabled and disabled
- Bandwidth for each channel is programmable
- Supports both stream mode and frame mode for data transmission and reception
- Maximum bandwidth up to 200Mbs in both transmit and receive directions
- Configurable transmit and receive FIFO
- Programmable transmission bit rate
- Support round-robin arbitration for transmission mode
- Run-time configurability of channel ID bits
- Data time out feature supported for receive operation

43.2 External Signals

43.2.1 External Signals Overview

MIPI_HSI has 8 associate I/O signals.

- HSI_RX_DATA: Serial data bus
- HSI_RX_FLAG: Flag signal. If data on HSI_RX_DATA is toggling then flag stays constant and if data is constant then flag should toggle. It can be used with HSI_RX_DATA.
- HSI_RX_READY: Link data flow control signal. It indicates HSI RX is ready to receive.
- HSI_RX_WAKE: Link Control signal used to wake up the receiver.
- HSI_TX_DATA: Serial data out
- HSI_TX_FLAG: Flag signal. If data on HSI_TX_DATA is toggling, then flag stays constant and if data is constant, then flag should toggle.
- HSI_TX_READY: Link data flow control signal. It indicates HSI RX of other die is ready.
- HSI_TX_WAKE: Assertion of this signal shows that HSI TX wants to transfer some data. This signal is used to wake up HSI RX of other die.

43.2.2 Ports Table

See the following table for the signal properties of the I/Os.

Table 43-1. Properties of I/O Signals

Name	Port	Function	Reset State	Pull up
HSI_RX_DATA	I	Serial data bus	N/A	N/A
HSI_RX_FLAG	I	Flag signal. If data on HSI_RX_DATA is toggling then flag stays constant and if data is constant then flag should toggle. It can be used with HSI_RX_DATA.	N/A	N/A
HSI_RX_READY	O	Link data flow control signal. It indicates HSI RX is ready to receive	N/A	N/A
HSI_RX_WAKE	I	Link Control signal used to wake up the receiver so that transmitter starts a transmission	N/A	N/A
HSI_TX_DATA	O	Serial data out	0	N/A
HSI_TX_FLAG	O	Flag signal. If data on tx_data is toggling, then flag stays constant and if data is constant, then flag should toggle.	0	N/A
HSI_TX_READY	I	Link data flow control signal. It indicates HSI RX of other die is ready	N/A	N/A
HSI_TX_WAKE	O	Assertion of this signal shows that HSI TX wants to transfer some data. This signal is used to wake up HSI RX of other die	0	N/A

43.3 Clocks

The table found here describes the clock sources for MIPI HSI.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 43-2. MIPI HSI Clocks

Clock name	Clock Root	Description
h_clk	ahb_clk_root	AHB clock
tx_ref_clk	hsi_tx_clk_root	Tx reference clock

43.4 Functional Description

The following sections provide a brief functional description of the major system blocks, including the Data Channel Buffer, DMA AHB interface, register bank as well as AHB slave Bus interface, dual-port memory wrapper, clock & reset manager and clock generator.

43.4.1 DMA AHB /PIO AHB slave and Data Channel Buffer

The HSI uses one configurable data buffer so data can be transferred between the system bus (AHB slave Bus or AHB Master Bus) and the HSI controller.

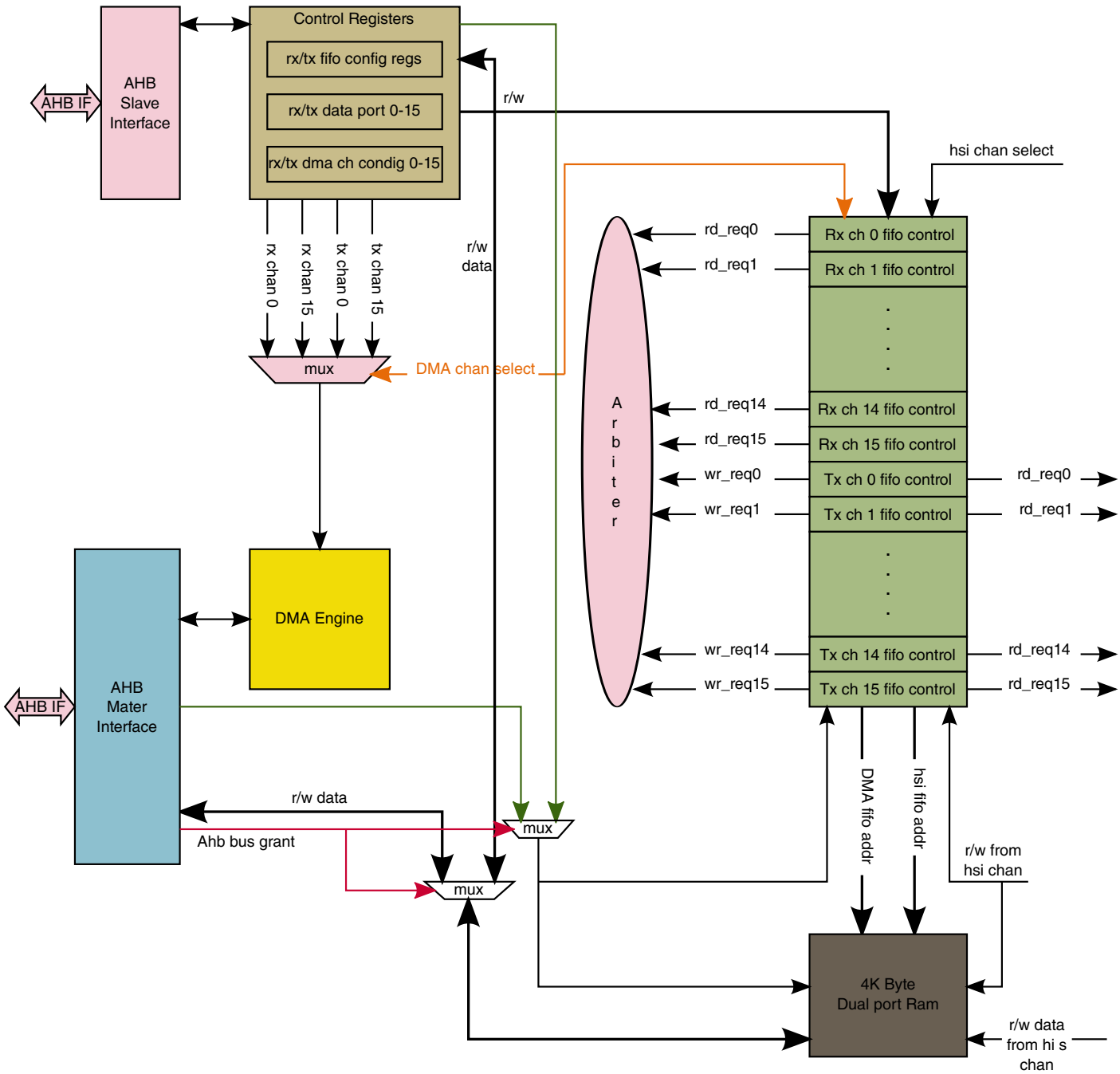


Figure 43-2. HSI DMA AHB/PIO AHB slave and data channel Buffer Scheme

43.4.1.1 AHB Slave Interface

This interface is compatible with AMBA AHB2.0 interface.

43.4.1.2 AHB Master Interface

This block access bus data by burst 8, burst 4 or single. AHB bus error is support.

43.4.1.3 Control Registers

This block has all the control registers of HSI Controller.

43.4.1.4 DMA Engine

This block generate DMA commands and send it to AHB master interface. These DMA commands are generated according current DMA channel configuration registers. AHB Master Interface should access AHB bus according the commands.

43.4.1.5 Rx/Tx FIFO controller

There are 32 sets of Rx/Tx FIFO Controller. Each FIFO Controller is corresponding to a Rx/Tx channel and a DMA channel. The depth for Each FIFO Controller is configurable. Once the number of data in the FIFO is less (for Tx) or more (for Rx) than the burst length of the corresponding DMA channel, it will send wr_req/rd_req signal to the Arbiter.

43.4.1.6 Arbiter

The arbiter arbitrates the requests from all 32 FIFO controllers according round robin priority.

43.4.1.7 4K Bytes Dual Port RAM

This block is used to save all FIFO data.

43.4.1.8 Typical Software Operation Flow for DMA

1. Configure Tx/Rx FIFO Size Configuration Registers. The total size of the FIFOs should not more than 1K DWords.

2. Enable RDMA/TDMA channel Complete Irq and Error Irq.

3. Configure RDMA/TDMA Start Address, Burst Size and Trans_length for each DMA channel. After that, set the RDMA/TDMA Enable bit. Once the Enable bit is set, the DMA channel will start to work right away.

4. Wait HSI Controller Irq and check RDMA/TDMA Complete Irq Stat bit.

43.4.1.9 Typical Software Operation Flow 1 for Data Port

1. Read corresponding Tx/Rx FIFO Status Register to make sure FIFO isn't empty/full.

2. Read/write Tx/Rx channel Data Port registers to access channel FIFO data.

43.4.1.10 Typical Software Operation Flow 2 for Data Port

1. Configure corresponding Tx/Rx FIFO Threshold Register.

2. Enable Tx/Rx FIFO Threshold interrupt.

3. Wait Tx/Rx FIFO Threshold interrupt.

4. Read/write Tx/Rx channel Data Port registers to access channel FIFO data.

43.4.2 HSI Reset

There are two kinds of reset signals within HSI:

1. Hardware reset
2. Software reset for all

These signals are fed into this module and stable signals are generated inside the module to reset all other modules. The module also gates off all the inside signals.

This module controls all the reset signals within the HSI.

43.5 HSI Memory Map/Register Definition

This section includes the module memory map and detailed descriptions of all registers.

NOTE

The HSI registers are 32 bits wide and only support 32-bit access.

NOTE

The term reset refers to power-on-reset. It will be specified whenever the software reset value differs from the power-on-reset.

MIPI_HSI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
220_8000	HSI Control Register (MIPI_HSI_CTRL)	32	R/W	C800_0000h	43.5.1/3742
220_8004	HSI Tx Config Register (MIPI_HSI_TX_CONF)	32	R/W	0000_0000h	43.5.2/3744
220_8008	HSI Rx Config Register (MIPI_HSI_RX_CONF)	32	R/W	0000_0000h	43.5.3/3746
220_800C	HSI Capability Register (MIPI_HSI_CAP)	32	R	0007_1FFFh	43.5.4/3749
220_8010	HSI Tx Water Mark Level 0 Register (MIPI_HSI_TX_WML0)	32	R/W	0000_0000h	43.5.5/3750
220_8014	HSI Tx Water Mark Level 1 Register (MIPI_HSI_TX_TML1)	32	R/W	0000_0000h	43.5.6/3752
220_8018	HSI Tx Arbiter Priority 0 Register (MIPI_HSI_TX_ARB_PRI0)	32	R/W	0000_0000h	43.5.7/3754
220_801C	HSI Tx Arbiter Priority 1 Register (MIPI_HSI_TX_ARB_PRI1)	32	R/W	0000_0000h	43.5.8/3756
220_8020	HSI Line Status Register (MIPI_HSI_LINE_ST)	32	R	0000_0000h	43.5.9/3759
220_8024	HSI ID Bits Register (MIPI_HSI_ID_BIT)	32	R/W	0000_0000h	43.5.10/ 3761
220_8028	Tx and Rx Fif0 Threshold Configuration Register (MIPI_HSI_FIFO_THR_CONF)	32	R/W	0000_0000h	43.5.11/ 3762
220_802C	Tx and Rx Channel Soft Reset Register (MIPI_HSI_CH_SFTRST)	32	W	0000_0000h	43.5.12/ 3765
220_8030	HSI Interrupt Status Register (MIPI_HSI_IRQSTAT)	32	R/W	0000_0000h	43.5.13/ 3767
220_8034	HSI Interrupt Status Enable Register (MIPI_HSI_IRQSTAT_EN)	32	R/W	0000_0000h	43.5.14/ 3769
220_8038	HSI Interrupt Signal Enable Register (MIPI_HSI_IRQSIG_EN)	32	R/W	0000_0000h	43.5.15/ 3771
220_803C	HSI FIFO Threshold Interrupt Status Register (MIPI_HSI_FIFO_THR_IRQSTAT)	32	R	FFFF_0000h	43.5.16/ 3773
220_8040	HSI FIFO Threshold Interrupt Status Enable Register (MIPI_HSI_FIFO_THR_IRQSTAT_EN)	32	R/W	0000_0000h	43.5.17/ 3776

Table continues on the next page...

MIPI_HSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
220_8044	HSI FIFO Threshold Interrupt Signal Enable Register (MIPI_HSI_FIFO_THR_IRQSIG_EN)	32	R/W	0000_0000h	43.5.18/ 3779
220_8050	Tx Channel n Data Port Register (MIPI_HSI_TX_CH0_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_8054	Tx Channel n Data Port Register (MIPI_HSI_TX_CH1_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_8058	Tx Channel n Data Port Register (MIPI_HSI_TX_CH2_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_805C	Tx Channel n Data Port Register (MIPI_HSI_TX_CH3_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_8060	Tx Channel n Data Port Register (MIPI_HSI_TX_CH4_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_8064	Tx Channel n Data Port Register (MIPI_HSI_TX_CH5_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_8068	Tx Channel n Data Port Register (MIPI_HSI_TX_CH6_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_806C	Tx Channel n Data Port Register (MIPI_HSI_TX_CH7_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_8070	Tx Channel n Data Port Register (MIPI_HSI_TX_CH8_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_8074	Tx Channel n Data Port Register (MIPI_HSI_TX_CH9_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_8078	Tx Channel n Data Port Register (MIPI_HSI_TX_CH10_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_807C	Tx Channel n Data Port Register (MIPI_HSI_TX_CH11_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_8080	Tx Channel n Data Port Register (MIPI_HSI_TX_CH12_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_8084	Tx Channel n Data Port Register (MIPI_HSI_TX_CH13_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_8088	Tx Channel n Data Port Register (MIPI_HSI_TX_CH14_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_808C	Tx Channel n Data Port Register (MIPI_HSI_TX_CH15_DP)	32	R/W	0000_0000h	43.5.19/ 3782
220_8090	Rx Channel n Data Port Register (MIPI_HSI_RX_CH0_DP)	32	R/W	0000_0000h	43.5.20/ 3783
220_8094	Rx Channel n Data Port Register (MIPI_HSI_RX_CH1_DP)	32	R/W	0000_0000h	43.5.20/ 3783
220_8098	Rx Channel n Data Port Register (MIPI_HSI_RX_CH2_DP)	32	R/W	0000_0000h	43.5.20/ 3783
220_809C	Rx Channel n Data Port Register (MIPI_HSI_RX_CH3_DP)	32	R/W	0000_0000h	43.5.20/ 3783
220_80A0	Rx Channel n Data Port Register (MIPI_HSI_RX_CH4_DP)	32	R/W	0000_0000h	43.5.20/ 3783

Table continues on the next page...

MIPI_HSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
220_80A4	Rx Channel n Data Port Register (MIPI_HSI_RX_CH5_DP)	32	R/W	0000_0000h	43.5.20/ 3783
220_80A8	Rx Channel n Data Port Register (MIPI_HSI_RX_CH6_DP)	32	R/W	0000_0000h	43.5.20/ 3783
220_80AC	Rx Channel n Data Port Register (MIPI_HSI_RX_CH7_DP)	32	R/W	0000_0000h	43.5.20/ 3783
220_80B0	Rx Channel n Data Port Register (MIPI_HSI_RX_CH8_DP)	32	R/W	0000_0000h	43.5.20/ 3783
220_80B4	Rx Channel n Data Port Register (MIPI_HSI_RX_CH9_DP)	32	R/W	0000_0000h	43.5.20/ 3783
220_80B8	Rx Channel n Data Port Register (MIPI_HSI_RX_CH10_DP)	32	R/W	0000_0000h	43.5.20/ 3783
220_80BC	Rx Channel n Data Port Register (MIPI_HSI_RX_CH11_DP)	32	R/W	0000_0000h	43.5.20/ 3783
220_80C0	Rx Channel n Data Port Register (MIPI_HSI_RX_CH12_DP)	32	R/W	0000_0000h	43.5.20/ 3783
220_80C4	Rx Channel n Data Port Register (MIPI_HSI_RX_CH13_DP)	32	R/W	0000_0000h	43.5.20/ 3783
220_80C8	Rx Channel n Data Port Register (MIPI_HSI_RX_CH14_DP)	32	R/W	0000_0000h	43.5.20/ 3783
220_80CC	Rx Channel n Data Port Register (MIPI_HSI_RX_CH15_DP)	32	R/W	0000_0000h	43.5.20/ 3783
220_80D0	HSI Error Interrupt Status Register (MIPI_HSI_ERR_IRQSTAT)	32	R	0000_0000h	43.5.21/ 3784
220_80D4	HSI Error Interrupt Status Enable Register (MIPI_HSI_ERR_IRQSTAT_EN)	32	R/W	0000_0000h	43.5.22/ 3787
220_80D8	HSI Error Interrupt Signal Enable Register (MIPI_HSI_ERR_IRQSIG_EN)	32	R/W	0000_0000h	43.5.23/ 3789
220_80DC	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA0_CONF)	32	R/W	0000_0000h	43.5.24/ 3791
220_80E0	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA1_CONF)	32	R/W	0000_0000h	43.5.24/ 3791
220_80E4	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA2_CONF)	32	R/W	0000_0000h	43.5.24/ 3791
220_80E8	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA3_CONF)	32	R/W	0000_0000h	43.5.24/ 3791
220_80EC	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA4_CONF)	32	R/W	0000_0000h	43.5.24/ 3791
220_80F0	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA5_CONF)	32	R/W	0000_0000h	43.5.24/ 3791
220_80F4	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA6_CONF)	32	R/W	0000_0000h	43.5.24/ 3791
220_80F8	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA7_CONF)	32	R/W	0000_0000h	43.5.24/ 3791

Table continues on the next page...

MIPI_HSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
220_80FC	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA8_CONF)	32	R/W	0000_0000h	43.5.24/ 3791
220_8100	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA9_CONF)	32	R/W	0000_0000h	43.5.24/ 3791
220_8104	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA10_CONF)	32	R/W	0000_0000h	43.5.24/ 3791
220_8108	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA11_CONF)	32	R/W	0000_0000h	43.5.24/ 3791
220_810C	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA12_CONF)	32	R/W	0000_0000h	43.5.24/ 3791
220_8110	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA13_CONF)	32	R/W	0000_0000h	43.5.24/ 3791
220_8114	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA14_CONF)	32	R/W	0000_0000h	43.5.24/ 3791
220_8118	Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA15_CONF)	32	R/W	0000_0000h	43.5.24/ 3791
220_811C	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA0_CONF)	32	R/W	0000_0000h	43.5.25/ 3792
220_8120	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA1_CONF)	32	R/W	0000_0000h	43.5.25/ 3792
220_8124	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA2_CONF)	32	R/W	0000_0000h	43.5.25/ 3792
220_8128	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA3_CONF)	32	R/W	0000_0000h	43.5.25/ 3792
220_812C	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA4_CONF)	32	R/W	0000_0000h	43.5.25/ 3792
220_8130	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA5_CONF)	32	R/W	0000_0000h	43.5.25/ 3792
220_8134	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA6_CONF)	32	R/W	0000_0000h	43.5.25/ 3792
220_8138	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA7_CONF)	32	R/W	0000_0000h	43.5.25/ 3792
220_813C	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA8_CONF)	32	R/W	0000_0000h	43.5.25/ 3792
220_8140	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA9_CONF)	32	R/W	0000_0000h	43.5.25/ 3792
220_8144	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA10_CONF)	32	R/W	0000_0000h	43.5.25/ 3792
220_8148	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA11_CONF)	32	R/W	0000_0000h	43.5.25/ 3792
220_814C	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA12_CONF)	32	R/W	0000_0000h	43.5.25/ 3792
220_8150	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA13_CONF)	32	R/W	0000_0000h	43.5.25/ 3792

Table continues on the next page...

MIPI_HSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
220_8154	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA14_CONF)	32	R/W	0000_0000h	43.5.25/ 3792
220_8158	Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA15_CONF)	32	R/W	0000_0000h	43.5.25/ 3792
220_815C	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA0_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_8160	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA1_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_8164	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA2_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_8168	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA3_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_816C	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA4_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_8170	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA5_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_8174	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA6_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_8178	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA7_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_817C	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA8_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_8180	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA9_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_8184	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA10_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_8188	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA11_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_818C	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA12_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_8190	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA13_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_8194	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA14_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_8198	Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA15_STA_ADDR)	32	R/W	0000_0000h	43.5.26/ 3793
220_819C	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA0_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793
220_81A0	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA1_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793
220_81A4	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA2_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793
220_81A8	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA3_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793

Table continues on the next page...

MIPI_HSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
220_81AC	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA4_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793
220_81B0	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA5_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793
220_81B4	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA6_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793
220_81B8	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA7_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793
220_81BC	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA8_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793
220_81C0	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA9_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793
220_81C4	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA10_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793
220_81C8	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA11_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793
220_81CC	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA12_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793
220_81D0	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA13_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793
220_81D4	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA14_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793
220_81D8	Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA15_STA_ADDR)	32	R/W	0000_0000h	43.5.27/ 3793
220_81DC	DMA Interrupt Status Register (MIPI_HSI_DMA_IRQSTAT)	32	R	0000_0000h	43.5.28/ 3794
220_81E0	DMA Interrupt Enable Register (MIPI_HSI_DMA_IRQSTAT_EN)	32	R/W	0000_0000h	43.5.29/ 3796
220_81E4	DMA Interrupt Status Signal Enable Register (MIPI_HSI_DMA_IRQSIG_EN)	32	R/W	0000_0000h	43.5.30/ 3798
220_81E8	DMA Error Interrupt Status Register (MIPI_HSI_DMA_ERR_IRQSTAT)	32	R	0000_0000h	43.5.31/ 3801
220_81EC	DMA Error Interrupt Enable Register (MIPI_HSI_DMA_ERR_IRQSTAT_EN)	32	R/W	0000_0000h	43.5.32/ 3803
220_81F0	DMA Error Interrupt Signal Enable Register (MIPI_HSI_DMA_ERR_IRQSIG_EN)	32	R/W	0000_0000h	43.5.33/ 3805
220_81F4	DMA Single Request Enable Register (MIPI_HSI_DMA_SINGLE_REQ_EN)	32	R	0000_0000h	43.5.34/ 3808
220_8200	Tx Fifo Size Configuration Register 0 (MIPI_HSI_TX_FIFO_SIZE_CONF0)	32	R/W	5555_5555h	43.5.35/ 3810
220_8204	Tx Fifo Size Configuration Register 1 (MIPI_HSI_TX_FIFO_SIZE_CONF1)	32	R/W	5555_5555h	43.5.36/ 3813
220_8208	Rx Fifo Size Configuration Register 0 (MIPI_HSI_RX_FIFO_SIZE_CONF0)	32	R/W	5555_5555h	43.5.37/ 3816

Table continues on the next page...

MIPI_HSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
220_820C	Rx Fifo Size Configuration Register 1 (MIPI_HSI_RX_FIFO_SIZE_CONF1)	32	R/W	5555_5555h	43.5.38/ 3819
220_8210	Tx Fifo Status Register (MIPI_HSI_TX_FIFO_STAT)	32	R	5555_5555h	43.5.39/ 3822
220_8214	Rx Fifo Status Register (MIPI_HSI_RX_FIFO_STAT)	32	R	5555_5555h	43.5.40/ 3824
220_8228	AHB Master Config Register (MIPI_HSI_AHB_MASTER_CONF)	32	R/W	0000_0180h	43.5.41/ 3826
220_822C	TX Break Length Register (MIPI_HSI_TX_BREAK_LEN)	32	R/W	0000_0025h	43.5.42/ 3827

43.5.1 HSI Control Register (MIPI_HSI_CTRL)

This register contains module soft reset, clock gating, clock divisor and so on.

Address: 220_8000h base + 0h offset = 220_8000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SFTRST	CLKGATE	Reserved		DMA_ DISABLE	RX_DLY_SEL			RX_FRAME_BRST_CNT							
Reset	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		RX_TAIL_ BIT_CNT		DATA_TIMEOUT_CNT				Reserved			TX_BREAK	TX_CLK_DIVISOR			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_CTRL field descriptions

Field	Description
31 SFTRST	Set this bit to zero to enable normal HSI operation. Set this bit to one (default) to disable clocking with the HSI and hold it in its reset (lowest power) state. This bit can be turned on and then off to reset the HSI block to its default state.
30 CLKGATE	This bit must be set to zero for normal operation. When set to one it gates off the clocks to the block.

Table continues on the next page...

MIPI_HSI_CTRL field descriptions (continued)

Field	Description
29–28 Reserved	This field is reserved. Reserved, always set to zero.
27 DMA_DISABLE	This bit must be set to zero for any DMA operation. When set to one it disables all the DMA channels.
26–24 RX_DLY_SEL	These values denote the tap delay values for reception of data and flag. 000 0ns ; 001 1ns ; 010 2ns ; 011 3ns ; 100 4ns ; 101 5ns ; 110 6ns ; 111 7ns ;
23–16 RX_FRAME_ BRST_CNT	This value is to limit the continuous Frame transmission count in Pipelined Data flow. The Receiver Frame Burst counter shall be able to support up to 256 frames of continuous transfer. 7'h00 256 frames transmission count is set. 7'h01 1 frames transmission count is set. 7'h02 2 frames transmission count is set. 7'hff 255 frames transmission count is set.
15–14 Reserved	This field is reserved. Reserved.
13–12 RX_TAIL_BIT_ CNT	The value determines the length of the Tailing bit counter. The receiver shall start Receiver Tailing bit counter after the nth frame programmed in Rx Frame Burst counter is received. The receiver shall then drive ready to logic one if the receiver Tailing-bit counter has completed with no errors detected, and the receiver has enough room for at least one new frame. 00 800-> tx_refclk 01 400-> tx_refclk 10 200-> tx_refclk 11 100-> tx_refclk
11–8 DATA_ TIMEOUT_CNT	This value determines the interval by which DATA timeouts are detected. This data timeout counter logic is used only for Receive operations. The counter should start counting when data in any of the RX channel fifo is less than the threshold value and resets to zero when there is a threshold reached interrupt from any of the RX buffers. The counter value should be zero, when RX fifo is empty. An interrupt will be asserted to the host driver, when the counter value reaches the data timeout counter value. 1110 HSI Tx Clock x 2 ^ 27 0001 HSI Tx Clock x 2 ^ 14 0000 HSI Tx Clock x 2 ^ 13

Table continues on the next page...

MIPI_HSI_CTRL field descriptions (continued)

Field	Description
7–5 Reserved	This field is reserved. Reserved.
4 TX_BREAK	Setting this bit to one triggers a transmission break at HSI Tx. Once this bit is set to one, the HSI controller will send a series of zeros on HSI_TX_DATA port according to the tx break count. It will be automatically cleared, when the send is finished.
TX_CLK_ DIVISOR	This register holds the divisor of the base clock (tx_refclk) frequency for HSI Tx clock (internal clock used to drive Transmitter interface). 1000 tx_refclk divided by 256 0111 tx_refclk divided by 128 0110 tx_refclk divided by 64 0101 tx_refclkdivided by 32 0100 tx_refclk divided by 16 0011 tx_refclk divided by 8 0010 tx_refclk divided by 4 0001 tx_refclk divided by 2 0000 tx_refclk divided by 1

43.5.2 HSI Tx Config Register (MIPI_HSI_TX_CONF)

This register contains the configurations of tx channel enable/disable, tx wakeup and tx trans mode.

Address: 220_8000h base + 4h offset = 220_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CH15_EN	CH14_EN	CH13_EN	CH12_EN	CH11_EN	CH10_EN	CH9_EN	CH8_EN	CH7_EN	CH6_EN	CH5_EN	CH4_EN	CH3_EN	CH2_EN	CH1_EN	CH0_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				TIMEOUT_CNT				Reserved						WAKEUP	TRANS_MODE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_TX_CONF field descriptions

Field	Description
31 CH15_EN	1 Tx Ch15 is Enabled. 0 Tx Ch15 is Disabled.
30 CH14_EN	1 Tx Ch14 is Enabled. 0 Tx Ch14 is Disabled.
29 CH13_EN	1 Tx Ch13 is Enabled. 0 Tx Ch13 is Disabled.
28 CH12_EN	1 Tx Ch12 is Enabled. 0 Tx Ch12 is Disabled.
27 CH11_EN	1 Tx Ch11 is Enabled. 0 Tx Ch11 is Disabled.
26 CH10_EN	1 Tx Ch10 is Enabled. 0 Tx Ch10 is Disabled.
25 CH9_EN	1 Tx Ch9 is Enabled. 0 Tx Ch9 is Disabled.
24 CH8_EN	1 Tx Ch8 is Enabled. 0 Tx Ch8 is Disabled.
23 CH7_EN	1 Tx Ch7 is Enabled. 0 Tx Ch7 is Disabled.
22 CH6_EN	1 Tx Ch6 is Enabled. 0 Tx Ch6 is Disabled.
21 CH5_EN	1 Tx Ch5 is Enabled. 0 Tx Ch5 is Disabled.
20 CH4_EN	1 Tx Ch4 is Enabled. 0 Tx Ch4 is Disabled.
19 CH3_EN	1 Tx Ch3 is Enabled. 0 Tx Ch3 is Disabled.
18 CH2_EN	1 Tx Ch2 is Enabled. 0 Tx Ch2 is Disabled.
17 CH1_EN	1 Tx Ch1 is Enabled. 0 Tx Ch1 is Disabled.
16 CH0_EN	1 Tx Ch0 is Enabled. 0 Tx Ch0 is Disabled.
15–12 Reserved	This field is reserved. Reserved.
11–8 TIMEOUT_CNT	0000 tx timeout value 2^14 tx_refclk 0001 tx timeout value 2^15 tx_refclk 0010 tx timeout value 2^16 tx_refclk 0011 tx timeout value 2^17 tx_refclk 1110 tx timeout value 2^28 tx_refclk 1111 tx timeout value 2^29 tx_refclk
7–2 Reserved	This field is reserved. Reserved.
1 WAKEUP	When this bit gets set to one, HSI transmitter sends HSI_TX_WAKE signal to Rx of other device. For a transmit operation this bit should be one.

Table continues on the next page...

MIPI_HSI_TX_CONF field descriptions (continued)

Field	Description
	0 Transmitter is in Sleep State 1 Transmitter is in Wakeup State.
0 TRANS_MODE	0 Stream Transmission Mode 1 Frame Transmission Mode

43.5.3 HSI Rx Config Register (MIPI_HSI_RX_CONF)

This register contains the configurations of Rx channel enable/disable, rx wakeup and rx trans mode, rx data flow.

Address: 220_8000h base + 8h offset = 220_8008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CH15_EN	CH14_EN	CH13_EN	CH12_EN	CH11_EN	CH10_EN	CH9_EN	CH8_EN	CH7_EN	CH6_EN	CH5_EN	CH4_EN	CH3_EN	CH2_EN	CH1_EN	CH0_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	TIMEOUT_CNT							Reserved			TAIL_BIT_CNT_EN	REC_MODE	WAKE	DATA_FLOW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_RX_CONF field descriptions

Field	Description
31 CH15_EN	1 Rx Ch15 is Enabled. 0 Rx Ch15 is Disabled.
30 CH14_EN	1 Rx Ch14 is Enabled. 0 Rx Ch14 is Disabled.

Table continues on the next page...

MIPI_HSI_RX_CONF field descriptions (continued)

Field	Description
29 CH13_EN	1 Rx Ch13 is Enabled. 0 Rx Ch13 is Disabled.
28 CH12_EN	1 Rx Ch12 is Enabled. 0 Rx Ch12 is Disabled.
27 CH11_EN	1 Rx Ch11 is Enabled. 0 Rx Ch11 is Disabled.
26 CH10_EN	1 Rx Ch10 is Enabled. 0 Rx Ch10 is Disabled.
25 CH9_EN	1 Rx Ch9 is Enabled. 0 Rx Ch9 is Disabled.
24 CH8_EN	1 Rx Ch8 is Enabled. 0 Rx Ch8 is Disabled.
23 CH7_EN	1 Rx Ch7 is Enabled. 0 Rx Ch7 is Disabled.
22 CH6_EN	1 Rx Ch6 is Enabled. 0 Rx Ch6 is Disabled.
21 CH5_EN	1 Rx Ch5 is Enabled. 0 Rx Ch5 is Disabled.
20 CH4_EN	1 Rx Ch4 is Enabled. 0 Rx Ch4 is Disabled.
19 CH3_EN	1 Rx Ch3 is Enabled. 0 Rx Ch3 is Disabled.
18 CH2_EN	1 Rx Ch2 is Enabled. 0 Rx Ch2 is Disabled.
17 CH1_EN	1 Rx Ch1 is Enabled. 0 Rx Ch1 is Disabled.
16 CH0_EN	1 Rx Ch0 is Enabled. 0 Rx Ch0 is Disabled.
15 Reserved	This field is reserved. Reserved.
14–8 TIMEOUT_CNT	Receive Frame Timeout Counter: The counter shall be started when the first bit of the Frame has been found. The counter shall be stopped once the receiver has received the correct number of bits for a Frame. If the counter expires before Frame reception is completed, the receiver will signal to the protocol layer that it has found an incomplete Frame and asserts Rx Error Interrupt. 7'h0 14800 ---> tx_refclk 7'h1 16400 ---> tx_refclk 7'h2 18000 ---> tx_refclk 7'h4 19600 ---> tx_refclk 7'h8 21200 ---> tx_refclk 7'h10 22800 ---> tx_refclk 7'h20 24400 ---> tx_refclk

Table continues on the next page...

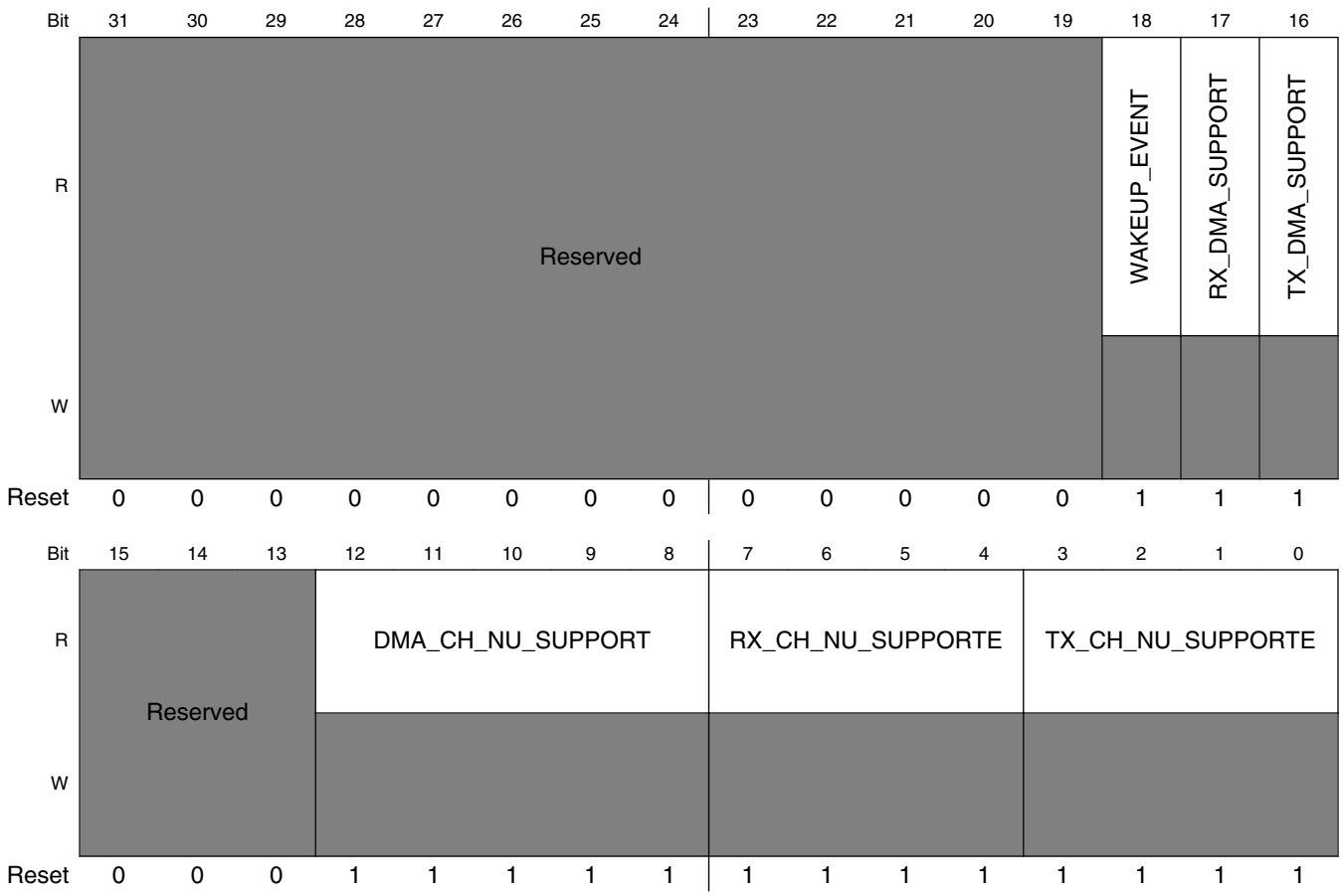
MIPI_HSI_RX_CONF field descriptions (continued)

Field	Description
	7'h40 26000 ---> tx_refclk
7–5 Reserved	This field is reserved. Reserved.
4 TAIL_BIT_CNT_EN	0 Tailing bit counter disable 1 Tailing bit counter Enable
3 REC_MODE	0 Stream Receive Mode 1 Frame Receive Mode
2 WAKE	0 Receiver is in Sleep State 1 Receiver is in Wakeup State
DATA_FLOW	00 Synchronized Data Flow 01 Pipelined Data Flow 10 Receiver Real-time Data Flow 11 Reserved

43.5.4 HSI Capability Register (MIPI_HSI_CAP)

This register contains the HSI controller Capability information.

Address: 220_8000h base + Ch offset = 220_800Ch



MIPI_HSI_CAP field descriptions

Field	Description
31–19 Reserved	This field is reserved. Reserved.
18 WAKEUP_ EVENT	0 Wakeup Event is supported 1 Wakeup Event is not supported.
17 RX_DMA_ SUPPORT	1 DMA is supported. 0 Not supported
16 TX_DMA_ SUPPORT	1 DMA is supported. 0 Not supported

Table continues on the next page...

MIPI_HSI_CAP field descriptions (continued)

Field	Description
15–13 Reserved	This field is reserved. Reserved.
12–8 DMA_CH_NU_ SUPPORT	00000 1 DMA supported 00001 2 DMA supported 00010 3 DMA supported 00011 4 DMA supported 00100 5 DMA supported 00101 6 DMA supported 11110 31 DMA supported 11111 32 DMA supported
7–4 RX_CH_NU_ SUPPORTE	0000 1 Rx channel supported 0001 2 Rx channels supported 1111 16 Rx channels supported
TX_CH_NU_ SUPPORTE	0000 1 Tx channel supported 0001 2 Tx channels supported 1111 16 Tx channels supported

43.5.5 HSI Tx Water Mark Level 0 Register (MIPI_HSI_TX_WML0)

This register contains HSI controller Tx channel Water Mark Level information.

Address: 220_8000h base + 10h offset = 220_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH15				CH14				CH13				CH12				CH11				CH10				CH9				CH8			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_TX_WML0 field descriptions

Field	Description
31–28 CH15	This value denotes the WML of Tx Channel 15. When > 1010 Reserved 0000 1 0001 2 0010 4 1000 256 1001 512 1010 1024
27–24 CH14	This value denotes the WML of Tx Channel 14. When > 1010 reserved

Table continues on the next page...

MIPI_HSI_TX_WML0 field descriptions (continued)

Field	Description
	0000 1 0001 2 0010 4 1000 256 1001 512 1010 1024
23–20 CH13	This value denotes the WML of Tx Channel 13. When > 1010 reserved 0000 1 0001 2 0010 4 1000 256 1001 512 1010 1024
19–16 CH12	This value denotes the WML of Tx Channel 12. When > 1010 reserved 0000 1 0001 2 0010 4 1000 256 1001 512 1010 1024
15–12 CH11	This value denotes the WML of Tx Channel 11. When > 1010 reserved 0000 1 0001 2 0010 4 1000 256 1001 512 1010 1024
11–8 CH10	This value denotes the WML of Tx Channel 10. When > 1010 reserved 0000 1 0001 2 0010 4 1000 256 1001 512 1010 1024
7–4 CH9	This value denotes the WML of Tx Channel 9. When > 1010 reserved 0000 1 0001 2

Table continues on the next page...

MIPI_HSI_TX_WML0 field descriptions (continued)

Field	Description
	0010 4 1000 256 1001 512 1010 1024
CH8	This value denotes the WML of Tx Channel 8. When > 1010 reserved 0000 1 0001 2 0010 4 1000 256 1001 512 1010 1024

43.5.6 HSI Tx Water Mark Level 1 Register (MIPI_HSI_TX_TML1)

This register contains HSI controller Tx channel Water Mark Level information.

This register contains HSI controller Tx channel bandwidth information.

Address: 220_8000h base + 14h offset = 220_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH7				CH6				CH5				CH4				CH3				CH2				CH1				CH0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_TX_TML1 field descriptions

Field	Description
31–28 CH7	This value denotes the WML of Tx Channel 7. When > 1010 reserved 0000 1 0001 2 0010 4 1000 256 1001 512 1010 1024
27–24 CH6	This value denotes the WML of Tx Channel 6. When > 1010 reserved 0000 1 0001 2 0010 4 1000 256

Table continues on the next page...

MIPI_HSI_TX_TML1 field descriptions (continued)

Field	Description
	1001 512 1010 1024
23–20 CH5	This value denotes the WML of Tx Channel 5. When > 1010 reserved 0000 1 0001 2 0010 4 1000 256 1001 512 1010 1024
19–16 CH4	This value denotes the WML of Tx Channel 4. When > 1010 reserved 0000 1 0001 2 0010 4 1000 256 1001 512 1010 1024
15–12 CH3	This value denotes the WML of Tx Channel 3. When > 1010 reserved 0000 1 0001 2 0010 4 1000 256 1001 512 1010 1024
11–8 CH2	This value denotes the WML of Tx Channel 2. When > 1010 reserved 0000 1 0001 2 0010 4 1000 256 1001 512 1010 1024
7–4 CH1	This value denotes the WML of Tx Channel 1. When > 1010 reserved 0000 1 0001 2 0010 4 1000 256

Table continues on the next page...

MIPI_HSI_TX_TML1 field descriptions (continued)

Field	Description
	1001 512 1010 1024
CH0	This value denotes the WML of Tx Channel 0. When > 1010 reserved 0000 1 0001 2 0010 4 1000 256 1001 512 1010 1024

43.5.7 HSI Tx Arbiter Priority 0 Register (MIPI_HSI_TX_ARB_PRI0)

This is HSI Tx Arbiter Priority Register.

Address: 220_8000h base + 18h offset = 220_8018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH7				CH6				CH5				CH4				CH3				CH2				CH1				CH0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_TX_ARB_PRI0 field descriptions

Field	Description
31–28 CH7	This value denotes the priority of Tx Channel 7. When > 1010 reserved 0000 1st priority 0001 2nd priority 0010 3rd priority 1101 14th priority 1110 15th priority 1111 16th priority
27–24 CH6	This value denotes the priority of Tx Channel 6. When > 1010 reserved 0000 1st priority 0001 2nd priority 0010 3rd priority 1101 14th priority

Table continues on the next page...

MIPI_HSI_TX_ARB_PRI0 field descriptions (continued)

Field	Description
	1110 15th priority 1111 16th priority
23–20 CH5	This value denotes the priority of Tx Channel 5. When > 1010 reserved 0000 1st priority 0001 2nd priority 0010 3rd priority 1101 14th priority 1110 15th priority 1111 16th priority
19–16 CH4	This value denotes the priority of Tx Channel 4. When > 1010 reserved 0000 1st priority 0001 2nd priority 0010 3rd priority 1101 14th priority 1110 15th priority 1111 16th priority
15–12 CH3	This value denotes the priority of Tx Channel 3. When > 1010 reserved 0000 1st priority 0001 2nd priority 0010 3rd priority 1101 14th priority 1110 15th priority 1111 16th priority
11–8 CH2	This value denotes the priority of Tx Channel 2. When > 1010 reserved 0000 1st priority 0001 2nd priority 0010 3rd priority 1101 14th priority 1110 15th priority 1111 16th priority
7–4 CH1	This value denotes the priority of Tx Channel 1. When > 1010 reserved 0000 1st priority 0001 2nd priority 0010 3rd priority 1101 14th priority

Table continues on the next page...

MIPI_HSI_TX_ARB_PRI0 field descriptions (continued)

Field	Description
	1110 15th priority 1111 16th priority
CH0	This value denotes the priority of Tx Channel 0. When > 1010 reserved 0000 1st priority 0001 2nd priority 0010 3rd priority 1101 14th priority 1110 15th priority 1111 16th priority

43.5.8 HSI Tx Arbiter Priority 1 Register (MIPI_HSI_TX_ARB_PRI1)

Address: 220_8000h base + 1Ch offset = 220_801Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH15				CH14				CH13				CH12				CH11				CH10				CH9				CH8			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_TX_ARB_PRI1 field descriptions

Field	Description
31–28 CH15	This value denotes the priority of Tx Channel 15. When > 1010 Reserved 0000 1st priority 0001 2nd priority 0010 3rd priority 1101 14th priority 1110 15th priority 1111 16th priority
27–24 CH14	This value denotes the priority of Tx Channel 14. When > 1010 reserved 0000 1st priority 0001 2nd priority 0010 3rd priority 1101 14th priority 1110 15th priority 1111 16th priority
23–20 CH13	This value denotes the priority of Tx Channel 13. When > 1010 reserved

Table continues on the next page...

MIPI_HSI_TX_ARB_PRI1 field descriptions (continued)

Field	Description
	0000 1st priority 0001 2nd priority 0010 3rd priority 1101 14th priority 1110 15th priority 1111 16th priority
19–16 CH12	This value denotes the priority of Tx Channel 12. When > 1010 reserved 0000 1st priority 0001 2nd priority 0010 3rd priority 1101 14th priority 1110 15th priority 1111 16th priority
15–12 CH11	This value denotes the priority of Tx Channel 11. When > 1010 reserved 0000 1st priority 0001 2nd priority 0010 3rd priority 1101 14th priority 1110 15th priority 1111 16th priority
11–8 CH10	This value denotes the priority of Tx Channel 10. When > 1010 reserved 0000 1st priority 0001 2nd priority 0010 3rd priority 1101 14th priority 1110 15th priority 1111 16th priority
7–4 CH9	This value denotes the priority of Tx Channel 9. When > 1010 reserved 0000 1st priority 0001 2nd priority 0010 3rd priority 1101 14th priority 1110 15th priority 1111 16th priority
CH8	This value denotes the priority of Tx Channel 8. When > 1010 reserved 0000 1st priority 0001 2nd priority

Table continues on the next page...

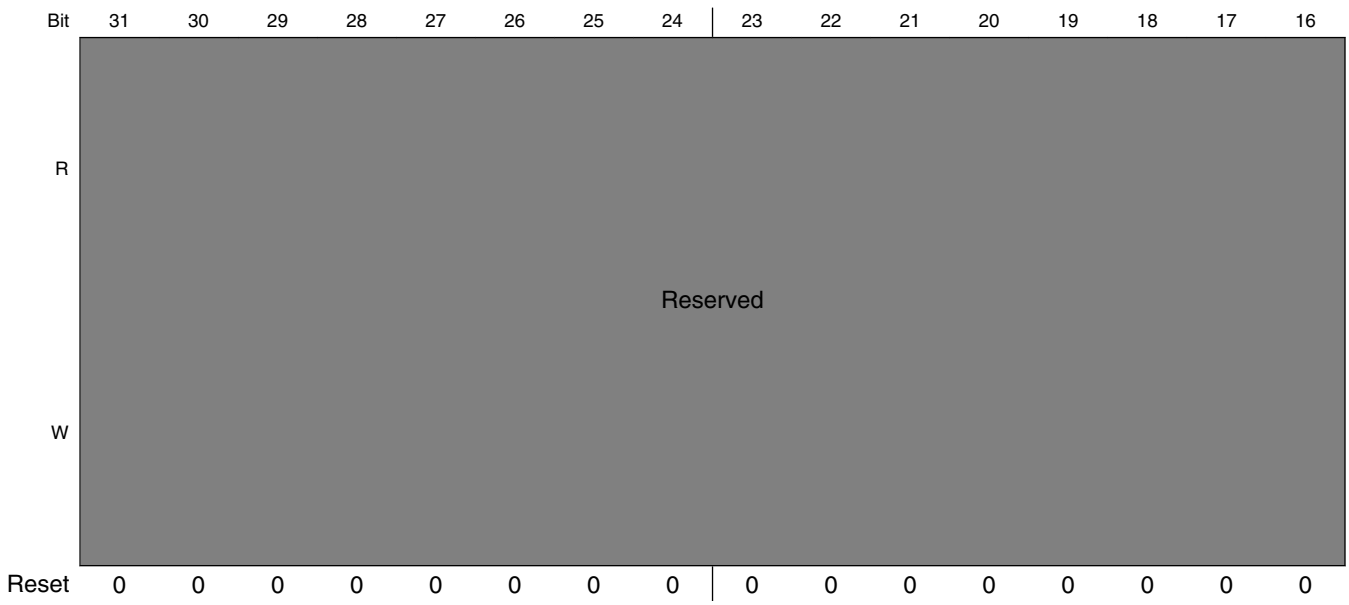
MIPI_HSI_TX_ARB_PRI1 field descriptions (continued)

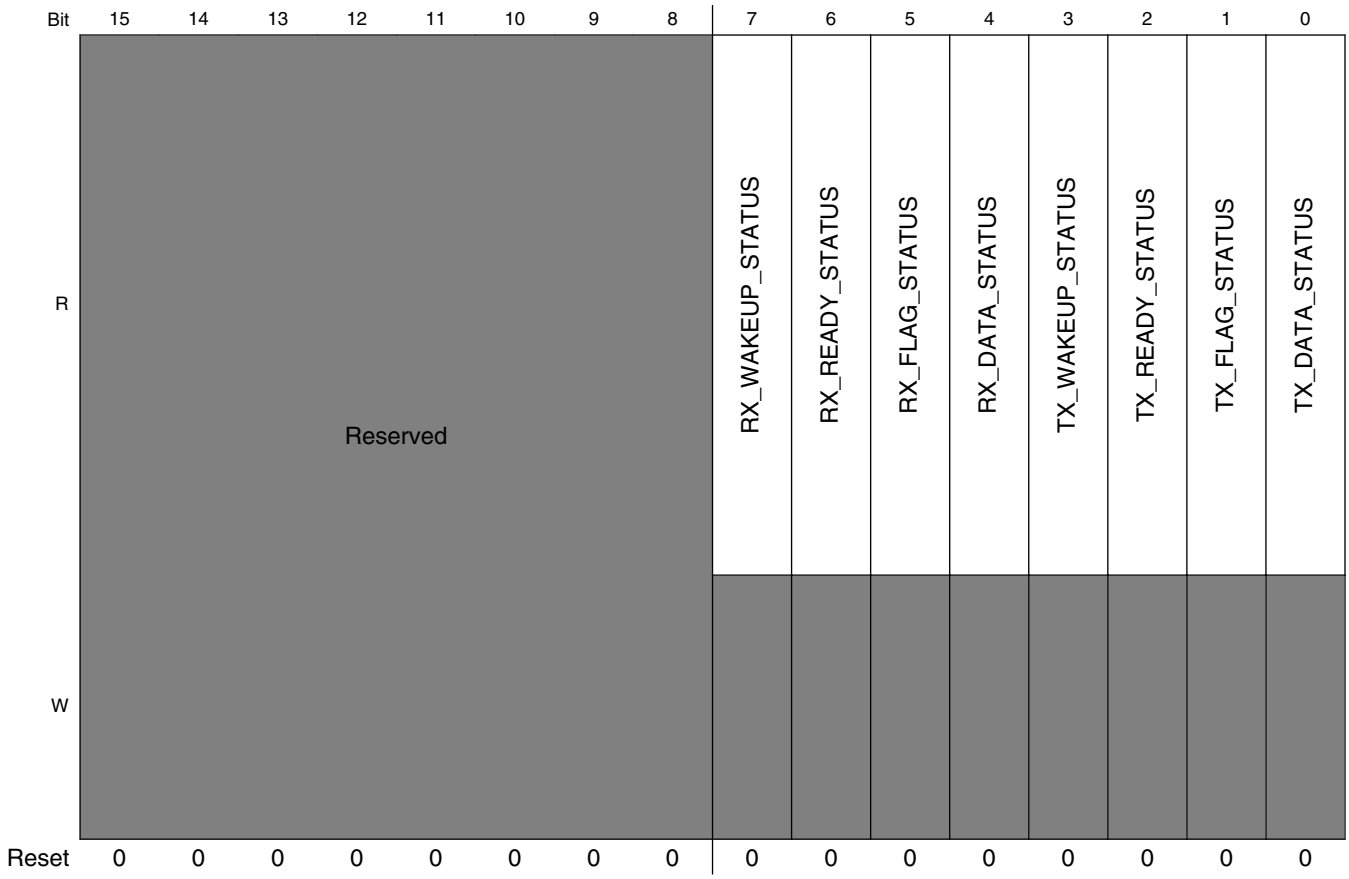
Field	Description
	0010 3rd priority 1101 14th priority 1110 15th priority 1111 16th priority

43.5.9 HSI Line Status Register (MIPI_HSI_LINE_ST)

This register contains the HSI controller line status for debug.

Address: 220_8000h base + 20h offset = 220_8020h





MIPI_HSI_LINE_ST field descriptions

Field	Description
31–8 Reserved	This field is reserved. Reserved, always set to zero.
7 RX_WAKEUP_ STATUS	This field reflects the rx_wake pin(only for debug).
6 RX_READY_ STATUS	This field reflects the rx_rdy pin(only for debug).
5 RX_FLAG_ STATUS	This field reflects the rx_flag pin(only for debug).
4 RX_DATA_ STATUS	This field reflects the rx_data pin(only for debug).
3 TX_WAKEUP_ STATUS	This field reflects the tx_wake pin(only for debug).
2 TX_READY_ STATUS	This field reflects the tx_ready pin(only for debug).

Table continues on the next page...

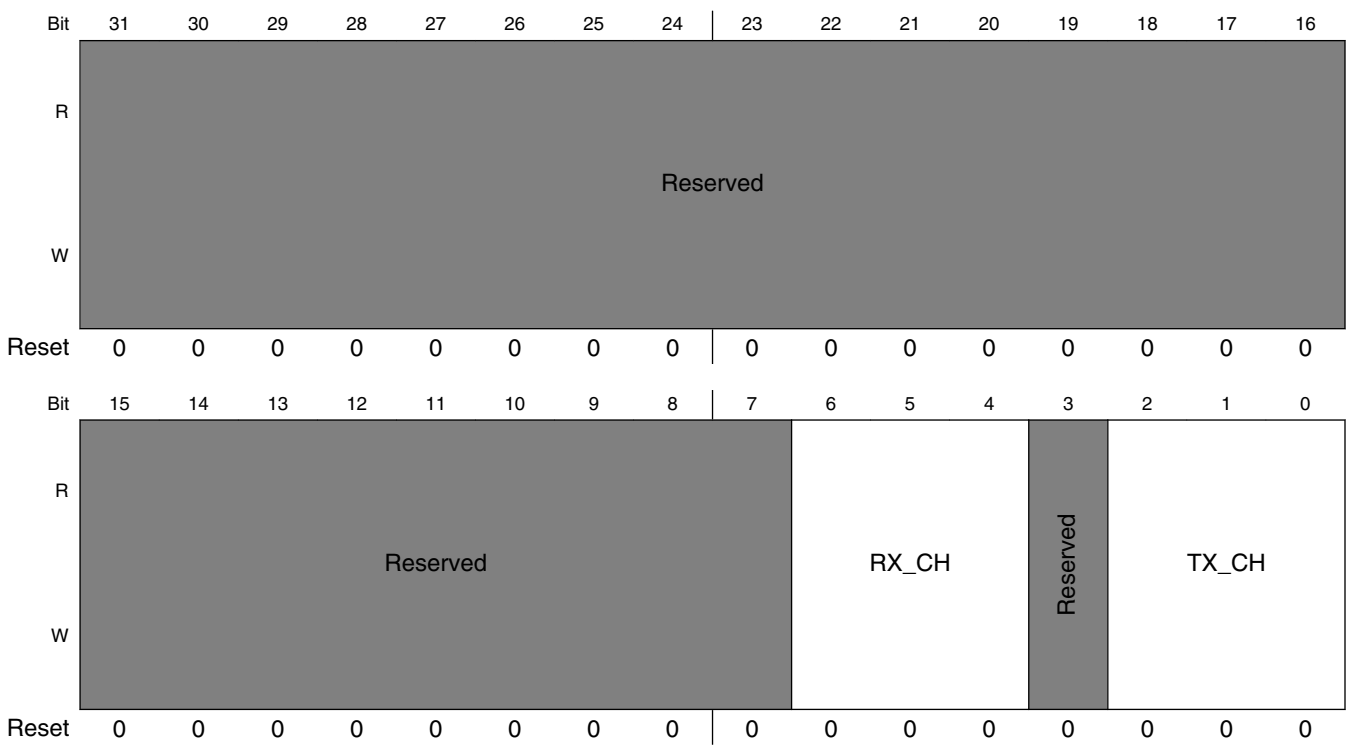
MIPI_HSI_LINE_ST field descriptions (continued)

Field	Description
1 TX_FLAG_ STATUS	This field reflects the tx_flag pin(only for debug).
0 TX_DATA_ STATUS	This field reflects the tx_data pin(only for debug).

43.5.10 HSI ID Bits Register (MIPI_HSI_ID_BIT)

This register contains the configurations of tx channel enable/disable, , tx wakeup and tx trans mode.

Address: 220_8000h base + 24h offset = 220_8024h



MIPI_HSI_ID_BIT field descriptions

Field	Description
31–7 Reserved	This field is reserved. Reserved, always set to zero.
6–4 RX_CH	This bit sets the number of channel ID bits per frame or stream for a Receive operation.

Table continues on the next page...

MIPI_HSI_ID_BIT field descriptions (continued)

Field	Description
	0 0 bit 1 1 bit 2 2 bits 3 3 bits 4 4 bits
3 Reserved	This field is reserved. Reserved, always set to zero.
TX_CH	This bit sets the number of channel ID bits per frame or stream for a transmit operation. 0 0 bit 1 1 bit 2 2 bits 3 3 bits 4 4 bits

43.5.11 Tx and Rx Fif0 Threshold Configuration Register (MIPI_HSI_FIFO_THR_CONF)

This register sets the threshold level for each Tx and Rx channel fifo

Address: 220_8000h base + 28h offset = 220_8028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TX_CH15	TX_CH14	TX_CH13	TX_CH12	TX_CH11	TX_CH10	TX_CH9	TX_CH8	TX_CH7	TX_CH6	TX_CH5	TX_CH4	TX_CH3	TX_CH2	TX_CH1	TX_CH0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX_CH15	RX_CH14	RX_CH13	RX_CH12	RX_CH11	RX_CH10	RX_CH9	RX_CH8	RX_CH7	RX_CH6	RX_CH5	RX_CH4	RX_CH3	RX_CH2	RX_CH1	RX_CH0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_FIFO_THR_CONF field descriptions

Field	Description
31 TX_CH15	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)

Table continues on the next page...

MIPI_HSI_FIFO_THR_CONF field descriptions (continued)

Field	Description
30 TX_CH14	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)
29 TX_CH13	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)
28 TX_CH12	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)
27 TX_CH11	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)
26 TX_CH10	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)
25 TX_CH9	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)
24 TX_CH8	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)
23 TX_CH7	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)
22 TX_CH6	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)
21 TX_CH5	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)
20 TX_CH4	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)
19 TX_CH3	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)
18 TX_CH2	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)
17 TX_CH1	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)
16 TX_CH0	0 Half Empty (fifo size / 2) 1 Almost Empty (fifo size / 4)
15 RX_CH15	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)
14 RX_CH14	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)
13 RX_CH13	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)
12 RX_CH12	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)
11 RX_CH11	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)
10 RX_CH10	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)

Table continues on the next page...

MIPI_HSI_FIFO_THR_CONF field descriptions (continued)

Field	Description
9 RX_CH9	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)
8 RX_CH8	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)
7 RX_CH7	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)
6 RX_CH6	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)
5 RX_CH5	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)
4 RX_CH4	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)
3 RX_CH3	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)
2 RX_CH2	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)
1 RX_CH1	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)
0 RX_CH0	0 Half Full (fifo size / 2) 1 Almost Full (3/4th of fifo size)

43.5.12 Tx and Rx Channel Soft Reset Register (MIPI_HSI_CH_SFTRST)

This register is used to reset each Tx and Rx Channel

Address: 220_8000h base + 2Ch offset = 220_802Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	TX_CH15	TX_CH14	TX_CH13	TX_CH12	TX_CH11	TX_CH10	TX_CH9	TX_CH8	TX_CH7	TX_CH6	TX_CH5	TX_CH4	TX_CH3	TX_CH2	TX_CH1	TX_CH0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	RX_CH15	RX_CH14	RX_CH13	RX_CH12	RX_CH11	RX_CH10	RX_CH9	RX_CH8	RX_CH7	RX_CH6	RX_CH5	RX_CH4	RX_CH3	RX_CH2	RX_CH1	RX_CH0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_CH_SFTRST field descriptions

Field	Description
31 TX_CH15	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 15 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
30 TX_CH14	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 14 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
29 TX_CH13	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 13 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
28 TX_CH12	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 12 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
27 TX_CH11	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 11 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
26 TX_CH10	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 10 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
25 TX_CH9	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 9 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.

Table continues on the next page...

MIPI_HSI_CH_SFTRST field descriptions (continued)

Field	Description
24 TX_CH8	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 8 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
23 TX_CH7	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 7 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
22 TX_CH6	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 6 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
21 TX_CH5	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 5 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
20 TX_CH4	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 4 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
19 TX_CH3	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 3 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
18 TX_CH2	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 2 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
17 TX_CH1	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 1 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
16 TX_CH0	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Tx Channel 0 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
15 RX_CH15	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 15 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
14 RX_CH14	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 14 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
13 RX_CH13	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 13 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
12 RX_CH12	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 12 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
11 RX_CH11	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 11 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
10 RX_CH10	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 10 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
9 RX_CH9	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 9 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
8 RX_CH8	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 8 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
7 RX_CH7	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 7 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
6 RX_CH6	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 6 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
5 RX_CH5	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 5 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
4 RX_CH4	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 4 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
3 RX_CH3	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 3 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
2 RX_CH2	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 2 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.

Table continues on the next page...

MIPI_HSI_CH_SFTRST field descriptions (continued)

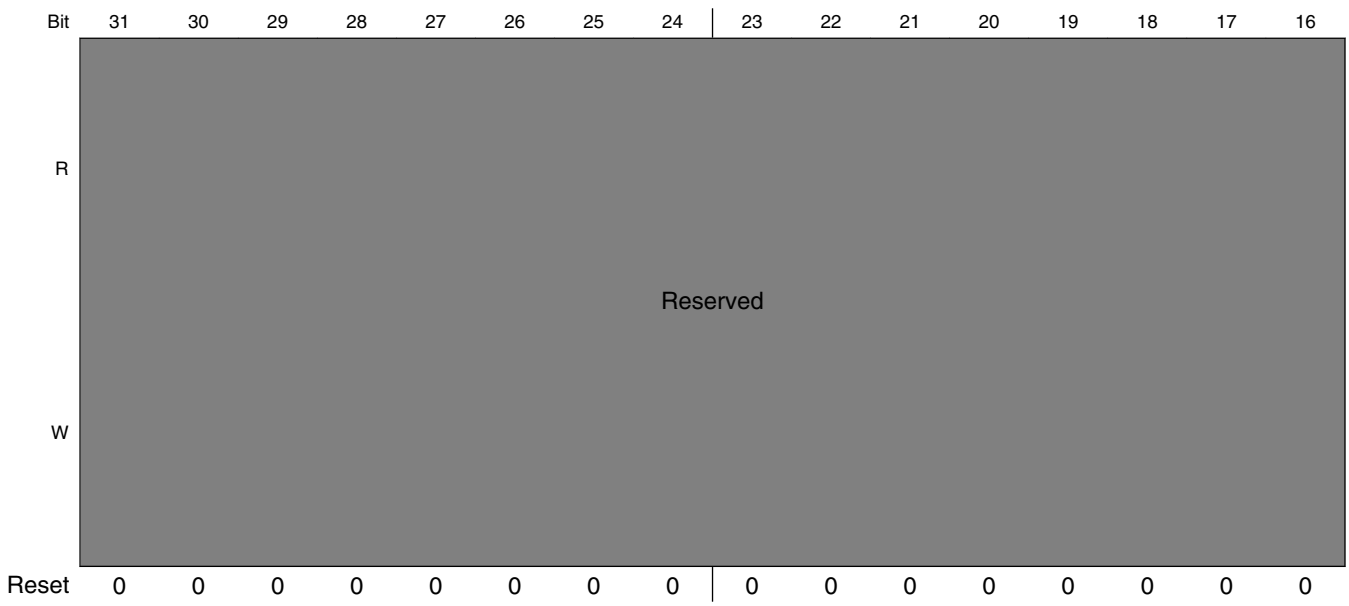
Field	Description
1 RX_CH1	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 1 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.
0 RX_CH0	Set this bit to zero to enable normal HSI operation. Set this bit to one to reset Rx Channel 0 (DMA and Fifo). When the reset operation complete, this bit will turn to zero automatically.

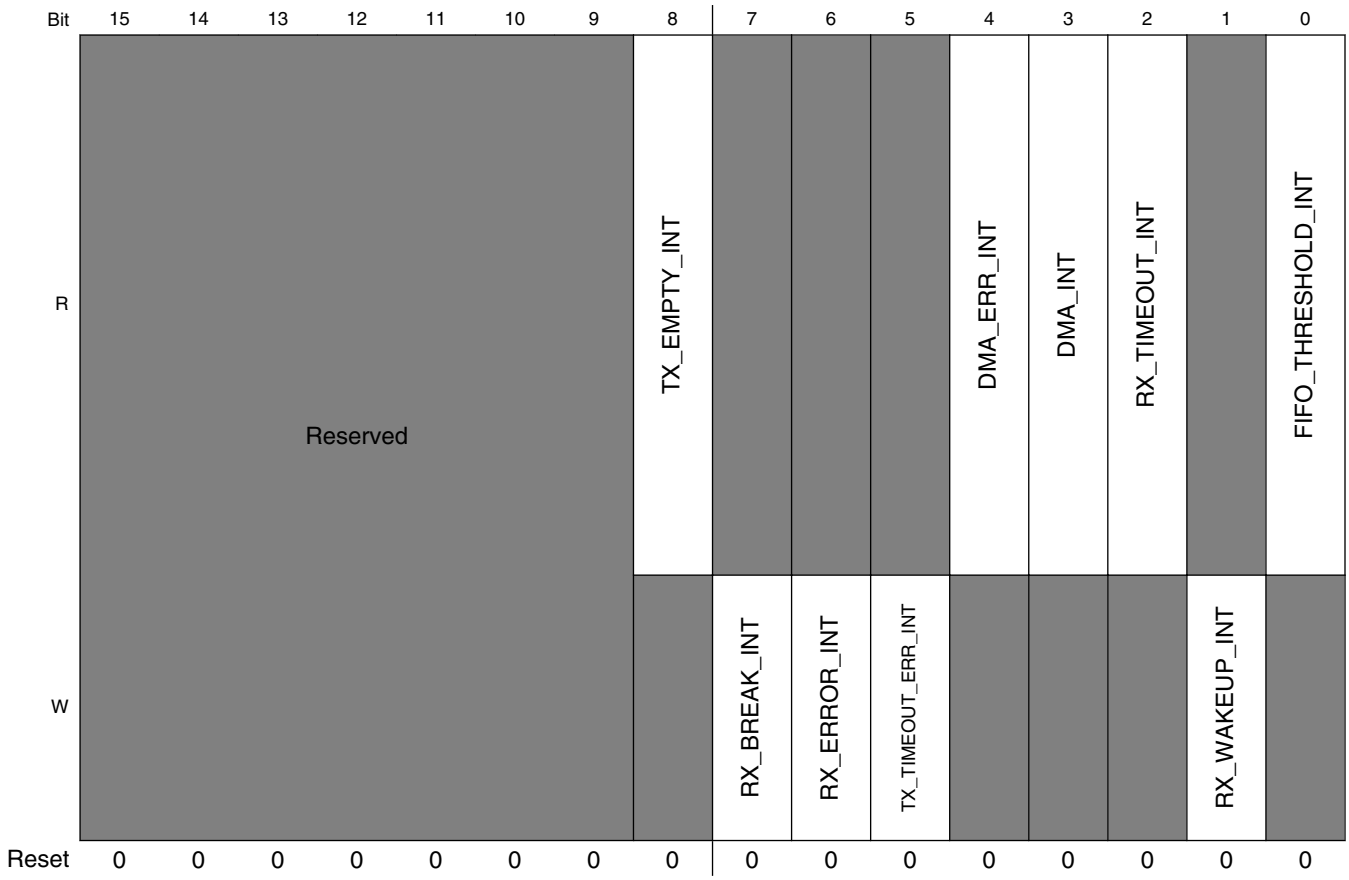
43.5.13 HSI Interrupt Status Register (MIPI_HSI_IRQSTAT)

This is HSI controller Interrupt Status Register.

This register contains the HSI controller Interrupt Status.

Address: 220_8000h base + 30h offset = 220_8030h





MIPI_HSI_IRQSTAT field descriptions

Field	Description
31–9 Reserved	This field is reserved. Reserved, always set to zero.
8 TX_EMPTY_INT	1 All tx channel empty and tx state IDLE Interrupt Status 0 not All tx channel empty and tx state IDLE Interrupt Status
7 RX_BREAK_INT	0 No Error. 1 Error.
6 RX_ERROR_INT	0 No Error. 1 Error.
5 TX_TIMEOUT_ERR_INT	0 No Error. 1 Error.
4 DMA_ERR_INT	If any bit in the DMA Error Interrupt Status Register is set, then this bit is set. on seeing this bit set, the ocp driver will read the Error Interrupt Status Register. 0 No Error. 1 Error.
3 DMA_INT	This bit is set when a Transmit or Receive Operation is completed for DMA.

Table continues on the next page...

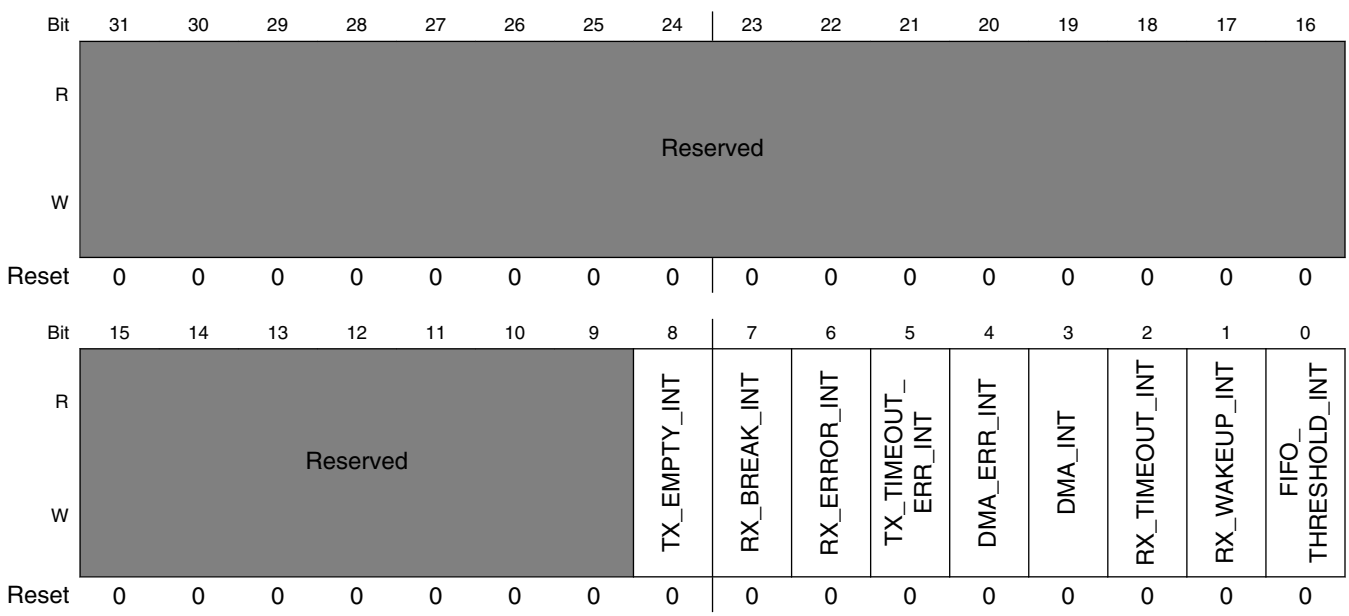
MIPI_HSI_IRQSTAT field descriptions (continued)

Field	Description
2 RX_TIMEOUT_INT	If any bit in the HSI Error Interrupt Status Register is set, then this bit is set. on seeing this bit set, the ocp driver will read the Error Interrupt Status Register. 0 No Error. 1 Error.
1 RX_WAKEUP_INT	1 Receiver Wakeup event is occurred 0 Receiver Wakeup event is not occurred
0 FIFO_THRESHOLD_INT	1 Threshold amount of data reached in TX/Rx FIFO Interrupt Status 0 Threshold amount of data not reached in TX/Rx FIFO Interrupt Status

43.5.14 HSI Interrupt Status Enable Register (MIPI_HSI_IRQSTAT_EN)

This register contains the HSI controller Interrupt Status Enable.

Address: 220_8000h base + 34h offset = 220_8034h



MIPI_HSI_IRQSTAT_EN field descriptions

Field	Description
31–9 Reserved	This field is reserved. Reserved, always set to zero.

Table continues on the next page...

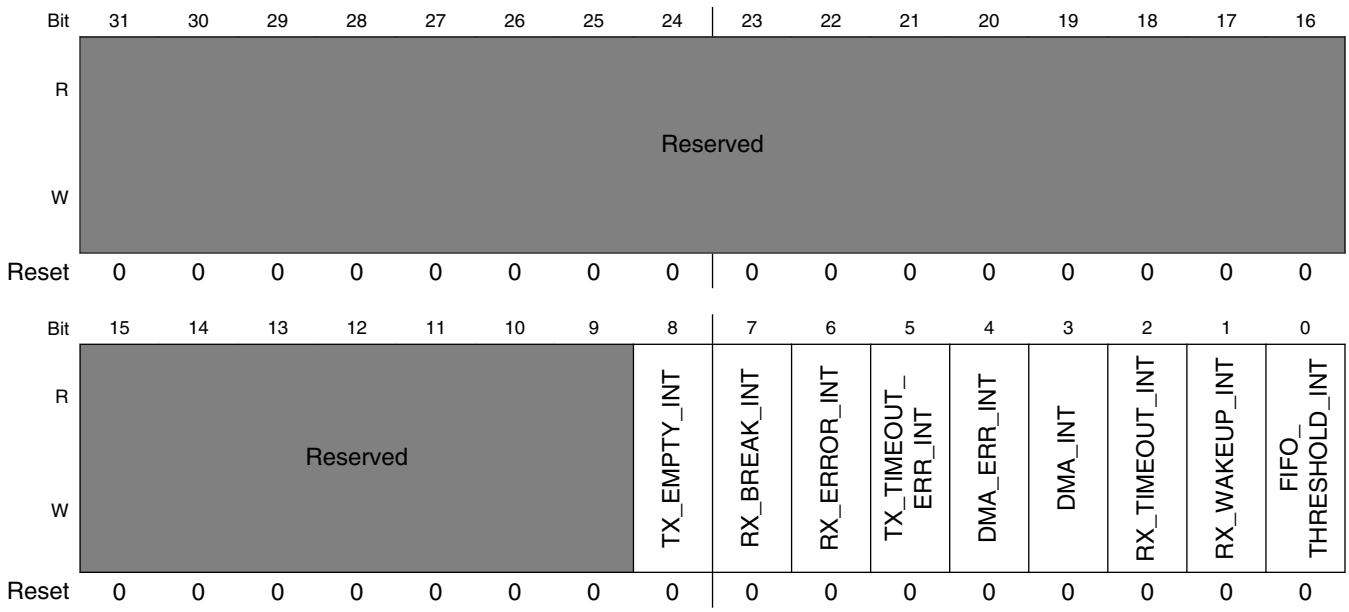
MIPI_HSI_IRQSTAT_EN field descriptions (continued)

Field	Description
8 TX_EMPTY_INT	1 Interrupt status enabled for TX_EMPTY_INT_STATUS interrupt. 0 Interrupt status masked TX_EMPTY_INT_STATUS interrupt.
7 RX_BREAK_INT	1 Interrupt status enabled for RX_BREAK status interrupt. 0 Interrupt status masked RX_BREAK status interrupt.
6 RX_ERROR_INT	1 Interrupt status enabled for RX_ERROR status interrupt. 0 Interrupt status masked RX_ERROR status interrupt.
5 TX_TIMEOUT_ERR_INT	1 Interrupt status enabled for TX_TIMEOUT_ERR status interrupt. 0 Interrupt status masked TX_TIMEOUT_ERR status interrupt.
4 DMA_ERR_INT	1 Interrupt status enabled for DMA_ERROR_INT_STATUS interrupt. 0 Interrupt status masked DMA_ERROR_INT_STATUS interrupt.
3 DMA_INT	1 Interrupt status enabled for DMA_INT_STATUS interrupt. 0 Interrupt status masked DMA_INT_STATUS interrupt.
2 RX_TIMEOUT_INT	1 Interrupt status enabled for RX_TIMEOUT_INT_STATUS interrupt. 0 Interrupt status masked RX_TIMEOUT_INT_STATUS interrupt.
1 RX_WAKEUP_INT	1 Interrupt status enabled for RX_WAKEUP_INT_STATUS interrupt. 0 Interrupt status masked RX_WAKEUP_INT_STATUS interrupt.
0 FIFO_THRESHOLD_INT	1 Interrupt status enabled for FIFO_THRESHOLD_INT_STATUS interrupt. 0 Interrupt status masked FIFO_THRESHOLD_INT_STATUS interrupt.

43.5.15 HSI Interrupt Signal Enable Register (MIPI_HSI_IRQSIG_EN)

This register contains the HSI controller Interrupt Signal Enable.

Address: 220_8000h base + 38h offset = 220_8038h



MIPI_HSI_IRQSIG_EN field descriptions

Field	Description
31–9 Reserved	This field is reserved. Reserved, always set to zero.
8 TX_EMPTY_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for HSI TX_EMPTY interrupt. 0 Interrupt signal masked for HSI TX_EMPTY interrupt.
7 RX_BREAK_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for RX_BREAK interrupt. 0 Interrupt signal masked for RX_BREAK interrupt.
6 RX_ERROR_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for RX Error interrupt. 0 Interrupt signal masked for RX Error interrupt.
5 TX_TIMEOUT_ERR_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for TX Timeout Error interrupt. 0 Interrupt signal masked for TX Timeout Error interrupt.

Table continues on the next page...

MIPI_HSI_IRQSIG_EN field descriptions (continued)

Field	Description
4 DMA_ERR_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for DMA Error interrupt. 0 Interrupt signal masked for DMA Error interrupt.
3 DMA_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for DMA Completed interrupt. 0 Interrupt signal masked for DMA Completed interrupt.
2 RX_TIMEOUT_ INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for RX TIMEOUT interrupt. 0 Interrupt signal masked for RX TIMEOUT interrupt.
1 RX_WAKEUP_ INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for HSI RX Wakeup interrupt. 0 Interrupt signal masked for HSI RX Wakeup interrupt.
0 FIFO_ THRESHOLD_ INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for HSI FIFO_THRESHOLD interrupt. 0 Interrupt signal masked for HSI FIFO_THRESHOLD interrupt.

43.5.16 HSI FIFO Threshold Interrupt Status Register (MIPI_HSI_FIFO_THR_IRQSTAT)

This register contains the HSI controller FIFO Threshold Interrupt Status.

Address: 220_8000h base + 3Ch offset = 220_803Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TX_CH15_INT	TX_CH14_INT	TX_CH13_INT	TX_CH12_INT	TX_CH11_INT	TX_CH10_INT	TX_CH9_INT	TX_CH8_INT	TX_CH7_INT	TX_CH6_INT	TX_CH5_INT	TX_CH4_INT	TX_CH3_INT	TX_CH2_INT	TX_CH1_INT	TX_CH0_INT
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX_CH15_INT	RX_CH14_INT	RX_CH13_INT	RX_CH12_INT	RX_CH11_INT	RX_CH10_INT	RX_CH9_INT	RX_CH8_INT	RX_CH7_INT	RX_CH6_INT	RX_CH5_INT	RX_CH4_INT	RX_CH3_INT	RX_CH2_INT	RX_CH1_INT	RX_CH0_INT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_FIFO_THR_IRQSTAT field descriptions

Field	Description
31 TX_CH15_INT	1 Threshold amount of data reached in Tx Channel 15 FIFO 0 Threshold amount of data not reached in Tx Channel 15 FIFO
30 TX_CH14_INT	1 Threshold amount of data reached in Tx Channel 14 FIFO 0 Threshold amount of data not reached in Tx Channel 14 FIFO
29 TX_CH13_INT	1 Threshold amount of data reached in Tx Channel 13 FIFO 0 Threshold amount of data not reached in Tx Channel 13 FIFO
28 TX_CH12_INT	1 Threshold amount of data reached in Tx Channel 12 FIFO 0 Threshold amount of data not reached in Tx Channel 12 FIFO
27 TX_CH11_INT	1 Threshold amount of data reached in Tx Channel 11 FIFO 0 Threshold amount of data not reached in Tx Channel 11 FIFO
26 TX_CH10_INT	1 Threshold amount of data reached in Tx Channel 10 FIFO 0 Threshold amount of data not reached in Tx Channel 10 FIFO
25 TX_CH9_INT	1 Threshold amount of data reached in Tx Channel 9 FIFO 0 Threshold amount of data not reached in Tx Channel 9 FIFO
24 TX_CH8_INT	1 Threshold amount of data reached in Tx Channel 8 FIFO 0 Threshold amount of data not reached in Tx Channel 8 FIFO
23 TX_CH7_INT	1 Threshold amount of data reached in Tx Channel 7 FIFO 0 Threshold amount of data not reached in Tx Channel 7 FIFO
22 TX_CH6_INT	1 Threshold amount of data reached in Tx Channel 6 FIFO 0 Threshold amount of data not reached in Tx Channel 6 FIFO
21 TX_CH5_INT	1 Threshold amount of data reached in Tx Channel 5 FIFO 0 Threshold amount of data not reached in Tx Channel 5 FIFO
20 TX_CH4_INT	1 Threshold amount of data reached in Tx Channel 4 FIFO 0 Threshold amount of data not reached in Tx Channel 4 FIFO
19 TX_CH3_INT	1 Threshold amount of data reached in Tx Channel 3 FIFO 0 Threshold amount of data not reached in Tx Channel 3 FIFO
18 TX_CH2_INT	1 Threshold amount of data reached in Tx Channel 2 FIFO 0 Threshold amount of data not reached in Tx Channel 2 FIFO
17 TX_CH1_INT	1 Threshold amount of data reached in Tx Channel 1 FIFO 0 Threshold amount of data not reached in Tx Channel 1 FIFO
16 TX_CH0_INT	1 Threshold amount of data reached in Tx Channel 0 FIFO 0 Threshold amount of data not reached in Tx Channel 0 FIFO
15 RX_CH15_INT	1 Threshold amount of data reached in Rx Channel 15 FIFO 0 Threshold amount of data not reached in Rx Channel 15 FIFO
14 RX_CH14_INT	1 Threshold amount of data reached in Rx Channel 14 FIFO 0 Threshold amount of data not reached in Rx Channel 14 FIFO
13 RX_CH13_INT	1 Threshold amount of data reached in Rx Channel 13 FIFO 0 Threshold amount of data not reached in Rx Channel 13 FIFO
12 RX_CH12_INT	1 Threshold amount of data reached in Rx Channel 12 FIFO 0 Threshold amount of data not reached in Rx Channel 12 FIFO
11 RX_CH11_INT	1 Threshold amount of data reached in Rx Channel 11 FIFO 0 Threshold amount of data not reached in Rx Channel 11 FIFO

Table continues on the next page...

MIPI_HSI_FIFO_THR_IRQSTAT field descriptions (continued)

Field	Description
10 RX_CH10_INT	1 Threshold amount of data reached in Rx Channel 10 FIFO 0 Threshold amount of data not reached in Rx Channel 10 FIFO
9 RX_CH9_INT	1 Threshold amount of data reached in Rx Channel 9 FIFO 0 Threshold amount of data not reached in Rx Channel 9 FIFO
8 RX_CH8_INT	1 Threshold amount of data reached in Rx Channel 8 FIFO 0 Threshold amount of data not reached in Rx Channel 8 FIFO
7 RX_CH7_INT	1 Threshold amount of data reached in Rx Channel 7 FIFO 0 Threshold amount of data not reached in Rx Channel 7 FIFO
6 RX_CH6_INT	1 Threshold amount of data reached in Rx Channel 6 FIFO 0 Threshold amount of data not reached in Rx Channel 6 FIFO
5 RX_CH5_INT	1 Threshold amount of data reached in Rx Channel 5 FIFO 0 Threshold amount of data not reached in Rx Channel 5 FIFO
4 RX_CH4_INT	1 Threshold amount of data reached in Rx Channel 4 FIFO 0 Threshold amount of data not reached in Rx Channel 4 FIFO
3 RX_CH3_INT	1 Threshold amount of data reached in Rx Channel 3 FIFO 0 Threshold amount of data not reached in Rx Channel 3 FIFO
2 RX_CH2_INT	1 Threshold amount of data reached in Rx Channel 2 FIFO 0 Threshold amount of data not reached in Rx Channel 2 FIFO
1 RX_CH1_INT	1 Threshold amount of data reached in Rx Channel 1 FIFO 0 Threshold amount of data not reached in Rx Channel 1 FIFO
0 RX_CH0_INT	1 Threshold amount of data reached in Rx Channel 0 FIFO 0 Threshold amount of data not reached in Rx Channel 0 FIFO

43.5.17 HSI FIFO Threshold Interrupt Status Enable Register (MIPI_HSI_FIFO_THR_IRQSTAT_EN)

This register contains the HSI controller FIFO Threshold Interrupt Status Enable.

Address: 220_8000h base + 40h offset = 220_8040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TX_CH15_INT	TX_CH14_INT	TX_CH13_INT	TX_CH12_INT	TX_CH11_INT	TX_CH10_INT	TX_CH9_INT	TX_CH8_INT	TX_CH7_INT	TX_CH6_INT	TX_CH5_INT	TX_CH4_INT	TX_CH3_INT	TX_CH2_INT	TX_CH1_INT	TX_CH0_INT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX_CH15_INT	RX_CH14_INT	RX_CH13_INT	RX_CH12_INT	RX_CH11_INT	RX_CH10_INT	RX_CH9_INT	RX_CH8_INT	RX_CH7_INT	RX_CH6_INT	RX_CH5_INT	RX_CH4_INT	RX_CH3_INT	RX_CH2_INT	RX_CH1_INT	RX_CH0_INT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_FIFO_THR_IRQSTAT_EN field descriptions

Field	Description
31 TX_CH15_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch15 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch15 threshold Reached interrupt.
30 TX_CH14_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch14 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch14 threshold Reached interrupt.
29 TX_CH13_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch13 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch13 threshold Reached interrupt.
28 TX_CH12_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch12 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch12 threshold Reached interrupt.
27 TX_CH11_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch11 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch11 threshold Reached interrupt.

Table continues on the next page...

MIPI_HSI_FIFO_THR_IRQSTAT_EN field descriptions (continued)

Field	Description
26 TX_CH10_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch10 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch10 threshold Reached interrupt.
25 TX_CH9_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch9 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch9 threshold Reached interrupt.
24 TX_CH8_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch8 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch8 threshold Reached interrupt.
23 TX_CH7_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch7 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch7 threshold Reached interrupt.
22 TX_CH6_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch6 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch6 threshold Reached interrupt.
21 TX_CH5_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch5 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch5 threshold Reached interrupt.
20 TX_CH4_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch4 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch4 threshold Reached interrupt.
19 TX_CH3_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch3 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch3 threshold Reached interrupt.
18 TX_CH2_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch2 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch2 threshold Reached interrupt.
17 TX_CH1_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch1 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch1 threshold Reached interrupt.
16 TX_CH0_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch0 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch0 threshold Reached interrupt.
15 RX_CH15_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch15 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch15 threshold Reached interrupt.

Table continues on the next page...

MIPI_HSI_FIFO_THR_IRQSTAT_EN field descriptions (continued)

Field	Description
14 RX_CH14_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch14 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch14 threshold Reached interrupt.
13 RX_CH13_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch13 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch13 threshold Reached interrupt.
12 RX_CH12_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch12 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch12 threshold Reached interrupt.
11 RX_CH11_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch11 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch11 threshold Reached interrupt.
10 RX_CH10_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch10 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch10 threshold Reached interrupt.
9 RX_CH9_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch9 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch9 threshold Reached interrupt.
8 RX_CH8_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch8 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch8 threshold Reached interrupt.
7 RX_CH7_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch7 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch7 threshold Reached interrupt.
6 RX_CH6_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch6 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch6 threshold Reached interrupt.
5 RX_CH5_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch5 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch5 threshold Reached interrupt.
4 RX_CH4_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch4 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch4 threshold Reached interrupt.
3 RX_CH3_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch3 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch3 threshold Reached interrupt.

Table continues on the next page...

MIPI_HSI_FIFO_THR_IRQSTAT_EN field descriptions (continued)

Field	Description
2 RX_CH2_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch2 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch2 threshold Reached interrupt.
1 RX_CH1_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch1 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch1 threshold Reached interrupt.
0 RX_CH0_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch0 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch0 threshold Reached interrupt.

43.5.18 HSI FIFO Threshold Interrupt Signal Enable Register (MIPI_HSI_FIFO_THR_IRQSIG_EN)

This register contains the HSI controller FIFO Threshold Interrupt Enable.

Address: 220_8000h base + 44h offset = 220_8044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TX_CH15_INT	TX_CH14_INT	TX_CH13_INT	TX_CH12_INT	TX_CH11_INT	TX_CH10_INT	TX_CH9_INT	TX_CH8_INT	TX_CH7_INT	TX_CH6_INT	TX_CH5_INT	TX_CH4_INT	TX_CH3_INT	TX_CH2_INT	TX_CH1_INT	TX_CH0_INT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX_CH15_INT	RX_CH14_INT	RX_CH13_INT	RX_CH12_INT	RX_CH11_INT	RX_CH10_INT	RX_CH9_INT	RX_CH8_INT	RX_CH7_INT	RX_CH6_INT	RX_CH5_INT	RX_CH4_INT	RX_CH3_INT	RX_CH2_INT	RX_CH1_INT	RX_CH0_INT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_FIFO_THR_IRQSIG_EN field descriptions

Field	Description
31 TX_CH15_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch15 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch15 threshold Reached interrupt.

Table continues on the next page...

MIPI_HSI_FIFO_THR_IRQSIG_EN field descriptions (continued)

Field	Description
30 TX_CH14_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch14 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch14 threshold Reached interrupt.
29 TX_CH13_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch13 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch13 threshold Reached interrupt.
28 TX_CH12_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch12 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch12 threshold Reached interrupt.
27 TX_CH11_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch11 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch11 threshold Reached interrupt.
26 TX_CH10_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch10 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch10 threshold Reached interrupt.
25 TX_CH9_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch9 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch9 threshold Reached interrupt.
24 TX_CH8_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch8 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch8 threshold Reached interrupt.
23 TX_CH7_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch7 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch7 threshold Reached interrupt.
22 TX_CH6_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch6 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch6 threshold Reached interrupt.
21 TX_CH5_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch5 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch5 threshold Reached interrupt.
20 TX_CH4_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch4 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch4 threshold Reached interrupt.
19 TX_CH3_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch3 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch3 threshold Reached interrupt.

Table continues on the next page...

MIPI_HSI_FIFO_THR_IRQSIG_EN field descriptions (continued)

Field	Description
18 TX_CH2_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch2 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch2 threshold Reached interrupt.
17 TX_CH1_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch1 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch1 threshold Reached interrupt.
16 TX_CH0_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Tx Ch0 threshold Reached interrupt. 0 Interrupt signal masked for Tx Ch0 threshold Reached interrupt.
15 RX_CH15_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch15 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch15 threshold Reached interrupt.
14 RX_CH14_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch14 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch14 threshold Reached interrupt.
13 RX_CH13_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch13 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch13 threshold Reached interrupt.
12 RX_CH12_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch12 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch12 threshold Reached interrupt.
11 RX_CH11_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch11 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch11 threshold Reached interrupt.
10 RX_CH10_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch10 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch10 threshold Reached interrupt.
9 RX_CH9_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch9 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch9 threshold Reached interrupt.
8 RX_CH8_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch8 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch8 threshold Reached interrupt.
7 RX_CH7_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch7 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch7 threshold Reached interrupt.

Table continues on the next page...

MIPI_HSI_FIFO_THR_IRQSIG_EN field descriptions (continued)

Field	Description
6 RX_CH6_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch6 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch6 threshold Reached interrupt.
5 RX_CH5_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch5 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch5 threshold Reached interrupt.
4 RX_CH4_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch4 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch4 threshold Reached interrupt.
3 RX_CH3_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch3 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch3 threshold Reached interrupt.
2 RX_CH2_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch2 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch2 threshold Reached interrupt.
1 RX_CH1_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch1 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch1 threshold Reached interrupt.
0 RX_CH0_INT	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for Rx Ch0 threshold Reached interrupt. 0 Interrupt signal masked for Rx Ch0 threshold Reached interrupt.

43.5.19 Tx Channel n Data Port Register (MIPI_HSI_TX_CH*n*_DP)

This Register is connected to fifo data port for Tx Channel n.

Address: 220_8000h base + 50h offset + (4d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_TX_CH*n*_DP field descriptions

Field	Description
DATA	Software could Write/Read this bits to access Tx Channel n.

43.5.20 Rx Channel n Data Port Register (MIPI_HSI_RX_CH*n*_DP)

This Register is connected to fifo data port for Rx Channel n.

Address: 220_8000h base + 90h offset + (4d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

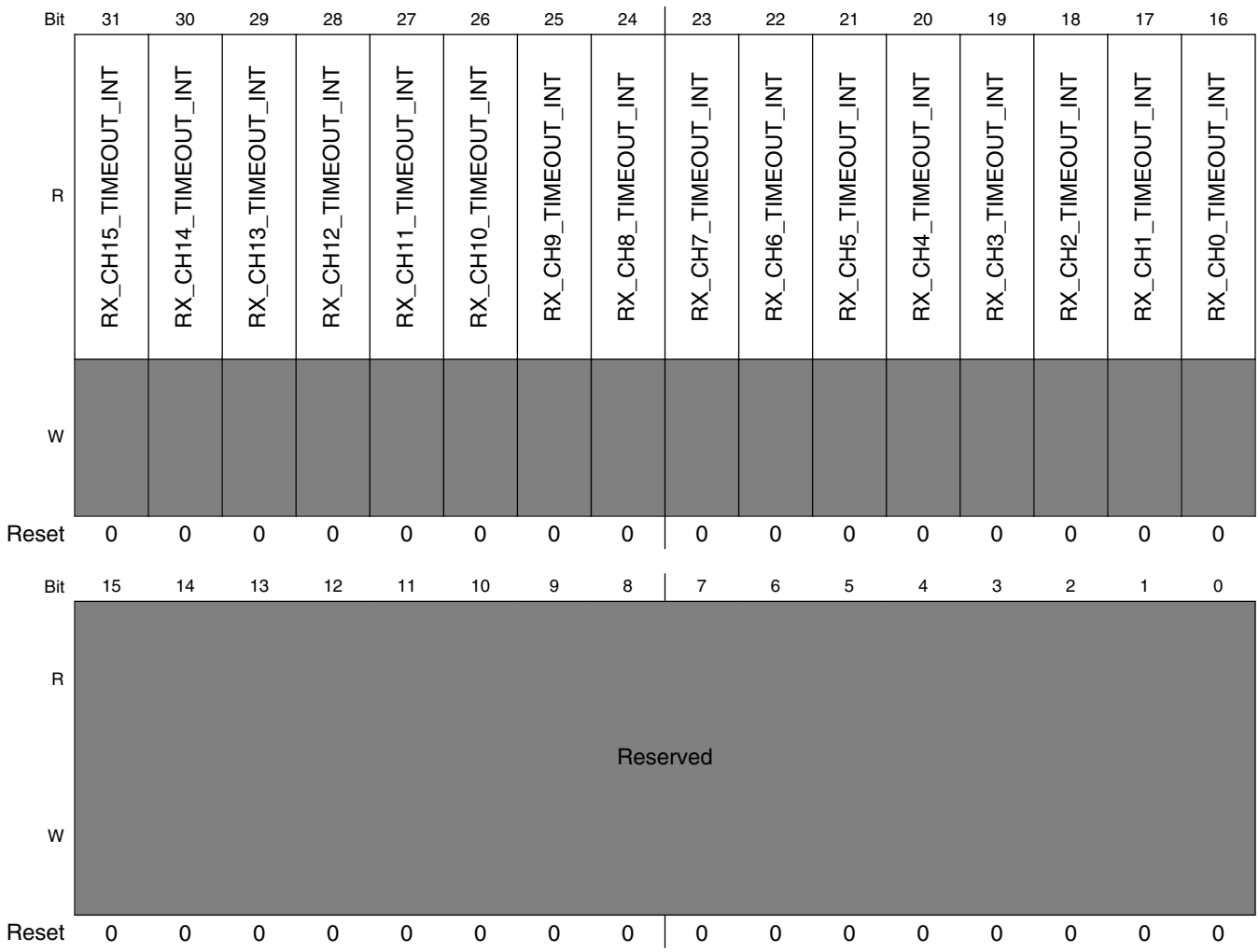
MIPI_HSI_RX_CH*n*_DP field descriptions

Field	Description
DATA	Software could Write/Read this bits to access Rx Channel n.

43.5.21 HSI Error Interrupt Status Register (MIPI_HSI_ERR_IRQSTAT)

This register contains the HSI controller Error Interrupt Status.

Address: 220_8000h base + D0h offset = 220_80D0h



MIPI_HSI_ERR_IRQSTAT field descriptions

Field	Description
31 RX_CH15_ TIMEOUT_INT	This status bit is set when data timeout counter for ch15 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch15 buffer and then read HSI Status register to find the further status of the Rx ch15 Buffer. The host driver has to read the Rx ch15 fifo on Dword basis, till the fifo is completely empty.

Table continues on the next page...

MIPI_HSI_ERR_IRQSTAT field descriptions (continued)

Field	Description
30 RX_CH14_ TIMEOUT_INT	This status bit is set when data timeout counter for ch14 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch14 buffer and then read HSI Status register to find the further status of the Rx ch14 Buffer. The host driver has to read the Rx ch14 fifo on Dword basis, till the fifo is completely empty.
29 RX_CH13_ TIMEOUT_INT	This status bit is set when data timeout counter for ch13 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch13 buffer and then read HSI Status register to find the further status of the Rx ch13 Buffer. The host driver has to read the Rx ch13 fifo on Dword basis, till the fifo is completely empty.
28 RX_CH12_ TIMEOUT_INT	This status bit is set when data timeout counter for ch12 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch12 buffer and then read HSI Status register to find the further status of the Rx ch12 Buffer. The host driver has to read the Rx ch12 fifo on Dword basis, till the fifo is completely empty.
27 RX_CH11_ TIMEOUT_INT	This status bit is set when data timeout counter for ch11 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch11 buffer and then read HSI Status register to find the further status of the Rx ch11 Buffer. The host driver has to read the Rx ch11 fifo on Dword basis, till the fifo is completely empty.
26 RX_CH10_ TIMEOUT_INT	This status bit is set when data timeout counter for ch10 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch10 buffer and then read HSI Status register to find the further status of the Rx ch10 Buffer. The host driver has to read the Rx ch10 fifo on Dword basis, till the fifo is completely empty.
25 RX_CH9_ TIMEOUT_INT	This status bit is set when data timeout counter for ch9 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch9 buffer and then read HSI Status register to find the further status of the Rx ch9 Buffer. The host driver has to read the Rx ch9 fifo on Dword basis, till the fifo is completely empty.
24 RX_CH8_ TIMEOUT_INT	This status bit is set when data timeout counter for ch8 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch8 buffer and then read HSI Status register to find the further status of the Rx ch8 Buffer. The host driver has to read the Rx ch8 fifo on Dword basis, till the fifo is completely empty.
23 RX_CH7_ TIMEOUT_INT	This status bit is set when data timeout counter for ch7 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch7 buffer and then read HSI Status register to find the further status of the Rx ch7 Buffer. The host driver has to read the Rx ch7 fifo on Dword basis, till the fifo is completely empty.
22 RX_CH6_ TIMEOUT_INT	This status bit is set when data timeout counter for ch6 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch6 buffer and then read HSI Status register to find the further status of the Rx ch6 Buffer. The host driver has to read the Rx ch6 fifo on Dword basis, till the fifo is completely empty.
21 RX_CH5_ TIMEOUT_INT	This status bit is set when data timeout counter for ch5 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch5 buffer and then read HSI Status register to find the further status of the Rx ch5 Buffer. The host driver has to read the Rx ch5 fifo on Dword basis, till the fifo is completely empty.
20 RX_CH4_ TIMEOUT_INT	This status bit is set when data timeout counter for ch4 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch4 buffer and then read HSI Status register to find the further status of the Rx ch4 Buffer. The host driver has to read the Rx ch4 fifo on Dword basis, till the fifo is completely empty.

Table continues on the next page...

MIPI_HSI_ERR_IRQSTAT field descriptions (continued)

Field	Description
19 RX_CH3_ TIMEOUT_INT	This status bit is set when data timeout counter for ch3 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch3 buffer and then read HSI Status register to find the further status of the Rx ch3 Buffer. The host driver has to read the Rx ch3 fifo on Dword basis, till the fifo is completely empty.
18 RX_CH2_ TIMEOUT_INT	This status bit is set when data timeout counter for ch2 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch2 buffer and then read HSI Status register to find the further status of the Rx ch2 Buffer. The host driver has to read the Rx ch2 fifo on Dword basis, till the fifo is completely empty.
17 RX_CH1_ TIMEOUT_INT	This status bit is set when data timeout counter for ch1 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch1 buffer and then read HSI Status register to find the further status of the Rx ch1 Buffer. The host driver has to read the Rx ch1 fifo on Dword basis, till the fifo is completely empty.
16 RX_CH0_ TIMEOUT_INT	This status bit is set when data timeout counter for ch0 reaches the data timeout counter value. On receiving the interrupt the host driver should read 1Dword of data from Rx ch0 buffer and then read HSI Status register to find the further status of the Rx ch0 Buffer. The host driver has to read the Rx ch0 fifo on Dword basis, till the fifo is completely empty.
Reserved	This field is reserved. Reserved, always set to zero.

43.5.22 HSI Error Interrupt Status Enable Register (MIPI_HSI_ERR_IRQSTAT_EN)

This register contains the HSI controller Error Interrupt Status Enable.

Address: 220_8000h base + D4h offset = 220_80D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RX_CH15_TIMEOUT_INT_EN	RX_CH14_TIMEOUT_INT_EN	RX_CH13_TIMEOUT_INT_EN	RX_CH12_TIMEOUT_INT_EN	RX_CH11_TIMEOUT_INT_EN	RX_CH10_TIMEOUT_INT_EN	RX_CH9_TIMEOUT_INT_EN	RX_CH8_TIMEOUT_INT_EN	RX_CH7_TIMEOUT_INT_EN	RX_CH6_TIMEOUT_INT_EN	RX_CH5_TIMEOUT_INT_EN	RX_CH4_TIMEOUT_INT_EN	RX_CH3_TIMEOUT_INT_EN	RX_CH2_TIMEOUT_INT_EN	RX_CH1_TIMEOUT_INT_EN	RX_CH0_TIMEOUT_INT_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_ERR_IRQSTAT_EN field descriptions

Field	Description
31 RX_CH15_TIMEOUT_INT_EN	1 Interrupt status enabled for data timeout for ch15 interrupt. 0 Interrupt status masked for data timeout for ch15 interrupt.
30 RX_CH14_TIMEOUT_INT_EN	1 Interrupt status enabled for data timeout for ch14 interrupt. 0 Interrupt status masked for data timeout for ch14 interrupt.
29 RX_CH13_TIMEOUT_INT_EN	1 Interrupt status enabled for data timeout for ch13 interrupt. 0 Interrupt status masked for data timeout for ch13 interrupt.
28 RX_CH12_TIMEOUT_INT_EN	1 Interrupt status enabled for data timeout for ch12 interrupt. 0 Interrupt status masked for data timeout for ch12 interrupt.

Table continues on the next page...

MIPI_HSI_ERR_IRQSTAT_EN field descriptions (continued)

Field	Description
27 RX_CH11_ TIMEOUT_INT_ EN	1 Interrupt status enabled for data timeout for ch11 interrupt. 0 Interrupt status masked for data timeout for ch11 interrupt.
26 RX_CH10_ TIMEOUT_INT_ EN	1 Interrupt status enabled for data timeout for ch10 interrupt. 0 Interrupt status masked for data timeout for ch10 interrupt.
25 RX_CH9_ TIMEOUT_INT_ EN	1 Interrupt status enabled for data timeout for ch9 interrupt. 0 Interrupt status masked for data timeout for ch9 interrupt.
24 RX_CH8_ TIMEOUT_INT_ EN	1 Interrupt status enabled for data timeout for ch8 interrupt. 0 Interrupt status masked for data timeout for ch8 interrupt.
23 RX_CH7_ TIMEOUT_INT_ EN	1 Interrupt status enabled for data timeout for ch7 interrupt. 0 Interrupt status masked for data timeout for ch7 interrupt.
22 RX_CH6_ TIMEOUT_INT_ EN	1 Interrupt status enabled for data timeout for ch6 interrupt. 0 Interrupt status masked for data timeout for ch6 interrupt.
21 RX_CH5_ TIMEOUT_INT_ EN	1 Interrupt status enabled for data timeout for ch5 interrupt. 0 Interrupt status masked for data timeout for ch5 interrupt.
20 RX_CH4_ TIMEOUT_INT_ EN	1 Interrupt status enabled for data timeout for ch4 interrupt. 0 Interrupt status masked for data timeout for ch4 interrupt.
19 RX_CH3_ TIMEOUT_INT_ EN	1 Interrupt status enabled for data timeout for ch3 interrupt. 0 Interrupt status masked for data timeout for ch3 interrupt.
18 RX_CH2_ TIMEOUT_INT_ EN	1 Interrupt status enabled for data timeout for ch2 interrupt. 0 Interrupt status masked for data timeout for ch2 interrupt.
17 RX_CH1_ TIMEOUT_INT_ EN	1 Interrupt status enabled for data timeout for ch1 interrupt. 0 Interrupt status masked for data timeout for ch1 interrupt.
16 RX_CH0_ TIMEOUT_INT_ EN	1 Interrupt status enabled for data timeout for ch0 interrupt. 0 Interrupt status masked for data timeout for ch0 interrupt.
Reserved	This field is reserved. Reserved, always set to zero.

43.5.23 HSI Error Interrupt Signal Enable Register (MIPI_HSI_ERR_IRQSIG_EN)

This register contains the HSI controller Error Interrupt Signal Enable.

Address: 220_8000h base + D8h offset = 220_80D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RX_CH15_TIMEOUT_INT_EN	RX_CH14_TIMEOUT_INT_EN	RX_CH13_TIMEOUT_INT_EN	RX_CH12_TIMEOUT_INT_EN	RX_CH11_TIMEOUT_INT_EN	RX_CH10_TIMEOUT_INT_EN	RX_CH9_TIMEOUT_INT_EN	RX_CH8_TIMEOUT_INT_EN	RX_CH7_TIMEOUT_INT_EN	RX_CH6_TIMEOUT_INT_EN	RX_CH5_TIMEOUT_INT_EN	RX_CH4_TIMEOUT_INT_EN	RX_CH3_TIMEOUT_INT_EN	RX_CH2_TIMEOUT_INT_EN	RX_CH1_TIMEOUT_INT_EN	RX_CH0_TIMEOUT_INT_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_ERR_IRQSIG_EN field descriptions

Field	Description
31 RX_CH15_TIMEOUT_INT_EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch15 interrupt. 0 Interrupt signal masked for data timeout for ch15 interrupt.
30 RX_CH14_TIMEOUT_INT_EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch14 interrupt. 0 Interrupt signal masked for data timeout for ch14 interrupt.
29 RX_CH13_TIMEOUT_INT_EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch13 interrupt. 0 Interrupt signal masked for data timeout for ch13 interrupt.
28 RX_CH12_TIMEOUT_INT_EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch12 interrupt. 0 Interrupt signal masked for data timeout for ch12 interrupt.

Table continues on the next page...

MIPI_HSI_ERR_IRQSIG_EN field descriptions (continued)

Field	Description
27 RX_CH11_ TIMEOUT_INT_ EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch11 interrupt. 0 Interrupt signal masked for data timeout for ch11 interrupt.
26 RX_CH10_ TIMEOUT_INT_ EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch10 interrupt. 0 Interrupt signal masked for data timeout for ch10 interrupt.
25 RX_CH9_ TIMEOUT_INT_ EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch9 interrupt. 0 Interrupt signal masked for data timeout for ch9 interrupt.
24 RX_CH8_ TIMEOUT_INT_ EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch8 interrupt. 0 Interrupt signal masked for data timeout for ch8 interrupt.
23 RX_CH7_ TIMEOUT_INT_ EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch7 interrupt. 0 Interrupt signal masked for data timeout for ch7 interrupt.
22 RX_CH6_ TIMEOUT_INT_ EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch6 interrupt. 0 Interrupt signal masked for data timeout for ch6 interrupt.
21 RX_CH5_ TIMEOUT_INT_ EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch5 interrupt. 0 Interrupt signal masked for data timeout for ch5 interrupt.
20 RX_CH4_ TIMEOUT_INT_ EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch4 interrupt. 0 Interrupt signal masked for data timeout for ch4 interrupt.
19 RX_CH3_ TIMEOUT_INT_ EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch3 interrupt. 0 Interrupt signal masked for data timeout for ch3 interrupt.
18 RX_CH2_ TIMEOUT_INT_ EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch2 interrupt. 0 Interrupt signal masked for data timeout for ch2 interrupt.
17 RX_CH1_ TIMEOUT_INT_ EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch1 interrupt. 0 Interrupt signal masked for data timeout for ch1interrupt.
16 RX_CH0_ TIMEOUT_INT_ EN	Setting this bit will enable interrupt generation on interrupt line. 1 Interrupt signal enabled for data timeout for ch0 interrupt. 0 Interrupt signal masked for data timeout for ch0 interrupt.
Reserved	This field is reserved.

Table continues on the next page...

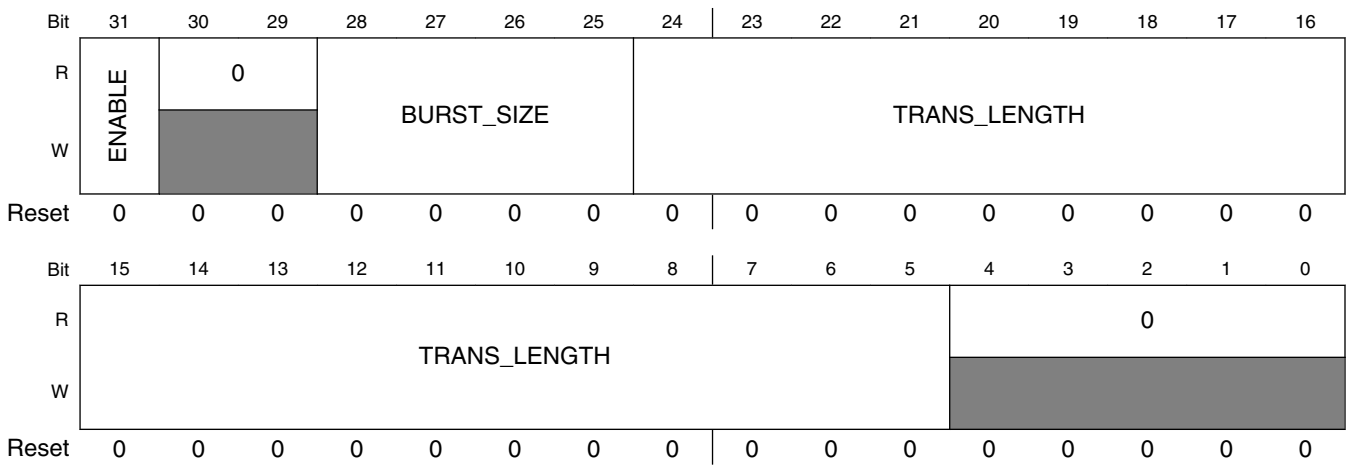
MIPI_HSI_ERR_IRQSIG_EN field descriptions (continued)

Field	Description
	Reserved, always set to zero.

43.5.24 Tx DMA Channel n Configuration Register (MIPI_HSI_TDMA*n*_CONF)

This register contains the configurations of enable/disable, burst size and transfer count for Tx DMA channel n.

Address: 220_8000h base + DCh offset + (4d × i), where i=0d to 15d



MIPI_HSI_TDMA*n*_CONF field descriptions

Field	Description
31 ENABLE	Setting this bit enables the internal Tx DMA channel n.
30–29 Reserved	This read-only field is reserved and always has the value 0.
28–25 BURST_SIZE	Burst size for Tx DMA channel n. The unit is Dword. The burst size should not be larger than relevant TRANS_LENGTH and FIFO_SIZE. h0 1Dword to transfer for each burst h1 2Dword to transfer for each burst h2 4Dword to transfer for each burst h10 1024Dword to transfer for each burst
24–5 TRANS_ LENGTH	Transfer data length for Tx DMA channel n. The unit is Dword. h1 1Dword to transfer h2 2Dwords to transfer

Table continues on the next page...

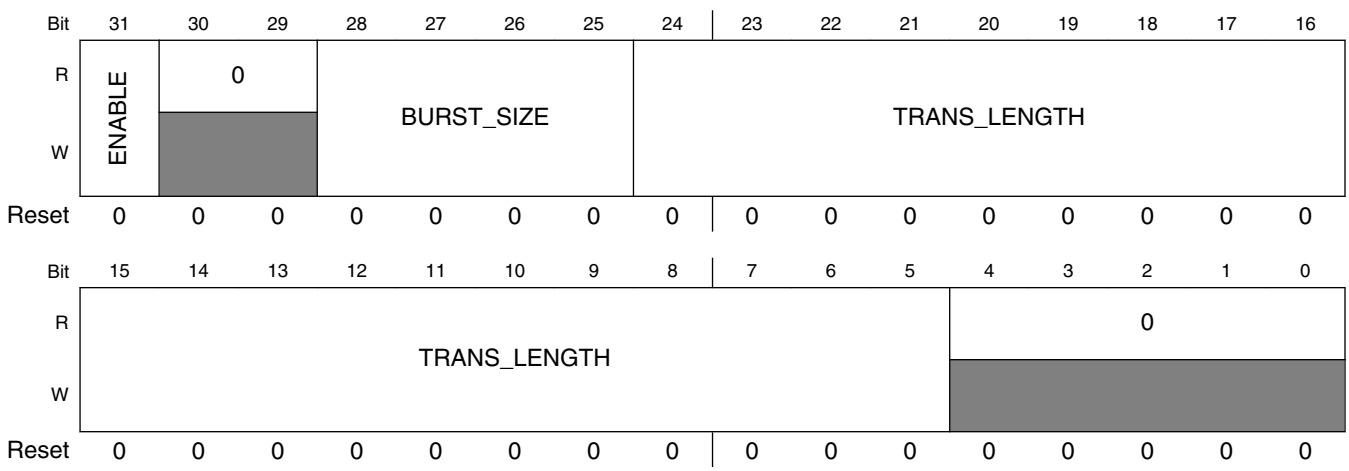
MIPI_HSI_TDMA*n*_CONF field descriptions (continued)

Field	Description
	hffffff 1048575Dwords to transfer
Reserved	This read-only field is reserved and always has the value 0.

43.5.25 Rx DMA Channel n Configuration Register (MIPI_HSI_RDMA*n*_CONF)

This register contains the configurations of enable/disable, burst size and transfer count for Rx DMA channel n.

Address: 220_8000h base + 11Ch offset + (4d × i), where i=0d to 15d



MIPI_HSI_RDMA*n*_CONF field descriptions

Field	Description
31 ENABLE	Setting this bit enables the internal Rx DMA channel n.
30–29 Reserved	This read-only field is reserved and always has the value 0.
28–25 BURST_SIZE	Burst size for Rx DMA channel n. The unit is Dword. The burst size should not be larger than relevant TRANS_LENGTH and FIFO_SIZE. h0 1Dword to transfer for each burst h1 2Dword to transfer for each burst h2 4Dword to transfer for each burst h10 1024Dword to transfer for each burst
24–5 TRANS_ LENGTH	Transfer data length for Rx DMA channel 0. The unit is Dword. h1 1Dword to transfer

Table continues on the next page...

MIPI_HSI_RDMA*n*_CONF field descriptions (continued)

Field	Description
	h2 2Dwords to transfer hffffff 1048575Dwords to transfer
Reserved	This read-only field is reserved and always has the value 0.

43.5.26 Tx DMA Channel n Start Address Register (MIPI_HSI_TDMA*n*_STA_ADDR)

This Register contains the physical Start Address HSI for Tx DMA Channel n.

Address: 220_8000h base + 15Ch offset + (4d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DS_ADDR																														0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_TDMA*n*_STA_ADDR field descriptions

Field	Description
31–2 DS_ADDR	The Physical Start Address for Tx DMA Channel n. DWord aligned
Reserved	This read-only field is reserved and always has the value 0.

43.5.27 Rx DMA Channel n Start Address Register (MIPI_HSI_RDMA*n*_STA_ADDR)

This Register contains the physical Start Address HSI for Rx DMA Channel n.

Address: 220_8000h base + 19Ch offset + (4d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DS_ADDR																														0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_RDMA*n*_STA_ADDR field descriptions

Field	Description
31–2 DS_ADDR	The Physical Start Address for Rx DMA Channel n. DWord aligned
Reserved	This read-only field is reserved and always has the value 0.

43.5.28 DMA Interrupt Status Register (MIPI_HSI_DMA_IRQSTAT)

This register contains all the interrupt status for HSI internal DMA

Address: 220_8000h base + 1DCh offset = 220_81DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TDMA15	TDMA14	TDMA13	TDMA12	TDMA11	TDMA10	TDMA9	TDMA8	TDMA7	TDMA6	TDMA5	TDMA4	TDMA3	TDMA2	TDMA1	TDMA0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDMA15	RDMA14	RDMA13	RDMA12	RDMA11	RDMA10	RDMA9	RDMA8	RDMA7	RDMA6	RDMA5	RDMA4	RDMA3	RDMA2	RDMA1	RDMA0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_DMA_IRQSTAT field descriptions

Field	Description
31 TDMA15	TDMA Channel 15 interrupt status
30 TDMA14	TDMA Channel 14 interrupt status
29 TDMA13	TDMA Channel 13 interrupt status
28 TDMA12	TDMA Channel 12 interrupt status
27 TDMA11	TDMA Channel 11 interrupt status
26 TDMA10	TDMA Channel 10 interrupt status
25 TDMA9	TDMA Channel 9 interrupt status
24 TDMA8	TDMA Channel 8 interrupt status
23 TDMA7	TDMA Channel 7 interrupt status
22 TDMA6	TDMA Channel 6 interrupt status
21 TDMA5	TDMA Channel 5 interrupt status
20 TDMA4	TDMA Channel 4 interrupt status
19 TDMA3	TDMA Channel 3 interrupt status
18 TDMA2	TDMA Channel 2 interrupt status
17 TDMA1	TDMA Channel 1 interrupt status
16 TDMA0	TDMA Channel 0 interrupt status
15 RDMA15	RDMA Channel 15 interrupt status
14 RDMA14	RDMA Channel 14 interrupt status
13 RDMA13	RDMA Channel 13 interrupt status
12 RDMA12	RDMA Channel 12 interrupt status
11 RDMA11	RDMA Channel 11 interrupt status
10 RDMA10	RDMA Channel 10 interrupt status
9 RDMA9	RDMA Channel 9 interrupt status

Table continues on the next page...

MIPI_HSI_DMA_IRQSTAT field descriptions (continued)

Field	Description
8 RDMA8	RDMA Channel 8 interrupt status
7 RDMA7	RDMA Channel 7 interrupt status
6 RDMA6	RDMA Channel 6 interrupt status
5 RDMA5	RDMA Channel 5 interrupt status
4 RDMA4	RDMA Channel 4 interrupt status
3 RDMA3	RDMA Channel 3 interrupt status
2 RDMA2	RDMA Channel 2 interrupt status
1 RDMA1	RDMA Channel 1 interrupt status
0 RDMA0	RDMA Channel 0 interrupt status

43.5.29 DMA Interrupt Enable Register (MIPI_HSI_DMA_IRQSTAT_EN)

This Register is used to select which DMA interrupt could send to HIS Interrupt Status Register

Address: 220_8000h base + 1E0h offset = 220_81E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDMA15	TDMA14	TDMA13	TDMA12	TDMA11	TDMA10	TDMA9	TDMA8	TDMA7	TDMA6	TDMA5	TDMA4	TDMA3	TDMA2	TDMA1	TDMA0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RDMA15	RDMA14	RDMA13	RDMA12	RDMA11	RDMA10	RDMA9	RDMA8	RDMA7	RDMA6	RDMA5	RDMA4	RDMA3	RDMA2	RDMA1	RDMA0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_DMA_IRQSTAT_EN field descriptions

Field	Description
31 TDMA15	TDMA Channel 15 interrupt Enable
30 TDMA14	TDMA Channel 14 interrupt Enable
29 TDMA13	TDMA Channel 13 interrupt Enable
28 TDMA12	TDMA Channel 12 interrupt Enable
27 TDMA11	TDMA Channel 11 interrupt Enable
26 TDMA10	TDMA Channel 10 interrupt Enable
25 TDMA9	TDMA Channel 9 interrupt Enable
24 TDMA8	TDMA Channel 8 interrupt Enable
23 TDMA7	TDMA Channel 7 interrupt Enable
22 TDMA6	TDMA Channel 6 interrupt Enable
21 TDMA5	TDMA Channel 5 interrupt Enable
20 TDMA4	TDMA Channel 4 interrupt Enable
19 TDMA3	TDMA Channel 3 interrupt Enable
18 TDMA2	TDMA Channel 2 interrupt Enable
17 TDMA1	TDMA Channel 1 interrupt Enable
16 TDMA0	TDMA Channel 0 interrupt Enable
15 RDMA15	RDMA Channel 15 interrupt Enable
14 RDMA14	RDMA Channel 14 interrupt Enable
13 RDMA13	RDMA Channel 13 interrupt Enable
12 RDMA12	RDMA Channel 12 interrupt Enable
11 RDMA11	RDMA Channel 11 interrupt Enable
10 RDMA10	RDMA Channel 10 interrupt Enable
9 RDMA9	RDMA Channel 9 interrupt Enable

Table continues on the next page...

MIPI_HSI_DMA_IRQSTAT_EN field descriptions (continued)

Field	Description
8 RDMA8	RDMA Channel 8 interrupt Enable
7 RDMA7	RDMA Channel 7 interrupt Enable
6 RDMA6	RDMA Channel 6 interrupt Enable
5 RDMA5	RDMA Channel 5 interrupt Enable
4 RDMA4	RDMA Channel 4 interrupt Enable
3 RDMA3	RDMA Channel 3 interrupt Enable
2 RDMA2	RDMA Channel 2 interrupt Enable
1 RDMA1	RDMA Channel 1 interrupt Enable
0 RDMA0	RDMA Channel 0 interrupt Enable

43.5.30 DMA Interrupt Status Signal Enable Register (MIPI_HSI_DMA_IRQSIG_EN)

This Register is used to select which DMA interrupt status could send to HIS Interrupt Status Register

Address: 220_8000h base + 1E4h offset = 220_81E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDMA15	TDMA14	TDMA13	TDMA12	TDMA11	TDMA10	TDMA9	TDMA8	TDMA7	TDMA6	TDMA5	TDMA4	TDMA3	TDMA2	TDMA1	TDMA0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RDMA15	RDMA14	RDMA13	RDMA12	RDMA11	RDMA10	RDMA9	RDMA8	RDMA7	RDMA6	RDMA5	RDMA4	RDMA3	RDMA2	RDMA1	RDMA0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_DMA_IRQSIG_EN field descriptions

Field	Description
31 TDMA15	TDMA Channel 15 interrupt status enable
30 TDMA14	TDMA Channel 14 interrupt status enable
29 TDMA13	TDMA Channel 13 interrupt status enable
28 TDMA12	TDMA Channel 12 interrupt status enable
27 TDMA11	TDMA Channel 11 interrupt status enable
26 TDMA10	TDMA Channel 10 interrupt status enable
25 TDMA9	TDMA Channel 9 interrupt status enable
24 TDMA8	TDMA Channel 8 interrupt status enable
23 TDMA7	TDMA Channel 7 interrupt status enable
22 TDMA6	TDMA Channel 6 interrupt status enable
21 TDMA5	TDMA Channel 5 interrupt status enable
20 TDMA4	TDMA Channel 4 interrupt status enable
19 TDMA3	TDMA Channel 3 interrupt status enable
18 TDMA2	TDMA Channel 2 interrupt status enable
17 TDMA1	TDMA Channel 1 interrupt status enable
16 TDMA0	TDMA Channel 0 interrupt status enable
15 RDMA15	RDMA Channel 15 interrupt status enable
14 RDMA14	RDMA Channel 14 interrupt status enable
13 RDMA13	RDMA Channel 13 interrupt status enable
12 RDMA12	RDMA Channel 12 interrupt status enable
11 RDMA11	RDMA Channel 11 interrupt status enable
10 RDMA10	RDMA Channel 10 interrupt status enable
9 RDMA9	RDMA Channel 9 interrupt status enable

Table continues on the next page...

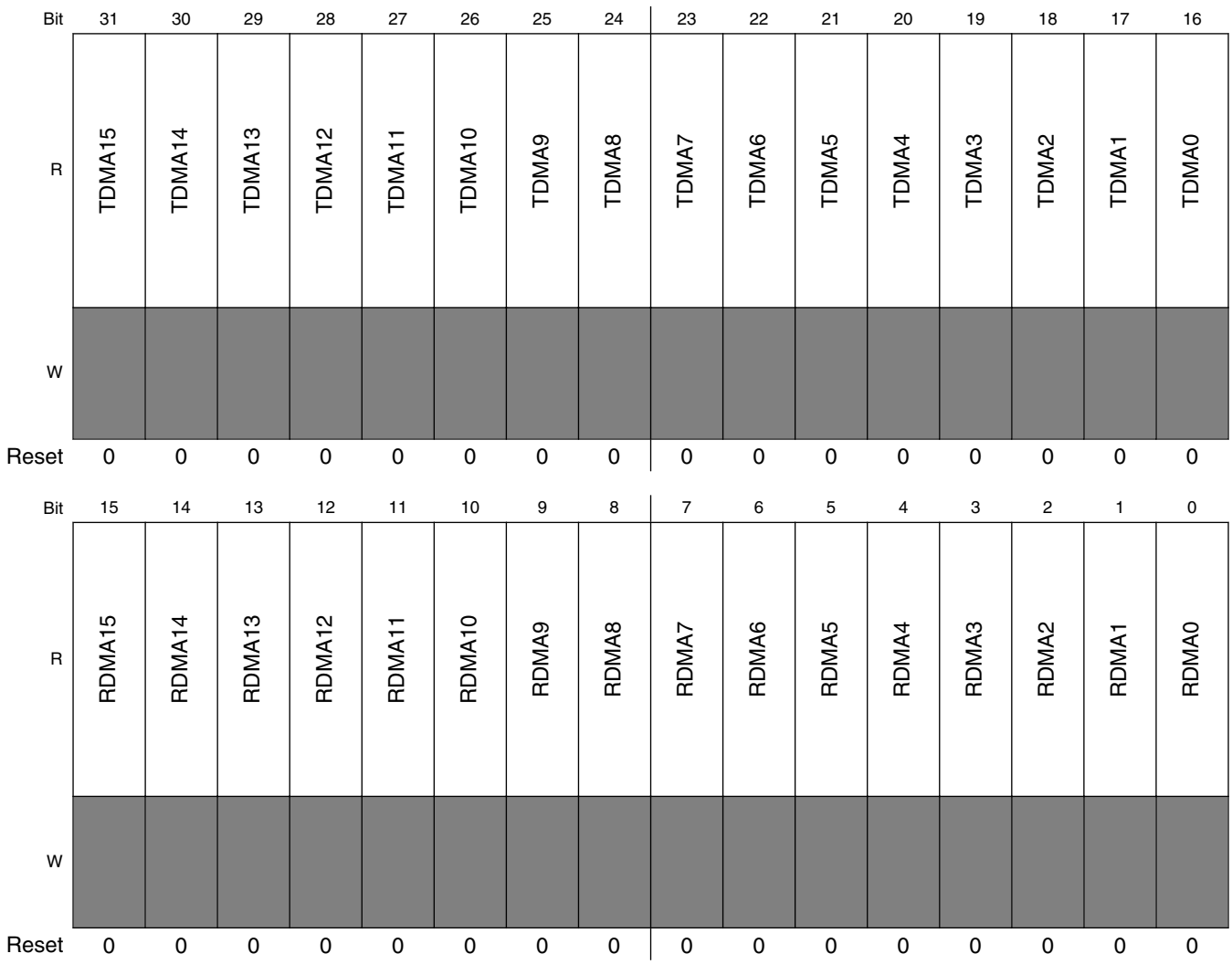
MIPI_HSI_DMA_IRQSIG_EN field descriptions (continued)

Field	Description
8 RDMA8	RDMA Channel 8 interrupt status enable
7 RDMA7	RDMA Channel 7 interrupt status enable
6 RDMA6	RDMA Channel 6 interrupt status enable
5 RDMA5	RDMA Channel 5 interrupt status enable
4 RDMA4	RDMA Channel 4 interrupt status enable
3 RDMA3	RDMA Channel 3 interrupt status enable
2 RDMA2	RDMA Channel 2 interrupt status enable
1 RDMA1	RDMA Channel 1 interrupt status enable
0 RDMA0	RDMA Channel 0 interrupt status enable

43.5.31 DMA Error Interrupt Status Register (MIPI_HSI_DMA_ERR_IRQSTAT)

This register contains all the error interrupt status for HSI internal DMA

Address: 220_8000h base + 1E8h offset = 220_81E8h



MIPI_HSI_DMA_ERR_IRQSTAT field descriptions

Field	Description
31 TDMA15	TDMA Channel 15 error interrupt status
30 TDMA14	TDMA Channel 14 error interrupt status

Table continues on the next page...

MIPI_HSI_DMA_ERR_IRQSTAT field descriptions (continued)

Field	Description
29 TDMA13	TDMA Channel 13 error interrupt status
28 TDMA12	TDMA Channel 12 error interrupt status
27 TDMA11	TDMA Channel 11 error interrupt status
26 TDMA10	TDMA Channel 10 error interrupt status
25 TDMA9	TDMA Channel 9 error interrupt status
24 TDMA8	TDMA Channel 8 error interrupt status
23 TDMA7	TDMA Channel 7 error interrupt status
22 TDMA6	TDMA Channel 6 error interrupt status
21 TDMA5	TDMA Channel 5 error interrupt status
20 TDMA4	TDMA Channel 4 error interrupt status
19 TDMA3	TDMA Channel 3 error interrupt status
18 TDMA2	TDMA Channel 2 error interrupt status
17 TDMA1	TDMA Channel 1 error interrupt status
16 TDMA0	TDMA Channel 0 error interrupt status
15 RDMA15	RDMA Channel 15 error interrupt status
14 RDMA14	RDMA Channel 14 error interrupt status
13 RDMA13	RDMA Channel 13 error interrupt status
12 RDMA12	RDMA Channel 12 error interrupt status
11 RDMA11	RDMA Channel 11 error interrupt status
10 RDMA10	RDMA Channel 10 error interrupt status
9 RDMA9	RDMA Channel 9 error interrupt status
8 RDMA8	RDMA Channel 8 error interrupt status
7 RDMA7	RDMA Channel 7 error interrupt status

Table continues on the next page...

MIPI_HSI_DMA_ERR_IRQSTAT field descriptions (continued)

Field	Description
6 RDMA6	RDMA Channel 6 error interrupt status
5 RDMA5	RDMA Channel 5 error interrupt status
4 RDMA4	RDMA Channel 4 error interrupt status
3 RDMA3	RDMA Channel 3 error interrupt status
2 RDMA2	RDMA Channel 2 error interrupt status
1 RDMA1	RDMA Channel 1 error interrupt status
0 RDMA0	RDMA Channel 0 error interrupt status

43.5.32 DMA Error Interrupt Enable Register (MIPI_HSI_DMA_ERR_IRQSTAT_EN)

This register is used to select which DMA error interrupt could send to HIS Interrupt Status Register

Address: 220_8000h base + 1ECh offset = 220_81ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDMA15	TDMA14	TDMA13	TDMA12	TDMA11	TDMA10	TDMA9	TDMA8	TDMA7	TDMA6	TDMA5	TDMA4	TDMA3	TDMA2	TDMA1	TDMA0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RDMA15	RDMA14	RDMA13	RDMA12	RDMA11	RDMA10	RDMA9	RDMA8	RDMA7	RDMA6	RDMA5	RDMA4	RDMA3	RDMA2	RDMA1	RDMA0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_DMA_ERR_IRQSTAT_EN field descriptions

Field	Description
31 TDMA15	TDMA Channel 15 error interrupt enable

Table continues on the next page...

MIPI_HSI_DMA_ERR_IRQSTAT_EN field descriptions (continued)

Field	Description
30 TDMA14	TDMA Channel 14 error interrupt enable
29 TDMA13	TDMA Channel 13 error interrupt enable
28 TDMA12	TDMA Channel 12 error interrupt enable
27 TDMA11	TDMA Channel 11 error interrupt enable
26 TDMA10	TDMA Channel 10 error interrupt enable
25 TDMA9	TDMA Channel 9 error interrupt enable
24 TDMA8	TDMA Channel 8 error interrupt enable
23 TDMA7	TDMA Channel 7 error interrupt enable
22 TDMA6	TDMA Channel 6 error interrupt enable
21 TDMA5	TDMA Channel 5 error interrupt enable
20 TDMA4	TDMA Channel 4 error interrupt enable
19 TDMA3	TDMA Channel 3 error interrupt enable
18 TDMA2	TDMA Channel 2 error interrupt enable
17 TDMA1	TDMA Channel 1 error interrupt enable
16 TDMA0	TDMA Channel 0 error interrupt enable
15 RDMA15	RDMA Channel 15 error interrupt enable
14 RDMA14	RDMA Channel 14 error interrupt enable
13 RDMA13	RDMA Channel 13 error interrupt enable
12 RDMA12	RDMA Channel 12 error interrupt enable
11 RDMA11	RDMA Channel 11 error interrupt enable
10 RDMA10	RDMA Channel 10 error interrupt enable
9 RDMA9	RDMA Channel 9 error interrupt enable
8 RDMA8	RDMA Channel 8 error interrupt enable

Table continues on the next page...

MIPI_HSI_DMA_ERR_IRQSTAT_EN field descriptions (continued)

Field	Description
7 RDMA7	RDMA Channel 7 error interrupt enable
6 RDMA6	RDMA Channel 6 error interrupt enable
5 RDMA5	RDMA Channel 5 error interrupt enable
4 RDMA4	RDMA Channel 4 error interrupt enable
3 RDMA3	RDMA Channel 3 error interrupt enable
2 RDMA2	RDMA Channel 2 error interrupt enable
1 RDMA1	RDMA Channel 1 error interrupt enable
0 RDMA0	RDMA Channel 0 error interrupt enable

43.5.33 DMA Error Interrupt Signal Enable Register (MIPI_HSI_DMA_ERR_IRQSIG_EN)

This Register is used to select which DMA error interrupt status could send to HIS Interrupt Status Register

Address: 220_8000h base + 1F0h offset = 220_81F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDMA15	TDMA14	TDMA13	TDMA12	TDMA11	TDMA10	TDMA9	TDMA8	TDMA7	TDMA6	TDMA5	TDMA4	TDMA3	TDMA2	TDMA1	TDMA0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RDMA15	RDMA14	RDMA13	RDMA12	RDMA11	RDMA10	RDMA9	RDMA8	RDMA7	RDMA6	RDMA5	RDMA4	RDMA3	RDMA2	RDMA1	RDMA0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_DMA_ERR_IRQSIG_EN field descriptions

Field	Description
31 TDMA15	TDMA Channel 15 error interrupt status enable
30 TDMA14	TDMA Channel 14 error interrupt status enable
29 TDMA13	TDMA Channel 13 error interrupt status enable
28 TDMA12	TDMA Channel 12 error interrupt status enable
27 TDMA11	TDMA Channel 11 error interrupt status enable
26 TDMA10	TDMA Channel 10 error interrupt status enable
25 TDMA9	TDMA Channel 9 error interrupt status enable
24 TDMA8	TDMA Channel 8 error interrupt status enable
23 TDMA7	TDMA Channel 7 error interrupt status enable
22 TDMA6	TDMA Channel 6 error interrupt status enable
21 TDMA5	TDMA Channel 5 error interrupt status enable
20 TDMA4	TDMA Channel 4 error interrupt status enable
19 TDMA3	TDMA Channel 3 error interrupt status enable
18 TDMA2	TDMA Channel 2 error interrupt status enable
17 TDMA1	TDMA Channel 1 error interrupt status enable
16 TDMA0	TDMA Channel 0 error interrupt status enable
15 RDMA15	RDMA Channel 15 error interrupt status enable
14 RDMA14	RDMA Channel 14 error interrupt status enable
13 RDMA13	RDMA Channel 13 error interrupt status enable
12 RDMA12	RDMA Channel 12 error interrupt status enable
11 RDMA11	RDMA Channel 11 error interrupt status enable
10 RDMA10	RDMA Channel 10 error interrupt status enable
9 RDMA9	RDMA Channel 9 error interrupt status enable

Table continues on the next page...

MIPI_HSI_DMA_ERR_IRQSIG_EN field descriptions (continued)

Field	Description
8 RDMA8	RDMA Channel 8 error interrupt status enable
7 RDMA7	RDMA Channel 7 error interrupt status enable
6 RDMA6	RDMA Channel 6 error interrupt status enable
5 RDMA5	RDMA Channel 5 error interrupt status enable
4 RDMA4	RDMA Channel 4 error interrupt status enable
3 RDMA3	RDMA Channel 3 error interrupt status enable
2 RDMA2	RDMA Channel 2 error interrupt status enable
1 RDMA1	RDMA Channel 1 error interrupt status enable
0 RDMA0	RDMA Channel 0 error interrupt status enable

43.5.34 DMA Single Request Enable Register (MIPI_HSI_DMA_SINGLE_REQ_EN)

This Register is used to debug

Address: 220_8000h base + 1F4h offset = 220_81F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TDMA15	TDMA14	TDMA13	TDMA12	TDMA11	TDMA10	TDMA9	TDMA8	TDMA7	TDMA6	TDMA5	TDMA4	TDMA3	TDMA2	TDMA1	TDMA0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDMA15	RDMA14	RDMA13	RDMA12	RDMA11	RDMA10	RDMA9	RDMA8	RDMA7	RDMA6	RDMA5	RDMA4	RDMA3	RDMA2	RDMA1	RDMA0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_HSI_DMA_SINGLE_REQ_EN field descriptions

Field	Description
31 TDMA15	When the remain DMA data less than one DMA burst size in Tx DMA Channel 15, this bit will be set automatically
30 TDMA14	When the remain DMA data less than one DMA burst size in Tx DMA Channel 14, this bit will be set automatically

Table continues on the next page...

MIPI_HSI_DMA_SINGLE_REQ_EN field descriptions (continued)

Field	Description
29 TDMA13	When the remain DMA data less than one DMA burst size in Tx DMA Channel 13, this bit will be set automatically
28 TDMA12	When the remain DMA data less than one DMA burst size in Tx DMA Channel 12, this bit will be set automatically
27 TDMA11	When the remain DMA data less than one DMA burst size in Tx DMA Channel 11, this bit will be set automatically
26 TDMA10	When the remain DMA data less than one DMA burst size in Tx DMA Channel 10, this bit will be set automatically
25 TDMA9	When the remain DMA data less than one DMA burst size in Tx DMA Channel 9, this bit will be set automatically
24 TDMA8	When the remain DMA data less than one DMA burst size in Tx DMA Channel 8, this bit will be set automatically
23 TDMA7	When the remain DMA data less than one DMA burst size in Tx DMA Channel 7, this bit will be set automatically
22 TDMA6	When the remain DMA data less than one DMA burst size in Tx DMA Channel 6, this bit will be set automatically
21 TDMA5	When the remain DMA data less than one DMA burst size in Tx DMA Channel 5, this bit will be set automatically
20 TDMA4	When the remain DMA data less than one DMA burst size in Tx DMA Channel 4, this bit will be set automatically
19 TDMA3	When the remain DMA data less than one DMA burst size in Tx DMA Channel 3, this bit will be set automatically
18 TDMA2	When the remain DMA data less than one DMA burst size in Tx DMA Channel 2, this bit will be set automatically
17 TDMA1	When the remain DMA data less than one DMA burst size in Tx DMA Channel 1, this bit will be set automatically
16 TDMA0	When the remain DMA data less than one DMA burst size in Tx DMA Channel 0, this bit will be set automatically
15 RDMA15	When the remain DMA data less than one DMA burst size in Rx DMA Channel 15, this bit will be set automatically
14 RDMA14	When the remain DMA data less than one DMA burst size in Rx DMA Channel 14, this bit will be set automatically
13 RDMA13	When the remain DMA data less than one DMA burst size in Rx DMA Channel 13, this bit will be set automatically
12 RDMA12	When the remain DMA data less than one DMA burst size in Rx DMA Channel 12, this bit will be set automatically
11 RDMA11	When the remain DMA data less than one DMA burst size in Rx DMA Channel 11, this bit will be set automatically
10 RDMA10	When the remain DMA data less than one DMA burst size in Rx DMA Channel 10, this bit will be set automatically
9 RDMA9	When the remain DMA data less than one DMA burst size in Rx DMA Channel 9, this bit will be set automatically
8 RDMA8	When the remain DMA data less than one DMA burst size in Rx DMA Channel 8, this bit will be set automatically
7 RDMA7	When the remain DMA data less than one DMA burst size in Rx DMA Channel 7, this bit will be set automatically

Table continues on the next page...

MIPI_HSI_DMA_SINGLE_REQ_EN field descriptions (continued)

Field	Description
6 RDMA6	When the remain DMA data less than one DMA burst size in Rx DMA Channel 6, this bit will be set automatically
5 RDMA5	When the remain DMA data less than one DMA burst size in Rx DMA Channel 5, this bit will be set automatically
4 RDMA4	When the remain DMA data less than one DMA burst size in Rx DMA Channel 4, this bit will be set automatically
3 RDMA3	When the remain DMA data less than one DMA burst size in Rx DMA Channel 3, this bit will be set automatically
2 RDMA2	When the remain DMA data less than one DMA burst size in Rx DMA Channel 2, this bit will be set automatically
1 RDMA1	When the remain DMA data less than one DMA burst size in Rx DMA Channel 1, this bit will be set automatically
0 RDMA0	When the remain DMA data less than one DMA burst size in Rx DMA Channel 0, this bit will be set automatically

43.5.35 Tx Fifo Size Configuration Register 0 (MIPI_HSI_TX_FIFO_SIZE_CONF0)

This register is used to configure each Tx fifo size

Address: 220_8000h base + 200h offset = 220_8200h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH15				CH14				CH13				CH12				CH11				CH10				CH9				CH8			
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

MIPI_HSI_TX_FIFO_SIZE_CONF0 field descriptions

Field	Description
31–28 CH15	This field is used to set the buffer size for channel 15. All the allowed combinations of bit setting are listed here . 0000 channel 15 buffer size is 1Dword 0001 channel 15 buffer size is 2Dwords 0010 channel 15 buffer size is 4Dwords 0011 channel 15 buffer size is 8Dwords 0100 channel 15 buffer size is 16Dwords 0101 channel 15 buffer size is 32Dwords 0110 channel 15 buffer size is 64Dwords 0111 channel 15 buffer size is 128Dwords 1000 channel 15 buffer size is 256Dwords 1001 channel 15 buffer size is 512Dwords

Table continues on the next page...

MIPI_HSI_TX_FIFO_SIZE_CONF0 field descriptions (continued)

Field	Description
	1010 channel 15 buffer size is 1024Dwords 1111-1011 Reserved
27–24 CH14	This field is used to set the buffer size for channel 14. All the allowed combinations of bit setting are listed here . 0000 channel 14 buffer size is 1Dword 0001 channel 14 buffer size is 2Dwords 0010 channel 14 buffer size is 4Dwords 0011 channel 14 buffer size is 8Dwords 0100 channel 14 buffer size is 16Dwords 0101 channel 14 buffer size is 32Dwords 0110 channel 14 buffer size is 64Dwords 0111 channel 14 buffer size is 128Dwords 1000 channel 14 buffer size is 256Dwords 1001 channel 14 buffer size is 512Dwords 1010 channel 14 buffer size is 1024Dwords 1111-1011 Reserved
23–20 CH13	This field is used to set the buffer size for channel 13. All the allowed combinations of bit setting are listed here . 0000 channel 13 buffer size is 1Dword 0001 channel 13 buffer size is 2Dwords 0010 channel 13 buffer size is 4Dwords 0011 channel 13 buffer size is 8Dwords 0100 channel 13 buffer size is 16Dwords 0101 channel 13 buffer size is 32Dwords 0110 channel 13 buffer size is 64Dwords 0111 channel 13 buffer size is 128Dwords 1000 channel 13 buffer size is 256Dwords 1001 channel 13 buffer size is 512Dwords 1010 channel 13 buffer size is 1024Dwords 1111-1011 Reserved
19–16 CH12	This field is used to set the buffer size for channel 12. All the allowed combinations of bit setting are listed here . 0000 channel 12 buffer size is 1Dword 0001 channel 12 buffer size is 2Dwords 0010 channel 12 buffer size is 4Dwords 0011 channel 12 buffer size is 8Dwords 0100 channel 12 buffer size is 16Dwords 0101 channel 12 buffer size is 32Dwords 0110 channel 12 buffer size is 64Dwords 0111 channel 12 buffer size is 128Dwords 1000 channel 12 buffer size is 256Dwords 1001 channel 12 buffer size is 512Dwords 1010 channel 12 buffer size is 1024Dwords 1111-1011 Reserved

Table continues on the next page...

MIPI_HSI_TX_FIFO_SIZE_CONF0 field descriptions (continued)

Field	Description
15–12 CH11	This field is used to set the buffer size for channel 11. All the allowed combinations of bit setting are listed here . 0000 channel 11 buffer size is 1Dword 0001 channel 11 buffer size is 2Dwords 0010 channel 11 buffer size is 4Dwords 0011 channel 11 buffer size is 8Dwords 0100 channel 11 buffer size is 16Dwords 0101 channel 11 buffer size is 32Dwords 0110 channel 11 buffer size is 64Dwords 0111 channel 11 buffer size is 128Dwords 1000 channel 11 buffer size is 256Dwords 1001 channel 11 buffer size is 512Dwords 1010 channel 11 buffer size is 1024Dwords 1111-1011 Reserved
11–8 CH10	This field is used to set the buffer size for channel 10. All the allowed combinations of bit setting are listed here . 0000 channel 10 buffer size is 1Dword 0001 channel 10 buffer size is 2Dwords 0010 channel 10 buffer size is 4Dwords 0011 channel 10 buffer size is 8Dwords 0100 channel 10 buffer size is 16Dwords 0101 channel 10 buffer size is 32Dwords 0110 channel 10 buffer size is 64Dwords 0111 channel 10 buffer size is 128Dwords 1000 channel 10 buffer size is 256Dwords 1001 channel 10 buffer size is 512Dwords 1010 channel 10 buffer size is 1024Dwords 1111-1011 Reserved
7–4 CH9	This field is used to set the buffer size for channel 9. All the allowed combinations of bit setting are listed here . 0000 channel 9 buffer size is 1Dword 0001 channel 9 buffer size is 2Dwords 0010 channel 9 buffer size is 4Dwords 0011 channel 9 buffer size is 8Dwords 0100 channel 9 buffer size is 16Dwords 0101 channel 9 buffer size is 32Dwords 0110 channel 9 buffer size is 64Dwords 0111 channel 9 buffer size is 128Dwords 1000 channel 9 buffer size is 256Dwords 1001 channel 9 buffer size is 512Dwords 1010 channel 9 buffer size is 1024Dwords 1111-1011 Reserved
CH8	This field is used to set the buffer size for channel 8.

Table continues on the next page...

MIPI_HSI_TX_FIFO_SIZE_CONF0 field descriptions (continued)

Field	Description
	All the allowed combinations of bit setting are listed here . 0000 channel 8 buffer size is 1Dword 0001 channel 8 buffer size is 2Dwords 0010 channel 8 buffer size is 4Dwords 0011 channel 8 buffer size is 8Dwords 0100 channel 8 buffer size is 16Dwords 0101 channel 8 buffer size is 32Dwords 0110 channel 8 buffer size is 64Dwords 0111 channel 8 buffer size is 128Dwords 1000 channel 8 buffer size is 256Dwords 1001 channel 8 buffer size is 512Dwords 1010 channel 8 buffer size is 1024Dwords 1111-1011 Reserved

43.5.36 Tx Fifo Size Configuration Register 1 (MIPI_HSI_TX_FIFO_SIZE_CONF1)

This register is used to configure each Tx fifo size

Address: 220_8000h base + 204h offset = 220_8204h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH7				CH6				CH5				CH4				CH3				CH2				CH1				CH0			
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

MIPI_HSI_TX_FIFO_SIZE_CONF1 field descriptions

Field	Description
31–28 CH7	This field is used to set the buffer size for channel 7. All the allowed combinations of bit setting are listed here . 0000 channel 7 buffer size is 1Dword 0001 channel 7 buffer size is 2Dwords 0010 channel 7 buffer size is 4Dwords 0011 channel 7 buffer size is 8Dwords 0100 channel 7 buffer size is 16Dwords 0101 channel 7 buffer size is 32Dwords 0110 channel 7 buffer size is 64Dwords 0111 channel 7 buffer size is 128Dwords 1000 channel 7 buffer size is 256Dwords 1001 channel 7 buffer size is 512Dwords

Table continues on the next page...

MIPI_HSI_TX_FIFO_SIZE_CONF1 field descriptions (continued)

Field	Description
	1010 channel 7 buffer size is 1024Dwords 1111-1011 Reserved
27–24 CH6	This field is used to set the buffer size for channel 6. All the allowed combinations of bit setting are listed here . 0000 channel 6 buffer size is 1Dword 0001 channel 6 buffer size is 2Dwords 0010 channel 6 buffer size is 4Dwords 0011 channel 6 buffer size is 8Dwords 0100 channel 6 buffer size is 16Dwords 0101 channel 6 buffer size is 32Dwords 0110 channel 6 buffer size is 64Dwords 0111 channel 6 buffer size is 128Dwords 1000 channel 6 buffer size is 256Dwords 1001 channel 6 buffer size is 512Dwords 1010 channel 6 buffer size is 1024Dwords 1111-1011 Reserved
23–20 CH5	This field is used to set the buffer size for channel 5. All the allowed combinations of bit setting are listed here . 0000 channel 5 buffer size is 1Dword 0001 channel 5 buffer size is 2Dwords 0010 channel 5 buffer size is 4Dwords 0011 channel 5 buffer size is 8Dwords 0100 channel 5 buffer size is 16Dwords 0101 channel 5 buffer size is 32Dwords 0110 channel 5 buffer size is 64Dwords 0111 channel 5 buffer size is 128Dwords 1000 channel 5 buffer size is 256Dwords 1001 channel 5 buffer size is 512Dwords 1010 channel 5 buffer size is 1024Dwords 1111-1011 Reserved
19–16 CH4	This field is used to set the buffer size for channel 4. All the allowed combinations of bit setting are listed here . 0000 channel 4 buffer size is 1Dword 0001 channel 4 buffer size is 2Dwords 0010 channel 4 buffer size is 4Dwords 0011 channel 4 buffer size is 8Dwords 0100 channel 4 buffer size is 16Dwords 0101 channel 4 buffer size is 32Dwords 0110 channel 4 buffer size is 64Dwords 0111 channel 4 buffer size is 128Dwords 1000 channel 4 buffer size is 256Dwords 1001 channel 4 buffer size is 512Dwords 1010 channel 4 buffer size is 1024Dwords 1111-1011 Reserved

Table continues on the next page...

MIPI_HSI_TX_FIFO_SIZE_CONF1 field descriptions (continued)

Field	Description
15–12 CH3	This field is used to set the buffer size for channel 3. All the allowed combinations of bit setting are listed here . 0000 channel 3 buffer size is 1Dword 0001 channel 3 buffer size is 2Dwords 0010 channel 3 buffer size is 4Dwords 0011 channel 3 buffer size is 8Dwords 0100 channel 3 buffer size is 16Dwords 0101 channel 3 buffer size is 32Dwords 0110 channel 3 buffer size is 64Dwords 0111 channel 3 buffer size is 128Dwords 1000 channel 3 buffer size is 256Dwords 1001 channel 3 buffer size is 512Dwords 1010 channel 3 buffer size is 1024Dwords 1111-1011 Reserved
11–8 CH2	This field is used to set the buffer size for channel 2. All the allowed combinations of bit setting are listed here . 0000 channel 2 buffer size is 1Dword 0001 channel 2 buffer size is 2Dwords 0010 channel 2 buffer size is 4Dwords 0011 channel 2 buffer size is 8Dwords 0100 channel 2 buffer size is 16Dwords 0101 channel 2 buffer size is 32Dwords 0110 channel 2 buffer size is 64Dwords 0111 channel 2 buffer size is 128Dwords 1000 channel 2 buffer size is 256Dwords 1001 channel 2 buffer size is 512Dwords 1010 channel 2 buffer size is 1024Dwords 1111-1011 Reserved
7–4 CH1	This field is used to set the buffer size for channel 1. All the allowed combinations of bit setting are listed here . 0000 channel 1 buffer size is 1Dword 0001 channel 1 buffer size is 2Dwords 0010 channel 1 buffer size is 4Dwords 0011 channel 1 buffer size is 8Dwords 0100 channel 1 buffer size is 16Dwords 0101 channel 1 buffer size is 32Dwords 0110 channel 1 buffer size is 64Dwords 0111 channel 1 buffer size is 128Dwords 1000 channel 1 buffer size is 256Dwords 1001 channel 1 buffer size is 512Dwords 1010 channel 1 buffer size is 1024Dwords 1111-1011 Reserved
CH0	This field is used to set the buffer size for channel 0.

Table continues on the next page...

MIPI_HSI_TX_FIFO_SIZE_CONF1 field descriptions (continued)

Field	Description
	All the allowed combinations of bit setting are listed here . 0000 channel 0 buffer size is 1Dword 0001 channel 0 buffer size is 2Dwords 0010 channel 0 buffer size is 4Dwords 0011 channel 0 buffer size is 8Dwords 0100 channel 0 buffer size is 16Dwords 0101 channel 0 buffer size is 32Dwords 0110 channel 0 buffer size is 64Dwords 0111 channel 0 buffer size is 128Dwords 1000 channel 0 buffer size is 256Dwords 1001 channel 0 buffer size is 512Dwords 1010 channel 0 buffer size is 1024Dwords 1111-1011 Reserved

43.5.37 Rx Fifo Size Configuration Register 0 (MIPI_HSI_RX_FIFO_SIZE_CONF0)

This register is used to configure each Rx fifo size

Address: 220_8000h base + 208h offset = 220_8208h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH15				CH14				CH13				CH12				CH11				CH10				CH9				CH8			
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

MIPI_HSI_RX_FIFO_SIZE_CONF0 field descriptions

Field	Description
31–28 CH15	This field is used to set the buffer size for channel 15. All the allowed combinations of bit setting are listed here . 0000 channel 15 buffer size is 1Dword 0001 channel 15 buffer size is 2Dwords 0010 channel 15 buffer size is 4Dwords 0011 channel 15 buffer size is 8Dwords 0100 channel 15 buffer size is 16Dwords 0101 channel 15 buffer size is 32Dwords 0110 channel 15 buffer size is 64Dwords 0111 channel 15 buffer size is 128Dwords 1000 channel 15 buffer size is 256Dwords 1001 channel 15 buffer size is 512Dwords

Table continues on the next page...

MIPI_HSI_RX_FIFO_SIZE_CONF0 field descriptions (continued)

Field	Description
	1010 channel 15 buffer size is 1024Dwords 1111-1011 Reserved
27–24 CH14	This field is used to set the buffer size for channel 14. All the allowed combinations of bit setting are listed here . 0000 channel 14 buffer size is 1Dword 0001 channel 14 buffer size is 2Dwords 0010 channel 14 buffer size is 4Dwords 0011 channel 14 buffer size is 8Dwords 0100 channel 14 buffer size is 16Dwords 0101 channel 14 buffer size is 32Dwords 0110 channel 14 buffer size is 64Dwords 0111 channel 14 buffer size is 128Dwords 1000 channel 14 buffer size is 256Dwords 1001 channel 14 buffer size is 512Dwords 1010 channel 14 buffer size is 1024Dwords 1111-1011 Reserved
23–20 CH13	This field is used to set the buffer size for channel 13. All the allowed combinations of bit setting are listed here . 0000 channel 13 buffer size is 1Dword 0001 channel 13 buffer size is 2Dwords 0010 channel 13 buffer size is 4Dwords 0011 channel 13 buffer size is 8Dwords 0100 channel 13 buffer size is 16Dwords 0101 channel 13 buffer size is 32Dwords 0110 channel 13 buffer size is 64Dwords 0111 channel 13 buffer size is 128Dwords 1000 channel 13 buffer size is 256Dwords 1001 channel 13 buffer size is 512Dwords 1010 channel 13 buffer size is 1024Dwords 1111 b1011 Reserved
19–16 CH12	This field is used to set the buffer size for channel 12. All the allowed combinations of bit setting are listed here . 0000 channel 12 buffer size is 1Dword 0001 channel 12 buffer size is 2Dwords 0010 channel 12 buffer size is 4Dwords 0011 channel 12 buffer size is 8Dwords 0100 channel 12 buffer size is 16Dwords 0101 channel 12 buffer size is 32Dwords 0110 channel 12 buffer size is 64Dwords 0111 channel 12 buffer size is 128Dwords 1000 channel 12 buffer size is 256Dwords 1001 channel 12 buffer size is 512Dwords 1010 channel 12 buffer size is 1024Dwords 1111-1011 Reserved

Table continues on the next page...

MIPI_HSI_RX_FIFO_SIZE_CONF0 field descriptions (continued)

Field	Description
15–12 CH11	This field is used to set the buffer size for channel 11. All the allowed combinations of bit setting are listed here . 0000 channel 11 buffer size is 1Dword 0001 channel 11 buffer size is 2Dwords 0010 channel 11 buffer size is 4Dwords 0011 channel 11 buffer size is 8Dwords 0100 channel 11 buffer size is 16Dwords 0101 channel 11 buffer size is 32Dwords 0110 channel 11 buffer size is 64Dwords 0111 channel 11 buffer size is 128Dwords 1000 channel 11 buffer size is 256Dwords 1001 channel 11 buffer size is 512Dwords 1010 channel 11 buffer size is 1024Dwords 1111-1011 Reserved
11–8 CH10	This field is used to set the buffer size for channel 10. All the allowed combinations of bit setting are listed here . 0000 channel 10 buffer size is 1Dword 0001 channel 10 buffer size is 2Dwords 0010 channel 10 buffer size is 4Dwords 0011 channel 10 buffer size is 8Dwords 0100 channel 10 buffer size is 16Dwords 0101 channel 10 buffer size is 32Dwords 0110 channel 10 buffer size is 64Dwords 0111 channel 10 buffer size is 128Dwords 1000 channel 10 buffer size is 256Dwords 1001 channel 10 buffer size is 512Dwords 1010 channel 10 buffer size is 1024Dwords 1111-1011 Reserved
7–4 CH9	This field is used to set the buffer size for channel 9. All the allowed combinations of bit setting are listed here . 0000 channel 9 buffer size is 1Dword 0001 channel 9 buffer size is 2Dwords 0010 channel 9 buffer size is 4Dwords 0011 channel 9 buffer size is 8Dwords 0100 channel 9 buffer size is 16Dwords 0101 channel 9 buffer size is 32Dwords 0110 channel 9 buffer size is 64Dwords 0111 channel 9 buffer size is 128Dwords 1000 channel 9 buffer size is 256Dwords 1001 channel 9 buffer size is 512Dwords 1010 channel 9 buffer size is 1024Dwords 1111-1011 Reserved
CH8	This field is used to set the buffer size for channel 8.

Table continues on the next page...

MIPI_HSI_RX_FIFO_SIZE_CONF0 field descriptions (continued)

Field	Description
	All the allowed combinations of bit setting are listed here . 0000 channel 8 buffer size is 1Dword 0001 channel 8 buffer size is 2Dwords 0010 channel 8 buffer size is 4Dwords 0011 channel 8 buffer size is 8Dwords 0100 channel 8 buffer size is 16Dwords 0101 channel 8 buffer size is 32Dwords 0110 channel 8 buffer size is 64Dwords 0111 channel 8 buffer size is 128Dwords 1000 channel 8 buffer size is 256Dwords 1001 channel 8 buffer size is 512Dwords 1010 channel 8 buffer size is 1024Dwords 1111-1011 Reserved

43.5.38 Rx Fifo Size Configuration Register 1 (MIPI_HSI_RX_FIFO_SIZE_CONF1)

This register is used to configure each Rx fifo size

Address: 220_8000h base + 20Ch offset = 220_820Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH7				CH6				CH5				CH4				CH3				CH2				CH1				CH0			
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

MIPI_HSI_RX_FIFO_SIZE_CONF1 field descriptions

Field	Description
31–28 CH7	This field is used to set the buffer size for channel 7. All the allowed combinations of bit setting are listed here . 0000 channel 7 buffer size is 1Dword 0001 channel 7 buffer size is 2Dwords 0010 channel 7 buffer size is 4Dwords 0011 channel 7 buffer size is 8Dwords 0100 channel 7 buffer size is 16Dwords 0101 channel 7 buffer size is 32Dwords 0110 channel 7 buffer size is 64Dwords 0111 channel 7 buffer size is 128Dwords 1000 channel 7 buffer size is 256Dwords 1001 channel 7 buffer size is 512Dwords

Table continues on the next page...

MIPI_HSI_RX_FIFO_SIZE_CONF1 field descriptions (continued)

Field	Description
	1010 channel 7 buffer size is 1024Dwords 1111-1011 Reserved
27–24 CH6	This field is used to set the buffer size for channel 6. All the allowed combinations of bit setting are listed here . 0000 channel 6 buffer size is 1Dword 0001 channel 6 buffer size is 2Dwords 0010 channel 6 buffer size is 4Dwords 0011 channel 6 buffer size is 8Dwords 0100 channel 6 buffer size is 16Dwords 0101 channel 6 buffer size is 32Dwords 0110 channel 6 buffer size is 64Dwords 0111 channel 6 buffer size is 128Dwords 1000 channel 6 buffer size is 256Dwords 1001 channel 6 buffer size is 512Dwords 1010 channel 6 buffer size is 1024Dwords 1111-1011 Reserved
23–20 CH5	This field is used to set the buffer size for channel 5. All the allowed combinations of bit setting are listed here . 0000 channel 5 buffer size is 1Dword 0001 channel 5 buffer size is 2Dwords 0010 channel 5 buffer size is 4Dwords 0011 channel 5 buffer size is 8Dwords 0100 channel 5 buffer size is 16Dwords 0101 channel 5 buffer size is 32Dwords 0110 channel 5 buffer size is 64Dwords 0111 channel 5 buffer size is 128Dwords 1000 channel 5 buffer size is 256Dwords 1001 channel 5 buffer size is 512Dwords 1010 channel 5 buffer size is 1024Dwords 1111-1011 Reserved
19–16 CH4	This field is used to set the buffer size for channel 4. All the allowed combinations of bit setting are listed here . 0000 channel 4 buffer size is 1Dword 0001 channel 4 buffer size is 2Dwords 0010 channel 4 buffer size is 4Dwords 0011 channel 4 buffer size is 8Dwords 0100 channel 4 buffer size is 16Dwords 0101 channel 4 buffer size is 32Dwords 0110 channel 4 buffer size is 64Dwords 0111 channel 4 buffer size is 128Dwords 1000 channel 4 buffer size is 256Dwords 1001 channel 4 buffer size is 512Dwords 1010 channel 4 buffer size is 1024Dwords 1111-1011 Reserved

Table continues on the next page...

MIPI_HSI_RX_FIFO_SIZE_CONF1 field descriptions (continued)

Field	Description
15–12 CH3	This field is used to set the buffer size for channel 3. All the allowed combinations of bit setting are listed here . 0000 channel 3 buffer size is 1Dword 0001 channel 3 buffer size is 2Dwords 0010 channel 3 buffer size is 4Dwords 0011 channel 3 buffer size is 8Dwords 0100 channel 3 buffer size is 16Dwords 0101 channel 3 buffer size is 32Dwords 0110 channel 3 buffer size is 64Dwords 0111 channel 3 buffer size is 128Dwords 1000 channel 3 buffer size is 256Dwords 1001 channel 3 buffer size is 512Dwords 1010 channel 3 buffer size is 1024Dwords 1111-1011 Reserved
11–8 CH2	This field is used to set the buffer size for channel 2. All the allowed combinations of bit setting are listed here . 0000 channel 2 buffer size is 1Dword 0001 channel 2 buffer size is 2Dwords 0010 channel 2 buffer size is 4Dwords 0011 channel 2 buffer size is 8Dwords 0100 channel 2 buffer size is 16Dwords 0101 channel 2 buffer size is 32Dwords 0110 channel 2 buffer size is 64Dwords 0111 channel 2 buffer size is 128Dwords 1000 channel 2 buffer size is 256Dwords 1001 channel 2 buffer size is 512Dwords 1010 channel 2 buffer size is 1024Dwords 1111-1011 Reserved
7–4 CH1	This field is used to set the buffer size for channel 1. All the allowed combinations of bit setting are listed here . 0000 channel 1 buffer size is 1Dword 0001 channel 1 buffer size is 2Dwords 0010 channel 1 buffer size is 4Dwords 0011 channel 1 buffer size is 8Dwords 0100 channel 1 buffer size is 16Dwords 0101 channel 1 buffer size is 32Dwords 0110 channel 1 buffer size is 64Dwords 0111 channel 1 buffer size is 128Dwords 1000 channel 1 buffer size is 256Dwords 1001 channel 1 buffer size is 512Dwords 1010 channel 1 buffer size is 1024Dwords 1111-1011 Reserved
CH0	This field is used to set the buffer size for channel 0.

Table continues on the next page...

MIPI_HSI_RX_FIFO_SIZE_CONF1 field descriptions (continued)

Field	Description
	All the allowed combinations of bit setting are listed here . 0000 channel 0 buffer size is 1Dword 0001 channel 0 buffer size is 2Dwords 0010 channel 0 buffer size is 4Dwords 0011 channel 0 buffer size is 8Dwords 0100 channel 0 buffer size is 16Dwords 0101 channel 0 buffer size is 32Dwords 0110 channel 0 buffer size is 64Dwords 0111 channel 0 buffer size is 128Dwords 1000 channel 0 buffer size is 256Dwords 1001 channel 0 buffer size is 512Dwords 1010 channel 0 buffer size is 1024Dwords 1111-1011 Reserved

43.5.39 Tx Fifo Status Register (MIPI_HSI_TX_FIFO_STAT)

This register contains full and empty status for each Tx channel fifo

Address: 220_8000h base + 210h offset = 220_8210h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CH15		CH14		CH13		CH12		CH11		CH10		CH9		CH8	
W																
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH7		CH6		CH5		CH4		CH3		CH2		CH1		CH0	
W																
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

MIPI_HSI_TX_FIFO_STAT field descriptions

Field	Description
31–30 CH15	00 Tx channel 15 fifo not Empty and Full; 01 Tx channel 15 fifo Empty; 10 Tx channel 15 fifo Full; 11 Reserved.
29–28 CH14	00 Tx channel 14 fifo not Empty and Full; 01 Tx channel 14 fifo Empty; 10 Tx channel 14 fifo Full; 11 Reserved.

Table continues on the next page...

MIPI_HSI_TX_FIFO_STAT field descriptions (continued)

Field	Description
27–26 CH13	00 Tx channel 13 fifo not Empty and Full; 01 Tx channel 13 fifo Empty; 10 Tx channel 13 fifo Full; 11 Reserved.
25–24 CH12	00 Tx channel 12 fifo not Empty and Full; 01 Tx channel 12 fifo Empty; 10 Tx channel 12 fifo Full; 11 Reserved.
23–22 CH11	00 Tx channel 11 fifo not Empty and Full; 01 Tx channel 11 fifo Empty; 10 Tx channel 11 fifo Full; 11 Reserved.
21–20 CH10	00 Tx channel 10 fifo not Empty and Full; 01 Tx channel 10 fifo Empty; 10 Tx channel 10 fifo Full; 11 Reserved.
19–18 CH9	00 Tx channel 9 fifo not Empty and Full; 01 Tx channel 9 fifo Empty; 10 Tx channel 9 fifo Full; 11 Reserved.
17–16 CH8	00 Tx channel 8 fifo not Empty and Full; 01 Tx channel 8 fifo Empty; 10 Tx channel 8 fifo Full; 11 Reserved.
15–14 CH7	00 Tx channel 7 fifo not Empty and Full; 01 Tx channel 7 fifo Empty; 10 Tx channel 7 fifo Full; 11 Reserved.
13–12 CH6	00 Tx channel 6 fifo not Empty and Full; 01 Tx channel 6 fifo Empty; 10 Tx channel 6 fifo Full; 11 Reserved.
11–10 CH5	00 Tx channel 5 fifo not Empty and Full; 01 Tx channel 5 fifo Empty; 10 Tx channel 5 fifo Full; 11 Reserved.
9–8 CH4	00 Tx channel 4 fifo not Empty and Full; 01 Tx channel 4 fifo Empty; 10 Tx channel 4 fifo Full; 11 Reserved.
7–6 CH3	00 Tx channel 3 fifo not Empty and Full; 01 Tx channel 3 fifo Empty; 10 Tx channel 3 fifo Full; 11 Reserved.

Table continues on the next page...

MIPI_HSI_TX_FIFO_STAT field descriptions (continued)

Field	Description
5–4 CH2	00 Tx channel 2 fifo not Empty and Full; 01 Tx channel 2 fifo Empty; 10 Tx channel 2 fifo Full; 11 Reserved.
3–2 CH1	00 Tx channel 1 fifo not Empty and Full; 01 Tx channel 1 fifo Empty; 10 Tx channel 1 fifo Full; 11 Reserved.
CH0	00 Tx channel 0 fifo not Empty and Full; 01 Tx channel 0 fifo Empty; 10 Tx channel 0 fifo Full; 11 Reserved.

43.5.40 Rx Fifo Status Register (MIPI_HSI_RX_FIFO_STAT)

This register contains full and empty status for each Rx channel fifo

Address: 220_8000h base + 214h offset = 220_8214h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CH15		CH14		CH13		CH12		CH11		CH10		CH9		CH8	
W																
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH7		CH6		CH5		CH4		CH3		CH2		CH1		CH0	
W																
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

MIPI_HSI_RX_FIFO_STAT field descriptions

Field	Description
31–30 CH15	00 Rx channel 15 fifo not Empty and Full; 01 Rx channel 15 fifo Empty; 10 Rx channel 15 fifo Full; 11 Reserved.
29–28 CH14	00 Rx channel 14 fifo not Empty and Full; 01 Rx channel 14 fifo Empty; 10 Rx channel 14 fifo Full; 11 Reserved.

Table continues on the next page...

MIPI_HSI_RX_FIFO_STAT field descriptions (continued)

Field	Description
27–26 CH13	00 Rx channel 13 fifo not Empty and Full; 01 Rx channel 13 fifo Empty; 10 Rx channel 13 fifo Full; 11 Reserved.
25–24 CH12	00 Rx channel 12 fifo not Empty and Full; 01 Rx channel 12 fifo Empty; 10 Rx channel 12 fifo Full; 11 Reserved.
23–22 CH11	00 Rx channel 11 fifo not Empty and Full; 01 Rx channel 11 fifo Empty; 10 Rx channel 11 fifo Full; 11 Reserved.
21–20 CH10	00 Rx channel 10 fifo not Empty and Full; 01 Rx channel 10 fifo Empty; 10 Rx channel 10 fifo Full; 11 Reserved.
19–18 CH9	00 Rx channel 9 fifo not Empty and Full; 01 Rx channel 9 fifo Empty; 10 Rx channel 9 fifo Full; 11 Reserved.
17–16 CH8	00 Rx channel 8 fifo not Empty and Full; 01 Rx channel 8 fifo Empty; 10 Rx channel 8 fifo Full; 11 Reserved.
15–14 CH7	00 Rx channel 7 fifo not Empty and Full; 01 Rx channel 7 fifo Empty; 10 Rx channel 7 fifo Full; 11 Reserved.
13–12 CH6	00 Rx channel 6 fifo not Empty and Full; 01 Rx channel 6 fifo Empty; 10 Rx channel 6 fifo Full; 11 Reserved.
11–10 CH5	00 Rx channel 5 fifo not Empty and Full; 01 Rx channel 5 fifo Empty; 10 Rx channel 5 fifo Full; 11 Reserved.
9–8 CH4	00 Rx channel 4 fifo not Empty and Full; 01 Rx channel 4 fifo Empty; 10 Rx channel 4 fifo Full; 11 Reserved.
7–6 CH3	00 Rx channel 3 fifo not Empty and Full; 01 Rx channel 3 fifo Empty; 10 Rx channel 3 fifo Full; 11 Reserved.

Table continues on the next page...

MIPI_HSI_RX_FIFO_STAT field descriptions (continued)

Field	Description
5–4 CH2	00 Rx channel 2 fifo not Empty and Full; 01 Rx channel 2 fifo Empty; 10 Rx channel 2 fifo Full; 11 Reserved.
3–2 CH1	00 Rx channel 1 fifo not Empty and Full; 01 Rx channel 1 fifo Empty; 10 Rx channel 1 fifo Full; 11 Reserved.
CH0	00 Rx channel 0 fifo not Empty and Full; 01 Rx channel 0 fifo Empty; 10 Rx channel 0 fifo Full; 11 Reserved.

43.5.41 AHB Master Config Register (MIPI_HSI_AHB_MASTER_CONF)

This register used to configure hsi internal ahb master

Address: 220_8000h base + 228h offset = 220_8228h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						DP_HOLD_CYCLE				DMA_MODE		DMA_INSERT_IDLE_NUM			
Reset	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

MIPI_HSI_AHB_MASTER_CONF field descriptions

Field	Description
31–10 Reserved	This field is reserved. Reserved, always set to zero.
9–6 DP_HOLD_ CYCLE	These bits used to set the number of cycles for DP access fifo.
5–4 DMA_MODE	00 Once AHB master get hgrant from bus, it will set htrans "IDLE" for several ahb cycles. In the several cycles, once it found dataport is accessing fifo, it will release bus. 01 Once AHB master get hgrant from bus, it will set htrans "IDLE" for several ahb cycles. After the several cycles, once it found dataport is accessing fifo, it will keep on sending "IDLE" out until dataport finish accessing fifo. 1x Once AHB master get hgrant from bus, dataport can not access fifo until a dma operation done.

Table continues on the next page...

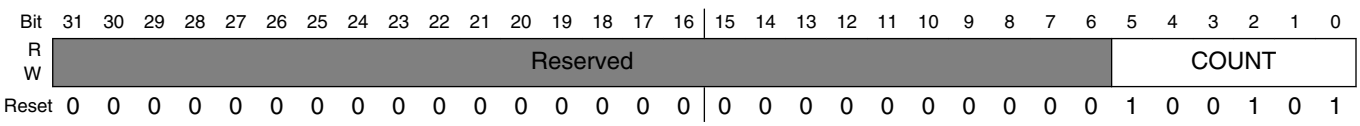
MIPI_HSI_AHB_MASTER_CONF field descriptions (continued)

Field	Description
DMA_INSERT_IDLE_NUM	These bits used to set the number of "IDLE" cycles when DMA_MODE == 2'b0x.

43.5.42 TX Break Length Register (MIPI_HSI_TX_BREAK_LEN)

This register used to set tx break length

Address: 220_8000h base + 22Ch offset = 220_822Ch



MIPI_HSI_TX_BREAK_LEN field descriptions

Field	Description
31–6 Reserved	This field is reserved. Reserved, always set to zero.
COUNT	The tx break length count. 6'h00 64 6'h01 1 6'h3f 63

Chapter 44
MediaLB (MLB)

44.1 Overview

The MLB150 implements the required functionality of a Media Local Bus (MediaLB) Device. Functionality includes:

- Transmission of commands and data when functioning as the transmitting device associated with a ChannelAddress
- Reception of data and transmission of RxStatus responses when functioning as the receiving device associated with a ChannelAddress
- MediaLB lock detection
- SystemChannel command handling

MediaLB Device functionality is implemented with an MediaLB 3-pin interface (single-ended) or MediaLB 6-pin interface (differential), however only one interface can be active at a time. The MediaLB interfaces are capable of exchanging data at speeds up to 1024xFs in 3-pin mode or 6144xFs in 6-pin mode.

A set of physical channels for exchanging data over the MediaLB bus is supported. These physical channels (4 bytes in length, or a quadlet) can be grouped into logical channels, where each logical channel is referenced using a ChannelAddress and represents a unidirectional data path between a specific MediaLB Device transmitting the data and the MediaLB Device(s) receiving the data. The MediaLB 6-pin interface provides support for up to 860 bytes of data per frame. The logical channels, configured by system software, can be any combination of channel types (synchronous, asynchronous, isochronous, or control) and direction (transmit or receive).

44.1.1 Block Diagram

The following figure is the top-level block diagram of the MLB150 behavioral models.

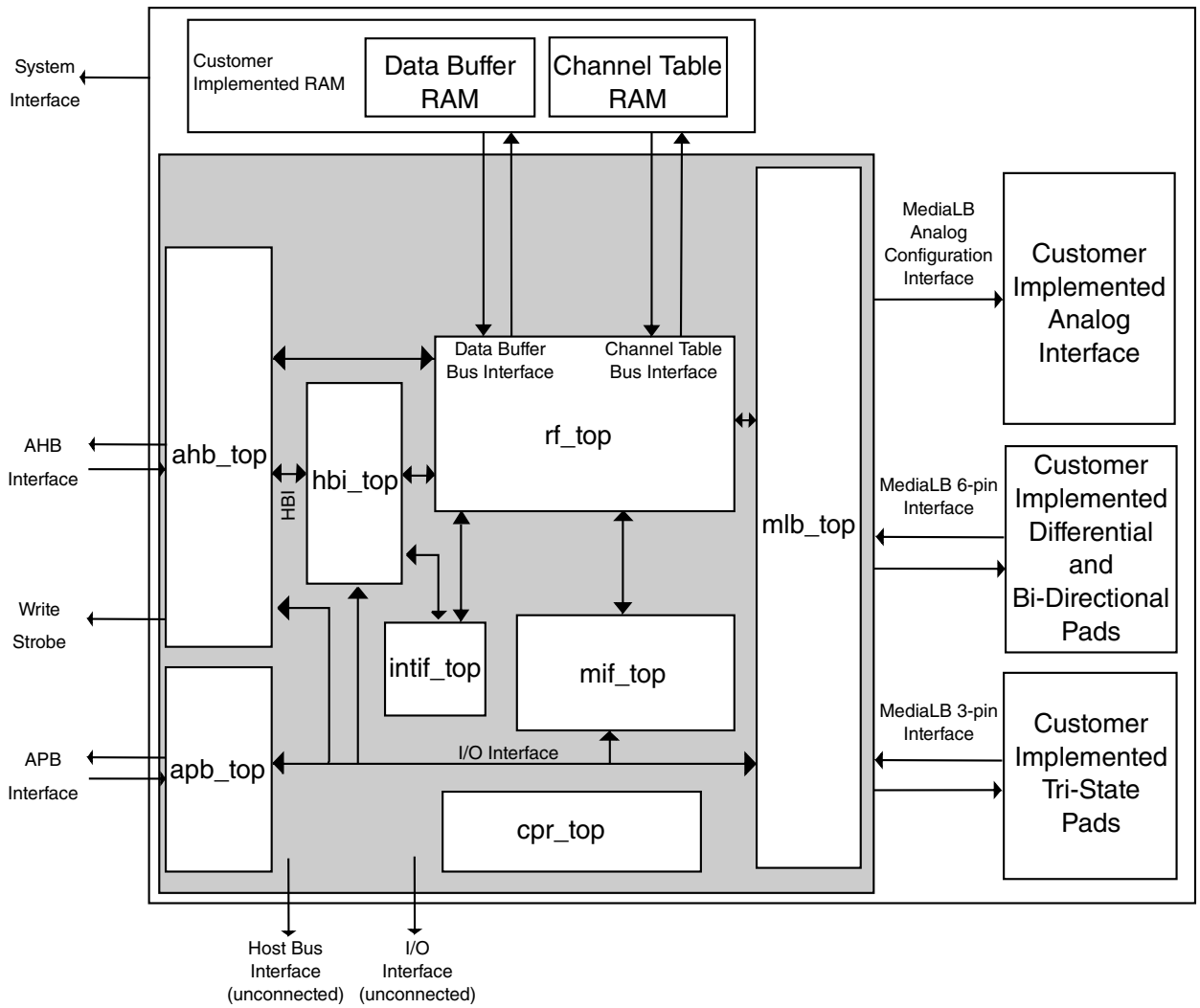


Figure 44-1. Block Diagram of MLB150

44.1.1.1 Bus Interfaces

The external bus interfaces include:

- MediaLB 3-pin Interface
- MediaLB 6-pin Interface
- MediaLB Analog Configuration Interface
- Channel Table Bus (CTB) Interface
- Data Buffer Bus (DBB) Interface
- System Interface

- AMBA Advanced High-performance Bus (AHB) Interface
- AMBA Advanced Peripheral Bus (APB) Interface

44.2 External Signals

The table found here describes the external signals of MLB150.

Table 44-1. MLB150 External Signals

Signal	Description	Pad	Mode	Direction
MLB_CLK	3-Wire clock signal	ENET_TXD1	ALT0	I
		GPIO_3	ALT7	
MLB_CLK_N	Negative clock signal	MLB_CN	No Muxing	I
MLB_CLK_P	Positive clock signal	MLB_CP	No Muxing	I
MLB_DATA	3-Wire data signal	ENET_MDC	ALT0	I/O
		GPIO_2	ALT7	
MLB_DATA_N	Negative data signal	MLB_DN	No Muxing	I/O
MLB_DATA_P	Positive data signal	MLB_DP	No Muxing	I/O
MLB_SIG	3-Wire signal	ENET_RXD1	ALT0	I/O
		GPIO_6	ALT7	
MLB_SIG_N	Negative signal	MLB_SIG_N	No Muxing	I/O
MLB_SIG_P	Positive signal	MLB_SIG_P	No Muxing	I/O

44.3 Clocks

The table found here describes the clock sources for MLB150.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 44-2. MLB150 Clocks

Clock name	Clock Root	Description
hclk	ahb_clk_root	AHB bus clock
ipg_clk_s	ipg_clk_root	Peripheral access clock
sys_clk	mlb_sys_clk	Module clock
mem_ct_CLK	mlb_sys_clk	Channel table bus clock
mem_db_CLK	mlb_sys_clk	Data buffer bus clock

44.4 Functional Description

This section describes the functional architecture of the MLB150.

The internal functional blocks of the MLB150 include:

- MediaLB Block (mlb_top) - Implements the physical and link-layer requirements of either a MediaLB 3-pin or 6-pin interface. Serial-to-parallel and parallel-to-serial data transformations are implemented, as well as MediaLB frame synchronization
- Host Bus Interface Block (hbi_top) - Provides 16-bit parallel slave access to all MOST channels and data types for the external Host Controller (HC). The HBI supports up to 64 independent channels with a minimum access latency of 40 ns per word and a maximum bandwidth of 400 Mbps
- Routing Fabric Block (rf_top) - Manages the flow of data between the MediaLB block and the HBI block, implementing a bus arbiter and muxing logic to the Channel Table RAM (CTR) and the Data Buffer RAM (DBR)
- Memory Interface Block (mif_top) - Implements a bridge between the I/O bus and the customer-implemented RAMs (i.e. Channel Table and Data Buffer)
- Interrupt Interface Block (intif_top) - Sends notifications to HBI that there are changes to the channel descriptors
- Clocks, Power, and Reset Block (cpr_top) - Implements clock and reset muxing and synchronization
- AMBA AHB Block (ahb_top) - Implements a bus bridge between the AHB master and the HBI slave interfaces
- AMBA APB Block (apb_top) - Implements a bus bridge that translates the two cycle APB interface signals to the single-cycle I/O interface signals

44.4.1 MediaLB Block

The Media Local Bus (MediaLB) block supports both a MediaLB 3-pin interface and MediaLB 6-pin interface. However, only one MediaLB interface can be active at any given time. Both MediaLB interfaces provide real-time access to all network data types including streaming, packet, control, and isochronous data.

- MediaLB 3-pin Interface - Supports the MediaLB protocol for single-ended 3-pin mode, with a maximum data rate of 1024xFs (49.152 MHz at Fs=48 kHz).
- MediaLB 6-pin Interface - Supports the MediaLB protocol for high-speed differential 6-pin mode, with a maximum data rate of 6144xFs (294.912 MHz at Fs=48 kHz).

44.4.1.1 MediaLB Channel Address to Logical Channel Mapping

The MediaLB channel addresses are mapped to the logical channels as follows.

Table 44-3. MediaLB Channel Address to Logical Channel Mapping

Channel Address	Logical Channel
0x0002	1
0x0004	2
0x0006	3
....	....
0x007C	62
0x007E	63
0x01FE	0*
* Logical Channel 0 is the System Channel and is reserved.	

44.4.2 Host Bus Interface Block

The Host Bus Interface (HBI) block provides a 16-bit parallel slave port that provides an external Host Controller (HC) with access to all MOST channels and data types.

Up to 64 independent HBI channels are available to the HC, each configurable for either transmitting or receiving a particular application data type (synchronous, isochronous, asynchronous, or control). The HBI block provides source and sink access to the full network data bandwidth.

44.4.2.1 HBI Physical Addresses

To access a particular HBI DMA channel, hardware must first translate the HBI channel address to a channel allocation table (CAT) physical address. This physical address is then used to retrieve the channel label (CL), which in turn retrieves the channel descriptor.

See Table 44-4 for more information on the mapping between the HBI channel address and physical address.

Table 44-4. HBI Channel Address to Physical Address Mapping

HBI Channel	CAT Address	CAT Offset
0x0	0x88	000

Table continues on the next page...

Table 44-4. HBI Channel Address to Physical Address Mapping (continued)

HBI Channel	CAT Address	CAT Offset
0x1	0x88	001
0x2	0x88	010
0x3	0x88	011
0x4	0x88	100
0x5	0x88	101
0x6	0x88	110
0x7	0x88	111
0x8	0x89	000
...	...	...
0x3E	0x8F	110
0x3F	0x8F	111

44.4.3 Routing Fabric Block

The Routing Fabric (RF) block manages the flow of data between the MediaLB Port and the HBI Port. Bus multiplexers and a bus arbiter are implemented in the RF block for accessing the channel table RAM (CTR) and data buffer RAM (DBR).

Each DMA controller in the routing fabric uses Channel Descriptors (stored in the CTR) to manage access to dynamic buffers in the DBR.

44.4.3.1 Data Buffer RAM

The MLB150 has an external data buffer RAM (DBR) that is 8-bit x 16k entries deep. The DBR provides dynamic circular buffering between the transmit and receive devices.

The size and location of each data buffer is defined by software in the channel descriptor table (CDT), which is located in the CTR.

Receive devices retain the write address pointer to the associated circular data buffer in the DBR, while transmit devices retain the read address pointer. The DMA controllers in the routing fabric are responsible for ensuring that the circular buffers do not overflow or underflow. Each channel type (e.g. synchronous, isochronous, asynchronous and control) has Full and Empty detection.

44.4.3.1.1 Synchronous Channels

For synchronous channels, two mechanisms prevent overflow and underflow of the data buffer:

- Hardware aligns the read pointer (RPTR) to the write pointer (WPTR) to ensure an offset of two sub-buffers.
- RPTR and WPTR are periodically synchronized to the start of the next sub-buffer (e.g. following a FRAMESYNC).

44.4.3.1.2 Isochronous Channels

For isochronous channels, hardware does not read from an empty data buffer or write to a full data buffer. The conditions used by hardware for detection include:

Data buffer Empty condition: (RPTR = WPTR) AND (BF = 0), and

Data buffer Full condition: (WPTR = RPTR) AND (BF = 1).

44.4.3.1.3 Asynchronous and Control Channels

For asynchronous and control channels, hardware does not read from an empty data buffer or write to a full data buffer. Hardware evaluates the DMA pointers (RPTR, WPTR) and packet count (RPC, WPC) to detect the data buffer condition, where:

- Data buffer Empty condition: (RPTR = WPTR) AND (RPC = WPC), and
- Data buffer Full condition: ((WPTR = RPTR) AND (WPC != RPC)) OR (WPC = (RPC - 1)).

44.4.3.2 Channel Table RAM

The MLB150 has an external Channel Table RAM (CTR) that is 128-bit x 144-entry. The CTR allows system software to dynamically configure channel routing and allocate data buffers in the DBR.

The CTR is logically divided into three sub-tables:

- Channel Descriptor Table (CDT)
- AHB Descriptor Table (ADT)
- Channel Allocation Table (CAT)

44.4.3.2.1 Address Mapping

Table 44-5. CTR Address Mapping

Label	Address	Bits 127…96	Bits 95…64	Bits 63…32	Bits 31…0
Channel Descriptor Table (CDT):					
CDT	0x00	CDT0[127:0], CL = 0			
	0x01	CDT1[127:0], CL = 1			
	0x02	CDT2[127:0], CL = 2			
	...	...			
	0x3D	CDT61[127:0], CL = 61			
	0x3E	CDT62[127:0], CL = 62			
	0x3F	CDT63[127:0], CL = 63			
AHB Descriptor Table (ADT):					
ADT*	0x40	ADT0[127:0]			
	0x41	ADT1[127:0]			
	0x42	ADT2[127:0]			
	...	...			
	0x7D	ADT61[127:0]			
	0x7E	ADT62[127:0]			
	0x7F	ADT63[127:0]			
Channel Allocation Table (CAT):					
CAT for MediaLB	0x80	CAT7	CAT6	CAT5	CAT4	CAT3	CAT2	CAT1	CAT0
	...	...	...	...	...	...	...	...	...
	0x87	CAT63	CAT62	CAT61	CAT60	CAT59	CAT58	CAT57	CAT56
CAT for HBI*	0x88	CAT71	CAT70	CAT69	CAT68	CAT67	CAT66	CAT65	CAT64
	...	...	...	...	...	...	...	...	...
	0x8F	CAT127	CAT126	CAT125	CAT124	CAT123	CAT122	CAT121	CAT120
* A fixed relationship exists between ADT entries and HBI CAT entries. When using HBI channel 0 (CAT64) on should program ADT0. When using HBI channel 1 (CAT65) on should program ADT1, and so on.									

44.4.3.2.2 Channel Allocation Table

The Channel Allocation Table (CAT) is comprised of 16 CTR entries (addresses 0x80 - 0x8F), as shown in Table 44-5. Each 16-bit CAT entry represents a logical connection to or from a transmit/receive device (e.g. MediaLB or HBI channel). All entries are indexed according to a fixed physical address assigned to every Rx/Tx channel (as shown in Table 44-6). The value stored in a CAT entry includes a 6-bit Connection Label, which provides a pointer to the CDT. To complete a logical channel and form a routing connection, system software must assign the same Connection Label to both the Rx and Tx channels.

Table 44-6. CAT Entry Map

Peripheral	Tx Channels	Rx Channels	CAT Start Index	CAT End Index	Entries
MediaLB	0 to 64	64 - Tx Channels	0	63	64
HBI	0 to 64	64 - Tx Channels	64	127	64

The format of a full CAT entry is shown in Table 44-7, with field descriptions described in Table 44-8. All reserved bits of a CAT entry field should be written as zero.

Table 44-7. CAT Entry Formats

Channel Type	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Isochronous	rsvd	FCE	rsvd	RNW	CE	CT[2:0] = 3			rsvd		CL[5:0]					
Asynchronous	rsvd		MT	RNW	CE	CT[2:0] = 2			rsvd		CL[5:0]					
Control	rsvd		MT	RNW	CE	CT[2:0] = 1			rsvd		CL[5:0]					
Synchronous	rsvd	MFE	MT	RNW	CE	CT[2:0] = 0			rsvd		CL[5:0]					

Table 44-8. CAT Field Definitions

Field	Description
CL[5:0]	Connection Label (offset into CDT)
CT[2:0]	Channel Type (Others): 111 = Reserved 110 = Reserved 101 = Reserved 100 = Reserved 011 = Isochronous 010 = Asynchronous 001 = Control 000 = Synchronous
CE	Channel Enable: 1 = Enabled 0 = Disabled
RNW	Read Not Write: 1 = Read 0 = Write
MT	Mute Enable: 1 1 = Enabled 0 = Disabled
FCE	Flow Control Enable: 2 1 = Enabled 0 = Disabled
MFE	Multi-Frame per Sub-buffer Enable: 3 1 = Enabled 0 = Disabled
rsvd	Reserved. Software writes a zero to all Reserved bits when the entry is initialized. The Reserved bits are Read-only after initialization.
1. When set for synchronous channels, the MT bit forces Rx channels to write zeros into the channel data buffer, and Tx channels to output zeros on the physical interface. When set for asynchronous and control channels, the MT bit causes DMA to halt at a packet boundary. Not valid for isochronous channels. 2. The FCE bit is used by MediaLB isochronous Rx channels only. 3. The MFE bit is used by MediaLB synchronous channels only.	

44.4.3.2.2.1 Channel Setup

Data direction in the MLB150 is in reference to the DBR. Therefore, the data direction of CAT entries corresponding to the same channel is reversed for the HBI CAT and the MediaLB CAT.

For a Tx channel (from the HC to the MediaLB interface):

- HBI CAT entry: RNW = 0 (write)
- MediaLB CAT entry: RNW = 1 (read)

Conversely, for a Rx channel (data from MediaLB to HC):

- HBI CAT entry: RNW = 1 (read)
- MediaLB CAT entry: RNW = 0 (write)

The figure below illustrates the directional relationship in the MLB150.

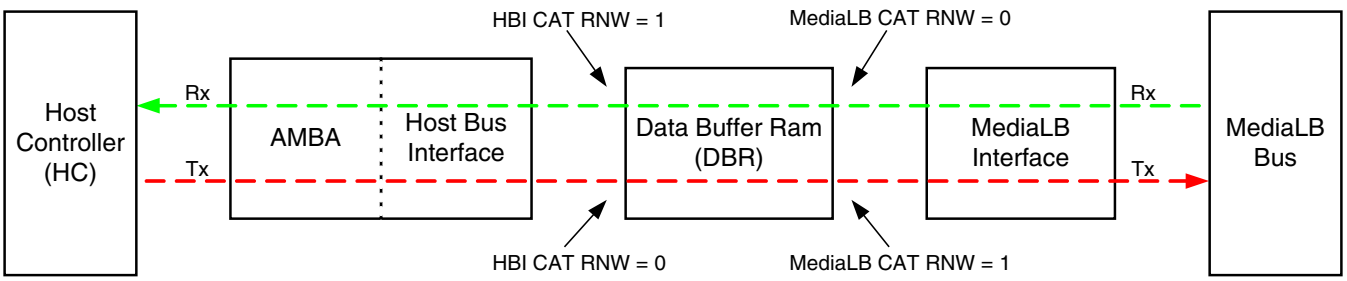


Figure 44-2. MLB DBR Directional Relationship

44.4.3.2.3 Channel Descriptor Table

The *Channel Descriptor Table* (CDT) is comprised of 64 CTR entries (addresses 0x00 - 0x3F), as shown in Table 44-5.

Each 128-bit CDT entry (also referred to as a *Channel Descriptor*) is referenced by a *Connection Label* and contains information about a data buffer in the DBR (e.g. buffer size, address pointers).

The format of each CDT entry (also referred to as a *Channel Descriptor*) is dependent on the channel type (e.g. synchronous, isochronous, asynchronous, or control).

NOTE

All reserved *Channel Descriptor* bits must be written to '0' by software when initialized.

44.4.3.2.3.1 Synchronous Channel Operation

The MLB150 provides two modes of operation (Standard and Multi-Frame per Sub-buffer) to provide flexibility for implementing synchronous channels.

Channels set up for Standard mode require less buffer space, but have higher interrupt rates and more stringent latency requirements. For channels configured for Standard mode, the Host Controller must transfer one full frame of streaming data in/out of each streaming channel's data buffer for each hbi_hsync period.

Channels set up for Multi-Frame per Sub-buffer mode require more buffer space, but have lower interrupt rates and less stringent latency requirements. For channels configured for Multi-Frame per Sub-buffer mode, the Host Controller must transfer N full frames of streaming data in/out of each streaming channel's data buffer for each hbi_hsync period.

To set up a channel in Multi-Frame per Sub-buffer mode:

- Program MLBC0[FCNT] to select the number of frames per sub-buffer
- Program the CAT to enable multi-frame sub-buffering (MFE = 1) for each particular channel
- Set the buffer depth in the CDT: BD = 4 × m × bpf - 1, where m = frames per sub-buffer, bpf = bytes per frame
- Repeat for additional synchronous channels

A sample synchronous data buffer is shown in the figure below. Each data buffer contains four sub-buffers and each sub-buffer contains space for 1 to 64 frames of data, determined by MLBC0.FCNT[2:0].

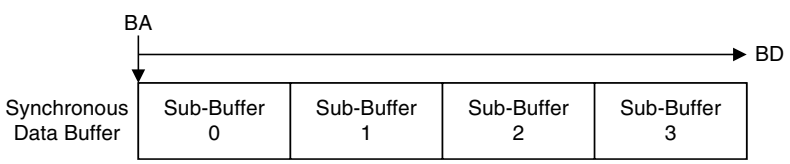


Figure 44-3. Synchronous Data Buffer Structure

44.4.3.2.3.2 Synchronous Channel Descriptors

The format and field definitions for a synchronous CDT entry are shown in the tables below respectively.

Table 44-9. Synchronous CDT Entry Format

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	WSBC		Reserved													
16	RSBC		Reserved													

Table continues on the next page...

Table 44-9. Synchronous CDT Entry Format (continued)

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
32	Reserved															
48	Reserved															
64	WSTS[3:0]				WPTR[11:0]											
80	RSTS[3:0]				RPTR[11:0]											
96	Reserved				BD[11:0]											
112	Reserved		BA[13:0]													

Table 44-10. Synchronous CDT Entry Field Definitions

Field	Description	Details	Accessibility
BA	Buffer Base Address	- BA can start at any byte in the 16k DBR	r,w
BD	Buffer Depth	- BD = size of buffer in bytes - 1 - Buffer end address = BA + BD - BD = 4 x m x bpf - 1, where: m = frames per sub-buffer (for MFE = 0, m = 1) bpf = bytes per frame.	r,w
RPTR	Read Pointer	- Software initializes to zero, hardware updates - Counts the read address offset within a buffer - DMA read address = BA + RPTR	r,w,u
WPTR	Write Pointer	- Software initializes to zero, hardware updates - Counts the write address offset within a buffer - DMA write address = BA + WPTR	r,w,u
RSBC	Read Sub-buffer Counter	- Software initializes to zero, hardware updates - Counts the read sub-buffer offset - DMA uses for pointer management	r,w,u
WSBC	Write Sub-buffer Counter	- Software initializes to zero, hardware updates - Counts the write sub-buffer offset - DMA uses for pointer management	r,w,u
RSTS	Read Status	- Software initializes to zero, hardware updates - RSTS states:* xxx0 = normal operation (no mute) xxx1 = normal operation (mute) xx0x = idle	r,w,u
WSTS	Write Status	- Software initializes to zero, hardware updates - WSTS states:* xxx0 = normal operation (no mute) xxx1 = normal operation (mute) xx0x = idle	r,w,u

Table continues on the next page...

Table 44-10. Synchronous CDT Entry Field Definitions (continued)

Field	Description	Details	Accessibility
		1xxx = command protocol error	
Reserved	Reserved	- Software writes a zero to all *Reserved* bits when the entry is initialized. The *Reserved* bits are *Read-only* after initialization.	r,w,u
* Only valid for DMA pointers associated with the MediaLB block (Not valid for HBI block related pointers).			

44.4.3.2.3.3 Isochronous Channel Descriptors

The format and field definitions for an isochronous CDT entry are shown in the tables below respectively.

Table 44-11. Isochronous CDT Entry Format

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Reserved															
16	Reserved															
32	Reserved							BS[8:0]								
48	Reserved															
64	WSTS[2:0]			WPTR[12:0]												
80	RSTS[2:0]			RPTR[12:0]												
96	Reserved			BD[12:0]												
112	BF	rsvd	BA[13:0]													

Table 44-12. Isochronous CDT Entry Field Definitions

Field	Description	Details	Accessibility
BA	Buffer Base Address	- BA can start at any byte in the 16k DBR	r,w
BD	Buffer Depth	- BD = size of buffer in bytes - 1 - Buffer end address = BA + BD - Isochronous buffers must be large enough to hold at least 3 blocks (packets) of data - Buffer depth must be a integer multiple of blocks	r,w
BF	Buffer Full	- Software initializes to zero, hardware updates - DMA write hardware sets BF when the buffer is full - DMA read hardware clears BF when the buffer is empty - BF is valid only when the buffer is full or empty, otherwise ignore	r,w,u
BS	Block Size	- BS defines when to begin the DMA to the data buffer - BS = buffer block size in bytes - 1	r,w,u

Table continues on the next page...

Table 44-12. Isochronous CDT Entry Field Definitions (continued)

Field	Description	Details	Accessibility
		- For Rx channels, the DMA writes start when the number of empty bytes (SPACE) in the data buffer >= the block size - For Tx channels, the DMA reads start when the number of valid bytes (VALID) in the data buffer >= the block size	
RPTR	Read Pointer	- Software initializes to zero, hardware updates - Counts the read address offset within a buffer - DMA read address = BA + RPTR	r,w,u
WPTR	Write Pointer	- Software initializes to zero, hardware updates - Counts the write address offset within a buffer - DMA write address = BA + WPTR	r,w,u
RSTS	Read Status	- Software initializes to zero, hardware updates - RSTS states:* xx1 = active xx0 = idle	r,w,u
WSTS	Write Status	- Software initializes to zero, hardware updates - WSTS states:* xx1 = active xx0 = idle x1x = command protocol error 1xx = buffer overflow (FCE = 0 only)	r,w,u
Reserved	Reserved	- Software writes a zero to all *Reserved* bits when the entry is initialized. The *Reserved* bits are *Read-only* after initialization.	r,w,u
* Only valid for DMA pointers associated with the MediaLB block (Not valid for HBI block related pointers).			

44.4.3.2.3.4 Asynchronous and Control Channel Descriptors

The format and field definitions for asynchronous and control CDT entries are shown in the tables below respectively.

Table 44-13. Asynchronous/Control CDT Entry Format

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	WPC[4:0]					Reserved										
16	RPC[4:0]					Reserved										
32	rsvd	WPC[7:5]			Reserved											
48	rsvd	RPC[7:5]			Reserved											
64	WSTS[3:0]				WPTR[11:0]											
80	RSTS[3:0]				RPTR[11:0]											

Table continues on the next page...

Table 44-13. Asynchronous/Control CDT Entry Format (continued)

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
96	RSTS[4]	WSTS[4]	rsvd		BD[11:0]											
112	Reserved		BA[13:0]													

Table 44-14. Asynchronous/Control CDT Entry Field Definitions

Field	Description	Details	Accessibility
BA	Buffer Base Address	- BA can start at any byte in the 16k DBR	r,w
BD	Buffer Depth	- BD = size of buffer in bytes - 1 - Buffer end address = BA + BD - BD >= max packet length - 1	r,w
RPC	Read Packet Count	- Software initializes to zero, hardware updates - Used in conjunction with WPC, RPTR and WPTR to determine if the buffer is empty or full	r,w,u
WPC	Write Packet Count	- Software initializes to zero, hardware updates - Used in conjunction with RPC, RPTR and WPTR to determine if the buffer is empty or full	r,w,u
RPTR	Read Pointer	- Software initializes to zero, hardware updates - Counts the read address offset within a buffer - DMA read address = BA + RPTR	r,w,u
WPTR	Write Pointer	- Software initializes to zero, hardware updates - Counts the write address offset within a buffer - DMA read address = BA + WPTR	r,w,u
RSTS	Read Status	- Software initializes to zero, hardware updates - Status states:* x0x00 = idle xx1xx = *ReceiverProtocolError* response received from Rx Device 1xxxx = *ReceiverBreak* command received from Rx Device	r,w,u
WSTS	Write Status	- Software initializes to zero, hardware updates - Status states:* x0x00 = idle xx1xx = command protocol error detected 1xxxx = *AsyncBreak*/*ControlBreak* command received from Tx Device	r,w,u
Reserved	Reserved	Software writes a zero to all *Reserved* bits when the entry is initialized. The *Reserved* bits are *Read-only* after initialization.	r,w,u
* Only valid for DMA pointers associated with the MediaLB block (Not valid for HBI block related pointers)			

44.4.4 Memory Interface Block

The Memory Interface (MIF) block implements a bridge between the I/O and the CTB or DBB interfaces.

The MIF block diagram is shown in the figure below. The targeted RAM (CTR or DBR) is determined by the target location bit in the memory address register (MADR.TB).

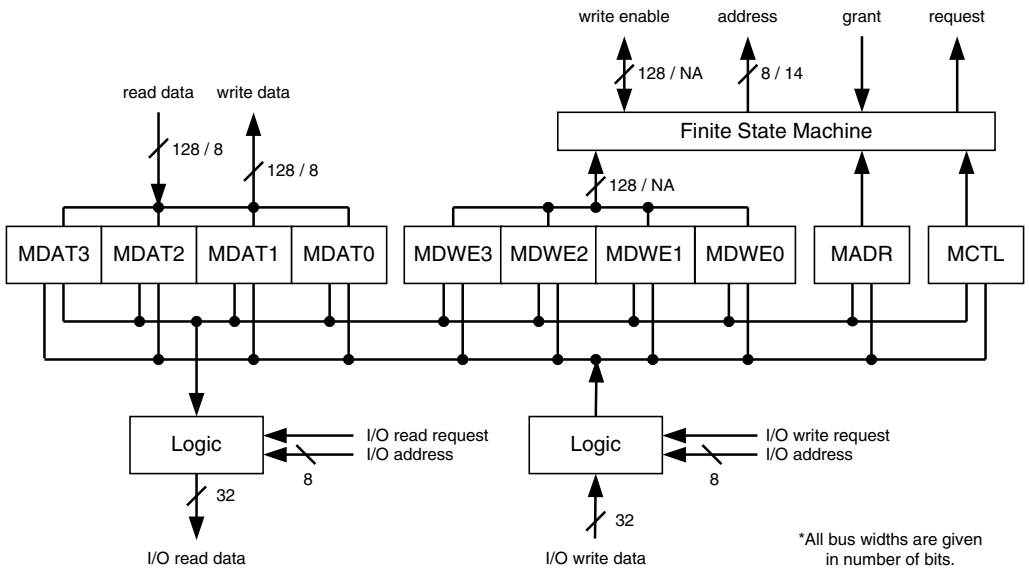


Figure 44-4. MIF Block Diagram

NOTE

The size of the read data, write data, write enable, and address buses are dependent on whether the CTB or DBB interfaces are used. In the figure above, the first number is for CTR accesses while the second is for DBR accesses (i.e. CTR bus width / DBR bus width).

44.4.4.1 CTR Access

The MIF block allows the HC to directly access the external Channel Table RAM (CTR) when MADR.TB is cleared. Any write to the MADR register triggers a single read or write cycle. Reading from the MADR register does not initiate read/write access.

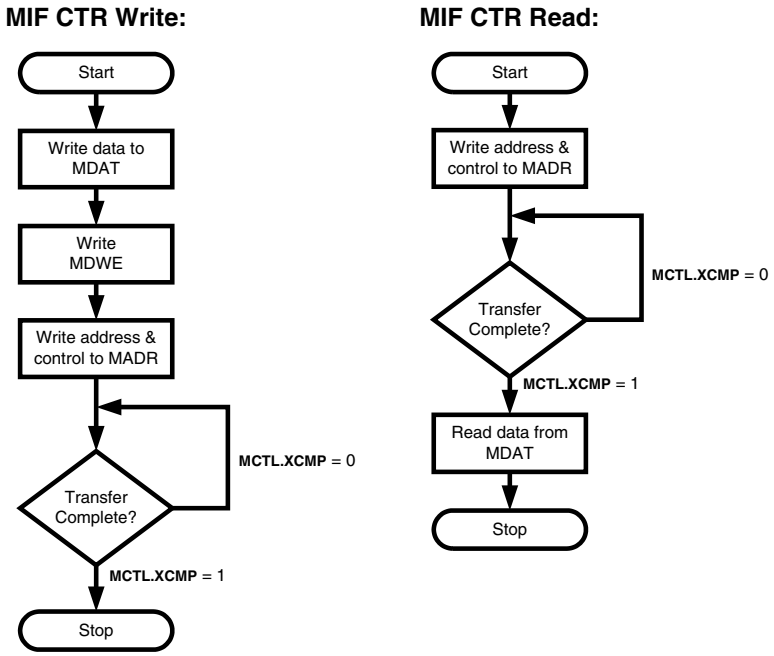


Figure 44-5. MIF CTR Read and Write Flow Diagrams

44.4.4.1.1 Direct CTR Writes

For a direct write of the CTR, the HC first loads the 128-bit data entry into the MDAT0-3 registers. Bitwise write enable control is available via the MDWE0-3 registers.

After the MDATn and MDWEn registers are set up, a write cycle is initiated by writing the address and control information to MADR as follows:

- MADR.WNR = 1
- MADR.TB = 0
- MADR.ADDR[7:0] = 8-bit Target Address

The MIF block sets MCTL.XCMP = 1 to inform the HC when the write is complete.

44.4.4.1.2 Direct CTR Reads

For a direct read of the CTR, the HC initiates a read cycle by writing the address and control information to MADR as follows:

- MADR.WNR = 0

- MADR.TB = 0
- MADR.ADDR[7:0] = 8-bit Target Address

The MIF block sets MCTL.XCMP = 1 to inform the HC when the read is complete. The HC can then read the 128-bit data entry from the MDAT0-3 registers.

44.4.4.1.3 CTR Addressing

The CTR is addressed as a 128-bit wide value. However, the MIF block can only access 32 bits of the addressed CTR data in a single access. Therefore, four 32-bit accesses through the MIF block are required to access a single 128-bit value (e.g. CDT entry).

To access a 16-bit CAT entry in the CTR, only a single access through the MIF is required. For example, to load a CAT61 entry for an isochronous Tx channel with mute and flow control enabled:

- Write MDAT2 = 7B070000h (assumes Connection Label = 7)
- MDWE2 = FFFF0000h (bitwise write enable for 16 msbs; assumes MDWE0/1/3 = 00000000h)
- MADR = 80000087h (write CTR address 87h)

44.4.4.2 DBR Access

The MIF block allows the HC to access the external Data Buffer RAM (DBR) directly when MADR.TB is set. Any write to the MADR register triggers a single read or write cycle. Reading from the MADR register does not initiate read/write access.

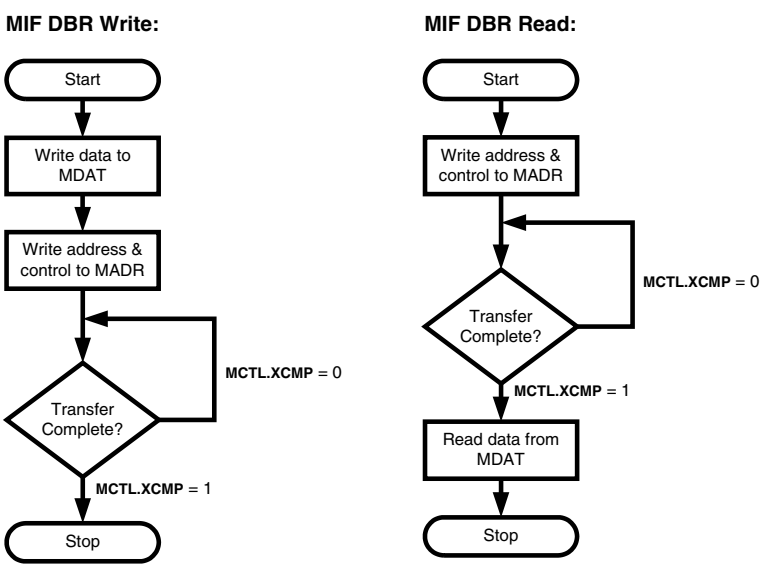


Figure 44-6. MIF DBR Read and Write Flow Diagrams

44.4.4.2.1 Direct DBR Writes

For a direct write of the DBR, the HC first loads the 8-bit data entry into the MDAT0 register at bits[7:0]. MDAT1-3 and MDWE0-3 are not used for DBR access.

After the MDAT0 register is set up, a write cycle is initiated by writing the address and control information to MADR as follows:

- MADR.WNR = 1
- MADR.TB = 1
- MADR.ADDR[13:0] = 14-bit Target Address

The MIF block sets MCTL.XCMP = 1 to inform the HC when the write is complete.

44.4.4.2.2 Direct DBR Reads

For a direct read of the DBR, the HC initiates a read cycle by writing the address and control information to MADR as follows:

- MADR.WNR = 0
- MADR.TB = 1
- MADR.ADDR[13:0] = 14-bit Target Address

The MIF block sets MCTL.XCMP = 1 to inform the HC when the read is complete. The HC can then read the 8-bit data entry from the MDAT0 register at bits[7:0].

44.4.5 Interrupt Interface Block

The Interrupt Interface (INTIF) block performs a low-priority polling algorithm of each of the HBI channel descriptors.

The INTIF alerts the HBI block when specific changes to HBI Channel Descriptors occur.

- For asynchronous and control read/write channels:
  - a packet is available to read in the channel buffer, or
  - sufficient empty space is available in the channel buffer to accept a requested packet write.
- For isochronous read/write channels:
  - the number of valid bytes in the channel buffer exceeds the block size, or
  - the number of empty bytes in the channel buffer exceeds the block size.

44.4.6 AMBA AHB Block

The AMBA AHB block manages data exchange between local channel data buffers within the MLB150 and the system memory buffer.

To support system memory buffering, a ping-pong memory structure is implemented on a per-channel basis using 128-bit descriptors for AHB Descriptor Table (ADT) entries.

NOTE

The 64 ADT entries are directly mapped to the 64 HBI physical channels.

Each logical channel is assigned a separate 128-bit descriptor, defining the data buffers in the system memory used by the DMA interface for that channel. The descriptors are stored at fixed addresses in the external CTR.

44.4.6.1 AHB Descriptor Table

The table below provides an overview of field definitions for ADT entries.

Table 44-15. ADT Field Definitions

Field	No. of Bits	Description	Accessibility
CE	1	Channel enable: 0 = Disabled 1 = Enabled	r,w,u
LE	1	Endianess select: 0 = Big Endian 1 = Little Endian	r,w
PG	1	Page pointer. Software initializes to zero, hardware writes thereafter. 0 = Ping buffer 1 = Pong buffer	r,w,u
RDY1	1	Buffer ready bit for ping buffer page: 0 = Not ready 1 = Ready	r,w
RDY2	1	Buffer ready bit for pong buffer page: 0 = Not ready 1 = Ready	r,w
DNE1	1	Buffer done bit for ping buffer page: 0 = Not done 1 = Done	r,u,c0
DNE2	1	Buffer done bit for pong buffer page: 0 = Not done 1 = Done	r,u,c0
ERR1	1	AHB error response detected for ping buffer page: 0 = No error 1 = Rrror	r,u,c0
ERR2	1	AHB error response detected for pong buffer page: 0 = No error 1 = Error	r,u,c0
PS1	1	Packet start bit for ping buffer page: 0 = No packet start 1 = Packet start Reserved for synchronous and isochronous channels.	r,w,u (both Tx and Rx)
PS2	1	Packet start bit for pong buffer page: 0 = No packet start 1 = Packet start Reserved for synchronous and isochronous channels.	r,w,u (both Tx and Rx)
MEP1	1	Most Ethernet Packet (MEP) indicator for ping buffer page: 0 = Not MEP 1 = MEP MEP1 only valid for the first page of a segmented buffer. Reserved for control, synchronous and isochronous channels.	Rsvd for Tx r,u,c0 for Rx

Table continues on the next page...

Table 44-15. ADT Field Definitions (continued)

Field	No. of Bits	Description	Accessibility
MEP2	1	MEP packet indicator for pong buffer page: 0 = not MEP 1 = MEP MEP2 only valid for the first page of a segmented buffer. Reserved for control, synchronous and isochronous channels.	Rsvd for Tx r,u,c0 for Rx
BD1*	11 to 13	Buffer depth for ping buffer page: 11 or 12-bits for asynchronous and control channels. 13-bits for synchronous and isochronous channels.	r,w
BD2*	11 to 13	Buffer depth for pong buffer page: 11 or 12-bits for asynchronous and control channels. 13-bits for synchronous and isochronous channels.	r,w
BA1	32	Buffer base address for ping buffer page	r,w
BA2	32	Buffer base address for pong buffer page	r,w
Reserved	varies	Software writes a zero to all *Reserved* bits when the entry is initialized. The *Reserved* bits are *Read-only* after initialization.	r,w,u
* The buffer depth (BD1 and BD2) for synchronous channels must consider if *Multi-Frame per Sub-buffer* mode is enabled.			

Data exchange across the AHB interface can be configured as Little Endian (LE = 1) or Big Endian (LE = 0). The figure below provides an overview of the endian options, chosen by an ADT descriptor field.

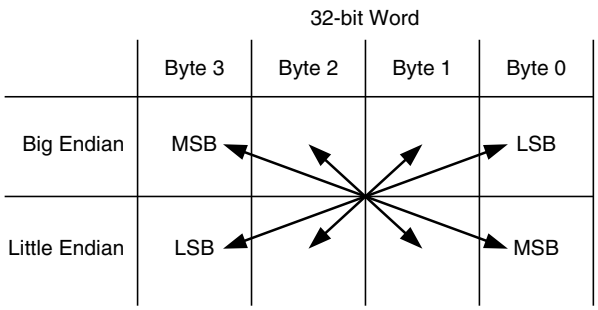


Figure 44-7. Endianness Overview

The figure below shows an example of the ping-pong system memory structure. This system memory structure is similar for all channel types and shows the relationship between the BAn, BDn, and PG descriptor fields.

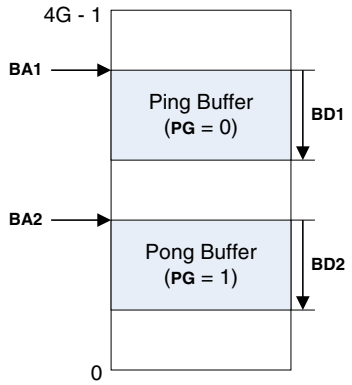


Figure 44-8. Ping-Pong System Memory Structure

Each ADT entry holds a 32-bit BAn field which defines the start of each ping or pong buffer within system memory. The BDn field is used to indicate the size for the respective ping or pong page. The maximum size is 2k-entries for asynchronous and control channels; 8k-entries for isochronous and synchronous channels.

44.4.6.2 AHB Synchronous Channel Descriptors

The table below shows the format for a synchronous ADT entry. The field definitions are defined in Table 44-15. Each synchronous channel buffer can be up to 8k-bytes deep.

Table 44-16. Synchronous ADT Entry Format

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CE	LE	PG	Reserved												
16	Reserved															
32	RDY 1	DNE 1	ERR 1	BD1[12:0]												
48	RDY 2	DNE 2	ERR 2	BD2[12:0]												
64	BA1[15:0]															
80	BA1[31:16]															
96	BA2[15:0]															
112	BA2[31:16]															

44.4.6.3 AHB Isochronous Channel Descriptors

The isochronous buffering scheme allows each ping or pong buffer to contain a single block or a multiple number of blocks. For this reason, the isochronous buffer depth (BDn) must be defined in terms of an *integer number* (n) and *block size* (BS) (e.g. BDn = n x (BS + 1) - 1).

Table 44-17 shows the format for an isochronous ADT entry. The field definitions are defined in Table 44-15. Each isochronous channel buffer can be up to 8k-bytes deep.

Table 44-17. Isochronous ADT Entry Format

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CE	LE	PG	Reserved												
16	Reserved															
32	RDY1	DNE1	ERR1	BD1[12:0]												
48	RDY2	DNE2	ERR2	BD2[12:0]												
64	BA1[15:0]															
80	BA1[31:16]															
96	BA2[15:0]															
112	BA2[31:16]															

44.4.6.4 AHB Asynchronous and Control Channel Descriptors

Every asynchronous and control packet adheres to the Port Message Protocol (PMP), which designates the first two bytes of each packet as the packet length (PML). Each packet must be no more than 2048 bytes.

Software must set the buffer ready bit (RDYn) for each buffer as it programs the DMA. As hardware processes each buffer, it sets the done bit (DNEn) and generates an interrupt to inform HC. When hardware finishes processing a buffer it can begin processing another buffer if RDYn is set. The application is responsible for setting up and configuring the channel buffer descriptor prior to every DMA access on the channel.

Two packet modes are supported by hardware for programming the DMA, single-packet mode and multiple-packet mode.

44.4.6.4.1 Single-packet Mode

The single-packet mode asynchronous and control buffering scheme supports a maximum of one packet per buffer (e.g. ping or pong). Both non-segmented and segmented data packets are allowed while using single-packet mode.

Non-segmented packets are exchanged when only one buffer (e.g. ping or pong) is needed for packet transfer. Segmented packets are exchanged when a single packet is too long for one buffer and the packet must span multiple buffers. The figure below shows the memory space usage for both non-segmented and segmented asynchronous or control packets along with the packet start bit (PSn). While using single-packet mode, buffer done (DNEn) is set in hardware when a packet is done or the buffer is full.

Table 44-18 shows the format for single-packet mode asynchronous and control ADT entries. The field definitions are defined in Table 44-15.

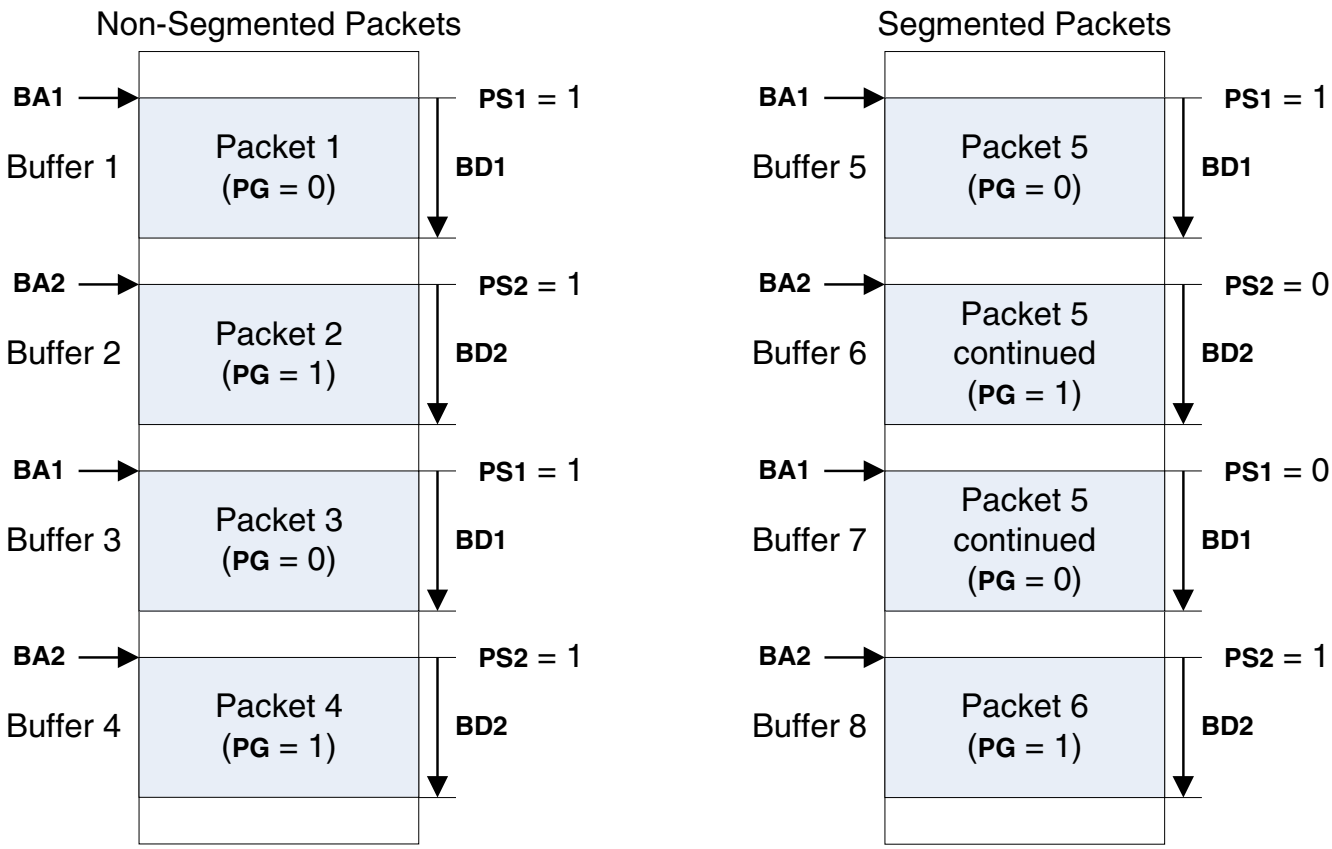


Figure 44-9. Single-packet Asynchronous or Control System Memory Structure

Table 44-18. Single-packet Asynchronous and Control Entry Format

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CE	LE	PG	Reserved												
16	Reserved															
32	RDY1	DNE1	ERR1	PS1	MEP1	BD1[10:0]										
48	RDY2	DNE2	ERR2	PS2	MEP2	BD2[10:0]										
64	BA1[15:0]															
80	BA1[31:16]															

Table continues on the next page...

Table 44-18. Single-packet Asynchronous and Control Entry Format (continued)

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
96	BA2[15:0]															
112	BA2[31:16]															

44.4.6.4.2 Multiple-packet Mode

The multiple-packet mode asynchronous and control buffering scheme supports more than one packet per system memory buffer, as shown in the figure below. Multiple-packet mode reduces the interrupt rate for packet channels at the cost of increasing buffering and latency.

For Tx packet channels in multiple-packet mode, software sets the packet start bit (PSn) for every buffer. Setting PSn informs hardware that the first two bytes of the buffer contains the port message length (PML) of the first packet. After the first packet, hardware keeps track of where packets start and end within the current buffer. Software should not write to PSn while the buffer is active (RDYn = 1 and DNEn = 0). For Tx packet channels, the buffer is done (DNEn= 1) when the last byte of the last packet in the buffer is read from system memory. Software should set the buffer depth to contain the exact number of complete packets for that buffer. Segmented buffers are not supported for Tx packet channels in multiple-packet mode.

For Rx packet channels in multiple-packet mode, PSn has no meaning and should be ignored. Software is responsible for keeping track of where each packet starts and ends within the multiple-packet buffer via the packet PML. The buffer done bit (DNEn) is set in hardware for Rx channels when a buffer is full (see Buffer 1 in the figure below) or if a packet ends exactly 1-byte before the end of the buffer (see Buffer 2 in the figure below). Multiple-packet mode also supports segmented Rx packets spanning two or more buffers (see Buffers 3 - 6in the figure below).

Table 44-19 The table below shows the format for multiple-packet mode asynchronous and control ADT entries. The field definitions are defined in Table 44-15.

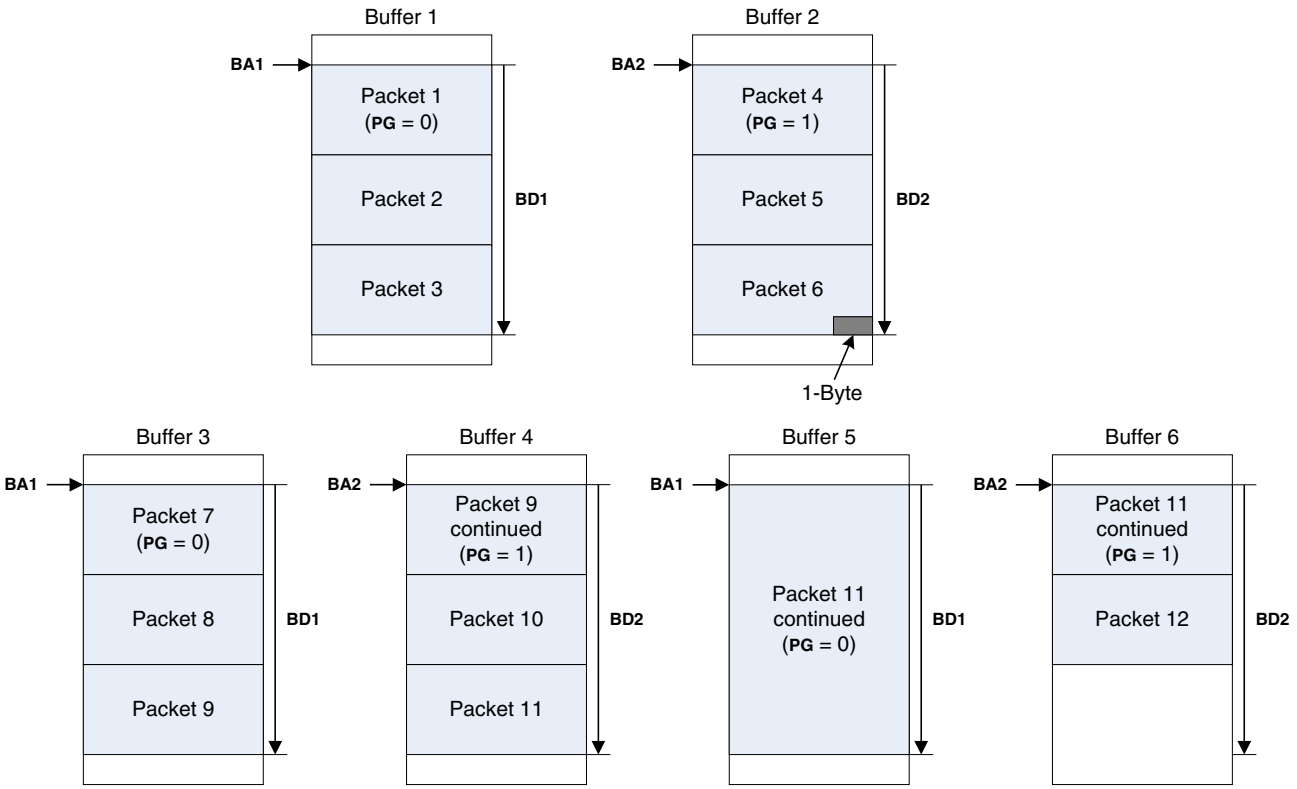


Figure 44-10. Multiple-packet Asynchronous or Control System Memory Structure

Table 44-19. Multiple-packet Asynchronous and Control Entry Format

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CE	LE	PG	Reserved												
16	Reserved															
32	RDY1	DNE1	ERR1	PS1*	BD1[11:0]											
48	RDY2	DNE2	ERR2	PS2*	BD2[11:0]											
64	BA1[15:0]															
80	BA1[31:16]															
96	BA2[15:0]															
112	BA2[31:16]															
* PSn is only valid for TX channels. Set PSn = 1 at the start of the buffer.																

44.5 Software Flow

The top-level software tasks the application must perform can be placed in two categories.

- Channel Initialization
- Channel Servicing

44.5.1 Channel Initialization

The software flow required to initialize a channel must be performed in order to ensure proper operation.

For clarity, the software flow is grouped as follows:

- Configure the Hardware
- Program the Routing Fabric Block
- Program the AMBA AHB Block DMAs
- Synchronize and Unmute Synchronous Channel

NOTE

In the event the external MLB INIC gets reset or disconnected, the MLB core and MLB PLL software initialization steps need to be performed again to resume channel operation.

44.5.1.1 Configure the Hardware

The MLBC0, HMCR0, HMCR1 and HCTL registers are accessible directly via APB reads and writes.

1. Initialize CTR and registers
   a. Set all bit of the CTR (CAT, CDT, and ADT) to'0'.
   b. Set all bits of all registers to'0'.
2. Configure the MediaLB interface
   a. Select 3-pin or 6-pin MediaLB operation: MLBC0.MLBPEN = 0 (3-pin), MLBC0.MLBPEN = 1 (6-pin)
   b. Select MediaLB clock speed via MLBC0.MLBCLK
   c. Set MediaLB enable via MLBC0.MLBEN
3. Configure the HBI interface
   a. Set HMCR0 and HMCR1 = FFFFFFFFh to activate all channels
   b. Set the HBI enable bit: HCTL.EN = 1

44.5.1.2 Program the Routing Fabric Block

The CAT and CDT reside in the external CTR and are programmed indirectly via APB or I/O reads and writes to the MIF block.

1. Initialize all bits of the CAT to '0'
2. Select a logical channel: N = 0 - 63
3. Program the CDT for channel N
    a. Set the 14-bit base address (BA)
    b. Set the 12-bit or 13-bit buffer depth (BD): BD = buffer depth in bytes -1
        - i. For synchronous channels: (BD + 1) = 4 x frames per sub-buffer (m) x bytes-per-frame (bpf)
        - ii. For isochronous channels: (BD + 1) mod (BS + 1) = 0
        - iii. For asynchronous channels: (BD + 1) >= max packet length (1024 for a MOST Data Packet (MDP); 1536 for a MOST Ethernet Packet (MEP))
        - iv. For control channels: (BD + 1) >= max packet length (64)
    c. For isochronous channels, set the block size (BS): BS = block size in bytes - 1
    d. Set all other bits of the CDT to '0'
4. Program the CAT for the inbound DMA
    a. For Tx channels (to MediaLB) HBI is the inbound DMA
    b. For Rx channels (from MediaLB) MediaLB is the inbound DMA
    c. Set the channel direction: RNW = 0
    d. Set the channel type: CT[2:0] = 010 (asynchronous), 001 (control), 011 (isochronous), or 000 (synchronous)
    e. Set the connection label: CL[5:0] = N
    f. If CT[2:0] = 000 (synchronous), set the mute bit (MT = 1).
    g. Set the channel enable: CE = 1
    h. Set all other bits of the CAT to '0'
5. Program the CAT for the outbound DMA
    a. For Tx channels (to MediaLB) MediaLB is the outbound DMA
    b. For Rx channels (from MediaLB) HBI is the outbound DMA
    c. Set the channel direction: RNW = 1
    d. Set the channel type: CT[2:0] = 010 (asynchronous), 001 (control), 011 (isochronous), or 000 (synchronous)
    e. Set the channel label: CL[5:0] = N
    f. If CT[2:0] = 000 (synchronous), set the mute bit (MT = 1)
    g. Set the channel enable: CE = 1
    h. Set all other bits of the CAT to '0'
6. Repeat steps 2-5 to initialize all logical channels

44.5.1.3 Program the AMBA AHB Block DMAs

The ADT resides in the external CTR and is programmed indirectly via APB reads and writes to the MIF.

1. Initialize all bits of the ADT to '0'
2. Select a logical channel: N = 0 - 63
3. Program the AMBA AHB block ping page for channel N
   a. Set the 32-bit base address (BA1)
   b. Set the 11-bit buffer depth (BD1): BD1 = buffer depth in bytes - 1
      - i. For synchronous channels: (BD1 + 1) = n x frames per sub-buffer (m) x bytes-per-frame (bpf)
      - ii. For isochronous channels: (BD1 + 1) mod (BS + 1) = 0
      - iii. For asynchronous channels: 5 <= (BD1 + 1) <= 4096 (max packet length)
      - iv. For control channels: 5 <= (BD1 + 1) <= 4096 (max packet length)
   c. For asynchronous and control Tx channels set the packet start bit (PS1) iff the page contains the start of the packet
   d. Clear the page done bit (DNE1)
   e. Clear the error bit (ERR1)
   f. Set the page ready bit (RDY1)
4. Program the AMBA AHB block pong page for channel N
   a. Set the 32-bit base address (BA2)
   b. Set the 11-bit buffer depth (BD2): BD2 = buffer depth in bytes - 1
      - i. For synchronous channels: (BD2 +1) = n x frames per sub-buffer (m) x bytes-per-frame (bpf)
      - ii. For isochronous channels: (BD2 + 1) mod (BS + 1) = 0
      - iii. For asynchronous channels: 5 <= (BD2 + 1) <= 4096 (max packet length)
      - iv. For control channels: 5 <= (BD2 + 1) <= 4096 (max packet length)
   c. For asynchronous and control Tx channels set the packet start bit (PS2) if the page contains the start of the packet
   d. Clear the page done bit (DNE2)
   e. Clear the error bit (ERR2)
   f. Set the page ready bit (RDY2)
5. Select Big Endian (LE = 0) or Little Endian (LE = 1)
6. Select the active page: PG = 0 (ping), PG = 1 (pong)
7. Set the channel enable (CE) bit for all active logical channels
8. Repeat steps 2-7 for all active logical channels

NOTE

All asynchronous and control packets must start with a PMP header. The first two bytes of the PMP header contains the Port Message Length (PML), which defines the length of the message that follows in bytes (not including PML itself). Hardware uses the PML to determine when a packet is complete. Asynchronous and control packets can also be segmented into two or more pages as well as contain multiple packets per page within system memory.

44.5.1.4 Synchronize and Unmute Synchronous Channel

The MLBC0 and MLBC1 registers are accessible directly via APB reads and writes.

1. Check that MediaLB clock is running (MLBC1.CLKM = 0)
2. If MLBC1.CLKM = 1, clear the register bit, wait one APB or I/O clock cycle and repeat step 1.
3. Poll for MediaLB lock (MLBC0.MLBLK = 1)
4. Wait four frames
5. Unmute synchronous channel(s)

44.5.2 Channel Servicing

After initialization, each channel will require periodic servicing.

The following software flows can be performed concurrently and in any order:

- Servicing the AMBA AHB Block (DMA) Interrupts
- Servicing the MediaLB Interrupts
- Polling for MediaLB System Commands

44.5.2.1 Servicing the AMBA AHB Block (DMA) Interrupts

The ACMR0, ACMR1, ACTL, ACSR0, and ACSR1 registers are accessible directly via APB reads and writes.

1. Program the ACMRn registers to enable interrupts from all active DMA channels
2. Select the status clear method: SCTL.SCE = 0 (hardware clears on read), SCTL.SCE = 1 (software writes a '1' to clear)

3. Select 1 or 2 interrupt signals: SCTL.SMX = 0 (one interrupt for channels 0-31 on *ahb_int[0]* and another interrupt for channels 32-63 on *ahb_int[1]*), SCTL.SMX = 1 (single interrupt for all channels on *ahb_int[0]*)
4. Wait for an interrupt from *ahb_int[1:0]*
5. Read the ACSRn registers to determine which channel or channels are causing the interrupt
6. If SCTL.SCE = 1, write the results of step 5 back to ACSR0 and ACSR1 to clear the interrupt
7. Select a logical channel (N = 0-63) with an interrupt to service
8. Read the ADT entry for channel N
   a. Determine the active page (ping or pong) via the PG bit
   b. Determine which page(s) are done via the DNEn bits
   c. Determine which channels encountered an AHB error via the ERRn bit
   d. Determine which asynchronous and control Rx channel pages contain a packet start via the PSn bit (extract the PML)
9. Reprogram the expired or broken AHB page(s) via steps 3 and 4 in Program the AMBA AHB Block DMAs,
10. Repeat steps 6-9 for all channels with pending interrupts
11. Repeat steps 4-10 while there are active channels

NOTE

Channels that receive an AHB error response are disabled (CE = 0) by hardware.

44.5.2.2 Servicing the MediaLB Interrupts

1. Select the MediaLB Channel Status Register (MSn) to be cleared by software, writing a '0' to the appropriate bits
2. Program MIEN to enable protocol error interrupts for all active MediaLB channels (MIEN.CTX_PE = 1, MIEN.CRX_PE = 1, MIEN.ATX_PE = 1, MIEN.ARX_PE = 1, MIEN.SYNC_PE = 1, and MIEN.ISOC_PE = 1)
3. Wait for an interrupt on the *mlb_int* signal.
4. Read the MSn registers to determine which channel(s) are causing the interrupt
5. Read RSTS/WSTS of the appropriate CDT(s) to determine the interrupt type
6. Clear RSTS/WSTS errors to resume channel operation
   a. For synchronous channels: WSTS[3] = 0
   b. For isochronous channels: WSTS[2:1] = 00
   c. For asynchronous and control channels: RSTS[4]/WSTS[4] = 0 and RSTS[2]/WSTS[2] = 0

44.5.2.3 Polling for MediaLB System Commands

The MLB supports the MediaLB System Commands (e.g. MlbScan, MlbReset, MOST_Unlock). The MediaLB System Status (MSS) Register is used to detect a System Command received from the MediaLB Controller. The MLB automatically sends the appropriate system response to the MediaLB Controller.

The procedure for the application is:

1. The application periodically polls the MSS register.
2. Clear by writing a '0' to the appropriate bit in MSS register after the application finishes the service.
3. If MSS.SWSYSCMD = 1, read the MSD register to receive the system data sent from MediaLB Controller.

44.5.3 Low Power Mode

MLB doesn't provide dedicated low power mode features.

In case the clocks of digital IP need to shut down to save power, the following operations are recommended before entering low power mode:

- Finish any active MLB transfer
- Disable MLB (clear the MLBEN and MLBPEN bits in MLBC0)
- Disable HBI (clear all bits in HCMR0 and HCMR1, clear EN bit in HCTL)
- Mask AHB interrupts (clear all bits in ACMR0 and ACMR1)

For information on configuring the MLB IP if the clocks are re-enabled, see Configure the Hardware.

NOTE

In the event the external MLB INIC gets reset or disconnected, the MLB core and MLB PLL software initialization steps must be performed again to resume channel operation.

44.6 MLB150 Memory Map/Register Definition

The MLB150 registers are divided into 4 sections.

The first section begins at offset 0h00 and describes the MediaLB block registers.

The second section begins at offset 0h80 and it shows the address mapping of the Internal HBI Registers.

The third set begins at offset 0hC0 and implements ten 32-bit I/O registers for CTR transfers, including: data registers (MDATn), write enable registers (MDWEn), a control register (MCTL), and an address register (MADR).

The fourth set, the AMBA AHB registers, begins at offset 0h3C0. They consist of:

- one 32-bit register for control
- two 32-bit registers for interrupt status
- two 32-bit registers for channel interrupt masks

MLB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_C000	MediaLB Control 0 Register (MLB_MLBC0)	32	R/W	0000_0000h	44.6.1/3863
218_C008	MediaLB 6-pin Control 0 Register (MLB_MLBPC0)	32	R/W	0000_0000h	44.6.2/3866
218_C00C	MediaLB Channel Status 0 Register (MLB_MS0)	32	R	0000_0000h	44.6.3/3866
218_C00D	MediaLB 6-pin Control 2 Register (MLB_MLBPC2)	32	R	0000_0000h	44.6.4/3867
218_C014	MediaLB Channel Status1 Register (MLB_MS1)	32	R	0000_0000h	44.6.5/3868
218_C020	MediaLB System Status Register (MLB_MSS)	32	R	0000_0000h	44.6.6/3869
218_C024	MediaLB System Data Register (MLB_MSD)	32	R	0000_0000h	44.6.7/3870
218_C02C	MediaLB Interrupt Enable Register (MLB_MIEN)	32	R/W	0000_0000h	44.6.8/3871
218_C038	MediaLB 6-pin Control 1 Register (MLB_MLBPC1)	32	R/W	0000_0000h	44.6.9/3872
218_C03C	MediaLB Control 1 Register (MLB_MLBC1)	32	R	0000_0000h	44.6.10/ 3873
218_C080	HBI Control Register (MLB_HCTL)	32	R/W	0000_0000h	44.6.11/ 3874
218_C088	HBI Channel Mask 0 Register (MLB_HCMR0)	32	R/W	0000_0000h	44.6.12/ 3875
218_C08C	HBI Channel Mask 1 Register (MLB_HCMR1)	32	R/W	0000_0000h	44.6.13/ 3875
218_C090	HBI Channel Error 0 Register (MLB_HCER0)	32	R	0000_0000h	44.6.14/ 3876
218_C094	HBI Channel Error 1 Register (MLB_HCER1)	32	R	0000_0000h	44.6.15/ 3876
218_C098	HBI Channel Busy 0 Register (MLB_HCBR0)	32	R	0000_0000h	44.6.16/ 3877
218_C09C	HBI Channel Busy 1 Register (MLB_HCBR1)	32	R	0000_0000h	44.6.17/ 3877
218_C0C0	MIF Data 0 Register (MLB_MDAT0)	32	R/W	0000_0000h	44.6.18/ 3878
218_C0C4	MIF Data 1 Register (MLB_MDAT1)	32	R/W	0000_0000h	44.6.19/ 3878

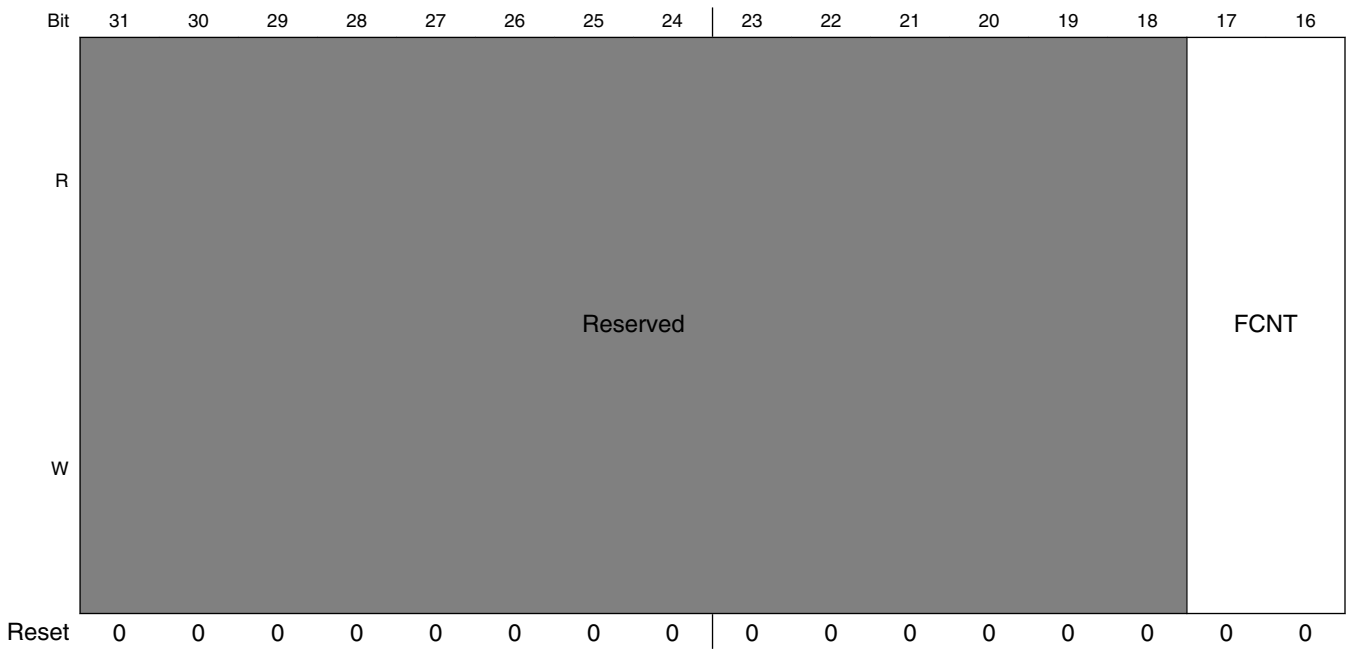
Table continues on the next page...

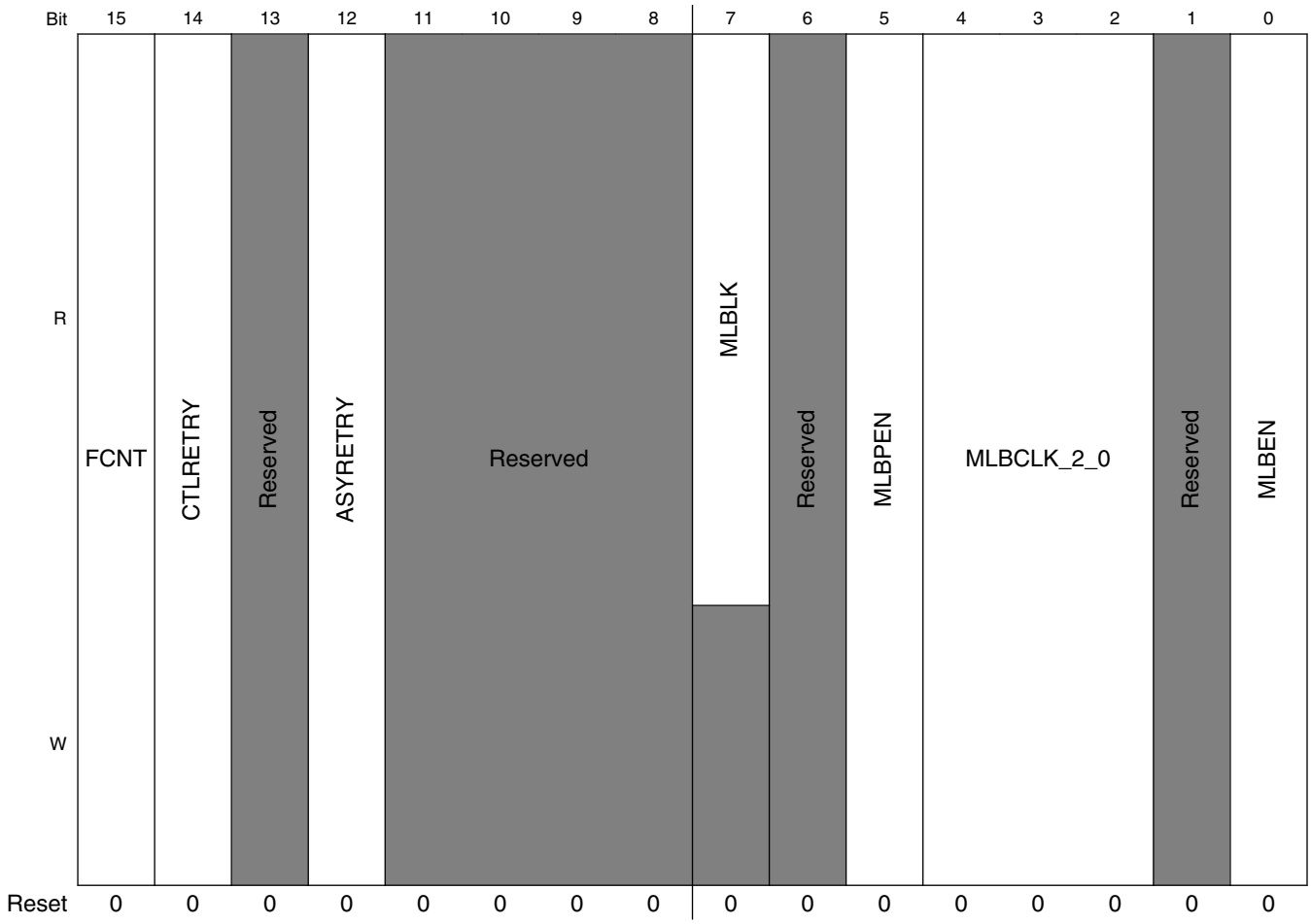
MLB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_C0C8	MIF Data 2 Register (MLB_MDAT2)	32	R/W	0000_0000h	44.6.20/ 3878
218_C0CC	MIF Data 3 Register (MLB_MDAT3)	32	R/W	0000_0000h	44.6.21/ 3879
218_C0D0	MIF Data Write Enable 0 Register (MLB_MDWE0)	32	R/W	0000_0000h	44.6.22/ 3879
218_C0D4	MIF Data Write Enable 1 Register (MLB_MDWE1)	32	R/W	0000_0000h	44.6.23/ 3879
218_C0D8	MIF Data Write Enable 2 Register (MLB_MDWE2)	32	R/W	0000_0000h	44.6.24/ 3880
218_C0DC	MIF Data Write Enable 3 Register (MLB_MDWE3)	32	R/W	0000_0000h	44.6.25/ 3880
218_C0E0	MIF Control Register (MLB_MCTL)	32	R	0000_0000h	44.6.26/ 3881
218_C0E4	MIF Address Register (MLB_MADR)	32	R/W	0000_0000h	44.6.27/ 3881
218_C3C0	AHB Control Register (MLB_ACTL)	32	R/W	0000_0000h	44.6.28/ 3882
218_C3D0	AHB Channel Status 0 Register (MLB_ACSR0)	32	R	0000_0000h	44.6.29/ 3883
218_C3D4	AHB Channel Status 1 Register (MLB_ACSR1)	32	R	0000_0000h	44.6.30/ 3884
218_C3D8	AHB Channel Mask 0 Register (MLB_ACMR0)	32	R/W	0000_0000h	44.6.31/ 3884
218_C3DC	AHB Channel Mask 1 Register (MLB_ACMR1)	32	R/W	0000_0000h	44.6.32/ 3885

44.6.1 MediaLB Control 0 Register (MLB_MLBC0)

Address: 218_C000h base + 0h offset = 218_C000h





MLB_MLBC0 field descriptions

Field	Description
31–18 -	This field is reserved. Reserved
17–15 FCNT	The number of frames per sub-buffer for synchronous channels. 000 1 frame per sub-buffer (Operation is the same as Standard mode.) 001 2 frames per sub-buffer 010 4 frames per sub-buffer 011 8 frames per sub-buffer 100 16 frames per sub-buffer 101 32 frames per sub-buffer 110 64 frames per sub-buffer 111 Reserved
14 CTLRETRY	Control Tx packet retry. When set, a control packet that is flagged with a Break or ProtocolError by the receiver is retransmitted. When cleared, a control packet that is flagged with a Break or ProtocolError by the receiver is skipped.
13 -	This field is reserved. Reserved

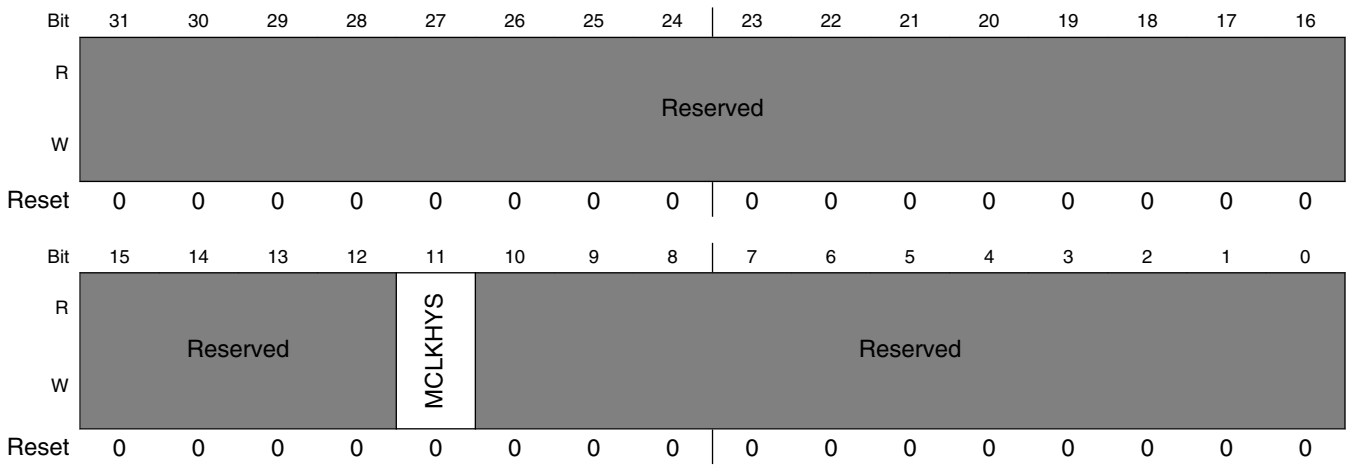
Table continues on the next page...

MLB_MLBC0 field descriptions (continued)

Field	Description
12 ASYRETRY	Asynchronous Tx packet retry. When set, an asynchronous packet that is flagged with a Break or ProtocolError by the receiver is retransmitted. When cleared, an asynchronous packet that is flagged with a Break or ProtocolError by the receiver is skipped.
11–8 -	This field is reserved. Reserved
7 MLBLK	MediaLB lock status. When set, indicates that the MediaLB block is synchronized to the incoming MediaLB frame. If **MLBLK** is clear (unlocked), **MLBLK** is set after FRAMESYNC is detected at the same position for three consecutive frames. If **MLBLK** is set (locked), **MLBLK** is cleared after not receiving FRAMESYNC at the expected time for two consecutive frames. While **MLBLK** is set, FRAMESYNC patterns occurring at locations other than the expected one are ignored. (read-only)
6 -	This field is reserved. Reserved
5 MLBPEN	MediaLB 6-pin enable. 0 MediaLB 3-pin interface enabled 1 MediaLB 6-pin interface enabled. MLB PLL and MLB PHY is enabled in this case.
4–2 MLBCLK_2_0	MLB_CLK (MediaLB clock) speed select. 000 256xFs (for **MLBPEN** = 0) 001 512xFs (for **MLBPEN** = 0) 010 1024xFs (for **MLBPEN** = 0) 011 2048xFs (for **MLBPEN** = 1) 100 3072xFs (for **MLBPEN** = 1) 101 4096xFs (for **MLBPEN** = 1) 110 6144xFs (for **MLBPEN** = 1) 111 reserved
1 -	This field is reserved. Reserved
0 MLBEN	MediaLB enable. When set, MLB_CLK (MediaLB clock), MLB_SIG (signal), and MLB_DATA (data) are received and transmitted on the appropriate MediaLB pins.

44.6.2 MediaLB 6-pin Control 0 Register (MLB_MLBPC0)

Address: 218_C000h base + 8h offset = 218_C008h

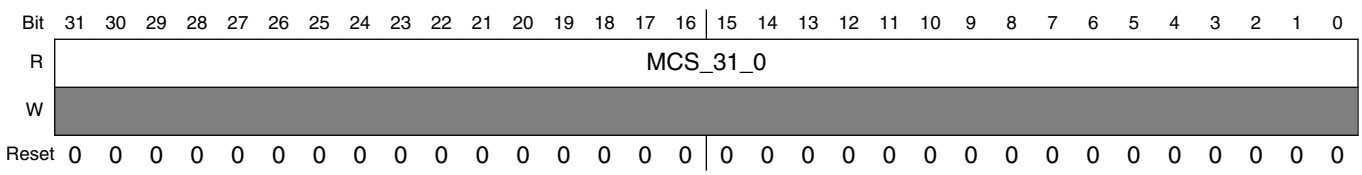


MLB_MLBPC0 field descriptions

Field	Description
31–12 -	This field is reserved. Reserved
11 MCLKHYS	MediaLB (6-pin) hysteresis enable. When set, enables hysteresis on the MLB_CLK (MediaLB clock). This value is driven on mlb_clk_hys_enable output pin and has no internal function.
-	This field is reserved. Reserved

44.6.3 MediaLB Channel Status 0 Register (MLB_MS0)

Address: 218_C000h base + Ch offset = 218_C00Ch

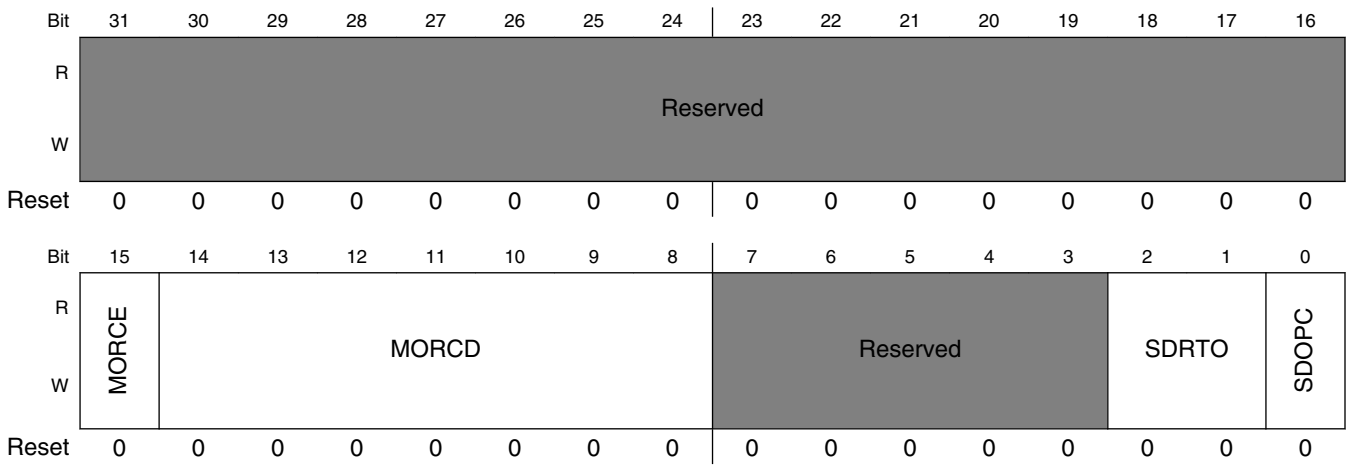


MLB_MS0 field descriptions

Field	Description
MCS_31_0	MediaLB channel status. Indicates the channel status for MediaLB channels 31 to 0. Channel status bits are set by hardware and cleared by software. Status is only set if the appropriate bits in the MIEN register are set.

44.6.4 MediaLB 6-pin Control 2 Register (MLB_MLBPC2)

Address: 218_C000h base + Dh offset = 218_C00Dh

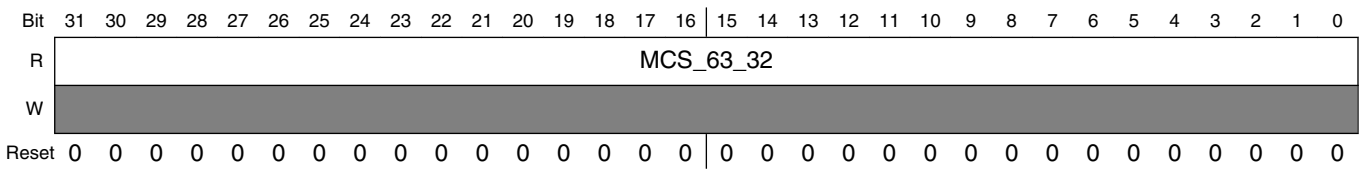


MLB_MLBPC2 field descriptions

Field	Description
31–16 -	Reserved. This field is reserved. Reserved
15 MORCE	Output reference clock enable. 0 Enable MLB output reference clock. 1 Disable MLB output reference clock.
14–8 MORCD	Divider factor of MLB output reference clock. MLB output reference clock is divided by MLB bus clock, the divider factor can be choose between 1 to 128. 0x00 Divider factor is 1. 0x01 Divider factor is 2. ... 0xff Divider factor is 256.
7–3 -	Reserved. This field is reserved. Reserved.
2–1 SDRTO	MLB 6-pin interface: Signal/Data receiver threshold offset control.
0 SDOPC	MLB 3-pin interface: Signal/Data output phase control. 0 MLB_SIG / MLB_DATA launch at rising edge of MLB_CLK(default) 1 MLB_SIG / MLB_DATA launch at falling edge of MLB_CLK

44.6.5 MediaLB Channel Status1 Register (MLB_MS1)

Address: 218_C000h base + 14h offset = 218_C014h

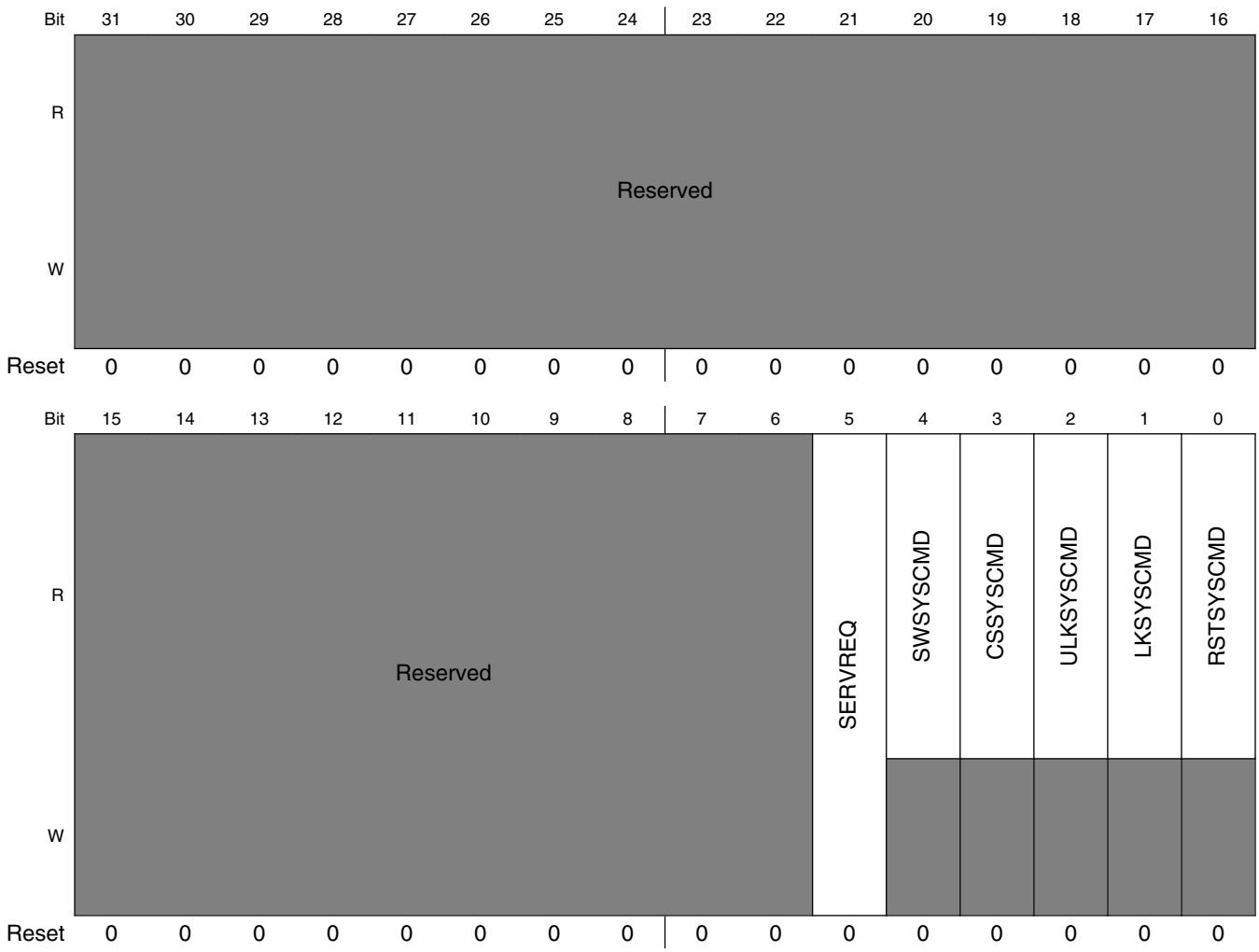


MLB_MS1 field descriptions

Field	Description
MCS_63_32	MediaLB channel status. Indicates the channel status for MediaLB channels 63 to 32. Channel status bits are set by hardware and cleared by software. Status is only set if the appropriate bits in the MIEN register are set.

44.6.6 MediaLB System Status Register (MLB_MSS)

Address: 218_C000h base + 20h offset = 218_C020h



MLB_MSS field descriptions

Field	Description
31–6 -	This field is reserved. Reserved
5 SERVREQ	Service request enabled. When set, the MediaLB block responds with a "device present, request service" system response if a matching channel scan system command is detected. When cleared, the MediaLB block responds with a "device present" system response.
4 SWSYSCMD	Software system command detected (in the system quadlet). Set by hardware, cleared by software. Data is stored in the MSD register for this command.
3 CSSYSCMD	Channel scan system command detected (in the system quadlet). Set by hardware, cleared by software. If the node address specified in *Data* quadlet matches the value in **MLBC1.NDA**, the device responds either "device present" or "device present, request service" system response in the next system quadlet.

Table continues on the next page...

MLB_MSS field descriptions (continued)

Field	Description
2 ULKSYSCMD	Network unlock system command detected (in the system quadlet). Set by hardware, cleared by software.
1 LKSYSCMD	Network lock system command detected (in the system quadlet). Set by hardware, cleared by software.
0 RSTSYSCMD	Reset system command detected (in the system quadlet). Set by hardware, cleared by software.

44.6.7 MediaLB System Data Register (MLB_MSD)

Address: 218_C000h base + 24h offset = 218_C024h

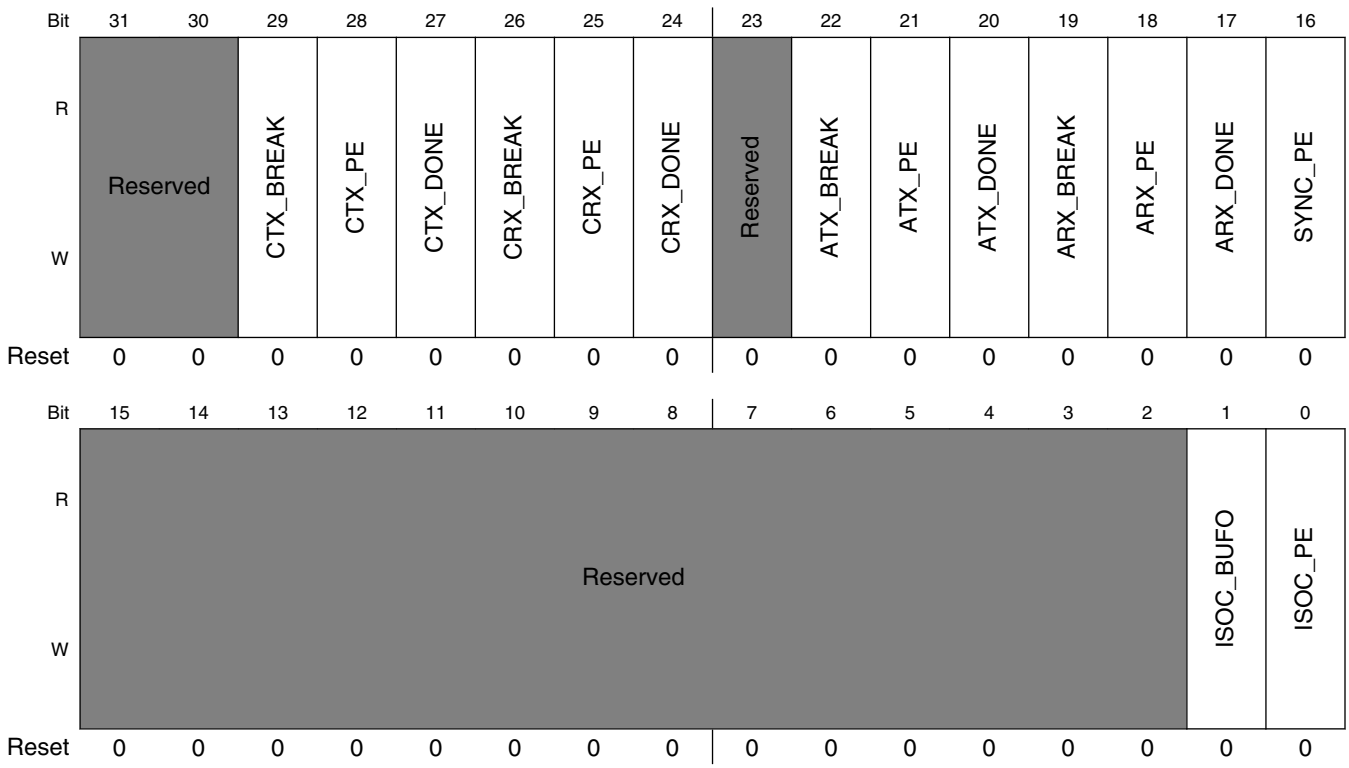
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SD3_7_0								SD2_7_0								SD1_7_0								SD0_7_0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_MSD field descriptions

Field	Description
31–24 SD3_7_0	System data (byte 3). Updated with MediaLB Data[31:24] when a MediaLB software system command is received in the system quadlet. If MSS.SWSYSCMD is already set, then SD3 is not updated. (read-only)
23–16 SD2_7_0	System data (byte 2). Updated with MediaLB Data[23:16] when a MediaLB software system command is received in the system quadlet. If MSS.SWSYSCMD is already set, then SD2 is not updated. (read-only)
15–8 SD1_7_0	System data (byte 1). Updated with MediaLB Data[15:8] when a MediaLB software system command is received in the system quadlet. If MSS.SWSYSCMD is already set, then SD1 is not updated. (read-only)
SD0_7_0	System data (byte 0). Updated with MediaLB Data[7:0] when a MediaLB software system command is received in the system quadlet. If MSS.SWSYSCMD is already set, then SD0 is not updated. (read-only)

44.6.8 MediaLB Interrupt Enable Register (MLB_MIEN)

Address: 218_C000h base + 2Ch offset = 218_C02Ch



MLB_MIEN field descriptions

Field	Description
31–30 -	This field is reserved. Reserved
29 CTX_BREAK	Control Tx break enable. When set, a *ReceiverBreak* response received from the receiver on a control Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
28 CTX_PE	Control Tx protocol error enable. When set, a *ProtocolError* generated by the receiver on a control Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
27 CTX_DONE	Control Tx packet done enable. When set, a packet transmitted with no errors on a control Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
26 CRX_BREAK	Control Rx break enable. When set, a *ControlBreak* command received from the transmitter on a control Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
25 CRX_PE	Control Rx protocol error enable. When set, a *ProtocolError* detected on a control Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
24 CRX_DONE	Control Rx packet done enable. When set, a packet received with no errors on a control Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
23 -	This field is reserved. Reserved
22 ATX_BREAK	Asynchronous Tx break enable. When set, a *ReceiverBreak* response received from the receiver on an asynchronous Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.

Table continues on the next page...

MLB_MIEN field descriptions (continued)

Field	Description
21 ATX_PE	Asynchronous Tx protocol error enable. When set, a *ProtocolError* generated by the receiver on an asynchronous Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
20 ATX_DONE	Asynchronous Tx packet done enable. When set, a packet transmitted with no errors on an asynchronous Tx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
19 ARX_BREAK	Asynchronous Rx break enable. When set, a *AsyncBreak* command received from the transmitter on an asynchronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
18 ARX_PE	Asynchronous Rx protocol error enable. When set, a *ProtocolError* detected on an asynchronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
17 ARX_DONE	Asynchronous Rx done enable. When set, a packet received with no errors on an asynchronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
16 SYNC_PE	Synchronous protocol error enable. When set, a *ProtocolError* detected on a synchronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.
15–2 -	This field is reserved. Reserved
1 ISOC_BUFO	Isochronous Rx buffer overflow enable. When set, a buffer overflow on an isochronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set. This occurs only when isochronous flow control is disabled.
0 ISOC_PE	Isochronous Rx protocol error enable. When set, a *ProtocolError* detected on an isochronous Rx channel causes the appropriate channel bit in the MS0 or MS1 registers to be set.

44.6.9 MediaLB 6-pin Control 1 Register (MLB_MLBPC1)

Address: 218_C000h base + 38h offset = 218_C038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

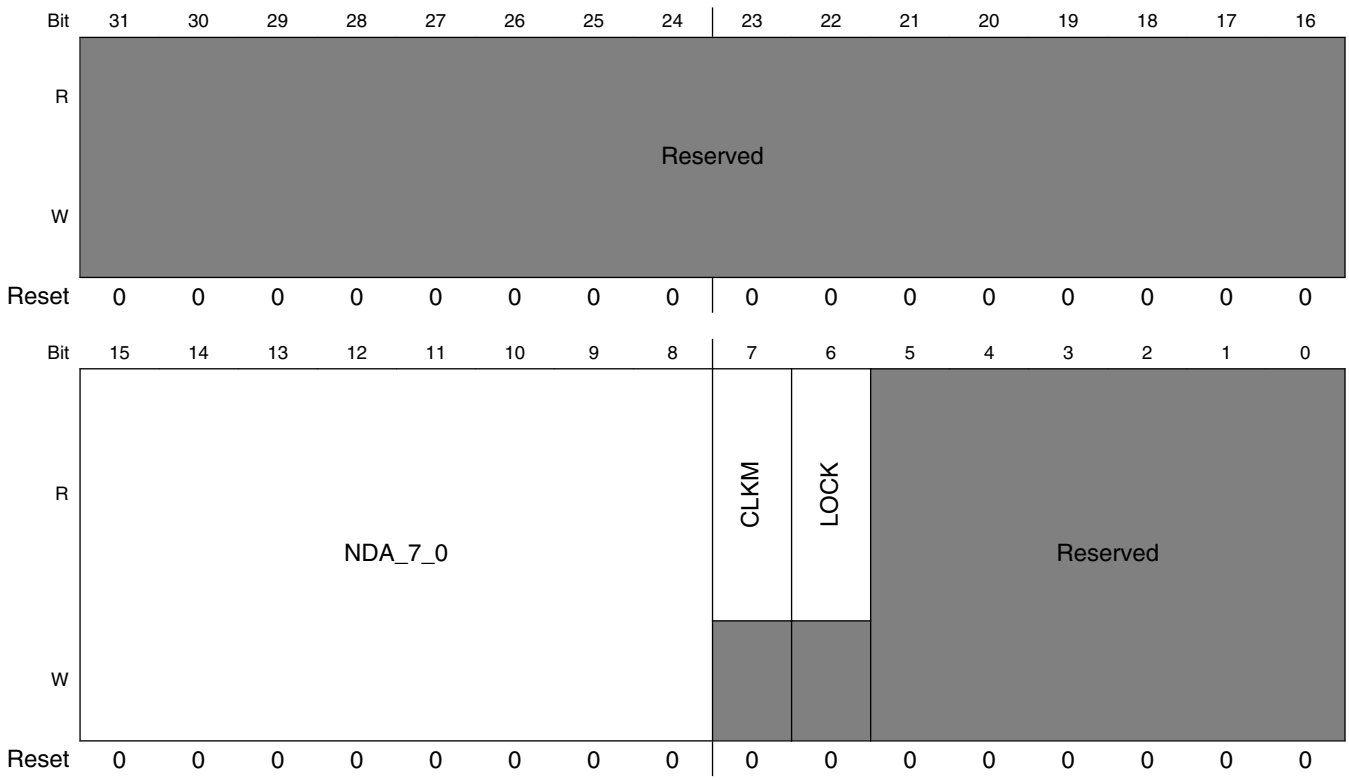
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				CKRCVBIAS_3_0				SDXMTBIAS_3_0				SDRCVBIAS_3_0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_MLBPC1 field descriptions

Field	Description
31–12 -	This field is reserved. Reserved
11–8 CKRCVBIAS_3_0	Clock receiver bias control (for MediaLB 6-pin interface). Must be written to 0xC when MediaLB 6-pin is initialized (final value needs to be determined through characterization). This value is driven on MLB_CLK, MLB_CLK_N and MLB_CLK_P output pins and has no internal function.
7–4 SDXMTBIAS_3_0	Signal/Data transmitter bias control (for MediaLB 6-pin interface). Must be written to 0xC when MediaLB 6-pin is initialized (final value needs to be determined through characterization). This value is driven on MLB_DATA, MLB_DATA_N and MLB_DATA_P output pins and has no internal function.
SDRCVBIAS_3_0	Signal/Data receiver bias control (for MediaLB 6-pin interface). Must be written to 0xC when MediaLB 6-pin is initialized (final value needs to be determined through characterization). This value is driven on MLB_SIG, MLB_SIG_N and MLB_SIG_P output pins and has no internal function.

44.6.10 MediaLB Control 1 Register (MLB_MLBC1)

Address: 218_C000h base + 3Ch offset = 218_C03Ch



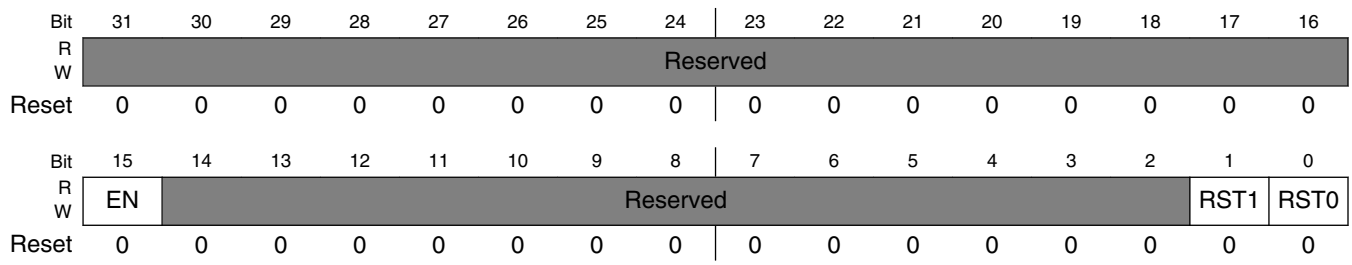
MLB_MLBC1 field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15–8 NDA_7_0	Node device address. Used for system commands directed to individual MediaLB nodes.
7 CLKM	MediaLB clock missing status. Set when MLB_CLK (MediaLB clock) is not toggling at the pin; cleared by software.
6 LOCK	MediaLB lock error status. Set when MediaLB is unlocked; cleared by software.
- 	This field is reserved. Reserved

44.6.11 HBI Control Register (MLB_HCTL)

The HC can control and monitor general operation of the HBI block by reading and writing the HBI Control Register (HCTL) through the I/O interface. Each bit of HCTL is read/write.

Address: 218_C000h base + 80h offset = 218_C080h



MLB_HCTL field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 EN	HBI enable 1 enabled 0 disabled
14–2 -	This field is reserved. Reserved
1 RST1	AGU1 software reset 1 reset 0 active
0 RST0	AGU0 software reset 1 reset 0 active

44.6.12 HBI Channel Mask 0 Register (MLB_HCMR0)

The HC can control which channel(s) are able to generate an HBI interrupt by writing the HBI Channel Mask Registers (HCMRn). The HCMRn registers mask the channel interrupt on the *hbi_hintb* signal (i.e. *hbi_hintb* will not become active for any masked channel). Each bit of HCMRn is read/write.

Address: 218_C000h base + 88h offset = 218_C088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CHM_31_0_P																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_HCMR0 field descriptions

Field	Description
CHM_31_0_P	Bitwise channel mask bit 0 masked 1 unmasked

44.6.13 HBI Channel Mask 1 Register (MLB_HCMR1)

Address: 218_C000h base + 8Ch offset = 218_C08Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CHM_63_32																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

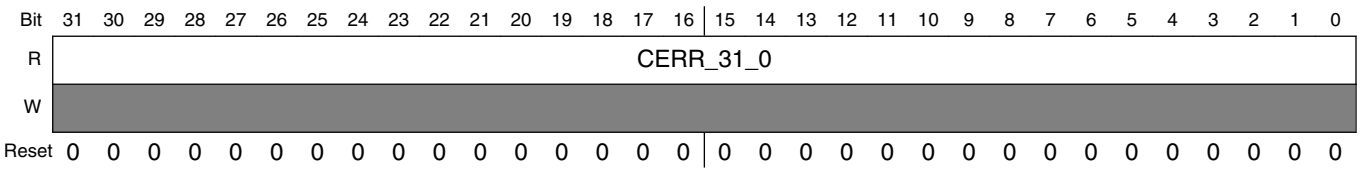
MLB_HCMR1 field descriptions

Field	Description
CHM_63_32	Bitwise channel mask bit 0 masked 1 unmasked

44.6.14 HBI Channel Error 0 Register (MLB_HCER0)

The HBI Channel Error Registers (HCERn) indicate which channel(s) have encountered fatal errors.

Address: 218_C000h base + 90h offset = 218_C090h



MLB_HCER0 field descriptions

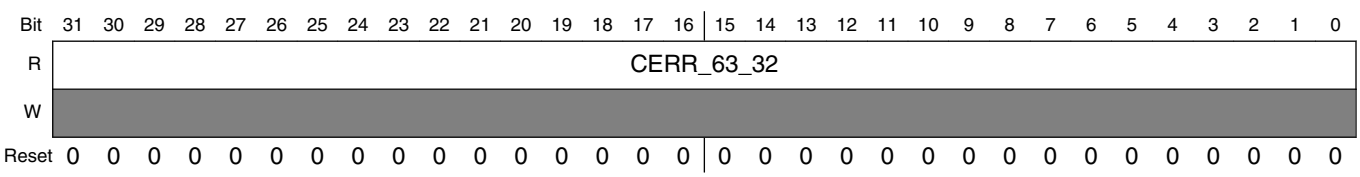
Field	Description
CERR_31_0	Bitwise channel error bit

44.6.15 HBI Channel Error 1 Register (MLB_HCER1)

HCERn status bits are set when hardware detects hardware errors on the given logical channel, including:

- Channel opened, but not enabled,
- Channel programmed with invalid channel type, or
- Out-of-range PML for asynchronous or control Tx channels

Address: 218_C000h base + 94h offset = 218_C094h



MLB_HCER1 field descriptions

Field	Description
CERR_63_32	Bitwise channel error bit

44.6.16 HBI Channel Busy 0 Register (MLB_HCBR0)

The HC can determine which channel(s) are busy by reading the HBI Channel Busy Registers (HCBRn). An HBI channel is busy if:

- it is currently loaded into one of the two AGUs
- the channel is enabled, CE = 1 from the Channel Allocation Table (Table 44-5), and
- the DMA is active

When an HBI channel is busy, hardware may write back its local copy of the channel descriptor at any time. System software should not write a CDT descriptor for a channel that is busy. Only two HBI channels can be busy at any given time. Each bit of HCBRn is read-only.

Address: 218_C000h base + 98h offset = 218_C098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHB_31_0																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_HCBR0 field descriptions

Field	Description
CHB_31_0	Bitwise channel busy bit 0 idle 1 busy

44.6.17 HBI Channel Busy 1 Register (MLB_HCBR1)

Address: 218_C000h base + 9Ch offset = 218_C09Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHB_63_32																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_HCBR1 field descriptions

Field	Description
CHB_63_32	Bitwise channel busy bit 0 idle 1 busy

44.6.18 MIF Data 0 Register (MLB_MDAT0)

Address: 218_C000h base + C0h offset = 218_C0C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA_31_0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_MDAT0 field descriptions

Field	Description
DATA_31_0	CTR data - bits[31:0] of 128-bit entry or DBR data - bits[7:0] of 8-bit entry

44.6.19 MIF Data 1 Register (MLB_MDAT1)

Address: 218_C000h base + C4h offset = 218_C0C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA_63_32																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_MDAT1 field descriptions

Field	Description
DATA_63_32	CTR data - bits[63:32] of 128-bit entry

44.6.20 MIF Data 2 Register (MLB_MDAT2)

Address: 218_C000h base + C8h offset = 218_C0C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA_95_64																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_MDAT2 field descriptions

Field	Description
DATA_95_64	CTR data - bits[95:64] of 128-bit entry

44.6.21 MIF Data 3 Register (MLB_MDAT3)

Address: 218_C000h base + CCh offset = 218_C0CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA_127_96																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_MDAT3 field descriptions

Field	Description
DATA_127_96	CTR data - bits[127:96] of 128-bit entry

44.6.22 MIF Data Write Enable 0 Register (MLB_MDWE0)

Address: 218_C000h base + D0h offset = 218_C0D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_31_0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_MDWE0 field descriptions

Field	Description
MASK_31_0	Bitwise write enable for CTR data - bits[31:0] 0 disabled 1 enabled

44.6.23 MIF Data Write Enable 1 Register (MLB_MDWE1)

Address: 218_C000h base + D4h offset = 218_C0D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_63_32																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_MDWE1 field descriptions

Field	Description
MASK_63_32	Bitwise write enable for CTR data - bits[63:32] 0 disabled 1 enabled

44.6.24 MIF Data Write Enable 2 Register (MLB_MDWE2)

Address: 218_C000h base + D8h offset = 218_C0D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_95_64																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_MDWE2 field descriptions

Field	Description
MASK_95_64	Bitwise write enable for CTR data - bits[95:64] 0 disabled 1 enabled

44.6.25 MIF Data Write Enable 3 Register (MLB_MDWE3)

Address: 218_C000h base + DCh offset = 218_C0DCh

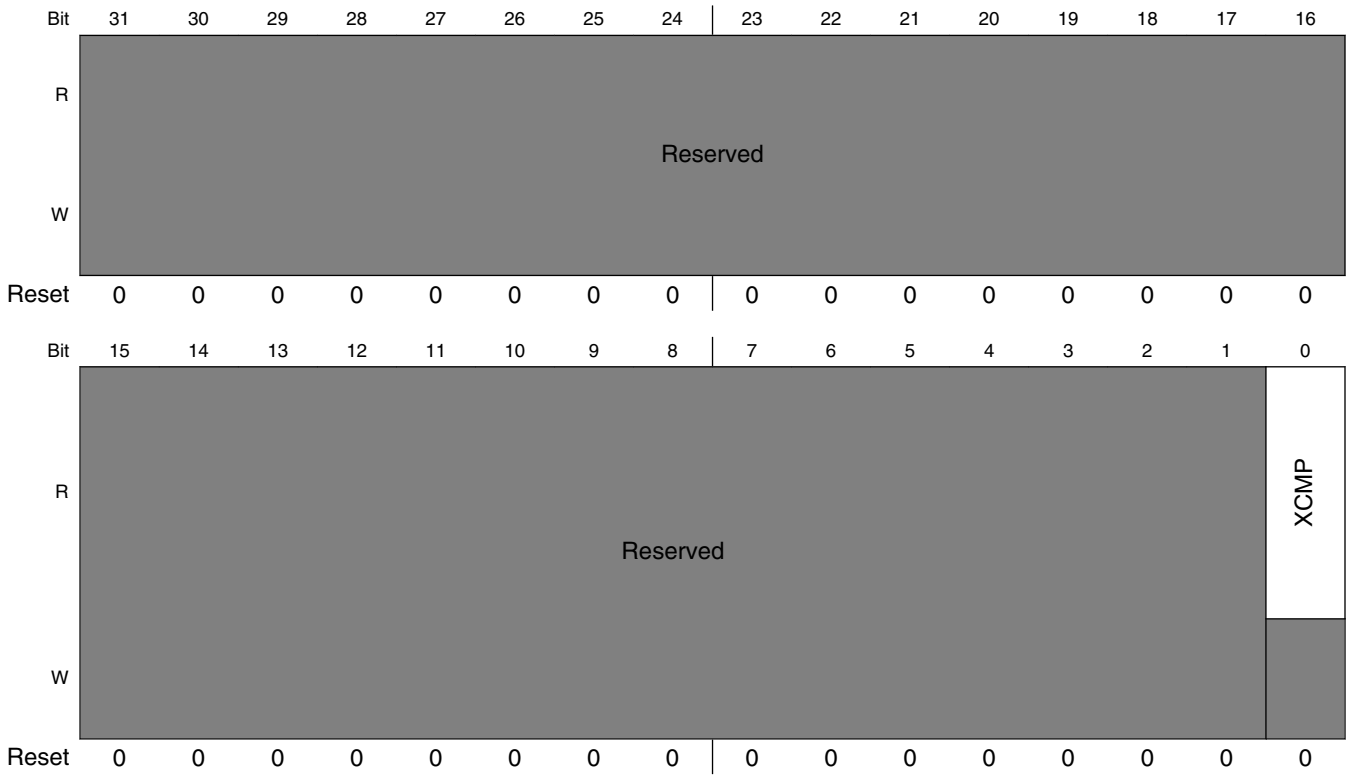
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_127_96																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_MDWE3 field descriptions

Field	Description
MASK_127_96	Bitwise write enable for CTR data - bits[127:96] 0 disabled 1 enabled

44.6.26 MIF Control Register (MLB_MCTL)

Address: 218_C000h base + E0h offset = 218_C0E0h

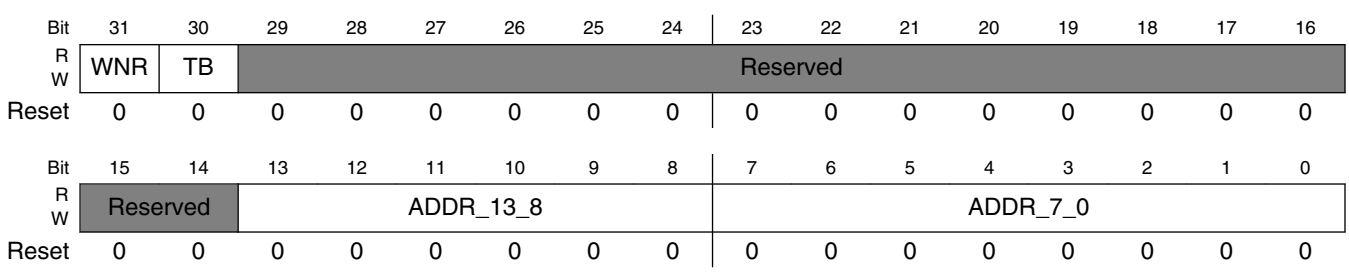


MLB_MCTL field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 XCMP	Transfer complete (write 0 to clear)

44.6.27 MIF Address Register (MLB_MADR)

Address: 218_C000h base + E4h offset = 218_C0E4h



MLB_MADR field descriptions

Field	Description
31 WNR	Write-Not-Read selection 0 read 1 write
30 TB	Target location bit 0 selects CTR 1 selects DBR
29–14 -	This field is reserved. Reserved
13–8 ADDR_13_8	DBR address of 8-bit entry - bits[13:8]
ADDR_7_0	CTR address of 128-bit entry or DBR address of 8-bit entry - bits[7:0]

44.6.28 AHB Control Register (MLB_ACTL)

The AHB Control (ACTL) register is written by the HC to configure the AMBA AHB block for channel interrupts. ACTL contains three configuration fields, one is used to select the DMA mode, one is used to mux channel interrupts onto a single interrupt signal, and the last selects the method of clearing channel interrupts (either software or hardware).

Address: 218_C000h base + 3C0h offset = 218_C3C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											MPB	-	DMA_MODE	SMX	SCE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_ACTL field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 MPB	DMA Packet buffering mode. 0 Single-packet mode 1 Multiple-packet mode
3 -	Reserved.
2 DMA_MODE	DMA Mode: 0 DMA Mode 0 1 DMA Mode 1
1 SMX	AHB interrupt mux enable: 0 ACSR0 generates an interrupt on *ahb_int[0]*; ACSR1 generates an interrupt on *ahb_int[1]* 1 ACSR0 and ACSR1 generate an interrupts on *ahb_int[0]* only
0 SCE	Software clear enable: 0 Hardware clears interrupt after a ACSRn register read 1 Software clears interrupt

44.6.29 AHB Channel Status 0 Register (MLB_ACSR0)

The AHB Channel Status (ACSRn) registers contain interrupt bits for each of the 64 physical channels. When an ACSRn register bit is set, it indicates that the corresponding physical channel has an interrupt pending.

An AHB interrupt is triggered when either DNEn or ERRn is set within the AHB Channel Descriptor. The HC is notified of the channel interrupt via *ahb_int[1:0]*. When an interrupt occurs in ACSR0 (for channels 31 to 0) *ahb_int[0]* is set. When an interrupt occurs in ACSR1 (for channels 63 to 32) *ahb_int[1]* is set.

Interrupts in ACSR0 and ACSR1 can be optionally muxed onto a single interrupt signal, *ahb_int[0]*, if ACTL.SMX = 1. If ACTL.SCE = 0, hardware automatically clears the interrupt bit(s) after the HC reads the ACSRn register. Alternatively, if ACTL.SCE = 1, software must write a 1 to the appropriate bit(s) of ACSRn to clear the interrupt(s).

Address: 218_C000h base + 3D0h offset = 218_C3D0h

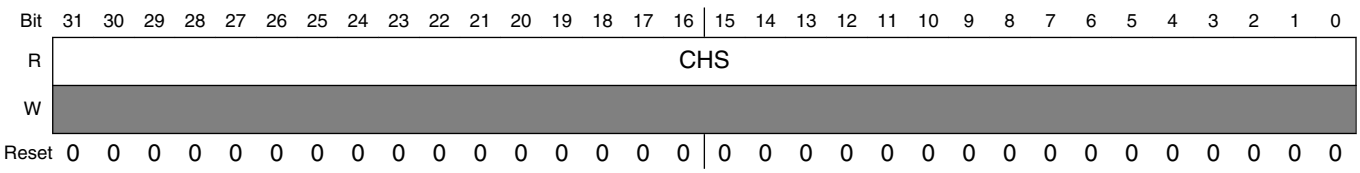
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHS																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_ACSR0 field descriptions

Field	Description
CHS	Interrupt status for logical channels 31 to 0: 0 None 1 Interrupt

44.6.30 AHB Channel Status 1 Register (MLB_ACSR1)

Address: 218_C000h base + 3D4h offset = 218_C3D4h



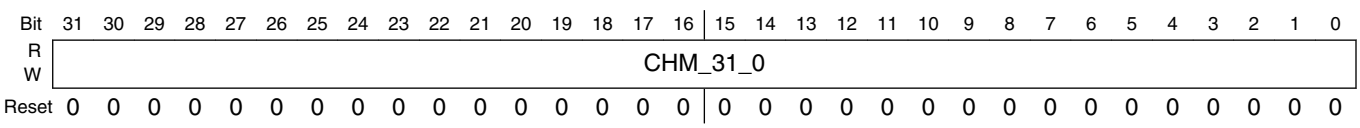
MLB_ACSR1 field descriptions

Field	Description
CHS	Interrupt status for logical channels 63 to 32: 0 None 1 Interrupt

44.6.31 AHB Channel Mask 0 Register (MLB_ACMR0)

Using the AHB Channel Mask (ACMRn) register, the HC can control which channel(s) generate interrupts on *ahb_int[1:0]*. All ACMRn register bits default as '0' ("masked"); therefore, the HC must initially write ACMRn to enable interrupts. Each bit of ACMRn is read/write accessible.

Address: 218_C000h base + 3D8h offset = 218_C3D8h



MLB_ACMR0 field descriptions

Field	Description
CHM_31_0	Bitwise channel mask bit:

MLB_ACMR0 field descriptions (continued)

Field	Description
	0 Masked 1 Unmasked

44.6.32 AHB Channel Mask 1 Register (MLB_ACMR1)

Address: 218_C000h base + 3DCh offset = 218_C3DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CHM																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_ACMR1 field descriptions

Field	Description
CHM	Bitwise channel mask bit: 0 Masked 1 Unmasked

Chapter 45
Multi Mode DDR Controller (MMDC)

45.1 Overview

MMDC is a multi-mode DDR controller that supports DDR3/DDR3L x16/x32/x64 and LPDDR2 two channel x16/x32 memory types. MMDC is configurable, high performance, and optimized.

The following figure shows the MMDC block diagram.

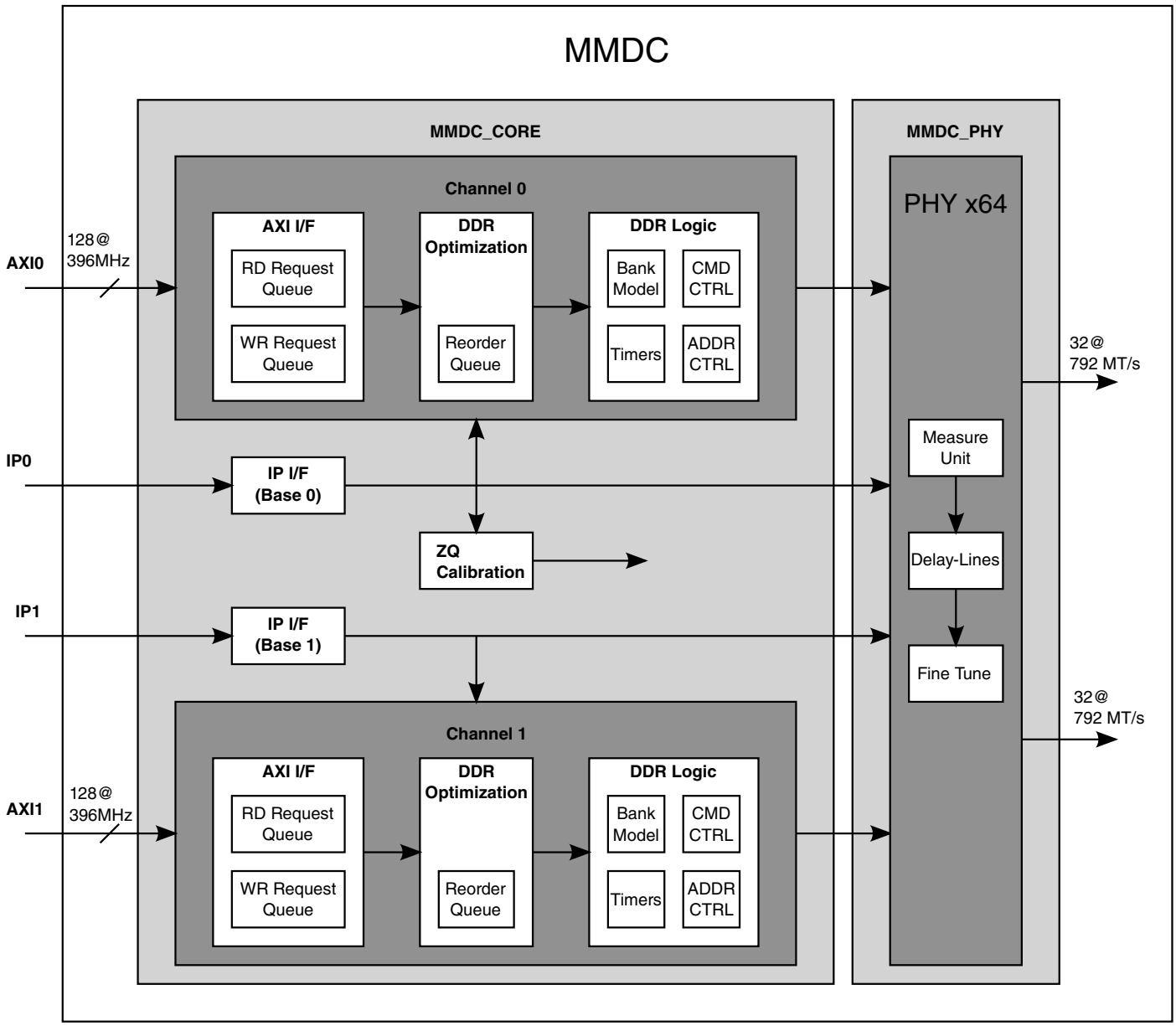


Figure 45-1. MMDC block diagram

MMDC consists of a core (MMDC_CORE) and PHY (MMDC_PHY).

- The core is responsible for communication with the system through an AXI interface, DDR command generation, DDR command optimizations, and a read/write data path.
- The PHY is responsible for the timing adjustment; it uses special calibration mechanisms to ensure data capture margin at a clock rate of up to 396 MHz.

NOTE

The core is composed of two channels, but both channels are only active in LPDDR2 mode. If DDR3 mode is selected,

channel 1 is not activated and the MMDC communicates with the system through AXI port0.

The internal memory map (configuration registers) of the MMDC can be configured through two IP channels (IP0 and IP1).

45.1.1 MMDC feature summary

The table found here summarizes the MMDC features.

Table 45-1. MMDC feature summary

Feature	Details
DDR standards	• DDR3L, DDR3 x16, x32, x64 (includes SODIMM) • LPDDR2 2-channels x16, x32 • Does not support LPDDR1MDDR or DDR2

DDR interface	• x16, x32, x64 data bus width
	• Density per DDR device of 256 Mbits–8 Gbits with the following column and row combinations:
	  • Column size of 8–12 bits
	  • Row size of 11–16 bits
	• Two chip selects per channel (one chip select for DDR3/DDR3L, LPDDR2)
	• Up to 4 Gbytes of address space with configurable partitioning between CS0 and CS1 (for LPDDR2 2ch x32 up to 2 Gbytes per channel)
	• Supports burst length of 8 (aligned) for DDR3
	• Supports burst length of 4 for LPDDR2

DDR performance	• MMDC running at up to 396 MHz (792MT/s), see CCM block for actual clock frequencies supported. • Supports Real-Time priority by means of QoS sideband priority signals from the chip to enable various priority levels in the re-ordering mechanism: real-time, latency sensitive, normal priority. • Page hit/page miss optimizations • Consecutive read/write access optimizations • Supports deep read and write request queues to enable bank prediction. • Drives back the critical word in a read transaction as soon as it is received by the DDR device (does not wait until the whole data phase has been completed). • Keeps tracking of open memory pages • Supports bank interleaving • Special optimization in case of non-aligned wrap accesses in DDR3 mode (burst length 8) **NOTE:** Due to reordering and optimization mechanisms (per different AXI Identifier (ID)), the transactions towards the DDR device may be driven in a different ID order than was received by the AXI master. In a similar fashion, the write response, read response or read data may be driven to the AXI master in a different ID order.
AXI interface	• AXI bus compliant • Supports bus transfers of 8, 16 , 32 and 128 bits (single accesses and bursts) running at 396 MHz. • Supports AXI bursts length of up to 16 • Supports burst types of WRAP, INCR and FIXED • Supports 16 bits AXI ID • Write data interleave depth is 1 (no support for Write Data Interleave) • Supports write data before address

Table continues on the next page...

Table 45-1. MMDC feature summary (continued)

Feature	Details

	• Supports buffered/non-buffered accesses (AWCACHE[0] = 0b means a non-bufferable access and AWCACHE[0] = 1b means a bufferable access). The rest of the CACHE options are not supported
	  • To keep data access coherency between write and read access of the same master, the response signal is sent as follows:
	  • Bufferable write access—BRESP will be sent when last data of the access has entered the MMDC.
	  • Non-bufferable write access—BRESP will be sent when the data was physically written into the external memory device.
	• Supports four exclusive monitors per configurable ID for only a single access with a size of up to 64 bits
	• Supports AXI responses as follows:
	  • Okay in case the access has been successful or exclusive access failure
	  • Slave error in case of security violation
	  • Exclusive okay in case the read or the write portion of an exclusive access has been successful

DDR calibration and delay-lines.	• Supports various calibration processes which can be performed either automatically (hardware) or manually (software) towards either CS0 or CS1. (At the end of the process the delay-lines will work with one set of results.) The following calibration processes are supported:
	  • ZQ calibration for external DDR device (in DDR3 through ZQ calibration command and in LPDDR2 through MRW command)
	  • Can be handled automatically for ZQ Short (periodically) and ZQ Long (at exit from self-refresh)
	  • Can be handled manually at ZQ INIT
	  • ZQ calibration for DDR I/O pads for calibrating the DDR driving strength
	  • The sequence can be handled automatically by hardware
	  • The sequence can be handled step by step manually by software
	  • Read data calibration. Adjustment of read DQS with read data byte.
	  • Read DQS gating calibration for DDR3 only. Adjustment of DQS gate with read preamble window.
	  • Write data calibration. Adjustment of write DQS with write data byte.
	  • Write leveling calibration. Adjustment of write DQS with CK (DDR differential clock).
	  • Read fine tuning. Adjustment of up to 7 delay-line units for each read data bit.
	  • Write fine tuning. Adjustment of up to 3 delay-line units for each read data bit.
	  • Periodic delay-line measurement for keeping its accuracy during refresh interval.
	  • Additional fine tuning delay lines to adjust DDR clock delay, DDR clock duty cycle, DQS duty cycle.

Power saving	• Support of dynamic voltage, frequency change and self-refresh mode entry through hardware and software negotiation with the system (request/acknowledge handshake)
	  • Upon hardware or software self-refresh request assertion, further AXI requests are blocked (even before the assertion of the acknowledge).
	  • During self-refresh mode the system may deassert the operating clock of the MMDC for power saving.
	  • During self-refresh mode the clock (CK) that is driven to the DDR device will be gated for power saving.
	• Supports automatic self-refresh and power down entry and exit
	  • In automatic self-refresh, the internal operating clock will be gated for power saving.
	• Supports fast and slow precharge power down in DDR3
	• Automatic active and precharge power down timer per chip select (one chip select can enter power down while the other is still working)

Table continues on the next page...

Table 45-1. MMDC feature summary (continued)

Feature	Details
	• While CS (chip-select) is inactive (high) the command and address buses are not toggling for power saving. • While DM (data masking) is high the associated DQ bus is not toggling (driven to "0") for power saving.

DDR general	• Configurable timing parameters
	• Configurable refresh scheme
	• Page boundary crossing support
	  • Automatically generates precharge command and activates the next row
	• Supports various ODT control schemes
	  • Assertion or deassertion of ODT control per read or write accesses and for active or passive CS (chip-select)
	• Supports MRW and MRR commands for LPDDR2
	• Software control in LPDDR2 mode for switching to derated timing parameters and/or update the refresh rate according to temperature sensor
	• Debug and profiling capabilities

45.2 External Signals

The table found here describes the external signals of MMDC.

Table 45-2. MMDC External Signals

Signal	Description	Pad	Mode	Direction
DRAM_ADDR[15:0]	Address Bus Signals	DRAM_A[15:0]	No Muxing	O
DRAM_CAS	Column Address Strobe Signal	DRAM_CAS	No Muxing	O
DRAM_CS[1:0]	Chip Selects	DRAM_CS[1:0]	No Muxing	O
DRAM_DATA[63:0]	Data Bus Signals	DRAM_D[63:0]	No Muxing	I/O
DRAM_DQM[7:0]	Data Mask Signals	DRAM_DQM[7:0]	No Muxing	O
DRAM_ODT[1:0]	On-Die Termination Signals	DRAM_SDODT[1:0]	No Muxing	O
DRAM_RAS	Row Address Strobe Signal	DRAM_RAS	No Muxing	O
DRAM_RESET	Reset Signal	DRAM_RESET	No Muxing	O
DRAM_SDBA[2:0]	Bank Select Signals	DRAM_SDBA[2:0]	No Muxing	O
DRAM_SDCKE[1:0]	Clock Enable Signals	DRAM_SDCKE[1:0]	No Muxing	O
DRAM_SDCLK[1:0]_N	Negative Clock Signals	DRAM_SDCLK_[1:0]_B	No Muxing	O
DRAM_SDCLK[1:0]_P	Positive Clock Signals	DRAM_SDCLK_[1:0]	No Muxing	O
DRAM_SDQS[7:0]_N	Negative DQS Signals	DRAM_SDQS[7:0]_B	No Muxing	I/O
DRAM_SDQS[7:0]_P	Positive DQS Signals	DRAM_SDQS[7:0]	No Muxing	I/O
DRAM_SDWE	WE signal	DRAM_SDWE	No Muxing	O

45.3 Clocks

The table found here describes the clock sources for MMDC.

See Clock Controller Module (CCM) for clock setting, configuration and gating information.

NOTE

The terms *clocks* and *cycles* are used interchangeably and refer to the clock period of the main ddr clock (mmdc_ch0_axi_clk_root), commonly referred to as the DDR frequency.

Table 45-3. MMDC Clocks

Clock name	Clock Root	Description
aclk_fast_core_p0	mmdc_ch0_axi_clk_root	Fast clock (channel 1)
aclk_fast_core_p1	GND	Fast clock (channel 2)
ipg_clk_p0	ipg_clk_root	Peripheral clock (channel 1)
ipg_clk_p1	ipg_clk_root	Peripheral clock (channel 2)
aclk_fast_phy_p0	mmdc_ch0_axi_clk_root	Fast clock (channel 1 - PHY)
aclk_fast_phy_p1	GND	Fast clock (channel 2 - PHY)

45.4 Functional Description

This section provides a complete functional description of the block.

45.4.1 Write/Read data flow

45.4.1.1 Write data flow

1. Write requests are received into an 8 entry request FIFO. Access is received only when there are at least two available entries. Each entry holds all of the AXI attributes.
   - If the burst length is greater than 8, the access splits into two accesses: one with burst length 8 and the other with the remainder.
   - The access can be performed as soon as the entire data phase of the associated write request is completed (all data beats were received).

2. A simple round-robin arbitration between the pending read and write accesses is performed, and the pointer to this stage's winner access is sent to the re-ordering buffer.
3. The reordering mechanism is activated to find the winner access, which is the access that best utilizes the DDR bus, based on its dynamic score. For further information see Dynamic scoring mode (Arbitration Winning Conditions).
4. The winner write access at the previous stage is received and is held for dispatch to the DDR logic.
5. When the DDR command control unit is ready to accept the write request, it issues (if needed) a precharge/active command to the DDR device according to the status of the bank model and the parameters of the timers.
6. The DDR logic drives the associated data to the DDR device through the DDR PHY.

45.4.1.2 Read data flow

1. Read requests are received into a 16 entry request FIFO in MMDC if there are at least two available entries. Each entry holds all of the AXI attributes.

NOTE

If the burst length is greater than 8, the access splits into 2 accesses (one with burst length 8 and the other with the remainder).

2. A simple round-robin arbitration between the pending read and write accesses is performed and the pointer to this phase's winner access is sent to the re-ordering buffer.
3. The reordering mechanism is activated to find the winner access, which is the access that best utilizes the DDR bus, based on its dynamic score. For further information see Dynamic scoring mode (Arbitration Winning Conditions).
4. The winner read access at the previous stage is sampled and is held for dispatch to the DDR logic. This read access will be dispatched when there is at least one free slot in the read data buffer to store the data.
5. When the DDR command control unit is ready to accept the read request, it issues (if needed) a precharge/active command to the DDR device according to the status of the bank model and the parameters of the timers.
6. The MMDC PHY samples the read data, and the DDR logic transfers the data to the associated slot in the read data buffer.
7. MMDC transfers the data back to the master.

45.4.2 MMDC initialization

Because the MMDC is disabled when the chip exits reset, no clock is driven to the DDR device and the whole interface towards the DDR device is inactive. The following steps are required to activate the MMDC properly.

NOTE

To guarantee that the DRAM_RESET and DRAM_SDCKE signals are kept low during the power-up and reset sequences of the chip (as defined by JEDEC), you must connect those signals to pull-down resistors.

1. Set MDSCR[CON_REQ], which sets the configuration request; note that because the MMDC is disabled, there is no need to poll the configuration acknowledge bit at MDSCR[CON_ACK].
2. Configure the desired timing parameters at the MDCFG0, MDCFG1, MDOTC, and MDCFG2 registers.
3. Configure the DDR type and other miscellaneous parameters at the MDMISC register.
4. Configure the required delay while leaving reset, at the MDOR register.
5. Configure the DDR physical parameters (density and burst length) at the MDCTL register.
6. Perform a ZQ calibration of the MMDC module to correctly initialize drive strengths.
7. Enable MMDC with the desired chip select at MDCTL[SDE_0] (for chip select 0) and MDCTL[SDE_1] (for chip select 1). At this point, MMDC starts the reset and initialization sequence related to DRAM_RESET/DRAM_SDCKE as defined by JEDEC.
8. Complete the initialization sequence as defined by JEDEC by issuing MRS/MRW commands for (ZQ, ODT, PRE, and so on). To issue those commands, configure the appropriate command and address at the MDSCR register.

NOTE

A Precharge All command must be issued prior to the MRW command to ensure robust DDR initialization. This command is required to be issued to both chip selects if two chip selects are utilized in the system.

9. Program the DDR mode registers by configuring the appropriate command and address at the MDSCR register.
10. Configure the power down and self-refresh entry and exit parameters at the MDPDC and MAPSR registers.
11. Configure the ZQ scheme at the MPZQHWCTRL and MPZQLP2CTL registers.
12. Configure and activate the periodic refresh scheme at the MDREF register.

13. Deassert the configuration request by clearing MDSCR[CON_REQ].

NOTE

Steps 1 through 6 are non-blocking and can be done in any order.

Upon completion of these steps, MMDC is ready for work and to process AXI accesses.

NOTE

To achieve better timing and better precision, it is recommended that users configure the MMDC PHY delay parameters by operating either the automatic or manual calibration process. Before starting any calibration process, you must disable the periodic refresh scheme (MDREF[REF_SEL] = 11) and then issue a manual refresh command by configuring MDSCR[CMD] to 2h. For further information, see Calibration Process.

45.4.3 Configuring the MMDC registers

To safely modify MMDC's internal configuration registers, MMDC must be placed into configuration mode.

Use the following steps to enter configuration mode.

1. Issue a configuration request by setting MDSCR[CON_REQ].
2. Poll on configuration acknowledge until it is set at MDSCR[CON_ACK].

At this point, MMDC enters configuration mode and accessing the MMDC registers is permitted.

NOTE

During configuration mode, MMDC prevents further AXI accesses from being acknowledged.

Upon deassertion of MDSCR[CON_REQ], MMDC leaves configuration mode and AXI accesses are processed.

45.4.4 MMDC Address Space

45.4.4.1 Address decoding

MMDC supports up to two consecutive chip selects, each with the same density.

In LPDDR2-2ch mode, up to two chip selects per channel are supported.

It is optional to configure the partition between the chip selects through MDASP[CS0_END].

The incoming AXI address bus is 32 bits. MMDC decodes each access as follows:

1. chip select
2. bank number
3. row number
4. column number

The following registers in the MMDC define the DDR address space:

- MDMISC[DDR_4_BANK]—Defines either 4 or 8 banks in the DDR device
- MDCTL[DSIZ]—Defines the DDR data bus width of x16, x32 or x64
- MDMISC[BI]—Defines whether bank interleaving is on or off
- MDCTL[COL]—Defines the column size of the DDR device
- MDCTL[ROW]—Defines the row size of the DDR device

The following tables show address decoding examples for x16 and x32 bit DDR devices when bank interleaving is both on and off. It is assumed that the configuration is as follows: 8 banks (3 bits), 15 bit assignment for the row, and 10 bit assignment for the column. The total density is 256 MWords (512 Mbytes for x16 and 1 Gbyte for x32).

NOTE

Chip selection is done by comparing the 7 most significant address bits (ARADDR[31:25]/AWADDR[31:25]) with MDASP[CS0_END].

Table 45-4. Address decoding—bank interleaving off

AXI ADDRESS	x16 DDR	x32 DDR
A29	—	BANK[2]
A28	BANK[2]	BANK[1]
A27	BANK[1]	BANK[0]
A26	BANK[0]	ROW[14]
A25	ROW[14]	ROW[13]
A24	ROW[13]	ROW[12]
A23	ROW[12]	ROW[11]
A22	ROW[11]	ROW[10]
A21	ROW[10]	ROW[9]

Table continues on the next page...

Table 45-4. Address decoding—bank interleaving off (continued)

AXI ADDRESS	x16 DDR	x32 DDR
A20	ROW[9]	ROW[8]
A19	ROW[8]	ROW[7]
A18	ROW[7]	ROW[6]
A17	ROW[6]	ROW[5]
A16	ROW[5]	ROW[4]
A15	ROW[4]	ROW[3]
A14	ROW[3]	ROW[2]
A13	ROW[2]	ROW[1]
A12	ROW[1]	ROW[0]
A11	ROW[0]	COL[9]
A10	COL[9]	COL[8]
A9	COL[8]	COL[7]
A8	COL[7]	COL[6]
A7	COL[6]	COL[5]
A6	COL[5]	COL[4]
A5	COL[4]	COL[3]
A4	COL[3]	COL[2]
A3	COL[2]	COL[1]
A2	COL[1]	COL[0]
A1	COL[0]	—
A0	—	—

Table 45-5. Address decoding—bank interleaving on

AXI ADDRESS	x16 DDR	x32 DDR
A29	—	ROW[14]
A28	ROW[14]	ROW[13]
A27	ROW[13]	ROW[12]
A26	ROW[12]	ROW[11]
A25	ROW[11]	ROW[10]
A24	ROW[10]	ROW[9]
A23	ROW[9]	ROW[8]
A22	ROW[8]	ROW[7]
A21	ROW[7]	ROW[6]
A20	ROW[6]	ROW[5]
A19	ROW[5]	ROW[4]
A18	ROW[4]	ROW[3]
A17	ROW[3]	ROW[2]
A16	ROW[2]	ROW[1]

Table continues on the next page...

Table 45-5. Address decoding—bank interleaving on (continued)

AXI ADDRESS	x16 DDR	x32 DDR
A15	ROW[1]	ROW[0]
A14	ROW[0]	BANK[2]
A13	BANK[2]	BANK[1]
A12	BANK[1]	BANK[0]
A11	BANK[0]	COL[9]
A10	COL[9]	COL[8]
A9	COL[8]	COL[7]
A8	COL[7]	COL[6]
A7	COL[6]	COL[5]
A6	COL[5]	COL[4]
A5	COL[4]	COL[3]
A4	COL[3]	COL[2]
A3	COL[2]	COL[1]
A2	COL[1]	COL[0]
A1	COL[0]	—
A0	—	—

NOTE

In cases where this is an access to a non-initialized or disconnected chip select, behavior may be unexpected.

45.4.4.2 Chip select settings

MMDC drives the incoming access to either CS0 or CS1 by comparing the 7 most significant address bits (ARADDR[31:25]/AWADDR[31:25]) with MDASP[CS0_END].

Generally, the total density per chip-select must be a power of two and the total density must be the same for each chip-select.

Creating 4 Gbyte address space with 2 Gbyte CS density and Creating 2 Gbyte address spaces with 1 Gbyte CS density show how to create a continous address space and configure the MMDC accordingly.

45.4.4.2.1 Creating 4 Gbyte address space with 2 Gbyte CS density

If the DDR memory space allocation is 4 Gbytes, only one configuration of chip select partition is allowed. The register MDASP[CS0_END] should be set to 2 Gbytes partition (16Gb).

The figure below shows the associated memory space. In the case of DDR3 x64, this address space can be acheived by connecting four devices per chip select. Each device is x16 with density of 4 Gbytes.

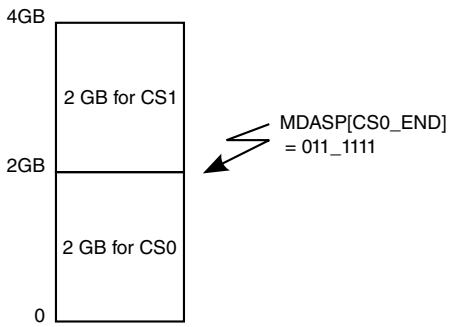


Figure 45-2. Chip select partion—2 Gbytes per chip select

45.4.4.2.2 Creating 2 Gbyte address spaces with 1 Gbyte CS density

If the DDR memory space allocation is 2 Gbytes, there are three options for configuring the chip select partition:MDASP[CS0_END] to 001_1111 (1 Gbyte), MDASP[CS0_END] to 011_1111 (2 Gbytes), and MDASP[CS0_END] to 101_1111 (3 Gbytes).

If DDR memory space allocation is 2 Gbytes, there are three options for configuring the chip select partition:

- MDASP[CS0_END] to 001_1111 (1 Gbyte)
- MDASP[CS0_END] to 011_1111 (2 Gbytes)
- MDASP[CS0_END] to 101_1111 (3 Gbytes)

The figure below shows the associated memory space:

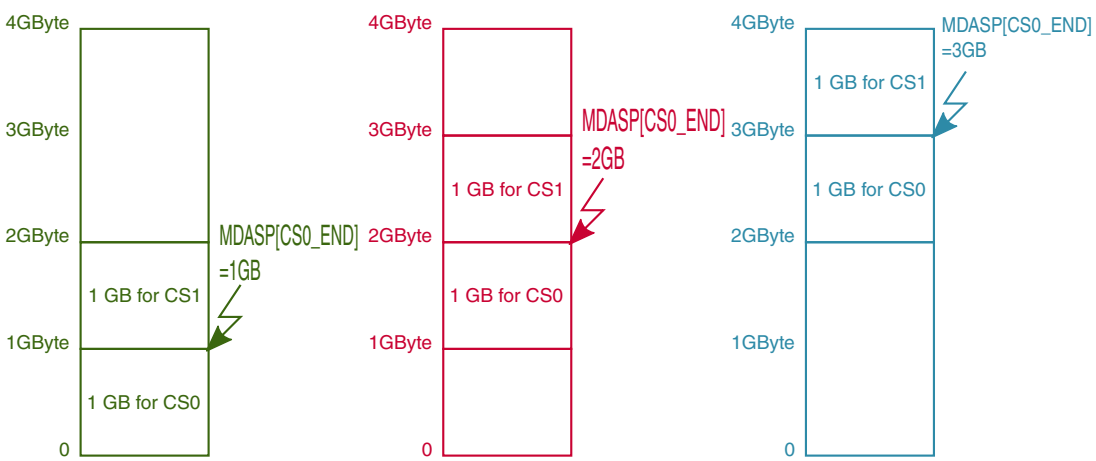


Figure 45-3. Chip select partion—1 Gbyte per chip select

45.4.4.3 Translation of AXI accesses to DDR accessess

45.4.4.3.1 Example 1

Assume the AXI read access has the following attributes:

- Wrap (arburst[1:0] = 10b)
- AXI size of 128 bits (arsize[2:0] = 100b)
- AXI length of 8 (arlen[3:0] = 0111b)
- AXI address with suffix B0h (non-aligned to AXI wrap boundary which is 16Bx8 = 128B = 0x80)

Toward DDR3(MDMISC[DDR_TYPE] = 00b) with the following attributes:

- x32 (MDCTL[DSIZ] = 01b)
- burst length of 8 (MDCTL[BL] = 1b)

In this case, the AXI wrap boundary is every 16Bx8=128B (0x80) and the DDR wrap boundary is every 4Bx8=32B(0x20).

The master expects to fetch the data that is associated with the following addresses: 0xB0, 0xC0, 0xD0, 0xE0, 0xF0, 0x80, 0x90, 0xA0.

The first aligned AXI address that is associated with suffix 0xB0 is with suffix 0x80, and the last wrap AXI address is with suffux 0xFF. Because the AXI master expects to get the first data from AXI address with suffix 0xB0, the MMDC issues the following accesses toward the DDR:

- Read access toward logic address with suffix 0xA0 (DDR boundary is 0x20 and 0xA0 is the closest to 0xB0)

NOTE

Logic address is the address of the column normalized to 1 byte.

- Read access toward logic address with suffix 0xC0
- Read access toward logic address with suffix 0xE0
- Read access toward logic address with suffix 0x80

The MMDC breaks the AXI access into four DDR accesses and returns the read data associated with address 0xB0 to the master first. The read data fetched from address 0xA0 is stored in the internal buffers and is driven back to the master at the end.

45.4.4.3.2 Example 2

Assume the AXI write access has the following attributes:

- Increment (awburst[1:0]=2'b01)
- AXI size of 64bits (awsize[2:0]=3'b011)
- AXI length of 8 (awlen[3:0]=4'b0111)
- AXI address with suffix 0xB0 (aligned as the size of the increment is 8B)

Toward DDR3(MDMISC[DDR_TYPE]=2'b00) with the following attributes:

- x64 (MDCTL[DSIZ]=2'b10)
- burst length of 8 (MDCTL[BL]=1'b1)

In this case, the AXI alignment is every 8B and the DDR boundary is every 8Bx8=64B(0x40).

The master expects to write the data to the following addresses: 0xB0, 0xB8, 0xC0, 0xC8, 0xD0, 0xD8, 0xE0, 0xE8.

The MMDC will issue the following accesses toward the DDR:

- Write access toward logic address with suffix 0x80 (DDR boundary is 0x40 and 0x80 is the closest to 0xB0) while address 0x80 to 0xAF are masked by DM (data masking signal).

NOTE

Logic address is the address of the column normalized to 1 byte.

- Write access toward logic address with suffix 0xC0 while addresses 0xF0 through 0xFF are masked by DM (data masking signal)

The MMDC will break the AXI access into two DDR accesses.

45.4.4.3.3 Example 3

Assume the AXI write access has the following attributes:

- Wrap (awburst[1:0]=2'b10)
- AXI size of 128bits (awsize[2:0]=3'b100)
- AXI length of 4 (awlen[3:0]=4'b0011)
- AXI address with suffix 0x80 (aligned)

Toward DDR3(MDMISC[DDR_TYPE]=2'b00) with the following attributes:

- x64 (MDCTL[DSIZ]=2'b10)
- burst length of 8 (MDCTL[BL]=1'b1)

In this case, the AXI wrap boundary is every 16Bx4=64B (0x40) and the DDR wrap boundary is every 8x8=64B(0x40).

The master expects to write the data to the following addresses: 0x80, 0x90, 0xA0, 0xB0.

Because the AXI wrap boundary and DDR wrap boundary are similar and the starting AXI address is aligned, the MMDC will issue only one access toward the DDR as follows:

- Write access towards logic address with suffix 0x80

NOTE

Logic address is the address of the column normalized to 1 byte.

45.4.4.3.4 Example 4

Assume the AXI write access has the following attributes:

- Increment (awburst[1:0]=2'b01)
- AXI size of 64bits (awsize[2:0]=3'b011)
- AXI length of 2 (awlen[3:0]=4'b0001)
- AXI address with suffix 0x5 (non aligned)

Toward DDR3(MDMISC[DDR_TYPE]=2'b00) with the following attributes:

- x32 (MDCTL[DSIZ]=2'b10)
- burst length of 8 (MDCTL[BL]=1'b1)

In this case the AXI alignment is every 8B (0x8) and the DDR boundary is every 4Bx8=32B(0x20).

The master expects to write the data to the following addresses: 0x5 (with WSTRB=0xE0), 0x8 (till 0xF).

The MMDC will issue one access toward the DDR as follows:

Write access toward logic address with suffix 0x0 (DDR boundary is 0x20 and 0x0 is the closest to 0x0) while address 0x0 till 0x4 are masked by DM (data masking signal) and address 0x10 till 0x1F are also masked by DM.

45.4.4.3.5 Example 5

Assume AXI write access has the following attributes:

- Increment (awburst[1:0]=2'b01)
- AXI size of 64bits (awsize[2:0]=3'b011)

- AXI length of 7 (awlen[3:0]=4'b0001)
- AXI address with suffix 0x10 (aligned)

Toward DDR3 (MDMISC[DDR_TYPE]=2'b00) with the following attributes:

- x64 (MDCTL[DSIZ]=2'b10)
- burst length of 8 (MDCTL[BL]=1'b1)

In this case the AXI alignment is every 8B (0x8) and the DDR boundary is every 8Bx8=64B(0x40).

The master expects to write the data to the following addresses: 0x10, 0x18, 0x20, 0x28, 0x30, 0x38, 0x40, 0x48.

Because the AXI access is not aligned to DDR boundary, which is every 0x40, the MMDC will issue two accesses toward the DDR as follows:

- Write access toward logic address with suffix 0x0 while address 0x0 till 0xF are masked by DM (data masking signal).

NOTE

Logic address is the address of the column normalized to 1 byte.

- Write access towards logic address with suffix 0x40 while addresses 0x50 till 0x7F are masked by DM (data masking signal).

45.4.4.4 Address mirroring

When enabling this feature, address bits DRAM_A3, DRAM_A4, DRAM_A5, DRAM_A6, DRAM_A7, DRAM_A8, DRAM_SDBA0, and DRAM_SDBA1 behave differently according to the associated chip select.

This feature facilitates PCB board routing for devices on chip select 1, which are typically populated on the opposite side of the PCB from the devices on chip select 0.

NOTE

This feature is only supported for DDR3 memories. It is not supported for LPDDR2 memories.

The following table specifies the address mirroring options:

Table 45-6. Address mirroring options

MMDC pin	Chip select 0 pin	Chip select 1 pin
DRAM_A3	DRAM_A3	DRAM_A4

Table continues on the next page...

Table 45-6. Address mirroring options (continued)

MMDC pin	Chip select 0 pin	Chip select 1 pin
DRAM_A4	DRAM_A4	DRAM_A3
DRAM_A5	DRAM_A5	DRAM_A6
DRAM_A6	DRAM_A6	DRAM_A5
DRAM_A7	DRAM_A7	DRAM_A8
DRAM_A8	DRAM_A8	DRAM_A7
DRAM_SDBA0	DRAM_SDBA0	DRAM_SDBA1
DRAM_SDBA1	DRAM_SDBA1	DRAM_SDBA0

45.4.5 LPDDR2 and DDR3 pin mux mapping

The following table shows the pin mux mapping between LPDDR2 and DDR3. The i.MX DDR I/O pads corresponds with the DDR3 standard.

- In DDR3, all DRAM_DATA, DRAM_SDQS, and DRAM_DQM data lines work with channel 0.
- In LPDDR2, DRAM_DDQS[3:0], DRAM_DATA[31:0] and DRAM_DQM[3:0] work with channel 0. DRAM_SDQS[7:4], DRAM_DATA[63:32], and DRAM_DQM[7:4] work with channel 1.

Table 45-7. LPDDR2 and DRAM pin mux mapping

DRAM I/O pad	LPDDR2 functionality
DRAM_ADDR00	LPDDR2_CA9_P0
DRAM_ADDR01	LPDDR2_CA7_P0
DRAM_ADDR02	LPDDR2_CA8_P0
DRAM_ADDR03	LPDDR2_CA1_P1
DRAM_ADDR04	LPDDR2_CA0_P1
DRAM_ADDR05	LPDDR2_CA3_P1
DRAM_ADDR06	LPDDR2_CA2_P1
DRAM_ADDR07	LPDDR2_CKE0_P1
DRAM-ADDR08	LPDDR2_CA4_P1
DRAM_ADDR09	LPDDR2_CKE1_P1
DRAM_ADDR10	LPDDR2_CA6_P0
DRAM_ADDR11	LPDDR2_CS1_B_P1
DRAM_ADDR12	LPDDR2_CS0_B_P1
DRAM_ADDR13	LPDDR2_CA1_P0
DRAM_ADDR14	LPDDR2_CA5_P1
DRAM_ADDR15	LPDDR2_CA7_P1
DRAM_CAS_B	LPDDR2_CA5_P0

Table continues on the next page...

Table 45-7. LPDDR2 and DRAM pin mux mapping (continued)

DRAM I/O pad	LPDDR2 functionality
DRAM_RAS_B	LPDDR2_CS1_B_P0
DRAM_WE_B	LPDDR2_CKE0_P0
DRAM_SDCKE0	LPDDR2_CA9_P1
DRAM_CKE1	LPDDR2_CA8_P1
DRAM_CS0_B	LPDDR2_CKE1_P0
DRAM_CS1_B	LPDDR2_CA2_P0
DRAM_ODT0	LPDDR2_CA3_P0
DRAM_ODT1	LPDDR2_CA0_P0
DRAM_SDCLK0_P	LPDDR2_CK_P0
DRAM_SDCLK1	LPDDR2_CK_P1
DRAM_BA0	LPDDR2_CS0_B_P0
DRAM_BA1	LPDDR2_CA4_P0
ddr3_ba2	LPDDR2_CA6_P1

45.4.6 Power Saving and Clock Frequency Change modes

45.4.6.1 Power saving general

MMDC supports multiple DDR power saving modes.

NOTE

At default, the power saving modes are disabled. These modes may dramatically decrease the power consumption of DDR memories.

1. Self-refresh entry to the entire DDR device (for both chip select 0 and 1) can be activated through two mechanisms:
   - LPMD (Low Power Mode)
     - Hardware handshaking (LPMD/LPACK) with the clock module in the system
     - Software handshaking by setting the field MAPSR[LPMD] and polling MAPSR[LPACK]
     - Automatic entry by configuring the amount of idle cycle for triggering self-refresh entry through MAPSR[PST] and by clearing MAPSR[PSD]
   - DVFS (Dynamic Voltage and Frequency Change)

- Hardware handshaking (DVFS/DVACK) with the clock module in the system
- Software handshaking by setting the field MAPSR[DVFS] and polling MAPSR[DVACK]

NOTE

If hardware or software requests for self-refresh entry were detected by the MMDC (even before the assertion of the LPACK), no write or read accesses will be acknowledged until the deassertion of those requests.

2. Automatic active/precharge power down entry to a specific chip select can be activated by configuring the ESDPDC register:
   - PWDT_0/PWDT_1 - define the number of idle cycles before entering power down, can be different value per chip select.
   - SLOW_PD - In case of DDR3 memory is configured to use slow precharge power down then this bit should be set as well.
   - BOTH_CS_PS - The MMDC can either set each chip select independently to power down, according to its idle state, or set both chip selects to power down only if both in idle state for the configured period.
   - Few parameters must be configured in addition:
     - Timing parameters at MDCFG0[tXP and tXPDLL].
     - ODT timing at MDOTC[tAOFPD, tAONPD, tANPD and tAXPD]

NOTE

It is possible to enter certain chip selects to low power consumption while the second chip select is activated.

3. Automatic precharge of all DDR banks to a specific chip select. Can be activated by configuring ESDPDC fields: PRCT_0 and PRCT_1. Each field determines a value loaded to a different chip select.

45.4.6.2 Self refresh and Frequency change entry/exit

As described in Power saving general, the MMDC supports two mechanisms that will cause the DDR device to enter self-refresh mode:

- LPMD (Low Power Mode) - For power saving purposes
- DVFS (Dynamic Voltage and Frequency Change) - For clock frequency changes

While the DDR device is in self-refresh mode, there is no need to provide periodic refresh commands.

The MMDC treats hardware/software handshaking of LPMD/DVFS in the same manner:

- Upon the assertion of LPMD/DVFS request, the following is done:
  - The MMDC blocks any further AXI accesses even before the acknowledge is asserted
  - Completes all opened AXI accesses
  - Closes (precharge) all banks in the appropriate timing
  - Drives self-refresh command by deasserting clock enable signal (DRAM_SDCKE is driven to "0") together with a refresh command. This occurs after satisfying tRP/tRPA from the Precharge All command.
  - Deasserts the clock (CK) that is driven to the DDR device
  - Asserts LPMD/DVFS acknowledge (LPACK/DVACK)
  - Allows deassertion of the operating clock of the MMDC (AXI clock)
- Upon the deassertion of LPMD/DVFS request, the following is done:
  - Operating clock of the MMDC must be turned on before LPMD/DVFS is deasserted
  - Starts driving the clock (CK) to the DDR device
  - After satisfying tCKSRX from clock renewal the clock enable signal (DRAM_SDCKE) is asserted
  - LPMD/DVFS acknowledge (LPACK/DVACK) is deasserted
  - After satisfying tXS from the assertion of DRAM_SDCKE, a refresh command is driven to the DDR device.
  - If ZQ calibration is enabled then tRFC is satisfied from the refresh command and a long ZQ command is driven.
  - tZQoper idle cycles are counted after the ZQ command.
  - After satisfying tDLLK from the assertion DRAM_SDCKE, the MMDC returns to normal operation.

The figure below shows the timing diagram of the hardware/software handshaking of LPMD/DVFS:

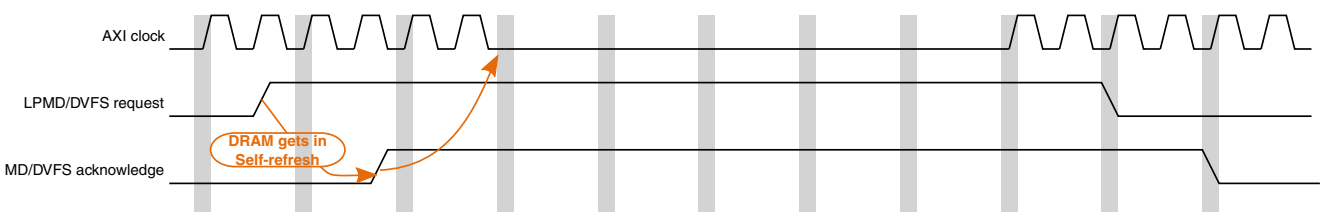


Figure 45-4. LPMD/DVFS Hardware/Software Handshaking

Note for self-refresh:

- As soon as LPMD or DVFS requests are detected by either hardware or software handshaking, the MMDC will deassert the AXI ARREADY/AWREADY signals immediately to block further requests from the system.
- In case of automatic self-refresh, the internal operating clock will be negated to save power.

45.4.7 Reset

45.4.7.1 Hard reset

When hard reset is asserted (aresetn is driven to "0") while warm reset is deasserted (warm_reset is driven to "0"), the entire MMDC will be initialized, including configuration/status registers and state machines.

In order to access the DDR device, the MMDC will then have to be reconfigured.

45.4.7.2 Warm reset

The MMDC supports warm reset signal. The warm reset signal must envelop the hard reset signal and then the MMDC will reset all the internal registers. The only registers that are not reset are those that are essential for returning it to normal operation without repeating the initialization sequence and without losing data stored in the memory (configuration/status registers won't be initialized).

For the successful operation of warm reset, the following steps must be performed:

- The MMDC must enter self-refresh mode. This can be achieved by either LPMD or DFVS requests
- Wait for LPMD or DVFS acknowledge
- Assert warm reset signal (i.e. drive warm_reset to "1")
- Assert hard reset signal (i.e. drive aresetn to "0")
- Deassert hard reset signal
- Deassert warm reset
- Get out of the LPMD/DVFS mode

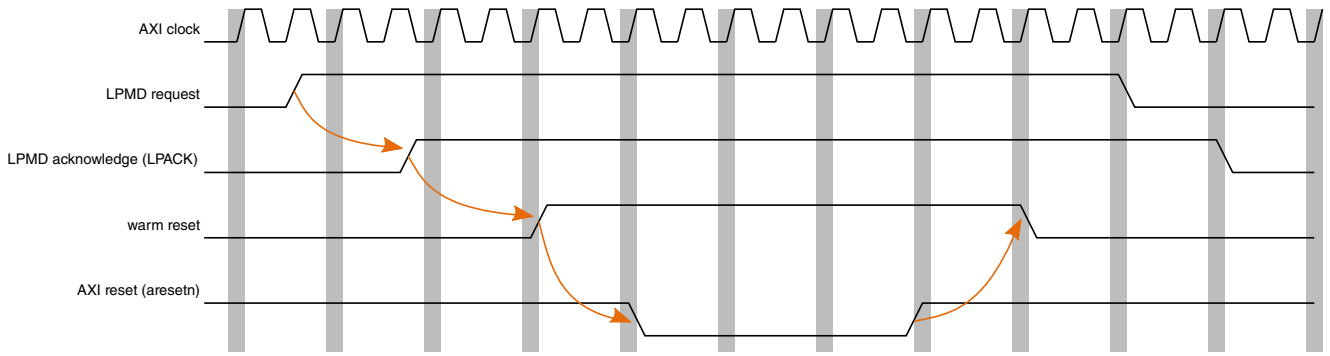


Figure 45-5. Warm Reset Diagram

45.4.7.3 Software reset

The MMDC supports software reset. When software reset is configured then the MMDC will reset all the internal registers except those that are essential for returning to normal operation without repeating the initialization sequence or without losing data stored in the memory (configuration/status registers won't be initialized).

The following steps should be performed for successful operation of software reset:

- The MMDC should enter self-refresh mode. This can be achieved by either LPMD or DFVS request.
- Wait for LPMD or DVFS acknowledge
- Assert software reset, by setting MDMISC[RST]
- Get out of the LPMD/DVFS mode

Normal operation can be resumed.

45.4.8 Refresh Scheme

The MMDC supports various automatic refresh options which can be configured via the MDREF register.

The periodic auto refresh can be triggered by the following clocks:

- 32 kHz clock
- 64 kHz clock
- MMDC operating clock

The refresh scheme of the MMDC is flexible and allows the system to configure the desired AXI accesses delay/latency in each refresh cycle.

The table below shows an example of four configurations of the refresh cycles that will be handled by the MMDC. Each configuration meets a refresh rate of 3.9µs (tREFI, refresh command every 3.9µs).

Table 45-8. MMDC Refresh Scheme

Option number	Description	REFR	REF_SEL	REF_CNT	DDR hang time
1	Issue 8 refresh commands every 31,250 ns	0x7 (8 refreshes)	0x0 (64 kHz)	not needed	tRFC x 8
2	Issue 4 refresh commands every 15,625ns	0x3 (4 refreshes)	0x1(32 kHz)	not needed	tRFC x 4
3	Issue 2 refresh commands every 7800ns	0x1(2 refreshes)	0x2 (REF_CNT)	7800/2.5 = 3120 (0xC30)	tRFC x 2
4	Issue 1 refresh command every 3900 ns	0x0 (1 refresh)	0x2 (REF_CNT)	3900/2.5 = 1560(0x618)	tRFC

45.4.9 Burst Length options towards DDR

The MMDC supports burst lengths which can be configured through MD+CTL[BL] as follows:

- In DDR3 mode, only burst length 8 can be used.
- In LPDDR2 mode, only burst length 4 can be used.

In DDR3 mode read/write accesses to the DDR are always 8 words (x16, x32, x64) and aligned in according to JEDEC standards.

In case of AXI INCREMENT, accesses that are not aligned the irrelevant data is masked in write accesses and ignored in read accesses. In case of AXI WRAP accesses, even if the access is not aligned, then the MMDC provides an internal optimization mechanism for better efficiency of the DDR data bus.

45.4.10 Exclusive accesses handling

The MMDC contains four exclusive monitors, each for dedicated ID as configured in MAEXIDR0 and MAEXIDR1.

- If legal read exclusive is received by the MMDC, the associated monitor is turned on.

- While the monitor is turned on upon legal write exclusive, the monitor will be turned off and the write will be completed successfully with EXOKAY.
- The following rules must be met for successful exclusive access:
  - Aligned access (the AXI address is aligned to the AXI size)
  - AXI single access (AXI burst length isn't greater than 1)
  - AXI size of up to 64 bits
  - AXI non-cachable access (i.e. ARCACHE[1]/AWCACHE[1] is equal "0" or ARCACHE[1]/AWCACHE[1] is equal "1" while ARCACHE[3:2]/ AWCACHE[3:2] are equal "00")
  - AXI ID that matches one of the four exclusive IDs

Exclusive read behavior (first bullet also correct for non-exclusive accesses):

- In case of security violation, the read is blocked and is not sent to DDR. There are two options for response:
  - If ARCR_SEC_ERR_EN (MAARCR[30]) is high, SLV error is issued towards the Master, otherwise OKAY response is sent to the Master.
- If AXI exclusive rules violation occurs (as described above), the read access is not blocked and is sent to DDR. The data will be fetched and be driven to the master, but the type of response may be unpredicted.
- If none of the above occurs, the read is sent to the DDR. The exclusive monitor will be turned on and the response is EXOKAY
- If additional legal AXI read exclusive is received with the same ID before the AXI exclusive write, the monitor will be updated with the latest attributes.

Exclusive write behavior (first bullet also correct for non-exclusive accesses):

- In case of security violation, the write is blocked and is not sent to DDR, but the monitor will be kept on. There are two options for response:
  - If ARCR_SEC_ERR_EN (MAARCR[30]) is high then SLV error is issued towards the Master, otherwise OKAY response is sent to the Master.
- In case of AXI exclusive rules violation (as described above), the write is blocked and is not sent to DDR. In that case the type of response may be unpredicted.
- In case the exclusive write access has different AXI attributes, but the same ID as the read exclusive access, the write is blocked and is not sent to DDR and the monitor will be turned off. There are two options for response:
  - If ARCR_EXC_ERR_EN (MAARCR[28]) is high then SLV error is issued towards the Master, otherwise OKAY response is sent to the Master.
- In case of regular (non exclusive) write access is received to the same address or overlapping addresses then the write will be sent to the DDR and the monitor will be turned off.
- In case of legal write exclusive access is received with the same attributes as the read exclusive access while the monitor is on (no write accesses occured to the same

address between the read exclusive and write exclusive), then the write is sent to DDR and the response is EXOKAY. But, if the legal write exclusive is received while the monitor is off, the write is blocked and there are two options for response.

- If ARCR_EXC_ERR_EN (MAARCR[28]) is high then SLV error is issued towards the Master, otherwise OKAY response is sent to the Master.

45.4.11 AXI Error Handling

The MMDC supports the AXI responses listed here.

- In case of AXI exclusive violation there are two options for response:
  - If MAARCR[28] is high then SLV Error is issued towards the Master, Otherwise OKAY response is sent to the Master

NOTE

In case of read error MMDC drives zeros on the read data bus

45.5 Performance

45.5.1 Arbitration and reordering mechanism

45.5.1.1 Arbitration General

The following specifies arbitration and reordering flow in MMDC towards the DDR.

- AXI read and write accesses are sampled in the associated queue.
- Read/write arbitration is handled to select the winning access.
- Winning access is sampled in the reordering queue
- Reordering mechanism is handled between valid requests that reside in the reordering queue to select the access that will be dispatched to the DDR.
  - The reordering is held in order to optimize the accesses and to maximize the utilization of the DDR bus
  - As soon as the reordered access is completed (indicated by end of response or data phase) then it is erased from the associated queue and the MMDC is ready to receive the next available access from the master

In general, the reordering/arbitration mechanism is based on dynamic priority mechanism, which compares dynamic priorities between valid entries in the reordering queue and issues the entry with highest dynamic priority towards the DDR Logic.

The selection of the winning access is based on two modes, which can be activated together, as following:

- Real time channel mode:
  - Accesses with QoS='f' (i.e. awqos[3:0]/arqos[3:0] = "f") will bypass all other requests towards the DDR
- Dynamic scoring mode:
  - The arbitration mechanism is based on dynamic priority. Relevant for the accesses with QoS smaller than 'f' or when real time channel mode is disabled.

NOTE

Due to re-ordering and optimization mechanism (per different AXI ID), the transactions towards the DDR may be driven in a different ID order they were received by the AXI master. In similar way, the write response, read response or read data may be driven to the AXI master in a different ID order.

45.5.1.2 Real time channel mode

When real time mode is enabled (i.e MAARCR[ARCR_RCH_EN] = "1") , all requests with QoS='f' (i.e. awqos[3:0]/aqqos[3:0] = "f") will bypass all other pending accesses towards the DDR. This mode is enabled by default.

45.5.1.3 Dynamic scoring mode (Arbitration Winning Conditions)

The arbitration between pending accesses in the MMDC is handled according to a dynamic priority of each access.

The dynamic priority (may be also called score) is calculated according to a sum of some factors (final_score[3:0]), where part of them may be updated dynamically. The following will specify each scoring factor:

- MAARCR[ARCR_PAG_HIT] (Page hit score) - A static score which is taken into account in case the pending access has a page hit
- MAARCR[ARCR_ACC_HIT] (Access hit score) - A static score, which is taken into account in case the current access type (read/write) is the same as the access that has been dispatched to the DDR previously

- MAARCR[ARCR_DYN_JMP] (Dynamic jump score) - A dynamic score which is given to any pending access in case it was not chosen in the arbitration. The dynamic jump counter is limited by maximum value which is set in MAARCR[ARCR_DYN_MAX] .
- QoS score which is indicated through a sideband 4bits AXI signals (awqos[3:0]/ aqqos[3:0]) and is driven by the AXI master per access

NOTE

In order to prevent an overflow in the total sum of scores, a clipping is held and selects the maximum score value of 'f' once a total scores sum is greater than 'f'.

The figure below shows the dynamic score calculations

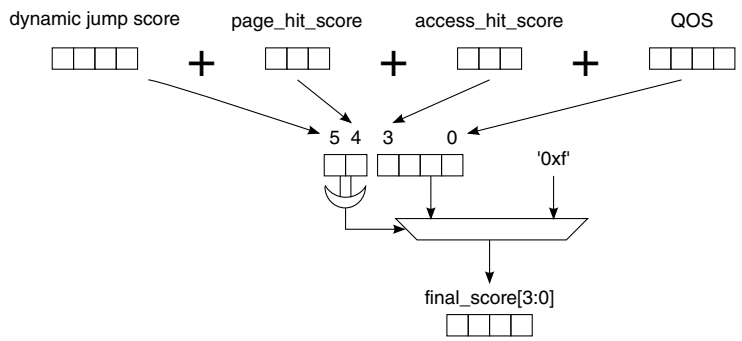


Figure 45-6. Dynamic score/priority calculation

45.5.1.4 Guarding (aging) mechanism

The guarding mechanism (may be also called aging) is used to prevent a starvation of accesses.

As soon as the dynamic jump score reaches its maximum value (MAARCR[ARCR_DYN_MAX]) then each time a pending request was not chosen in the arbitration, the "guarding" counter is incremented by 1. When the "guarding" counter reaches its predefined value, set in MAARCR[ARCR_GUARD], the associated request gets the highest priority and will be chosen in the next arbitration cycle towards the DDR unless a real time channel (i.e access with QoS ="f") is arrived.

Note: In case real time channel has arrived then the dynamic score of the non real time channels won't increment in order to prevent a case where the "guarding" counter of more than one access has reached its limit.

45.5.2 Prediction mechanism

When prediction mechanism is enabled (i.e by configuring MDMISC[MIF3_MODE]) then the MMDC predicts the chip-select, bank address and row address that is going to be issued towards the DDR before the access is physically dispatched towards DDR device.

That mechanism enables to prepare the DDR device with future accesses and improves the overall DDR performance.

This prediction mechanism operates in parallel to the reordering mechanism and may yield a prediction based on 3 levels of pending accesses:

1. Access in first stage of pipeline.
2. Valid access on AXI bus either read channel or write channel.
3. Valid access on special bus from arbitration - this access is chosen by the arbitration as the next miss access in its buffers

45.5.3 Special Optimization for accesses towards DDR3

In case an AXI read/write wrap non-aligned access is acknowledged in DDR3 mode with the same wrap boundary as the DDR wrap boundary then the MMDC will make an optimization and issue only one access towards the DDR, although all the accesses towards the DDR3 must be aligned.

For example: AXI write access with size of 128bits (awsize[2:0]=3'b100), length of 4 (awlen[3:0]=4'b0011) towards DDR3 x64 (burst length 8). In that case the AXI wrap boundary is 16Bx4=64B (0x40) and the DDR3 wrap boundary is 8Bx8=64B (0x40). If, for example, the AXI access is towards AXI address with suffix of 0x10 (non-aligned to 64B boundary) then the MMDC will get from the AXI master the data that is associated with addresses 0x10, 0x20, 0x30, 0x0. The MMDC will rearrange internally the data so it will match DDR3 alignment as following: 0x0, 0x10, 0x20, 0x30 and drive it in one access towards the DDR to address 0x0. The alternative was to issue two accesses towards the DDR with address 0x0 with different data masking

NOTE

In read wrap access the same optimization is handled, while as soon as the critical AXI word is fetched from the DDR then it is driven immediately to the AXI master without buffering. Based on the example above, the master expects to fetch first the data that is associated with address 0x10. Therefore the MMDC will issue read access from address 0x0 of the DDR and as soon as the data that is associated with address 0x10 is received then it

will be driven back immediately to the master even before fetching the data of the furhter addresses.

45.6 MMDC Debug

45.6.1 Hardware debug monitor

The MMDC has a hardware debugging mechanism that monitors each access that is driven to the MMDC from channel 0. Every time this mechanism is enabled (setting of MADPCR0[DBG_EN] to "1") then each access that will be dispatched to the DDR will be also observed in the I/O pads (i.e. over ipp_do_ddr_debug[50:0]). The content of this bus is described in the table below.

Table 45-9. Hardware monitor debugging

Signal Name	Number of Bits	Description
acc_addr	[31:0]	AXI ADDRESS of the selected access
acc_type	1	access type of the selected access. "0 indicates write. "1" indicates read.
acc_id	[15:0]	AXI transaction ID of the selected access
valid_strobe	1	indication for a valid request. This signal will be asserted for 1 clock cycle

The fields above are organized as following:

MMDC_DEBUG[50:0] = {1'b0,valid_strobe,acc_id,access_type,addr}

These signals are sent to IOMUX, in IOMUX user can configure it to be output from the chip for debug usage.

45.6.2 Step By Step (SBS) software monitor

The MMDC has a Step By Step (SBS) software debugging mechanism that monitors each access that is driven to the MMDC. Every time this mechanism is trigerred then one AXI access will be dispatched to the DDR and in parallel its attributes will be observed in a status register.

Once the "step by step" is enabled (i.e. MADPCR0[SBS_EN] is "1") then all accesses to the DDR device will be halted. This SBS feature is used during DDR SDCLK frequency scaling to prevent any accesses to the DDR device during the transition. Setting MADPCR0[SBS] to "1" will dispatch the access that is pending in the head of the MMDC queue (read or write). Upon every setting of MADPCR0[SBS]:

- The AXI attributes of the access will be sampled in the associated MASBS0 and MASBS1 fields
- MADPCR0[SBS] will be cleared automatically.

Setting again MADPCR0[SBS] to "1" will dispatch the next pending access in the MMDC queue.

NOTE

Setting the ARCR_ARB_REO_DIS bit in the MMDC0_MAARCR Register disables the SBS function

45.7 MMDC Profiling

The profiling mechanism provides the ability to calculate the DDR utilization together with read and write accesses statistics towards DDR per given period of time.

MMDC supports the following profiling counters:

- MADPSR0 (Total cycles count) - Indicates the total amount of cycles of the profiling period (up to 2^32 cycles)
- MADPSR1 (Busy cycles count) - Indicates the total busy cycles during the profiling period. Busy cycles are any MMDC clock cycles where the internal state machine is not idle. If any read or write requests are pending in the FIFOs, the MMDC is not idle.
- MADPSR2 (Total read accesses count) - Indicates the total read accesses towards MMDC during the profiling period
- MADPSR3 (Total write accesses count) - Indicates the total write accesses towards MMDC during the profiling period
- MADPSR4 (Total read bytes count) - Indicates total bytes that were read from MMDC during the profiling period
- MADPSR5 (Total write bytes count) - Indicates total bytes that were written to MMDC during the profiling period

All profiling items described above are disabled by default. The following describes how to control the profiling mechanism:

- MADPCR0[DBG_EN] enables profiling.

- MADPCR0[PRF_FRZ] stops/freezes the profiling for example in case user wishes to perform DDR profiling per specific task. In order to resume profiling then MADPCR0[PRF_FRZ] should be cleared.
- MADPCR0[DBG_RST] clears all profiling counters
- MADPCR0[CYC_OVF] indicates whether an overflow occured in the total cycles counter (i.e. total amount of cycles are greater than 2^32) . This field can only cleared by writing '0'.

Read/Write statistics can be collected per specific AXI ID (16bits). The following fields in MADPCR1 register determines which AXI-ID or AXI-ID's to monitor:

- PRF_AXI_ID defines which AXI IDs are taken for profiling. Default values is 16'h0.
- PRF_AXI_ID_MASK defines which bits from PRF_AXI_ID will be compared with AXI ID of read/write access. "1" means to monitor the associated bit and "0" means don't care. Default value is 16'h0000, meaning all IDs are not monitored

So the AXI-IDs to be monitored are calculated according to the following equation:

(AXI-ID & PRF_AXI_ID_MASK) Xnor (PRF_AXI_ID & PRF_AXI_ID_MASK)

For example if AXI ID's between A100 till A1FF are wished to be monitored then the following should be configured:

- PRF_AXI_ID = A100
- PRF_AXI_ID_MASK = FF00

Table 45-10. i.MX 6Solo/6DualLite AXI ID

Master	AXI ID at DDR controllers ('x' denotes ID from master)
ARM_S0	14'b000xxxxxxxx000
ARM_S1	14'b000xxxxxxxx001
IPU	14'b000000000xx100
GPU3D	14'b0000xxxx000010
GPU2D_a	14'b0000xxxx001010
VDOA	14'b000000xx010010
HDMI	14'b00000100011010
SDMA (Burst)	14'b00000101011010
SDMA (Periph)	14'b00000110011010
CAAM	14'b0xxxx000011010
USB	14'b000xx001011010
ENET	14'b00000010011010
HSI	14'b00000011011010
uSDHC1	14'b00000111011010
EPDC	14'b0000xxxx100010
GPU2D_b	14'b0000xxxx000011

Table continues on the next page...

Table 45-10. i.MX 6Solo/6DualLite AXI ID (continued)

Master	AXI ID at DDR controllers ('x' denotes ID from master)
VPU Prime	14'b0000xxxx001011
PCIe	14'b000xxxxx010011
DAP	14'b00000000011011
APBH DMA	14'b00000010011011
BCH40	14'b0xxxx001011011
MLB150	14'b00000011011011
uSDHC2	14'b00000100011011
uSDHC3	14'b00000101011011
uSDHC4	14'b00000110011011
Test port	14'b00000111011011
PXP	14'b0000xxxx100011

45.8 LPDDR2 Refresh Rate Update and Timing Derating

LPDDR2 devices may have a temperature sensor that is used to determine an appropriate refresh rate and whether AC timing derating is required. The status of the temperature sensor can be read through MRR command from LPDDR2 MR4 register.

The MMDC supports refresh update and timing derating mechanism on the fly. The following steps specify how to use that mechanism:

- Perform periodic polling on MR4 LPDDR2 register using MRR command
- Read MDMRR register and analyze the MR4 indication
- In case refresh rate update and/or AC timing derating is required then it is needed to update MDREF and/or MDMR4[tRCD_DE, tRC_DE, tRAS_DE, tRP_DE, tRRD_DE] parameters

NOTE

MDMR4[tRCD_DE, tRC_DE, tRAS_DE, tRP_DE, tRRD_DE] are referred to the associated values configured at MDCFG3LP[tRC_LP, tRP_LP, tRCD_LP], MDCFG1[tRAS], MDCFG2[tRRD]

- Assert MDMR4[UPDATE_DE_REQ]
- When the MMDC switch to the new values then an acknowledge will be indicated at MDMR4[UPDATE_DE_ACK]

The enabled power saving features may be interfering with the MR4 reads. When these features are enabled the MMDC will automatically enter self-refresh and hence prevent the MR4 reads. Hence the Power Saving and refresh functionality should be disabled when SW performs the MR4 reads.

To disable the “Power Down” mode, set bit 0 of MAPSR to 1’b.

To disable the “Self-Refresh Power Down Timer” mode, set bits [15:8] of MDPDC to 0x00.

Overall Sequence to Perform MR4 reads

1. Set MAPSR to 0x00010107
2. Set MDPDC to 0x00020076
3. MRR command to read MR4
4. Write MDPDC to 0x00025576 to re-enable this feature
5. Write MAPSR to 0x00010106 to re-enable this feature

45.9 DLL Switching

45.9.1 DLL Off mode

DLL Off mode is supported only in DDR3 and allows operation of the DDR in low frequency (i.e. below 125 MHz as defined in JEDEC standard).

For further details refer to DLL-off Mode chapter in the standard.

The following steps should be executed in order to switch from DLL on to DLL off mode:

- Assert CON_REQ signal and wait to CON_ACK assertion.
- Disable power down timers that can conflict with this sequence, such as: MAPSR[PSD], MDPDC[PWDT_1], MDPDC[PWDT_0], MDPDC[PRCT_0], MDPDC[PRCT_1].
- Execute precharge all banks command (via MDSCR).
- Execute MRW command to MR1 and disable RTT Nom (A9,A6,A2 =0) and DLL ON (A0 =1).
- Execute MRW command to MR2 in order to update CWL to 6.
- Execute MRW command to MR0 in order to update CL to 6.
- De-assert CON_REQ signal.
- Enter self refresh mode. For further information refer to Self refresh and Frequency change entry/exit.
- At self refresh entry acknowledge , change to the desired frequency.
- Exit self refresh mode.

- Assert CON_REQ and wait to CON_ACK assertion.
- Enable Pull Down resistors on DQS (through the I/O-MUX).
- Configure the MMDC register as following:
- Update tCWL =6 and tCL =6 to meet the values configured in the DDR device. (MDCGFG0, MDCFG1)
- Disable ODT resistor (i.e. set MPODTCTRL to "0").
- Disable DQS gating (i.e. set MPDGCTRL0[DG_DIS] to "1").
- Enable required power down timers that were disabled, such as: MAPSR[PSD], MDPDC[PWDT_1], MDPDC[PWDT_0], MDPDC[PRCT_0], MDPDC[PRCT_1].
- De-assert CON_REQ and wait for de-assertion of CON_ACK.

The following steps should be executed in order to switch from DLL off to DLL on:

- Execute precharge all banks command (via MDSCR).
- Enter self refresh mode. For further information refer to Self refresh and Frequency change entry/exit.
- At self refresh entry acknowledge, change to the desired frequency.
- Exit self refresh mode
- Assert CON_REQ and wait to CON_ACK assertion.
- Disable power down timers that can conflict with this sequence, such as: MAPSR[PSD], MDPDC[PWDT_1], MDPDC[PWDT_0], MDPDC[PRCT_0], MDPDC[PRCT_1].
- Execute MRW command to MR1 and enable RTT Nom (A9,A6,A2) and DLL ON (A0 =0).
- Execute MRW command to MR0 to reset the DLL (A8) and update CL value
- Execute MRW command to MR2 in order to update CWL value.
- Execute ZQ commnad.
- Reconfigure MMDC BLOCK
- Update tCWL and tCL to meet the values configured to the memory. (MDCGFG0, MDCFG1)
- Enable ODT resistor (i.e. MPODTCTRL register)
- Enable DQS gating (i.e. set MPDGCTRL0[DG_DIS] to "0").
- Disable Pull Down resistors on DQS (through the I/O-MUX).
- Enable required power down timers that were disabled, such as: MAPSR[PSD], MDPDC[PWDT_1], MDPDC[PWDT_0], MDPDC[PRCT_0], MDPDC[PRCT_1].
- De-assert CON_REQ and wait for de-assertion of CON_ACK.

45.10 ODT Configuration

The MMDC supports two DRAM_ODT signals (DRAM_ODT for each DRAM_CS) in DDR3 mode to allow the DDR device to turn on/off its termination resistors. The MMDC suggests various configurations for the assertion of the ODT signals as well as several timing related configurations.

The following specifies the options for configuring the assertion of DRAM_ODT signals:

- Assert DRAM_ODT signal for the non active DRAM_CS in write. For example : when this bit asserted - if writing to DRAM_CS0 the DRAM_ODT of DRAM_CS1 will be asserted. This is done by setting MPODTCTRL[0] to "1"
- Assert DRAM_ODT signal for the active DRAM_CS in write by by setting MPODTCTRL[1] to "1"
- Assert DRAM_ODT signal for the non active DRAM_CS in read by by setting MPODTCTRL[2] to "1"
- Assert DRAM_ODT signal for the active DRAM_CS in read by by setting MPODTCTRL[3] to "1"

MDOTC register controls the timing for the DRAM_ODT signals assertion.

NOTE

tODTLon determines the delay between DRAM_ODT signal and the associated RTT, where according to JEDEC standard it equals WL(write latency) - 2. Therefore, the value configured to MDOTC[tODTLon] field should correspond with the value configured to MDCGFG1[tCWL].

In precharge power down mode, when all banks are closed, the assertion of ODT corresponds with tAOFPD and tAONPD which are configured in MDOTC register.

The figure below shows timing diagram of DRAM_ODT and RTT signals while MPODTCTRL[0] is set to "1" (i.e. assertion of DRAM_ODT to the non active DRAM_CS in write access command) and MDOTC[tODTLon] is set to 4.

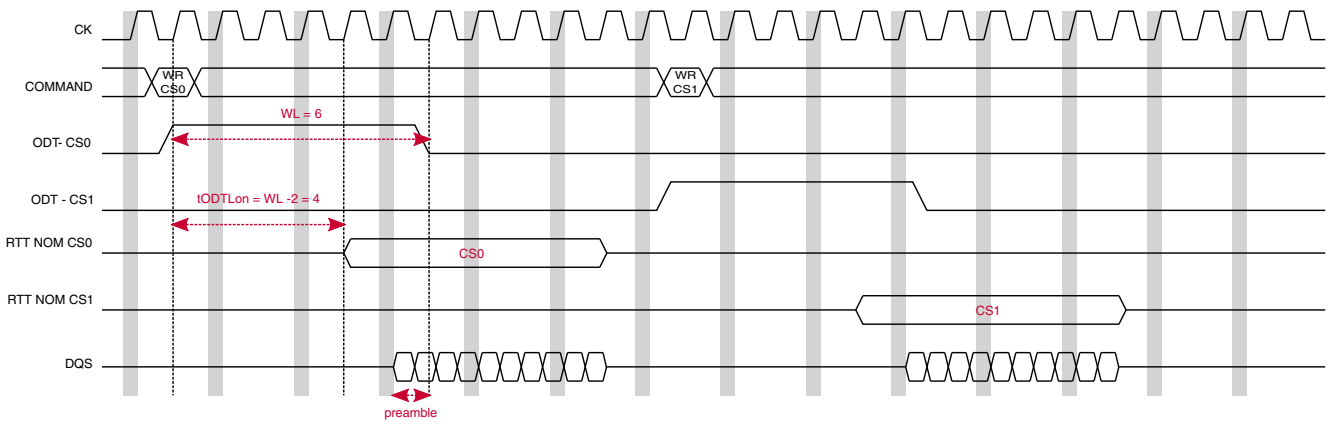


Figure 45-7. ODT - Timing Diagram DDR3 WL=6, BL=8

45.11 Calibration Process

The MMDC offers various calibration processes that are used to obtain better timing accuracy, board skew compensation and I/O pad driving strength adjustment. Each calibration process can be performed either automatically (hardware) or manually (software), though the manual method is typically reserved for debugging purposes. The following calibration processes are supported:

NOTE

Power saving features must be disabled before beginning the write leveling, DQS gating, and read/write calibration processes. ZQ calibration is allowed with power saving features enabled.

- ZQ calibration for external DDR device (in DDR3 through ZQ calibration command and in LPDDR2/LPDDR3 through MRW command)
  - Can be handled automatically for ZQ Short (periodically) and ZQ Long (at exit from self-refresh)
  - Can be handled manually at ZQ INIT
- ZQ calibration for i.MX DDR I/O pads for calibrating the DDR driving strength
  - The sequence can be handled automatically by hardware
  - The sequence can be handled step by step manually by software
- Read DQS gating calibration for DDR3 only. Adjustment of DQS gate with read preamble window. For further information refer to Read DQS Gating Calibration
- Read DQS gating calibration not supported for LPDDR2/LPDDR3
- Read data calibration. Adjustment of read DQS with read data byte. For further information refer to Read Calibration

- Write data calibration. Adjustment of write DQS with write data byte. For further information refer to Write Calibration
- Write leveling calibration. Adjustment of write DQS with CK (DDR differential clock). For further information refer to Write leveling Calibration
- Read fine tuning. Adjustment of up to 7 delay-line units for each read data bit.
- Write fine tuning. Adjustment of up to 3 delay-line units for each read data bit.

NOTE

Before starting any calibration process that involves the DDR3 device MPR mode or write leveling calibration, the following should be done:

- Disable the periodic refresh scheme (i.e. setting MDREF[REF_SEL] = "11") and then issue manual refresh command burst by configuring MDSCR[CMD]= 0x2. At the end of the calibration it is needed to enable the periodic refresh scheme.
- Disable the automatic power saving mode (i.e set MAPSR[PSD] = "1").

45.11.1 Delay-line

Each of the calibration processes controls several delay-lines for aligning data and strobes.

By default the delay-line is configured to generate 1/4 clock cycle of delay.

Moreover, when the operating clock is at the maximum allowed frequency, as appeared in the features list, then the delay-line is capable to issue a configurable delay of up to 1/2 clock cycle.

NOTE

At the beginning of the calibration process the initial value of the delay-line must be a valid value (i.e. the strobes must be somewhere among the associated data window) though it might not be the optimal value. The delay-line calibration should be done after Read DQS gating and write-leveling calibrations.

In order to generate an adequate delay during normal operation of the MMDC the delay-line is going through an automatic measurement process during the refresh period of the DDR device

45.11.2 ZQ calibration

The MMDC supports ZQ calibration process to calibrate the driving strength of the DDR I/O pads as well as driving ZQ commands to calibrate the external DDR device driving strength.

The first ZQ calibration (after booting the processor) is performed prior to turning on the MMDC. Subsequent ZQ calibrations may be executed in parallel to the DDR ZQ calibration. The MMDC supports 2 types of ZQ calibration commands: short and long.

The ZQ long calibration is executed during power up sequence, when exiting self-refresh mode or when exiting slow precharge power down (DLL lock can be done in parallel). The ZQ short calibration is executed periodically according to a configurable timer defined by MPZQHWCTRL[ZQ_HW_PER].

The field MPZQHWCTRL[ZQ_MODE] determines whether the MMDC will execute ZQ calibration to DDR I/O pads and/or issue ZQ short/long command to the DDR device.

The MMDC supports both automatic (hardware) and manual (software) ZQ calibration process for the DDR I/O pads.

It is possible to perform automatic (hardware) ZQ calibration only once (i.e. non-periodical) by asserting MPZQHWCTRL[ZQ_HW_FOR].

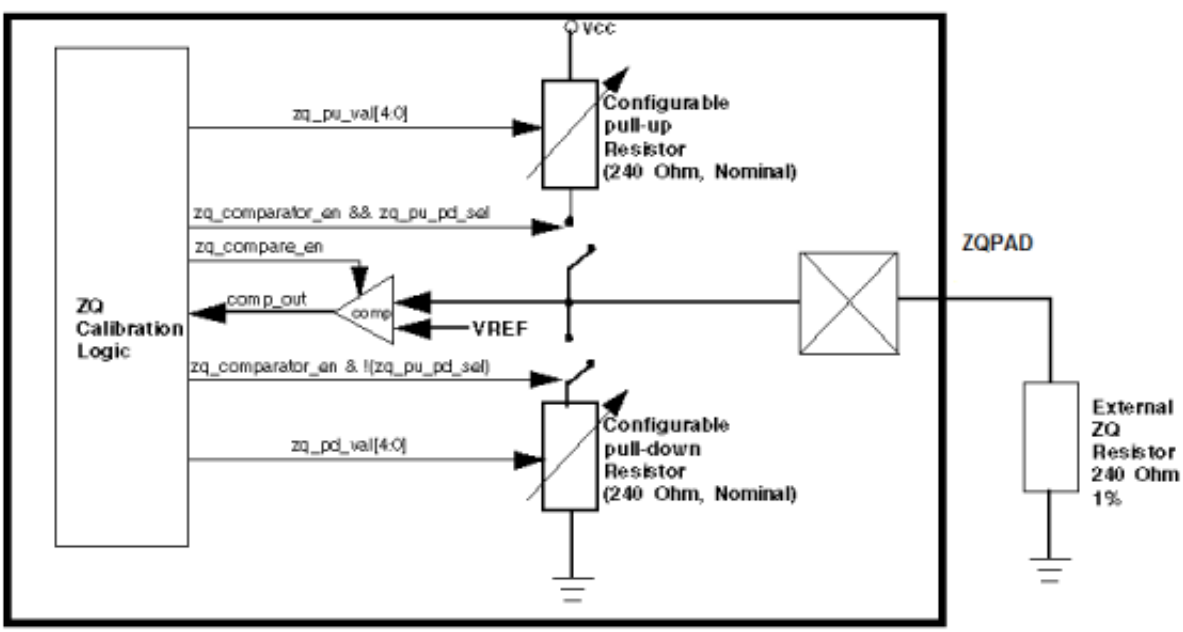


Figure 45-8. MMDC ZQ IF with PAD

45.11.2.1 ZQ automatic (hardware) calibration process

The ZQ automatic calibration occurs over multiple steps isolated by pull-up resistor calibration and by pull-down resistor calibration as follows.

45.11.2.1.1 ZQ automatic Pull-up calibration

The MMDC automatically performs a handshaking mechanism with the ZQ calibration pad as follows:

1. The MMDC drives zq_comparator_en to "1"
2. The MMDC waits few cycles according to MPZQHWCTRL[ZQ_EARLY_COMPARATOR_EN_TIMER]
3. The MMDC drives zq_pu_pd_sel to "1" for indication of pull-up calibration and drives zq_pu_val[4:0] = 5'b00000
4. MMDC drives zq_pu_val[4] to "1"
5. MMDC asserts zq_compare_en
6. MMDC waits few cycles according to MPZQSWCTRL[ZQ_CMP_OUT_SMP] before sampling the comparator output (i.e zq_comp_out). If zq_comp_out is "1" then it means that the output voltage is greater than Vdd/2 (i.e. internal resistor is less than 240 ohm) and drives bit zq_pu_val[4] to "1" else it drives zq_pu_val[4] to "0"
7. MMDC deasserts zq_compare_en
8. MMDC repeats steps 4-7 for zq_pu_val bits 3 to 0
9. MMDC drives ZQ calibration result to MPZQHWCTRL[ZQ_HW_PU_RES]
10. MMDC advances to pull-down calibration

45.11.2.1.2 ZQ automatic Pull-down calibration

1. The MMDC drives zq_pu_pd_sel to "0" for indication of pull-down calibration and drives zq_pd_val[4:0] = 5'b00000
2. MMDC drives zq_pd_val[4] to "1"
3. MMDC asserts zq_compare_en
4. MMDC waits few cycles according to MPZQSWCTRL[ZQ_CMP_OUT_SMP] before sampling the comparator output (i.e zq_comp_out). If zq_comp_out is "1" then it means that the output voltage is greater than Vdd/2 (i.e. internal resistor is less than 240 ohm) and drives bit zq_pd_val[4] to "0" else it drives zq_pd_val[4] to "1"
5. MMDC deasserts zq_compare_en
6. MMDC repeats steps 2-5 for zq_pd_val bits 3 to 0
7. MMDC drives ZQ calibration result to MPZQHWCTRL[ZQ_HW_PD_RES]
8. MMDC deassert zq_comparator_en to indicate the completion of the ZQ calibration

45.11.2.2 ZQ software calibration process

The ZQ calibration can be done also in software. However since software ZQ calibration is much slower than hardware calibration it should be used mainly for debugging.

Software should configure the ZQ calibration parameters (Pull-up or Pull-down and their value) then assert the MPZQSWCTRL[ZQ_SW_FOR] bit. Then software should wait till ZQ_SW_FOR is de-asserted and use ZQ_SW_RES status bit in order to calculate the next ZQ calibration parameters.

45.11.2.3 ZQ calibration commands

Before the MMDC can issue a ZQCL/ZQCS command to the memory it should precharge all memory banks and wait tRP period. A single ZQ command can be issued to all devices as long as the devices don't share the same ZQ resistor.

When the MMDC issues the ZQ command it should also drive A10 (long or short command) and CS (0, 1 or both).

The MMDC must keep the memory lines quiet (except for CK) for the ZQ calibration time as defined in the Jedec (512 cycles for ZQCL after reset, 256 for other ZQCL and 64 for ZQCS).

45.11.3 Read DQS Gating Calibration

The read DQS gating calibration is used to adjust the read DQS gating with the middle of the read DQS preamble. The DQS gating includes a delay of up to 7 cycles (The delay is chosen according to two fields MPDGCTRLn[DG_HC_DELn] and MPDGCTRLn[DG_DL_ABS_OFFSETn]

Each DQS has its own delay-line. The DQS gating process can be done for all DQS in parallel.

NOTE

In LPDDR2/LPDDR3 mode hardware Read DQS gating should be disabled and Pull-up/pull-down resistors on DQS/DQSn should be enabled while ODT resistors must be disconnected.

NOTE

In DDR3_x64 mode activation of the calibration is done by setting MPDGCTRL0[HW_DG_EN]

45.11.3.1 Hardware DQS Gating Calibration

- There are two modes of operations:
  - Calibration with the MPR (Multi Purpose Register)
  - Calibration with MMDC pre-defined values

45.11.3.1.1 Hardware DQS Calibration with MPR

The following steps should be executed:

1. Precharge all active banks (Can be done through MDSCR) as requried by the standard.
2. Enter the DDR device into MPR mode through MRS commands
3. Configure the MMDC to work with MPR mode by asserting MPPDCMPR2[MPR_CMP]
4. Make sure that the initial value that is configured in the read delay line absolute offset of each byte (i.e. MPRDDLCTL[RD_DL_ABS_OFFSETn]) will place the read DQS somewhere inside the read DQ window
5. Start the calibration process by asserting MPDGCTRL0[HW_DG_EN]

45.11.3.1.2 Hardware DQS Calibration with pre-defined value

In case pre-defined mode is used, (i.e. MPPDCMPR2[MPR_CMP]) is cleared, then the following steps should be executed:

1. Precharge all active banks (Can be done through MDSCR) as required by the standard.
2. Configure the pre-defined value, which reflects the value that will be written and compared through the read calibration, to MPPDCMPR1[PDV1, PDV2]
3. Issue write access to the external DDR device by setting MPSWDAR0[SW_DUMMY_WR] = 1 (MMDC will generate internally write access without intervention of the system towards bank 0, row 0, column 0)
4. Make sure that the initial value that is configured in the read delay line absolute offset of each byte (i.e. MPRDDLCTL[RD_DL_ABS_OFFSETn]) will place the read DQS somewhere inside the read DQ window
5. Start the calibration process by asserting MPDGCTRL0[HW_DG_EN]

   The following steps will be executed automatically by the MMDC for both modes (MPR and Pre-defined value):

6. MMDC waits till the read DQS delay-line is updated with the absolute delay value for all bytes at MPDGCTRLn[DG_HC_DELn] and MPDGCTRLn[DG_DL_ABS_OFFSETn] and also satisfying the Tmod + 4 requirement

7. MMDC drives read command to the external DDR devices and waits 16 or 32 cycles (according to MPDGCTRL0[DG_CMP_CYC] assuming that the data has arrived from the DDR device.
8. MMDC compares the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8). If the comparison fails then it indicates that the read DQS gating is asserted in illegal time point. If the comparison passes then MMDC advances to step 14
9. MMDC resets the read FIFO (to the inverted pre-defined/MPR value) and it's pointers by setting MPDGCTRL[RST_RD_FIFO] = 1
10. MMDC increments the read DQS gating delay of each byte by half cycle (i.e. MPDGCTRLn[DG_HC_DELn] + 1)
11. MMDC drives read command to the external DDR devices and waits 16 or 32 cycles (according to MPDGCTRL0[DG_CMP_CYC] assuming that the data has arrived from the DDR device.
12. MMDC compares the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8).
13. If the comparison fails then it indicates that the read DQS gating is asserted in illegal time point and it is needed to repeat steps 9-12. If the comparison passes then MMDC stores the value of the temporary low boundary and advances to next step
14. MMDC increments the read DQS gating delay-line of each byte by half cycle (i.e. MPDGCTRLn[DG_HC_DELn] + 1) and issue measurement process of the read DQS gating delay-line to update itself with the new value
15. MMDC drives read command to the external DDR devices and waits 16 or 32 cycles (according to MPDGCTRL0[DG_CMP_CYC] assuming that the data has arrived from the DDR device.
16. MMDC compares the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8)
17. If the comparison passes then it indicates that the read DQS gating is asserted inside the read preamble window and it is needed to repeat steps 14-16. If the comparison fails then MMDC stores the value of the temporary upper boundary and starts searching the adequate low and high boundaries
18. MMDC returns to the temporary low boundary minus half cycle and issue measurement process of the read DQS gating delay-line to update itself with the new value
19. MMDC drives read command to the external DDR devices and waits 16 or 32 cycles (according to MPDGCTRL0[DG_CMP_CYC] assuming that the data has arrived from the DDR device.
20. MMDC compares the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8)

21. If the comparison fails then it indicates that the read DQS gating is asserted in illegal time point and it is needed to repeat steps 22-23. If the comparison passes then MMDC stores the value of the adequate low boundary and advances to step 24
22. MMDC resets the read FIFO (to the inverted pre-defined/MPR value) and it's pointers by setting MPDGCTRL[RST_RD_FIFO] = 1
23. MMDC increments the read DQS gating delay of each byte by 1 (i.e. MPDGCTRLn[DG_DL_ABS_OFFSETn] + 1) and issue measurement process of the read DQS gating delay-line to update itself with the new value and advances to step 19
24. MMDC returns to the temporary upper boundary minus half cycle and issue measurement process of the read DQS gating delay-line to update itself with the new value
25. MMDC drives read command to the external DDR devices and waits 16 or 32 cycles (according to MPDGCTRL0[DG_CMP_CYC] assuming that the data has arrived from the DDR device.
26. MMDC compares the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8)
27. If the comparison passes then it indicates that the read DQS gating is asserted inside the read preamble window and it is needed to repeat steps 28-29. If the comparison fails then MMDC stores the value minus 1 of the adequate upper boundary and advances to step 30
28. MMDC resets the read FIFO (to the inverted pre-defined/MPR value) and it's pointers by setting MPDGCTRL[RST_RD_FIFO] = 1
29. MMDC increments the read DQS gating delay of each byte by 1 (i.e. MPDGCTRLn[DG_DL_ABS_OFFSETn] + 1) and issue measurement process of the read DQS gating delay-line to update itself with the new value and advances to step 25
30. After the MMDC finds the window boundary (lower and upper) of each read data byte then it stores the average between lower and upper boundaries at the associated MPDGCTRLn[DG_DL_ABS_OFFSETn] and issue measurement process of the read DQS delay-line to update itself with the new value.
31. MMDC indicates that the read DQS gating calibration had finished by setting MPDGCTRL0[HW_DG_EN] = 0
32. Exit the DDR device from MPR mode through MRS command
33. Read the upper boundary that was found: MPDGHWSTn[HW_DG_UPn]. This field is 11 bits, 7 LSB bits correspond to MPDGCTRLn[DG_DL_ABS_OFFSETn] upper limit value and 4 MSB bits correspond to MPDGCTRLn[DG_HC_DELn] upper limit value.
34. Set MPDGHWSTn[HW_DG_UPn][6:0] to MPDGCTRLn[DG_DL_ABS_OFFSETn].

35. Set (MPDGHWSTn[HW_DG_UPn][10:7] - 1) to MPDGCTRLn[DG_HC_DELn]. (We set the DQS gating value to be the upper limit value minus 1 half cycle)

45.11.3.2 SW read DQS gating Calibration

There are two modes of operations:

- Calibration with the MPR (Multi Purpose Register)
- Calibration with MMDC pre-defined values

45.11.3.2.1 SW read Calibration with MPR

Execute the following steps:

1. Precharge all active banks (Can be done through MDSCR) as required.
2. Enter the DDR device into MPR mode through MRS commands
3. Configure the MMDC to work with MPR mode by asserting MPPDCMPR2[MPR_CMP]
4. Ensure that the initial value that is configured in the read delay line absolute offset of each byte (i.e. MPRDDLCTL[RD_DL_ABS_OFFSET#]) will place the read DQS somewhere inside the read DQ window

45.11.3.2.2 SW read Calibration with pre-defined value

In case pre-defined mode is used, (i.e. MPPDCMPR2[MPR_CMP]) is cleared, then the following steps should be executed:

1. Precharge all active banks (Can be done through MDSCR) as required by the standard.
2. Configure the pre-defined value, which reflects the value that will be written and compared through the read calibration, to MPPDCMPR1[PDV1, PDV2]
3. Issue write access (with any legal DDR address) to external DDR device
4. Ensure that the initial value that is configured in the read delay line absolute offset of each byte (i.e. MPRDDLCTL[RD_DL_ABS_OFFSETn]) will place the read DQS somewhere inside the read DQ window

The following steps should be executed automatically by the MMDC for both modes (MPR and Pre-defined value):

1. Configure the read DQS delay-line to issue zero delay by setting MPDGCTRLn[DG_DL_ABS_OFFSETn] = 0 and MPDGCTRLn[DG_HC_DELn] = 0
2. Force the delay line to measure itself and to issue the requested read delay by configuring MPMUR[FRC_MSR] = 1

3. Wait 16 DDR cycles till the read DQS delay-line is updated with the absolute delay value for all bytes
4. Issue read command (with the legal DDR address chosen in step 3) from the external DDR device
5. Waits 16 or 32 cycles (according to MPDGCTRL0[DG_CMP_CYC]) assuming that the data has arrived from the DDR device.
6. Compare the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8). If the comparison fails then it indicates that the read DQS gating is asserted in illegal time point. If the comparison passes then advance to step 11.
7. MMDC resets the read FIFO (to the inverted pre-defined/MPR value) and it's pointers by setting MPDGCTRL[RST_RD_FIFO] = 1
8. Increment the read DQS gating delay of each byte by half cycle (i.e. MPDGCTRLn[DG_HC_DELn] + 1)
9. MMDC drives read command to the external DDR devices and waits 16 or 32 cycles (according to MPDGCTRL0[DG_CMP_CYC]) assuming that the data has arrived from the DDR device.
10. Compare the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8). If the comparison fails then it indicates that the read DQS gating is asserted in illegal time point and it is needed to repeat steps 7-10. If the comparison passes then advance to step 11.
11. Store the temporary lower boundary and start searching the temporary upper boundary
12. MMDC resets the read FIFO (to the inverted pre-defined/MPR value) and it's pointers by setting MPDGCTRL[RST_RD_FIFO] = 1
13. Increment the read DQS gating delay of each byte by half cycle (i.e. MPDGCTRLn[DG_HC_DELn] + 1)
14. MMDC drives read command to the external DDR devices and waits 16 or 32 cycles (according to MPDGCTRL0[DG_CMP_CYC] assuming that the data has arrived from the DDR device.
15. Compare the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8).
16. If the comparison passes then it indicates that the read DQS gating is asserted inside the read preamble window and it is needed to repeat steps 12-15. If the comparison fails then it is needed to store the value of the temporary upper boundary and starts searching the adequate low and high boundaries
17. Load the temporary low boundary minus half cycle into the associated MPDGCTRLn[DG_HC_DELn]
18. Reset the read FIFO (to the inverted pre-defined/MPR value) and its pointers by setting MPDGCTRL[RST_RD_FIFO] = 1

19. Increment the read DQS gating delay of each byte by 1 (i.e. MPDGCTRLn[DG_DL_ABS_OFFSETn] + 1) and force the delay line to measure itself and to issue the requested read DQS delay by configuring MPMUR0[FRC_MSR] = 1
20. Issue read command to the external DDR devices and waits 16 or 32 cycles (according to MPDGCTRL0[DG_CMP_CYC]) assuming that the data has arrived from the DDR device.
21. Compare the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8)
22. If the comparison fails then it indicates that the read DQS gating is asserted in illegal time point and it is needed to repeat steps 18-22. If the comparisons passes then advance to the next step.
23. Store the adequate lower boundary
24. Load the temporary upper boundary minus half cycle into the associated MPDGCTRLn[DG_HC_DELn]
25. Reset the read FIFO (to the inverted pre-defined/MPR value) and it's pointers by setting MPDGCTRL[RST_RD_FIFO] = 1
26. Increment the read DQS gating delay of each byte by 1 (i.e. MPDGCTRLn[DG_DL_ABS_OFFSETn] + 1) and force the delay line to measure itself and to issue the requested read DQS delay by configuring MPMUR[FRC_MSR] = 1
27. Issue read command to the external DDR devices and waits 16 or 32 cycles (according to MPDGCTRL0[DG_CMP_CYC] assuming that the data has arrived from the DDR device.
28. Compare the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8)
29. If the comparison passes then it is needed to repeat steps 25-28. If the comparisons fails then advance to the next step.
30. Reset the read FIFO (to the inverted pre-defined/MPR value) and it's pointers by setting MPDGCTRL[RST_RD_FIFO] = 1
31. Store the adequate upper boundary.
32. Keep the MPDGCTRLn[DG_DL_ABS_OFFSETn] value of the upper limit.
33. Set MPDGCTRLn[DG_HC_DELn] = (MPDGCTRLn[DG_HC_DELn] - 1). (We set the DQS gating value to be the upper limit value minus 1 half cycle)
34. Issue the requested read DQS delay by configuring MPMUR[FRC_MSR] = 1
35. Exit the DDR device from MPR mode through MRS command

45.11.4 Read Calibration

The read calibration is used to adjust the read DQS with read data byte. It is assumed that the read DQS gating calibration process is completed prior to the read calibration.

NOTE

In DDR3 mode, the activation of the calibration is done by setting MPRDDLHWCTL[HW_RD_DL_EN].

NOTE

In LP2_x16, LP2_x32 the activation of the calibration of each channel is done by setting MPRDDLHWCTL[HW_RD_DL_EN] and MPRDDLHWCTL[HW_RD_DL_EN].

45.11.4.1 Hardware (automatic) Read Calibration

There are two modes of operations:

- Calibration with the MPR (Multi Purpose Register)/DQ calibration(LPDDR2)
- Calibration with MMDC pre-defined values

45.11.4.1.1 Hardware (automatic) Calibration with MPR/DQ Calibration

Execute the following steps:

1. Precharge all active banks (can be done through MDSCR) as required.
2. Enter the DDR device into MPR/DQ calibration mode through MRS/MRW commands.
3. Configure the MMDC to work with MPR/DQ calibration mode by asserting MPPDCMPR2[MPR_CMP].
4. Make sure that the initial value that is configured in the read delay line absolute offset of each byte (MPRDDLCTL[RD_DL_ABS_OFFSET#]) will place the read DQS somewhere inside the read DQ window.
5. Start the calibration process by asserting MPRDDLHWCTL[HW_RD_DL_EN].

45.11.4.1.2 Hardware (automatic) Calibration with pre-defined value

In case pre-defined mode is used, i.e. MPPDCMPR2[MPR_CMP] is cleared, then the following steps should be executed:

1. Precharge all active banks (Can be done through MDSCR) as required.
2. Configure the pre-defined value, which reflects the value that will be written and compared through the read calibration, to MPPDCMPR1[PDV1, PDV2]
3. Issue write access to the external DDR device by setting MPSWDAR0[SW_DUMMY_WR] = 1 (MMDC will generate internally write access without intervention of the system towards bank 0, row 0, column 0)

4. Make sure that the initial value that is configured in the read delay line absolute offset of each byte (i.e. MPRDDLCTL[RD_DL_ABS_OFFSETn]) will place the read DQS somewhere inside the read DQ window
5. Start the calibration process by asserting MPRDDLHWCTL[HW_RD_DL_EN]

The following steps will be executed automatically by the MMDC for both modes (MPR and Pre-defined value):

1. MMDC waits till the read delay-line is updated with the absolute delay value for all bytes at MPRDDLCTL[RD_DL_ABS_OFFSETn] and also satisfying the Tmod + 4 requirement
2. MMDC drives read command to the external DDR devices and waits 16 or 32 cycles (according to MPRDDLHWCTL[HW_RD_DL_CMP_CYC]) assuming that the data has arrived from the DDR device.
3. MMDC compares the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8). If the comparison fails then it indicates that the initial read DQS isn't inside the read DQ window and the MMDC generates an error for the associated byte at MPRDDLHWCTL[HW_RD_DL_ERRn] . If the comparison passes then MMDC advances to next step.
4. MMDC resets the rd fifo (to the inverted pre-defined/MPR value) and it's pointers by setting MPDGCTRL[RST_RD_FIFO] = 1
5. MMDC decrements the read delay line absolute offset of each byte by 1 (i.e. MPRDDLCTL[RD_DL_ABS_OFFSETn]) and issue measurement process of the read delay-line to update itself with the new value.
6. MMDC drives read command to the DDR external devices and waits 16 or 32 cycles (according to MPRDDLHWCTL[HW_RD_DL_CMP_CYC]) assuming that the data has arrived from the DDR device
7. MMDC compares the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8). If the comparison fails then it stores the low read boundary of the associated byte for each byte at MPRDDLHWST0/1[HW_RD_DL_LOWn] . If the comparison passes then MMDC repeats steps 4-6. If all read data comparisons fail then the MMDC advances to the next step
8. The MMDC start seeking the upper boundary and sets the read delay line absolute offset of each byte to the initial value + 1 as determined at step 4 and issue measurement process of the read delay-line to update itself with the new value
9. MMDC resets the rd fifo (to the inverted pre-defined value) and it's pointers by setting MPDGCTRL[RST_RD_FIFO] = 1
10. MMDC drives read command to the DDR external devices and waits 16 or 32 cycles (according to MPRDDLHWCTL[HW_RD_DL_CMP_CYC]) assuming that the data has arrived from the DDR device

11. MMDC compares the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8). If the comparison fails then it stores the upper read boundary of the associated byte for each byte at MPRDDLHWST0/1[HW_RD_DL_UPn] . If the comparison passes then MMDC increments the read delay line absolute offset of each byte by 1 (i.e. MPRDDLCTL[RD_DL_ABS_OFFSETn]) and issue measurement process of the read delay-line to update itself with the new value.
12. If all read data comparisons fail then the MMDC advances to the next step. otherwise, MMDC repeats steps 9-11.
13. After the MMDC finds the window boundary (lower and upper) of each read data byte then it stores the average between lower and upper boundaries at the associated MPRDDLCTL[RD_DL_ABS_OFFSETn] and issue measurement process of the read delay-line to update itself with the new value.
14. MMDC indicates that the read data calibration had finished by setting MPRDDLHWCTL[HW_RD_DL_EN] = 0
15. Exit the DDR device from MPR/DQ calibration mode through MRS/MRW commands

45.11.4.2 SW Read Calibration

There are two modes of operations:

- Calibration with the MPR (Multi Purpose Register)/DQ calibration (LPDDR2)
- Calibration with MMDC pre-defined values

45.11.4.2.1 Calibration with MPR/DQ calibration

Execute the following steps:

1. Precharge all active banks (Can be done through MDSCR) as required.
2. Enter the DDR device into MPR/DQ calibration mode through MRS/MRW commands
3. Configure the MMDC to work with MPR/DQ calibration mode by asserting MPPDCMPR2[MPR_CMP]
4. Make sure that the initial value that is configured in the read delay line absolute offset of each byte (i.e. MPRDDLCTL[RD_DL_ABS_OFFSETn]) will place the read DQS somewhere inside the read DQ window

45.11.4.2.2 Calibration with pre-defined value

In case pre-defined mode is used, i.e. MPPDCMPR2[MPR_CMP] is cleared, then the following steps should be executed:

1. Precharge all active banks (Can be done through MDSCR) as requried by the standard.
2. Configure the pre-defined value, which reflects the value that will be written and compared through the read calibration, to MPPDCMPR1[PDV1, PDV2]
3. Issue write access (with any legal DDR address) to external DDR device.
4. Make sure that the initial value that is configured in the read delay line absolute offset of each byte (i.e. MPRDDLCTL[RD_DL_ABS_OFFSETn]) will place the read DQS somewhere inside the read DQ window

The following steps will be executed manually by SW for both modes (MPR/DQ calibration and Pre-defined value):

1. Force the delay line to measure itself and to issue the requested read delay by configuring MPMUR[FRC_MSR] = 1
2. Wait 16 DDR cycles till the read delay-line is updated with the absolute delay value for all bytes
3. Issue read command (with any legal DDR address) from the external DDR device
4. Compare the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8). If the comparison fails then it indicates that the initial read DQS isn't inside the read DQ window. If the comparison passes then advance to next step.
5. Reset the rd fifo (to the inverted pre-defined/MPR value) and it's pointers by setting MPDGCTRL[RST_RD_FIFO] = 1
6. Decrement the read delay line absolute offset of each byte by 1 (i.e. MPRDDLCTL[RD_DL_ABS_OFFSETn])
7. Force the delay line to measure itself and to issue the requested read delay by configuring MPMUR[FRC_MSR] = 1
8. Issue read command (with the legal DDR address chosen in step 7) from the external DDR device and waits 16 or 32 cycles (according to MPRDDLHWCTL[HW_RD_DL_CMP_CYC]) assuming that the data has arrived from the DDR device
9. Compare the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8). If the comparison fails then it is needed to store the low read boundary of the associated byte at of each byte . If the comparison passes then repeat steps 5-8. If all read data comparisons fail then advance to the next step.
10. Start seeking the upper boundary and set the read delay line absolute offset of each byte to the initial value + 1 as determined at step 4
11. Force the delay line to measure itself and to issue the requested read delay by configuring MPMUR[FRC_MSR] = 1
12. Resets the rd fifo (to the inverted pre-defined/MPR value) and it's pointers by setting MPDGCTRL[RST_RD_FIFO] = 1

13. Issue read command (with the legal DDR address chosen in step 7) from the external DDR device
14. Compare the read data byte to the associated byte in the pre-defined/MPR value for all the bytes in the DDR burst (burst length 4 or 8). If the comparison fails then it is needed to store the upper read boundary of the associated byte at of each byte. If the comparison passes then increment the read delay line absolute offset of each byte by 1 (i.e. MPRDDLCTL[RD_DL_ABS_OFFSETn])
15. Force the delay line to measure itself and to issue the requested read delay by configuring MPMUR[FRC_MSR] = 1
16. If all read data comparisons fail then advance to the next step, else repeat steps 12-15.
17. After finding the window boundary (lower and upper) of each read data byte then calculate the average between lower and upper boundaries and store the associated average at MPRDDLCTL[RD_DL_ABS_OFFSETn]
18. Force the delay line to measure itself and to issue the requested read delay by configuring MPMUR[FRC_MSR] = 1
19. Exit the DDR device from MPR/DQ calibration mode through MRS/MRW commands.

45.11.5 Write Calibration

The write calibration is used to adjust the write DQS with write data byte. It is assumed that the read calibration process is completed prior to the write calibration.

NOTE

In DDR3_x64 and LP2_1ch_x64 modes, the activation of the calibration is done by setting MPWRDLHWCTL0[HW_WR_DL_EN]

NOTE

In LP2_2ch_x16, LP2_2ch_x32 the activation of the calibration of each channel is done by setting MPWRDLHWCTL0[HW_WR_DL_EN].

45.11.5.1 HW (automatic) Write Calibration

The following steps should be executed:

1. Make sure that the initial value that is configured in the write delay line absolute offset of each byte (i.e. MPWRDLCTL[WR_DL_ABS_OFFSET#]) will place the write DQS somewhere inside the write DQ window

2. Configure the pre-defined value, which reflects the value that will be written and compared through the write calibration, to MPPDCMPR1[PDV1, PDV2]
3. Assert MPWRDLHWCTL0[HW_WR_DL_EN]

The following steps will be executed automatically:

1. MMDC waits till the write delay-line is updated with the absolute delay value for all bytes at MPWRDCTL[WR_DL_ABS_OFFSET#]
2. MMDC drives write command to the external DDR devices (to bank 0 address 0) and waits 16 or 32 cycles (according to MPWRDLHWCTL[HW_WR_DL_CMP_CYC]) assuming that the data has arrived to the DDR device.
3. MMDC drives read command to the same address from the external DDR
4. MMDC compares the read data byte to the associated byte in the pre-defined value for all the bytes in the DDR burst (burst length 4 or 8). If the comparison fails then it indicates that the initial write DQS isn't inside the write DQ window and the MMDC generates an error for the associated byte at MPWRDLHWCTL[HW_WR_DL_ERR#] . If the comparison passes then MMDC advances to next step.
5. MMDC resets the rd fifo (to the inverted pre-defined value) and it's pointers by setting MPDGCTRL[RST_RD_FIFO] = 1
6. MMDC decrements the write delay line absolute offset of each byte by 1 (i.e. MPWRDLCTL[WR_DL_ABS_OFFSET#]) and issue measurement process of the write delay-line to update itself with the new value.
7. MMDC drives write command to the external DDR devices (to bank 0 address 0) and waits 16 or 32 cycles (according to MPWRDLHWCTL[HW_WR_DL_CMP_CYC]) assuming that the data has arrived to the DDR device
8. MMDC drives read command to the same address from the external DDR
9. MMDC compares the read data byte to the associated byte in the pre-defined value for all the bytes in the DDR burst (burst length 4 or 8). If the comparison fails then it stores the low write boundary of the associated byte of each byte at MPWRDLHWST0/1[HW_WR_DL_LOW#] . If the comparison passes then MMDC repeats steps 8-11. If all data comparisons fail then the MMDC advances to the next step
10. The MMDC start seeking the upper boundary and sets the write delay line absolute offset of each byte to the initial value + 1 as determined at step 4 and issue measurement process of the write delay-line to update itself with the new value
11. MMDC resets the rd fifo (to the inverted pre-defined value) and its pointers by setting MPDGCTRL[RST_RD_FIFO] = 1
12. MMDC drives write command to the external DDR devices (to bank 0 address 0) and waits 16 or 32 cycles (according to MPWRDLHWCTL[HW_WR_DL_CMP_CYC]) assuming that the data has arrived to the DDR device.
13. MMDC drives read command to the same address from the external DDR

14. MMDC compares the read data byte to the associated byte in the pre-defined value for all the bytes in the DDR burst (burst length 4 or 8). If the comparison fails then it stores the upper write boundary of the associated byte of each byte at MPWRDLHWST0/1[HW_WR_DL_UP#] . If the comparison passes then MMDC increments the write delay line absolute offset of each byte by 1 (i.e. MPWRDLCTL[WR_DL_ABS_OFFSET#]) and issue measurement process of the write delay-line to update itself with the new value.
15. MMDC repeats steps 14-17. If all data comparisons fail then the MMDC advances to the next step
16. .After the MMDC finds the window boundary (lower and upper) of each write data byte then it stores the average between lower and upper boundaries at the associated MPWRDLCTL[WR_DL_ABS_OFFSET#] and issue measurement process of the write delay-line to update itself with the new value.
17. MMDC indicates that the write data calibration had finished by setting MPWRDLHWCTL[HW_WR_DL_EN] = 0

45.11.5.2 SW Write Calibration

The following steps should be executed:

NOTE

It is recommended to perform the write calibration using the HW method. The SW method is provided for debug purposes only.

1. Make sure that the initial value that is configured in the write delay line absolute offset of each byte (i.e. MPWRDLCTL[WR_DL_ABS_OFFSETn]) will place the write DQS somewhere inside the write DQ window
2. Configure the pre-defined value, which reflects the value that will be written and compared through the write calibration, to MPPDCMPR1[PDV1, PDV2]
3. Force the delay line to measure itself and to issue the requested write delay by configuring MPMUR0[FRC_MSR] = 1
4. Wait 16 DDR cycles till the write delay-line is updated with the absolute delay value for all bytes
5. Issue write command to any legal DDR address of the external DDR device
6. Issue read command, to the address written previously, from the external DDR device
7. Compare the read data byte to the associated byte in the pre-defined value for all bytes in the DDR burst (burst length 4 or 8). If the comparison fails then it indicates that the initial write DQS isn't inside the write DQ window. If the comparison passes then advance to next step.

8. MMDC resets the rd fifo (to the inverted pre-defined value) and it's pointers by setting MPDGCTRL[RST_RD_FIFO] = 1
9. Decrement the write delay line absolute offset of each byte by 1 (i.e. MPWRDLCTL[WR_DL_ABS_OFFSETn])
10. Force the delay line to measure itself and to issue the requested write delay by configuring MPMUR0[FRC_MSR] = 1
11. Issue write command to any legal DDR address of the external DDR device
12. Issue read command, to the address written previously, from the external DDR device
13. Compare the read data byte to the associated byte in the pre-defined value for all the bytes in the DDR burst (burst length 4 or 8). If the comparison fails then it is needed to store the low write boundary of the associated byte of each byte at MPWRDLHWST0/1[HW_WR_DL_LOWn] . If the comparison passes then repeat steps 8-12. If all data comparisons fail then advance to the next step.
14. Start seeking the upper boundary and set the write delay line absolute offset of each byte to the initial value + 1
15. Force the delay line to measure itself and to issue the requested write delay by configuring MPMUR0[FRC_MSR] = 1
16. Reset the rd fifo (to the inverted pre-defined value) and it's pointers by setting MPDGCTRL[RST_RD_FIFO] = 1
17. Issue write command to any legal DDR address of the external DDR device
18. Issue read command, to the address written previously, from the external DDR device
19. Compare the read data byte to the associated byte in the pre-defined value for all the bytes in the DDR burst (burst length 4 or 8). If the comparison fails then it is needed to store the upper write boundary of the associated byte of each byte at MPWRDLHWST0/1[HW_WR_DL_UPn]. If the comparison passes then increment the write delay line absolute offset of each byte by 1.
20. Force the delay line to measure itself and to issue the requested write delay by configuring MPMUR0[FRC_MSR] = 1
21. If all read data comparisons fail then advance to the next step else repeat steps 16-20.
22. .After finding the window boundary (lower and upper) of each write data byte then calculate the average between lower and upper boundaries and store the associated average at MPWRDLCTL[WR_DL_ABS_OFFSETn]
23. Force the delay line to measure itself and to issue the requested write delay by configuring MPMUR0[FRC_MSR] = 1

45.11.6 Write leveling Calibration

The write leveling calibration can generate a delay between the clock and the associate DQS of up to 3 cycles as following: (WL_DL_ABS_OFFSET/256*cycle) + (WL_HC_DEL*half cycle) + (WL_CYC_DEL*cycle). Write leveling calibration can be executed automatically(HW) or manually (SW).

The automatic calibration process can only detect the optimal DQS to clock delay to within 1 cycle. In extreme cases in which the DDR3 memory is placed far from the microcontroller (long address/command/clock trace lengths), the skew between the DQS and clock may exceed 1 cycle. If this is the case, it is the user's responsibility to both understand that their design causes the DQS to clock skew to exceed 1 cycle and to indicate this manually in the MPWLDECTRL0/1[WL_CYC_DEL#]. It is highly recommended to keep the DDR3 memory as close to the microcontroller as possible, especially in embedded system designs. When using fly-by topology, the user should calculate the PCB flight time of the clock signal to the furthest placed DDR3 memory to ensure less than 1 cycle skew between DQS and clock.

NOTE

In LPDDR2 mode Write-leveling calibration should be disabled.

NOTE

In DDR3_x64 mode activation of the calibration is done by setting MPWLGCR[HW_WL_EN]

NOTE

It is essential to route the first bit in each data byte group (D0, D8, D16, D24, D32, D40, D48, D56) from the DDR3 memory to the same data bus bits on the controller. The DDR3 memory outputs the state of the DRAM clock during the write leveling calibration. If any of these are not routed properly, the controller will have no information regarding the state of the DRAM clock during calibration. In previous designs in which write leveling calibration was not performed, the board designer would often swap data bits within each byte group to make the data bus routing cleaner and less susceptible to noise and impedance mismatch. This can still be done with DDR3 so long as the requirement of properly routing D0, D8, D16, D24, D32, D40, D48, D56 is maintained.

45.11.6.1 Hardware Write Leveling Calibration

The following steps should be executed:

1. Configure the external DDR device to enter write leveling mode through MRS command
2. Activate the DQS output enable by setting MDSCR[WL_EN]
3. Active automatic calibration by setting MPWLGCR[HW_WL_EN]

   The following steps will be executed automatically by the MMDC:

4. MMDC enters write leveling mode, counts 25 + 15 cycles and drives the DQS pads as output while the DQ pads will remain inputs. In parallel the MMDC configures the write leveling delay line to "0" (i.e. MPWLDECTRL0[WL_DL_ABS_OFFSET#] = 0) and issue measurement process of the writ-leveling delay-line to update itself with the new value
5. MMDC drives one DQS pulse to the DDR external device
6. MMDC waits 16 cycles (to guarantee that the DQ prime data is stable) and samples the associated prime DQ bit (for example for DQS1 the MMDC samples DQ[8])
7. MMDC increments the write leveling delay line by 1/8 cycle and perform measurement process in order to load the updated value to the associated delay-line
8. MMDC repeates steps 5-7 till the write leveling delay is 1 cycle
9. MMDC checks the 8 bit prime DQ results for each DQS and finds the first transition from 0 to 1. If no transition is found then the MMDC indicates an error at MPWLGCR[HW_WL_ERR#]
10. MMDC stores the value that issues the last "0" on the prime DQ before the transition and loads it to the write leveling delay-line. The MMDC initiates a fine-tune process by incrementing the delay-line values by 1 step (which is 1/256 part of a cycle) till detecting the most accurate transition from 0 to 1
11. Upon completion of this process the MMDC de-asserts the MPWLGCR[HW_WL_EN] and update the most accurate value of the delay-line at the associated MPWLDECTRL#[WL_DL_ABS_OFFSET#]
12. MMDC perform measurement process in order to load the most accurate value to the associated delay-line
13. User should issue MRS command to exit write leveling mode
14. The user should read the results of the associated delay-line at MPWLDECTRL#[WL_DL_ABS_OFFSET#] and in case the user estimates that the reasonable delay may be above 1 cycle then the user should indicate it at MPWLDECTRL#[WL_CYC_DEL#]. Moreover the user should indicate it in MDMISC[WALAT] field. For example, if the result of the write leveling calibration is 100/256 parts of a cycle, but the user estimates that the delay is above 2 cycles then MPWLDECTRL#[WL_CYC_DEL#] should be configured to 2, so the total delay will be 2 and 100/256 parts of a cycle

15. Return the DQS output enable to functional mode by deasserting MDSCR[WL_EN]

45.11.6.2 SW Write Leveling Calibration

The following steps should be executed:

NOTE

It is recommended to perform the write calibration using the HW method. The SW method is provided for debug purposes only.

1. Configure the external DDR device to enter write leveling mode through MRS command
2. Activate the DQS output enable by setting MDSCR[WL_EN]
3. Set the write-leveling delay-line offset to "0" by configuring MPWLDECTRL0[WL_DL_ABS_OFFSET#] = 0
4. Force the delay line to measure itself and to issue the requested write-leveling delay by configuring MPMUR[FRC_MSR] = 1
5. Activate SW write-leveling calibration and issue one DQS pulse by setting MPWLGCR[SW_WL_EN] = 1 together with MPWLGCR[SW_WL_CNT_EN] = 1
6. Issue an IP read command from MPWLGCR. If MPWLGCR[SW_WL_EN] = 0 then the SW write-leveling result is valid at MPWLGCR[WL_SW_RES#].
7. Increment the write leveling delay line by 1/8 cycle—that is, add 0x20 to {MPWLDECTRL0[WL_HC_DEL#],MPWLDECTRL0[WL_DL_ABS_OFFSET#]}
8. Force the delay line to measure itself and to issue the requested write-leveling delay by configuring MPMUR[FRC_MSR] = 1
9. Activate SW write-leveling calibration and issue one DQS pulse by setting MPWLGCR[SW_WL_EN] = 1
10. Repeate steps 6-9 till the edge of CK was detected (i.e the write-leveling result switched from "0" to "1")
11. Store the value that issues the last "0" on the prime DQ before the transition and load it to the write leveling delay-line and start fine tuning process to detect the exact switch from "0" to "1"
12. Force the delay line to measure itself and to issue the requested write-leveling delay by configuring MPMUR[FRC_MSR] = 1
13. Activate SW write-leveling calibration and issue one DQS pulse by setting MPWLGCR[SW_WL_EN] = 1
14. Issue a IP read command from MPWLGCR. If MPWLGCR[SW_WL_EN] = 0 then the SW write-leveling result is valid at MPWLGCR[WL_SW_RES#].
15. Increment the write leveling delay line by 1 step (i.e add 0x01 to MPWLDECTRL0[WL_DL_ABS_OFFSET#])

16. Force the delay line to measure itself and to issue the requested write-leveling delay by configuring MPMUR[FRC_MSR] = 1
17. Issue an IP read command from MPWLGCR. If MPWLGCR[SW_WL_EN] = 0 then the SW write-leveling result is valid at MPWLGCR[WL_SW_RES#].
18. Activate SW write-leveling calibration and issue one DQS pulse by setting MPWLGCR[SW_WL_EN] = 1
19. Repeates step 15-18 till the exact edge of CK was detected (i.e the write-leveling result switched from "0" to "1")
20. Issue MRS command to exit write leveling mode
21. Return the DQS output enable to functional mode by deasserting MDSCR[WL_EN]

45.11.7 Write fine tuning

Write fine tuning is an additional circuit that provides the ability to fine tune the timing of each of the DQ/DM bits (relative to DQS) by up to 100 ps. To use the write fine tuning, select the number of delay units in each DQ/DM I/O (maximum 3 delay units of around 30-35 ps each). The delay can be configured independently for each DQ/DM. This configuration is controlled by the MPWRDQBYnDL registers.

45.11.8 Read fine tuning

Read fine tuning is an additional circuit that provides the ability to fine tune the timing of each coming dq bits (relative to coming dqs) by up to +/-100 ps.

This is done by reducing the delay between the incoming rd_dqs by 100 ps and adding a configurable delay of up to 200 ps (6 delay units of around 30-35 ps each) for each DQ input. The delay can be configured independently for each DQ. The calibration of this mechanism can be done only by writing and reading data from the memory. Controlled by register MPRDDQBY#DL.

45.11.9 ZQ Fine Tuning

An offset can be added to the PU /PD values determined by the ZQ calibration process. The offset range is programmable from -7 to +7, controlled by the MMDC_MPPDCMPR2[ZQ_PU_OFFSET] and MMDC_MPPDCMPR2[ZQ_PD_OFFSET] fields. The offset is enabled/disabled by MMDC_MPPDCMPR2[ZQ_OFFSET_EN]. See MMDC_MPPDCMPR2 register for more information.

45.11.10 Duty cycle adjustment

Duty cycle adjustments can be made to the SDCLKx and SDQSx signals, see the MMDCx_MPDCCR register for more information.

45.12 MMDC Memory Map/Register Definition

MMDC may be configured to several modes of operation. See Table 45-11.

Table 45-11. MMDC - Modes of Operation

Mode of Operation	Abbreviation
DDR3 x16	DDR3_x16
DDR3 x32	DDR3_x32
DDR3 x64	DDR3_x64
LPDDR2 2-channels x16	LP2_2ch_x16
LPDDR2 2-channels x32	LP2_2ch_x32

NOTE

In case of LPDDR2 2-channels mode while the external memory devices are exactly the same for both channels then it is recommended to configure the registers located at 0x021B_0000 - 0x021B_0440 and 0x021B_4000 - 0x021B_4440 to the same values

In case of DDR3_x64 then IP port1 is used only to configure the parameters related to the calibration process of Byte4 - Byte7 at addresses 0x021B_4808 - 0x021B_48C0 as shown in Table 45-12 .

Table 45-12. MMDC - Modes of Operation

Mode of Operation	Associated Memory Map
DDR3 x16, x32	0x021B_0000 - 0x021B_08C4
DDR3 x64	0x021B_0000 - 0x021B_08C4 0x021B_4808 - 0x021B_48C0
LPDDR2 2-channels x16, x32	0x021B_0000 - 0x021B_08C4 0x021B_4808 - 0x021B_48C0

Table 45-13 shows the register mode of operations.

Table 45-13. Register Mode of Operations

Registers	1-Channel Mode of Operations	2-Channel Mode of Operations
MMDC Core Control Register	all	LP2_2ch_x16, LP2_2ch_x32
MMDC Core Power Down Control Register		all
MMDC Core ODT Timing Control Register		LP2_2ch_x16, LP2_2ch_x32
MMDC Core Timing Configuration Register 0		
MMDC Core Timing Configuration Register 1		
MMDC Core Timing Configuration Register 2		
MMDC Core Miscellaneous Register		
MMDC Core Special Command Register		
MMDC Core Refresh Control Register		
Reserved		all
Reserved		
MMDC Core Read/Write Command Delay Register		LP2_2ch_x16, LP2_2ch_x32
MMDC Core Out of Reset Delays Register		
MMDC Core MRR Data Register	LP2_2ch_x16, LP2_2ch_x32	LP2_2ch_x16, LP2_2ch_x32
MMDC Core Timing Configuration Register 3		LP2_2ch_x16, LP2_2ch_x32
MMDC Core MR4 Derating Register		
MMDC Core Address Space Partition Register	all	
MMDC Core AXI Reordering Control Register		
MMDC Core Power Saving Control and Status Register		
MMDC Core Exclusive ID Monitor Register0		
MMDC Core Exclusive ID Monitor Register1		
MMDC Core Debug and Profiling Control Register 0		
MMDC Core Debug and Profiling Control Register 1		
MMDC Core Debug and Profiling Status Register 0		
MMDC Core Debug and Profiling Status Register 1		
MMDC Core Debug and Profiling Status Register 2		
MMDC Core Debug and Profiling Status Register 3		
MMDC Core Debug and Profiling Status Register 4		
MMDC Core Debug and Profiling Status Register 5		
MMDC Core Step By Step Address Register		
MMDC Core Step By Step Address Attributes Register		
Reserved		all
MMDC Core General Purpose Register		LP2_2ch_x16, LP2_2ch_x32
MMDC PHY ZQ HW control register		all
MMDC PHY ZQ SW control register		
MMDC PHY Write Leveling Configuration and Error Status Register		DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Table continues on the next page...

Table 45-13. Register Mode of Operations (continued)

Registers	1-Channel Mode of Operations	2-Channel Mode of Operations
MMDC PHY Write Leveling Delay Control Register 0		
MMDC PHY Write Leveling Delay Control Register 1		DDR3_x64, LP2_2ch_x32
MMDC PHY Write Leveling delay-line Status Register		DDR3_x64, LP2_2ch_x16, LP2_2ch_x32
MMDC PHY ODT control register	DDR3_x16, DDR3_x32, DDR3_x64	DDR3_x64
MMDC PHY Read DQ Byte0 Delay Register	all	DDR3_x64, LP2_2ch_x16, LP2_2ch_x32
MMDC PHY Read DQ Byte1 Delay Register		
MMDC PHY Read DQ Byte2 Delay Register		DDR3_x64, LP2_2ch_x32
MMDC PHY Read DQ Byte3 Delay Register		
MMDC PHY Write DQ Byte0 Delay Register		DDR3_x64, LP2_2ch_x16, LP2_2ch_x32
MMDC PHY Write DQ Byte1 Delay Register		
MMDC PHY Write DQ Byte2 Delay Register		DDR3_x64, LP2_2ch_x32
MMDC PHY Write DQ Byte3 Delay Register		
MMDC PHY Read DQS Gating Control Register 0	DDR3_x16, DDR3_x32, DDR3_x64	DDR3_x64
MMDC PHY Read DQS Gating Control Register 1		
MMDC PHY Read DQS Gating delay-line Status Register		
MMDC PHY Read delay-lines Configuration Register	all	DDR3_x64, LP2_2ch_x16, LP2_2ch_x32
MMDC PHY Read delay-lines Status Register		
MMDC PHY Write delay-lines Configuration Register		
MMDC PHY Write delay-lines Status Register		
MMDC PHY CK Control Register		LP2_2ch_x16, LP2_2ch_x32
MMDC ZQ LPDDR2 HW Control Register	LP2_2ch_x16, LP2_2ch_x32	
MMDC PHY Read Delay HW Calibration Control Register	all	DDR3_x64, LP2_2ch_x16, LP2_2ch_x32
MMDC PHY Write Delay HW Calibration Control Register		
MMDC PHY Read Delay HW Calibration Status Register 0		
MMDC PHY Read Delay HW Calibration Status Register 1		DDR3_x64, LP2_2ch_x32
MMDC PHY Write Delay HW Calibration Status Register 0		DDR3_x64, LP2_2ch_x16, LP2_2ch_x32
MMDC PHY Write Delay HW Calibration Status Register 1		DDR3_x64, LP2_2ch_x32
MMDC PHY Write Leveling HW Error Register		DDR3_x64, LP2_2ch_x16, LP2_2ch_x32
MMDC PHY Read DQS Gating HW Status Register 0	DDR3_x16, DDR3_x32, DDR3_x64	DDR3_x64
MMDC PHY Read DQS Gating HW Status Register 1		
MMDC PHY Read DQS Gating HW Status Register 2		
MMDC PHY Read DQS Gating HW Status Register 3		
MMDC PHY Pre-defined Compare Register 1	all	LP2_2ch_x16, LP2_2ch_x32
MMDC PHY Pre-defined Compare and CA delay-line Configuration Register		

Table continues on the next page...

Table 45-13. Register Mode of Operations (continued)

Registers	1-Channel Mode of Operations	2-Channel Mode of Operations
MMDC PHY SW Dummy Access Register		DDR3_x64, LP2_2ch_x16, LP2_2ch_x32
MMDC PHY SW Dummy Read Data Register 0		
MMDC PHY SW Dummy Read Data Register 1		
MMDC PHY SW Dummy Read Data Register 2		
MMDC PHY SW Dummy Read Data Register 3		
MMDC PHY SW Dummy Read Data Register 4		
MMDC PHY SW Dummy Read Data Register 5		
MMDC PHY SW Dummy Read Data Register 6		
MMDC PHY SW Dummy Read Data Register 7		
MMDC PHY Measure Unit Register		LP2_2ch_x16, LP2_2ch_x32
MMDC Write CA delay-line controller	LP2_2ch_x16, LP2_2ch_x32	
MMDC Duty Cycle Control Register	all	DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Table 45-14 shows the maximum AXI address space for the main modes of operation.

Table 45-14. MMDC - Maximum AXI Address Space Per Operation Mode

Mode of Operation	Maximum AXI Address Space	Comment
LP2_2ch_x32	AXI port0: 0x8000_0000 - 0xFFFF_FFFFF	Up to 2GB for LPDDR2 Channel 0
	AXI port1: 0x1000_0000 - 0x7FFF_FFFFF	Up to 1.75GB for LPDDR2 Channel 1
DDR3_x64	AXI port0: 0x1000_0000 - 0xFFFF_FFFFF	All address space is associated with AXI port0

The Memory Map is shown below.

NOTE

The terms *clocks* and *cycles* are used interchangeably and refer to the clock period of the main ddr clock (mmdc_ch0_axi_clk_root), commonly referred to as the DDR frequency.

MMDC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21B_0000	MMDC Core Control Register (MMDC1_MDCTL)	32	R/W	0311_0000h	45.12.1/ 3957
21B_0004	MMDC Core Power Down Control Register (MMDC1_MDPDC)	32	R/W	0003_0012h	45.12.2/ 3959
21B_0008	MMDC Core ODT Timing Control Register (MMDC1_MDOTC)	32	R/W	1227_2000h	45.12.3/ 3961
21B_000C	MMDC Core Timing Configuration Register 0 (MMDC1_MDCFG0)	32	R/W	3236_22D3h	45.12.4/ 3963
21B_0010	MMDC Core Timing Configuration Register 1 (MMDC1_MDCFG1)	32	R/W	B6B1_8A23h	45.12.5/ 3965
21B_0014	MMDC Core Timing Configuration Register 2 (MMDC1_MDCFG2)	32	R/W	00C7_0092h	45.12.6/ 3967
21B_0018	MMDC Core Miscellaneous Register (MMDC1_MDMISC)	32	R/W	0000_1600h	45.12.7/ 3969
21B_001C	MMDC Core Special Command Register (MMDC1_MDSCR)	32	R/W	0000_0000h	45.12.8/ 3972
21B_0020	MMDC Core Refresh Control Register (MMDC1_MDREF)	32	R/W	0000_C000h	45.12.9/ 3975
21B_002C	MMDC Core Read/Write Command Delay Register (MMDC1_MDRWD)	32	R/W	0F9F_26D2h	45.12.10/ 3978
21B_0030	MMDC Core Out of Reset Delays Register (MMDC1_MDOR)	32	R/W	009F_0E0Eh	45.12.11/ 3980
21B_0034	MMDC Core MRR Data Register (MMDC1_MDMRR)	32	R	0000_0000h	45.12.12/ 3981
21B_0038	MMDC Core Timing Configuration Register 3 (MMDC1_MDCFG3LP)	32	R/W	0000_0000h	45.12.13/ 3982
21B_003C	MMDC Core MR4 Derating Register (MMDC1_MDMR4)	32	R/W	0000_0000h	45.12.14/ 3983
21B_0040	MMDC Core Address Space Partition Register (MMDC1_MDASP)	32	R/W	0000_003Fh	45.12.15/ 3985
21B_0400	MMDC Core AXI Reordering Control Register (MMDC1_MAARCR)	32	R/W	5142_01F0h	45.12.16/ 3986
21B_0404	MMDC Core Power Saving Control and Status Register (MMDC1_MAPSR)	32	R/W	0000_1007h	45.12.17/ 3988
21B_0408	MMDC Core Exclusive ID Monitor Register0 (MMDC1_MAEXIDR0)	32	R/W	0020_0000h	45.12.18/ 3991
21B_040C	MMDC Core Exclusive ID Monitor Register1 (MMDC1_MAEXIDR1)	32	R/W	0060_0040h	45.12.19/ 3991
21B_0410	MMDC Core Debug and Profiling Control Register 0 (MMDC1_MADPCR0)	32	R/W	0000_0000h	45.12.20/ 3992
21B_0414	MMDC Core Debug and Profiling Control Register 1 (MMDC1_MADPCR1)	32	R/W	0000_0000h	45.12.21/ 3994
21B_0418	MMDC Core Debug and Profiling Status Register 0 (MMDC1_MADPSR0)	32	R	0000_0000h	45.12.22/ 3994

Table continues on the next page...

MMDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21B_041C	MMDC Core Debug and Profiling Status Register 1 (MMDC1_MADPSR1)	32	R	0000_0000h	45.12.23/ 3995
21B_0420	MMDC Core Debug and Profiling Status Register 2 (MMDC1_MADPSR2)	32	R	0000_0000h	45.12.24/ 3995
21B_0424	MMDC Core Debug and Profiling Status Register 3 (MMDC1_MADPSR3)	32	R	0000_0000h	45.12.25/ 3996
21B_0428	MMDC Core Debug and Profiling Status Register 4 (MMDC1_MADPSR4)	32	R	0000_0000h	45.12.26/ 3996
21B_042C	MMDC Core Debug and Profiling Status Register 5 (MMDC1_MADPSR5)	32	R	0000_0000h	45.12.27/ 3997
21B_0430	MMDC Core Step By Step Address Register (MMDC1_MASBS0)	32	R	0000_0000h	45.12.28/ 3997
21B_0434	MMDC Core Step By Step Address Attributes Register (MMDC1_MASBS1)	32	R	0000_0000h	45.12.29/ 3998
21B_0440	MMDC Core General Purpose Register (MMDC1_MAGENP)	32	R/W	0000_0000h	45.12.30/ 3999
21B_0800	MMDC PHY ZQ HW control register (MMDC1_MPZQHWCTRL)	32	R/W	A138_0000h	45.12.31/ 4000
21B_0804	MMDC PHY ZQ SW control register (MMDC1_MPZQSWCTRL)	32	R/W	0000_0000h	45.12.32/ 4002
21B_0808	MMDC PHY Write Leveling Configuration and Error Status Register (MMDC1_MPWLGCR)	32	R/W	0000_0000h	45.12.33/ 4004
21B_080C	MMDC PHY Write Leveling Delay Control Register 0 (MMDC1_MPWLDECTRL0)	32	R/W	0000_0000h	45.12.34/ 4007
21B_0810	MMDC PHY Write Leveling Delay Control Register 1 (MMDC1_MPWLDECTRL1)	32	R/W	0000_0000h	45.12.35/ 4009
21B_0814	MMDC PHY Write Leveling delay-line Status Register (MMDC1_MPWLDLST)	32	R	0000_0000h	45.12.36/ 4012
21B_0818	MMDC PHY ODT control register (MMDC1_MPODTCTRL)	32	R/W	0000_0000h	45.12.37/ 4014
21B_081C	MMDC PHY Read DQ Byte0 Delay Register (MMDC1_MPRDDQBY0DL)	32	R/W	0000_0000h	45.12.38/ 4016
21B_0820	MMDC PHY Read DQ Byte1 Delay Register (MMDC1_MPRDDQBY1DL)	32	R/W	0000_0000h	45.12.39/ 4019
21B_0824	MMDC PHY Read DQ Byte2 Delay Register (MMDC1_MPRDDQBY2DL)	32	R/W	0000_0000h	45.12.40/ 4021
21B_0828	MMDC PHY Read DQ Byte3 Delay Register (MMDC1_MPRDDQBY3DL)	32	R/W	0000_0000h	45.12.41/ 4024
21B_082C	MMDC PHY Write DQ Byte0 Delay Register (MMDC1_MPWRDQBY0DL)	32	R/W	0000_0000h	45.12.42/ 4027
21B_0830	MMDC PHY Write DQ Byte1 Delay Register (MMDC1_MPWRDQBY1DL)	32	R/W	0000_0000h	45.12.43/ 4029
21B_0834	MMDC PHY Write DQ Byte2 Delay Register (MMDC1_MPWRDQBY2DL)	32	R/W	0000_0000h	45.12.44/ 4031

Table continues on the next page...

MMDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21B_0838	MMDC PHY Write DQ Byte3 Delay Register (MMDC1_MPWRDQBY3DL)	32	R/W	0000_0000h	45.12.45/ 4033
21B_083C	MMDC PHY Read DQS Gating Control Register 0 (MMDC1_MPDGCTRL0)	32	R/W	0000_0000h	45.12.46/ 4036
21B_0840	MMDC PHY Read DQS Gating Control Register 1 (MMDC1_MPDGCTRL1)	32	R/W	0000_0000h	45.12.47/ 4038
21B_0844	MMDC PHY Read DQS Gating delay-line Status Register (MMDC1_MPDGDLST0)	32	R	0000_0000h	45.12.48/ 4040
21B_0848	MMDC PHY Read delay-lines Configuration Register (MMDC1_MPRDDLCTL)	32	R/W	4040_4040h	45.12.49/ 4042
21B_084C	MMDC PHY Read delay-lines Status Register (MMDC1_MPRDDLST)	32	R	0000_0000h	45.12.50/ 4043
21B_0850	MMDC PHY Write delay-lines Configuration Register (MMDC1_MPWRDLCTL)	32	R/W	4040_4040h	45.12.51/ 4044
21B_0854	MMDC PHY Write delay-lines Status Register (MMDC1_MPWRDLST)	32	R	0000_0000h	45.12.52/ 4046
21B_0858	MMDC PHY CK Control Register (MMDC1_MPSDCTRL)	32	R/W	0000_0000h	45.12.53/ 4047
21B_085C	MMDC ZQ LPDDR2 HW Control Register (MMDC1_MPZQLP2CTL)	32	R/W	1B5F_0109h	45.12.54/ 4048
21B_0860	MMDC PHY Read Delay HW Calibration Control Register (MMDC1_MPRDDLHWCTL)	32	R/W	0000_0000h	45.12.55/ 4049
21B_0864	MMDC PHY Write Delay HW Calibration Control Register (MMDC1_MPWRDLHWCTL)	32	R/W	0000_0000h	45.12.56/ 4051
21B_0868	MMDC PHY Read Delay HW Calibration Status Register 0 (MMDC1_MPRDDLHWST0)	32	R	0000_0000h	45.12.57/ 4053
21B_086C	MMDC PHY Read Delay HW Calibration Status Register 1 (MMDC1_MPRDDLHWST1)	32	R	0000_0000h	45.12.58/ 4054
21B_0870	MMDC PHY Write Delay HW Calibration Status Register 0 (MMDC1_MPWRDLHWST0)	32	R	0000_0000h	45.12.59/ 4055
21B_0874	MMDC PHY Write Delay HW Calibration Status Register 1 (MMDC1_MPWRDLHWST1)	32	R	0000_0000h	45.12.60/ 4056
21B_0878	MMDC PHY Write Leveling HW Error Register (MMDC1_MPWLHWERR)	32	R/W	0000_0000h	45.12.61/ 4057
21B_087C	MMDC PHY Read DQS Gating HW Status Register 0 (MMDC1_MPDGHWST0)	32	R	0000_0000h	45.12.62/ 4057
21B_0880	MMDC PHY Read DQS Gating HW Status Register 1 (MMDC1_MPDGHWST1)	32	R	0000_0000h	45.12.63/ 4058
21B_0884	MMDC PHY Read DQS Gating HW Status Register 2 (MMDC1_MPDGHWST2)	32	R	0000_0000h	45.12.64/ 4059
21B_0888	MMDC PHY Read DQS Gating HW Status Register 3 (MMDC1_MPDGHWST3)	32	R	0000_0000h	45.12.65/ 4059
21B_088C	MMDC PHY Pre-defined Compare Register 1 (MMDC1_MPPDCMPR1)	32	R/W	0000_0000h	45.12.66/ 4060

Table continues on the next page...

MMDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21B_0890	MMDC PHY Pre-defined Compare and CA delay-line Configuration Register (MMDC1_MPPDCMPR2)	32	R/W	0040_0000h	45.12.67/ 4061
21B_0894	MMDC PHY SW Dummy Access Register (MMDC1_MPSWDAR0)	32	R/W	0000_0000h	45.12.68/ 4062
21B_0898	MMDC PHY SW Dummy Read Data Register 0 (MMDC1_MPSWDRDR0)	32	R	FFFF_FFFFh	45.12.69/ 4064
21B_089C	MMDC PHY SW Dummy Read Data Register 1 (MMDC1_MPSWDRDR1)	32	R	FFFF_FFFFh	45.12.70/ 4065
21B_08A0	MMDC PHY SW Dummy Read Data Register 2 (MMDC1_MPSWDRDR2)	32	R	FFFF_FFFFh	45.12.71/ 4065
21B_08A4	MMDC PHY SW Dummy Read Data Register 3 (MMDC1_MPSWDRDR3)	32	R	FFFF_FFFFh	45.12.72/ 4066
21B_08A8	MMDC PHY SW Dummy Read Data Register 4 (MMDC1_MPSWDRDR4)	32	R	FFFF_FFFFh	45.12.73/ 4066
21B_08AC	MMDC PHY SW Dummy Read Data Register 5 (MMDC1_MPSWDRDR5)	32	R	FFFF_FFFFh	45.12.74/ 4067
21B_08B0	MMDC PHY SW Dummy Read Data Register 6 (MMDC1_MPSWDRDR6)	32	R	FFFF_FFFFh	45.12.75/ 4067
21B_08B4	MMDC PHY SW Dummy Read Data Register 7 (MMDC1_MPSWDRDR7)	32	R	FFFF_FFFFh	45.12.76/ 4068
21B_08B8	MMDC PHY Measure Unit Register (MMDC1_MPMUR0)	32	R/W	0000_0000h	45.12.77/ 4068
21B_08BC	MMDC Write CA delay-line controller (MMDC1_MPWRCADL)	32	R/W	0000_0000h	45.12.78/ 4069
21B_08C0	MMDC Duty Cycle Control Register (MMDC1_MPDCCR)	32	R/W	2492_2492h	45.12.79/ 4071
21B_4000	MMDC Core Control Register (MMDC2_MDCTL)	32	R/W	0311_0000h	45.12.1/ 3957
21B_4004	MMDC Core Power Down Control Register (MMDC2_MDPDC)	32	R/W	0003_0012h	45.12.2/ 3959
21B_4008	MMDC Core ODT Timing Control Register (MMDC2_MDOTC)	32	R/W	1227_2000h	45.12.3/ 3961
21B_400C	MMDC Core Timing Configuration Register 0 (MMDC2_MDCFG0)	32	R/W	3236_22D3h	45.12.4/ 3963
21B_4010	MMDC Core Timing Configuration Register 1 (MMDC2_MDCFG1)	32	R/W	B6B1_8A23h	45.12.5/ 3965
21B_4014	MMDC Core Timing Configuration Register 2 (MMDC2_MDCFG2)	32	R/W	00C7_0092h	45.12.6/ 3967
21B_4018	MMDC Core Miscellaneous Register (MMDC2_MDMISC)	32	R/W	0000_1600h	45.12.7/ 3969
21B_401C	MMDC Core Special Command Register (MMDC2_MDSCR)	32	R/W	0000_0000h	45.12.8/ 3972
21B_4020	MMDC Core Refresh Control Register (MMDC2_MDREF)	32	R/W	0000_C000h	45.12.9/ 3975

Table continues on the next page...

MMDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21B_402C	MMDC Core Read/Write Command Delay Register (MMDC2_MDRWD)	32	R/W	0F9F_26D2h	45.12.10/ 3978
21B_4030	MMDC Core Out of Reset Delays Register (MMDC2_MDOR)	32	R/W	009F_0E0Eh	45.12.11/ 3980
21B_4034	MMDC Core MRR Data Register (MMDC2_MDMRR)	32	R	0000_0000h	45.12.12/ 3981
21B_4038	MMDC Core Timing Configuration Register 3 (MMDC2_MDCFG3LP)	32	R/W	0000_0000h	45.12.13/ 3982
21B_403C	MMDC Core MR4 Derating Register (MMDC2_MDMR4)	32	R/W	0000_0000h	45.12.14/ 3983
21B_4040	MMDC Core Address Space Partition Register (MMDC2_MDASP)	32	R/W	0000_003Fh	45.12.15/ 3985
21B_4400	MMDC Core AXI Reordering Control Register (MMDC2_MAARCR)	32	R/W	5142_01F0h	45.12.16/ 3986
21B_4404	MMDC Core Power Saving Control and Status Register (MMDC2_MAPSR)	32	R/W	0000_1007h	45.12.17/ 3988
21B_4408	MMDC Core Exclusive ID Monitor Register0 (MMDC2_MAEXIDR0)	32	R/W	0020_0000h	45.12.18/ 3991
21B_440C	MMDC Core Exclusive ID Monitor Register1 (MMDC2_MAEXIDR1)	32	R/W	0060_0040h	45.12.19/ 3991
21B_4410	MMDC Core Debug and Profiling Control Register 0 (MMDC2_MADPCR0)	32	R/W	0000_0000h	45.12.20/ 3992
21B_4414	MMDC Core Debug and Profiling Control Register 1 (MMDC2_MADPCR1)	32	R/W	0000_0000h	45.12.21/ 3994
21B_4418	MMDC Core Debug and Profiling Status Register 0 (MMDC2_MADPSR0)	32	R	0000_0000h	45.12.22/ 3994
21B_441C	MMDC Core Debug and Profiling Status Register 1 (MMDC2_MADPSR1)	32	R	0000_0000h	45.12.23/ 3995
21B_4420	MMDC Core Debug and Profiling Status Register 2 (MMDC2_MADPSR2)	32	R	0000_0000h	45.12.24/ 3995
21B_4424	MMDC Core Debug and Profiling Status Register 3 (MMDC2_MADPSR3)	32	R	0000_0000h	45.12.25/ 3996
21B_4428	MMDC Core Debug and Profiling Status Register 4 (MMDC2_MADPSR4)	32	R	0000_0000h	45.12.26/ 3996
21B_442C	MMDC Core Debug and Profiling Status Register 5 (MMDC2_MADPSR5)	32	R	0000_0000h	45.12.27/ 3997
21B_4430	MMDC Core Step By Step Address Register (MMDC2_MASBS0)	32	R	0000_0000h	45.12.28/ 3997
21B_4434	MMDC Core Step By Step Address Attributes Register (MMDC2_MASBS1)	32	R	0000_0000h	45.12.29/ 3998
21B_4440	MMDC Core General Purpose Register (MMDC2_MAGENP)	32	R/W	0000_0000h	45.12.30/ 3999
21B_4800	MMDC PHY ZQ HW control register (MMDC2_MPZQHWCTRL)	32	R/W	A138_0000h	45.12.31/ 4000

Table continues on the next page...

MMDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21B_4804	MMDC PHY ZQ SW control register (MMDC2_MPZQSWCTRL)	32	R/W	0000_0000h	45.12.32/ 4002
21B_4808	MMDC PHY Write Leveling Configuration and Error Status Register (MMDC2_MPWLGCR)	32	R/W	0000_0000h	45.12.33/ 4004
21B_480C	MMDC PHY Write Leveling Delay Control Register 0 (MMDC2_MPWLDECTRL0)	32	R/W	0000_0000h	45.12.34/ 4007
21B_4810	MMDC PHY Write Leveling Delay Control Register 1 (MMDC2_MPWLDECTRL1)	32	R/W	0000_0000h	45.12.35/ 4009
21B_4814	MMDC PHY Write Leveling delay-line Status Register (MMDC2_MPWLDLST)	32	R	0000_0000h	45.12.36/ 4012
21B_4818	MMDC PHY ODT control register (MMDC2_MPODTCTRL)	32	R/W	0000_0000h	45.12.37/ 4014
21B_481C	MMDC PHY Read DQ Byte0 Delay Register (MMDC2_MPRDDQBY0DL)	32	R/W	0000_0000h	45.12.38/ 4016
21B_4820	MMDC PHY Read DQ Byte1 Delay Register (MMDC2_MPRDDQBY1DL)	32	R/W	0000_0000h	45.12.39/ 4019
21B_4824	MMDC PHY Read DQ Byte2 Delay Register (MMDC2_MPRDDQBY2DL)	32	R/W	0000_0000h	45.12.40/ 4021
21B_4828	MMDC PHY Read DQ Byte3 Delay Register (MMDC2_MPRDDQBY3DL)	32	R/W	0000_0000h	45.12.41/ 4024
21B_482C	MMDC PHY Write DQ Byte0 Delay Register (MMDC2_MPWRDQBY0DL)	32	R/W	0000_0000h	45.12.42/ 4027
21B_4830	MMDC PHY Write DQ Byte1 Delay Register (MMDC2_MPWRDQBY1DL)	32	R/W	0000_0000h	45.12.43/ 4029
21B_4834	MMDC PHY Write DQ Byte2 Delay Register (MMDC2_MPWRDQBY2DL)	32	R/W	0000_0000h	45.12.44/ 4031
21B_4838	MMDC PHY Write DQ Byte3 Delay Register (MMDC2_MPWRDQBY3DL)	32	R/W	0000_0000h	45.12.45/ 4033
21B_483C	MMDC PHY Read DQS Gating Control Register 0 (MMDC2_MPDGCTRL0)	32	R/W	0000_0000h	45.12.46/ 4036
21B_4840	MMDC PHY Read DQS Gating Control Register 1 (MMDC2_MPDGCTRL1)	32	R/W	0000_0000h	45.12.47/ 4038
21B_4844	MMDC PHY Read DQS Gating delay-line Status Register (MMDC2_MPDGDLST0)	32	R	0000_0000h	45.12.48/ 4040
21B_4848	MMDC PHY Read delay-lines Configuration Register (MMDC2_MPRDDLCTL)	32	R/W	4040_4040h	45.12.49/ 4042
21B_484C	MMDC PHY Read delay-lines Status Register (MMDC2_MPRDDLST)	32	R	0000_0000h	45.12.50/ 4043
21B_4850	MMDC PHY Write delay-lines Configuration Register (MMDC2_MPWRDLCTL)	32	R/W	4040_4040h	45.12.51/ 4044
21B_4854	MMDC PHY Write delay-lines Status Register (MMDC2_MPWRDLST)	32	R	0000_0000h	45.12.52/ 4046
21B_4858	MMDC PHY CK Control Register (MMDC2_MPSDCTRL)	32	R/W	0000_0000h	45.12.53/ 4047

Table continues on the next page...

MMDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21B_485C	MMDC ZQ LPDDR2 HW Control Register (MMDC2_MPZQLP2CTL)	32	R/W	1B5F_0109h	45.12.54/ 4048
21B_4860	MMDC PHY Read Delay HW Calibration Control Register (MMDC2_MPRDDLHWCTL)	32	R/W	0000_0000h	45.12.55/ 4049
21B_4864	MMDC PHY Write Delay HW Calibration Control Register (MMDC2_MPWRDLHWCTL)	32	R/W	0000_0000h	45.12.56/ 4051
21B_4868	MMDC PHY Read Delay HW Calibration Status Register 0 (MMDC2_MPRDDLHWST0)	32	R	0000_0000h	45.12.57/ 4053
21B_486C	MMDC PHY Read Delay HW Calibration Status Register 1 (MMDC2_MPRDDLHWST1)	32	R	0000_0000h	45.12.58/ 4054
21B_4870	MMDC PHY Write Delay HW Calibration Status Register 0 (MMDC2_MPWRDLHWST0)	32	R	0000_0000h	45.12.59/ 4055
21B_4874	MMDC PHY Write Delay HW Calibration Status Register 1 (MMDC2_MPWRDLHWST1)	32	R	0000_0000h	45.12.60/ 4056
21B_4878	MMDC PHY Write Leveling HW Error Register (MMDC2_MPWLHWERR)	32	R/W	0000_0000h	45.12.61/ 4057
21B_487C	MMDC PHY Read DQS Gating HW Status Register 0 (MMDC2_MPDGHWST0)	32	R	0000_0000h	45.12.62/ 4057
21B_4880	MMDC PHY Read DQS Gating HW Status Register 1 (MMDC2_MPDGHWST1)	32	R	0000_0000h	45.12.63/ 4058
21B_4884	MMDC PHY Read DQS Gating HW Status Register 2 (MMDC2_MPDGHWST2)	32	R	0000_0000h	45.12.64/ 4059
21B_4888	MMDC PHY Read DQS Gating HW Status Register 3 (MMDC2_MPDGHWST3)	32	R	0000_0000h	45.12.65/ 4059
21B_488C	MMDC PHY Pre-defined Compare Register 1 (MMDC2_MPPDCMPR1)	32	R/W	0000_0000h	45.12.66/ 4060
21B_4890	MMDC PHY Pre-defined Compare and CA delay-line Configuration Register (MMDC2_MPPDCMPR2)	32	R/W	0040_0000h	45.12.67/ 4061
21B_4894	MMDC PHY SW Dummy Access Register (MMDC2_MPSWDAR0)	32	R/W	0000_0000h	45.12.68/ 4062
21B_4898	MMDC PHY SW Dummy Read Data Register 0 (MMDC2_MPSWDRDR0)	32	R	FFFF_FFFFh	45.12.69/ 4064
21B_489C	MMDC PHY SW Dummy Read Data Register 1 (MMDC2_MPSWDRDR1)	32	R	FFFF_FFFFh	45.12.70/ 4065
21B_48A0	MMDC PHY SW Dummy Read Data Register 2 (MMDC2_MPSWDRDR2)	32	R	FFFF_FFFFh	45.12.71/ 4065
21B_48A4	MMDC PHY SW Dummy Read Data Register 3 (MMDC2_MPSWDRDR3)	32	R	FFFF_FFFFh	45.12.72/ 4066
21B_48A8	MMDC PHY SW Dummy Read Data Register 4 (MMDC2_MPSWDRDR4)	32	R	FFFF_FFFFh	45.12.73/ 4066
21B_48AC	MMDC PHY SW Dummy Read Data Register 5 (MMDC2_MPSWDRDR5)	32	R	FFFF_FFFFh	45.12.74/ 4067
21B_48B0	MMDC PHY SW Dummy Read Data Register 6 (MMDC2_MPSWDRDR6)	32	R	FFFF_FFFFh	45.12.75/ 4067

Table continues on the next page...

MMDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21B_48B4	MMDC PHY SW Dummy Read Data Register 7 (MMDC2_MPSWDRDR7)	32	R	FFFF_FFFFh	45.12.76/ 4068
21B_48B8	MMDC PHY Measure Unit Register (MMDC2_MPMUR0)	32	R/W	0000_0000h	45.12.77/ 4068
21B_48BC	MMDC Write CA delay-line controller (MMDC2_MPWRCADL)	32	R/W	0000_0000h	45.12.78/ 4069
21B_48C0	MMDC Duty Cycle Control Register (MMDC2_MPDCCR)	32	R/W	2492_2492h	45.12.79/ 4071

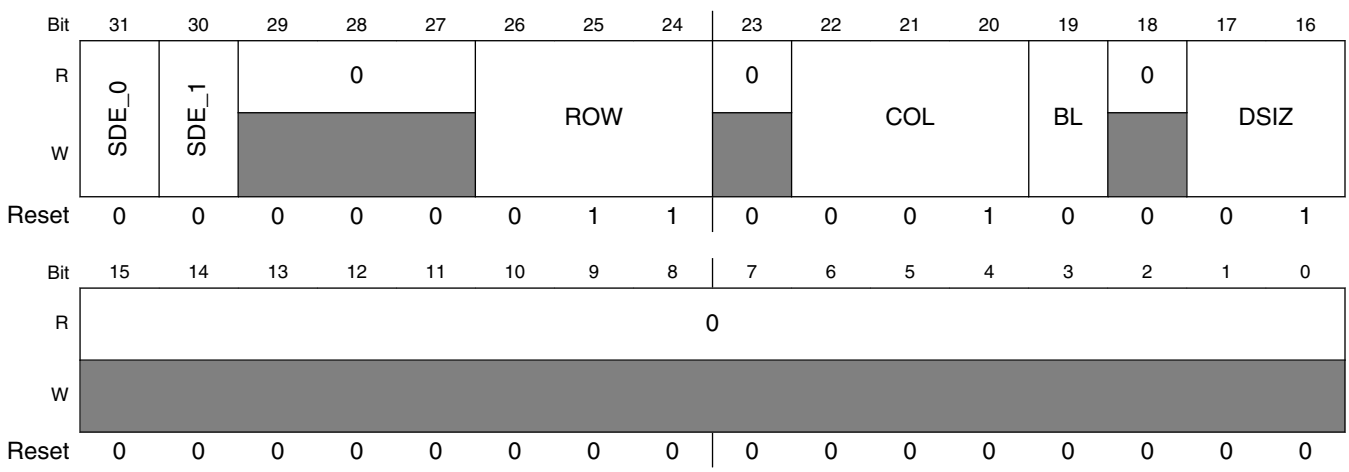
45.12.1 MMDC Core Control Register (MMDCx_MDCTL)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 0h offset



MMDCx_MDCTL field descriptions

Field	Description
31 SDE_0	MMDC Enable CS0. This bit enables/disables accesses from the MMDC toward Chip Select 0. The reset value of this bit is "0" (i.e. No clocks and clock enable will be driven to the memory). At the enabling point the MMDC will perform an initialization process (including a delay on RESET and/or CKE) for both chip selects. The initialization length depends on the configured memory type. 0 Disabled 1 Enabled
30 SDE_1	MMDC Enable CS1. This bit enables/disables accesses from the MMDC toward Chip Select 1. The reset value of this bit is "0" (i.e. No clocks and clock enable will be driven to the memory).

Table continues on the next page...

MMDCx_MDCTL field descriptions (continued)

Field	Description
	At the enabling point the MMDC will perform an initialization process (including a delay on RESET and/or CKE) for both chip selects. The initialization length depends on the configured memory type. 0 Disabled 1 Enabled
29–27 Reserved	This read-only field is reserved and always has the value 0.
26–24 ROW	Row Address Width. This field specifies the number of row addresses used by the memory array. It will affect the way an incoming address will be decoded. Settings 110-111 are reserved 000 11 bits Row 001 12 bits Row 010 13 bits Row 011 14 bits Row 100 15 bits Row 101 16 bits Row
23 Reserved	This read-only field is reserved and always has the value 0.
22–20 COL	Column Address Width. This field specifies the number of column addresses used by the memory array. It will determine how an incoming address will be decoded. 0x0 9 bits column 0x1 10 bits column 0x2 11 bits column 0x3 8 bits column 0x4 12 bits column 0x5-0xF Reserved
19 BL	Burst Length. This field determines the burst length of the DDR device. In LPDDR2 mode the MMDC supports burst length 4. In DDR3 mode the MMDC supports burst length 8. 0 Burst Length 4 is used 1 Burst Length 8 is used
18 Reserved	This read-only field is reserved and always has the value 0.
17–16 DSIZ	DDR data bus size. This field determines the size of the data bus of the DDR memory 0 16-bit data bus 1 32-bit data bus — 2 64-bit data bus 3 Reserved —
Reserved	This read-only field is reserved and always has the value 0.

45.12.2 MMDC Core Power Down Control Register (MMDCx_MDPDC)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

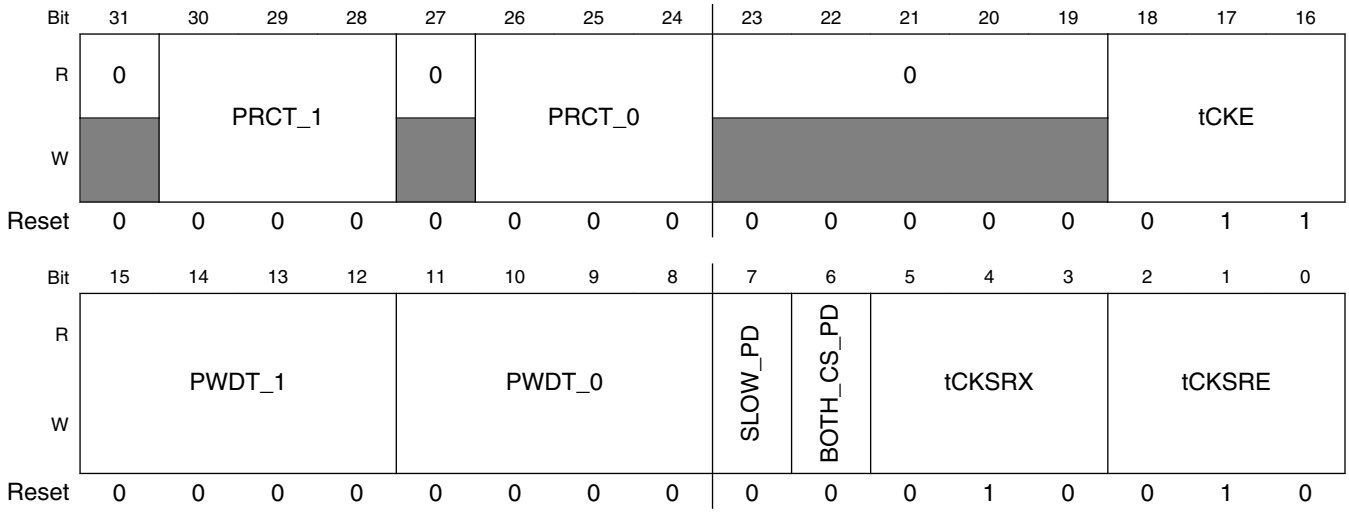
Table 45-15. PRCT field encoding

PRCT[2:0]	Precharge Timer
000	Disabled (Bit field reset value)
001	2 clocks
010	4 clocks
011	8 clocks
100	16 clocks
101	32 clocks
110	64 clocks
111	128 clocks

Table 45-16. PWDT field encoding

PWDT[3:0]	Power Down Time-out
0000	Disabled (bit field reset value)
0001	16 cycles
0010	32 cycles
0011	64 cycles
0100	128 cycles
0101	256 cycles
0110	512 cycles
0111	1024 cycles
1000	2048 cycles
1001	4096 cycles
1010	8196 cycles
1011	16384 cycles
1100	32768 cycles
1101-1111	Reserved

Address: Base address + 4h offset



MMDCx_MDPDC field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–28 PRCT_1	Precharge Timer - Chip Select 1. This field determines the amount of idle cycle for which chip select 1 will be automatically precharged. The amount of cycles are determined according to the PRCT Field Encoding table above.
27 Reserved	This read-only field is reserved and always has the value 0.
26–24 PRCT_0	Precharge Timer - Chip Select 0. This field determines the amount of idle cycle for which chip select 0 will be automatically precharged. The amount of cycles are determined according to the table below.
23–19 Reserved	This read-only field is reserved and always has the value 0.
18–16 tCKE	CKE minimum pulse width. This field determines the minimum pulse width of CKE. 0x0 1 cycle 0x1 2 cycles 0x6 7 cycles 0x7 8 cycles
15–12 PWDT_1	Power Down Timer - Chip Select 1. This field determines the amount of idle cycle for which chip select 1 will be automatically get into precharge/active power down. The amount of cycles are determined according to the PWDT Field Encoding table above.
11–8 PWDT_0	Power Down Timer - Chip Select 0. This field determines the amount of idle cycle for which chip select 0 will be automatically get into precharge/active power down. The amount of cycles are determined according to the PWDT Field Encoding table above.
7 SLOW_PD	Slow/fast power down. In DDR3 mode this field is referred to slow precharge power-down. LPDDR2 mode: This field is not relevant and should remain in its default reset value.

Table continues on the next page...

MMDCx_MDPDC field descriptions (continued)

Field	Description
	NOTE: Memory should be configured the same. 0 Fast mode. 1 Slow mode.
6 BOTH_CS_PD	Parallel power down entry to both chip selects. When power down timer is used for both chip-selects (i.e. PWDT_0 and PWDT1 don't equal "0") , then if this bit is enabled, the MMDC will enter power down only if the amount of idle cycles of both chip selects was obtained. 0 Each chip select can enter power down independently according to its configuration. 1 Chip selects can enter power down only if the amount of idle cycles of both chip selects was obtained.
5–3 tCKSRX	Valid clock cycles before self-refresh exit. This field determines the amount of clock cycles before self-refresh exit. 0x0 0 cycle 0x1 1 cycles 0x6 6 cycles 0x7 7 cycles
tCKSRE	Valid clock cycles after self-refresh entry. This field determines the amount of clock cycles after self-refresh entry. 0x0 0 cycle 0x1 1 cycles 0x6 6 cycles 0x7 7 cycles

45.12.3 MMDC Core ODT Timing Control Register (MMDCx_MDOTC)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

For further information see ODT Configuration .

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		tAOFPD			tAONPD			tANPD				tAXPD			
W																
Reset	0	0	0	1	0	0	1	0	0	0	1	0	0	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	tODTLon			0				tODT_idle_off				0			
W																
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MDOTC field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value 0.
29–27 tAOFPD	Asynchronous RTT turn-off delay (power down with DLL frozen). This field determines the time between termination circuit starts to turn off the ODT resistance till termination has reached high impedance. LPDDR2 mode: This field is not relevant and should remain in its default reset value. 0x0 1 cycle 0x1 2 cycles 0x6 7 cycles 0x7 8 cycles
26–24 tAONPD	Asynchronous RTT turn-on delay (power down with DLL frozen). This field determines the time between termination circuit gets out of high impedance and begins to turn on till ODT resistance are fully on. LPDDR2 mode: This field is not relevant and should remain in its default reset value. 0x0 1 cycle 0x1 2 cycles 0x6 7 cycles 0x7 8 cycles
23–20 tANPD	Asynchronous ODT to power down entry delay. In DDR3 should be set to tCWL-1 LPDDR2 mode: This field is not relevant and should remain in its default reset value. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xE 15 clocks 0xF 16 clocks
19–16 tAXPD	Asynchronous ODT to power down exit delay. In DDR3 should be set to tCWL-1 LPDDR2 mode: This field is not relevant and should remain in its default reset value. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xE 15 clocks 0xF 16 clocks
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 tODTLon	ODT turn on latency. This field determines the delay between ODT signal and the associated RTT, where according to JEDEC standard it equals WL(write latency) - 2. Therefore, the value that is configured to tODTLon field should correspond the value that is configured to MDCGFG1[tCWL] LPDDR2 mode: This field is not relevant and should remain in its default reset value. 0x0 - 0x1 Reserved 0x2 2 cycles 0x3 3 cycles 0x4 4 cycles 0x5 5 cycles

Table continues on the next page...

MMDCx_MDOTC field descriptions (continued)

Field	Description
	0x6 6 cycles 0x7 Reserved
11–9 Reserved	This read-only field is reserved and always has the value 0.
8–4 tODT_idle_off	ODT turn off latency. This field determines the Idle period before turning memory ODT off. **NOTE:** LPDDR2 mode: This field is not relevant and should remain in its default reset value. 0x0 0 cycle (turned off at the earliest possible time) 0x1 1 cycle 0x2 2 cycles 0x1E 30 cycles 0x1F 31 cycles
Reserved	This read-only field is reserved and always has the value 0.

45.12.4 MMDC Core Timing Configuration Register 0 (MMDCx_MDCFG0)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	tRFC								tXS								tXP			tXPDLL				tFAW						tCL			
Reset	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	0	1	0	0	0	1	0	1	1	0	1	0	0	1	1

MMDCx_MDCFG0 field descriptions

Field	Description
31–24 tRFC	Refresh command to Active or Refresh command time. See DDR3 SDRAM Specification JESD79-3E (July 2010) and LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xFE 255 clocks 0xFF 256 clocks
23–16 tXS	Exit self refresh to non READ command. In LPDDR2 it is called tXSR, self-refresh exit to next valid command delay.

Table continues on the next page...

MMDCx_MDCFG0 field descriptions (continued)

Field	Description
	See DDR3 SDRAM Specification JESD79-3E (July 2010) , LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 - 0x15 reserved 0x16 23 clocks 0x17 24 clocks 0xFE 255 clocks 0xFF 256 clocks
15–13 tXP	Exit power down with DLL-on to any valid command. Exit power down with DLL-frozen to commands not requiring a locked DLL In LPDDR2/LPDDR3 mode this field is referred to Exit power-down to next valid command delay. See DDR3 SDRAM Specification JESD79-3E (July 2010) , LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 1 cycle 0x1 2 cycles 0x6 7 cycles 0x7 8 cycles
12–9 tXPDLL	Exit precharge power down with DLL frozen to commands requiring DLL. LPDDR2 mode: This field is not relevant and should remain in its default reset value. See DDR3 SDRAM Specification JESD79-3E (July 2010) , LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xE 15 clocks 0xF 16 clocks
8–4 tFAW	Four Active Window (all banks). See DDR3 SDRAM Specification JESD79-3E (July 2010) , LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0x1E 31 clocks 0x1F 32 clocks
tCL	CAS Read Latency. In DDR3 mode this field is referred to CL. In LPDDR2 mode this field is referred to RL. **NOTE:** In LPDDR2/LPDDR3 mode only the RL/WL pairs are allowed as specified in MR2 register. See DDR3 SDRAM Specification JESD79-3E (July 2010) , LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 3 cycles 0x1 4 cycles

Table continues on the next page...

MMDCx_MDCFG0 field descriptions (continued)

Field	Description
	0x2 5 cycles 0x3 6 cycles 0x4 7 cycles 0x5 8 cycles 0x6 9 cycles 0x7 10 cycles 0x8 11 cycles 0x9 - 0xF Reserved

45.12.5 MMDC Core Timing Configuration Register 1 (MMDCx_MDCFG1)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	tRCD			tRP			tRC					tRAS					0				tWR			tMRD				0			tCWL		
Reset	1	0	1	1	0	1	1	0	1	0	1	1	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	0	0	0	1	1

MMDCx_MDCFG1 field descriptions

Field	Description
31–29 tRCD	Active command to internal read or write delay time (same bank). This field is valid only for DDR2/DDR3 memories In LPDDR2/LPDDR3 mode, this parameter should be configured at tRCD_LP. See DDR3 SDRAM Specification JESD79-3E (July 2010) , LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0x3 4 clocks 0x4 5 clocks 0x5 6 clocks 0x6 7 clocks 0x7 8 clocks
28–26 tRP	Precharge command period (same bank). This field is valid only for DDR2/DDR3 memories

Table continues on the next page...

MMDCx_MDCFG1 field descriptions (continued)

Field	Description
	In LPDDR2 mode this parameter should be configured at tRPpb_LP. See DDR3 SDRAM Specification JESD79-3E (July 2010) , LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0x3 4 clocks 0x4 5 clocks 0x5 6 clocks 0x6 7 clocks 0x7 8 clocks
25–21 tRC	Active to Active or Refresh command period (same bank). This field is valid only for DDR2/DDR3 memories In LPDDR2/LPDDR3 mode, this parameter should be configured at tRC_LP. See DDR3 SDRAM Specification JESD79-3E (July 2010) , LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0x1E 31 clocks 0x1F 32 clocks
20–16 tRAS	Active to Precharge command period (same bank). See DDR3 SDRAM Specification JESD79-3E (July 2010) , LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0x1E 31 clocks 0x1F Reserved
15–12 Reserved	This read-only field is reserved and always has the value 0.
11–9 tWR	WRITE recovery time (same bank). See DDR3 SDRAM Specification JESD79-3E (July 2010) , LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 1cycle 0x1 2cycles 0x2 3cycles 0x3 4cycles 0x4 5cycles 0x5 6cycles 0x6 7cycles 0x7 8 cycles

Table continues on the next page...

MMDCx_MDCFG1 field descriptions (continued)

Field	Description
8–5 tMRD	Mode Register Set command cycle (all banks). In DDR3 mode this field must be set to max (tMRD,tMOD). In LPDDR2/LPDDR3 mode this field should be set to max(tMRR,tMRW). See DDR3 SDRAM Specification JESD79-3E (July 2010) , LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xE 15 clocks 0xF 16 clocks
4–3 Reserved	This read-only field is reserved and always has the value 0.
tCWL	CAS Write Latency. In DDR3 mode this field is referred to CWL. In LPDDR2/LPDDR3 mode this field is referred to WL. 0x0 2 cycles (DDR2/DDR3) ,1 cycles (LPDDR2/LPDDR3) 0x1 3 cycles (DDR2/DDR3) ,2 cycles (LPDDR2/LPDDR3) 0x2 4 cycles (DDR2/DDR3) ,3 cycles (LPDDR2/LPDDR3) 0x3 5 cycles (DDR2/DDR3) ,4 cycles (LPDDR2/LPDDR3) 0x4 6 cycles (DDR2/DDR3) ,5 cycles (LPDDR2/LPDDR3) 0x5 7 cycles (DDR2/DDR3) ,6 cycles (LPDDR2/LPDDR3) 0x6 8 cycles (DDR2/DDR3) ,7 cycles (LPDDR2/LPDDR3) 0x7 Reserved

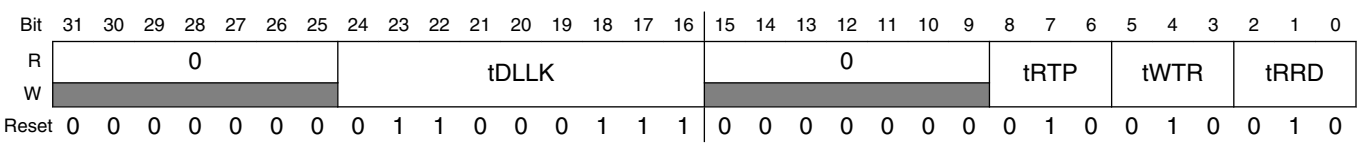
45.12.6 MMDC Core Timing Configuration Register 2 (MMDCx_MDCFG2)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 14h offset



MMDCx_MDCFG2 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–16 tDLLK	DLL locking time. LPDDR2 mode: This field is not relevant and should remain in its default reset value. See DDR3 SDRAM Specification JESD79-3E (July 2010) , LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 1 cycle. 0x1 2 cycles. 0x2 3 cycles. 0xC7 200 cycles 0x1FE 511 cycles. 0x1FF 512 cycles (JEDEC value for DDR3).
15–9 Reserved	This read-only field is reserved and always has the value 0.
8–6 tRTP	Internal READ command to Precharge command delay (same bank). See DDR3 SDRAM Specification JESD79-3E (July 2010) , LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 1cycle 0x1 2cycles 0x2 3cycles 0x3 4cycles 0x4 5cycles 0x5 6cycles 0x6 7cycles 0x7 8 cycles
5–3 tWTR	Internal WRITE to READ command delay (same bank). See DDR3 SDRAM Specification JESD79-3E (July 2010) , LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 1cycle 0x1 2cycles 0x2 3cycles 0x3 4cycles 0x4 5cycles 0x5 6cycles 0x6 7cycles 0x7 8 cycles
tRRD	Active to Active command period (all banks). See DDR3 SDRAM Specification JESD79-3E (July 2010) , LPDDR2 SDRAM Specification JESD209-2B (February 2010) for a detailed description of this parameter. 0x0 1cycle 0x1 2cycles 0x2 3cycles 0x3 4cycles

Table continues on the next page...

MMDCx_MDCFG2 field descriptions (continued)

Field	Description
	0x4 5cycles 0x5 6cycles 0x6 7cycles 0x7 Reserved

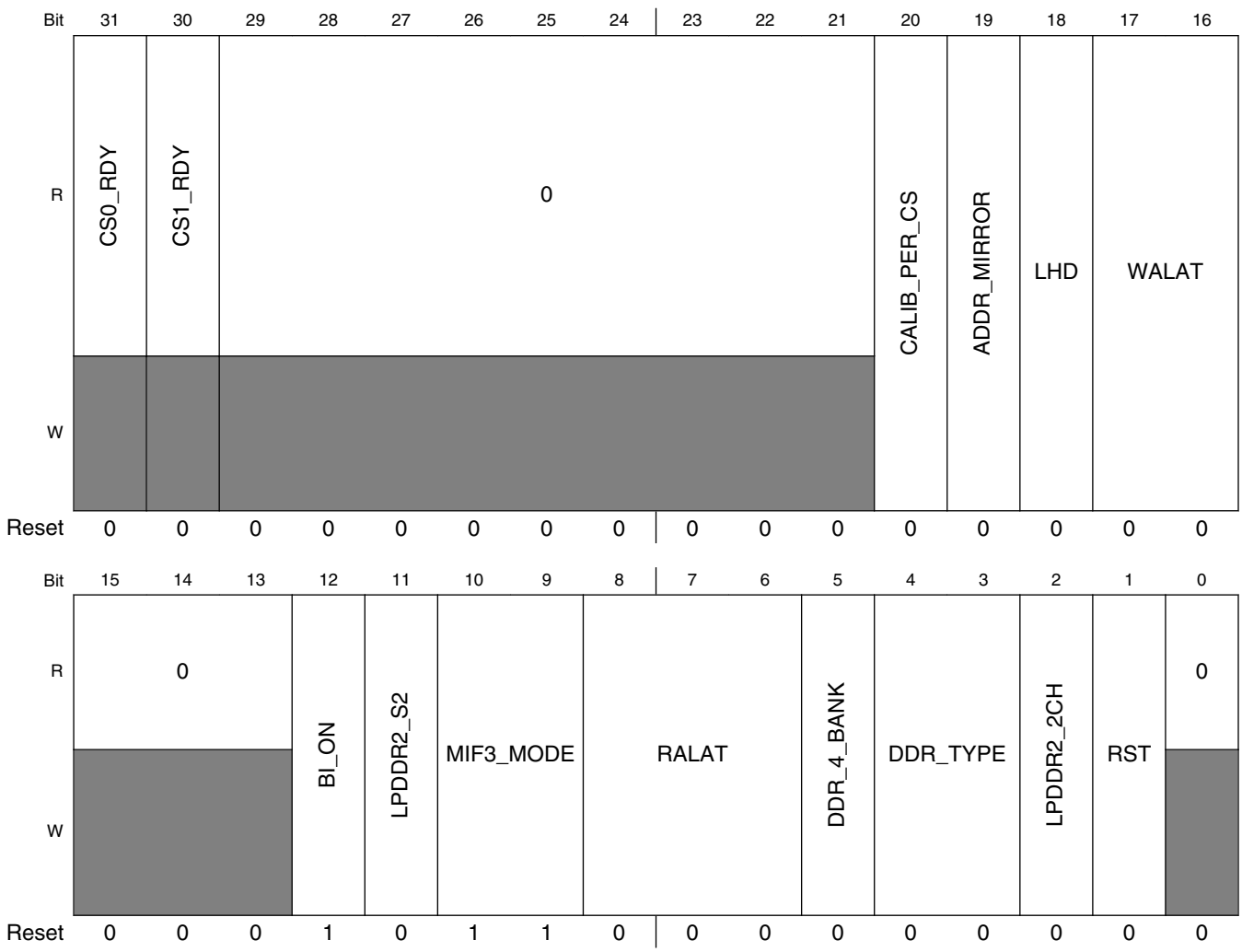
45.12.7 MMDC Core Miscellaneous Register (MMDCx_MDMISC)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 18h offset



MMDCx_MDMISC field descriptions

Field	Description
31 CS0_RDY	External status device on CS0. This is a read-only status bit, that indicates whether the external memory is in wake-up period. 0 Device in wake-up period. 1 Device is ready for initialization.
30 CS1_RDY	External status device on CS1. This is a read-only status bit, that indicates whether the external memory is in wake-up period. 0 Device in wake-up period. 1 Device is ready for initialization.
29–21 Reserved	This read-only field is reserved and always has the value 0.
20 CALIB_PER_CS	Number of chip-select for calibration process. This bit determines the chip-select index that the associated calibration is targeted to. Relevant for read, write, write leveling and read DQS gating calibrations 0 Calibration is targeted to CS0 1 Calibration is targeted to CS1
19 ADDR_MIRROR	Address mirroring. **NOTE:** This feature is not supported for LPDDR2/LPDDR3 memories. But only for DDR2/DDR3 memories. For further information see Address mirroring . 0 Address mirroring disabled. 1 Address mirroring enabled.
18 LHD	Latency hiding disable. This is a debug feature. When set to "1" the MMDC will handle one read/write access at a time. Meaning that the MMDC pipe-line will be limited to 1 open access (next AXI address phase will be acknowledged if the current AXI data phase had finished) 0 Latency hiding on. 1 Latency hiding disable.
17–16 WALAT	Write Additional latency. In case the write-leveling calibration process indicates a delay of greater than one-eighth a clock cycle (between CK and any of the DQS strobe lines), then this field must be configured accordingly. This field will add delay on the strobe I/O control, which will compensate on the additional write leveling delay on DQS and prevent the DQS from being cropped. **NOTE:** The purpose of WALAT is to add time delay at the end of a burst write operation to ensure that the JEDEC time specification for Write Post Amble Delay (tWPST) is met (DQS strobe is held low at the end of a write burst for > 30% a clock cycle before it is released). If the value of any of the WL_DL_ABS_OFFSETn register fields are greater than ‘1F’, WALAT should be set to ‘1’ (cycle additional delay). WALAT should be further increased for any full cycle delays added by the WL_CYC_DELn register fields. 0x0 No additional latency required. 0x1 1 cycle additional delay 0x2 2 cycles additional delay 0x3 3 cycles additional delay

Table continues on the next page...

MMDCx_MDMISC field descriptions (continued)

Field	Description
15–13 Reserved	This read-only field is reserved and always has the value 0.
12 BI_ON	Bank Interleaving On. This bit controls the organization of the bank, row and column address bits. For further information see Address decoding . 0 Banks are not interleaved, and address will be decoded as bank-row-column 1 Banks are interleaved, and address will be decoded as row-bank-column
11 LPDDR2_S2	LPDDR2 S2 device type indication. In case LPDDR2 device is used (DDR_TYPE = 0x1), this bit will indicate whether S2 or S4 device is used. This bit should be cleared in DDR3 mode 0x0 LPDDR2-S4 device is used. 0x1 LPDDR2-S2 device is used.
10–9 MIF3_MODE	Command prediction working mode. This field determines the level of command prediction that will be used by the MMDC 00 Disable prediction. 01 Enable prediction based on : Valid access on first pipe line stage. 10 Enable prediction based on: Valid access on first pipe line stage, Valid access on axi bus. 11 Enable prediction based on: Valid access on first pipe line stage, Valid access on axi bus, Next miss access from access queue.
8–6 RALAT	Read Additional Latency. This field determines the additional read latency which is added to CAS latency and internal delays for which the MMDC will retrieve the read data from the internal FIFO. This field is used to compensate on board/chip delays. **NOTE:** In LPDDR2 mode 2 extra cycles will be added internally in order to compensate tDQSCK delay. 0x0 no additional latency. 0x1 1 cycle additional latency. 0x2 2 cycles additional latency. 0x3 3 cycles additional latency. 0x4 4 cycles additional latency. 0x5 5 cycles additional latency. 0x6 6 cycles additional latency. 0x7 7 cycles additional latency.
5 DDR_4_BANK	Number of banks per DDR device. When this bit is set to "1" then the MMDC will work with DDR device of 4 banks. 0 8 banks device is being used. (Default) 1 4 banks device is being used
4–3 DDR_TYPE	DDR TYPE. This field determines the type of the external DDR device. 0x0 DDR3 device is used. 0x1 LPDDR2 device is used. 0x2 Reserved 0x3 Reserved
2 LPDDR2_2CH	LPDDR2 2-channels mode. When this bit is set to "1" then dual channel mode is activated. This field should be cleared for DDR3 mode.

Table continues on the next page...

MMDCx_MDMISC field descriptions (continued)

Field	Description
	0 1-channel mode (DDR3) 1 2-channels mode (LPDDR2)
1 RST	Software Reset. When this bit is asserted then the internal FSMs and registers of the MMDC will be initialized. **NOTE:** This bit once asserted gets deasserted automatically. 0 Do nothing. 1 Assert reset to the MMDC.
0 Reserved	This read-only field is reserved and always has the value 0.

45.12.8 MMDC Core Special Command Register (MMDCx_MDSCR)

This register is used to issue special commands manually toward the external DDR device (such as load mode register, manual self refresh, manual precharge and so on). Every write to this register will be interpreted as a command, and a read from this register will show the last command that was executed.

Every write to this register will result in one special command, and the IP bus will assert ips_xfr_wait as long as the special command is being carried out.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 1Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CMD_ADDR_MSB_MR_OP								CMD_ADDR_LSB_MR_ADDR							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CON_REQ	CON_ACK	0			MRR_READ_DATA_VALID	WL_EN	0		CMD			CMD_CS	CMD_BA		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MDSCR field descriptions

Field	Description
31–24 CMD_ADDR_ MSB_MR_OP	Command/Address MSB. This field indicates the MSB of the command/Address. In LPDDR2/LPDDR3 this field indicates the MRW operand
23–16 CMD_ADDR_ LSB_MR_ADDR	Command/Address LSB. This field indicates the LSB of the command/Address In LPDDR2/LPDDR3 this field indicates the MRR/MRW address
15 CON_REQ	Configuration request. When this bit is set then the MMDC will clean the pending AXI accesses and will prevent from further AXI accesses to be acknowledged. This field guarantee safe configuration (or change configuration) of the MMDC while no access is in process and prevents an unexpected behaviour. After setting this bit, it is needed to poll on CON_ACK until it is set to "1". When CON_ACK is asserted then configuration is permitted. After configuration is completed then this bit must be deasserted in order to process further AXI accesses. **NOTE:** This bit is asserted at the end of the reset sequence, meaning that the MMDC is waiting to configure and initialize the external memory before accepting any AXI accesses. Configuration request/acknowledge mechanism should be used for the following procedures: changing of timing parameters , during calibration process or driving commands via MDSCR[CMD] 0 No request to configure MMDC. 1 A request to configure MMDC is valid
14 CON_ACK	Configuration acknowledge. Whenever this bit is set, it is permitted to configure MMDC IP registers. 0 Configuration of MMDC registers is forbidden. 1 Configuration of MMDC registers is permitted.
13–11 Reserved	This read-only field is reserved and always has the value 0.
10 MRR_READ_ DATA_VALID	MRR read data valid. This field indicates that read data is valid at MDMRR register **NOTE:** This field is relevant only for LPDDR2/LPDDR3 mode 0 Cleared upon the assertion of MRR command 1 Set after MRR data is valid and stored at MDMRR register.
9 WL_EN	DQS pads direction. This bit controls the DQS pads direction during write-leveling calibration process. Before starting the write-leveling calibration process this bit should be set to "1". It should be set to "0" when sending write leveling exit command. For further information see Write leveling Calibration . 0 Exit write leveling mode or stay in normal mode. 1 Write leveling entry command was sent.
8–7 Reserved	This read-only field is reserved and always has the value 0.
6–4 CMD	Command. This field contains the command to be executed. This field will be automatically cleared after the command will be send to the DDR memory. **NOTE:** Only the 0x5 Precharge All command should be used, memory operation is unpredictable when using the 0x1 command

Table continues on the next page...

MMDCx_MDSCR field descriptions (continued)

Field	Description
	NOTE: A Precharge All command must be issued prior to the MRW command to ensure robust DDR initialization. This command is required to be issued to both chip selects if two chip selects are utilized in the system. 0x0 Normal operation 0x1 Precharge All, command is sent independently of bank status (set correct CMD_CS). Will be issued even if banks are closed. Primarily used for initialization sequence purposes. Not to be used during run-time operation. 0x2 Auto-Refresh Command (set correct CMD_CS). 0x3 Load Mode Register Command DDR2/DDR3, set correct CMD_CS, CMD_BA, CMD_ADDR_LSB, CMD_ADDR_MSB), MRW Command (LPDDR2/LPDDR3, set correct CMD_CS, MR_OP, MR_ADDR) 0x4 ZQ calibration (DDR2/DDR3, set correct CMD_CS, {CMD_ADDR_MSB,CMD_ADDR_LSB} = 0x400 or 0x0) 0x5 Precharge All, only if banks open (set correct CMD_CS). 0x6 MRR command (LPDDR2/LPDDR3, set correct CMD_CS, MR_ADDR) 0x7 Reserved
3 CMD_CS	Chip Select. This field determines which chip select the command is targeted to 0 to Chip-select 0 1 to Chip-select 1
CMD_BA	Bank Address. This field determines the address of the bank within the selected chip-select to where the command is targeted. 0x0 bank address 0 0x1 bank address 1 0x2 bank address 2 0x7 bank address 7

45.12.9 MMDC Core Refresh Control Register (MMDCx_MDREF)

This register determines the refresh scheme that will be executed toward the DDR device. It specifies how often a refresh cycle occurs and how many refresh commands will be executed every refresh cycle.

For further information see Refresh Scheme .

The following tables show examples of possible refresh schemes.

Table 45-17. Refresh rate example for REF_SEL = 0

REFR[2:0]	Number of refresh commands every 64KHz	Average periodic refresh rate (tREFI)	System Refresh period
0x0	1	15.6 µs	tRFC
0x1	2	7.8 µs	2*tRFC

Table continues on the next page...

Table 45-17. Refresh rate example for REF_SEL = 0 (continued)

REFR[2:0]	Number of refresh commands every 64KHz	Average periodic refresh rate (tREFI)	System Refresh period
0x3	4	3.9µs	4*tRFC
0x7	8	1.95 µs	8*tRFC

Table 45-18. Refresh rate example for REF_SEL = 1

REFR[2:0]	Number of refresh commands every 32KHz	Average periodic refresh rate (tREFI)	System Refresh period
0x1	2	15.6 µs	2*tRFC
0x3	4	7.8 µs	4*tRFC
0x7	8	3.9µs	8*tRFC

Table 45-19. Refresh rate example for REF_SEL = 2@ 400MHz

REFR[2:0]	Number of refresh commands every refresh cycle	REF_CNT	Average periodic refresh rate (tREFI)	System Refresh period
0x0	1	0x618	3.9 µs	tRFC
0x1	2	0xC30	3.9 µs	2*tRFC
0x2	3	0x1248	3.9µs	3*tRFC
0x3	4	0x1860	3.9 µs	4*tRFC

Other refresh configurations are also allowed; the configuration values in the tables above are only examples for obtaining the desired average periodic refresh rate.

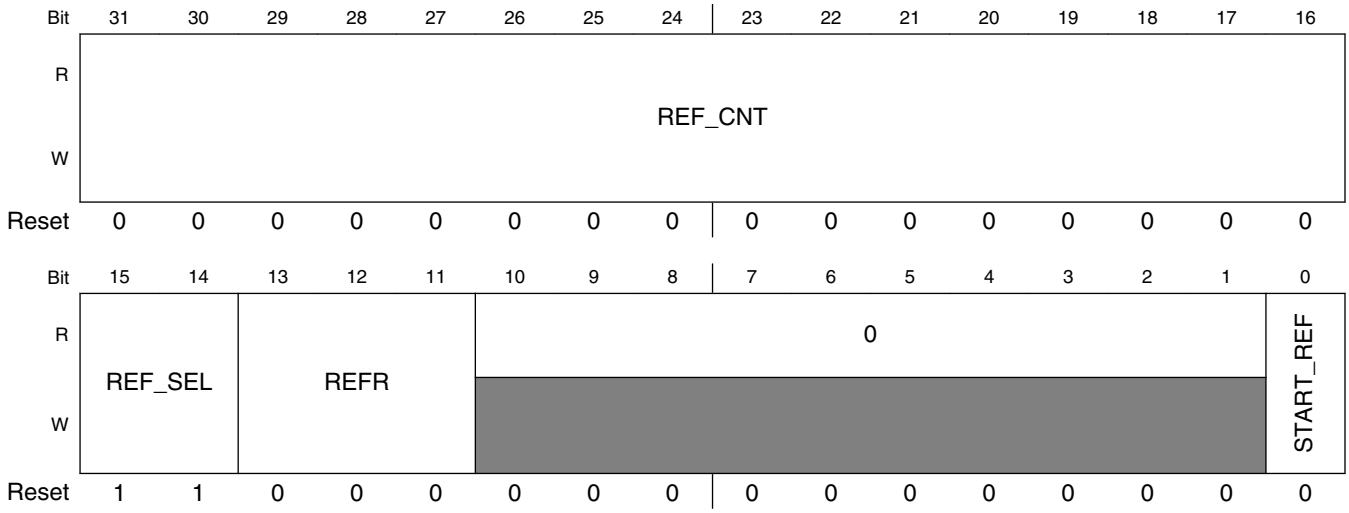
If the required average periodic refresh rate (tREFI) is kept, all of the rows will be refreshed in every refresh window. Because the memory device issues additional refresh commands for every refresh it receives, the tREFI remains the same across the device, regardless of its number of rows. This is particularly relevant in the tRFC parameter, which becomes bigger as the density increases.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 20h offset



MMDCx_MDREF field descriptions

Field	Description
31–16 REF_CNT	Refresh Counter at DDR clock period If REF_SEL equals '2' a refresh cycle will begin every amount of DDR cycles configured in this field. 0x0 Reserved. 0x1 1 cycle. 0xFFFE 65534 cycles. 0xFFFF 65535 cycles.
15–14 REF_SEL	Refresh Selector. This bit selects the source of the clock that will trigger each refresh cycle: 0 Periodic refresh cycles will be triggered in frequency of 64KHz. 1 Periodic refresh cycles will be triggered in frequency of 32KHz. 2 Periodic refresh cycles will be triggered every amount of cycles that are configured in REF_CNT field. 3 No refresh cycles will be triggered.
13–11 REFR	Refresh Rate. This field determines how many refresh commands will be issued every refresh cycle. After every refresh command the MMDC won't drive any command to the DDR device until satisfying tRFC period 0x0 1 refresh 0x1 2 refreshes 0x2 3 refreshes 0x3 4 refreshes 0x4 5 refreshes 0x5 6 refreshes 0x6 7 refreshes 0x7 8 refreshes
10–1 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

MMDCx_MDREF field descriptions (continued)

Field	Description
0 START_REF	Manual start of refresh cycle. When this field is set to '1' the MMDC will start a refresh cycle immediately according to number of refresh commands that are configured in 'REFR' field. This bit returns to zero automatically. 0 Do nothing. 1 Start a refresh cycle.

45.12.10 MMDC Core Read/Write Command Delay Register (MMDCx_MDRWD)

This register determines the delay between back to back read and write accesses. The register reset values are set to the minimum required value.

NOTE

As the default values are set to achieve optimal results, changing them is discouraged.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 2Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			tDAI												
W																
Reset	0	0	0	0	1	1	1	1	1	0	0	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	RTW_SAME			WTR_DIFF			WTW_DIFF			RTW_DIFF			RTR_DIFF		
W																
Reset	0	0	1	0	0	1	1	0	1	1	0	1	0	0	1	0

MMDCx_MDRWD field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–16 tDAI	Device auto initialization period.(maximum) **NOTE:** This field is relevant only to LPDDR2 mode 0x0 1 cycle

Table continues on the next page...

MMDCx_MDRWD field descriptions (continued)

Field	Description
	0xF9F 4000 cycles (Default, JEDEC value for LPDDR2, gives 10us at 400MHz clock). 0x1FFF 8192 cycles
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 RTW_SAME	Read to write delay for the same chip-select. This field controls the delay between read to write commands toward the same chip select. The total delay is calculated according to: BL/2 + RTW_SAME + (tCL-tCWL) + RALAT 0x0 0 cycle 0x1 1 cycle 0x2 2 cycles (Default) 0x3 3 cycles 0x4 4 cycles 0x5 5 cycles 0x6 6 cycles 0x7 7 cycles
11–9 WTR_DIFF	Write to read delay for different chip-select. This field controls the delay between write to read commands toward different chip select. The total delay is calculated according to: BL/2 + WTR_DIFF + (tCL-tCWL) + RALAT 0x0 0 cycle 0x1 1 cycle 0x2 2 cycles 0x3 3 cycles (Default) 0x4 4 cycles 0x5 5 cycles 0x6 6 cycles 0x7 7 cycles
8–6 WTW_DIFF	Write to write delay for different chip-select. This field controls the delay between write to write commands toward different chip select. The total delay is calculated according to: BL/2 + WTW_DIFF 0x0 0 cycle 0x1 1 cycle 0x2 2 cycles 0x3 3 cycles (Default) 0x4 4 cycles 0x5 5 cycles 0x6 6 cycles 0x7 7 cycles
5–3 RTW_DIFF	Read to write delay for different chip-select. This field controls the delay between read to write commands toward different chip select. The total delay is calculated according to: BL/2 + RTW_DIFF + (tCL - tCWL) + RALAT 0x0 0 cycle 0x1 1 cycle 0x2 2 cycles (Default)

Table continues on the next page...

MMDCx_MDRWD field descriptions (continued)

Field	Description
	0x3 3 cycles 0x4 4 cycles 0x5 5 cycles 0x6 6 cycles 0x7 7 cycles
RTR_DIFF	Read to read delay for different chip-select. This field controls the delay between read to read commands toward different chip select. The total delay is calculated according to: BL/2 + RTR_DIFF 0x0 0 cycle 0x1 1 cycle 0x2 2 cycles (Default) 0x3 3 cycles 0x4 4 cycles 0x5 5 cycles 0x6 6 cycles 0x7 7 cycles

45.12.11 MMDC Core Out of Reset Delays Register (MMDCx_MDOR)

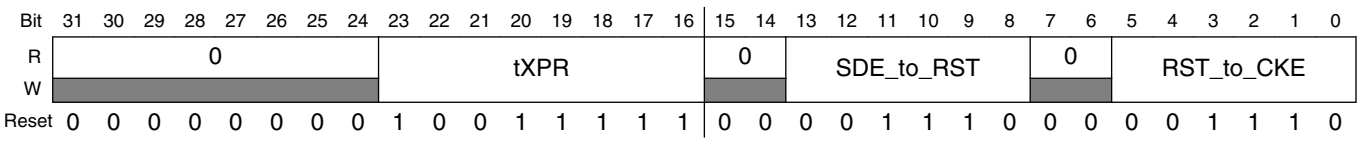
This register defines delays that must be kept when MMDC exits reset.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 30h offset



MMDCx_MDOR field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value 0.
23–16 tXPR	LPDDR2: Not relevant to this mode and should remain in default reset value. DDR3: As defined in timing parameter table. 0x0 Reserved 0x1 2 cycles

Table continues on the next page...

MMDCx_MDOR field descriptions (continued)

Field	Description
	0x2 3 cycles 0xFE 255 cycles 0xFF 256 cycles
15–14 Reserved	This read-only field is reserved and always has the value 0.
13–8 SDE_to_RST	DDR3 mode: Time from SDE enable until DDR reset# is high. LPDDR2 mode: This field is not relevant and should remain in its default reset value. **NOTE:** Each cycle in this field is 15.258 us. 0x0 Reserved 0x1 Reserved 0x2 Reserved 0x3 1 cycles 0x4 2 cycles 0x10 14 cycles (JEDEC value for DDR3) - total of 200 us 0x3E 60 cycles 0x3F 61 cycles
7–6 Reserved	This read-only field is reserved and always has the value 0.
RST_to_CKE	DDR3: Time from SDE enable to CKE rise. In case that DDR reset# is low, will wait until it's high and then wait this period until rising CKE. (JEDEC value is 500 us) LPDDR2: Idle time after first CKE assertion (JEDEC value is 200 us). **NOTE:** Each cycle in this field is 15.258 us. 0x0 Reserved 0x1 Reserved 0x2 Reserved 0x3 1 cycles 0x10 14 cycles (JEDEC value for LPDDR2) - total of 200 us 0x23 33 cycles (JEDEC value for DDR3) - total of 500 us 0x3E 60 cycles 0x3F 61 cycles

45.12.12 MMDC Core MRR Data Register (MMDCx_MDMRR)

This register contains data that was collected after issuing MRR command. The data in this register is valid only when MDSCR[MRR_READ_DATA_VALID] is set to "1".

This register is relevant only in LPDDR2 mode. For further information see LPDDR2 Refresh Rate Update and Timing Derating.

Supported Mode Of Operations:

For Channel 0: LP2_2ch_x16, LP2_2ch_x32

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 34h offset

Bit	31–24	23–16	15–8	7–0
R	MRR_READ_DATA3	MRR_READ_DATA2	MRR_READ_DATA1	MRR_READ_DATA0
W				
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

MMDCx_MDMRR field descriptions

Field	Description
31–24 MRR_READ_DATA3	MRR DATA that arrived on DQ[31:24]
23–16 MRR_READ_DATA2	MRR DATA that arrived on DQ[23:16]
15–8 MRR_READ_DATA1	MRR DATA that arrived on DQ[15:8]
MRR_READ_DATA0	MRR DATA that arrived on DQ[7:0]

45.12.13 MMDC Core Timing Configuration Register 3 (MMDCx_MDCFG3LP)

This register is relevant only for LPDDR2 mode.

Supported Mode Of Operations:

For Channel 0: LP2_2ch_x16, LP2_2ch_x32

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 38h offset

Bit	31–22	21–16	15–12	11–8	7–4	3–0
R	0	RC_LP	0	tRCD_LP	tRPpb_LP	tRPab_LP
W						
Reset	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

MMDCx_MDCFG3LP field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value 0.
21–16 RC_LP	Active to Active or Refresh command period (same bank). This field is valid only for LPDDR2 memories

Table continues on the next page...

MMDCx_MDCFG3LP field descriptions (continued)

Field	Description
	0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0x3E 63 clocks 0x3F Reserved
15–12 Reserved	This read-only field is reserved and always has the value 0.
11–8 tRCD_LP	Active command to internal read or write delay time (same bank). This field is valid only for LPDDR2 memories 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xE 15 clocks 0xF Reserved
7–4 tRPpb_LP	Precharge (per bank) command period (same bank). This field is valid only for LPDDR2 memories 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xE 15 clocks 0xF Reserved
tRPab_LP	Precharge (all banks) command period. This field is valid only for LPDDR2 memories 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xE 15 clocks 0xF Reserved

45.12.14 MMDC Core MR4 Derating Register (MMDCx_MDMR4)

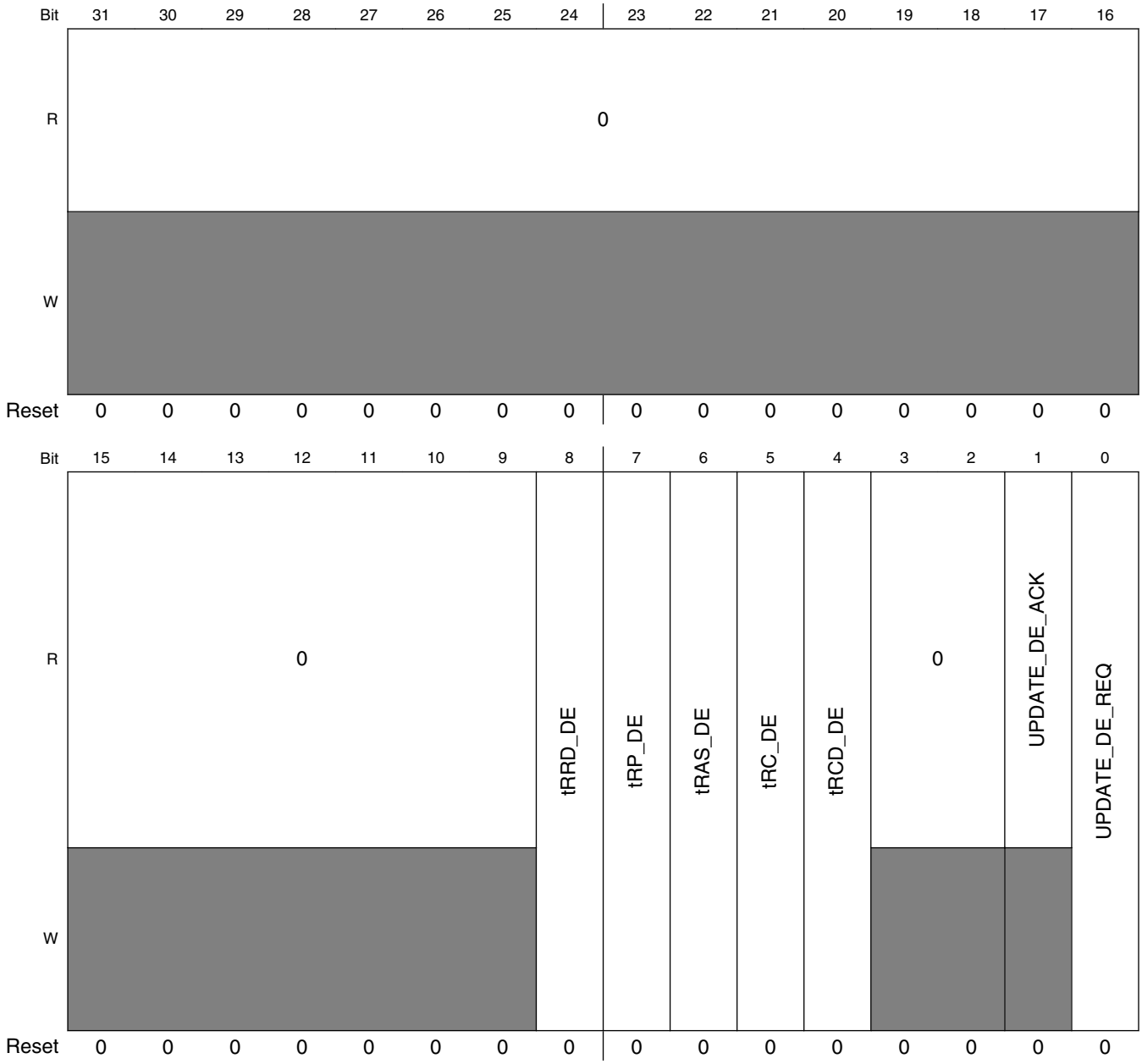
This register is relevant only for LPDDR2 mode. It is used to dynamically change certain values depending on MR4 read result, which is based on memory temperature sensor result.

Supported Mode of Operations:

For Channel 0: LP2_2ch_x16, LP2_2ch_x32

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 3Ch offset



MMDCx_MDMR4 field descriptions

Field	Description
31–9 Reserved	This read-only field is reserved and always has the value 0.
8 tRRD_DE	tRRD derating value. 0 Original tRRD is used. 1 tRRD is derated in 1 cycle.
7 tRP_DE	tRP derating value.

Table continues on the next page...

MMDCx_MDMR4 field descriptions (continued)

Field	Description
	0 Original tRP is used. 1 tRP is derated in 1 cycle.
6 tRAS_DE	tRAS derating value. 0 Original tRAS is used. 1 tRAS is derated in 1 cycle.
5 tRC_DE	tRC derating value. 0 Original tRC is used. 1 tRC is derated in 1 cycle.
4 tRCD_DE	tRCD derating value. 0 Original tRCD is used. 1 tRCD is derated in 1 cycle.
3–2 Reserved	This read-only field is reserved and always has the value 0.
1 UPDATE_DE_ACK	Update Derated Values Acknowledge. This read only bit will be cleared upon UPDATE_DE_REQ assertion and will be set after the new values are taken.
0 UPDATE_DE_REQ	Update Derated Values Request. This read modify write field is automatically cleared after the request is issued. 0 Do nothing. 1 Request to update the following values: tRRD, tRCD, tRP, tRC, tRAS and refresh related fields(MDREF register): REF_CNT, REF_SEL, REFR

45.12.15 MMDC Core Address Space Partition Register (MMDCx_MDASP)

This register defines the partitioning between chip select 0 and chip select 1. For further information see Chip select settings .

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 40h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																CS0_END															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

MMDCx_MDASP field descriptions

Field	Description
31–7 Reserved	This read-only field is reserved and always has the value 0.
CS0_END	Defines the absolute last address associated with CS0 with increments of 256Mb. CS0_END=AXI_ADDRESS[31:25] bits. In DDR3 and 1-channel LPDDR2 mode: MMDCx_MDASP[CS0_END] should be set to DDR_CS_SIZE/32MB + 0x7 (DDR base address begins at 0x10000000) In 2-channel LPDDR2 mode: MMDC0_MDASP[CS0_END] should be set to DDR_CS_SIZE/32M + 0x3f (channel 0 base address begins at 0x80000000) MMDC1_MDASP[CS0_END] should be set to DDR_CS_SIZE/32M + 0x7 (channel 1 base address begins at 0x10000000) In 2-channel LPDDR2 with 4k-interleave mode: MMDC0_MDASP[CS0_END] should be set to DDR_CS_SIZE/32M + 0x43 MMDC1_MDASP[CS0_END] should be set to DDR_CS_SIZE/32M + 0x3

45.12.16 MMDC Core AXI Reordering Control Register (MMDCx_MAARCR)

This register determines the values of the weights used for the re-ordering arbitration engine. For further information see Performance.

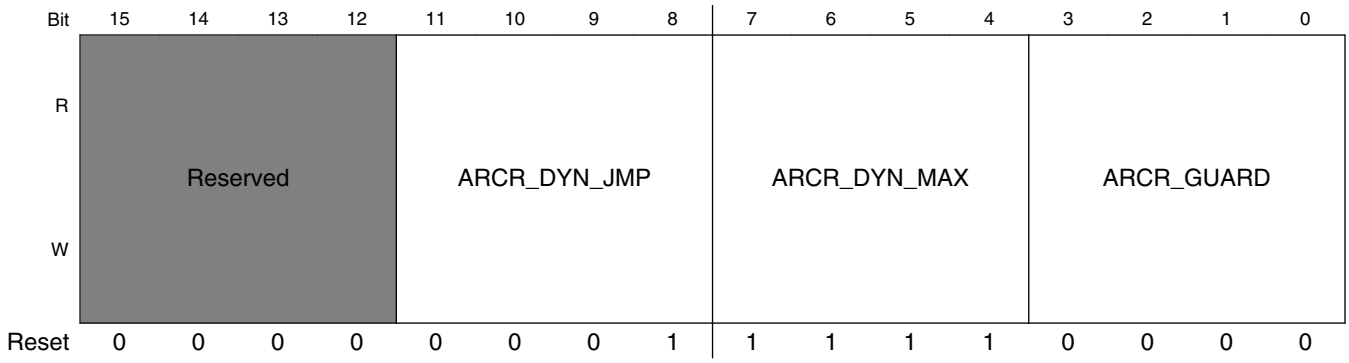
Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 400h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ARCR_SEC_ERR_LOCK	ARCR_SEC_ERR_EN	Reserved	ARCR_EXC_ERR_EN	Reserved			ARCR_RCH_EN	Reserved	ARCR_PAG_HIT			Reserved	ARCR_ACC_HIT		
Reset	0	1	0	1	0	0	0	1	0	1	0	0	0	0	1	0



MMDCx_MAARCR field descriptions

Field	Description
31 ARCR_SEC_ERR_LOCK	Once set, this bit locks ARCR_SEC_ERR_EN and prevents from its updating. This bit can be only cleared by reset Default value is 0x0 - encoding 0 (unlocked) 0 ARCR_SEC_ERR_EN is unlocked, so can be updated any moment 1 ARCR_SEC_ERR_EN is locked, so it can't be updated
30 ARCR_SEC_ERR_EN	This bit defines whether security read/write access violation result in SLV Error response or in OKAY response Default value is 0x1 - encoding 1(response is SLV Error, rresp/bresp=2'b10) 0 security violation results in OKAY response (rresp/bresp=2'b00) 1 security violation results in SLAVE Error response (rresp/bresp=2'b10)
29 -	This field is reserved. Reserved
28 ARCR_EXC_ERR_EN	This bit defines whether exclusive read/write access violation of AXI 6.2.4 rule result in SLV Error response or in OKAY response Default value is 0x1 - encoding 1(response is SLV Error) 0 violation of AXI exclusive rules (6.2.4) result in OKAY response (rresp/bresp=2'b00) 1 violation of AXI exclusive rules (6.2.4) result in SLAVE Error response (rresp/bresp=2'b10)
27–25 -	This field is reserved. Reserved
24 ARCR_RCH_EN	This bit defines whether Real time channel is activated and bypassed all other pending accesses, So accesses with QoS=='F' will be granted the highest priority in the optimization/reordering mechanism Default value is 0x1 - encoding 1 (Enabled) 0 normal prioritization, no bypassing 1 accesses with QoS=='F' bypass the arbitration
23 -	This field is reserved. Reserved
22–20 ARCR_PAG_HIT	ARCR Page Hit Rate. This value will be added by the optimization/reordering mechanism to any pending access that is targeted to an open DDR row. Default value of ARCR_PAG_HIT is 0x00100 - encoding 4.
19 -	This field is reserved. Reserved

Table continues on the next page...

MMDCx_MAARCR field descriptions (continued)

Field	Description
18–16 ARCR_ACC_HIT	ARCR Access Hit Rate. This value will be added by the optimization/reordering mechanism to any pending access that has the same access type (read/write) as the previous access. Default value of is ARCR_ACC_HIT 0x0010 - encoding 2.
15–12 -	This field is reserved. Reserved
11–8 ARCR_DYN_JMP	ARCR Dynamic Jump. Each time an access is not chosen by the optimization/reordering mechanism then its dynamic score will be incremented by ARCR_DYN_JMP value. **NOTE:** Setting ARCR_DYN_JMP may cause starvation of low priority accesses **NOTE:** ARCR_DYN_JMP must be smaller than ARCR_DYN_MAX Default ARCR_DYN_JMP value is 0x0001 - encoding 1
7–4 ARCR_DYN_MAX	ARCR Dynamic Maximum. ARCR_DYN_MAX is the maximum dynamic score value that each access inside the optimization/reordering mechanism can get. 0000 0 0001 1 1111 15 (default)
ARCR_GUARD	ARCR Guard. After an access reached the maximum dynamic score value, it will wait additional ARCR_GUARD arbitration cycles and then will gain the highest priority in the optimization/reordering mechanism. 0000 15 (default) 0001 16 1111 30

45.12.17 MMDC Core Power Saving Control and Status Register (MMDCx_MAPSR)

The MAPSR determines the power saving features of MMDC. For further information see Power Saving and Clock Frequency Change modes .

Supported Mode Of Operations:

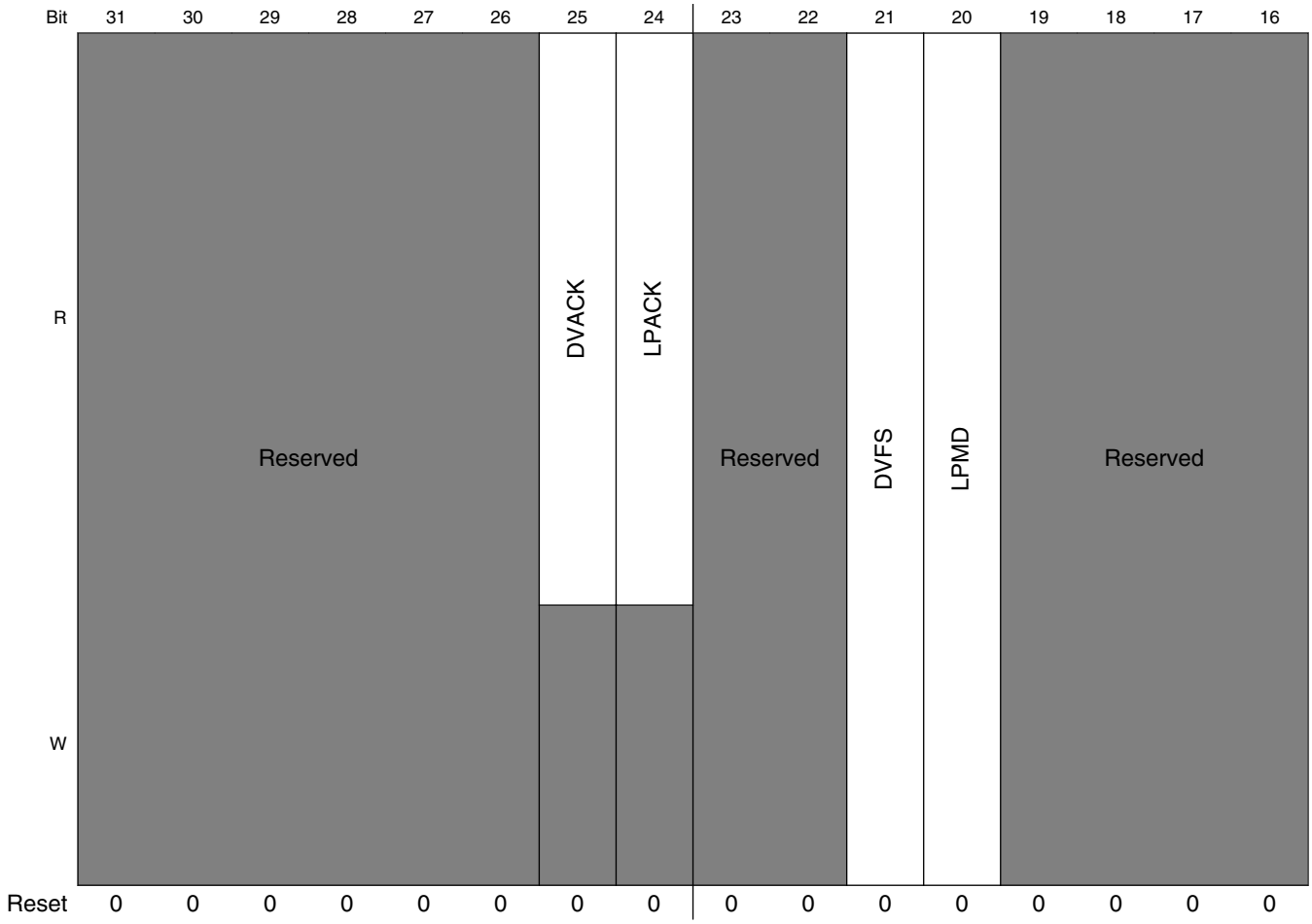
For Channel 0: All

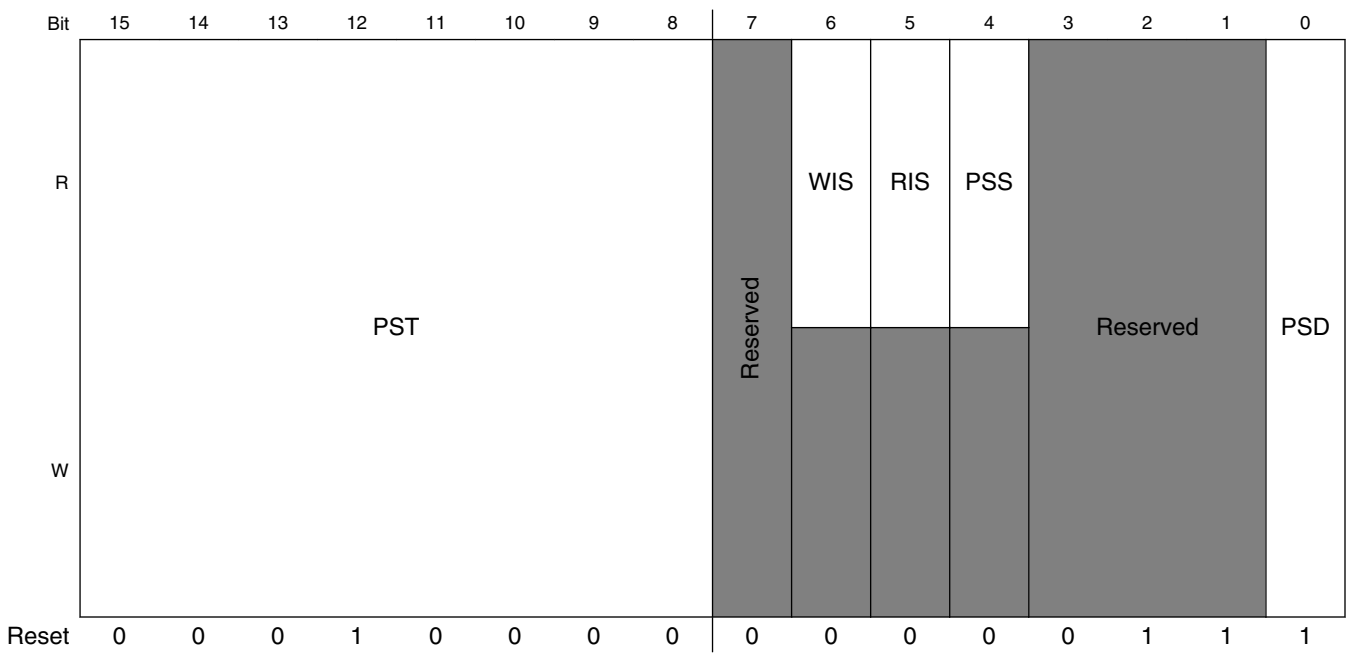
For Channel 1: LP2_2ch_x16, LP2_2ch_x32

NOTE

See chip-specific information for implementation on your device.

Address: Base address + 404h offset





MMDCx_MAPSR field descriptions

Field	Description
31–26 -	This field is reserved. Reserved
25 DVACK	DVFS/Self-Refresh acknowledge. This read only bit indicates whether a DVFS/self-refresh acknowledge was asserted and that the MMDC is in self-refresh mode.
24 LPACK	General low-power acknowledge. This read only bit indicates whether a low-power acknowledge was asserted and that MMDC is in self-refresh mode
23–22 -	This field is reserved. Reserved
21 DVFS	DVFS/Self-Refresh request. Software request for DVFS/self-refresh. Assertion of this bit will initiate a self-refresh entry sequence. 0 no DVFS/Self-Refresh entry request 1 DVFS/Self-Refresh entry request
20 LPMD	General LPMD request. Software request for LPMD. Assertion of this bit will yield in self-refresh entry sequence 0 no lpmd request 1 lpmd request
19–16 -	This field is reserved. Reserved
15–8 PST	Automatic Power saving timer. Valid only when PSD is set to "0". When the MMDC is idle for amount of cycles specified in that field then the DDR device will be entered automatically into self-refresh mode. The real value which is used is register-value multiplied by 64. 00000000 Reserved - this value is forbidden. 00000001 timer is configured to 64 clock cycles. 00000010 timer is configured to 128 clock cycles. 00010000 (Default)- 1024 clock cycles. 11111111 timer clock is configured to 16320 clock cycles.
7 -	This field is reserved. Reserved.
6 WIS	Write Idle Status. This read only bit indicates whether write request buffer is idle (empty) or not. 0 not idle 1 idle
5 RIS	Read Idle Status. This read only bit indicates whether read request buffer is idle (empty) or not. 0 not idle 1 idle
4 PSS	Power Saving Status. This read only bit indicates whether the MMDC is in automatic power saving mode. 0 not in power saving mode 1 power saving mode
3–1 -	This field is reserved. Reserved.

Table continues on the next page...

MMDCx_MAPSR field descriptions (continued)

Field	Description
0 PSD	Automatic Power Saving Disable. When the value of PSD is "0" (i.e. automatic power saving is enabled) then the PST is activated and MMDC will enter automatically to self-refresh while the number of idle cycle reached. **NOTE:** This bit must be disabled (i.e. set to "1") during calibration process 0 power saving enabled 1 power saving disabled (default)

45.12.18 MMDC Core Exclusive ID Monitor Register0 (MMDCx_MAEXIDR0)

This register defines the ID to be monitored for exclusive accesses of monitor0 and monitor1. For further information see Exclusive accesses handling .

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 408h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EXC_ID_MONITOR1																EXC_ID_MONITOR0															
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MAEXIDR0 field descriptions

Field	Description
31–16 EXC_ID_ MONITOR1	This field defines ID for Exclusive monitor#1. Default value is 0x0020
EXC_ID_ MONITOR0	This field defines ID for Exclusive monitor#0. Default value is 0x0000

45.12.19 MMDC Core Exclusive ID Monitor Register1 (MMDCx_MAEXIDR1)

This register defines the ID to be monitored for exclusive accesses of monitor2 and monitor3. For further information see Exclusive accesses handling .

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 40Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EXC_ID_MONITOR3																EXC_ID_MONITOR2															
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

MMDCx_MAEXIDR1 field descriptions

Field	Description
31–16 EXC_ID_ MONITOR3	This field defines ID for Exclusive monitor#3. Default value is 0x0060
EXC_ID_ MONITOR2	This field defines ID for Exclusive monitor#2. Default value is 0x0040

45.12.20 MMDC Core Debug and Profiling Control Register 0 (MMDCx_MADPCR0)

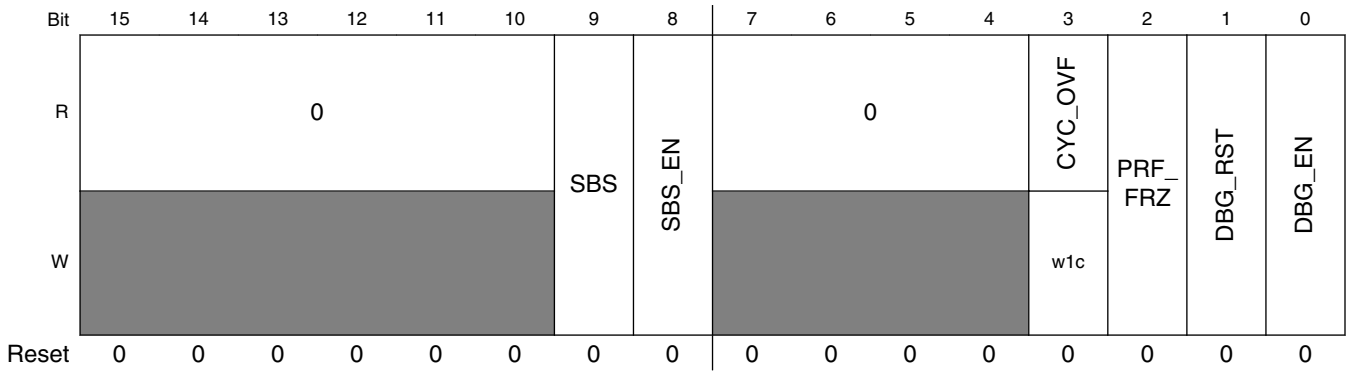
Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 410h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



MMDCx_MADPCR0 field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value 0.
9 SBS	Step By Step trigger. If SBS_EN is set to "1" then dispatching AXI pending access toward the DDR will done only if this bit is set to "1", otherwise no access will be dispatched toward the DDR. This bit is cleared when the pending access has been issued toward the DDR device. 1 Launch AXI pending access toward the DDR 0 No access will be launched toward the DDR
8 SBS_EN	Step By Step debug Enable. Enable step by step mode. Every time this mechanism is enabled then setting SBS to "1" will dispatch one pending AXI access to the DDR and in parallel its attributes will be observed in the status registers (MASBS0 and MASBS1). For further information see Step By Step (SBS) software monitor . 0 disable 1 enable
7–4 Reserved	This read-only field is reserved and always has the value 0.
3 CYC_OVF	Total Profiling Cycles Count Overflow. When profiling mechanism is enabled (DBG_EN is set to "1") then this bit is asserted when overflow of CYC_COUNT occurred. Cleared by writing 1 to it. 0 no overflow 1 overflow
2 PRF_FRZ	Profiling freeze. When this bit is asserted then the profiling mechanism will be froze and the associated status registers (MADPSR0-MADPSR5) will hold the current profiling values. 0 profiling counters are not frozen 1 profiling counters are frozen
1 DBG_RST	Debug and Profiling Reset. Reset all debug and profiling counters and components. 0 no reset 1 reset
0 DBG_EN	Debug and Profiling Enable. Enable debug and profiling mechanism. When this bit is asserted then the MMDC will perform a profiling based on the ID that is configured to MADPCR1. Upon assertion of PRF_FRZ the profiling will be froze and the profiling results will be sampled to the status registers (MADPSR0-MADPSR5). For further information see MMDC Profiling. 0 disable (default) 1 enable

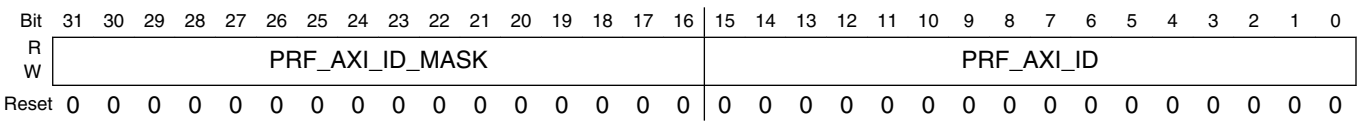
45.12.21 MMDC Core Debug and Profiling Control Register 1 (MMDCx_MADPCR1)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 414h offset



MMDCx_MADPCR1 field descriptions

Field	Description
31–16 PRF_AXI_ID_ MASK	Profiling AXI ID Mask. AXI ID bits which masked by this value are chosen for profiling. 1 AXI ID specific bit is chosen for profiling 0 AXI ID specific bit is ignored (don't care)
PRF_AXI_ID	Profiling AXI ID. AXI IDs that match a bit-wise AND logic operation between PRF_AXI_ID and PRF_AXI_ID_MASK are chosen for profiling. Default value is 0x0, to choose any ID-s for profiling

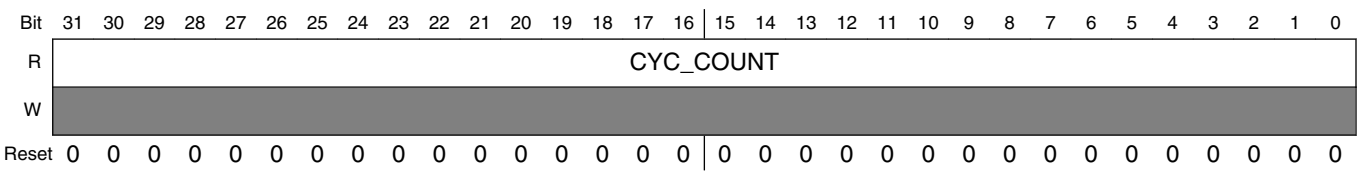
45.12.22 MMDC Core Debug and Profiling Status Register 0 (MMDCx_MADPSR0)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 418h offset



MMDCx_MADPSR0 field descriptions

Field	Description
CYC_COUNT	Total Profiling cycle Count. This field reflects the total cycle count in case the profiling mechanism is enabled from assertion of DBG_EN and until PRF_FRZ is asserted

45.12.23 MMDC Core Debug and Profiling Status Register 1 (MMDCx_MADPSR1)

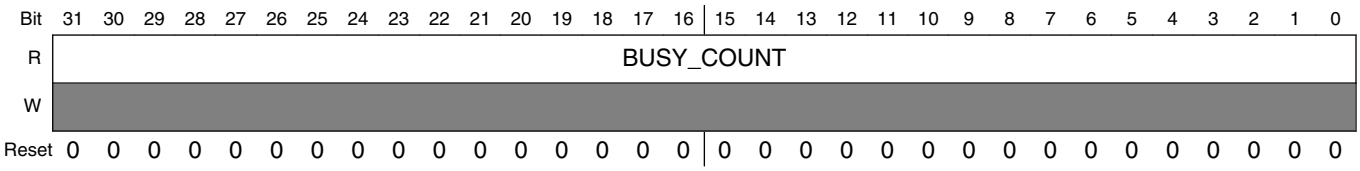
The register reflects the total cycles during which the MMDC state machines were busy (both writes and reads). This information can be used for DDR Utilization calculation.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 41Ch offset



MMDCx_MADPSR1 field descriptions

Field	Description
BUSY_COUNT	Profiling Busy Cycles Count. This field reflects the total number of cycles where the MMDC read and write state machines were busy during the profiling period. Can be used for DDR utilization calculations. Busy cycles are any MMDC clock cycles where the internal state machine is not idle. If any read or write requests are pending in the FIFOs, the MMDC is not idle.

45.12.24 MMDC Core Debug and Profiling Status Register 2 (MMDCx_MADPSR2)

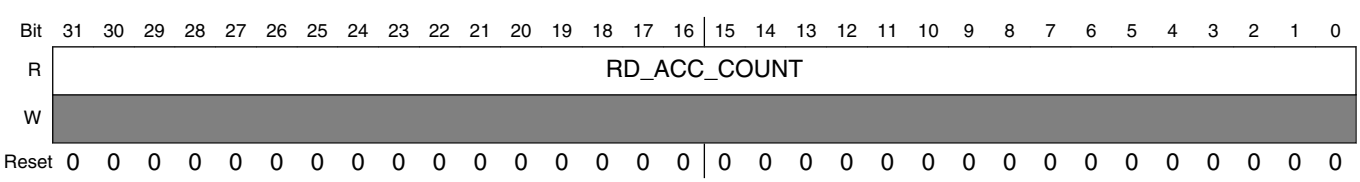
This register reflects the total number of read accesses (per AXI ID) toward MMDC.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 420h offset



MMDCx_MADPSR2 field descriptions

Field	Description
RD_ACC_ COUNT	Profiling Read Access Count. This register reflects the total number of read accesses (per AXI ID) toward MMDC.

45.12.25 MMDC Core Debug and Profiling Status Register 3 (MMDCx_MADPSR3)

This register reflects the total number of write accesses (per AXI ID) toward MMDC.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 424h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WR_ACC_COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MADPSR3 field descriptions

Field	Description
WR_ACC_ COUNT	Profiling Write Access Count. This register reflects the total number of write accesses (per AXI ID) toward MMDC.

45.12.26 MMDC Core Debug and Profiling Status Register 4 (MMDCx_MADPSR4)

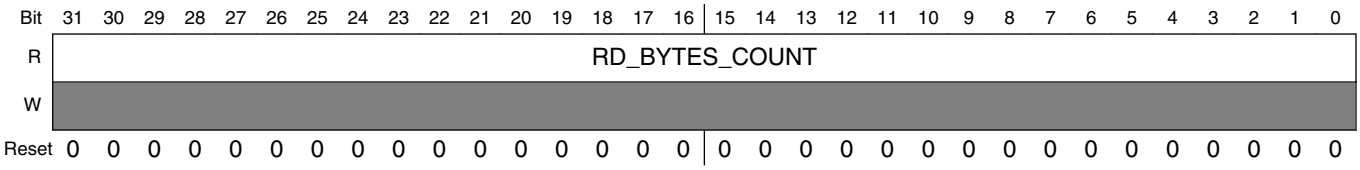
This register reflects the total number of bytes that were transferred during read access (per AXI ID) toward MMDC.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 428h offset



MMDCx_MADPSR4 field descriptions

Field	Description
RD_BYTES_ COUNT	Profiling Read Bytes Count. This register reflects the total number of bytes that were transferred during read access (per AXI ID) toward MMDC.

45.12.27 MMDC Core Debug and Profiling Status Register 5 (MMDCx_MADPSR5)

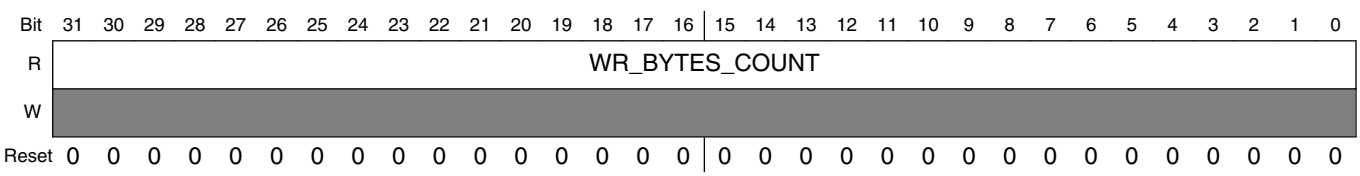
This register reflects the total number of bytes that were transferred during write access (per AXI ID) toward MMDC.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 42Ch offset



MMDCx_MADPSR5 field descriptions

Field	Description
WR_BYTES_ COUNT	Profiling Write Bytes Count. This register reflects the total number of bytes that were transferred during write access (per AXI ID) toward MMDC.

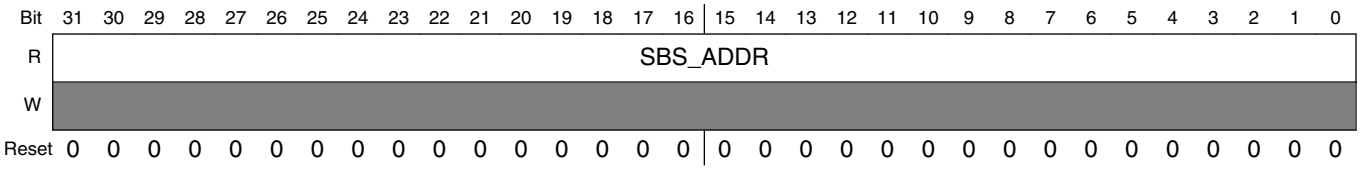
45.12.28 MMDC Core Step By Step Address Register (MMDCx_MASBS0)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 430h offset



MMDCx_MASBS0 field descriptions

Field	Description
SBS_ADDR	Step By Step Address. These bits reflect the address of the pending request in case of step by step mode.

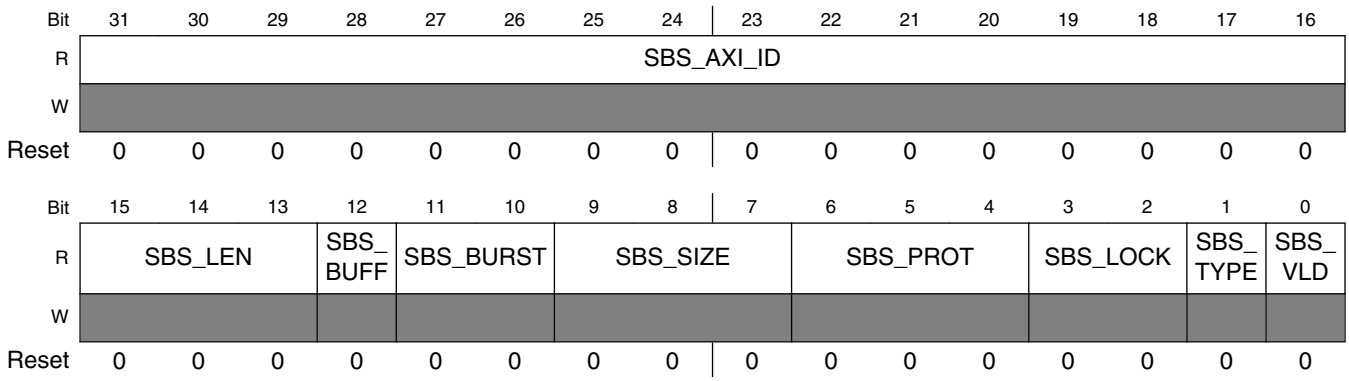
45.12.29 MMDC Core Step By Step Address Attributes Register (MMDCx_MASBS1)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 434h offset



MMDCx_MASBS1 field descriptions

Field	Description
31–16 SBS_AXI_ID	Step By Step AXI ID. These bits reflect the AXI ID of the pending request in case of step by step mode.
15–13 SBS_LEN	Step By Step Length. These bits reflect the AXI LENGTH of the pending request in case of step by step mode. 000 burst of length 1 001 burst of length 2 111 burst of length 8
12 SBS_BUFF	Step By Step Buffered. This bit reflect the AXI CACHE[0] of the pending request in case of step by step mode. Relevant only for write requests

Table continues on the next page...

MMDCx_MASBS1 field descriptions (continued)

Field	Description
11–10 SBS_BURST	Step By Step Burst. These bits reflect the AXI BURST of the pending request in case of step by step mode. 00 FIXED 01 INCR burst 10 WRAP burst 11 reserved
9–7 SBS_SIZE	Step By Step Size. These bits reflect the AXI SIZE of the pending request in case of step by step mode. 000 8 bits 001 16 bits 010 32 bits 011 64 bits 100 128bits 101-111 Reserved
6–4 SBS_PROT	Step By Step Protection. These bits reflect the AXI PROT of the pending request in case of step by step mode.
3–2 SBS_LOCK	Step By Step Lock. These bits reflect the AXI LOCK of the pending request in case of step by step mode.
1 SBS_TYPE	Step By Step Request Type. These bits reflect the type (read/write) of the pending request in case of step by step mode. 0 write 1 read
0 SBS_VLD	Step By Step Valid. This bit reflects whether there is a pending request in case of step by step mode. 0 not valid 1 valid

45.12.30 MMDC Core General Purpose Register (MMDCx_MAGENP)

This register is a general 32 bit read/write register.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 440h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GP31_GP0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MAGENP field descriptions

Field	Description
GP31_GP0	General purpose read/write bits.

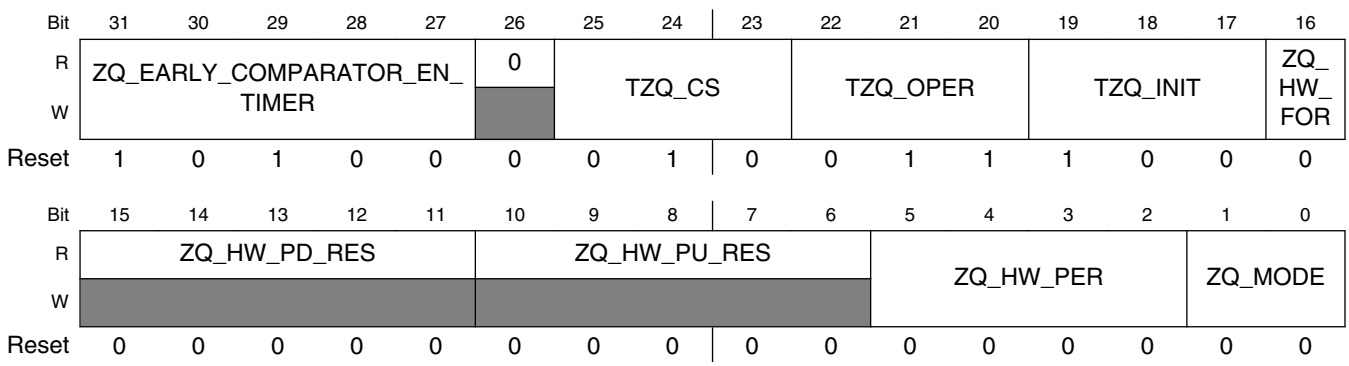
45.12.31 MMDC PHY ZQ HW control register (MMDCx_MPZQHWCTRL)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LPDDR2 - When using LPDDR2, ZQ calibration must be applied to all channels in use.

Address: Base address + 800h offset



MMDCx_MPZQHWCTRL field descriptions

Field	Description
31–27 ZQ_EARLY_COMPARATOR_EN_TIMER	ZQ early comparator enable timer. This timer defines the interval between the warming up of the comparator of the i.MX ZQ calibration pad and the beginning of the ZQ calibration process with the pad 0x0 - 0x6 Reserved 0x7 8 cycles 0x14 21 cycles (Default) 0x1E 31 cycles 0x1F 32 cycles
26 Reserved	This read-only field is reserved and always has the value 0.
25–23 TZQ_CS	Device ZQ short time. This field holds the number of cycles that are required by the external DDR device to perform ZQ short calibration. Upon driving the command to the DDR device then no further accesses will be issued to the DDR device till satisfying that time. **NOTE:** In LPDDR2 the ZQ short time is taken from MPZQLP2CTL[ZQ_LP2_HW_ZQCS] **NOTE:** This field should not be update during ZQ calibration. 000 Reserved

Table continues on the next page...

MMDCx_MPZQHWCTRL field descriptions (continued)

Field	Description
	001 Reserved 010 128 cycles (Default) 011 256 cycles 100 512 cycles 101 1024 cycles 110- 111 Reserved
22–20 TZQ_OPER	Device ZQ long/oper time. This field holds the number of cycles that are required by the external DDR device to perform ZQ long calibration except the first ZQ long command that is issued after reset. Upon driving the command to the DDR device then no further accesses will be issued to the DDR device till satisfying that time. **NOTE:** In LPDDR2 the ZQ oper time is taken from MPZQLP2CTL[ZQ_LP2_HW_ZQCL] **NOTE:** This field should not be update during ZQ calibration. 000 Reserved 001 Reserved 010 128 cycles 011 256 cycles - Default (JEDEC value for DDR3) 100 512 cycles 101 1024 cycles 110- 111 Reserved
19–17 TZQ_INIT	Device ZQ long/init time. This field holds the number of cycles that are required by the external DDR device to perform ZQ long calibration right after reset. Upon driving the command to the DDR device then no further accesses will be issued to the DDR device till satisfying that time. **NOTE:** In LPDDR2 the ZQ init time is taken from MPZQLP2CTL[ZQ_LP2_HW_ZQINIT] **NOTE:** This field should not be update during ZQ calibration. 000 Reserved 001 Reserved 010 128 cycles 011 256 cycles 100 512 cycles - Default (JEDEC value for DDR3) 101 1024 cycles 110- 111 Reserved
16 ZQ_HW_FOR	Force ZQ automatic calibration process with the i.MX ZQ calibration pad. When this bit is asserted then the MMDC will issue one ZQ automatic calibration process with the i.MX ZQ calibration pad. It is the user responsibility to make sure that all the accesses to DDR will be finished before asserting this bit using CON_REQ/CON_ACK mechanism. HW will negate this bit upon completion of the ZQ calibration process. Upon negation of this bit the ZQ HW calibration pull-up and pull-down results (ZQ_HW_PU_RES and ZQ_HW_PD_RES respectively) are valid **NOTE:** In order to enable this bit ZQ_MODE must be set to either "1" or "3"
15–11 ZQ_HW_PD_ RES	ZQ HW calibration pull-down result. This field holds the pull-down resistor value calculated at the end of the ZQ automatic calibration process with the i.MX ZQ calibration pad. **NOTE:** An offset can be applied to this result, see MMDC_MPPDCMPR2[ZQ_PD_OFFSET]. 00000 Max. resistance. 11111 Min. resistance.

Table continues on the next page...

MMDCx_MPZQHWCTRL field descriptions (continued)

Field	Description
10–6 ZQ_HW_PU_RES	ZQ automatic calibration pull-up result. This field holds the pull-up resistor value calculated at the end of the ZQ automatic calibration process with the i.MX ZQ calibration pad. **NOTE:** An offset can be applied to this result, see MMDC_MPPDCMPR2[ZQ_PU_OFFSET] 00000 Min. resistance. 11111 Max. resistance.
5–2 ZQ_HW_PER	ZQ periodic calibration time. This field determines how often the periodic ZQ calibration is performed. This field is applied for both ZQ short calibration and ZQ automatic calibration process with i.MX ZQ calibration pad. Whenever this timer is expired then according to ZQ_MODE the ZQ automatic calibration process with the i.MX ZQ calibration pad will be issued and/or short/long command will be issued to the external DDR device. This field is ignored if ZQ_MODE equals "00" 0000 ZQ calibration is performed every 1 ms. 0001 ZQ calibration is performed every 2 ms. 0010 ZQ calibration is performed every 4 ms. 1010 ZQ calibration is performed every 1 sec. 1110 ZQ calibration is performed every 16 sec. 1111 ZQ calibration is performed every 32 sec.
ZQ_MODE	ZQ calibration mode: 0x0 No ZQ calibration is issued. (Default) 0x1 ZQ calibration is issued to i.MX ZQ calibration pad together with ZQ long command to the external DDR device only when exiting self refresh. 0x2 ZQ calibration command long/short is issued only to the external DDR device periodically and when exiting self refresh 0x3 ZQ calibration is issued to i.MX ZQ calibration pad together with ZQ calibration command long/short to the external DDR device periodically and when exiting self refresh

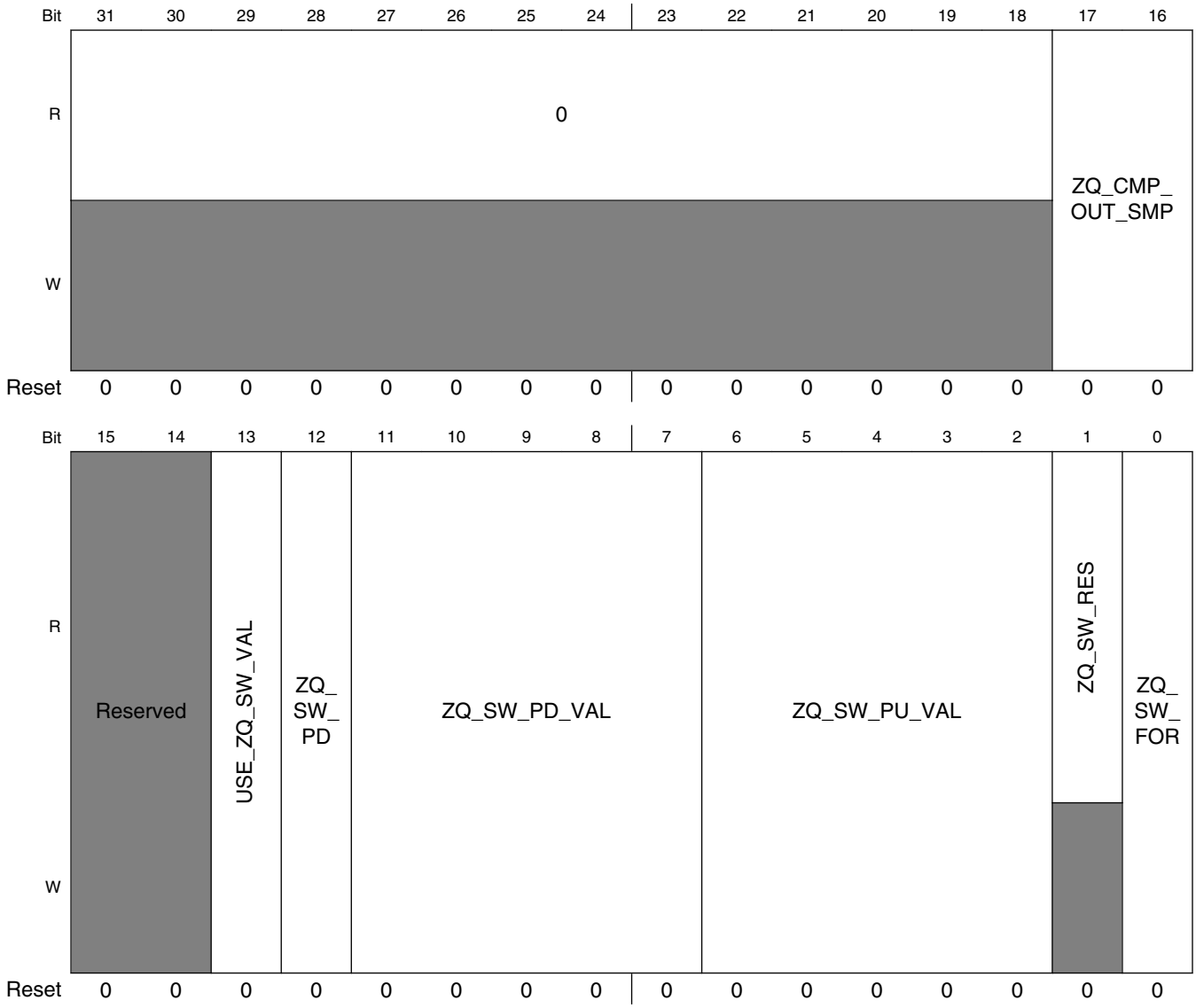
45.12.32 MMDC PHY ZQ SW control register (MMDCx_MPZQSWCTRL)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: When using LPDDR2, ZQ calibration must be applied to all channels in use.

Address: Base address + 804h offset



MMDCx_MPZQSWCTRL field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value 0.
17–16 ZQ_CMP_OUT_ SMP	Defines the amount of cycles between driving the ZQ signals to the ZQ pad and till sampling the comparator enable output while performing ZQ calibration process with the i.MX ZQ calibration pad 00 7 cycles 01 15 cycles 10 23 cycles 11 31 cycles
15–14 -	This field is reserved. Reserved

Table continues on the next page...

MMDCx_MPZQSWCTRL field descriptions (continued)

Field	Description
13 USE_ZQ_SW_VAL	Use SW ZQ configured value for I/O pads resistor controls. This bit selects whether ZQ SW value or ZQ HW value will be driven to the I/O pads resistor controls. By default this bit is cleared and MMDC drives the HW ZQ status bits on the resistor controls of the I/O pads. **NOTE:** This bit should not be updated during ZQ calibration. 0 Fields ZQ_HW_PD_VAL & ZQ_HW_PU_VAL will be driven to I/O pads resistor controls. 1 Fields ZQ_SW_PD_VAL & ZQ_SW_PU_VAL will be driven to I/O pads resistor controls.
12 ZQ_SW_PD	ZQ software PU/PD calibration. This bit determines the calibration stage (PU or PD). 0 PU resistor calibration 1 PD resistor calibration
11–7 ZQ_SW_PD_VAL	ZQ software pull-down resistance. This field determines the value of the PD resistor during SW ZQ calibration. 00000 Max. resistance. 11111 Min. resistance.
6–2 ZQ_SW_PU_VAL	ZQ software pull-up resistance. This field determines the value of the PU resistor during SW ZQ calibration. 00000 Min. resistance. 11111 Max. resistance.
1 ZQ_SW_RES	ZQ software calibration result. This bit reflects the ZQ calibration voltage comparator value. 0 Current ZQ calibration voltage is less than VDD/2. 1 Current ZQ calibration voltage is more than VDD/2
0 ZQ_SW_FOR	ZQ SW calibration enable. This bit when asserted enables ZQ SW calibration. HW negates this bit upon completion of the ZQ SW calibration. Upon negation of this bit the ZQ SW calibration result (i.e. ZQ_SW_RES) is valid

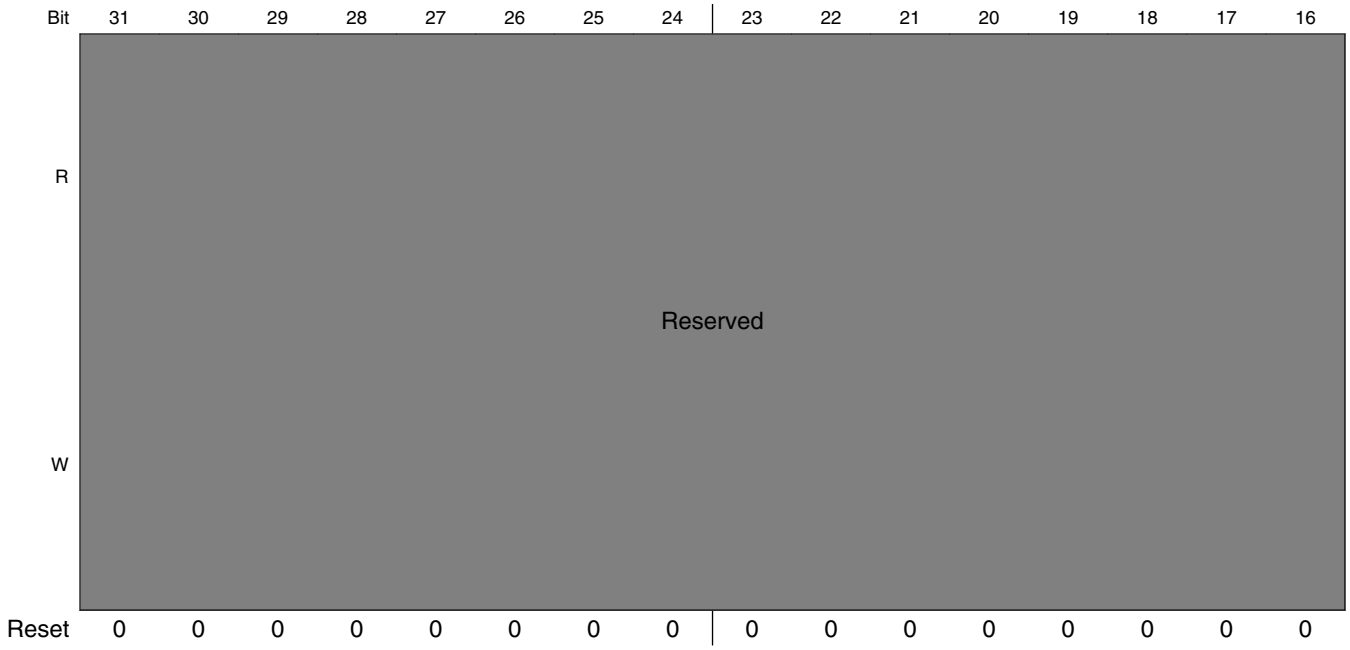
45.12.33 MMDC PHY Write Leveling Configuration and Error Status Register (MMDCx_MPWLGCR)

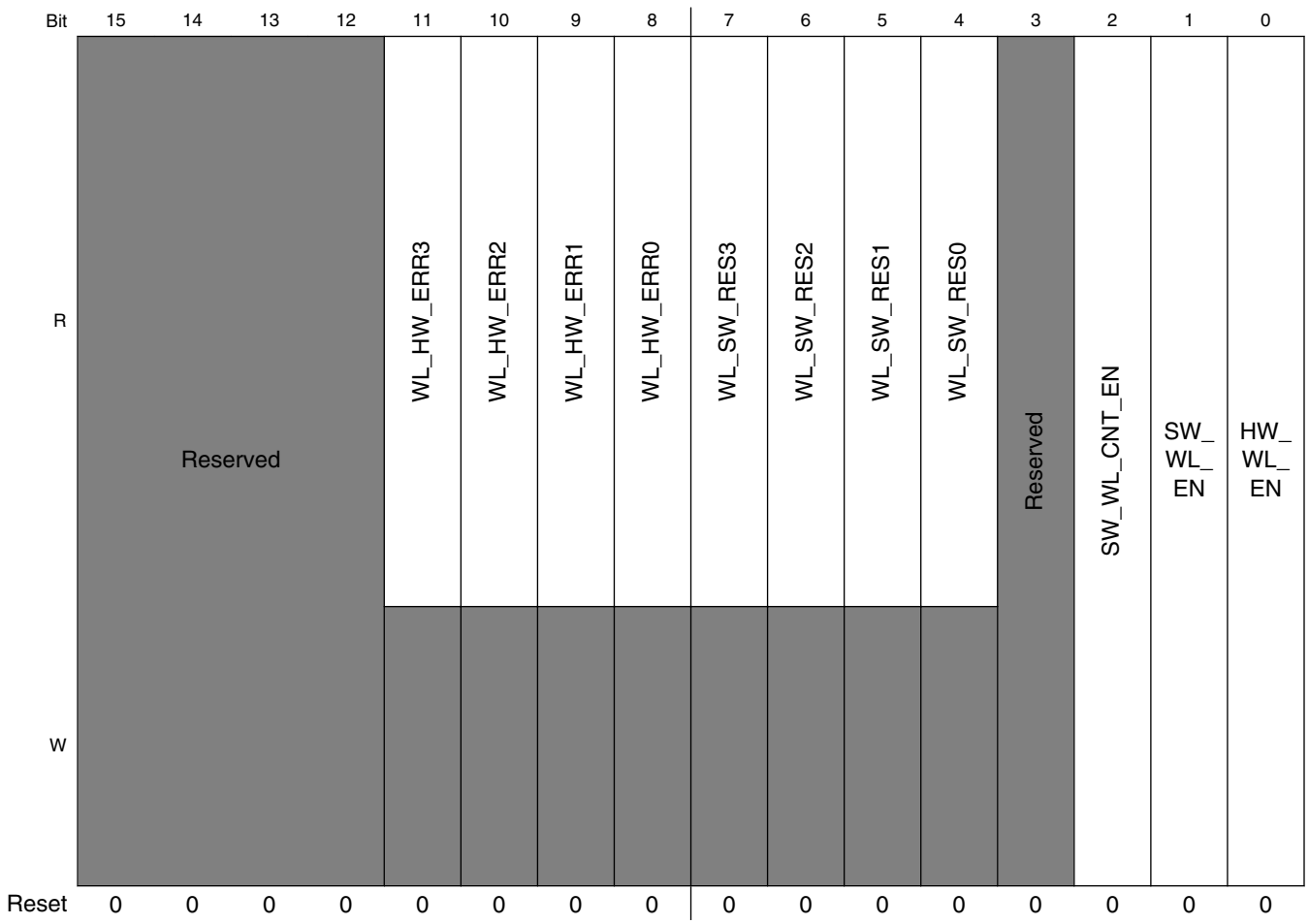
Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 808h offset





MMDCx_MPWLGCR field descriptions

Field	Description
31–12 -	This field is reserved. Reserved
11 WL_HW_ERR3	Byte3 write-leveling HW calibration error. This bit is asserted when an error was found on byte3 during write-leveling HW calibration. This bit is valid only upon completion of the write-leveling HW calibration (i.e. HW_WL_EN bit is de-asserted) 0 No error was found on byte3 during write-leveling HW calibration. 1 An error was found on byte3 during write-leveling HW calibration.
10 WL_HW_ERR2	Byte2 write-leveling HW calibration error. This bit is asserted when an error was found on byte2 during write-leveling HW calibration. This bit is valid only upon completion of the write-leveling HW calibration (i.e. HW_WL_EN bit is de-asserted) 0 No error was found on byte2 during write-leveling HW calibration. 1 An error was found on byte2 during write-leveling HW calibration.
9 WL_HW_ERR1	Byte1 write-leveling HW calibration error. This bit is asserted when an error was found on byte1 during write-leveling HW calibration. This bit is valid only upon completion of the write-leveling HW calibration (i.e. HW_WL_EN bit is de-asserted) 0 No error was found on byte1 during write-leveling HW calibration. 1 An error was found on byte1 during write-leveling HW calibration.
8 WL_HW_ERR0	Byte0 write-leveling HW calibration error. This bit is asserted when an error was found on byte0 during write-leveling HW calibration. This bit is valid only upon completion of the write-leveling HW calibration (i.e. HW_WL_EN bit is de-asserted) 0 No error was found on byte0 during write-leveling HW calibration. 1 An error was found on byte0 during write-leveling HW calibration.
7 WL_SW_RES3	Byte3 write-leveling software result. This bit reflects the value that is driven by the DDR device on DQ24 during SW write-leveling. 0 DQS3 sampled low CK during SW write-leveling. 1 DQS3 sampled high CK during SW write-leveling.
6 WL_SW_RES2	Byte2 write-leveling software result. This bit reflects the value that is driven by the DDR device on DQ16 during SW write-leveling. 0 DQS2 sampled low CK during SW write-leveling. 1 DQS2 sampled high CK during SW write-leveling.
5 WL_SW_RES1	Byte1 write-leveling software result. This bit reflects the value that is driven by the DDR device on DQ8 during SW write-leveling. 0 DQS1 sampled low CK during SW write-leveling. 1 DQS1 sampled high CK during SW write-leveling.
4 WL_SW_RES0	Byte0 write-leveling software result. This bit reflects the value that is driven by the DDR device on DQ0 during SW write-leveling.

Table continues on the next page...

MMDCx_MPWLGCR field descriptions (continued)

Field	Description
	0 DQS0 sampled low CK during SW write-leveling. 1 DQS0 sampled high CK during SW write-leveling.
3 -	This field is reserved. Reserved
2 SW_WL_CNT_EN	SW write-leveling count down enable. This bit when asserted set a certain delay of (25+15) cycles from the setting of SW_WL_EN and before driving the DQS to the DDR device. This bit should be asserted before the first SW write-leveling request and after issuing the write leveling MRS command 0 MMDC doesn't count 25+15 cycles before issuing write-leveling DQS. 1 MMDC counts 25+15 cycles before issuing write-leveling DQS.
1 SW_WL_EN	Write-Leveling SW enable. If this bit is asserted then the MMDC will perform one write-leveling iteration with the DDR device (assuming that Write-Leveling procedure is already enabled in the DDR device through MRS command). HW negate this bit upon completion of the SW write-leveling. Negation of this bit also points that the write-leveling SW calibration result is valid **NOTE:** If this bit and the SW_WL_CNT_EN are enabled the MMDC counts 25 + 15 cycles before issuing the SW write-leveling DQS.
0 HW_WL_EN	Write-Leveling HW (automatic) enable. If this bit is asserted then the MMDC will perform the whole Write-Leveling sequence with the DDR device (assuming that Write-Leveling procedure is already enabled in the DDR device through MRS command). HW negates this bit upon completion of the HW write-leveling. Negation of this bit also points that the write-leveling HW calibration results are valid **NOTE:** Before issuing the first DQS the MMDC counts 25 + 15 cycles automatically as required by the standard.

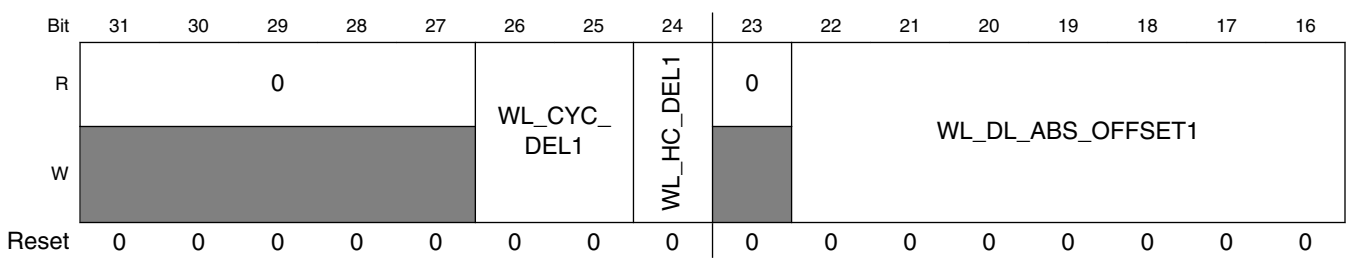
45.12.34 MMDC PHY Write Leveling Delay Control Register 0 (MMDCx_MPWLDECTRL0)

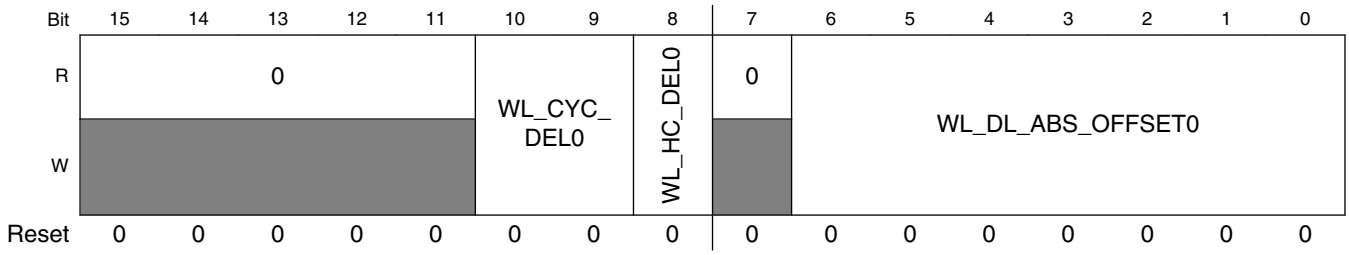
Supported Mode OF Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 80Ch offset





MMDCx_MPWLDECTRL0 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26–25 WL_CYC_DEL1	Write leveling cycle delay for Byte 1. This field indicates whether a delay of 1 or 2 cycles between CK and write DQS is added to the delay that is indicated in the associated WR_DL_ABS_OFFSET and WL_HC_DEL. So the total delay is the sum of (WL_DL_ABS_OFFSET/256*cycle) + (WL_HC_DEL*half cycle) + (WL_CYC_DEL*cycle). When both SW write-leveling is enabled (i.e. SW_WL_EN = 1) or HW write-leveling is enabled (i.e. HW_WL_EN = 1) then this value will be taken as is and will be added to the associated delay that is configured in WL_DL_OFFSET and WL_HC_DEL. **NOTE:** In HW write-leveling this field is not used for indication, as in WL_DL_OFFSET and WL_HC_DEL, but for configuration. 0 No delay is added. 1 1 cycle delay is added. 2 2 cycles delay is added. 3 Reserved.
24 WL_HC_DEL1	Write leveling half cycle delay for Byte 1. This field indicates whether a delay of half cycle between CK and write DQS is added to the delay that is indicated in the associated WR_DL_ABS_OFFSET and WL_CYC_DEL. So the total delay is the sum of (WL_DL_ABS_OFFSET/256*cycle) + (WL_HC_DEL*half cycle) + (WL_CYC_DEL*cycle). When SW write-leveling is enabled (i.e. SW_WL_EN = 1) then this value will be taken as is and will be added to the associated delay that is configured in WL_DL_OFFSET and WL_CYC_DEL. When HW write-leveling is enabled (i.e. HW_WL_EN = 1) then this value will indicate (status) whether a delay of half cycle was added or not to the associated WL_DL_OFFSET and WL_CYC_DEL. 0 No delay is added. 1 Half cycle delay is added.
23 Reserved	This read-only field is reserved and always has the value 0.
22–16 WL_DL_ABS_ OFFSET1	Absolute write-leveling delay offset for Byte 1. This field indicates the absolute delay between CK and write DQS of Byte1 with fractions of a clock period and up to half cycle. This value is process and frequency independent. The value of the delay can be calculated using the following equation (WR_DL_ABS_OFFSET1 / 256) * clock period When SW write-leveling is enabled (i.e. SW_WL_EN = 1) then this value will be taken as is to the associated delay-line. When HW write-leveling is enabled (i.e. HW_WL_EN = 1) then this value will indicate (status) the value that is taken to the associated delay-line at the end of the write-leveling calibration. **NOTE:** The delay-line has a resolution that may vary between device to device, therefore is some cases an increment of the delay by 1 step may be smaller than the delay-line resolution.

Table continues on the next page...

MMDCx_MPWLDECTRL0 field descriptions (continued)

Field	Description
15–11 Reserved	This read-only field is reserved and always has the value 0.
10–9 WL_CYC_DEL0	Write leveling cycle delay for Byte 0. This field indicates whether a delay of 1 or 2 cycles between CK and write DQS is added to the delay that is indicated in the associated WR_DL_ABS_OFFSET and WL_HC_DEL. So the total delay is the sum of (WL_DL_ABS_OFFSET/256*cycle) + (WL_HC_DEL*half cycle) + (WL_CYC_DEL*cycle). When both SW write-leveling is enabled (i.e. SW_WL_EN = 1) or HW write-leveling is enabled (i.e. HW_WL_EN = 1) then this value will be taken as is and will be added to the associated delay that is configured in WL_DL_OFFSET and WL_HC_DEL. Note that in HW write-leveling this field is not used for indication, as in WL_DL_OFFSET and WL_HC_DEL, but for configuration. 0 No delay is added. 1 1 cycle delay is added. 2 2 cycles delay is added. 3 Reserved.
8 WL_HC_DEL0	Write leveling half cycle delay for Byte 0. This field indicates whether a delay of half cycle between CK and write DQS is added to the delay that is indicated in the associated WR_DL_ABS_OFFSET and WL_CYC_DEL. So the total delay is the sum of (WL_DL_ABS_OFFSET/256*cycle) + (WL_HC_DEL*half cycle) + (WL_CYC_DEL*cycle). When SW write-leveling is enabled (i.e. SW_WL_EN = 1) then this value will be taken as is and will be added to the associated delay that is configured in WL_DL_OFFSET and WL_CYC_DEL. When HW write-leveling is enabled (i.e. HW_WL_EN = 1) then this value will indicate (status) whether a delay of half cycle was added or not to the associated WL_DL_OFFSET and WL_CYC_DEL. 0 No delay is added. 1 Half cycle delay is added.
7 Reserved	This read-only field is reserved and always has the value 0.
WL_DL_ABS_ OFFSET0	Absolute write-leveling delay offset for Byte 0. This field indicates the absolute delay between CK and write DQS of Byte0 with fractions of a clock period and up to half cycle. This value is process and frequency independent. The value of the delay can be calculated using the following equation (WR_DL_ABS_OFFSET1 / 256) * clock period When SW write-leveling is enabled (i.e. SW_WL_EN = 1) then this value will be taken as is to the associated delay-line. When HW write-leveling is enabled (i.e. HW_WL_EN = 1) then this value will indicate (status) the value that is taken to the associated delay-line at the end of the write-leveling calibration. **NOTE:** The delay-line has a resolution that may vary between device to device, therefore is some cases an increment of the delay by 1 step may be smaller than the delay-line resolution.

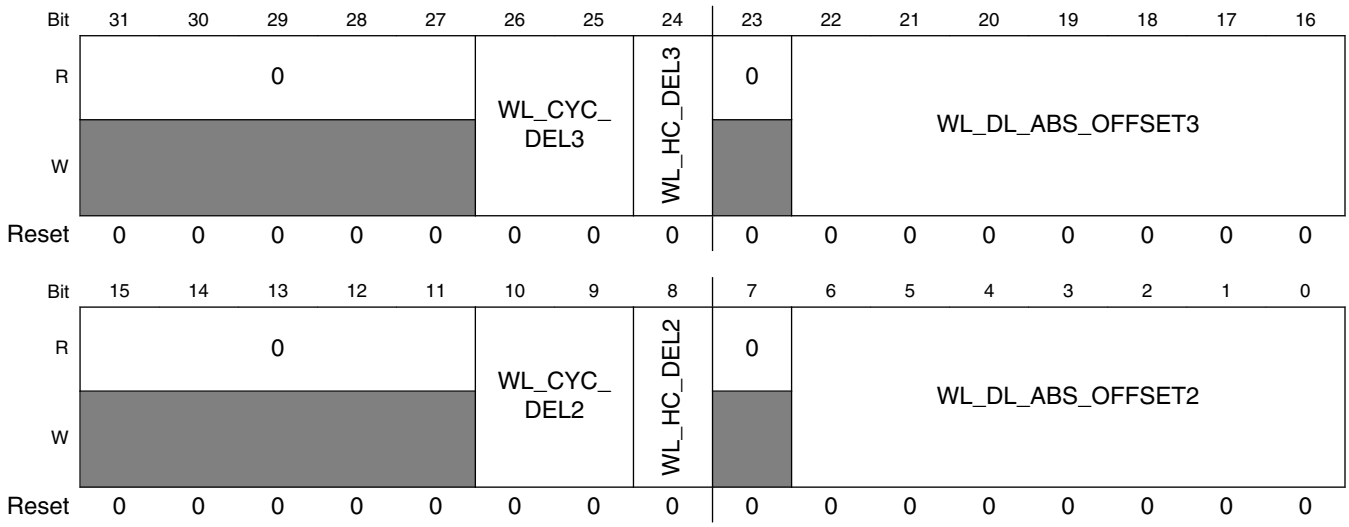
45.12.35 MMDC PHY Write Leveling Delay Control Register 1 (MMDCx_MPWLDECTRL1)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x32

Address: Base address + 810h offset



MMDCx_MPWLDECTRL1 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26–25 WL_CYC_DEL3	Write leveling cycle delay for Byte 3. This field indicates whether a delay of 1 or 2 cycles between CK and write DQS is added to the delay that is indicated in the associated WL_DL_ABS_OFFSET and WL_HC_DEL. So the total delay is the sum of (WL_DL_ABS_OFFSET/256*cycle) + (WL_HC_DEL*half cycle) + (WL_CYC_DEL*cycle). When both SW write-leveling is enabled (i.e. SW_WL_EN = 1) or HW write-leveling is enabled (i.e. HW_WL_EN = 1) then this value will be taken as is and will be added to the associated delay that is configured in WL_DL_OFFSET and WL_HC_DEL. **NOTE:** In HW write-leveling this field is not used for indication, as in WL_DL_OFFSET and WL_HC_DEL, but for configuration. 0 No delay is added. 1 1 cycle delay is added. 2 2 cycles delay is added. 3 Reserved.
24 WL_HC_DEL3	Write leveling half cycle delay for Byte 3. This field indicates whether a delay of half cycle between CK and write DQS is added to the delay that is indicated in the associated WL_DL_ABS_OFFSET and WL_CYC_DEL. So the total delay is the sum of (WL_DL_ABS_OFFSET/256*cycle) + (WL_HC_DEL*half cycle) + (WL_CYC_DEL*cycle). When SW write-leveling is enabled (i.e. SW_WL_EN = 1) then this value will be taken as is and will be added to the associated delay that is configured in WL_DL_OFFSET and WL_CYC_DEL. When HW write-leveling is enabled (i.e. HW_WL_EN = 1) then this value will indicate (status) whether a delay of half cycle was added or not to the associated WL_DL_OFFSET and WL_CYC_DEL. 0 No delay is added. 1 Half cycle delay is added.
23 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

MMDCx_MPWLDECTRL1 field descriptions (continued)

Field	Description
22–16 WL_DL_ABS_OFFSET3	Absolute write-leveling delay offset for Byte 3. This field indicates the absolute delay between CK and write DQS of Byte3 with fractions of a clock period and up to half cycle. This value is process and frequency independent. The value of the delay can be calculated using the following equation (WL_DL_ABS_OFFSET3 / 256) * clock period When SW write-leveling is enabled (i.e. SW_WL_EN = 1) then this value will be taken as is to the associated delay-line. When HW write-leveling is enabled (i.e. HW_WL_EN = 1) then this value will indicate (status) the value that is taken to the associated delay-line at the end of the write-leveling calibration. **NOTE:** The delay-line has a resolution that may vary between device to device, therefore is some cases an increment of the delay by 1 step may be smaller than the delay-line resolution.
15–11 Reserved	This read-only field is reserved and always has the value 0.
10–9 WL_CYC_DEL2	Write leveling cycle delay for Byte 2. This field indicates whether a delay of 1 or 2 cycles between CK and write DQS is added to the delay that is indicated in the associated WR_DL_ABS_OFFSET and WL_HC_DEL. So the total delay is the sum of (WL_DL_ABS_OFFSET/256*cycle) + (WL_HC_DEL*half cycle) + (WL_CYC_DEL*cycle). When both SW write-leveling is enabled (i.e. SW_WL_EN = 1) or HW write-leveling is enabled (i.e. HW_WL_EN = 1) then this value will be taken as is and will be added to the associated delay that is configured in WL_DL_OFFSET and WL_HC_DEL. **NOTE:** In HW write-leveling this field is not used for indication, as in WL_DL_OFFSET and WL_HC_DEL, but for configuration. 0 No delay is added. 1 1 cycle delay is added. 2 2 cycles delay is added. 3 Reserved.
8 WL_HC_DEL2	Write leveling half cycle delay for Byte 2. This field indicates whether a delay of half cycle between CK and write DQS is added to the delay that is indicated in the associated WR_DL_ABS_OFFSET and WL_CYC_DEL. So the total delay is the sum of (WL_DL_ABS_OFFSET/256*cycle) + (WL_HC_DEL*half cycle) + (WL_CYC_DEL*cycle). When SW write-leveling is enabled (i.e. SW_WL_EN = 1) then this value will be taken as is and will be added to the associated delay that is configured in WL_DL_OFFSET and WL_CYC_DEL. When HW write-leveling is enabled (i.e. HW_WL_EN = 1) then this value will indicate (status) whether a delay of half cycle was added or not to the associated WL_DL_OFFSET and WL_CYC_DEL. 0 No delay is added. 1 Half cycle delay is added.
7 Reserved	This read-only field is reserved and always has the value 0.
WL_DL_ABS_OFFSET2	Absolute write-leveling delay offset for Byte 2. This field indicates the absolute delay between CK and write DQS of Byte1 with fractions of a clock period and up to half cycle. This value is process and frequency independent. The value of the delay can be calculated using the following equation (WR_DL_ABS_OFFSET2 / 256) * clock period When SW write-leveling is enabled (i.e. SW_WL_EN = 1) then this value will be taken as is to the associated delay-line. When HW write-leveling is enabled (i.e. HW_WL_EN = 1) then this value will indicate (status) the value that is taken to the associated delay-line at the end of the write-leveling calibration.

Table continues on the next page...

MMDCx_MPWLDECTRL1 field descriptions (continued)

Field	Description
	NOTE: The delay-line has a resolution that may vary between device to device, therefore is some cases an increment of the delay by 1 step may be smaller than the delay-line resolution.

45.12.36 MMDC PHY Write Leveling delay-line Status Register (MMDCx_MPWLDLST)

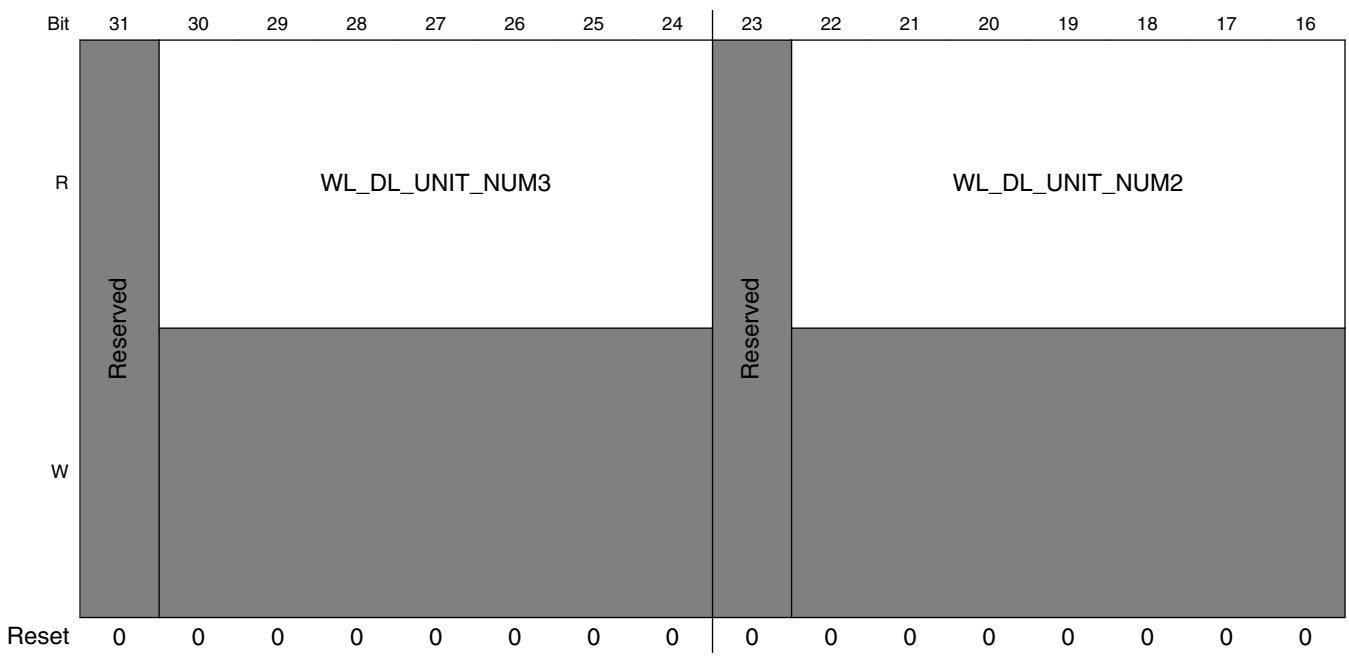
This register holds the status of the four write leveling delay-lines.

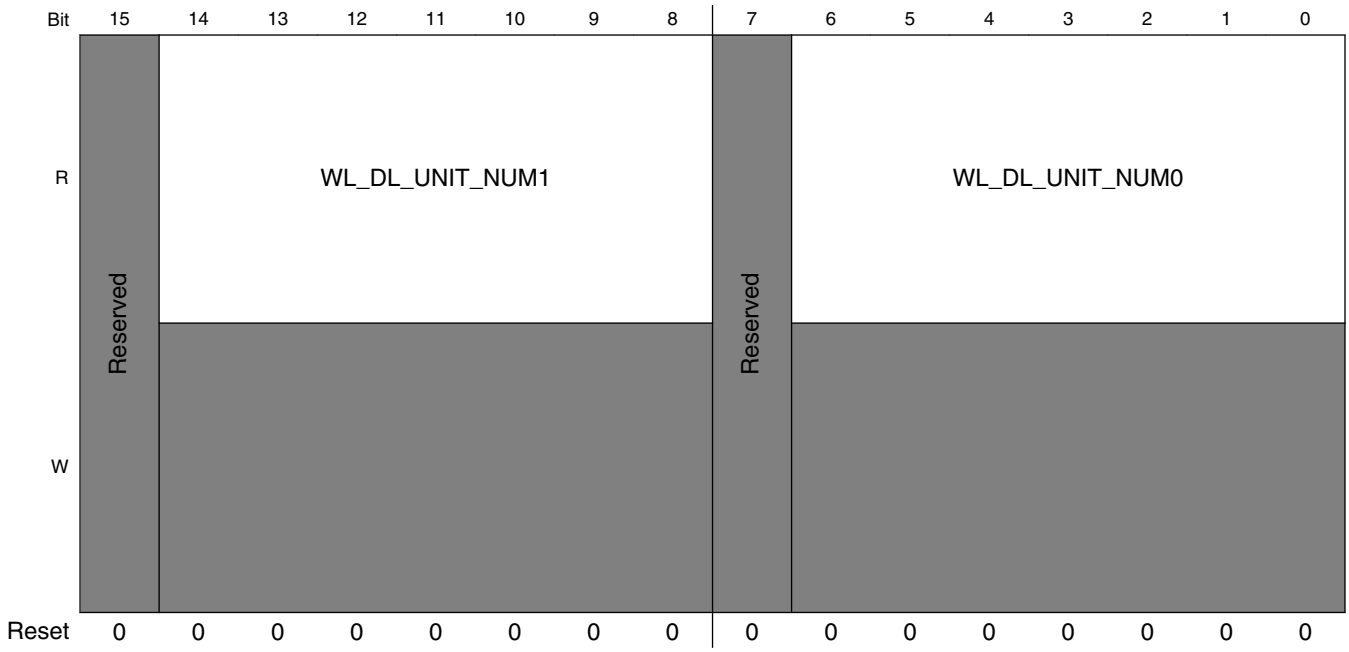
Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 814h offset





MMDCx_MPWLDLST field descriptions

Field	Description
31 -	This field is reserved. Reserved
30–24 WL_DL_UNIT_NUM3	This field reflects the number of delay units that are actually used by write leveling delay-line 3.
23 -	This field is reserved. Reserved
22–16 WL_DL_UNIT_NUM2	This field reflects the number of delay units that are actually used by write leveling delay-line 2.
15 -	This field is reserved. Reserved
14–8 WL_DL_UNIT_NUM1	This field reflects the number of delay units that are actually used by write leveling delay-line 1.
7 -	This field is reserved. Reserved
WL_DL_UNIT_NUM0	This field reflects the number of delay units that are actually used by write leveling delay-line 0.

45.12.37 MMDC PHY ODT control register (MMDCx_MPODTCTRL)

NOTE

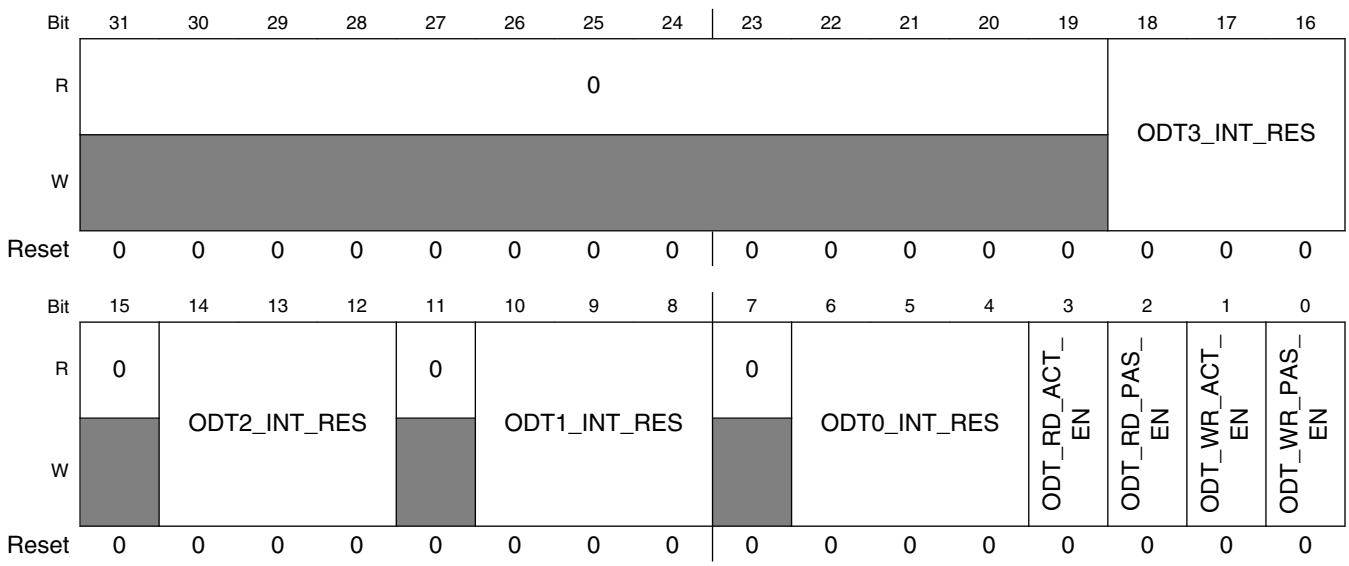
In LPDDR2 mode this register should be cleared, so no termination will be activated

Supported Mode Of Operations:

For Channel 0: DDR3_x16, DDR3_x32, DDR3_x64

For Channel 1: DDR3_x64

Address: Base address + 818h offset



MMDCx_MPODTCTRL field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value 0.
18–16 ODT3_INT_RES	On chip ODT byte3 resistor - This field determines the Rtt_Nom of the on chip ODT byte3 resistor during read accesses. 000 Rtt_Nom Disabled. 001 Rtt_Nom 120 Ohm 010 Rtt_Nom 60 Ohm 011 Rtt_Nom 40 Ohm 100 Rtt_Nom 30 Ohm 101 Rtt_Nom 24 Ohm 110 Rtt_Nom 20 Ohm 111 Rtt_Nom 17 Ohm

Table continues on the next page...

MMDCx_MPODTCTRL field descriptions (continued)

Field	Description
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 ODT2_INT_RES	On chip ODT byte2 resistor - This field determines the Rtt_Nom of the on chip ODT byte2 resistor during read accesses. 000 Rtt_Nom Disabled. 001 Rtt_Nom 120 Ohm 010 Rtt_Nom 60 Ohm 011 Rtt_Nom 40 Ohm 100 Rtt_Nom 30 Ohm 101 Rtt_Nom 24 Ohm 110 Rtt_Nom 20 Ohm 111 Rtt_Nom 17 Ohm
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 ODT1_INT_RES	On chip ODT byte1 resistor - This field determines the Rtt_Nom of the on chip ODT byte1 resistor during read accesses. 0000 Rtt_Nom Disabled. 001 Rtt_Nom 120 Ohm 010 Rtt_Nom 60 Ohm 011 Rtt_Nom 40 Ohm 100 Rtt_Nom 30 Ohm 101 Rtt_Nom 24 Ohm 110 Rtt_Nom 20 Ohm 111 Rtt_Nom 17 Ohm
7 Reserved	This read-only field is reserved and always has the value 0.
6–4 ODT0_INT_RES	On chip ODT byte0 resistor - This field determines the Rtt_Nom of the on chip ODT byte0 resistor during read accesses. 000 Rtt_Nom Disabled. 001 Rtt_Nom 120 Ohm 010 Rtt_Nom 60 Ohm 011 Rtt_Nom 40 Ohm 100 Rtt_Nom 30 Ohm 101 Rtt_Nom 24 Ohm 110 Rtt_Nom 20 Ohm 111 Rtt_Nom 17 Ohm
3 ODT_RD_ACT_EN	Active read CS ODT enable. The bit determines if ODT pin of the active CS will be asserted during read accesses. 0 Active CS ODT pin is disabled during read access. 1 Active CS ODT pin is enabled during read access.
2 ODT_RD_PAS_EN	Inactive read CS ODT enable. The bit determines if ODT pin of the inactive CS will be asserted during read accesses. 0 Inactive CS ODT pin is disabled during read accesses to other CS. 1 Inactive CS ODT pin is enabled during read accesses to other CS.

Table continues on the next page...

MMDCx_MPODTCTRL field descriptions (continued)

Field	Description
1 ODT_WR_ACT_EN	Active write CS ODT enable. The bit determines if ODT pin of the active CS will be asserted during write accesses. 0 Active CS ODT pin is disabled during write access. 1 Active CS ODT pin is enabled during write access.
0 ODT_WR_PAS_EN	Inactive write CS ODT enable. The bit determines if ODT pin of the inactive CS will be asserted during write accesses. 0 Inactive CS ODT pin is disabled during write accesses to other CS. 1 Inactive CS ODT pin is enabled during write accesses to other CS.

45.12.38 MMDC PHY Read DQ Byte0 Delay Register (MMDCx_MPRDDQBY0DL)

This register is used to add fine-tuning adjustment to every bit in the read DQ byte0 relative to the read DQS. This delay is in addition to the read data calibration. If operating in 64-bit mode, there is an identical register that is mapped at the second base address.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 81Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	rd_dq7_del			0	rd_dq6_del			0	rd_dq5_del			0	rd_dq4_del		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	rd_dq3_del			0	rd_dq2_del			0	rd_dq1_del			0	rd_dq0_del		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPRDDQBY0DL field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–28 rd_dq7_del	Read dqs0 to dq7 delay fine-tuning. This field holds the number of delay units that are added to dq7 relative to dqs0. 000 No change in dq7 delay 001 Add dq7 delay of 1 delay unit 010 Add dq7 delay of 2 delay units.

Table continues on the next page...

MMDCx_MPRDDQBY0DL field descriptions (continued)

Field	Description
	011 Add dq7 delay of 3 delay units. 100 Add dq7 delay of 4 delay units. 101 Add dq7 delay of 5 delay units. 110 Add dq7 delay of 6 delay units. 111 Add dq7 delay of 7 delay units.
27 Reserved	This read-only field is reserved and always has the value 0.
26–24 rd_dq6_del	Read dqs0 to dq6 delay fine-tuning. This field holds the number of delay units that are added to dq6 relative to dqs0. 000 No change in dq6 delay 001 Add dq6 delay of 1 delay unit 010 Add dq6 delay of 2 delay units. 011 Add dq6 delay of 3 delay units. 100 Add dq6 delay of 4 delay units. 101 Add dq6 delay of 5 delay units. 110 Add dq6 delay of 6 delay units. 111 Add dq6 delay of 7 delay units.
23 Reserved	This read-only field is reserved and always has the value 0.
22–20 rd_dq5_del	Read dqs0 to dq5 delay fine-tuning. This field holds the number of delay units that are added to dq5 relative to dqs0. 000 No change in dq5 delay 001 Add dq5 delay of 1 delay unit 010 Add dq5 delay of 2 delay units. 011 Add dq5 delay of 3 delay units. 100 Add dq5 delay of 4 delay units. 101 Add dq5 delay of 5 delay units. 110 Add dq5 delay of 6 delay units. 111 Add dq5 delay of 7 delay units.
19 Reserved	This read-only field is reserved and always has the value 0.
18–16 rd_dq4_del	Read dqs0 to dq4 delay fine-tuning. This field holds the number of delay units that are added to dq4 relative to dqs0. 000 No change in dq4 delay 001 Add dq4 delay of 1 delay unit 010 Add dq4 delay of 2 delay units. 011 Add dq4 delay of 3 delay units. 100 Add dq4 delay of 4 delay units. 101 Add dq4 delay of 5 delay units. 110 Add dq4 delay of 6 delay units. 111 Add dq4 delay of 7 delay units.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 rd_dq3_del	Read dqs0 to dq3 delay fine-tuning. This field holds the number of delay units that are added to dq3 relative to dqs0.

Table continues on the next page...

MMDCx_MPRDDQBY0DL field descriptions (continued)

Field	Description
	000 No change in dq3 delay 001 Add dq3 delay of 1 delay unit 010 Add dq3 delay of 2 delay units. 011 Add dq3 delay of 3 delay units. 100 Add dq3 delay of 4 delay units. 101 Add dq3 delay of 5 delay units. 110 Add dq3 delay of 6 delay units. 111 Add dq3 delay of 7 delay units.
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 rd_dq2_del	Read dqs0 to dq2 delay fine-tuning. This field holds the number of delay units that are added to dq2 relative to dqs0. 000 No change in dq2 delay 001 Add dq2 delay of 1 delay unit 010 Add dq2 delay of 2 delay units. 011 Add dq2 delay of 3 delay units. 100 Add dq2 delay of 4 delay units. 101 Add dq2 delay of 5 delay units. 110 Add dq2 delay of 6 delay units. 111 Add dq2 delay of 7 delay units.
7 Reserved	This read-only field is reserved and always has the value 0.
6–4 rd_dq1_del	Read dqs0 to dq1 delay fine-tuning. This field holds the number of delay units that are added to dq1 relative to dqs0. 000 No change in dq1 delay 001 Add dq1 delay of 1 delay unit 010 Add dq1 delay of 2 delay units. 011 Add dq1 delay of 3 delay units. 100 Add dq1 delay of 4 delay units. 101 Add dq1 delay of 5 delay units. 110 Add dq1 delay of 6 delay units. 111 Add dq1 delay of 7 delay units.
3 Reserved	This read-only field is reserved and always has the value 0.
rd_dq0_del	Read dqs0 to dq0 delay fine-tuning. This field holds the number of delay units that are added to dq0 relative to dqs0. 000 No change in dq0 delay 001 Add dq0 delay of 1 delay unit 010 Add dq0 delay of 2 delay units. 011 Add dq0 delay of 3 delay units. 100 Add dq0 delay of 4 delay units. 101 Add dq0 delay of 5 delay units. 110 Add dq0 delay of 6 delay units. 111 Add dq0 delay of 7 delay units.

45.12.39 MMDC PHY Read DQ Byte1 Delay Register (MMDCx_MPRDDQBY1DL)

This register is used to add fine-tuning adjustment to every bit in the read DQ byte1 relative to the read DQS

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 820h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	rd_dq15_del			0	rd_dq14_del			0	rd_dq13_del			0	rd_dq12_del		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	rd_dq11_del			0	rd_dq10_del			0	rd_dq9_del			0	rd_dq8_del		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPRDDQBY1DL field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–28 rd_dq15_del	Read dqs1 to dq15 delay fine-tuning. This field holds the number of delay units that are added to dq15 relative to dqs1. 000 No change in dq15 delay 001 Add dq15 delay of 1 delay unit 010 Add dq15 delay of 2 delay units. 011 Add dq15 delay of 3 delay units. 100 Add dq15 delay of 4 delay units. 101 Add dq15 delay of 5 delay units. 110 Add dq15 delay of 6 delay units. 111 Add dq15 delay of 7 delay units.
27 Reserved	This read-only field is reserved and always has the value 0.
26–24 rd_dq14_del	Read dqs1 to dq14 delay fine-tuning. This field holds the number of delay units that are added to dq14 relative to dqs1. 000 No change in dq14 delay 001 Add dq14 delay of 1 delay unit 010 Add dq14 delay of 2 delay units. 011 Add dq14 delay of 3 delay units. 100 Add dq14 delay of 4 delay units.

Table continues on the next page...

MMDCx_MPRDDQBY1DL field descriptions (continued)

Field	Description
	101 Add dq14 delay of 5 delay units. 110 Add dq14 delay of 6 delay units. 111 Add dq14 delay of 7 delay units.
23 Reserved	This read-only field is reserved and always has the value 0.
22–20 rd_dq13_del	Read dqs1 to dq13 delay fine-tuning. This field holds the number of delay units that are added to dq13 relative to dqs1. 000 No change in dq13 delay 001 Add dq13 delay of 1 delay unit 010 Add dq13 delay of 2 delay units. 011 Add dq13 delay of 3 delay units. 100 Add dq13 delay of 4 delay units. 101 Add dq13 delay of 5 delay units. 110 Add dq13 delay of 6 delay units. 111 Add dq13 delay of 7 delay units.
19 Reserved	This read-only field is reserved and always has the value 0.
18–16 rd_dq12_del	Read dqs1 to dq12 delay fine-tuning. This field holds the number of delay units that are added to dq12 relative to dqs1. 000 No change in dq12 delay 001 Add dq12 delay of 1 delay unit 010 Add dq12 delay of 2 delay units. 011 Add dq12 delay of 3 delay units. 100 Add dq12 delay of 4 delay units. 101 Add dq12 delay of 5 delay units. 110 Add dq12 delay of 6 delay units. 111 Add dq12 delay of 7 delay units.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 rd_dq11_del	Read dqs1 to dq11 delay fine-tuning. This field holds the number of delay units that are added to dq11 relative to dqs1. 000 No change in dq11 delay 001 Add dq11 delay of 1 delay unit 010 Add dq11 delay of 2 delay units. 011 Add dq11 delay of 3 delay units. 100 Add dq11 delay of 4 delay units. 101 Add dq11 delay of 5 delay units. 110 Add dq11 delay of 6 delay units. 111 Add dq11 delay of 7 delay units.
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 rd_dq10_del	Read dqs1 to dq10 delay fine-tuning. This field holds the number of delay units that are added to dq10 relative to dqs1. 000 No change in dq10 delay

Table continues on the next page...

MMDCx_MPRDDQBY1DL field descriptions (continued)

Field	Description
	001 Add dq10 delay of 1 delay unit 010 Add dq10 delay of 2 delay units. 011 Add dq10 delay of 3 delay units. 100 Add dq10 delay of 4 delay units. 101 Add dq10 delay of 5 delay unit 110 Add dq10 delay of 6 delay units. 111 Add dq10 delay of 7 delay units.
7 Reserved	This read-only field is reserved and always has the value 0.
6–4 rd_dq9_del	Read dqs1 to dq9 delay fine-tuning. This field holds the number of delay units that are added to dq9 relative to dqs1. 000 No change in dq9 delay 001 Add dq9 delay of 1 delay unit 010 Add dq9 delay of 2 delay units. 011 Add dq9 delay of 3 delay units. 100 Add dq9 delay of 4 delay units. 101 Add dq9 delay of 5 delay units. 110 Add dq9 delay of 6 delay units. 111 Add dq9 delay of 7 delay units.
3 Reserved	This read-only field is reserved and always has the value 0.
rd_dq8_del	Read dqs1 to dq8 delay fine-tuning. This field holds the number of delay units that are added to dq8 relative to dqs1. 000 No change in dq8 delay 001 Add dq8 delay of 1 delay unit 010 Add dq8 delay of 2 delay units. 011 Add dq8 delay of 3 delay units. 100 Add dq8 delay of 4 delay units. 101 Add dq8 delay of 5 delay units. 110 Add dq8 delay of 6 delay units. 111 Add dq8 delay of 7 delay units.

45.12.40 MMDC PHY Read DQ Byte2 Delay Register (MMDCx_MPRDDQBY2DL)

This register is used to add fine-tuning adjustment to every bit in the read DQ byte2 relative to the read DQS

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x32

Address: Base address + 824h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	rd_dq23_del			0	rd_dq22_del			0	rd_dq21_del			0	rd_dq20_del		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	rd_dq19_del			0	rd_dq18_del			0	rd_dq17_del			0	rd_dq16_del		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPRDDQBY2DL field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–28 rd_dq23_del	Read dqs2 to dq23 delay fine-tuning. This field holds the number of delay units that are added to dq23 relative to dqs2. 000 No change in dq23 delay 001 Add dq23 delay of 1 delay unit 010 Add dq23 delay of 2 delay units. 011 Add dq23 delay of 3 delay units. 100 Add dq23 delay of 4 delay units. 101 Add dq23 delay of 5 delay units. 110 Add dq23 delay of 6 delay units. 111 Add dq23 delay of 7 delay units.
27 Reserved	This read-only field is reserved and always has the value 0.
26–24 rd_dq22_del	Read dqs2 to dq22 delay fine-tuning. This field holds the number of delay units that are added to dq22 relative to dqs2. 000 No change in dq22 delay 001 Add dq22 delay of 1 delay unit 010 Add dq22 delay of 2 delay units. 011 Add dq22 delay of 3 delay units. 100 Add dq22 delay of 4 delay units. 101 Add dq22 delay of 5 delay units. 110 Add dq22 delay of 6 delay units. 111 Add dq22 delay of 7 delay units.
23 Reserved	This read-only field is reserved and always has the value 0.
22–20 rd_dq21_del	Read dqs2 to dq21 delay fine-tuning. This field holds the number of delay units that are added to dq21 relative to dqs2. 000 No change in dq21 delay 001 Add dq21 delay of 1 delay unit 010 Add dq21 delay of 2 delay units. 011 Add dq21 delay of 3 delay units. 100 Add dq21 delay of 4 delay units. 101 Add dq21 delay of 5 delay units.

Table continues on the next page...

MMDCx_MPRDDQBY2DL field descriptions (continued)

Field	Description
	110 Add dq21 delay of 6 delay units. 111 Add dq21 delay of 7 delay units.
19 Reserved	This read-only field is reserved and always has the value 0.
18–16 rd_dq20_del	Read dqs2 to dq20 delay fine-tuning. This field holds the number of delay units that are added to dq20 relative to dqs2. 000 No change in dq20 delay 001 Add dq20 delay of 1 delay unit 010 Add dq20 delay of 2 delay units. 011 Add dq20 delay of 3 delay units. 100 Add dq20 delay of 4 delay units. 101 Add dq20 delay of 5 delay units. 110 Add dq20 delay of 6 delay units. 111 Add dq20 delay of 7 delay units.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 rd_dq19_del	Read dqs2 to dq19 delay fine-tuning. This field holds the number of delay units that are added to dq19 relative to dqs2. 000 No change in dq19 delay 001 Add dq19 delay of 1 delay unit 010 Add dq19 delay of 2 delay units. 011 Add dq19 delay of 3 delay units. 100 Add dq19 delay of 4 delay units. 101 Add dq19 delay of 5 delay units. 110 Add dq19 delay of 6 delay units. 111 Add dq19 delay of 7 delay units.
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 rd_dq18_del	Read dqs2 to dq18 delay fine-tuning. This field holds the number of delay units that are added to dq18 relative to dqs2. 000 No change in dq18 delay 001 Add dq18 delay of 1 delay unit 010 Add dq18 delay of 2 delay units. 011 Add dq18 delay of 3 delay units. 100 Add dq18 delay of 4 delay units. 101 Add dq18 delay of 5 delay units. 110 Add dq18 delay of 6 delay units. 111 Add dq18 delay of 7 delay units.
7 Reserved	This read-only field is reserved and always has the value 0.
6–4 rd_dq17_del	Read dqs2 to dq17 delay fine-tuning. This field holds the number of delay units that are added to dq17 relative to dqs2. 000 No change in dq17 delay 001 Add dq17 delay of 1 delay unit

Table continues on the next page...

MMDCx_MPRDDQBY2DL field descriptions (continued)

Field	Description
	010 Add dq17 delay of 2 delay units. 011 Add dq17 delay of 3 delay units. 100 Add dq17 delay of 4 delay units. 101 Add dq17 delay of 5 delay units. 110 Add dq17 delay of 6 delay units. 111 Add dq17 delay of 7 delay units.
3 Reserved	This read-only field is reserved and always has the value 0.
rd_dq16_del	Read dqs2 to dq16 delay fine-tuning. This field holds the number of delay units that are added to dq16 relative to dqs2. 000 No change in dq16 delay 001 Add dq16 delay of 1 delay unit 010 Add dq16 delay of 2 delay units. 011 Add dq16 delay of 3 delay units. 100 Add dq16 delay of 4 delay units. 101 Add dq16 delay of 5 delay units. 110 Add dq16 delay of 6 delay units. 111 Add dq16 delay of 7 delay units.

45.12.41 MMDC PHY Read DQ Byte3 Delay Register (MMDCx_MPRDDQBY3DL)

This register is used to add fine-tuning adjustment to every bit in the read DQ byte3 relative to the read DQS.

The bit assignments and the bit field descriptions for the register are shown below.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x32

Address: Base address + 828h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	rd_dq31_del			0	rd_dq30_del			0	rd_dq29_del			0	rd_dq28_del		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	rd_dq27_del			0	rd_dq26_del			0	rd_dq25_del			0	rd_dq24_del		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPRDDQBY3DL field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–28 rd_dq31_del	Read dqs3 to dq31 delay fine-tuning. This field holds the number of delay units that are added to dq31 relative to dqs3. 000 No change in dq31 delay 001 Add dq31 delay of 1 delay unit 010 Add dq31 delay of 2 delay units. 011 Add dq31 delay of 3 delay units. 100 Add dq31 delay of 4 delay units. 101 Add dq31 delay of 5 delay units. 110 Add dq31 delay of 6 delay units. 111 Add dq31 delay of 7 delay units.
27 Reserved	This read-only field is reserved and always has the value 0.
26–24 rd_dq30_del	Read dqs3 to dq30 delay fine-tuning. This field holds the number of delay units that are added to dq30 relative to dqs3. 000 No change in dq30 delay 001 Add dq30 delay of 1 delay unit 010 Add dq30 delay of 2 delay units. 011 Add dq30 delay of 3 delay units. 100 Add dq30 delay of 4 delay units. 101 Add dq30 delay of 5 delay units. 110 Add dq30 delay of 6 delay units. 111 Add dq30 delay of 7 delay units.
23 Reserved	This read-only field is reserved and always has the value 0.
22–20 rd_dq29_del	Read dqs3 to dq29 delay fine-tuning. This field holds the number of delay units that are added to dq29 relative to dqs3. 000 No change in dq29 delay 001 Add dq29 delay of 1 delay unit 010 Add dq29 delay of 2 delay units. 011 Add dq29 delay of 3 delay units. 100 Add dq29 delay of 4 delay units. 101 Add dq29 delay of 5 delay units. 110 Add dq29 delay of 6 delay units. 111 Add dq29 delay of 7 delay units.
19 Reserved	This read-only field is reserved and always has the value 0.
18–16 rd_dq28_del	Read dqs3 to dq28 delay fine-tuning. This field holds the number of delay units that are added to dq28 relative to dqs3. 000 No change in dq28 delay 001 Add dq28 delay of 1 delay unit 010 Add dq28 delay of 2 delay units. 011 Add dq28 delay of 3 delay units.

Table continues on the next page...

MMDCx_MPRDDQBY3DL field descriptions (continued)

Field	Description
	100 Add dq28 delay of 4 delay units. 101 Add dq28 delay of 5 delay units. 110 Add dq28 delay of 6 delay units. 111 Add dq28 delay of 7 delay units.
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 rd_dq27_del	Read dqs3 to dq27 delay fine-tuning. This field holds the number of delay units that are added to dq27 relative to dqs3. 000 No change in dq27 delay 001 Add dq27 delay of 1 delay unit 010 Add dq27 delay of 2 delay units. 011 Add dq27 delay of 3 delay units. 100 Add dq27 delay of 4 delay units. 101 Add dq27 delay of 5 delay units. 110 Add dq27 delay of 6 delay units. 111 Add dq27 delay of 7 delay units.
11 Reserved	This read-only field is reserved and always has the value 0.
10–8 rd_dq26_del	Read dqs3 to dq26 delay fine-tuning. This field holds the number of delay units that are added to dq26 relative to dqs3. 000 No change in dq26 delay 001 Add dq26 delay of 1 delay unit 010 Add dq26 delay of 2 delay units. 011 Add dq26 delay of 3 delay units. 100 Add dq26 delay of 4 delay units. 101 Add dq26 delay of 5 delay units. 110 Add dq26 delay of 6 delay units. 111 Add dq26 delay of 7 delay units.
7 Reserved	This read-only field is reserved and always has the value 0.
6–4 rd_dq25_del	Read dqs3 to dq25 delay fine-tuning. This field holds the number of delay units that are added to dq25 relative to dqs3. 000 No change in dq25 delay 001 Add dq25 delay of 1 delay unit 010 Add dq25 delay of 2 delay units. 011 Add dq25 delay of 3 delay units. 100 Add dq25 delay of 4 delay units. 101 Add dq25 delay of 5 delay units. 110 Add dq25 delay of 6 delay units. 111 Add dq25 delay of 7 delay units.
3 Reserved	This read-only field is reserved and always has the value 0.
rd_dq24_del	Read dqs3 to dq24 delay fine-tuning. This field holds the number of delay units that are added to dq24 relative to dqs3.

Table continues on the next page...

MMDCx_MPRDDQBY3DL field descriptions (continued)

Field	Description
	000 No change in dq24 delay 001 Add dq24 delay of 1 delay unit 010 Add dq24 delay of 2 delay units. 011 Add dq24 delay of 3 delay units. 100 Add dq24 delay of 4 delay units. 101 Add dq24 delay of 5 delay units. 110 Add dq24 delay of 6 delay units. 111 Add dq24 delay of 7 delay units.

45.12.42 MMDC PHY Write DQ Byte0 Delay Register (MMDCx_MPWRDQBY0DL)

This register is used to add fine-tuning adjustment to every bit in the write DQ byte0 relative to the write DQS

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 82Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	wr_dm0_del		wr_dq7_del		0		wr_dq6_del		0		wr_dq5_del		0		wr_dq4_del	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	0		wr_dq3_del		0		wr_dq2_del		0		wr_dq1_del		0		wr_dq0_del	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPWRDQBY0DL field descriptions

Field	Description
31–30 wr_dm0_del	Write dm0 delay fine-tuning. This field holds the number of delay units that are added to dm0 relative to dqs0. 00 No change in dm0 delay 01 Add dm0 delay of 1 delay unit. 10 Add dm0 delay of 2 delay units. 11 Add dm0 delay of 3 delay units.
29–28 wr_dq7_del	Write dq7 delay fine-tuning. This field holds the number of delay units that are added to dq7 relative to dqs0. 00 No change in dq7 delay

Table continues on the next page...

MMDCx_MPWRDQBY0DL field descriptions (continued)

Field	Description
	01 Add dq7 delay of 1 delay unit. 10 Add dq7 delay of 2 delay units. 11 Add dq7 delay of 3 delay units.
27–26 Reserved	This read-only field is reserved and always has the value 0.
25–24 wr_dq6_del	Write dq6 delay fine-tuning. This field holds the number of delay units that are added to dq6 relative to dqs0. 00 No change in dq6 delay 01 Add dq6 delay of 1 delay unit. 10 Add dq6 delay of 2 delay units. 11 Add dq6 delay of 3 delay units.
23–22 Reserved	This read-only field is reserved and always has the value 0.
21–20 wr_dq5_del	Write dq5 delay fine-tuning. This field holds the number of delay units that are added to dq5 relative to dqs0. 00 No change in dq5 delay 01 Add dq5 delay of 1 delay unit. 10 Add dq5 delay of 2 delay units. 11 Add dq5 delay of 3 delay units.
19–18 Reserved	This read-only field is reserved and always has the value 0.
17–16 wr_dq4_del	Write dq4 delay fine-tuning. This field holds the number of delay units that are added to dq4 relative to dqs0. 00 No change in dq4 delay 01 Add dq4 delay of 1 delay unit.. 10 Add dq4 delay of 2 delay units. 11 Add dq4 delay of 3 delay units.
15–14 Reserved	This read-only field is reserved and always has the value 0.
13–12 wr_dq3_del	Write dq3 delay fine-tuning. This field holds the number of delay units that are added to dq3 relative to dqs0. 00 No change in dq3 delay 01 Add dq3 delay of 1 delay unit. 10 Add dq3 delay of 2 delay units. 11 Add dq3 delay of 3 delay units.
11–10 Reserved	This read-only field is reserved and always has the value 0.
9–8 wr_dq2_del	Write dq2 delay fine-tuning. This field holds the number of delay units that are added to dq2 relative to dqs0. 00 No change in dq2 delay 01 Add dq2 delay of 1 delay unit. 10 Add dq2 delay of 2 delay units. 11 Add dq2 delay of 3 delay units.

Table continues on the next page...

MMDCx_MPWRDQBY0DL field descriptions (continued)

Field	Description
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–4 wr_dq1_del	Write dq1 delay fine-tuning. This field holds the number of delay units that are added to dq1 relative to dqs0. 00 No change in dq1 delay 01 Add dq1 delay of 1 delay unit. 10 Add dq1 delay of 2 delay units. 11 Add dq1 delay of 3 delay units.
3–2 Reserved	This read-only field is reserved and always has the value 0.
wr_dq0_del	Write dq0 delay fine-tuning. This field holds the number of delay units that are added to dq0 relative to dqs0. 00 No change in dq0 delay 01 Add dq0 delay of 1 delay unit. 10 Add dq0 delay of 2 delay units. 11 Add dq0 delay of 3 delay units.

45.12.43 MMDC PHY Write DQ Byte1 Delay Register (MMDCx_MPWRDQBY1DL)

This register is used to add fine-tuning adjustment to every bit in the write DQ byte1 relative to the write DQS

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 830h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	wr_dm1_del		wr_dq15_del		0		wr_dq14_del		0		wr_dq13_del		0		wr_dq12_del	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		wr_dq11_del		0		wr_dq10_del		0		wr_dq9_del		0		wr_dq8_del	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPWRDQBY1DL field descriptions

Field	Description
31–30 wr_dm1_del	Write dm1 delay fine-tuning. This field holds the number of delay units that are added to dm1 relative to dqs1. 00 No change in dm1 delay 01 Add dm1 delay of 1 delay unit. 10 Add dm1 delay of 2 delay units. 11 Add dm1 delay of 3 delay units.
29–28 wr_dq15_del	Write dq15 delay fine-tuning. This field holds the number of delay units that are added to dq15 relative to dqs1. 00 No change in dq15 delay 01 Add dq15 delay of 1 delay unit. 10 Add dq15 delay of 2 delay units. 11 Add dq15 delay of 3 delay units.
27–26 Reserved	This read-only field is reserved and always has the value 0.
25–24 wr_dq14_del	Write dq14 delay fine-tuning. This field holds the number of delay units that are added to dq14 relative to dqs1. 00 No change in dq14 delay 01 Add dq14 delay of 1 delay unit. 10 Add dq14 delay of 2 delay units. 11 Add dq14 delay of 3 delay units.
23–22 Reserved	This read-only field is reserved and always has the value 0.
21–20 wr_dq13_del	Write dq13 delay fine-tuning. This field holds the number of delay units that are added to dq13 relative to dqs1. 00 No change in dq13 delay 01 Add dq13 delay of 1 delay unit. 10 Add dq13 delay of 2 delay units. 11 Add dq13 delay of 3 delay units.
19–18 Reserved	This read-only field is reserved and always has the value 0.
17–16 wr_dq12_del	Write dq12 delay fine-tuning. This field holds the number of delay units that are added to dq12 relative to dqs1. 00 No change in dq12 delay 01 Add dq12 delay of 1 delay unit. 10 Add dq12 delay of 2 delay units. 11 Add dq12 delay of 3 delay units.
15–14 Reserved	This read-only field is reserved and always has the value 0.
13–12 wr_dq11_del	Write dq11 delay fine-tuning. This field holds the number of delay units that are added to dq11 relative to dqs1. 00 No change in dq11 delay 01 Add dq11 delay of 1 delay unit.

Table continues on the next page...

MMDCx_MPWRDQBY1DL field descriptions (continued)

Field	Description
	10 Add dq11 delay of 2 delay units. 11 Add dq11 delay of 3 delay units.
11–10 Reserved	This read-only field is reserved and always has the value 0.
9–8 wr_dq10_del	Write dq10 delay fine-tuning. This field holds the number of delay units that are added to dq10 relative to dqs1. 00 No change in dq10 delay 01 Add dq10 delay of 1 delay unit. 10 Add dq10 delay of 2 delay units. 11 Add dq10 delay of 3 delay units.
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–4 wr_dq9_del	Write dq9 delay fine-tuning. This field holds the number of delay units that are added to dq9 relative to dqs1. 00 No change in dq9 delay 01 Add dq9 delay of 1 delay unit. 10 Add dq9 delay of 2 delay units. 11 Add dq9 delay of 3 delay units.
3–2 Reserved	This read-only field is reserved and always has the value 0.
wr_dq8_del	Write dq8 delay fine-tuning. This field holds the number of delay units that are added to dq8 relative to dqs1. 00 No change in dq8 delay 01 Add dq8 delay of 1 delay unit. 10 Add dq8 delay of 2 delay units. 11 Add dq8 delay of 3 delay units.

45.12.44 MMDC PHY Write DQ Byte2 Delay Register (MMDCx_MPWRDQBY2DL)

This register is used to add fine-tuning adjustment to every bit in the write DQ byte2 relative to the write DQS

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x32

Address: Base address + 834h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	wr_dm2_del		wr_dq23_del		0		wr_dq22_del		0		wr_dq21_del		0		wr_dq20_del	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		wr_dq19_del		0		wr_dq18_del		0		wr_dq17_del		0		wr_dq16_del	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPWRDQBY2DL field descriptions

Field	Description
31–30 wr_dm2_del	Write dm2 delay fine-tuning. This field holds the number of delay units that are added to dm2 relative to dqs2. 00 No change in dm2 delay 01 Add dm2 delay of 1 delay unit. 10 Add dm2 delay of 2 delay units. 11 Add dm2 delay of 3 delay units.
29–28 wr_dq23_del	Write dq23 delay fine tuning. This field holds the number of delay units that are added to dq23 relative to dqs2. 00 No change in dq23 delay 01 Add dq23 delay of 1 delay unit. 10 Add dq23 delay of 2 delay units. 11 Add dq23 delay of 3 delay units.
27–26 Reserved	This read-only field is reserved and always has the value 0.
25–24 wr_dq22_del	Write dq22 delay fine tuning. This field holds the number of delay units that are added to dq22 relative to dqs2. 00 No change in dq22 delay 01 Add dq22 delay of 1 delay unit. 10 Add dq22 delay of 2 delay units. 11 Add dq22 delay of 3 delay units.
23–22 Reserved	This read-only field is reserved and always has the value 0.
21–20 wr_dq21_del	Write dq21 delay fine tuning. This field holds the number of delay units that are added to dq21 relative to dqs2. 00 No change in dq21 delay 01 Add dq21 delay of 1 delay unit. 10 Add dq21 delay of 2 delay units. 11 Add dq21 delay of 3 delay units.
19–18 Reserved	This read-only field is reserved and always has the value 0.
17–16 wr_dq20_del	Write dq20 delay fine tuning. This field holds the number of delay units that are added to dq20 relative to dqs2. 00 No change in dq20 delay

Table continues on the next page...

MMDCx_MPWRDQBY2DL field descriptions (continued)

Field	Description
	01 Add dq20 delay of 1 delay unit. 10 Add dq20 delay of 2 delay units. 11 Add dq20 delay of 3 delay units.
15–14 Reserved	This read-only field is reserved and always has the value 0.
13–12 wr_dq19_del	Write dq19 delay fine tuning. This field holds the number of delay units that are added to dq19 relative to dqs2. 00 No change in dq19 delay 01 Add dq19 delay of 1 delay unit. 10 Add dq19 delay of 2 delay units. 11 Add dq19 delay of 3 delay units.
11–10 Reserved	This read-only field is reserved and always has the value 0.
9–8 wr_dq18_del	Write dq18 delay fine tuning. This field holds the number of delay units that are added to dq18 relative to dqs2. 00 No change in dq18 delay 01 Add dq18 delay of 1 delay unit. 10 Add dq18 delay of 2 delay units. 11 Add dq18 delay of 3 delay units.
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–4 wr_dq17_del	Write dq17 delay fine tuning. This field holds the number of delay units that are added to dq17 relative to dqs2. 00 No change in dq17 delay 01 Add dq17 delay of 1 delay unit. 10 Add dq17 delay of 2 delay units. 11 Add dq17 delay of 3 delay units.
3–2 Reserved	This read-only field is reserved and always has the value 0.
wr_dq16_del	Write dq16 delay fine tuning. This field holds the number of delay units that are added to dq16 relative to dqs2. 00 No change in dq16 delay 01 Add dq16 delay of 1 delay unit. 10 Add dq16 delay of 2 delay units. 11 Add dq16 delay of 3 delay units.

45.12.45 MMDC PHY Write DQ Byte3 Delay Register (MMDCx_MPWRDQBY3DL)

This register is used to add fine-tuning adjustment to every bit in the write DQ byte3 relative to the write DQS

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x32

Address: Base address + 838h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	wr_dm3_del		wr_dq31_del		0		wr_dq30_del		0		wr_dq29_del		0		wr_dq28_del	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		wr_dq27_del		0		wr_dq26_del		0		wr_dq25_del		0		wr_dq24_del	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPWRDQBY3DL field descriptions

Field	Description
31–30 wr_dm3_del	Write dm3 delay fine tuning. This field holds the number of delay units that are added to dm3 relative to dqs3. 00 No change in dm3 delay 01 Add dm3 delay of 1 delay unit. 10 Add dm3 delay of 2 delay units. 11 Add dm3 delay of 3 delay units.
29–28 wr_dq31_del	Write dq31 delay fine tuning. This field holds the number of delay units that are added to dq31 relative to dqs3. 00 No change in dq31 delay 01 Add dq31 delay of 1 delay unit. 10 Add dq31 delay of 2 delay units. 11 Add dq31 delay of 3 delay units.
27–26 Reserved	This read-only field is reserved and always has the value 0.
25–24 wr_dq30_del	Write dq30 delay fine tuning. This field holds the number of delay units that are added to dq30 relative to dqs3. 00 No change in dq30 delay 01 Add dq30 delay of 1 delay unit. 10 Add dq30 delay of 2 delay units. 11 Add dq30 delay of 3 delay units.
23–22 Reserved	This read-only field is reserved and always has the value 0.
21–20 wr_dq29_del	Write dq29 delay fine tuning. This field holds the number of delay units that are added to dq29 relative to dqs3. 00 No change in dq29 delay 01 Add dq29 delay of 1 delay unit. 10 Add dq29 delay of 2 delay units. 11 Add dq29 delay of 3 delay units.

Table continues on the next page...

MMDCx_MPWRDQBY3DL field descriptions (continued)

Field	Description
19–18 Reserved	This read-only field is reserved and always has the value 0.
17–16 wr_dq28_del	Write dq28 delay fine tuning. This field holds the number of delay units that are added to dq28 relative to dqs3. 00 No change in dq28 delay 01 Add dq28 delay of 1 delay unit. 10 Add dq28 delay of 2 delay units. 11 Add dq28 delay of 3 delay units.
15–14 Reserved	This read-only field is reserved and always has the value 0.
13–12 wr_dq27_del	Write dq27 delay fine tuning. This field holds the number of delay units that are added to dq27 relative to dqs3. 00 No change in dq27 delay 01 Add dq27 delay of 1 delay unit. 10 Add dq27 delay of 2 delay units. 11 Add dq27 delay of 3 delay units.
11–10 Reserved	This read-only field is reserved and always has the value 0.
9–8 wr_dq26_del	Write dq26 delay fine tuning. This field holds the number of delay units that are added to dq26 relative to dqs3. 00 No change in dq26 delay 01 Add dq26 delay of 1 delay unit. 10 Add dq26 delay of 2 delay units. 11 Add dq26 delay of 3 delay units.
7–6 Reserved	This read-only field is reserved and always has the value 0.
5–4 wr_dq25_del	Write dq25 delay fine tuning. This field holds the number of delay units that are added to dq25 relative to dqs3. 00 No change in dq25 delay 01 Add dq25 delay of 1 delay unit. 10 Add dq25 delay of 2 delay units. 11 Add dq25 delay of 3 delay units.
3–2 Reserved	This read-only field is reserved and always has the value 0.
wr_dq24_del	Write dq24 delay fine tuning. This field holds the number of delay units that are added to dq24 relative to dqs3. 00 No change in dq24 delay 01 Add dq24 delay of 1 delay unit. 10 Add dq24 delay of 2 delay units. 11 Add dq24 delay of 3 delay units.

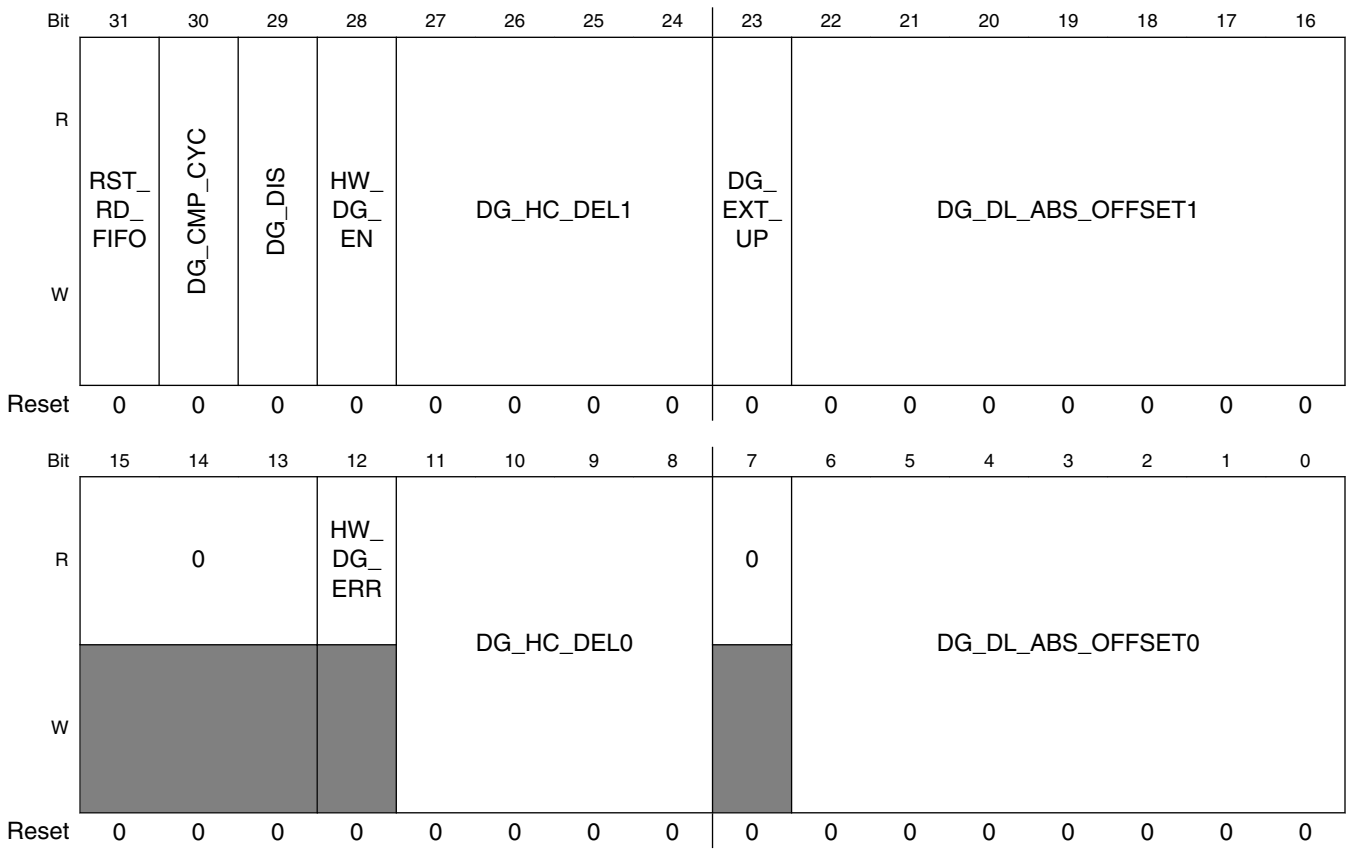
45.12.46 MMDC PHY Read DQS Gating Control Register 0 (MMDCx_MPDGCTRL0)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x32

Address: Base address + 83Ch offset



MMDCx_MPDGCTRL0 field descriptions

Field	Description
31 RST_RD_FIFO	Reset Read Data FIFO and associated pointers. If this bit is asserted then the MMDC resets the read data FIFO and the associated pointers. This bit is self cleared after the FIFO reset is done.
30 DG_CMP_CYC	Read DQS gating sample cycle. If this bit is asserted then the MMDC waits 32 cycles before comparing the read data, Otherwise it waits 16 DDR cycles. 0 MMDC waits 16 DDR cycles 1 MMDC waits 32 DDR cycles
29 DG_DIS	Read DQS gating disable. If this bit is asserted then the MMDC disables the read DQS gating mechanism.

Table continues on the next page...

MMDCx_MPDGCTRL0 field descriptions (continued)

Field	Description
	If this bits is asserted (read DQS gating is disabled) then pull-up and pull-down resistors suppose to be used on DQS and DQS# respectively 0 Read DQS gating mechanism is enabled 1 Read DQS gating mechanism is disabled
28 HW_DG_EN	Enable automatic read DQS gating calibration. If this bit is asserted then the MMDC performs automatic read DQS gating calibration. HW negates this bit upon completion of the automatic read DQS gating. **NOTE:** Before issuing the first read command the MMDC counts 12 cycles. In LPDDR2 mode automatic (HW) read DQS gating should be disabled and Pull-up/pull-down resistors on DQS/DQS# should be enabled while ODT resistors must be disconnected. 0 Disable automatic read DQS gating calibration 1 Start automatic read DQS gating calibration
27–24 DG_HC_DEL1	Read DQS gating half cycles delay for Byte1. (channel 0 register) and Byte5 in 64-bit mode (channel 1 register) This field indicates the delay in half cycles between read DQS gate and the middle of the read DQS preamble of Byte1. This delay is added to the delay that is generated by the read DQS1 gating delay-line, So the total read DQS gating delay is (DG_HC_DEL#)*0.5*cycle + (DG_DL_ABS_OFFSET#)*1/256*cycle Upon completion of the automatic read DQS gating calibration this field gets the value of the 4 MSB of ((HW_DG_LOW1 + HW_DG_UP1) /2). 0000 0 cycles delay. 0001 Half cycle delay. 0010 1 cycle delay 1101 6.5 cycles delay 1110 Reserved 1111 Reserved
23 DG_EXT_UP	DG extend upper boundary. By default the upper boundary of DQS gating HW calibration is set according to first failing comparison after at least one passing comparison. If this bit is asserted then the upper boundary is set according to the last passing comparison.
22–16 DG_DL_ABS_ OFFSET1	Absolute read DQS gating delay offset for Byte1. This field indicates the absolute delay between read DQS gate and the middle of the read DQS preamble of Byte1 with fractions of a clock period and up to half cycle. The fraction is process and frequency independent. The delay of the delay-line would be (DG_DL_ABS_OFFSET1 / 256)* MMDC_CH0 AXI clock (fast clock). This field can also bit written by HW. Upon completion of the automatic read DQS gating calibration this field gets the value of the 7 LSB of ((HW_DG_LOW1 + HW_DG_UP1) /2). **NOTE:** Not all changes will have effect on the actual delay. If the requested change is smaller than the delay-line resolution, then no change will occur.
15–13 Reserved	This read-only field is reserved and always has the value 0.
12 HW_DG_ERR	HW DQS gating error. This bit valid is asserted when an error was found during the read DQS gating HW calibration process. Error can occur when no valid value was found during HW calibration. This bit is valid only after HW_DG_EN is de-asserted. 0 No error was found during the DQS gating HW calibration process. 1 An error was found during the DQS gating HW calibration process.

Table continues on the next page...

MMDCx_MPDGCTRL0 field descriptions (continued)

Field	Description
11–8 DG_HC_DEL0	Read DQS gating half cycles delay for Byte0 (Channel 0 register) and Byte4 in 64-bit mode (Channel 1 register) This field indicates the delay in half cycles between read DQS gate and the middle of the read DQS preamble of Byte0/4. This delay is added to the delay that is generated by the read DQS1 gating delay-line, So the total read DQS gating delay is (DG_HC_DEL#)*0.5*cycle + (DG_DL_ABS_OFFSET#)*1/256*cycle Upon completion of the automatic read DQS gating calibration this field gets the value of the 4 MSB of ((HW_DG_LOW1 + HW_DG_UP1) /2). 0000 0 cycles delay. 0001 Half cycle delay. 0010 1 cycle delay 1101 6.5 cycles delay 1110 Reserved 1111 Reserved
7 Reserved	This read-only field is reserved and always has the value 0.
DG_DL_ABS_OFFSET0	Absolute read DQS gating delay offset for Byte0. This field indicates the absolute delay between read DQS gate and the middle of the read DQS preamble of Byte0 with fractions of a clock period and up to half cycle. The fraction is process and frequency independent. The delay of the delay-line would be (DG_DL_ABS_OFFSET0 / 256)* MMDC_CH0 AXI clock (fast clock). This field can also bit written by HW. Upon completion of the automatic read DQS gating calibration this field gets the value of the 7 LSB of ((HW_DG_LOW0 + HW_DG_UP0) /2). **NOTE:** Not all changes will have effect on the actual delay. If the requested change is smaller than the delay-line resolution, then no change will occur.

45.12.47 MMDC PHY Read DQS Gating Control Register 1 (MMDCx_MPDGCTRL1)

Supported Mode Of Operations:

For Channel 0: DDR3_x16, DDR3_x32, DDR3_x64

For Channel 1: DDR3_x64

Address: Base address + 840h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				DG_HC_DEL3				0	DG_DL_ABS_OFFSET3						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DG_HC_DEL2				0	DG_DL_ABS_OFFSET2						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPDGCTRL1 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–24 DG_HC_DEL3	Read DQS gating half cycles delay for Byte3 (Channel 0 register) and Byte7 for 64-bit data (Channel 1 register) . This field indicates the delay in half cycles between read DQS gate and the middle of the read DQS preamble of Byte3/7. This delay is added to the delay that is generated by the read DQS1 gating delay-line, So the total read DQS gating delay is (DG_HC_DEL#)*0.5*cycle + (DG_DL_ABS_OFFSET#)*1/256*cycle Upon completion of the automatic read DQS gating calibration this field gets the value of the 4 MSB of ((HW_DG_LOW3 + HW_DG_UP3) /2). 0000 0 cycles delay. 0001 Half cycle delay. 0010 1 cycle delay 1101 6.5 cycles delay 1110 Reserved 1111 Reserved
23 Reserved	This read-only field is reserved and always has the value 0.
22–16 DG_DL_ABS_ OFFSET3	Absolute read DQS gating delay offset for Byte3. This field indicates the absolute delay between read DQS gate and the middle of the read DQS preamble of Byte3 with fractions of a clock period and up to half cycle. The fraction is process and frequency independent. The delay of the delay-line would be (DG_DL_ABS_OFFSET3 / 256)* MMDC_CH0 AXI clock (fast clock). This field can also bit written by HW. Upon completion of the automatic read DQS gating calibration this field gets the value of the 7 LSB of ((HW_DG_LOW3 + HW_DG_UP3) /2). **NOTE:** Not all changes will have effect on the actual delay. If the requested change is smaller than the delay-line resolution, then no change will occur.
15–12 Reserved	This read-only field is reserved and always has the value 0.
11–8 DG_HC_DEL2	Read DQS gating half cycles delay for Byte2 (Channel 0 register) and Byte6 for 64-bit mode(channel 1 register) . This field indicates the delay in half cycles between read DQS gate and the middle of the read DQS preamble of Byte2/5. This delay is added to the delay that is generated by the read DQS1 gating delay-line, So the total read DQS gating delay is (DG_HC_DEL#)*0.5*cycle + (DG_DL_ABS_OFFSET#)*1/256*cycle Upon completion of the automatic read DQS gating calibration this field gets the value of the 4 MSB of ((HW_DG_LOW2 + HW_DG_UP2) /2). 0000 0 cycles delay. 0001 Half cycle delay. 0010 1 cycle delay 1101 6.5 cycles delay 1110 Reserved 1111 Reserved
7 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

MMDCx_MPDGCTRL1 field descriptions (continued)

Field	Description
DG_DL_ABS_ OFFSET2	Absolute read DQS gating delay offset for Byte2. This field indicates the absolute delay between read DQS gate and the middle of the read DQS preamble of Byte2 with fractions of a clock period and up to half cycle. The fraction is process and frequency independent. The delay of the delay-line would be (DG_DL_ABS_OFFSET2 / 256)* MMDC_CH0 AXI clock (fast clock). This field can also bit written by HW. Upon completion of the automatic read DQS gating calibration this field gets the value of the 7 LSB of ((HW_DG_LOW2 + HW_DG_UP2) /2). **NOTE:** Not all changes will have effect on the actual delay. If the requested change is smaller than the delay-line resolution, then no change will occur.

45.12.48 MMDC PHY Read DQS Gating delay-line Status Register (MMDCx_MPDGDLST0)

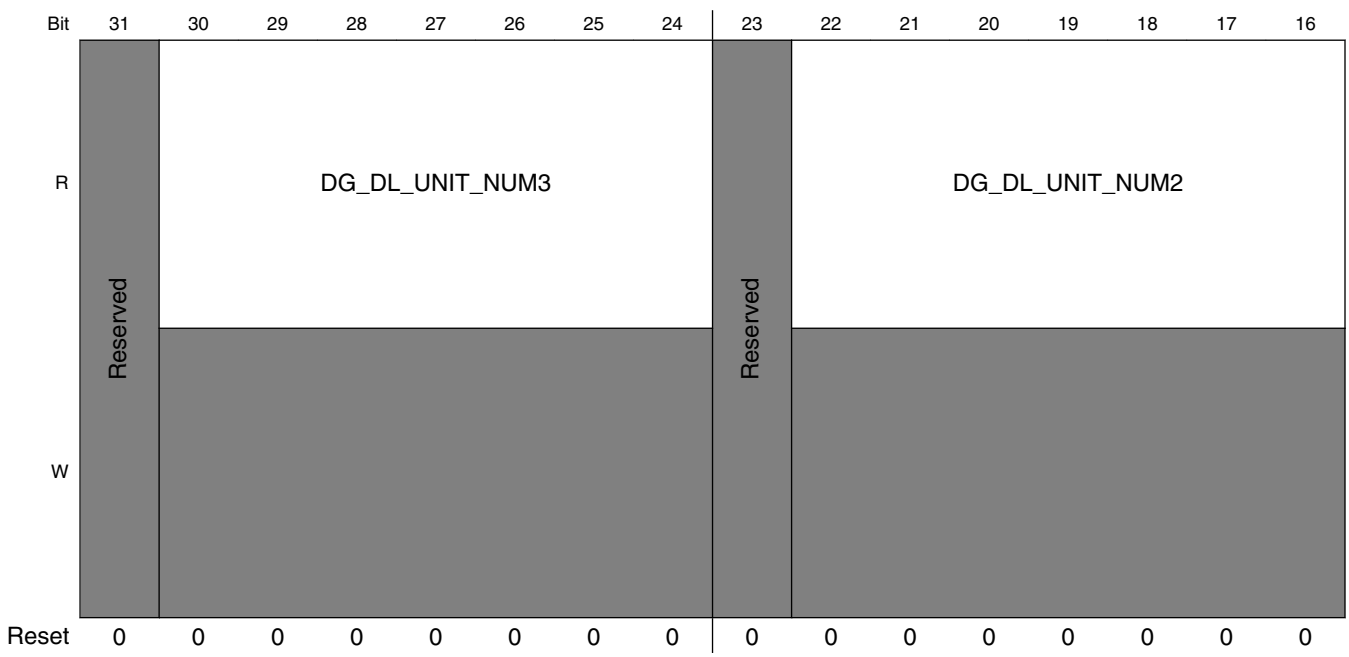
This register holds the status of the 4 dqs gating delay-lines.

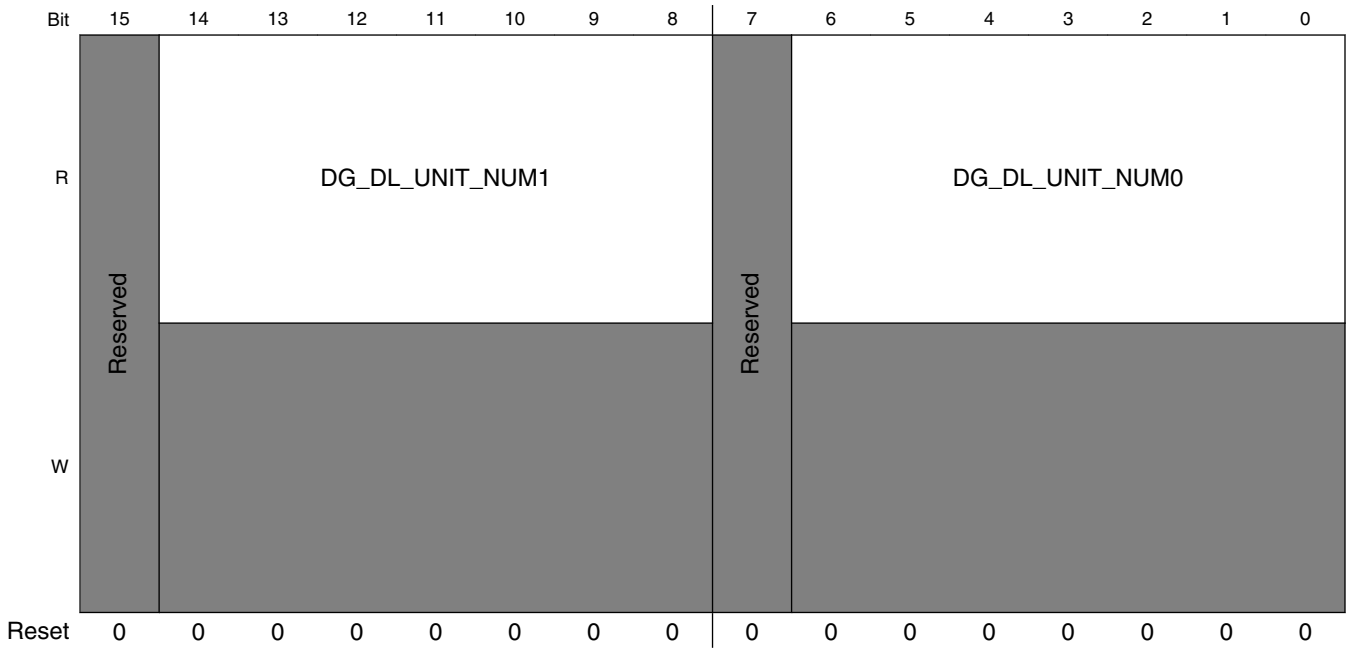
Supported Mode Of Operations:

For Channel 0: DDR3_x16, DDR3_x32, DDR3_x64

For Channel 1: DDR3_x64

Address: Base address + 844h offset





MMDC*x*_MPDGDLST0 field descriptions

Field	Description
31 -	This field is reserved. Reserved
30–24 DG_DL_UNIT_NUM3	This field reflects the number of delay units that are actually used by read DQS gating delay-line 3.
23 -	This field is reserved. Reserved
22–16 DG_DL_UNIT_NUM2	This field reflects the number of delay units that are actually used by read DQS gating delay-line 2.
15 -	This field is reserved. Reserved
14–8 DG_DL_UNIT_NUM1	This field reflects the number of delay units that are actually used by read DQS gating delay-line 1.
7 -	This field is reserved. Reserved
DG_DL_UNIT_NUM0	This field reflects the number of delay units that are actually used by read DQS gating delay-line 0.

45.12.49 MMDC PHY Read delay-lines Configuration Register (MMDCx_MPRDDLCTL)

This register controls read delay-lines functionality; it determines DQS delay relative to the associated DQ read access. The delay-line compensates for process variations and produces a constant delay regardless of the process, temperature and voltage.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 848h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	RD_DL_ABS_OFFSET3							0	RD_DL_ABS_OFFSET2						
W																
Reset	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	RD_DL_ABS_OFFSET1							0	RD_DL_ABS_OFFSET0						
W																
Reset	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0

MMDCx_MPRDDLCTL field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–24 RD_DL_ABS_ OFFSET3	Absolute read delay offset for Byte3. This field indicates the absolute delay between read DQS strobe and the read data of Byte3 with fractions of a clock period and up to half cycle. The fraction is process and frequency independent. The delay of the delay-line would be (RD_DL_ABS_OFFSET3 / 256) * MMDC_CH0 AXI clock (fast clock). So for the default value of 64 we get a quarter cycle delay. This field can also bit written by HW. Upon completion of the read delay-line HW calibration this field gets the value of (HW_RD_DL_LOW3 + HW_RD_DL_UP3) /2 **NOTE:** Not all changes will have effect on the actual delay. If the requested change is smaller than the delay-line resolution, then no change will occur.
23 Reserved	This read-only field is reserved and always has the value 0.
22–16 RD_DL_ABS_ OFFSET2	Absolute read delay offset for Byte2. This field indicates the absolute delay between read DQS strobe and the read data of Byte2 with fractions of a clock period and up to half cycle. The fraction is process and frequency independent. The delay of the delay-line would be (RD_DL_ABS_OFFSET2 / 256) * MMDC_CH0 AXI clock (fast clock). So for the default value of 64 we get a quarter cycle delay. This field can also bit written by HW. Upon completion of the read delay-line HW calibration this field gets the value of (HW_RD_DL_LOW2 + HW_RD_DL_UP2) /2 **NOTE:** Not all changes will have effect on the actual delay. If the requested change is smaller than the delay-line resolution, then no change will occur.

Table continues on the next page...

MMDCx_MPRDDLCTL field descriptions (continued)

Field	Description
15 Reserved	This read-only field is reserved and always has the value 0.
14–8 RD_DL_ABS_ OFFSET1	Absolute read delay offset for Byte1. This field indicates the absolute delay between read DQS strobe and the read data of Byte1 with fractions of a clock period and up to half cycle. The fraction is process and frequency independent. The delay of the delay-line would be (RD_DL_ABS_OFFSET1 / 256) * MMDC_CH0 AXI clock (fast clock). So for the default value of 64 we get a quarter cycle delay. This field can also bit written by HW. Upon completion of the read delay-line HW calibration this field gets the value of (HW_RD_DL_LOW1 + HW_RD_DL_UP1) /2 **NOTE:** Not all changes will have effect on the actual delay. If the requested change is smaller than the delay-line resolution, then no change will occur.
7 Reserved	This read-only field is reserved and always has the value 0.
RD_DL_ABS_ OFFSET0	Absolute read delay offset for Byte0. This field indicates the absolute delay between read DQS strobe and the read data of Byte0 with fractions of a clock period and up to half cycle. The fraction is process and frequency independent. The delay of the delay-line would be (RD_DL_ABS_OFFSET0 / 256) * MMDC_CH0 AXI clock (fast clock). So for the default value of 64 we get a quarter cycle delay. This field can also bit written by HW. Upon completion of the read delay-line HW calibration this field gets the value of (HW_RD_DL_LOW0 + HW_RD_DL_UP0) /2 **NOTE:** Not all changes will have effect on the actual delay. If the requested change is smaller than the delay-line resolution, then no change will occur.

45.12.50 MMDC PHY Read delay-lines Status Register (MMDCx_MPRDDLST)

This register holds the status of the 4 read delay-lines.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 84Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	RD_DL_UNIT_NUM3							0	RD_DL_UNIT_NUM2						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	RD_DL_UNIT_NUM1							0	RD_DL_UNIT_NUM0						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPRDDLST field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–24 RD_DL_UNIT_NUM3	This field reflects the number of delay units that are actually used by read delay-line 3.
23 Reserved	This read-only field is reserved and always has the value 0.
22–16 RD_DL_UNIT_NUM2	This field reflects the number of delay units that are actually used by read delay-line 2.
15 Reserved	This read-only field is reserved and always has the value 0.
14–8 RD_DL_UNIT_NUM1	This field reflects the number of delay units that are actually used by read delay-line 1.
7 Reserved	This read-only field is reserved and always has the value 0.
RD_DL_UNIT_NUM0	This field reflects the number of delay units that are actually used by read delay-line 0.

45.12.51 MMDC PHY Write delay-lines Configuration Register (MMDCx_MPWRDLCTL)

This register controls write delay-lines functionality, it determines DQ/DM delay relative to the associated DQS in write access. The delay-line compensates for process variations, and produces a constant delay regardless of the process, temperature and voltage.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 850h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	WR_DL_ABS_OFFSET3							0	WR_DL_ABS_OFFSET2						
W																
Reset	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	WR_DL_ABS_OFFSET1							0	WR_DL_ABS_OFFSET0						
W																
Reset	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0

MMDCx_MPWRDLCTL field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–24 WR_DL_ABS_ OFFSET3	Absolute write delay offset for Byte3. This field indicates the absolute delay between write DQS strobe and the write data of Byte3 with fractions of a clock period and up to half cycle. The fraction is process and frequency independent. The delay of the delay-line would be (WR_DL_ABS_OFFSET3 / 256) * MMDC_CH0 AXI clock (fast clock). So for the default value of 64 we get a quarter cycle delay. This field can also bit written by HW. Upon completion of the write delay-line HW calibration this field gets the value of (HW_WR_DL_LOW3 + HW_WR_DL_UP3) /2 **NOTE:** Not all changes will have effect on the actual delay. If the requested change is smaller than the delay-line resolution, then no change will occur.
23 Reserved	This read-only field is reserved and always has the value 0.
22–16 WR_DL_ABS_ OFFSET2	Absolute write delay offset for Byte2. This field indicates the absolute delay between write DQS strobe and the write data of Byte2 with fractions of a clock period and up to half cycle. The fraction is process and frequency independent. The delay of the delay-line would be (WR_DL_ABS_OFFSET2/ 256) * MMDC_CH0 AXI clock (fast clock). So for the default value of 64 we get a quarter cycle delay. This field can also bit written by HW. Upon completion of the write delay-line HW calibration this field gets the value of (HW_WR_DL_LOW2 + HW_WR_DL_UP2) /2 **NOTE:** Not all changes will have effect on the actual delay. If the requested change is smaller than the delay-line resolution, then no change will occur.
15 Reserved	This read-only field is reserved and always has the value 0.
14–8 WR_DL_ABS_ OFFSET1	Absolute write delay offset for Byte1. This field indicates the absolute delay between write DQS strobe and the write data of Byte1 with fractions of a clock period and up to half cycle. The fraction is process and frequency independent. The delay of the delay-line would be (WR_DL_ABS_OFFSET1 / 256) * MMDC_CH0 AXI clock (fast clock). So for the default value of 64 we get a quarter cycle delay. This field can also bit written by HW. Upon completion of the write delay-line HW calibration this field gets the value of (HW_WR_DL_LOW1 + HW_WR_DL_UP1) /2 **NOTE:** Not all changes of this value will affect the actual delay. If the requested change is smaller than the delay-line resolution, then no change will occur.
7 Reserved	This read-only field is reserved and always has the value 0.
WR_DL_ABS_ OFFSET0	Absolute write delay offset for Byte0. This field indicates the absolute delay between write DQS strobe and the write data of Byte3 with fractions of a clock period and up to half cycle. The fraction is process and frequency independent. The delay of the delay-line would be (WR_DL_ABS_OFFSET0 / 256) * MMDC_CH0 AXI clock (fast clock). So for the default value of 64 we get a quarter cycle delay. This field can also bit written by HW. Upon completion of the write delay-line HW calibration this field gets the value of (HW_WR_DL_LOW0 + HW_WR_DL_UP0) /2 **NOTE:** Not all changes of this value will affect the actual delay. If the requested change is smaller than the delay-line resolution, then no change will occur.

45.12.52 MMDC PHY Write delay-lines Status Register (MMDCx_MPWRDLST)

This register holds the status of the 4 write delay-line.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 854h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	WR_DL_UNIT_NUM3							0	WR_DL_UNIT_NUM2						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	WR_DL_UNIT_NUM1							0	WR_DL_UNIT_NUM0						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPWRDLST field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–24 WR_DL_UNIT_NUM3	This field reflects the number of delay units that are actually used by write delay-line 3.
23 Reserved	This read-only field is reserved and always has the value 0.
22–16 WR_DL_UNIT_NUM2	This field reflects the number of delay units that are actually used by write delay-line 2.
15 Reserved	This read-only field is reserved and always has the value 0.
14–8 WR_DL_UNIT_NUM1	This field reflects the number of delay units that are actually used by write delay-line 1.
7 Reserved	This read-only field is reserved and always has the value 0.
WR_DL_UNIT_NUM0	This field reflects the number of delay units that are actually used by write delay-line 0.

45.12.53 MMDC PHY CK Control Register (MMDCx_MPSDCTRL)

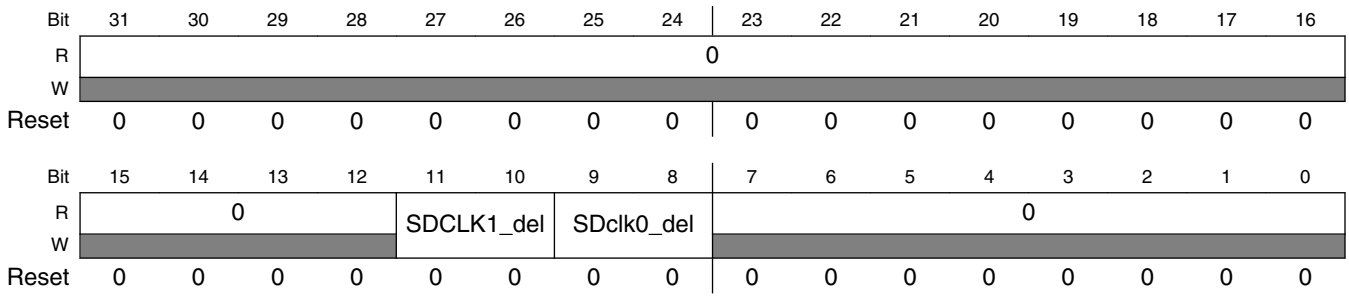
This register controls the fine tuning of the primary clock (CK0).

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 858h offset



MMDCx_MPSDCTRL field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value 0.
11–10 SDCLK1_del	DDR clock1 delay fine tuning. This field holds the number of delay units that are added to DDR clock1 (CK1). 00 No change in DDR clock delay 01 Add DDR clock delay of 1 delay unit. 10 Add DDR clock delay of 2 delay units. 11 Add DDR clock delay of 3 delay units.
9–8 SDclk0_del	DDR clock0 delay fine tuning. This field holds the number of delay units that are added to DDR clock (CK0). **NOTE:** In case of LPDDR2 2-ch mode this registers controls the fine tuning of the clock that is driven to channel0 In case of DDR3 the fine tuning of the secondary clock is controlled by 0x021B_4858[SDCLK] 00 No change in DDR clock0 delay 01 Add DDR clock0 delay of 1 delay unit. 10 Add DDR clock0 delay of 2 delay units. 11 Add DDR clock0 delay of 3 delay units.
Reserved	This read-only field is reserved and always has the value 0.

45.12.54 MMDC ZQ LPDDR2 HW Control Register (MMDCx_MPZQLP2CTL)

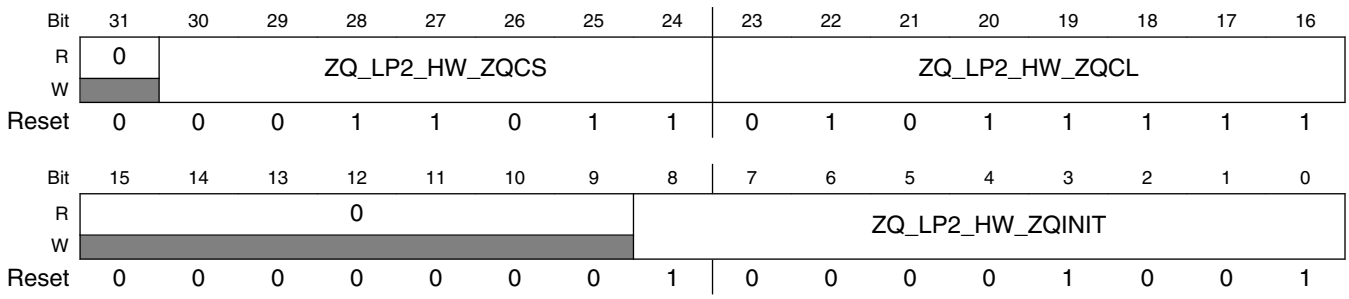
This register controls the idle time that takes the LPDDR2 device to perform ZQ calibration.

Supported Mode Of Operations:

For Channel 0: LP2_2ch_x16, LP2_2ch_x32

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 85Ch offset



MMDCx_MPZQLP2CTL field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–24 ZQ_LP2_HW_ZQCS	This register defines the period in cycles that it takes the memory device to perform a short ZQ calibration. This is the period of time that the MMDC has to wait after sending a short ZQ calibration and before sending other commands. This delay will also be used if ZQ reset is sent. 0x0-0x1A Reserved 0x1B 112 cycles (default) 0x1C 116 cycles 0x7E 508 cycles 0x7F 512 cycles
23–16 ZQ_LP2_HW_ZQCL	This register defines the period in cycles that it takes the memory device to perform a long ZQ calibration. This is the period of time that the MMDC has to wait after sending a long ZQ calibration and before sending other commands. 0x0-0x36 Reserved 0x37 112 cycles 0x38 114 cycles 0x5F 192 cycles (Default. This may need to be adjusted to achieve the correct timing depending on the DDR clock frequency.)

Table continues on the next page...

MMDCx_MPZQLP2CTL field descriptions (continued)

Field	Description
	0xFE 510 cycles 0xFF 512 cycles
15–9 Reserved	This read-only field is reserved and always has the value 0.
ZQ_LP2_HW_ZQINIT	This register defines the period in cycles that it takes the memory device to perform a Init ZQ calibration. This is the period of time that the MMDC has to wait after sending a init ZQ calibration and before sending other commands. 0x0-0x36 Reserved 0x37 112 cycles 0x38 114 cycles 0x109 532 cycles (Default. This may need to be adjusted to achieve the correct timing depending on the DDR clock frequency.) 0x1FE 1022 cycles 0x1FF 1024 cycles

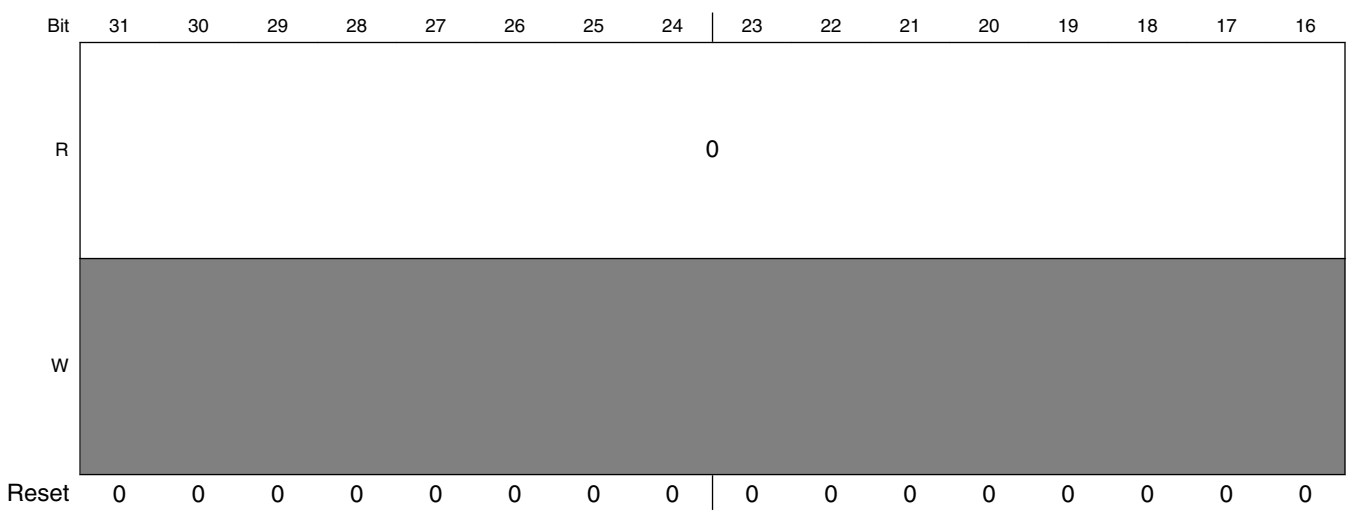
45.12.55 MMDC PHY Read Delay HW Calibration Control Register (MMDCx_MPRDDLHWCTL)

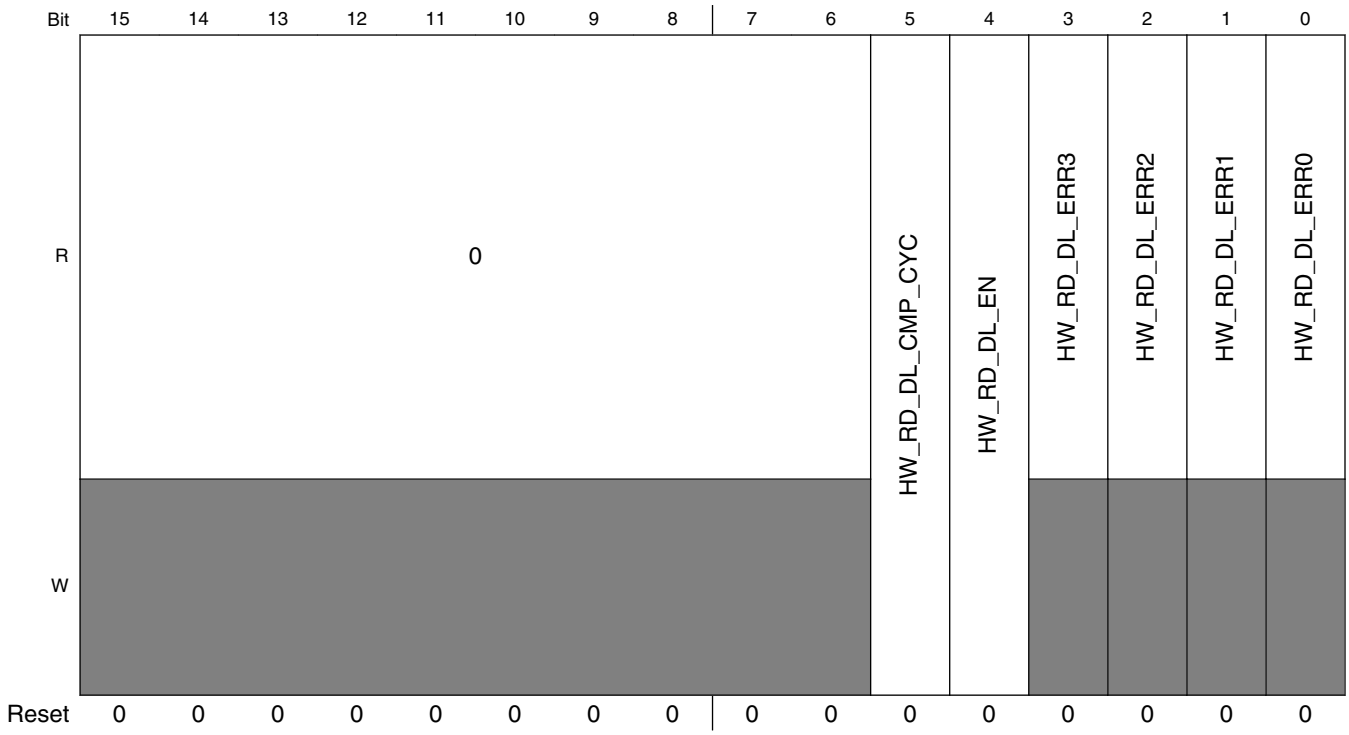
Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 860h offset





MMDCx_MPRDDLHWCTL field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.
5 HW_RD_DL_ CMP_CYC	Automatic (HW) read sample cycle. If this bit is asserted then the MMDC will compare the read data 32 cycles after the MMDC sent the read command enable pulse else it compares the data after 16 cycles.
4 HW_RD_DL_EN	Enable automatic (HW) read calibration. If this bit is asserted then the MMDC will perform an automatic read calibration. HW should negate this bit upon completion of the calibration. Negation of this bit also points that the read calibration results are valid **NOTE:** Before issuing the first read command MMDC counts 12 cycles.
3 HW_RD_DL_ ERR3	Automatic (HW) read calibration error of Byte3. If this bit is asserted then it indicates that an error was found during the HW calibration process of read delay-line 3. In case this bit is zero at the end of the calibration process then the boundary results can be found at MPRDDLHWST1 register. This bit is valid only after HW_RD_DL_EN is de-asserted. 0 No error was found in read delay-line 3 during the automatic (HW) read calibration process of read delay-line 3. 1 An error was found in read delay-line 3 during the automatic (HW) read calibration process of read delay-line 3.
2 HW_RD_DL_ ERR2	Automatic (HW) read calibration error of Byte2. If this bit is asserted then it indicates that an error was found during the HW calibration process of read delay-line 2. In case this bit is zero at the end of the calibration process then the boundary results can be found at MPRDDLHWST1 register. This bit is valid only after HW_RD_DL_EN is de-asserted.

Table continues on the next page...

MMDCx_MPRDDLHWCTL field descriptions (continued)

Field	Description
	0 No error was found in read delay-line 2 during the automatic (HW) read calibration process of read delay-line 2. 1 An error was found in read delay-line 2 during the automatic (HW) read calibration process of read delay-line 2.
1 HW_RD_DL_ERR1	Automatic (HW) read calibration error of Byte1. If this bit is asserted then it indicates that an error was found during the HW calibration process of read delay-line 1. In case this bit is zero at the end of the calibration process then the boundary results can be found at MPRDDLHWST0 register. This bit is valid only after HW_RD_DL_EN is de-asserted. 0 No error was found in read delay-line 1 during the automatic (HW) read calibration process of read delay-line 1. 1 An error was found in read delay-line 1 during the automatic (HW) read calibration process of read delay-line 1.
0 HW_RD_DL_ERR0	Automatic (HW) read calibration error of Byte0. If this bit is asserted then it indicates that an error was found during the HW calibration process of read delay-line 0. In case this bit is zero at the end of the calibration process then the boundary results can be found at MPRDDLHWST0 register. This bit is valid only after HW_RD_DL_EN is de-asserted. 0 No error was found in read delay-line 0 during the automatic (HW) read calibration process of read delay-line 0. 1 An error was found in read delay-line 0 during the automatic (HW) read calibration process of read delay-line 0.

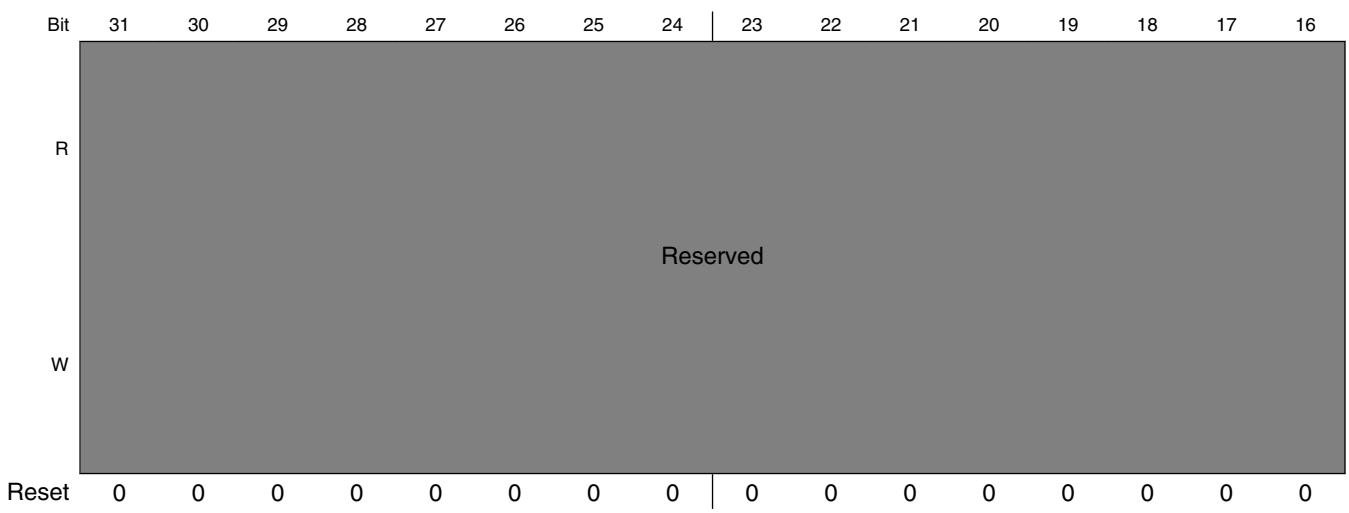
45.12.56 MMDC PHY Write Delay HW Calibration Control Register (MMDCx_MPWRDLHWCTL)

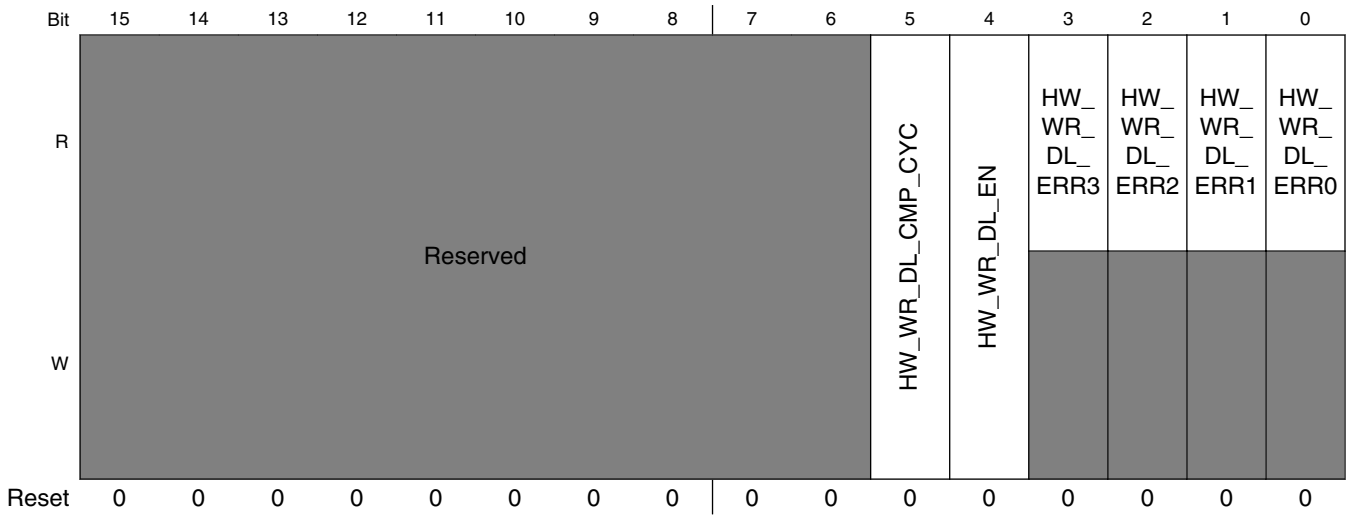
Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 864h offset





MMDCx_MPWRDLHWCTL field descriptions

Field	Description
31–6 -	This field is reserved.
5 HW_WR_DL_CMP_CYC	Write sample cycle. If this bit is asserted then the MMDC will compare the data 32 cycles after the MMDC sent the read command enable pulse else it compares the data after 16 cycles.
4 HW_WR_DL_EN	Enable automatic (HW) write calibration. If this bit is asserted then the MMDC will perform an automatic write calibration. HW should negate this bit upon completion of the calibration. Negation of this bit also indicates that the write calibration results are valid **NOTE:** Before issuing the first read command MMDC counts 12 cycles.
3 HW_WR_DL_ERR3	Automatic (HW) write calibration error of Byte3. If this bit is asserted then it indicates that an error was found during the HW calibration process of write delay-line 3. In case this bit is zero at the end of the calibration process then the boundary results can be found at MPWRDLHWST1 register. This bit is valid only after HW_WR_DL_EN is de-asserted. 0 No error was found during the automatic (HW) write calibration process of write delay-line 3. 1 An error was found during the automatic (HW) write calibration process of write delay-line 3.
2 HW_WR_DL_ERR2	Automatic (HW) write calibration error of Byte2. If this bit is asserted then it indicates that an error was found during the HW calibration process of write delay-line 2. T In case this bit is zero at the end of the calibration process then the boundary results can be found at MPWRDLHWST1 register. This bit is valid only after HW_WR_DL_EN is de-asserted. 0 No error was found during the automatic (HW) write calibration process of write delay-line 2. 1 An error was found during the automatic (HW) write calibration process of write delay-line 2.
1 HW_WR_DL_ERR1	Automatic (HW) write calibration error of Byte1. If this bit is asserted then it indicates that an error was found during the HW calibration process of write delay-line 1. In case this bit is zero at the end of the calibration process then the boundary results can be found at MPWRDLHWST0 register. This bit is valid only after HW_WR_DL_EN is de-asserted. 0 No error was found during the automatic (HW) write calibration process of write delay-line 1. 1 An error was found during the automatic (HW) write calibration process of write delay-line 1.
0 HW_WR_DL_ERR0	Automatic (HW) write calibration error of Byte0. If this bit is asserted then it indicates that an error was found during the HW calibration process of write delay-line 0. In case this bit is zero at the end of the

Table continues on the next page...

MMDCx_MPWRDLHWCTL field descriptions (continued)

Field	Description
	calibration process then the boundary results can be found at MPWRDLHWST0 register. This bit is valid only after HW_WR_DL_EN is de-asserted. 0 No error was found during the automatic (HW) write calibration process of write delay-line 0. 1 An error was found during the automatic (HW) write calibration process of write delay-line 0.

45.12.57 MMDC PHY Read Delay HW Calibration Status Register 0 (MMDCx_MPRDDLHWST0)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 868h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	HW_RD_DL_UP1							0	HW_RD_DL_LOW1						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	HW_RD_DL_UP0							0	HW_RD_DL_LOW0						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPRDDLHWST0 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–24 HW_RD_DL_ UP1	Automatic (HW) read calibration result of the upper boundary of Byte1. This field holds the automatic (HW) read calibration result of the upper boundary of Byte1
23 Reserved	This read-only field is reserved and always has the value 0.
22–16 HW_RD_DL_ LOW1	Automatic (HW) read calibration result of the lower boundary of Byte1. This field holds the automatic (HW) read calibration result of the lower boundary of Byte1
15 Reserved	This read-only field is reserved and always has the value 0.
14–8 HW_RD_DL_ UP0	Automatic (HW) read calibration result of the upper boundary of Byte0. This field holds the automatic (HW) read calibration result of the upper boundary of Byte0.

Table continues on the next page...

MMDCx_MPRDDLHWST0 field descriptions (continued)

Field	Description
7 Reserved	This read-only field is reserved and always has the value 0.
HW_RD_DL_ LOW0	Automatic (HW) read calibration result of the lower boundary of Byte0. This field holds the automatic (HW) read calibration result of the lower boundary of Byte0.

45.12.58 MMDC PHY Read Delay HW Calibration Status Register 1 (MMDCx_MPRDDLHWST1)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 86Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	HW_RD_DL_UP3							0	HW_RD_DL_LOW3						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	HW_RD_DL_UP2							0	HW_RD_DL_LOW2						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPRDDLHWST1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–24 HW_RD_DL_ UP3	Automatic (HW) read calibration result of the upper boundary of Byte3. This field holds the automatic (HW) read calibration result of the upper boundary of Byte3
23 Reserved	This read-only field is reserved and always has the value 0.
22–16 HW_RD_DL_ LOW3	Automatic (HW) read calibration result of the lower boundary of Byte3. This field holds the automatic (HW) read calibration result of the lower boundary of Byte3
15 Reserved	This read-only field is reserved and always has the value 0.
14–8 HW_RD_DL_ UP2	Automatic (HW) read calibration result of the upper boundary of Byte2. This field holds the automatic (HW) read calibration result of the upper boundary of Byte2.

Table continues on the next page...

MMDCx_MPRDDLHWST1 field descriptions (continued)

Field	Description
7 Reserved	This read-only field is reserved and always has the value 0.
HW_RD_DL_ LOW2	Automatic (HW) read calibration result of the lower boundary of Byte2. This field holds the automatic (HW) read calibration result of the lower boundary of Byte2.

45.12.59 MMDC PHY Write Delay HW Calibration Status Register 0 (MMDCx_MPWRDLHWST0)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 870h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	HW_WR_DL_UP1							0	HW_WR_DL_LOW1						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	HW_WR_DL_UP0							0	HW_WR_DL_LOW0						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPWRDLHWST0 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–24 HW_WR_DL_ UP1	Automatic (HW) write automatic (HW) write calibration result of the upper boundary of Byte1. This field holds the automatic (HW) write calibration result of the upper boundary of Byte1.
23 Reserved	This read-only field is reserved and always has the value 0.
22–16 HW_WR_DL_ LOW1	Automatic (HW) write calibration result of the lower boundary of Byte1. This field holds the automatic (HW) write calibration result of the lower boundary of Byte1.
15 Reserved	This read-only field is reserved and always has the value 0.
14–8 HW_WR_DL_ UP0	Automatic (HW) write calibration result of the upper boundary of Byte0. This field holds the automatic (HW) write calibration result of the upper boundary of Byte0.

Table continues on the next page...

MMDCx_MPWRDLHWST0 field descriptions (continued)

Field	Description
7 Reserved	This read-only field is reserved and always has the value 0.
HW_WR_DL_ LOW0	Automatic (HW) write calibration result of the lower boundary of Byte0. This field holds the automatic (HW) write calibration result of the lower boundary of Byte0.

45.12.60 MMDC PHY Write Delay HW Calibration Status Register 1 (MMDCx_MPWRDLHWST1)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x32

Address: Base address + 874h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	HW_WR_DL_UP3							0	HW_WR_DL_LOW3						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	HW_WR_DL_UP2							0	HW_WR_DL_LOW2						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPWRDLHWST1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–24 HW_WR_DL_ UP3	Automatic (HW) write calibration result of the upper boundary of Byte3. This field holds the automatic (HW) write calibration result of the upper boundary of Byte3.
23 Reserved	This read-only field is reserved and always has the value 0.
22–16 HW_WR_DL_ LOW3	Automatic (HW) write calibration result of the lower boundary of Byte3. This field holds the automatic (HW) write calibration result of the lower boundary of Byte3.
15 Reserved	This read-only field is reserved and always has the value 0.
14–8 HW_WR_DL_ UP2	Automatic (HW) write calibration result of the upper boundary of Byte2. This field holds the automatic (HW) write calibration result of the upper boundary of Byte2.

Table continues on the next page...

MMDCx_MPWRDLHWST1 field descriptions (continued)

Field	Description
7 Reserved	This read-only field is reserved and always has the value 0.
HW_WR_DL_ LOW2	Automatic (HW) write calibration result of the lower boundary of Byte2. This field holds the automatic (HW) write calibration result of the lower boundary of Byte2.

45.12.61 MMDC PHY Write Leveling HW Error Register (MMDCx_MPWLHWERR)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 878h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HW_WL3_DQ								HW_WL2_DQ								HW_WL1_DQ								HW_WL0_DQ							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPWLHWERR field descriptions

Field	Description
31–24 HW_WL3_DQ	HW write-leveling calibration result of Byte3. This field holds the results for all the 8 write-leveling steps of Byte3. i.e. bit 0 holds the result of the write-leveling calibration of 0 delay, bit 1holds the result of the write-leveling calibration of 1/8delay till bit 7 that holds the result of the write-leveling calibration of 7/8 delay
23–16 HW_WL2_DQ	HW write-leveling calibration result of Byte2. This field holds the results for all the 8 write-leveling steps of Byte2. i.e. bit 0 holds the result of the write-leveling calibration of 0 delay, bit 1holds the result of the write-leveling calibration of 1/8delay till bit 7 that holds the result of the write-leveling calibration of 7/8 delay
15–8 HW_WL1_DQ	HW write-leveling calibration result of Byte1. This field holds the results for all the 8 write-leveling steps of Byte1. i.e. bit 0 holds the result of the write-leveling calibration of 0 delay, bit 1holds the result of the write-leveling calibration of 1/8delay till bit 7 that holds the result of the write-leveling calibration of 7/8 delay
HW_WL0_DQ	HW write-leveling calibration result of Byte0. This field holds the results for all the 8 write-leveling steps of Byte0. i.e. bit 0 holds the result of the write-leveling calibration of 0 delay, bit 1holds the result of the write-leveling calibration of 1/8delay till bit 7 that holds the result of the write-leveling calibration of 7/8 delay

45.12.62 MMDC PHY Read DQS Gating HW Status Register 0 (MMDCx_MPDGHWST0)

Supported Mode Of Operations:

For Channel 0: DDR3_x16, DDR3_x32, DDR3_x64

For Channel 1: DDR3_x64

Address: Base address + 87Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					HW_DG_UP0											Reserved					HW_DG_LOW0										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPDGHWST0 field descriptions

Field	Description
31–27 -	This field is reserved. Reserved
26–16 HW_DG_UP0	HW DQS gating calibration result of the upper boundary of Byte0. This field holds the HW DQS gating calibration result of the upper boundary of Byte0.
15–11 -	This field is reserved. Reserved
HW_DG_LOW0	HW DQS gating calibration result of the lower boundary of Byte0. This field holds the HW DQS gating calibration result of the lower boundary of Byte0.

45.12.63 MMDC PHY Read DQS Gating HW Status Register 1 (MMDCx_MPDGHWST1)

Supported Mode Of Operations:

For Channel 0: DDR3_x16, DDR3_x32, DDR3_x64

For Channel 1: DDR3_x64

Address: Base address + 880h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					HW_DG_UP1											Reserved					HW_DG_LOW1										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPDGHWST1 field descriptions

Field	Description
31–27 -	This field is reserved. Reserved
26–16 HW_DG_UP1	HW DQS gating calibration result of the upper boundary of Byte1. This field holds the HW DQS gating calibration result of the upper boundary of Byte1.
15–11 -	This field is reserved. Reserved
HW_DG_LOW1	HW DQS gating calibration result of the lower boundary of Byte1. This field holds the HW DQS gating calibration result of the lower boundary of Byte1.

45.12.64 MMDC PHY Read DQS Gating HW Status Register 2 (MMDCx_MPDGHWST2)

Supported Mode Of Operations:

For Channel 0: DDR3_x16, DDR3_x32, DDR3_x64

For Channel 1: DDR3_x64

Address: Base address + 884h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					HW_DG_UP2											Reserved					HW_DG_LOW2										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPDGHWST2 field descriptions

Field	Description
31–27 -	This field is reserved. Reserved
26–16 HW_DG_UP2	HW DQS gating calibration result of the upper boundary of Byte2. This field holds the HW DQS gating calibration result of the upper boundary of Byte2.
15–11 -	This field is reserved. Reserved
HW_DG_LOW2	HW DQS gating calibration result of the lower boundary of Byte2. This field holds the HW DQS gating calibration result of the lower boundary of Byte2.

45.12.65 MMDC PHY Read DQS Gating HW Status Register 3 (MMDCx_MPDGHWST3)

Supported Mode Of Operations:

For Channel 0: DDR3_x16, DDR3_x32, DDR3_x64

For Channel 1: DDR3_x64

Address: Base address + 888h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					HW_DG_UP3											Reserved					HW_DG_LOW3										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPDGHWST3 field descriptions

Field	Description
31–27 -	This field is reserved. Reserved
26–16 HW_DG_UP3	HW DQS gating calibration result of the upper boundary of Byte3. This field holds the HW DQS gating calibration result of the upper boundary of Byte3.
15–11 -	This field is reserved. Reserved
HW_DG_LOW3	HW DQS gating calibration result of the lower boundary of Byte3. This field holds the HW DQS gating calibration result of the lower boundary of Byte3.

45.12.66 MMDC PHY Pre-defined Compare Register 1 (MMDCx_MPPDCMPR1)

This register holds the MMDC pre-defined compare value that will be used during automatic read, read DQS gating and write calibration process. The compare value can be the MPR value (as defined in the JEDEC) or can be programmed by the PDV1 and PDV2 fields. In case of DDR3 (BL=8) the MMDC will duplicate PDV1, PDV2 and drive that data on Beat4-7 of the same byte.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 88Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PDV2																PDV1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPPDCMPR1 field descriptions

Field	Description
31–16 PDV2	MMDC Pre defined compare value2. This field holds the 2 MSB of the data that will be driven to the DDR device during automatic read, read DQS gating and write calibrations in case MPR(DDR3)/ DQ calibration (LPDDR2/LPDDR3) mode are disabled (MPR_CMP is disabled). Upon read access during the calibration the MMDC will compare the read data with the data that is stored in this field. **NOTE:** Before issue the read access the MMDC will invert the value of this field and drive it to the associate entry in the read comparison FIFO. For further information, see Calibration with pre-defined value and Hardware (automatic) Calibration with MPR/DQ Calibration
PDV1	MMDC Pre defined compare value2. This field holds the 2 LSB of the data that will be driven to the DDR device during automatic read, read DQS gating and write calibrations in case MPR(DDR3)/ DQ calibration (LPDDR2/LPDDR3) mode are disabled (MPR_CMP is disabled). Upon read access during the calibration the MMDC will compare the read data with the data that is stored in this field.

Table continues on the next page...

MMDCx_MPPDCMPR1 field descriptions (continued)

Field	Description
	NOTE: Before issuing the read access, the MMDC will invert the value of this field and drive it to the associated entry in the read comparison FIFO.

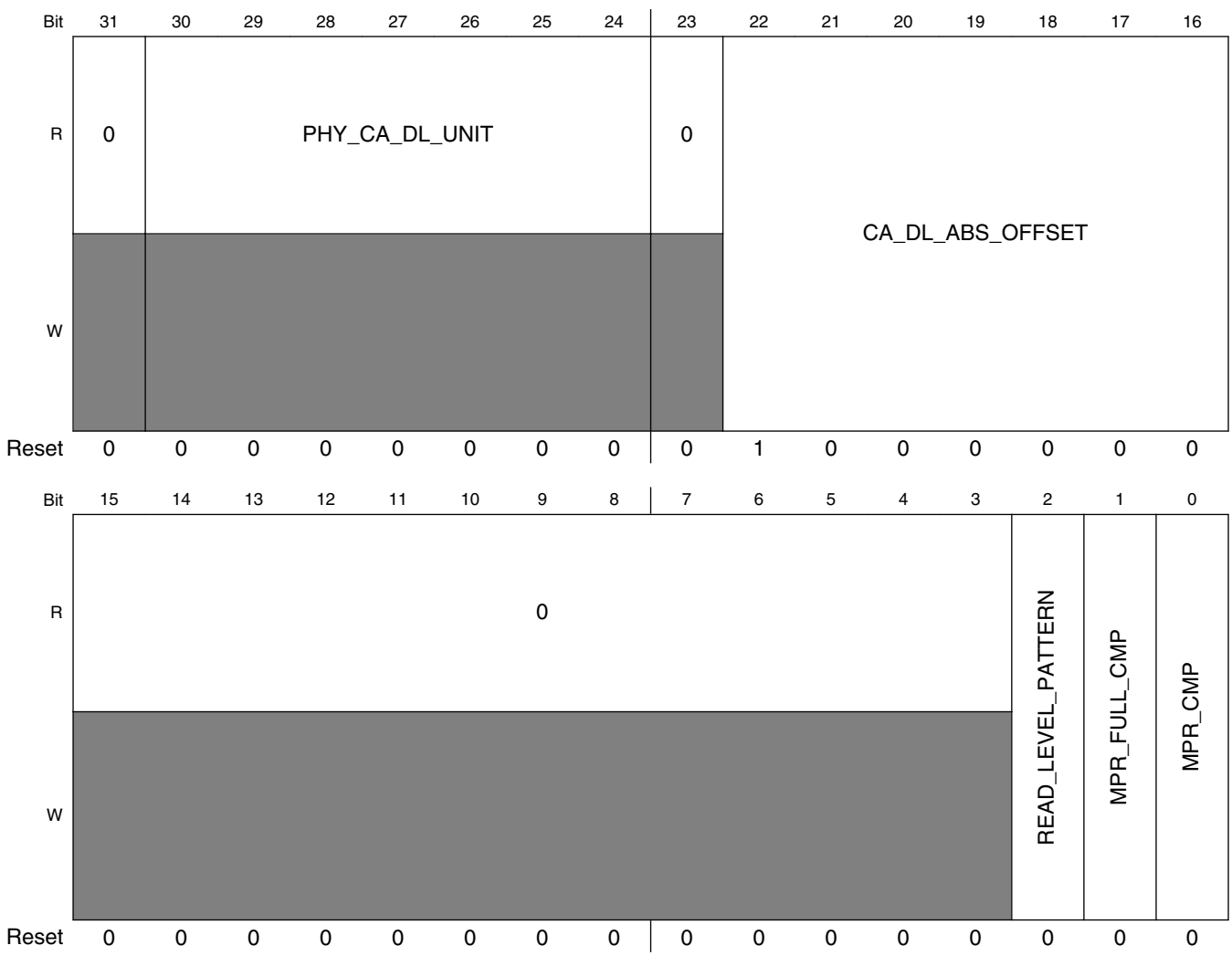
45.12.67 MMDC PHY Pre-defined Compare and CA delay-line Configuration Register (MMDCx_MPPDCMPR2)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 890h offset



MMDCx_MPPDCMPR2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–24 PHY_CA_DL_UNIT	This field reflects the number of delay units that are actually used by CA(Command/Address of LPDDR2) delay-line
23 Reserved	This read-only field is reserved and always has the value 0.
22–16 CA_DL_ABS_OFFSET	Absolute CA (Command/Address of LPDDRR2) offset. This field indicates the absolute delay between CA (Command/Address) bus and the DDR clock (CK) with fractions of a clock period and up to half cycle. The fraction is process and frequency independent. The delay of the delay-line would be (CA_DL_ABS_OFFSET / 256) * MMDC_CH0 AXI clock (fast clock). So for the default value of 64 we get a quarter cycle delay.
15–3 Reserved	This read-only field is reserved and always has the value 0.
2 READ_LEVEL_PATTERN	MPR(DDR3)/ DQ calibration (LPDDR2/LPDDR3) read compare pattern. In case MPR(DDR3)/ DQ calibration (LPDDR2/LPDDR3) modes are used during the calibration process (MPR_CMP is asserted) then this field indicates the read pattern for the comparison. 0 Compare with read pattern 1010 1 Compare with read pattern 0011 (Used only in LPDDR2/LPDDR3 mode)
1 MPR_FULL_CMP	MPR(DDR3)/ DQ calibration (LPDDR2/LPDDR3) full compare enable. In case MPR(DDR3)/ DQ calibration (LPDDR2/LPDDR3) modes are used during the calibration process (MPR_CMP is asserted) then this field indicates whether the MMDC will compare all the bits of the data that is read from the DDR device to the MPR pre-defined pattern. When this bit is de-asserted only LSB of each byte is compared.
0 MPR_CMP	MPR(DDR3)/ DQ calibration (LPDDR2/LPDDR3) compare enable. This bit indicates whether the MMDC will compare the read data during automatic read and read DQS calibration processes to the pre-defined patterns that are driven by the DDR device (READ_LEVEL_PATTERN as defined by JEDEC) or general pre-defined value that are stored in PDV1 and PDV2. When this bit is disabled data is compared to the data of the pre defined compare value field For further information see Read DQS Gating Calibration and Read Calibration .

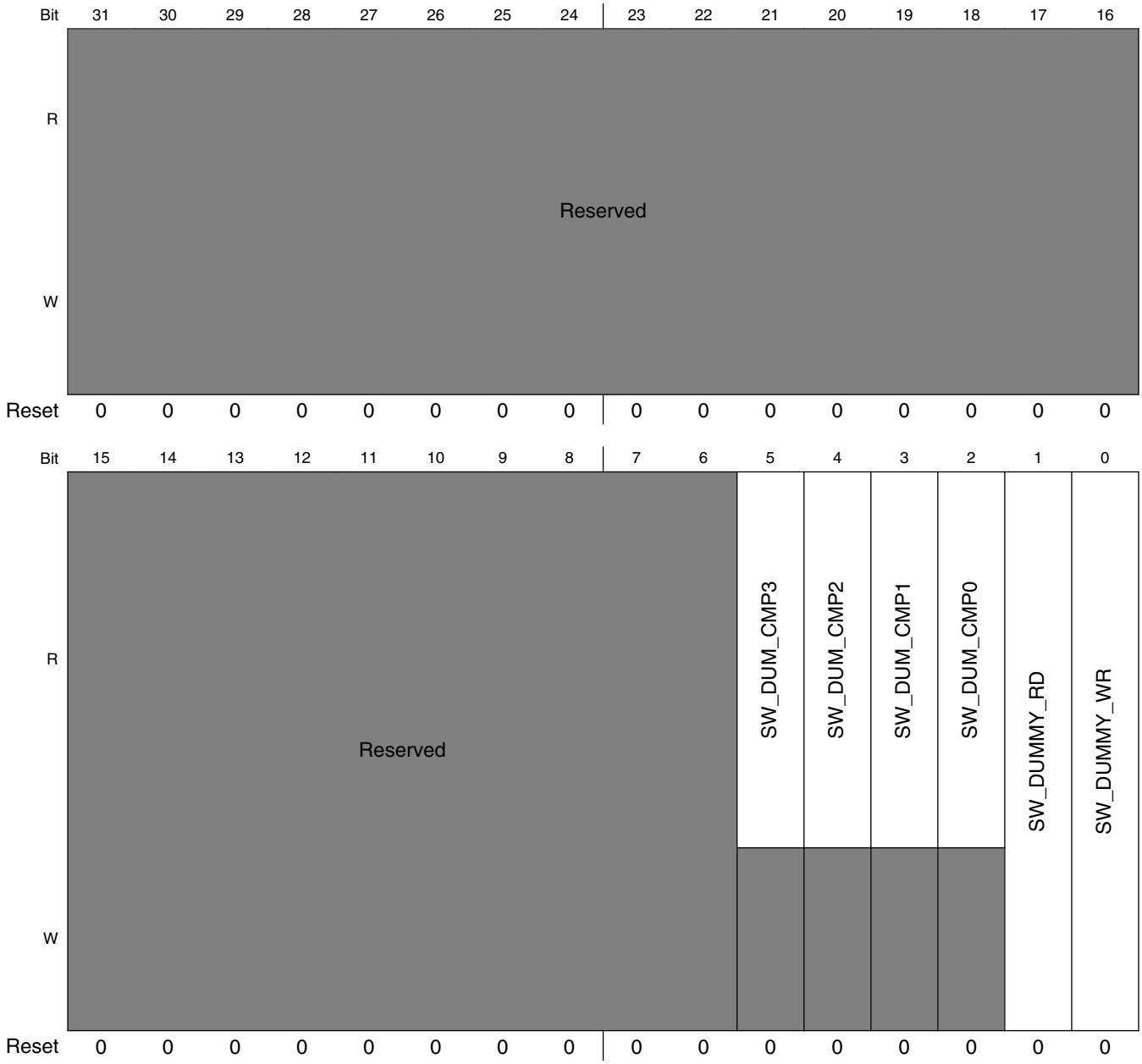
45.12.68 MMDC PHY SW Dummy Access Register (MMDCx_MPSWDAR0)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 894h offset



MMDCx_MPSWDAR0 field descriptions

Field	Description
31–6 -	This field is reserved.
5 SW_DUM_CMP3	SW dummy read byte3 compare results. This bit indicates the result of the read data comparison of Byte3 at the completion of SW_DUMMY_RD. This bit is valid only when SW_DUMMY_RD is de-asserted. 0 Dummy read fail 1 Dummy read pass
4 SW_DUM_CMP2	SW dummy read byte2 compare results. This bit indicates the result of the read data comparison of Byte2 at the completion of SW_DUMMY_RD. This bit is valid only when SW_DUMMY_RD is de-asserted.

Table continues on the next page...

MMDCx_MPSWDAR0 field descriptions (continued)

Field	Description
	0 Dummy read fail 1 Dummy read pass
3 SW_DUM_CMP1	SW dummy read byte1 compare results. This bit indicates the result of the read data comparison of Byte1 at the completion of SW_DUMMY_RD. This bit is valid only when SW_DUMMY_RD is de-asserted. 0 Dummy read fail 1 Dummy read pass
2 SW_DUM_CMP0	SW dummy read byte0 compare results. This bit indicates the result of the read data comparison of Byte0 at the completion of SW_DUMMY_RD. This bit is valid only when SW_DUMMY_RD is de-asserted. 0 Dummy read fail 1 Dummy read pass
1 SW_DUMMY_ RD	SW dummy read. When this bit is asserted the MMDC will generate internally read access without intervention of the system toward bank 0, row 0, column 0. If MPR_CMP = 1then the read data will be compared to MPPDCMPR2[READ_LEVEL_PATTERN] . If MPR_CMP =0 then the read data will be compared to MPPDCMPR1[PDV1], MPPDCMPR1[PDV2]. Upon completion of the access this bit is de-asserted automatically and the read data and comparison results are valid at MPSWDAR0[SW_DUM_CMP#] and MPSWDRDR0-MPSWDRDR7 respectively.
0 SW_DUMMY_ WR	SW dummy write. When this bit is asserted the MMDC will generate internally write access without intervention of the system toward bank 0, row 0, column 0, while the data is driven from MPPDCMPR1[PDV1] and MPPDCMPR1[PDV2]. The bit is de-asserted automatically upon completion of the access.

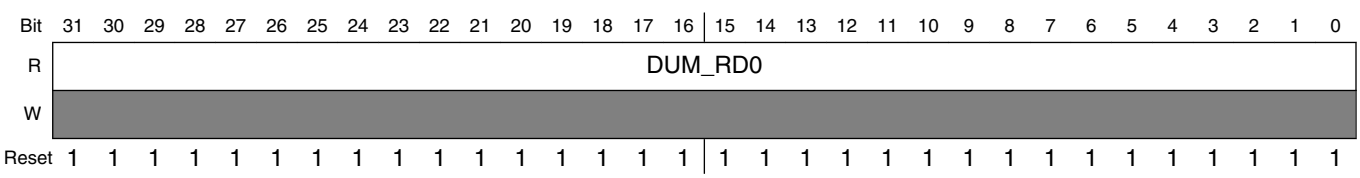
45.12.69 MMDC PHY SW Dummy Read Data Register 0 (MMDCx_MPSWDRDR0)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 898h offset



MMDCx_MPSWDRDR0 field descriptions

Field	Description
DUM_RD0	Dummy read data0. This field holds the first data that is read from the DDR during SW dummy read access (i.e. when SW_DUMMY_RD = 1). This field is valid only when SW_DUMMY_RD is de-asserted

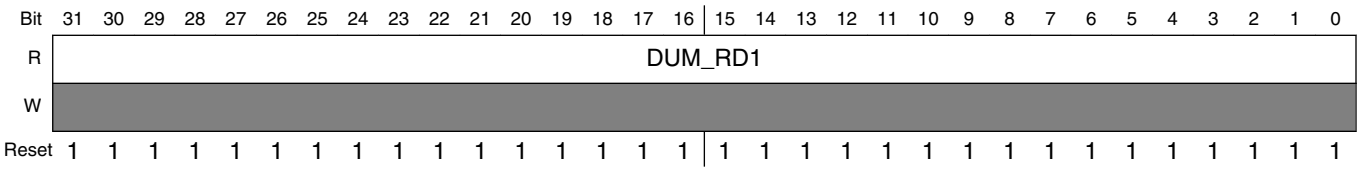
45.12.70 MMDC PHY SW Dummy Read Data Register 1 (MMDCx_MPSWDRDR1)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 89Ch offset



MMDCx_MPSWDRDR1 field descriptions

Field	Description
DUM_RD1	Dummy read data1. This field holds the second data that is read from the DDR during SW dummy read access (i.e. when SW_DUMMY_RD = 1). This field is valid only when SW_DUMMY_RD is de-asserted

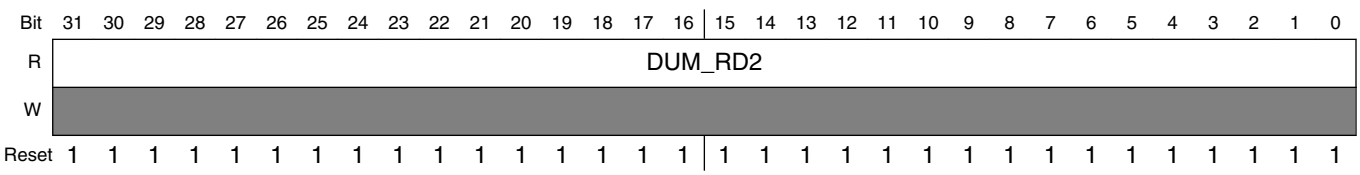
45.12.71 MMDC PHY SW Dummy Read Data Register 2 (MMDCx_MPSWDRDR2)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 8A0h offset



MMDCx_MPSWDRDR2 field descriptions

Field	Description
DUM_RD2	Dummy read data2. This field holds the third data that is read from the DDR during SW dummy read access (i.e. when SW_DUMMY_RD = 1). This field is valid only when SW_DUMMY_RD is de-asserted.

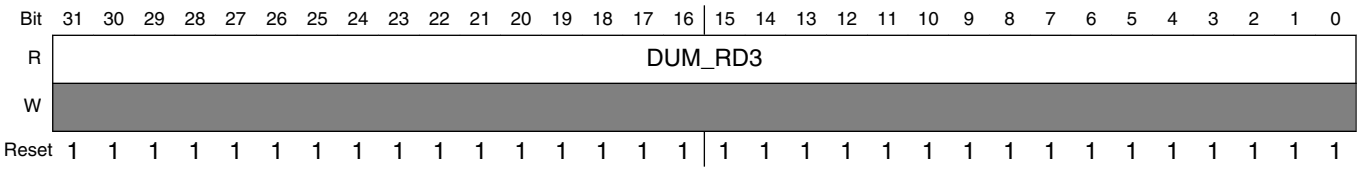
45.12.72 MMDC PHY SW Dummy Read Data Register 3 (MMDCx_MPSWDRDR3)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 8A4h offset



MMDCx_MPSWDRDR3 field descriptions

Field	Description
DUM_RD3	Dummy read data3. This field holds the forth data that is read from the DDR during SW dummy read access (i.e. when SW_DUMMY_RD = 1). This field is valid only when SW_DUMMY_RD is de-asserted.

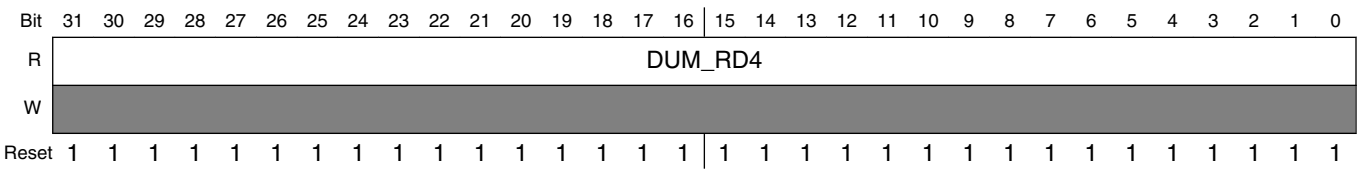
45.12.73 MMDC PHY SW Dummy Read Data Register 4 (MMDCx_MPSWDRDR4)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 8A8h offset



MMDCx_MPSWDRDR4 field descriptions

Field	Description
DUM_RD4	Dummy read data4. This field holds the fifth data (only in case of burst length 8 (BL =1)) that is read from the DDR during SW dummy read access (i.e. when SW_DUMMY_RD = 1). This field is valid only when SW_DUMMY_RD is de-asserted.

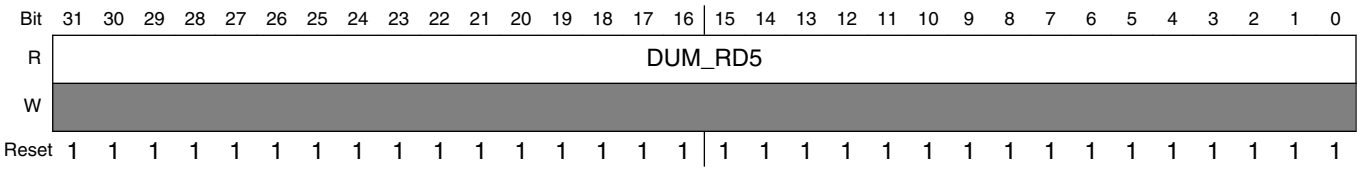
45.12.74 MMDC PHY SW Dummy Read Data Register 5 (MMDCx_MPSWDRDR5)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 8ACh offset



MMDCx_MPSWDRDR5 field descriptions

Field	Description
DUM_RD5	Dummy read data5. This field holds the sixth data (only in case of burst length 8 (BL =1)) that is read from the DDR during SW dummy read access (i.e. when SW_DUMMY_RD = 1). This field is valid only when SW_DUMMY_RD is de-asserted.

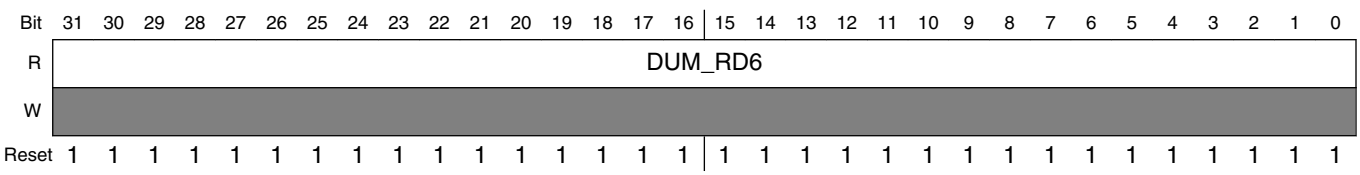
45.12.75 MMDC PHY SW Dummy Read Data Register 6 (MMDCx_MPSWDRDR6)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 8B0h offset



MMDCx_MPSWDRDR6 field descriptions

Field	Description
DUM_RD6	Dummy read data6. This field holds the seventh data (only in case of burst length 8 (BL =1)) that is read from the DDR during SW dummy read access (i.e. when SW_DUMMY_RD = 1). This field is valid only when SW_DUMMY_RD is de-asserted.

45.12.76 MMDC PHY SW Dummy Read Data Register 7 (MMDCx_MPSWDRDR7)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 8B4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DUM_RD7																															
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

MMDCx_MPSWDRDR7 field descriptions

Field	Description
DUM_RD7	Dummy read data7. This field holds the eight data (only in case of burst length 8 (BL =1)) that is read from the DDR during SW dummy read access (i.e. when SW_DUMMY_RD = 1). This field is valid only when SW_DUMMY_RD is de-asserted.

45.12.77 MMDC PHY Measure Unit Register (MMDCx_MPMUR0)

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 8B8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						MU_UNIT_DEL_NUM									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				FRC_MSR	MU_BYP_EN	MU_BYP_VAL									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPMUR0 field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.
25–16 MU_UNIT_DEL_NUM	Number of delay units measured per cycle. This field is used in debug mode and holds the number of delay units that were measured by the measure unit per DDR clock cycle. The delay-lines that are used in every calibration process use that number for generating the desired delay.
15–12 Reserved	This read-only field is reserved and always has the value 0.
11 FRC_MSR	Force measurement on delay-lines. When this bit is asserted then a measurement process will be performed, where at the completion of the process the delay-lines will issue the desired delay. Upon completion of the measurement process the measure unit and the delay-lines will return to functional mode. This bit is self cleared. **NOTE:** This bit should be used only during manual (SW) calibration and not while the DDR is functional (being accessed). After initial calibration is done the hardware performs periodic measurements to track any operating conditions changes. Hence, force measurements (FRC_MSR) should not be used. See Calibration Process for more information. **NOTE:** User should make sure that there is no active accesses to/from DDR before asserting this bit. 0 No measurement is performed 1 Perform measurement process
10 MU_BYP_EN	Measure unit bypass enable. This field is used in debug mode and when it is asserted then the delay-lines will use the number of delay units that are indicated at MU_BYP_VAL, otherwise the delay-lines will use the number of delay units that was measured by the measurement unit and are indicated at MU_UNIT_DEL_NUM 0 The delay-lines use delay units as indicated at MU_UNIT_DEL_NUM. 1 The delay-lines use delay units as indicated at MU_BYPASS_VAL.
MU_BYP_VAL	Number of delay units for measurement bypass. This field is used in debug mode and holds the number of delay units that will be used by the delay-lines when MU_BYP_EN is asserted.

45.12.78 MMDC Write CA delay-line controller (MMDCx_MPWRCADL)

This register is used to add fine-tuning adjustment to the CA (command/Address of LPDDR2 bus) relative to the DDR clock.

Supported Mode Of Operations:

For Channel 0: LP2_2ch_x16, LP2_2ch_x32

For Channel 1: LP2_2ch_x16, LP2_2ch_x32

Address: Base address + 8BCh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												WR_CA9_DEL		WR_CA8_DEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WR_CA7_DEL		WR_CA6_DEL		WR_CA5_DEL		WR_CA4_DEL		WR_CA3_DEL		WR_CA2_DEL		WR_CA1_DEL		WR_CA0_DEL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMDCx_MPWRCADL field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
19–18 WR_CA9_DEL	CA (Command/Address LPDDR2 bus) bit 9 delay fine tuning. This field holds the number of delay units that are added to CA (Command/Address bus) bit 9 relative to the clock. 00 No change in CA9 delay 01 Add CA9 delay of 1 delay unit 10 Add CA9 delay of 2 delay units. 11 Add CA9 delay of 3 delay units.
17–16 WR_CA8_DEL	CA (Command/Address LPDDR2 bus) bit 8 delay fine tuning. This field holds the number of delay units that are added to CA (Command/Address bus) bit 8 relative to the clock. 00 No change in CA8 delay 01 Add CA8 delay of 1 delay unit 10 Add CA8 delay of 2 delay units. 11 Add CA8 delay of 3 delay units.
15–14 WR_CA7_DEL	CA (Command/Address LPDDR2 bus) bit 7 delay fine tuning. This field holds the number of delay units that are added to CA (Command/Address bus) bit 7 relative to the clock. 00 No change in CA7 delay 01 Add CA7 delay of 1 delay unit 10 Add CA7 delay of 2 delay units. 11 Add CA7 delay of 3 delay units.
13–12 WR_CA6_DEL	CA (Command/Address LPDDR2 bus) bit 6 delay fine tuning. This field holds the number of delay units that are added to CA (Command/Address bus) bit 6 relative to the clock. 00 No change in CA6 delay 01 Add CA6 delay of 1 delay unit 10 Add CA6 delay of 2 delay units. 11 Add CA6 delay of 3 delay units.
11–10 WR_CA5_DEL	CA (Command/Address LPDDR2 bus) bit 5 delay fine tuning. This field holds the number of delay units that are added to CA (Command/Address bus) bit 5 relative to the clock. 00 No change in CA5 delay 01 Add CA5 delay of 1 delay unit 10 Add CA5 delay of 2 delay units. 11 Add CA5 delay of 3 delay units.
9–8 WR_CA4_DEL	CA (Command/Address LPDDR2 bus) bit 4 delay fine tuning. This field holds the number of delay units that are added to CA (Command/Address bus) bit 4 relative to the clock. 00 No change in CA4 delay 01 Add CA4 delay of 1 delay unit 10 Add CA4 delay of 2 delay units. 11 Add CA4 delay of 3 delay units.

Table continues on the next page...

MMDCx_MPWRCADL field descriptions (continued)

Field	Description
7–6 WR_CA3_DEL	CA (Command/Address LPDDR2 bus) bit 3 delay fine tuning. This field holds the number of delay units that are added to CA (Command/Address bus) bit 3 relative to the clock. 00 No change in CA3 delay 01 Add CA3 delay of 1 delay unit 10 Add CA3 delay of 2 delay units. 11 Add CA3 delay of 3 delay units.
5–4 WR_CA2_DEL	CA (Command/Address LPDDR2 bus) bit 2 delay fine tuning. This field holds the number of delay units that are added to CA (Command/Address bus) bit 2 relative to the clock. 00 No change in CA2 delay 01 Add CA2 delay of 1 delay unit 10 Add CA2 delay of 2 delay units. 11 Add CA2 delay of 3 delay units.
3–2 WR_CA1_DEL	CA (Command/Address LPDDR2 bus) bit 1 delay fine tuning. This field holds the number of delay units that are added to CA (Command/Address bus) bit 1 relative to the clock. 00 No change in CA1 delay 01 Add CA1 delay of 1 delay unit 10 Add CA1 delay of 2 delay units. 11 Add CA1 delay of 3 delay units.
WR_CA0_DEL	CA(Command/Address LPDDR2 bus) bit 0 delay fine tuning. This field holds the number of delay units that are added to CA (Command/Address bus) bit 0 relative to the clock. 00 No change in CA0 delay 01 Add CA0 delay of 1 delay unit 10 Add CA0 delay of 2 delay units. 11 Add CA0 delay of 3 delay units.

45.12.79 MMDC Duty Cycle Control Register (MMDCx_MPDCCR)

This register is used to control the duty cycle of the DQS and the primary clock (CK0). Programming of this register is permitted by entering the DDR device into self-refresh mode through LPMD/DVFS mechanism.

MMDC1_MPDCCR only affects SDCLK0.

MMDC2_MPDCCR only affects SDCLK1.

Supported Mode Of Operations:

For Channel 0: All

For Channel 1: DDR3_x64, LP2_2ch_x16, LP2_2ch_x32

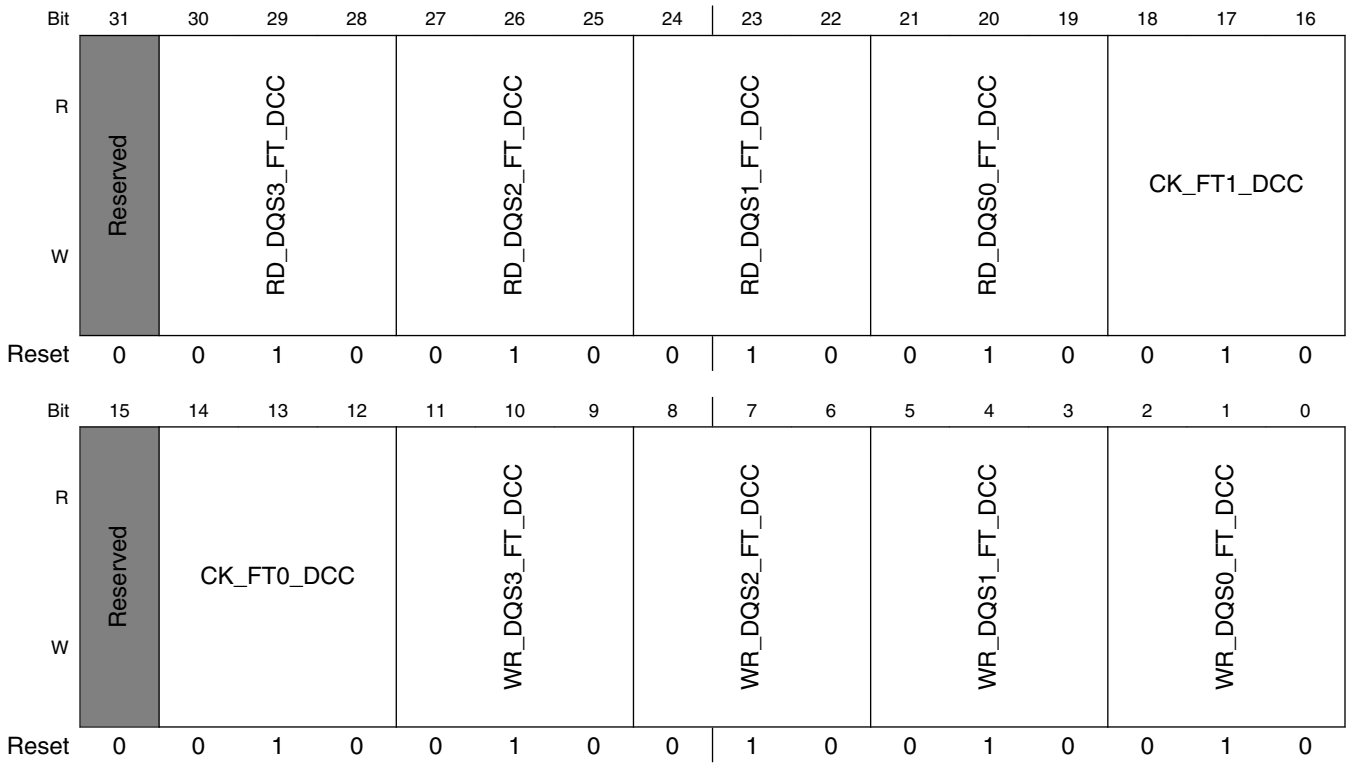
NOTE

If the duty cycle is modified after DDR initialization, the DDR will have to be placed in self-refresh mode.

NOTE

The duty cycle may be changed during initial DDR initialization without having to be placed in self-refresh mode.

Address: Base address + 8C0h offset



MMDCx_MPDCCR field descriptions

Field	Description
31 -	This field is reserved. reserved
30–28 RD_DQS3_FT_DCC	Read DQS duty cycle fine tuning control of Byte3. This field controls the duty cycle of read DQS of Byte3 **NOTE:** All the other options are not allowed 001 51.5% low 48.5% high 010 50% duty cycle (default) 100 48.5% low 51.5% high
27–25 RD_DQS2_FT_DCC	Read DQS duty cycle fine tuning control of Byte2. This field controls the duty cycle of read DQS of Byte2 **NOTE:** All the other options are not allowed 001 51.5% low 48.5% high 010 50% duty cycle (default) 100 48.5% low 51.5% high

Table continues on the next page...

MMDCx_MPDCCR field descriptions (continued)

Field	Description
24–22 RD_DQS1_FT_DCC	Read DQS duty cycle fine tuning control of Byte1. This field controls the duty cycle of read DQS of Byte1 **NOTE:** All the other options are not allowed 001 51.5% low 48.5% high 010 50% duty cycle (default) 100 48.5% low 51.5% high
21–19 RD_DQS0_FT_DCC	Read DQS duty cycle fine tuning control of Byte0. This field controls the duty cycle of read DQS of Byte0 **NOTE:** All the other options are not allowed 001 51.5% low 48.5% high 010 50% duty cycle (default) 100 48.5% low 51.5% high
18–16 CK_FT1_DCC	Secondary duty cycle fine tuning control of DDR clock. This field controls the duty cycle of the DDR clock and is cascaded to CK_FT0_DCC **NOTE:** All the other options are not allowed 001 48.5% low 51.5% high 010 50% duty cycle (default) 100 51.5% low 48.5% high
15 -	This field is reserved. Reserved
14–12 CK_FT0_DCC	Primary duty cycle fine tuning control of DDR clock. This field controls the duty cycle of the DDR clock **NOTE:** All the other options are not allowed 001 48.5% low 51.5% high 010 50% duty cycle (default) 100 51.5% low 48.5% high
11–9 WR_DQS3_FT_DCC	Write DQS duty cycle fine tuning control of Byte0. This field controls the duty cycle of write DQS of Byte0 **NOTE:** All the other options are not allowed 001 51.5% low 48.5% high 010 50% duty cycle (default) 100 48.5% low 51.5% high
8–6 WR_DQS2_FT_DCC	Write DQS duty cycle fine tuning control of Byte1. This field controls the duty cycle of write DQS of Byte1 **NOTE:** All the other options are not allowed 001 51.5% low 48.5% high 010 50% duty cycle (default) 100 48.5% low 51.5% high
5–3 WR_DQS1_FT_DCC	Write DQS duty cycle fine tuning control of Byte1. This field controls the duty cycle of write DQS of Byte1 **NOTE:** All the other options are not allowed 001 51.5% low 48.5% high 010 50% duty cycle (default) 100 48.5% low 51.5% high

Table continues on the next page...

MMDCx_MPDCCR field descriptions (continued)

Field	Description
WR_DQS0_FT_DCC	Write DQS duty cycle fine tuning control of Byte0. This field controls the duty cycle of write DQS of Byte0 **NOTE:** All the other options are not allowed 001 51.5% low 48.5% high 010 50% duty cycle (default) 100 48.5% low 51.5% high

Chapter 46
Network Interconnect Bus System (NIC-301)

46.1 Overview

This section provides an overview of the NIC-301 (Network Inter-Connect) AXI arbiter IP.

The NIC-301 (by Arm Ltd.) is a configurable AXI arbiter between several masters and slaves. The NIC-301 IP is designed so that many configuration options are selected at the hardware design stage, determined by SoC characteristics and needs, while several other configuration options are software-controlled.

This chapter covers in brief the NIC-301 functionality, while providing configuration details on the NIC-301 instances used in the chip. For complete details on the NIC-301 design, see the Arm specification, *AMBA® Network Interconnect (NIC-301) Technical Reference Manual, version r2p3*.

NOTE

The NIC-301 default settings are configured by NXP's board support package (BSP), and in most cases should not be modified by the customer. The default settings have gone through exhaustive testing during the validation of the part, and have proven to work well for the part's intended target applications. Changes to the default settings may result in a degradation in system performance.

46.1.1 Block diagram

The NIC-301 AXI arbiter (or "CoreLink Network Interconnect") by Arm, provides configurable AXI-based interconnect logic, for connecting a number of masters (initiators) to several slaves (targets), via configurable bus switches and bridging components.

The bus system is composed of five instances: PER1, PER2, FAST1, FAST2, and FAST3.

Each instance can include one or more bus switches, with additional logic. This block provides details of the various instances and the selected configuration parameters. The top level diagram of the bus system is shown in the following figure.

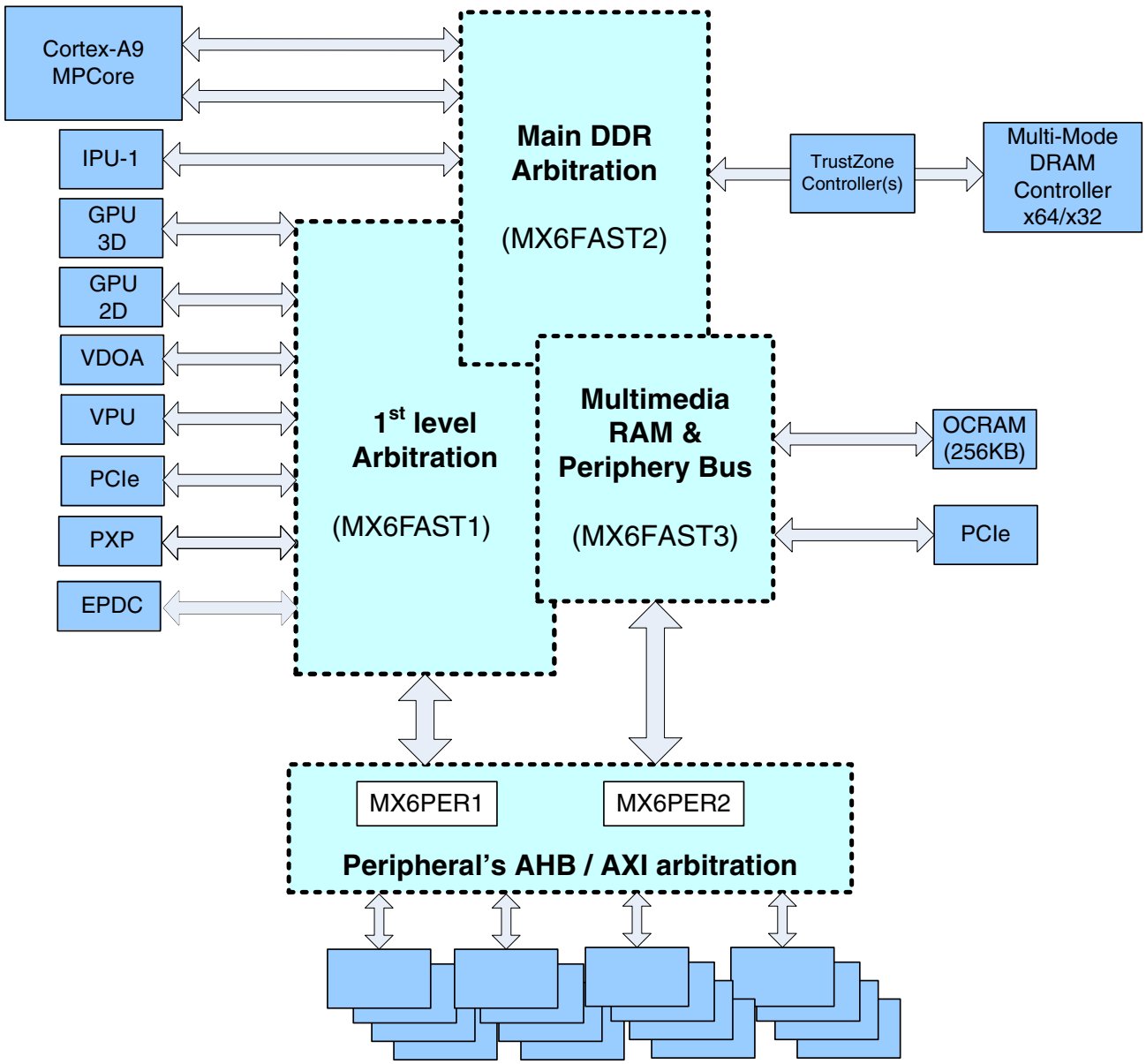


Figure 46-1. NIC-301 Bus System

46.1.2 NIC-301 Main Features

Key features of the NIC-301 module include the following:

- Address space memory mapping, including 'remap' functions.
- Programmer's view, for software-configured parameters, via "GPV" ports.
- Support for cross-clock domain synchronization.

46.1.3 Modes and Operations

The NIC-301 supports a normal functional mode only.

46.2 External Signals

There are no external I/O interfaces for NIC-301.

46.3 Memory Map and Register Definition

This section includes the block memory map and detailed register descriptions.

Access to NIC-301 registers is provided through the global programmer's view (GPV) ports. Each GPV port provides access to the configuration registers of certain IP as listed in Table 46-5 . The GPV base addresses are listed in the table below:

Table 46-1. GPV ports memory allocations

GPV	Associated NIC-301	System Bus	Chip Address	
			Start	End
GPV_0	FAST2	Main DDR Arbitration	00B0_0000	00BF_FFFF
GPV_1	FAST1	First Level Arbitration	00C0_0000	00CF_FFFF
GPV_2	PER1	Peripheral's AHB/AXI Arbitration	0020_0000	002F_FFFF
GPV_3	PER2	Reserved (Internal Use Only)	0030_0000	003F_FFFF
GPV_4	FAST3	Multimedia RAM and Peripheral Bus	0080_0000	008F_FFFF

46.3.1 Memory Map

The NIC-301 memory map, is dependent on the selected configuration option at time of creation.

A "template" map is provided in Table 46-2 below. For specific features, see the configuration tables in NIC-specific parameters to check whether specific options are selected.

46.3.2 Configuration programmers model

The GPV's contain configuration registers, partitioned into a number of individual 4KB blocks.

The general structure of the registers is provided by the following tables:

- Address map of the programmers model, Table 46-2
- AMIB Registers, Table 46-3
- ASIB Registers, Table 46-4

Table 46-2. Address map of the programmers model

Address Offset from Base Address	Registers	Notes
0x000F_F000	Internal interface p registers	Maximum p = 61
	...	Note [1]
0x000C_4000	Internal interface 2 registers	
0x000C_3000	Internal interface 1 registers	
0x000C_2000	Internal interface 0 registers	
0x000C_1000	Slave interface m registers	Maximum m = 127
	...	Note [2]
0x0004_4000	Slave interface 2 registers	
0x0004_3000	Slave interface 1 registers	
0x0004_2000	Slave interface 0 registers	
0x0004_1000	Master interface n registers	Maximum n = 63
	...	Note [3]
0x0000_4000	Master interface 2 registers	
0x0000_3000	Master interface 1 registers	
0x0000_2000	Master interface 0 registers	
0x0000_1000	ID registers	
0x0000_0000	Address control registers	Configurable base address [4]

1. Index refers to BI registers index
2. Index refer to ASIB registers index
3. Index refer to AMIB registers index
4. Reserved for internal use

46.3.2.1 Address control and ID registers

Registers at offsets 0x0–0xFFC are reserved for internal use.

46.3.2.2 AMBA master interface block (AMIB) configuration registers

The table below lists only the registers that affect the user. All other addresses are treated as "reserved". Non-implemented or reserved addresses inside NIC domains are read as ZEROS and writes operations are ignored.

Table 46-3. AMIB Registers

Offset	Register	Access	Width	Reset Value
0x024	fn_mod2 Bypass merge. This register is only present if upsizing or downsizing. See upsizing/downsizing data width functions in AMBA Network Interconnect TRM.	RW	1	0
0x040	wr_tidemark	RW	4	Note [1]

1. Reset value varies, default value chosen at RTL creation time is designed to suit normal operation.

46.3.2.3 ASIB (AMBA slave interface block) configuration registers

The table below lists only the registers that affect the user. All other addresses are treated as "reserved". Non-implemented or reserved addresses inside NIC domains are read as ZEROS and writes operations are ignored.

Table 46-4. ASIB Registers

Offset	Register	Access	Width	Reset Value
0x040	wr_tidemark Valid only for AXI slaves with WFIFO >=4.	RW	1	Note [1]
0x100	read_qos	RW	4	Note [2]
0x104	write_qos	RW	4	Note [2]

1. Reset value varies, default value chosen at RTL creation time is designed to suit typical operation cases.
2. QoS default is set at RTL creation time, and is listed in specific NIC-301 configuration tables in NIC-specific parameters, as parameters "QoS qv_value" in ASIB / AMIB parameter tables.

46.3.3 Register Descriptions

The NIC-301 registers are dependent upon the selected configuration, the type of ports, hardware-selected features and whether they have a GPV view. The addressing is associated to a specific port, by looking at the port's index number, under "apb_slave" column.

The memory map template is provided in the Configuration programmers model above.

46.3.3.1 QoS registers' address look-up example

The flow below describes the steps needed in determining the memory addresses for GPU3D QoS read/write registers:

- Look up the GPU3D port index ("apb_slave"[1] column in Table 46-5): index 66. Note that the GPU3D is listed under GPV_1 interface.
- Look up the GPV_1 base address: 0x00C0_0000, in Table 46-1 .
- The master/slave ports configuration base address, is always obtained by: "GPV base" + ("APB index" x 0x1000), regardless of whether it is a slave or a master port. This is somewhat contrary to the way indexes are presented in the "Address map of the NIC-301 programmers model" in the ARM documentation as shown in Table 46-2, where, instead, the indexes are provided in absolute values (of 0x1000 jumps), starting from the GPV base address.

  Hence, for the GPU3Dport (index '66'), the "slave interface" programming base address is 0x00C4_2000 (= GPV_1 base + 66 x 0x1000).

- To access any of the slave/master interface registers (in this example - the QoS read/write ones), their offset(s) must be added, as shown in Table 46-4 :

  "0x100" for Read_QoS and "0x104" for Write_QoS, in our example.

Final addresses obtained for QoS registers for the GPU3D primary port (index 66), are as follows:

- Read QoS: "0x00C4_2100"
- Write QoS: "0x00C4_2104"

46.3.4 NIC-specific parameters

This section details the configuration parameters of the NIC-301.

1. The "slave", "master" type is assumed from NIC-301 point of view.

General notes:

1. All accesses to GPV_4 port, must be of "Supervisor" type.
2. The associated master/slave port interface ID (for each "GPV_N") is specified by the "apb_slave" / "apb_master" [2] index. The slave/master interface, configuration register's start address is obtained by the following:

   Start address (for "apb" index= "n") = GPV_N base + (n x 0x1000).

3. The security features of NIC-301 are not used, since master-slaves permissions are controlled by the CSU security policy/scheme.

Table 46-5. QoS and tidemark parameters

Master	apb_slave	Tidemark	QOS qv_value	Comments
Configured via GPV_1 port				
GPU3D	66	0	2	
GPU2D (Primary)	67	0	2	
GPU2D (Secondary)	72		2	
VDOA	68	4	2	
PXP	78	16	2	
EPDC	76	16	2	
VPU Prim.	73	4	2	
PCIe	74	16	2	
Configured via GPV_0 port				
MPCore-0	66		2	
MPCore-1 (not included in i.MX 6Solo)	67		2	
IPU1	68	4	0	QoS configurable via GPR bits in IOMUXC.
Configured via GPV_4 port				
VPU (to internal RAM)	68		2	
Configured via GPV_2 port (switch 0)				
HDMI	66		2	
SDMA burst	67		3	
SDMA Peripheral	68		3	
CAAM	69		2	
USBOH3A	70		2	
ENET	71		2	
HSI	72		2	
uSDHC1	73		2	
Configured via GPV_2 port (switch 1)				

Table continues on the next page...

2. "APB" stands for "ARM Peripheral Bus".

Table 46-5. QoS and tidemark parameters (continued)

Master	apb_slave	Tidemark	QOS qv_value	Comments
DAP	74		2	
APBH	75		2	
BCH40	76		2	
MLB150	78		2	
uSDHC2	79		2	
uSDHC3	80		2	
uSDHC4	81		2	

Chapter 47
On-Chip OTP Controller (OCOTP_CTRL)

47.1 Overview

This section contains information describing the requirements for the on-chip eFuse OTP controller along with details about the block functionality and implementation.

In this document, the words "eFuse" and "OTP" are interchangeable. OCOTP refers to the hardware block itself.

47.1.1 Features

The OCOTP provides the following features:

- 32-bit word restricted program and read to 4 kbit of eFuse OTP (512 x 8).
- Loading and housing of fuse content into shadow registers.
- Memory-mapped (restricted) access to 4 kbit of shadow registers.
- Generation of HWV_FUSE (hardware visible fuse bus) and the HWV_REG bus which is made up of volatile PIO register based "fuses". The HWV_REG bits come from the SCS (Software Controllable Signals) register.
- Generation of STICKY_REG which is consist of sticky register bits.
- Provide program-protect and read-protect eFuse.
- Provide override and read protection of shadow register.

47.2 Clocks

The table found here describes the clock sources for OCOTP.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 47-1. OCOTP Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

47.3 Top-Level Symbol and Functional Overview

The figure found here shows the OCOTP system level diagram.

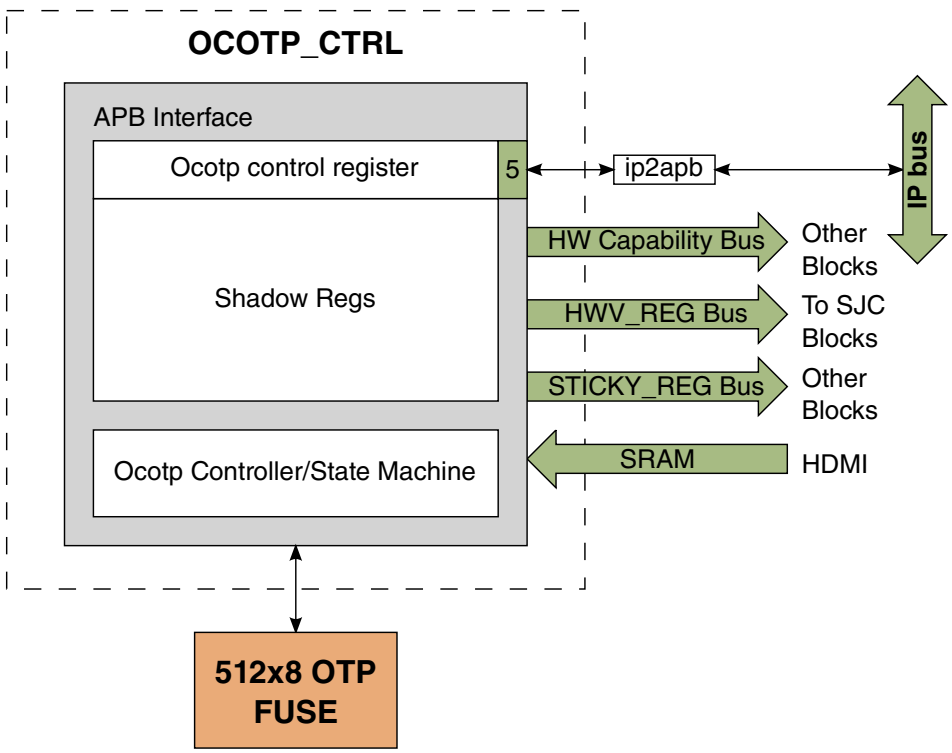


Figure 47-1. OCOTP System Level Diagram

47.3.1 Operation

The IP bus interface of the OCOTP provides two functions.

- Configure control registers for programming and reading fuse word.
- Override and read shadow registers.

For efuse, program can only be performed on bit and read is based on byte. OCOTP configuration for program and read are performed on 32-bit words for SW convenience. For writes, the 32-bit word reflects the "write-mask". Bit fields with 0 will not be programmed and bit fields with 1 will be programmed. OCOTP will program bit field with 1 in the fuse word one bit by one bit. For reads, OCOTP will read 4 times to get 4 bytes in the fuse word in order.

In this document, 4-kbit fuse are divided into 16 banks by function. Each bank has 8 fuse words. In physical, 4k bits fuse are in one 512x8 efusebox.

47.3.1.1 Shadow Register Reload

All fuse words in efusebox are shadowed. Therefore, fuse information is available through memory mapped shadow registers. If fuses are subsequently programmed, the shadow registers should be reloaded to keep them coherent with the fuse bank arrays.

The "reload shadows" feature allows the user to force a reload of the shadow registers (including HW_OCOTP_LOCK) without having to reset the device. To force a reload, complete the following steps:

1. Set the HW_OCOTP_TIMING[STROBE_READ] and HW_OCOTP_TIMING[RELAX] field value appropriately (as explained in a later section).
2. Check that HW_OCOTP_CTRL[BUSY] and HW_OCOTP_CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write , read or reload must be completed before a new access can be requested.
3. Set the HW_OCOTP_CTRL[RELOAD_SHADOWS] bit. OCOTP will read all the fuse one by one and put it into corresponding shadow register.
4. Wait for HW_OCOTP_CTRL[BUSY] and HW_OCOTP_CTRL[RELOAD_SHADOWS] to be cleared by the controller.

The controller will automatically clear the HW_OCOTP_CTRL[RELOAD_SHADOWS] bit after the successful completion of the operation.

47.3.1.2 Fuse and Shadow Register Read

All shadow registers are always readable through the APB bus except some secret keys regions. When their corresponding fuse lock bits are set, the shadow registers also become read locked. After read locking, reading from these registers will return 0xBADABADA.

In addition HW_OCOTP_CTRL[ERROR] will be set. It must be cleared by software before any new write , read or reload access can be issued. Subsequent reads to unlocked shadow locations will still work successfully however.

To read fuse words directly from fusebox correctly complete the following steps:

1. Program HW_OCOTP_TIMING[STROBE_READ] and HW_OCOTP_TIMING[RELAX] fields with timing values to match the current frequency of the ipg_clk. OTP read will work at maximum bus frequencies as long as the HW_OCOTP_TIMING parameters are set correctly.
2. Check that HW_OCOTP_CTRL[BUSY] and HW_OCOTP_CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write, read or reload must be completed before a read access can be requested.
3. Write the requested address to HW_OCOTP_CTRL[ADDR].
4. Set HW_OCOTP_READ_CTRL[READ_FUSE] to 1. OCOTP will auto read 4 bytes in requested word address in fusebox one by one. Then put read value into HW_OCOTP_READ_FUSE_DATA register.
5. Once complete, the controller will clear BUSY. A read request to a protected or locked region will result in no OTP access and no setting of HW_OCOTP_CTRL[BUSY]. In addition HW_OCOTP_CTRL[ERROR] will be set. It must be cleared by software before any new access can be issued.
6. Read HW_OCOTP_READ_FUSE_DATA register to get fuse word value.HW_OCOTP_READ_FUSE_DATA will be 0xBADABADA when HW_OCOTP_CTRL[ERROR] is set.

47.3.1.3 Fuse and Shadow Register Writes

Shadow register bits can be overridden by software until the corresponding fuse lock bit for the region is set. When the lock shadow bit is set, the shadow registers for that lock region become write locked. The LOCK shadow register also has no shadow or fuse lock bits but it is always read only.

In order to avoid "rogue" code performing erroneous writes to OTP, a special unlocking sequence is required for writes to the fuse banks. To program fuse bank correctly complete the following steps:

1. Program HW_OCOTP_TIMING[STROBE_PROG] and HW_OCOTP_TIMING[RELAX] fields with timing values to match the current frequency of the ipg_clk. OTP writes will work at maximum bus frequencies as long as the HW_OCOTP_TIMING parameters are set correctly.

2. Check that HW_OCOTP_CTRL[BUSY] and HW_OCOTP_CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write or reload must be completed before a write access can be requested.
3. Write the requested address to HW_OCOTP_CTRL[ADDR] and program the unlock code into HW_OCOTP_CTRL[WR_UNLOCK]. This must be programmed for each write access. The lock code is documented in the register description. Both the unlock code and address can be written in the same operation.
4. Write the data to the HW_OCOTP_DATA register. This will automatically set HW_OCOTP_CTRL[BUSY] and clear HW_OCOTP_CTRL[WR_UNLOCK]. To protect programming same OTP bit twice, before program OCOTP will automatically read fuse value in OTP and use read value to mask program data. The controller will use masked program data to program a 32-bit word in the OTP per the address in HW_OCOTP_CTRL[ADDR]. Bit fields with 1's will result in that OTP bit being programmed. Bit fields with 0's will be ignored. At the same time that the write is accepted, the controller makes an internal copy of HW_OCOTP_CTRL[ADDR] which cannot be updated until the next write sequence is initiated. This copy guarantees that erroneous writes to HW_OCOTP_CTRL[ADDR] will not affect an active write operation. It should also be noted that during the programming HW_OCOTP_DATA will shift right (with zero fill). This shifting is required to program the OTP serially. During the write operation, HW_OCOTP_DATA cannot be modified.
5. Once complete, the controller will clear BUSY. A write request to a protected or locked region will result in no OTP access and no setting of HW_OCOTP_CTRL[BUSY]. In addition HW_OCOTP_CTRL[ERROR] will be set. It must be cleared by software before any new write access can be issued.

It should be noted that write latencies to OTP are numbers of 10 micro-seconds per word. Write latencies is based on amount of bit filed which is 1. For example : program half fuse bits in one word need 10 us x 16.

For further details of OTP read/write operations see [eFUSE].

HW_OCOTP_CTRL[ERROR] will be set under the following conditions:

- A write is performed to a shadow register during a shadow reload (essentially, while HW_OCOTP_CTRL[RELOAD_SHADOWS] is set. In addition, the contents of the shadow register shall not be updated.
- A write is performed to a shadow register which has been locked.
- A read is performed to from a shadow register which has been read locked.
- A program is performed to a fuse word which has been locked.
- A read is performed to from a fuse word which has been read locked.

47.3.1.4 Write Postamble

Due to internal electrical characteristics of the OTP during writes, all OTP operations following a write must be separated by 2 us after the clearing of HW_OCOTP_CTRL_BUSY following the write. This guarantees programming voltages on-chip to reach a steady state when exiting a write sequence. This includes reads, shadow reloads, or other writes.

A recommended software sequence to meet the postamble requirements is as follows:

- Issue the write and poll for BUSY (as per Fuse Shadow Memory Footprint).
- After BUSY is clear, wait an additional 2 us.
- Perform the next OTP operation.

47.3.2 Fuse Shadow Memory Footprint

The OTP memory footprint shows in the following figure. The registers are grouped by lock region. Their names correspond to the PIO register and fusemap names.

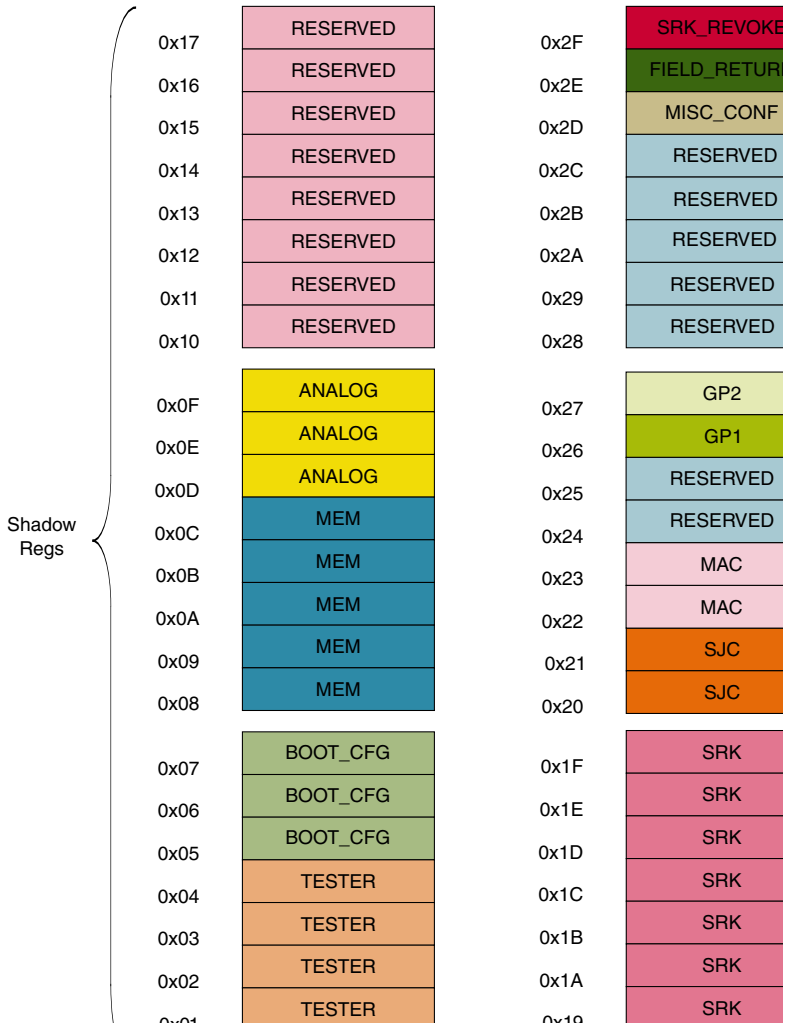


Figure 47-2. OTP Memory Footprint

47.3.3 OTP Read/Write Timing Parameters

There are three timing fields contained in the HW_OCOTP_TIMING register that specify counter limit values, which are used to time how long the state machine remains in the various states, as well as specify the STROBE signal timing.

TheyBoth two timing parameters are all specified in ipg_clk cycles. Since the ipg_clk frequency can be set to a range of values, these parameters must be adjusted with the clock to yield the appropriate delay.

The HW_OCOTP_TIMING[RELAX] field specifies how long to remain in states to meet setup and hold timing requirement in fuse spec. This parameter should be set by the following equation:

tRELAX = tHP_PG = (HW_OCOTP_TIMING[RELAX]+1)/ipg_frequency > 16.2ns

HW_OCOTP_TIMING[RELAX] field is used to create other setup and hold timing delays in addition to tHP_PG. For all timing to be met, this is the max delay that must be programmed.

Except for setup and hold timing delay, there are 2 timing parameters for STROBE signal pulse width in program and read.

The HW_OCOTP_TIMING[STROBE_PROG] field specifies the period of the STROBE signal for fuse writes and is given in units of ipg_clk cycles. This value should be specified so that the requirement for the time when the STROBE signal is asserted high is met: 9000ns < tPGM < 11000ns is met. Even though a range is given for tPGM, it is advised in **[eFUSE]** to program for a value of 10000ns. Therefore, this field should be set according to the following equation:

tPGM = ((HW_OCOTP_TIMING[STROBE_PROG]+1)-2*(HW_OCOTP_TIMING[RELAX]+1))/ipg_frequency = 10000ns.

The HW_OCOTP_TIMING[STROBE_READ] field specifies the period of the STROBE signal for fuse reads and is given in units of ipg_clk cycles. This field should be set according to the following equation:

tRD = ((HW_OCOTP_TIMING[STROBE_READ]+1)-2*(HW_OCOTP_TIMING[RELAX]+1))/ipg_frequency > 36ns.

The figure below illustrates the relationship between the STROBE signal in programming and reading mode, as well as the timing PIO register fields that affect it. The implementation uses one counter to generate the STROBE waveform within one period and a second counter counts the number of cycles to create for programming the designated word.

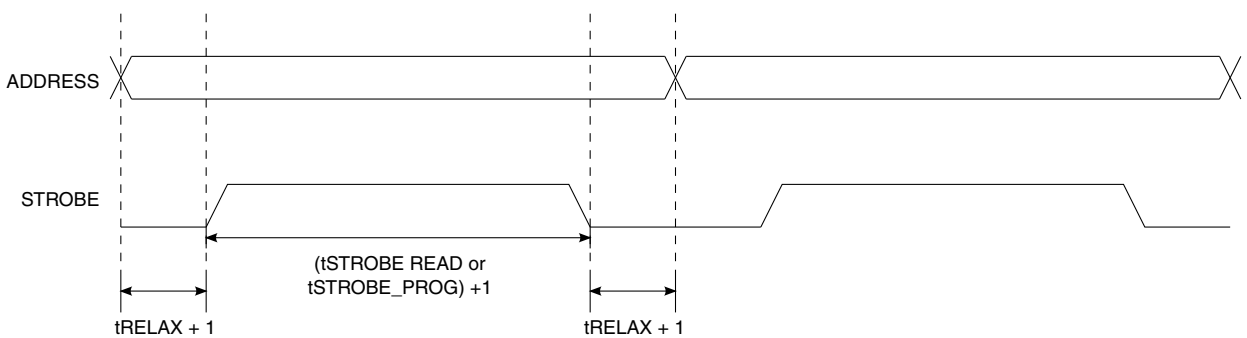


Figure 47-3. STROBE Signal Creation and Timing

47.3.4 Hardware Visible Fuses

The hwv_fuse bus emanates from the OCOTP block and goes to various other blocks inside the chip. This bus is made up of the shadow register bits for banks 0, 1, 2 and 4.

Only a subset of these fuse bits are currently used by the hardware. The fuse bits are initially copied from the eFuse banks after reset is deasserted. When all fuse bits are loaded into their shadow registers, the OCOTP asserts the fuse_latched output signal.

The hwv_reg bus also comes from the OCOTP. Its source is the HW_OCOTP_SCS register. This register has 1 defined bit, the HAB_JDE bit, that is connected to the SJC block. The SCS bits are intended to be used as volatile fuse bits under software control. Additional bits will be defined as needed in future implementations.

The system-wide reset sequence must be coordinated by the system reset controller, so that the hwv_fuse and hwv_reg buses are stable and reflect the values of the fuses before they are used by the rest of the system.

47.3.5 Behavior During Reset

The OCOTP is always active. The shadow registers automatically load the appropriate OTP contents after reset is deasserted. During this load-time HW_OCOTP_CTRL_BUSY is set. The load time is similar to that of a "reload shadow" operation.

47.4 Fuse Map

See the Fusemap chapter of this reference manual for more information.

47.5 OCOTP Memory Map/Register Definition

NOTE

When write/read unimplemented register address in ocotp_ctrl, ocotp_ctrl will not send error and read data will be 0.

OCOTP Hardware Register Format Summary

OCOTP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21B_C000	OTP Controller Control Register (OCOTP_CTRL)	32	R/W	0000_0000h	47.5.1/4094
21B_C004	OTP Controller Control Register (OCOTP_CTRL_SET)	32	R/W	0000_0000h	47.5.1/4094
21B_C008	OTP Controller Control Register (OCOTP_CTRL_CLR)	32	R/W	0000_0000h	47.5.1/4094
21B_C00C	OTP Controller Control Register (OCOTP_CTRL_TOG)	32	R/W	0000_0000h	47.5.1/4094
21B_C010	OTP Controller Timing Register (OCOTP_TIMING)	32	R/W	0146_1299h	47.5.2/4096
21B_C020	OTP Controller Write Data Register (OCOTP_DATA)	32	R/W	0000_0000h	47.5.3/4096
21B_C030	OTP Controller Write Data Register (OCOTP_READ_CTRL)	32	R/W	0000_0000h	47.5.4/4097
21B_C040	OTP Controller Read Data Register (OCOTP_READ_FUSE_DATA)	32	R/W	0000_0000h	47.5.5/4098
21B_C050	Sticky bit Register (OCOTP_SW_STICKY)	32	R/W	0000_0000h	47.5.6/4098
21B_C060	Software Controllable Signals Register (OCOTP_SCS)	32	R/W	0000_0000h	47.5.7/4099
21B_C064	Software Controllable Signals Register (OCOTP_SCS_SET)	32	R/W	0000_0000h	47.5.7/4099
21B_C068	Software Controllable Signals Register (OCOTP_SCS_CLR)	32	R/W	0000_0000h	47.5.7/4099
21B_C06C	Software Controllable Signals Register (OCOTP_SCS_TOG)	32	R/W	0000_0000h	47.5.7/4099
21B_C090	OTP Controller Version Register (OCOTP_VERSION)	32	R	0200_0000h	47.5.8/4100
21B_C400	Value of OTP Bank0 Word0 (Lock controls) (OCOTP_LOCK)	32	R	0000_0000h	47.5.9/4101
21B_C410	Value of OTP Bank0 Word1 (Configuration and Manufacturing Info.) (OCOTP_CFG0)	32	R/W	0000_0000h	47.5.10/ 4104
21B_C420	Value of OTP Bank0 Word2 (Configuration and Manufacturing Info.) (OCOTP_CFG1)	32	R/W	0000_0000h	47.5.11/ 4105
21B_C430	Value of OTP Bank0 Word3 (Configuration and Manufacturing Info.) (OCOTP_CFG2)	32	R/W	0000_0000h	47.5.12/ 4105
21B_C440	Value of OTP Bank0 Word4 (Configuration and Manufacturing Info.) (OCOTP_CFG3)	32	R/W	0000_0000h	47.5.13/ 4106
21B_C450	Value of OTP Bank0 Word5 (Configuration and Manufacturing Info.) (OCOTP_CFG4)	32	R/W	0000_0000h	47.5.14/ 4106
21B_C460	Value of OTP Bank0 Word6 (Configuration and Manufacturing Info.) (OCOTP_CFG5)	32	R/W	0000_0000h	47.5.15/ 4107
21B_C470	Value of OTP Bank0 Word7 (Configuration and Manufacturing Info.) (OCOTP_CFG6)	32	R/W	0000_0000h	47.5.16/ 4107
21B_C480	Value of OTP Bank1 Word0 (Memory Related Info.) (OCOTP_MEM0)	32	R/W	0000_0000h	47.5.17/ 4108
21B_C490	Value of OTP Bank1 Word1 (Memory Related Info.) (OCOTP_MEM1)	32	R/W	0000_0000h	47.5.18/ 4108
21B_C4A0	Value of OTP Bank1 Word2 (Memory Related Info.) (OCOTP_MEM2)	32	R/W	0000_0000h	47.5.19/ 4109
21B_C4B0	Value of OTP Bank1 Word3 (Memory Related Info.) (OCOTP_MEM3)	32	R/W	0000_0000h	47.5.20/ 4109
21B_C4C0	Value of OTP Bank1 Word4 (Memory Related Info.) (OCOTP_MEM4)	32	R/W	0000_0000h	47.5.21/ 4110

Table continues on the next page...

OCOTP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21B_C4D0	Value of OTP Bank1 Word5 (General Purpose Customer Defined Info.) (OCOTP_ANA0)	32	R/W	0000_0000h	47.5.22/ 4110
21B_C4E0	Value of OTP Bank1 Word6 (General Purpose Customer Defined Info.) (OCOTP_ANA1)	32	R/W	0000_0000h	47.5.23/ 4111
21B_C4F0	Value of OTP Bank1 Word7 (General Purpose Customer Defined Info.) (OCOTP_ANA2)	32	R/W	0000_0000h	47.5.24/ 4111
21B_C580	Shadow Register for OTP Bank3 Word0 (SRK Hash) (OCOTP_SRK0)	32	R/W	0000_0000h	47.5.25/ 4112
21B_C590	Shadow Register for OTP Bank3 Word1 (SRK Hash) (OCOTP_SRK1)	32	R/W	0000_0000h	47.5.26/ 4112
21B_C5A0	Shadow Register for OTP Bank3 Word2 (SRK Hash) (OCOTP_SRK2)	32	R/W	0000_0000h	47.5.27/ 4113
21B_C5B0	Shadow Register for OTP Bank3 Word3 (SRK Hash) (OCOTP_SRK3)	32	R/W	0000_0000h	47.5.28/ 4113
21B_C5C0	Shadow Register for OTP Bank3 Word4 (SRK Hash) (OCOTP_SRK4)	32	R/W	0000_0000h	47.5.29/ 4114
21B_C5D0	Shadow Register for OTP Bank3 Word5 (SRK Hash) (OCOTP_SRK5)	32	R/W	0000_0000h	47.5.30/ 4114
21B_C5E0	Shadow Register for OTP Bank3 Word6 (SRK Hash) (OCOTP_SRK6)	32	R/W	0000_0000h	47.5.31/ 4115
21B_C5F0	Shadow Register for OTP Bank3 Word7 (SRK Hash) (OCOTP_SRK7)	32	R/W	0000_0000h	47.5.32/ 4115
21B_C600	Value of OTP Bank4 Word0 (Secure JTAG Response Field) (OCOTP_SJC_RESP0)	32	R/W	0000_0000h	47.5.33/ 4116
21B_C610	Value of OTP Bank4 Word1 (Secure JTAG Response Field) (OCOTP_SJC_RESP1)	32	R/W	0000_0000h	47.5.34/ 4116
21B_C620	Value of OTP Bank4 Word2 (MAC Address) (OCOTP_MAC0)	32	R/W	0000_0000h	47.5.35/ 4117
21B_C630	Value of OTP Bank4 Word3 (MAC Address) (OCOTP_MAC1)	32	R/W	0000_0000h	47.5.36/ 4117
21B_C660	Value of OTP Bank4 Word6 (HW Capabilities) (OCOTP_GP1)	32	R/W	0000_0000h	47.5.37/ 4118
21B_C670	Value of OTP Bank4 Word7 (HW Capabilities) (OCOTP_GP2)	32	R/W	0000_0000h	47.5.38/ 4118
21B_C6D0	Value of OTP Bank5 Word5 (HW Capabilities) (OCOTP_MISC_CONF)	32	R/W	0000_0000h	47.5.39/ 4119
21B_C6E0	Value of OTP Bank5 Word6 (HW Capabilities) (OCOTP_FIELD_RETURN)	32	R/W	0000_0000h	47.5.40/ 4119
21B_C6F0	Value of OTP Bank5 Word7 (HW Capabilities) (OCOTP_SRK_REVOKE)	32	R/W	0000_0000h	47.5.41/ 4120

47.5.1 OTP Controller Control Register (OCOTP_CTRL*n*)

The OCOTP Control and Status Register specifies the copy state, as well as the control required for random access of the OTP memory

OCOTP_CTRL: 0x000

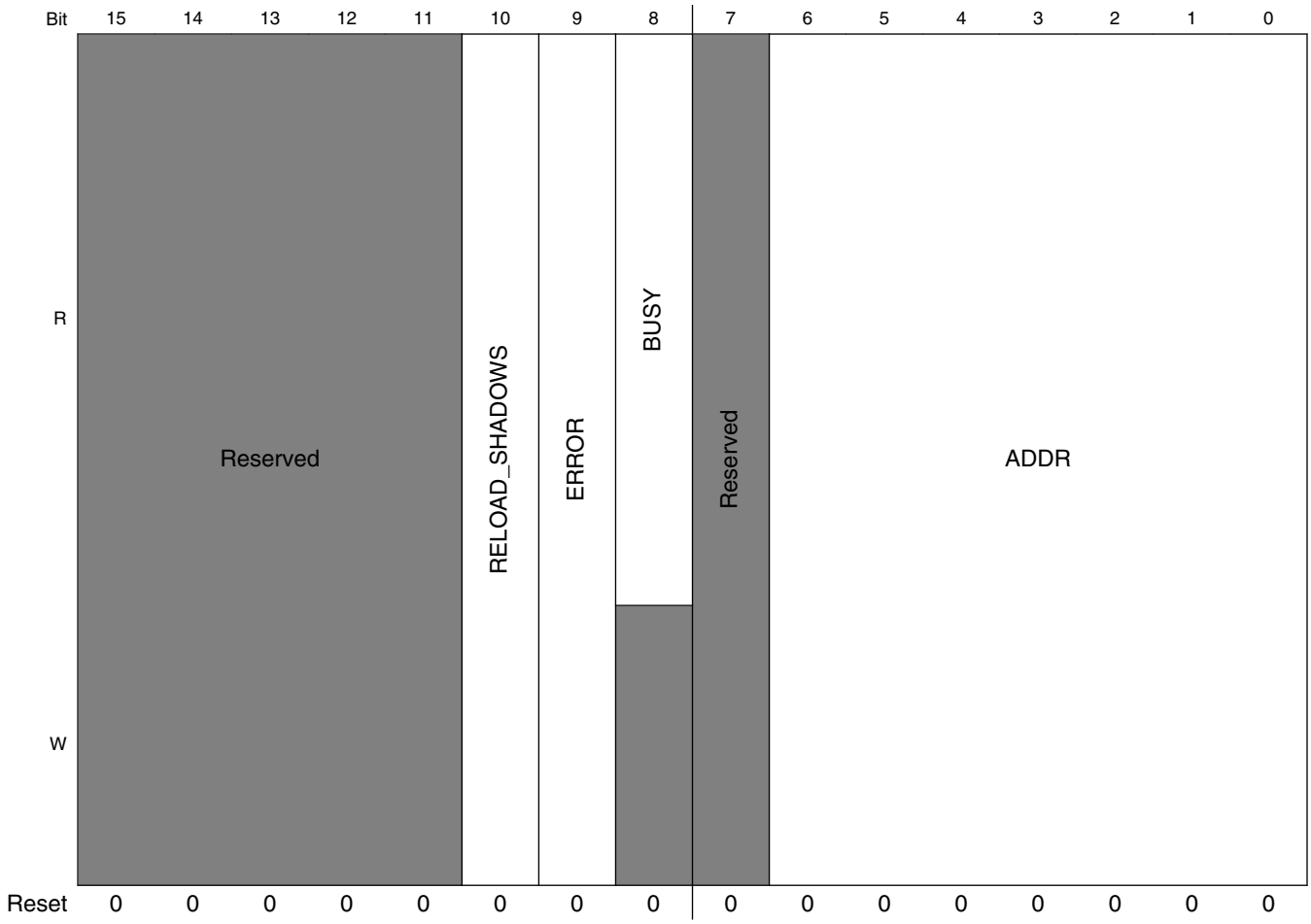
The OCOTP Control and Status Register provides the necessary software interface for performing read and write operations to the On-Chip OTP (One-Time Programmable ROM). The control fields such as WR_UNLOCK, ADDR and BUSY/ERROR may be used in conjuction with the HW_OCOTP_DATA register to perform write operations. Read operations to the On-Chip OTP are involving ADDR, BUSY/ERROR bit field and HW_OCOTP_READ_CTRL register. Read value is saved in HW_OCOTP_READ_FUSE_DATA register.

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WR_UNLOCK															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



OCOTP_CTRL*n* field descriptions

Field	Description
31–16 WR_UNLOCK	Write 0x3E77 to enable OTP write accesses. NOTE: This register must be unlocked on a write-by-write basis (a write is initiated when HW_OCOTP_DATA is written), so the UNLOCK bitfield must contain the correct key value during all writes to HW_OCOTP_DATA, otherwise a write shall not be initiated. This field is automatically cleared after a successful write completion (clearing of BUSY). 0x3E77 **KEY** — Key needed to unlock HW_OCOTP_DATA register.
15–11 -	This field is reserved. Reserved
10 RELOAD_ SHADOWS	Set to force re-loading the shadow registers (HW/SW capability and LOCK). This operation will automatically set BUSY. Once the shadow registers have been re-loaded, BUSY and RELOAD_SHADOWS are automatically cleared by the controller.
9 ERROR	Set by the controller when an access to a locked region(OTP or shadow register) is requested. Must be cleared before any further access can be performed. This bit can only be set by the controller. This bit is also set if the Pin interface is active and software requests an access to the OTP. In this instance, the ERROR bit cannot be cleared until the Pin interface access has completed. Reset this bit by writing a one to the SCT clear address space and not by a general write.
8 BUSY	OTP controller status bit. When active, no new write access or read access to OTP(including RELOAD_SHADOWS) can be performed. Cleared by controller when access complete. After reset (or after setting RELOAD_SHADOWS), this bit is set by the controller until the HW/SW and LOCK registers are successfully copied, after which time it is automatically cleared by the controller.

Table continues on the next page...

OCOTP_CTRL*n* field descriptions (continued)

Field	Description
7 -	This field is reserved. Reserved
ADDR	OTP write and read access address register. Specifies one of 128 word address locations (0x00 - 0x7f). If a valid access is accepted by the controller, the controller makes an internal copy of this value. This internal copy will not update until the access is complete.

47.5.2 OTP Controller Timing Register (OCOTP_TIMING)

The OCOTP Data Register is used for OTP Programming

This register specifies timing parameters for programming and reading the OCOTP fuse array.

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 10h offset = 21B_C010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				WAIT						STROBE_READ						RELAX				STROBE_PROG											
W																																
Reset	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	0	0	0	0	1	0	0	1	0	1	0	0	1	1	0	0	1

OCOTP_TIMING field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–22 WAIT	This count value specifies time interval between auto read and write access in one time program. It is given in number of ipg_clk periods.
21–16 STROBE_READ	This count value specifies the strobe period in one time read OTP. Trd = ((STROBE_READ+1)- 2*(RELAX +1)) /ipg_clk_freq. It is given in number of ipg_clk periods.
15–12 RELAX	This count value specifies the time to add to all default timing parameters other than the Tpgm and Trd. It is given in number of ipg_clk periods.
STROBE_PROG	This count value specifies the strobe period in one time write OTP. Tpgm = ((STROBE_PROG+1)- 2*(RELAX+1)) /ipg_clk_freq. It is given in number of ipg_clk periods.

47.5.3 OTP Controller Write Data Register (OCOTP_DATA)

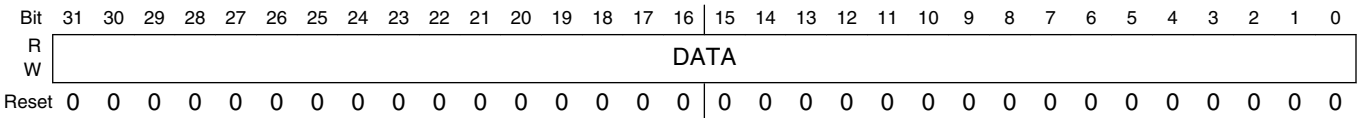
The OCOTP Data Register is used for OTP Programming

This register is used in conjuction with HW_OCOTP_CTRL to perform one-time writes to the OTP.

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 20h offset = 21B_C020h



OCOTP_DATA field descriptions

Field	Description
DATA	Used to initiate a write to OTP.

47.5.4 OTP Controller Write Data Register (OCOTP_READ_CTRL)

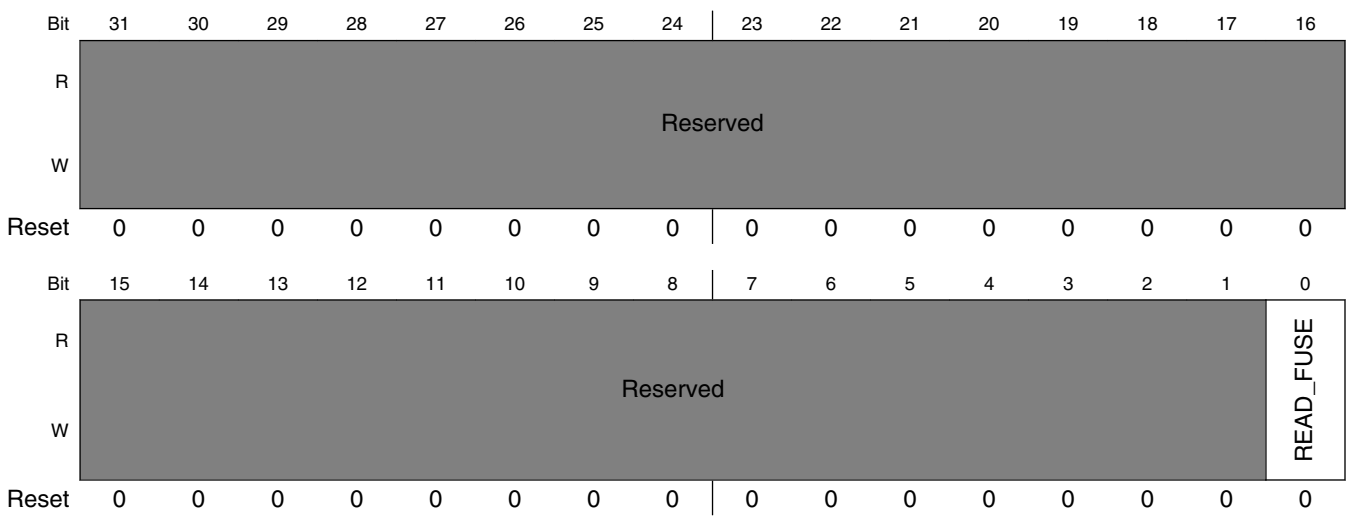
The OCOTP Register is used for OTP Read

This register is used in conjuction with HW_OCOTP_CTRL to perform one time read to the OTP.

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 30h offset = 21B_C030h



OCOTP_READ_CTRL field descriptions

Field	Description
31–1 -	This field is reserved. Reserved

Table continues on the next page...

OCOTP_READ_CTRL field descriptions (continued)

Field	Description
0 READ_FUSE	Used to initiate a read to OTP.

47.5.5 OTP Controller Read Data Register (OCOTP_READ_FUSE_DATA)

The OCOTP Data Register is used for OTP Read

The data read from OTP

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 40h offset = 21B_C040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_READ_FUSE_DATA field descriptions

Field	Description
DATA	The data read from OTP

47.5.6 Sticky bit Register (OCOTP_SW_STICKY)

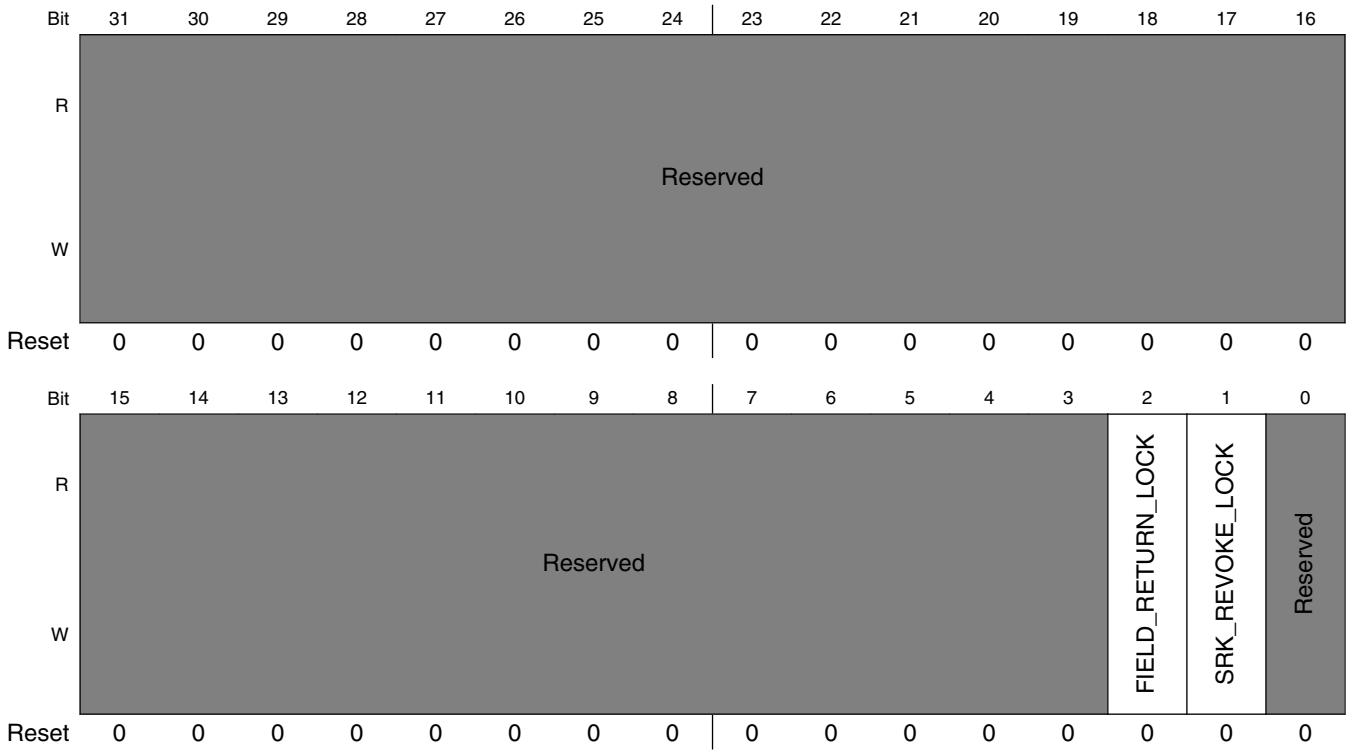
Some SW sticky bits .

Some sticky bits are used by SW to lock some fuse area , shadow registers and other features.

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 50h offset = 21B_C050h



OCOTP_SW_STICKY field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
2 FIELD_ RETURN_LOCK	Shadow register write and OTP write lock for FIELD_RETURN region. When set, the writing of this region's shadow register and OTP fuse word are blocked.Once this bit is set, it is always high unless a POR is issued.
1 SRK_REVOKE_ LOCK	Shadow register write and OTP write lock for SRK_REVOKE, MC_ERA and AP_BI_VER regions. When set, the writing of these region's shadow register and OTP fuse word are blocked. Once this bit is set, it is always high unless a POR is issued.
0 -	This field is reserved. Reserved.

47.5.7 Software Controllable Signals Register (OCOTP_SCS*n*)

HW_OCOTP_SCS: 0x060

This register holds volatile configuration values that can be set and locked by trusted software. All values are returned to their defualt values after POR.

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 60h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	SPARE														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SPARE															HAB_JDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_SCS*n* field descriptions

Field	Description
31 LOCK	When set, all of the bits in this register are locked and can not be changed through SW programming. This bit is only reset after a POR is issued.
30–1 SPARE	Unallocated read/write bits for implementation specific software use.
0 HAB_JDE	HAB JTAG Debug Enable. This bit is used by the HAB to enable JTAG debugging, assuming that a properly signed command to do so is found and validated by the HAB. The HAB must lock the register before passing control to the OS whether or not JTAG debugging has been enabled. Once JTAG is enabled by this bit, it can not be disabled unless the system is reset by POR. 0: JTAG debugging is not enabled by the HAB (it may still be enabled by other mechanisms). 1: JTAG debugging is enabled by the HAB (though this signal may be gated off). 1 JTAG debugging is enabled by the HAB (though this signal may be gated off)

47.5.8 OTP Controller Version Register (OCOTP_VERSION)

This register always returns a known read value for debug purposes it indicates the version of the block.

This register indicates the RTL version in use.

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 90h offset = 21B_C090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR								MINOR								STEP															
W																																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_VERSION field descriptions

Field	Description
31–24 MAJOR	Fixed read-only value reflecting the MAJOR field of the RTL version.
23–16 MINOR	Fixed read-only value reflecting the MINOR field of the RTL version.
STEP	Fixed read-only value reflecting the stepping of the RTL version.

47.5.9 Value of OTP Bank0 Word0 (Lock controls) (OCOTP_LOCK)

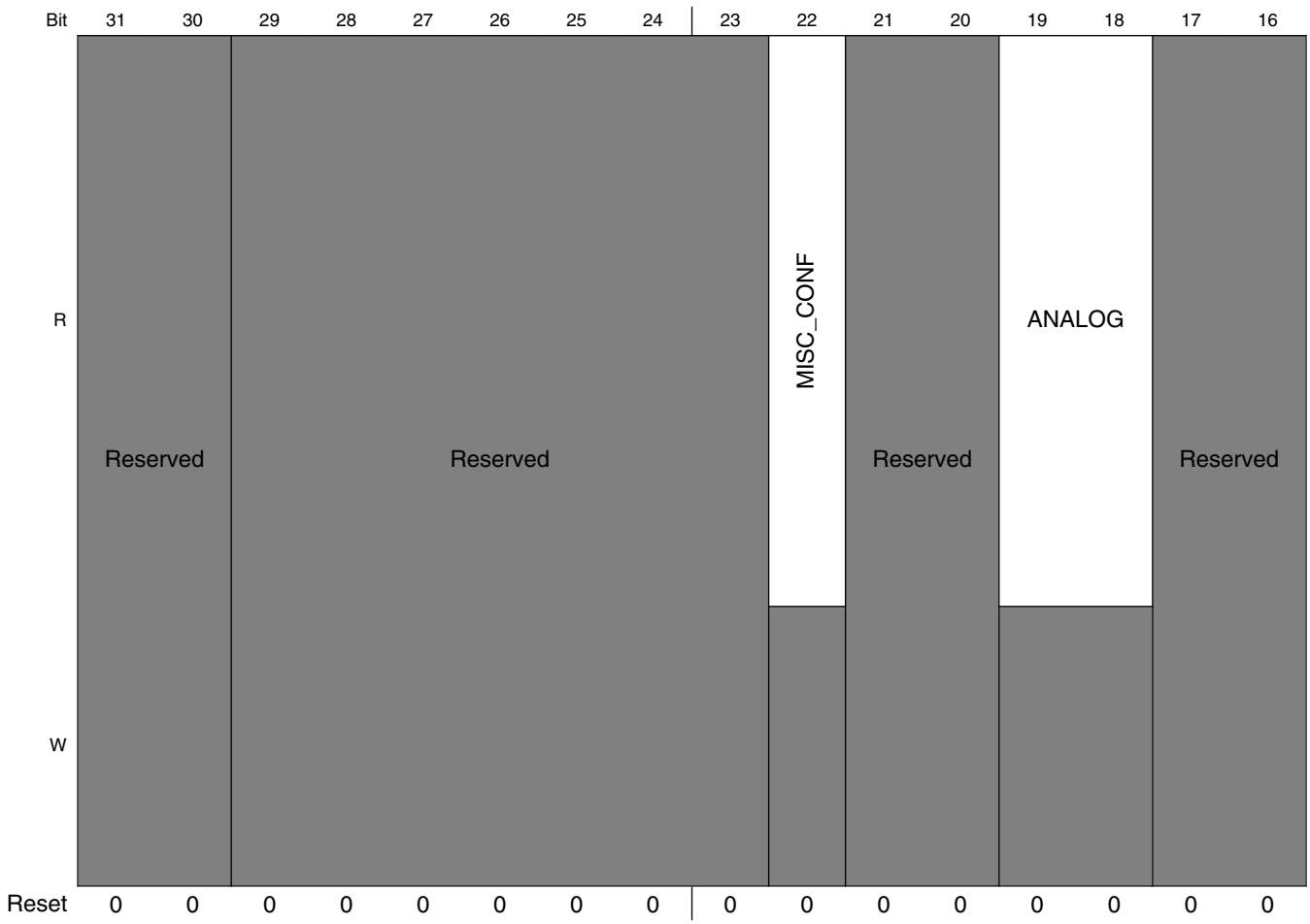
Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

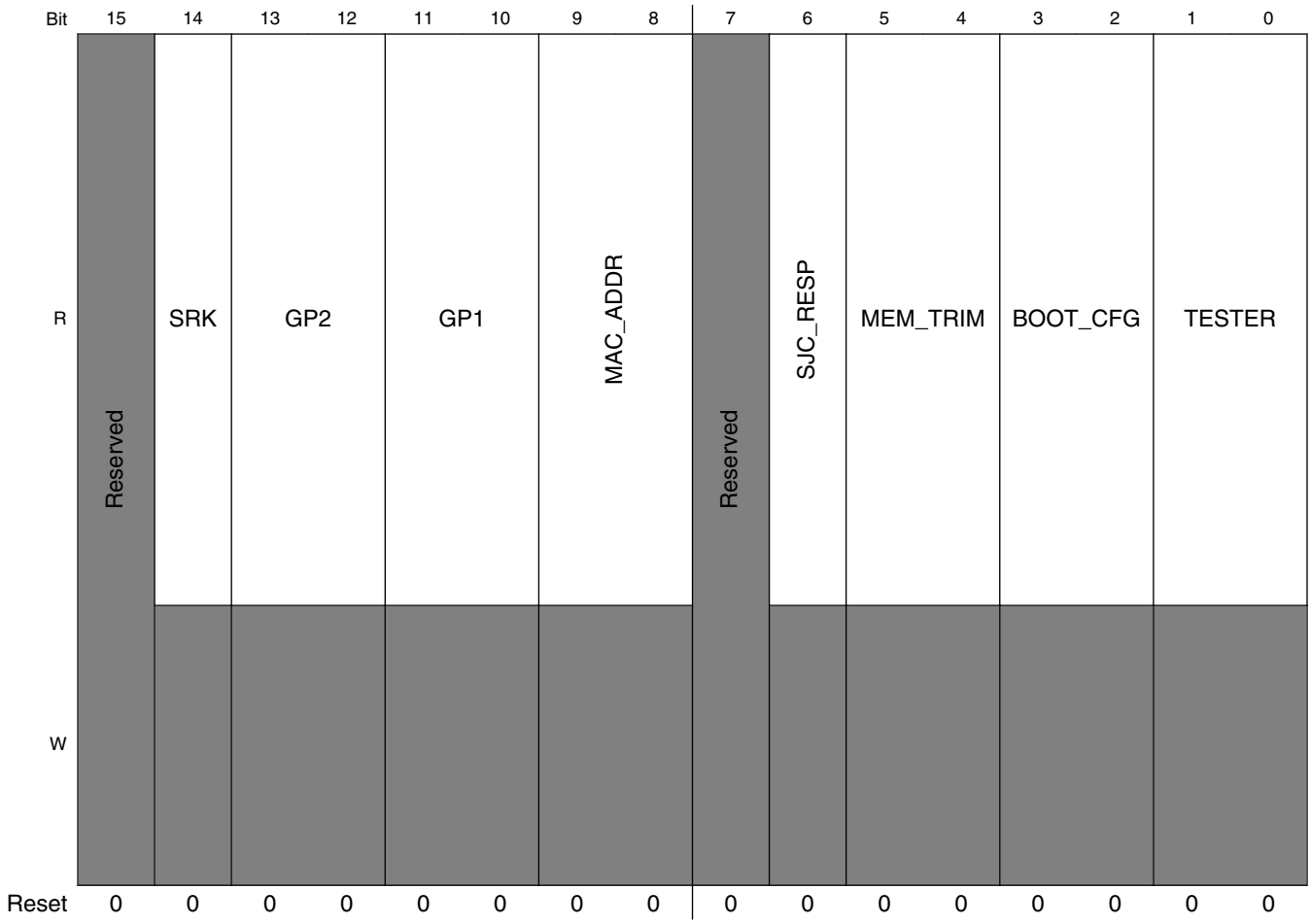
Shadowed memory mapped access to OTP Bank 0, word 0 (ADDR = 0x00).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 400h offset = 21B_C400h





OCOTP_LOCK field descriptions

Field	Description
31–30 -	This field is reserved. Reserved
29–23 -	This field is reserved. Reserved
22 MISC_CONF	Status of shadow register and OTP write lock for misc_conf region. When set, the writing of this region's shadow register and OTP fuse word are blocked.
21–20 -	This field is reserved. Reserved
19–18 ANALOG	Status of shadow register and OTP write lock for analog region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
17–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 SRK	Status of shadow register and OTP write lock for srk region. When set, the writing of this region's shadow register and OTP fuse word are blocked.

Table continues on the next page...

OCOTP_LOCK field descriptions (continued)

Field	Description
13–12 GP2	Status of shadow register and OTP write lock for gp2 region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
11–10 GP1	Status of shadow register and OTP write lock for gp2 region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
9–8 MAC_ADDR	Status of shadow register and OTP write lock for mac_addr region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
7 -	This field is reserved. Reserved
6 SJC_RESP	Status of shadow register read and write, OTP read and write lock for sjc_resp region. When set, the writing of this region's shadow register and OTP fuse word are blocked. The read of this region's shadow register and OTP fuse word are also blocked.
5–4 MEM_TRIM	Status of shadow register and OTP write lock for mem_trim region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
3–2 BOOT_CFG	Status of shadow register and OTP write lock for boot_cfg region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
TESTER	Status of shadow register and OTP write lock for tester region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.

47.5.10 Value of OTP Bank0 Word1 (Configuration and Manufacturing Info.) (OCOTP_CFG0)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP Bank 0, word 1 (ADDR = 0x01).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 410h offset = 21B_C410h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CFG0 field descriptions

Field	Description
BITS	This register contains 32 bits of the Unique ID and SJC_CHALLENGE field. Reflects value of OTP Bank 0, word 1 (ADDR = 0x01). These bits become read-only after the HW_OCOTP_LOCK_TESTER[1] bit is set.

47.5.11 Value of OTP Bank0 Word2 (Configuration and Manufacturing Info.) (OCOTP_CFG1)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

shadowed memory mapped access to OTP Bank 0, word 2 (ADDR = 0x02).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 420h offset = 21B_C420h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CFG1 field descriptions

Field	Description
BITS	This register contains 32 bits of the Unique ID and SJC_CHALLENGE field. Reflects value of OTP Bank 0, word 2 (ADDR = 0x02). These bits become read-only after the HW_OCOTP_LOCK_TESTER[1] bit is set.

47.5.12 Value of OTP Bank0 Word3 (Configuration and Manufacturing Info.) (OCOTP_CFG2)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP Bank 0, word 3 (ADDR = 0x03).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 430h offset = 21B_C430h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CFG2 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 0, word 3 (ADDR = 0x03). These bits become read-only after the HW_OCOTP_LOCK_TESTER[1] bit is set.

47.5.13 Value of OTP Bank0 Word4 (Configuration and Manufacturing Info.) (OCOTP_CFG3)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Non-shadowed memory mapped access to OTP Bank 0, word 4 (ADDR = 0x04).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 440h offset = 21B_C440h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CFG3 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 0, word 4 (ADDR = 0x04). These bits become read-only after the HW_OCOTP_LOCK_TESTER[1] bit is set.

47.5.14 Value of OTP Bank0 Word5 (Configuration and Manufacturing Info.) (OCOTP_CFG4)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP Bank 0, word 5 (ADDR = 0x05).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 450h offset = 21B_C450h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CFG4 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 0, word 5 (ADDR = 0x05). These bits become read-only after the HW_OCOTP_LOCK_BOOT_CFG[1] bit is set.

47.5.15 Value of OTP Bank0 Word6 (Configuration and Manufacturing Info.) (OCOTP_CFG5)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP Bank 0, word 6 (ADDR = 0x06).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 460h offset = 21B_C460h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CFG5 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 0, word 6 (ADDR = 0x06). These bits become read-only after the HW_OCOTP_LOCK_BOOT_CFG[1] bit is set.

47.5.16 Value of OTP Bank0 Word7 (Configuration and Manufacturing Info.) (OCOTP_CFG6)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP Bank 0, word 7 (ADDR = 0x07).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 470h offset = 21B_C470h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CFG6 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 0, word 7 (ADDR = 0x07). These bits become read-only after the HW_OCOTP_LOCK_BOOT_CFG[1] bit is set.

47.5.17 Value of OTP Bank1 Word0 (Memory Related Info.) (OCOTP_MEM0)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP bank 1, word 0 (ADDR = 0x08).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 480h offset = 21B_C480h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_MEM0 field descriptions

Field	Description
BITS	Reflects value of OTP bank 1, word 0 (ADDR = 0x08). These bits become read-only after the HW_OCOTP_LOCK_MEM_TRIM[1] bit is set.

47.5.18 Value of OTP Bank1 Word1 (Memory Related Info.) (OCOTP_MEM1)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP bank 1, word 1 (ADDR = 0x09).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 490h offset = 21B_C490h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_MEM1 field descriptions

Field	Description
BITS	Reflects value of OTP bank 1, word 1 (ADDR = 0x09). These bits become read-only after the HW_OCOTP_LOCK_MEM_TRIM[1] bit is set.

47.5.19 Value of OTP Bank1 Word2 (Memory Related Info.) (OCOTP_MEM2)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP bank 1, word 2 (ADDR = 0x0A).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 4A0h offset = 21B_C4A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_MEM2 field descriptions

Field	Description
BITS	Reflects value of OTP bank 1, word 2 (ADDR = 0x0A). These bits become read-only after the HW_OCOTP_LOCK_MEM_TRIM[1] bit is set.

47.5.20 Value of OTP Bank1 Word3 (Memory Related Info.) (OCOTP_MEM3)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP bank 1, word 3 (ADDR = 0x0B).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 4B0h offset = 21B_C4B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_MEM3 field descriptions

Field	Description
BITS	Reflects value of OTP bank 1, word 3 (ADDR = 0x0B). These bits become read-only after the HW_OCOTP_LOCK_MEM_TRIM[1] bit is set.

47.5.21 Value of OTP Bank1 Word4 (Memory Related Info.) (OCOTP_MEM4)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP bank 1, word 4 (ADDR = 0x0C).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 4C0h offset = 21B_C4C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_MEM4 field descriptions

Field	Description
BITS	Reflects value of OTP bank 1, word 4 (ADDR = 0x0C). These bits become read-only after the HW_OCOTP_LOCK_MEM_TRIM[1] bit is set.

47.5.22 Value of OTP Bank1 Word5 (General Purpose Customer Defined Info.) (OCOTP_ANA0)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP bank 1, word 5 (ADDR = 0x0D).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 4D0h offset = 21B_C4D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_ANA0 field descriptions

Field	Description
BITS	Reflects value of OTP bank 1, word 5 (ADDR = 0x0D). These bits become read-only after the HW_OCOTP_LOCK_ANALOG[1] bit is set.

47.5.23 Value of OTP Bank1 Word6 (General Purpose Customer Defined Info.) (OCOTP_ANA1)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP bank 1, word 6 (ADDR = 0x0E).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 4E0h offset = 21B_C4E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_ANA1 field descriptions

Field	Description
BITS	Reflects value of OTP bank 1, word 6 (ADDR = 0x0E). These bits become read-only after the HW_OCOTP_LOCK_ANALOG[1] bit is set.

47.5.24 Value of OTP Bank1 Word7 (General Purpose Customer Defined Info.) (OCOTP_ANA2)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP bank 1, word 7 (ADDR = 0x0F).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 4F0h offset = 21B_C4F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_ANA2 field descriptions

Field	Description
BITS	Reflects value of OTP bank 1, word 7 (ADDR = 0x0F). These bits become read-only after the HW_OCOTP_LOCK_ANALOG[1] bit is set.

47.5.25 Shadow Register for OTP Bank3 Word0 (SRK Hash) (OCOTP_SRK0)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS].

Shadowed memory mapped access to OTP Bank 3, word 0 (ADDR = 0x18).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 580h offset = 21B_C580h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_SRK0 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word0 (Copy of OTP Bank 3, word 0 (ADDR = 0x1C)). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

47.5.26 Shadow Register for OTP Bank3 Word1 (SRK Hash) (OCOTP_SRK1)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS].

Shadowed memory mapped access to OTP Bank 3, word 1 (ADDR = 0x19).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 590h offset = 21B_C590h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_SRK1 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word1 (Copy of OTP Bank 3, word 1 (ADDR = 0x1D)). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

47.5.27 Shadow Register for OTP Bank3 Word2 (SRK Hash) (OCOTP_SRK2)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS].

Shadowed memory mapped access to OTP Bank 3, word 2 (ADDR = 0x1A).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 5A0h offset = 21B_C5A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_SRK2 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word2 (Copy of OTP Bank 3, word 2 (ADDR = 0x1E)). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

47.5.28 Shadow Register for OTP Bank3 Word3 (SRK Hash) (OCOTP_SRK3)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS].

Shadowed memory mapped access to OTP Bank 3, word 3 (ADDR = 0x1B).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 5B0h offset = 21B_C5B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_SRK3 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word3 (Copy of OTP Bank 3, word 3 (ADDR = 0x1F)). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

47.5.29 Shadow Register for OTP Bank3 Word4 (SRK Hash) (OCOTP_SRK4)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS].

Shadowed memory mapped access to OTP Bank 3, word 4 (ADDR = 0x1C).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 5C0h offset = 21B_C5C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_SRK4 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word4 (Copy of OTP Bank 3, word 4 (ADDR = 0x20)). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

47.5.30 Shadow Register for OTP Bank3 Word5 (SRK Hash) (OCOTP_SRK5)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS].

Shadowed memory mapped access to OTP Bank 3, word 5 (ADDR = 0x1D).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 5D0h offset = 21B_C5D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_SRK5 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word5 (Copy of OTP Bank 3, word 5 (ADDR = 0x21)). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

47.5.31 Shadow Register for OTP Bank3 Word6 (SRK Hash) (OCOTP_SRK6)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS].

Shadowed memory mapped access to OTP Bank 3, word 6 (ADDR = 0x1E).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 5E0h offset = 21B_C5E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_SRK6 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word6 (Copy of OTP Bank 3, word 6 (ADDR = 0x22)). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

47.5.32 Shadow Register for OTP Bank3 Word7 (SRK Hash) (OCOTP_SRK7)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS].

Shadowed memory mapped access to OTP Bank 3, word 7 (ADDR = 0x1F).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 5F0h offset = 21B_C5F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_SRK7 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word7 (Copy of OTP Bank 3, word 7 (ADDR = 0x23)). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

47.5.33 Value of OTP Bank4 Word0 (Secure JTAG Response Field) (OCOTP_SJC_RESP0)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP Bank 4, word 0 (ADDR = 0x20).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 600h offset = 21B_C600h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_SJC_RESP0 field descriptions

Field	Description
BITS	Shadow register for the SJC_RESP Key word0 (Copy of OTP Bank 4, word 0 (ADDR = 0x20)). These bits can be not read and written after the HW_OCOTP_LOCK_SJC_RESP bit is set. If read, returns 0xBADA_BADA and sets HW_OCOTP_CTRL[ERROR].

47.5.34 Value of OTP Bank4 Word1 (Secure JTAG Response Field) (OCOTP_SJC_RESP1)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP Bank 4, word 1 (ADDR = 0x21).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 610h offset = 21B_C610h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_SJC_RESP1 field descriptions

Field	Description
BITS	Shadow register for the SJC_RESP Key word1 (Copy of OTP Bank 4, word 1 (ADDR = 0x21)). These bits can be not read and written after the HW_OCOTP_LOCK_SJC_RESP bit is set. If read, returns 0xBADA_BADA and sets HW_OCOTP_CTRL[ERROR].

47.5.35 Value of OTP Bank4 Word2 (MAC Address) (OCOTP_MAC0)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP Bank 4, word 2 (ADDR = 0x22).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 620h offset = 21B_C620h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_MAC0 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 4, word 2 (ADDR = 0x22).

47.5.36 Value of OTP Bank4 Word3 (MAC Address) (OCOTP_MAC1)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP Bank 4, word 3 (ADDR = 0x23).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 630h offset = 21B_C630h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_MAC1 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 4, word 3 (ADDR = 0x23).

47.5.37 Value of OTP Bank4 Word6 (HW Capabilities) (OCOTP_GP1)

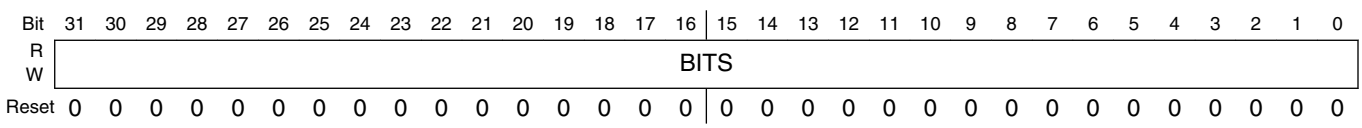
Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP Bank 4, word 6 (ADDR = 0x26).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 660h offset = 21B_C660h



OCOTP_GP1 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 4, word 6 (ADDR = 0x26).

47.5.38 Value of OTP Bank4 Word7 (HW Capabilities) (OCOTP_GP2)

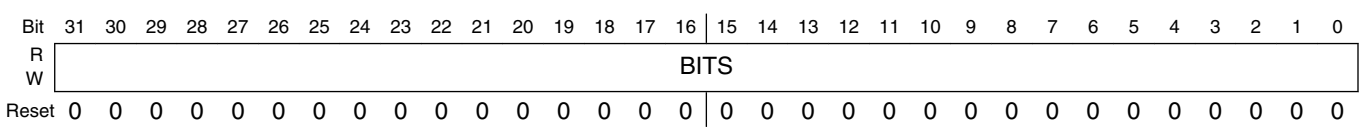
Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP Bank 4, word 7 (ADDR = 0x27).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 670h offset = 21B_C670h



OCOTP_GP2 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 4, word 7 (ADDR = 0x27).

47.5.39 Value of OTP Bank5 Word5 (HW Capabilities) (OCOTP_MISC_CONF)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP Bank 5, word 5 (ADDR = 0x2d).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 6D0h offset = 21B_C6D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_MISC_CONF field descriptions

Field	Description
BITS	Reflects value of OTP Bank 5, word 5 (ADDR = 0x2d).

47.5.40 Value of OTP Bank5 Word6 (HW Capabilities) (OCOTP_FIELD_RETURN)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP Bank 5, word 6 (ADDR = 0x2e).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 6E0h offset = 21B_C6E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_FIELD_RETURN field descriptions

Field	Description
BITS	Reflects value of OTP Bank 5, word 6 (ADDR = 0x2e).

47.5.41 Value of OTP Bank5 Word7 (HW Capabilities) (OCOTP_SRK_REVOKE)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

Shadowed memory mapped access to OTP Bank 5, word 7 (ADDR = 0x2f).

EXAMPLE

```
Empty Example.
```

Address: 21B_C000h base + 6F0h offset = 21B_C6F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_SRK_REVOKE field descriptions

Field	Description
BITS	Reflects value of OTP Bank 5, word 7 (ADDR = 0x2f).

Chapter 48
On-Chip RAM Memory Controller (OCRAM)

48.1 Overview

The on-chip RAM block is implemented as an slave module on the 64-bit system AXI bus. Designed as a simple on-chip memory controller, it supports only one AXI port with memory banks. For the AXI port, the read and write transactions are handled by two independent modules. As it is possible to have simultaneous read and write request from the AXI bus, each memory bank has an arbiter with round-robin scheme. After arbitration, the granted read or write access command can then be issued to the memory cell through a read/write MUX.

The memory banks are organized with the lower 2 bits of the address which is the AXI bus address and is 64 bits aligned interleaved. This allows a read access and a write access to be processed at the same time if they are targeted to different memory banks.

Various options are provided for adding a pipeline or wait-states in a read/write access, in order to ensure flexible timing control at both high and low frequencies.

The internal block diagram is shown in the figure below.

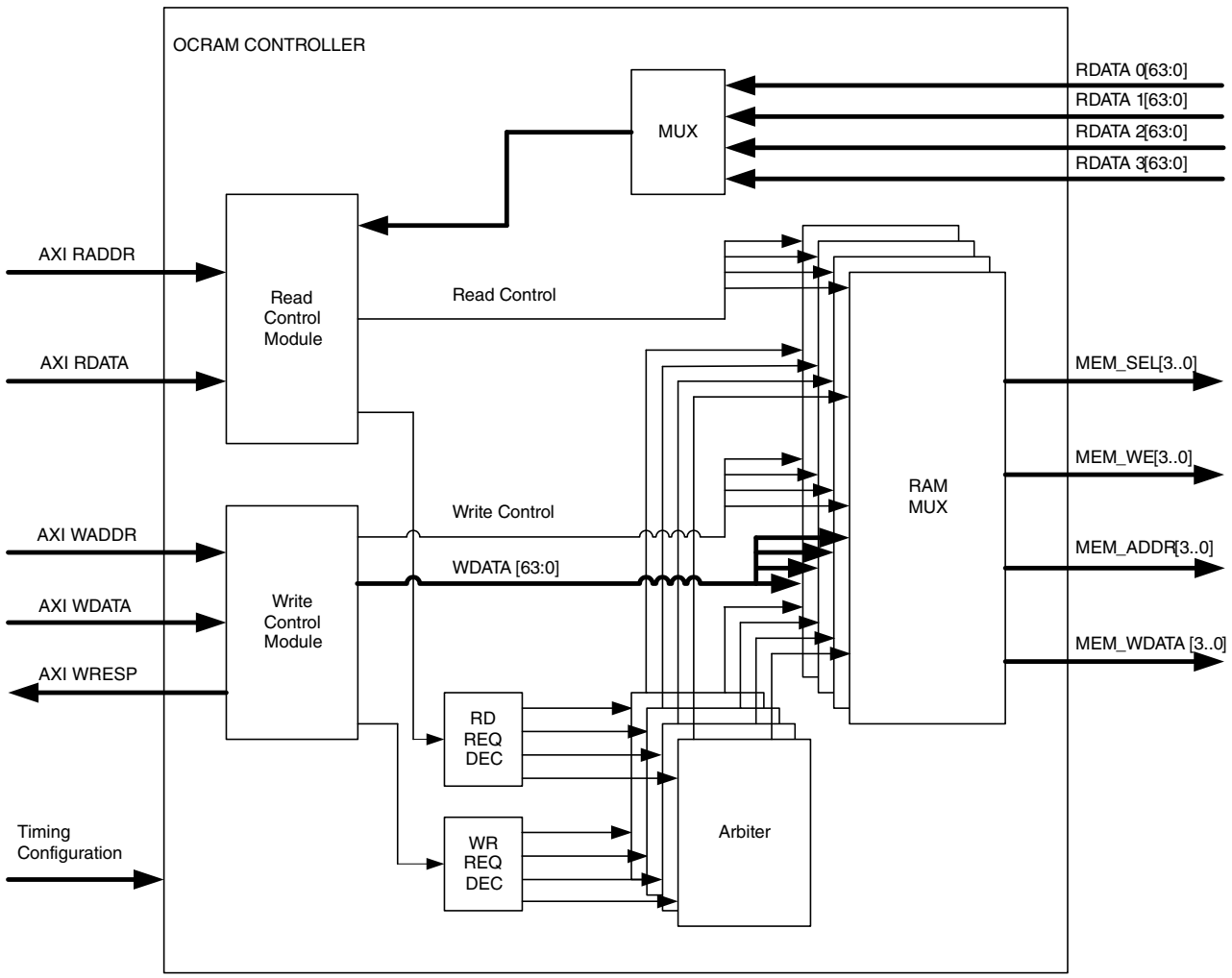


Figure 48-1. On-chip RAM Block Diagram

48.2 Basic Functions

48.2.1 Memory Map

The total on-chip RAM size for the chip is 128 Kbytes, organized as 16K x 64 bits, mapped from 0x00900000 to 0x0091FFFF .

Memory alias is also supported for on-chip RAM. Address from 0x00920000 to 0x009FFFFF are alias addresses for it, any read/write operation to this area will be mapped back to the 128 Kbytes area.

The 32-bit AXI address are arranged as shown in the table below.

Table 48-1. On-Chip RAM Address Bits

AHB ADDR Bits	Usage	Description
29:5	Address	Selects one of the 16 K words in the memory (64-bit a word).
4:3	Bank Address	Selects one of the Bank memory in the 4 memory Bank (64-bit a word).
2:0	Byte Address	Selects/masks out specific bytes within a word.

48.2.2 Read/Write Arbitration

The arbiter is used to handle the read/write request from the read control module and write control module. It grants read/write access to the memory cell according to the internal arbitration state and the request/switch signal from the read control module and write control module. Generally, the arbitration is using the round-robin method.

The detailed rules used in arbitration are as follows:

- If there is no granted read or write in the last cycle, and there is only a read request or a write request, the request will be granted.
- If there is no granted read or write in the last cycle, and there are both read or write requests coming in at the same time, the read request will be granted first.
- If a granted read/write transaction has just finished, the write/read request will have the higher priority in the next cycle.
- If the first read/write access request in a transaction is granted, all the data transfer in this burst will be finished before the next arbitration begins, that is, the round-robin arbitration mechanism is based on AXI transaction, not data access.

48.3 Advanced Features

This section describes some advanced features designed to avoid timing issues when the on-chip RAM is working at high frequency.

All of the features can be disabled/enabled by programming the corresponding fields of the General Purpose Register (IOMUXC.GPR3) bits [24:21] in the IOMUX chapter.

48.3.1 Read Data Wait State

When the wait state is enabled, it will take 2 cycles for each read access (each beat of a read burst).

This can avoid the potential timing problem caused by the longer memory access time at higher frequency.

When this feature is disabled, it only takes 1 clock cycle to finish a read transaction. That is, read data is available in the next cycle of read request becomes valid on the bus.

The read data wait state is configurable via IOMUXC.GPR3[21].

48.3.2 Read Address Pipeline

When this feature is enabled, the read address from the AXI master is delayed 1 cycle before it can be accepted by the on-chip RAM.

This can avoid setup time issues for the read access on the memory cell at high frequency. Enabling this feature can cost, at most, 1 more clock cycle for each AXI read transaction, that is, at most 1 more clock cycle for each read burst with multiple beats of data.

When this feature is disabled, the read address from the AXI master can be accepted by the on-chip RAM without delay, and data can become ready for master at next clock cycle (if no other access and no read data wait).

The read address pipeline is configurable via IOMUXC.GPR3[22].

48.3.3 Write Data Pipeline

When this feature is enabled, the write data from the AXI master would be delayed 1 cycle before it can be accepted by the on-chip RAM.

This can avoid setup time issue for the write access on the memory cell at high frequency. Enabling this feature would cost at most 1 more clock cycle for each AXI write transaction, that is, at most 1 more clock cycle for each write burst with multiple beats of data.

When this feature is disabled, the write data from the AXI master can be accepted by the on-chip RAM without delay, and data can be written to memory at this cycle (if no other access and write address is also ready at this cycle).

The write data pipeline is configurable via IOMUXC.GPR3[23].

48.3.4 Write Address Pipeline

When this feature is enabled, the write address from the AXI master would be delayed 1 cycle before it can be accepted by the on-chip RAM.

This can avoid setup time issue for the write access on the memory cell at high frequency. Enabling this feature would take at most 1 more clock cycle for each AXI write transaction, that is, at most 1 more clock cycle for each write burst with multiple beats of data.

When this feature is disabled, the write address from the AXI master can be accepted by the on-chip RAM without delay, and data can be written to memory at this cycle (if no other access and write data is also ready at this cycle).

The write address pipeline is configurable via IOMUXC.GPR3[24].

48.4 Programmable Registers

There are no programmable registers in this block; however, OCRAM configurable bits can be found in the IOMUX Controller (IOMUXC) general purpose registers found here.

- TrustZone bits: IOMUXC_GPR10
- WAIT state / Pipeline bits: IOMUXC_GPR3
- L2 Cache OCRAM enable bits: IOMUXC_GPR11

(See IOMUXC Memory Map/Register Definition for more details).

Chapter 49
PCI Express (PCIe)

49.1 Overview

PCI Express includes the following cores:

- PCI Express Dual Mode (DM) core
- PCI Express Root Complex (RC) core
- PCI Express Endpoint (EP) core

NOTE

Throughout this chapter, content that is specific to a core is identified by using the abbreviated core name; for example, DM, RC, EP. Content that applies to the DM core specifically in either RC mode or EP mode is further qualified by using RC Mode or EP Mode. Content that applies to all cores is not identified as being specific to any core; rather, the term core or cores is used.

49.1.1 Terms and Abbreviations

The following terms are used throughout this chapter:

Table 49-1. Terms and Abbreviations

Term	Description
DM	PCI Express Dual Mode (DM) core
RC	PCI Express Root Complex (RC) core
EP	PCI Express Endpoint (EP) core
PCIe	PCI Express
CXPL	Common Xpress Port Logic: An internal Port Logic core module that implements the majority of the PCI Express protocol
x1/x2/x4/x8/x16	1/2/4/8/16 lanes

Table continues on the next page...

Table 49-1. Terms and Abbreviations (continued)

PIPE	PHY Interface for the PCI Express Architecture
NW	Number of double words; 1 stands for a 32-bit DWORD
DW	Data width: 32, 64, or 128 bits
NF	Number of functions; 1 stands for one function
TLP	Transaction Layer Packet
DLLP	Data Link Layer Packet
VC	Virtual Channel
BAR	Base Address Register
XADM	Transmit application-dependent module.
RADM	Receive application-dependent module.
PMC	Power management controller.
RAMI	External RAM interface.
Core	Identifies the entire core. The core includes the native PCIe-core and AXI/AHB bridge if present.
Native PCIe core	Identifies the basic PCIe core which has its own non-standard, proprietary dedicated bus interface to the application.
CDM	Configuration Dependent Module This is an internal block in the native core that houses the PCI configuration registers and some user-accessible registers that reside in the core. In general, an application may use our core, but will add other registers that are unique to their applications. Those new application registers will be referred to as External Application Registers (EAR).
LBC	Local Bus Controller This is an internal block that resides in the native core. It allows the DBI interface (from the application side) or the wire side interface (via the radm_TRGT0 interface) to access the core's Configuration Dependent Module (CDM) registers and/or the External Application Registers (EAR). (For additional details on this module, please refer to the native core documentation.)
ELBI	External Local Bus Interface This is an interface on the native core that processes read/write access to the external application registers (EAR). For applications that require external registers, the application can access the EAR through the bridge or an entirely different interface (outside the scope of this core). (For additional details on this interface, please refer to the native core's documentation.)
DBI	Data Bus Interface This bus is internal to the native core. This interface is used to access the core's internal registers (CDM) or the external application's device-specific registers attached to the PCIe native core ELBI interface.
VMI	Vendor Message Interface.
Inbound traffic	PCIe transactions that enter the native core from the wire side of the core (PCIe wire). These transactions will be delivered to the application side.
Outbound traffic	Transactions that enter the native core from the application side of the core. These are passed to the native core, where they will be sent out onto the PCIe wire.
MTU	Maximum Transfer Unit Specifies the maximum packet payload size supported. This indicates the maximum allowed transfer size for a write or completion.
Page boundary	Specifies the address page boundary size supported by the bridge. No packet can have an address that crosses the specified address boundary.
big-endian	Data format in which most significant byte comes first; normal order of bytes in a word.

Table continues on the next page...

Table 49-1. Terms and Abbreviations (continued)

RBYP	Receive Bypass Interface
RTRGT1	Receive Target 1 Interface
RCPL	Receive Completion Interface
XALI0/1/2	Transmit Application Client Interfaces
SII	System Information Interface.
iATU	Internal Address Translation Unit.

49.2 Architecture

This information describes the architecture of the PCI Express core.

The topics for this section are:

- Common Xpress Port Logic (CXPL)
- Transmit Application-Dependent Module (XADM)
- Receive Application-Dependent Module (RADM)
- Configuration-Dependent Module (CDM)
- Power Management Controller (PMC)
- Local Bus Controller (LBC) and Data bus Interface (DBI)
- Message Generation
- Debug and Diagnostics

The Common Xpress Port Logic (CXPL) module implements the basic functionality for the PCI Express Physical, Link, and Transaction Layers. In addition to the CXPL, there are several top-level modules that provide the configurationim and mode- specific features:

- Transmit Application-Dependent Module (XADM)
- Receive Application-Dependent Module (RADM)
- Configuration-Dependent Module (CDM)
- Power Management Controller (PMC)
- Local Bus Controller (LBC) and Data bus Interface (DBI)
- Message Generation

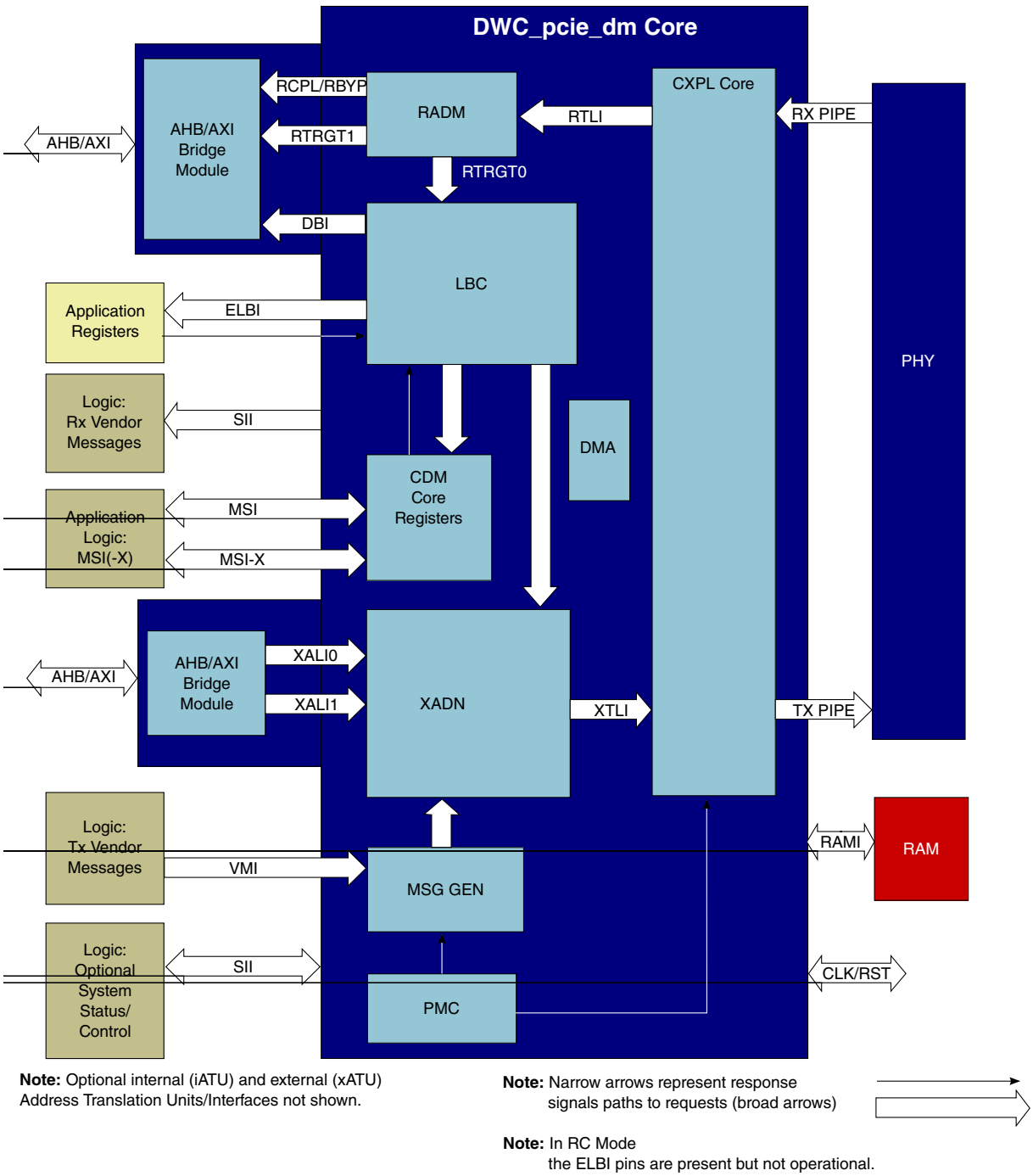


Figure 49-1. DM Core Block Diagram (with AXI Bridge Module)

NOTE

- Optional internal (iATU) and external (xATU) Address Translation Units/Interfaces not shown.
- Narrow arrows represent response signal paths to requests (broad arrows)
- In RC mode the ELBI pins are present but not operational.

For definitions of acronyms used for block and interface names, see Terms and Abbreviations.

Table 49-2. Core Interface Summary

Interface	Function
Transmit Client 0 Interface (XALI0)	Transmit interface for outbound Request or Completion TLPs.
Transmit Client 1 Interface (XALI1)	Additional application transmit interface, identical to XALI0. The usage of XALI0 and XALI1 is up to the application. For example, an application may use XALI0 for Requests and XALI1 for Completions.
Receive Completion Interface (RCPL)/Bypass Interface (RBYP)	When the core is configured in single or multiple queue architecture, RCPL delivers Completions received by the core to the application client that requested them. When the core is configured in segmented buffer queue mode, Bypass delivers all transactions that are configured as bypass.
Receive Target 1 Interface (RTRGT1)	Delivers inbound Requests received by the core after the Requests are qualified by the filter rules of the core.
Data Bus Interface (DBI)	Delivers an RD/WR request from application logic such as EEPROM to internal registers of the core or application registers at ELBI.
Message Signaled Interrupt (MSI) Interface	Allows the application to request a transmission of an MSI independent from the client interfaces. MSI interface is used only in EP mode.
MSI-X Interface	Allows the application to request a transmission of an MSI-X independent from the client interfaces. MSI-X interface is used only in EP mode.
Vendor Message Interface (VMI)	Allows the application to request a transmission of a vendor message independent from the client interfaces.
System Information Interface (SII)	Exchanges system information between the core and the application.
PIPE	Standard PIPE interface between the PCI Express PHY and the core.
RAM Interface (RAMI)	Optional top-level interface to connect external RAMs for the retry buffer and receive queues. If you do not select the optional top-level RAMI, the RAMs reside inside the top-level hierarchy of the core.

49.2.1 Common Xpress Port Logic (CXPL)

The CXPL module implements a large portion of the Transaction Layer logic, all of the Data Link Layer logic, and the MAC portion of the Physical Layer, including the Link Training and Status State Machine (LTSSM).

The CXPL connects to the external PHY though the PIPE.

Important aspects of the CXPL and overall core implementation include:

- Layer 3 (Transaction Layer) functionality is split between the XADM, RADM, CDM, and CXPL.

- Layer 1(Physical Layer) is split across the PIPE such that the MAC functionality is in the core and the PHY functionality is implemented in the PIPE-compliant PHY.the PHYmodule resides outside of the core, interfacing through the standard PIPE.
- Receive and transmit path functionality is decoupled except where communication between the two is required (such as Flow Control and other low-level Link management functions).

CXPL contains six modules, three for transmission and three for reception, as shown in the figure below.

- RTLH: Receive Transaction Layer Handler
- XTLH: Transmit Transaction Layer Handler
- RDLH: Receive Data Link Layer Handler
- XDLH: Transmit Data Link Layer Handler
- RMLH: Receive MAC Layer Handler
- XMLH: Transmit MAC Layer Handler

CXPL is compliant with the PCI Express 3.0 Specification with regards to the physical layer, data link layer and transaction layer.

1. PCS soft logic and timing model for mixed signal PMA

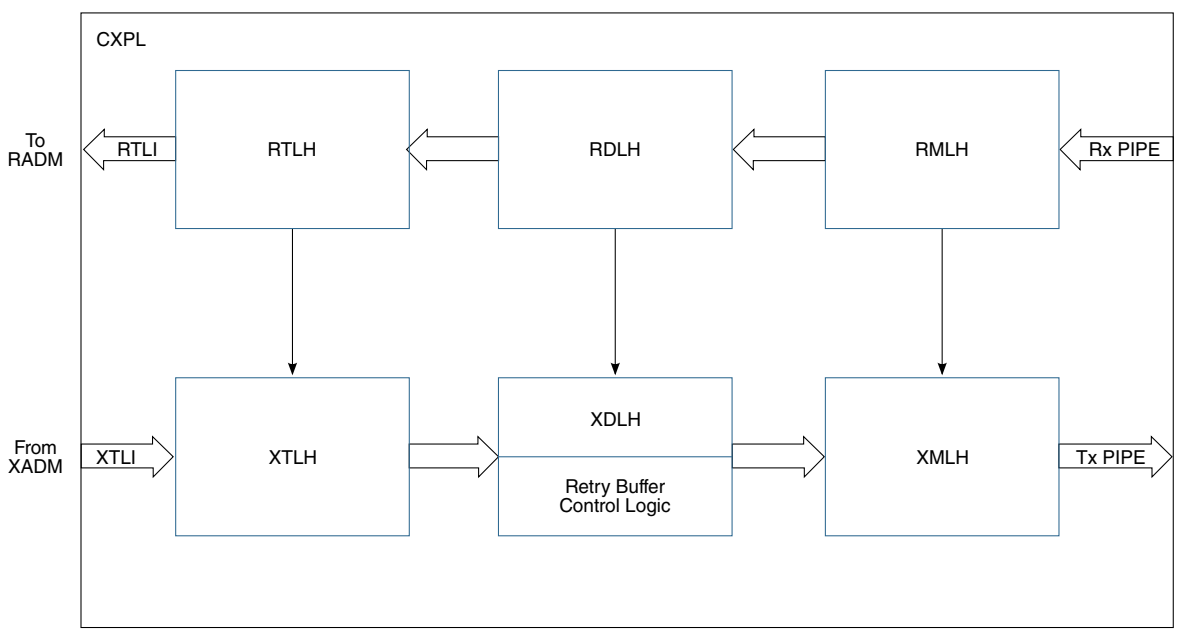


Figure 49-2. CXPL Module Block Diagram

49.2.2 Transmit Application-Dependent Module (XADM)

The XADM sits between the application logic and the CXPL core and implements the mode-specific functionality of the PCI Express Transaction Layer for packet transmission.

The figure below is a block diagram of the XADM. Its functions include arbitration, TLP formation, and credit checking.

The transmit path uses a cut-through architecture. It does not implement transmit buffering/queues (other than the retry buffer). Depending on system design, an externally-implemented transmit queue can be used to handle rate matching if the CXPL and application transfer rates are different. For relevant information on this in the context of using the AHB/ AXI Bridge as an application to the XADM.

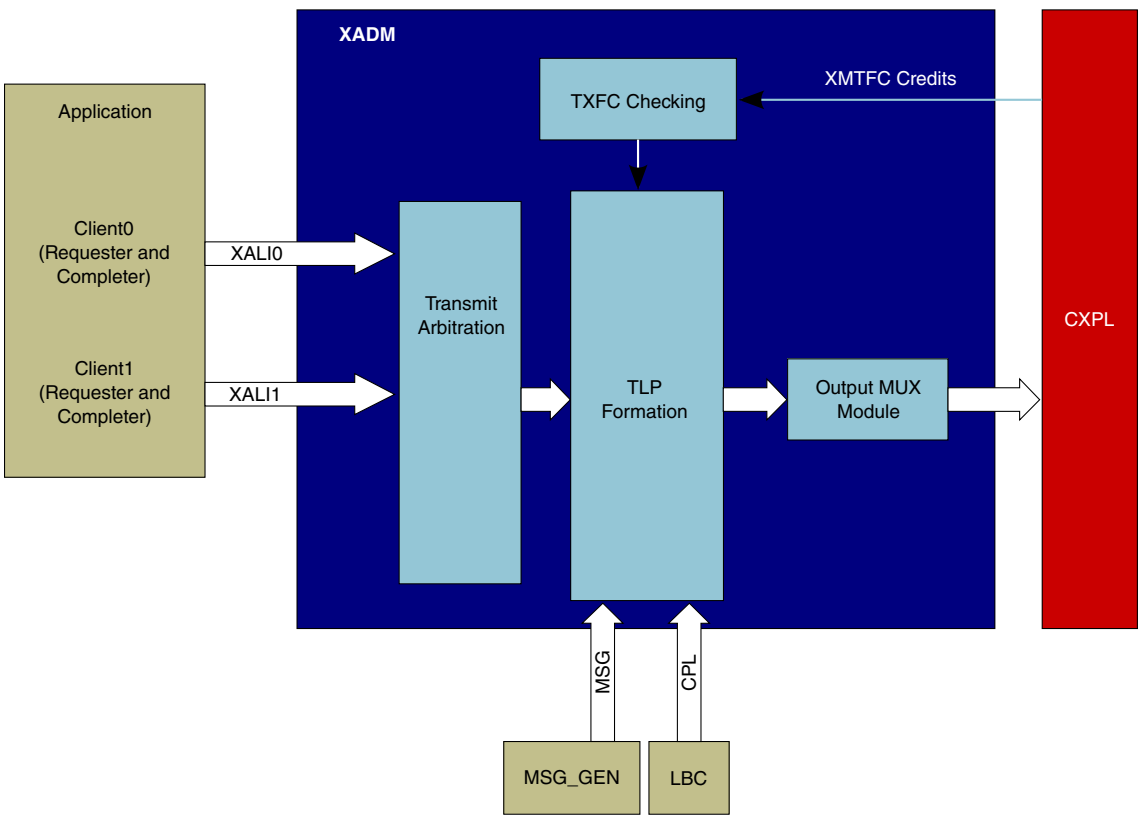


Figure 49-3. XADM Block Diagram

49.2.2.1 Arbitration

XADM provides the arbitration of TLP transmission between the following:

- The transmit client interfaces (XALI0, XALI1)
- Internally generated Messages from the MSG_GEN, triggered by PME, INTx (EP mode), errors, or application logic
- Internally generated Completions:
- EP mode: Internally generated Completions are responses for type 0 Configuration Read and Write Requests from upstream components, memory or I/O-mapped application register space Read and Write Requests, or responses to error conditions (Unsupported Requests).
- RC mode: Internally generated Completions are Unsupported Request or Completer Abort, as required by the incoming Request filtering function of the RADM.

In general, all internally generated TLP requests have higher priority than client interfaces.

For details about how the arbitration methods work and how to configure the arbitration methods, see Transmit TLP Arbitration.

49.2.2.2 Credit Checking

The core checks that enough FC credits are available in the remote device for the specific type of transaction (P, NP, CPL) before allowing a transmission of a TLP. TLPs that passed the credit check are arbitrated according to the supported arbitration method. Internally generated Completions and Messages are also gated by the arbitration logic, though at highest priority, and must also pass the FC credit test before they are accepted for transmission.

49.2.3 Receive Application-Dependent Module (RADM)

The RADM sits between the application logic and the CXPL core and implements the mode-specific functionality of the PCI Express Transaction Layer for TLP packet reception.

The figure below shows a block diagram of the RADM. The RADM serves four major functionalities as following:

- Sort/Filter received TLPs
- Completion Lookup Table (CLT), which is used for Completion tracking and Completion timeout monitoring of transmitted Non-Posted requests.
- Provide queuing (or bypass) of the received TLP
- Output received TLP to the core's receive interface (Demux function)

The filtering rules and routing for all TLP receive options are configurable for all TLPs received.

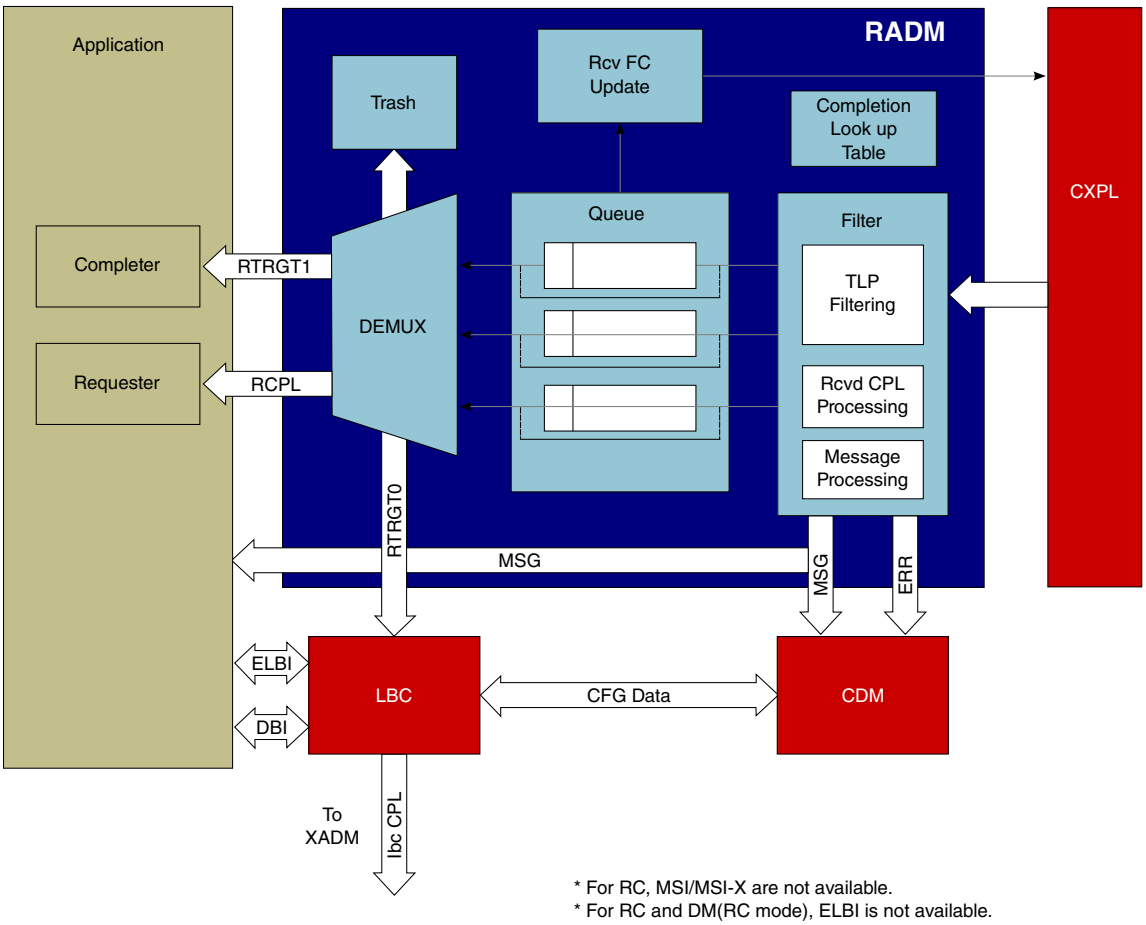


Figure 49-4. RADM Block Diagram

49.2.3.1 Posted and Non-Posted Request and Completion TLP Processing

The RADM filter passes the Posted and Non-Posted Request and Completion transactions (such as Write Transactions and Memory Reads) directly to the application through the RTRGT1 interface or to RTRGT0 for internal modules, as determined by the filtering and routing rules for the current operating mode, as described in Receive Filtering.

The RADM filter segregates Posted and Non-Posted TLPs into valid supported and valid un-supported Requests, and forwards them to the queue. The filter processes each Request and determines each TLP's destination along with other controls that may be needed to generate TLPs.

For Requests that the core forwards to the RTRGT1 or Bypass interface, the application must process the Request and generate the Completion.

For Requests that the core forwards to RTRGT0, the core automatically generates the Completion. The core automatically executes any required ELBI access before generating the Completion.

49.2.3.1.1 TLP Routing

The RADM demux is designed to mux out a received TLP to the RTRGT1 and RCPL/RBYP interfaces from single queue or multiple queue (DM/RC/EP) configurations. The filter determines the destination and the action for each TLP, then sends this to the queue. The demux decides whether to discard or forward the TLP onto the RTRGT1, RTRGT0, RCPL or RBYP interfaces.

For more details see Receive Routing and Queue to Port Mapping.

49.2.3.2 Received Completion TLP Processing

Received Completions are filtered against the completion lookup table content before presenting the Completion to the queue.

The RADM also implements a Completion time-out mechanism (via the Completion Lookup Table) and notifies the application when an expected Completion-corresponding to a transmitted Non Posted TLP-does not arrive within a specified time.

The Completion Lookup Table (and Completion Timeout event) should not be confused with the Target Completion Lookup table (and Target Completion Timeout event) .

The Target Completion Lookup Table is watching for received application completions (on XALI0/1) corresponding to previously received Non Posted requests.

The Completion Lookup Table is watching for received PCIe completions corresponding to previously transmitted Non Posted requests.

Typically, infinite Completion credits are advertised and the received completion is configured in bypass mode which means that there is no queue in the core to store completions.

NOTE

It is fully expected that the application will have enough buffering space ready for its requests, so no backpressure mechanism is needed. By default, the Completion queue operates in bypass mode.

Completions can be configured in store and forward mode if the application has chosen to do so. If a completion lookup has failed or other completion filtering has failed, the core will assert an abort signal at the end of the transaction. If the core is configured to have Completions in bypass mode, it is the application's responsibility to roll back any actions at the application's queue when an abort signal is asserted. If the core is configured with Completions enqueued, the Completion will be discarded by the core and flow control credits will be updated, as necessary, when an abort signal is detected.

49.2.3.3 Message Processing

The RADM filter provides a Message interface (grouped as part of the SII) to handle the Message TLPs received from the upstream component. The RADM filter processes the Message and decodes the header before sending it to the application logic on the SII. You can also select a configuration option to send the entire Message TLP to the application in addition to providing the decoded Message on the SII. For more details see Message Reception.

49.2.4 Configuration-Dependent Module (CDM)

The CDM implements the standard PCI Express configuration space and the core-specific register space. The CDM also requests the Message generation module to send Messages, as required, including MSI and interrupts.

The specific PCI Express configuration structures implemented in the CDM include the following:

PCI-Compatible Configuration Registers

- RC mode: Type 1 header
- EP mode: Type 0 header

PCI Capability Structures:

- PCI Power Management Capability Structure
- MSI Capability Structure

- MSI-X Capability Structure
- VPD (Vital Product Data) Capability

PCI Express Capability Structure

PCI Express Extended Capabilities:

- Advanced Error Reporting Capability
- Virtual Channel Capability
- Device Serial Number Capability
- Power Budgeting Extended Capability

The configured device type (determined by the device_type[3:0] input signal) affects the behavior of the Message generation engine, error reporting mechanism, as well as some PCI Express configuration space registers.

The CDM communicates with application's host bus controller through the DBI. The host bus controller controls accesses to registers within each CDM in multiple instances of the core in a multi-port design.

49.2.5 Local Bus Controller (LBC) and Data bus Interface (DBI)

This following topics are covered in this section:

- Overview (LBC)
- ELBI
- CDM Register Space Layout
- PCI Configuration Header and Capability Registers (in CDM)
- Port Logic (PL) Registers (in CDM)
- PCIe Wire Access (EP mode)
- PCIe Wire Access (RC mode)
- DBI Access
- LBC/DBI Feature Availability
- LBC/DBI Size Limitations
- AXI DBI Limitations

49.2.5.1 Overview (LBC)

The LBC module provides a mechanism for a link partner PCIe device (in EP mode only) or a local CPU (through the DBI) to access internal registers (in the CDM) .

NOTE

In RC mode:

The application can access CDM registers through the DBI.

PCIe wire access (through RTRGT0) to the CDM registers is NOT possible.

Figure 49-5 shows the location of the LBC within the PCIe core and its role in routing transactions.

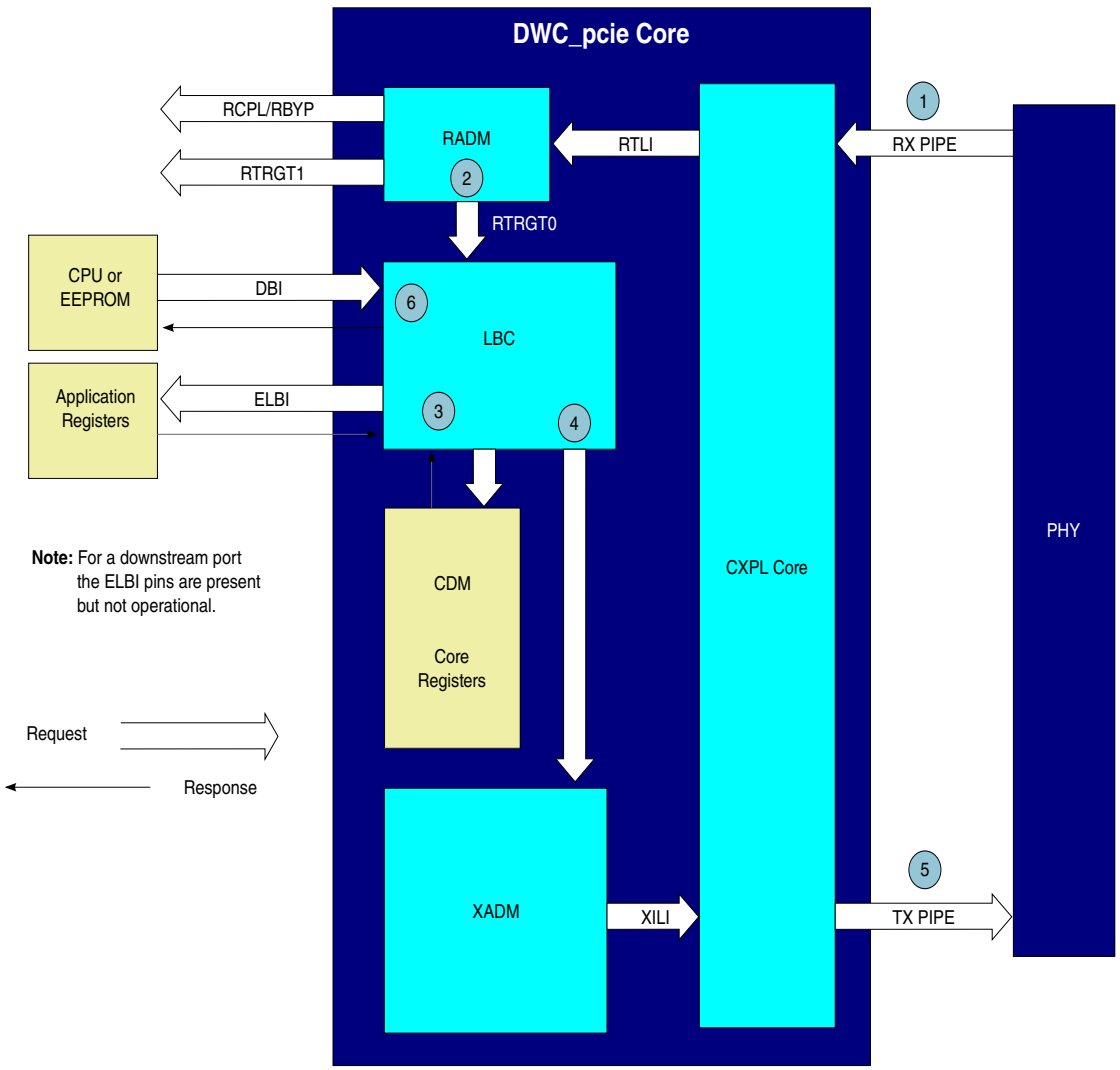


Figure 49-5. LBC Context

The LBC provides a Switched access function to internal registers (in the CDM)from the local application processor (CPU) via the DBI or the remote application software (off the PCIe RX wire) via RTRGT0. Figure 49-6 illustrates the four possible request paths through the LBC.

For more information on the filtering and routing of received inbound TLPs, see Receive Routing.

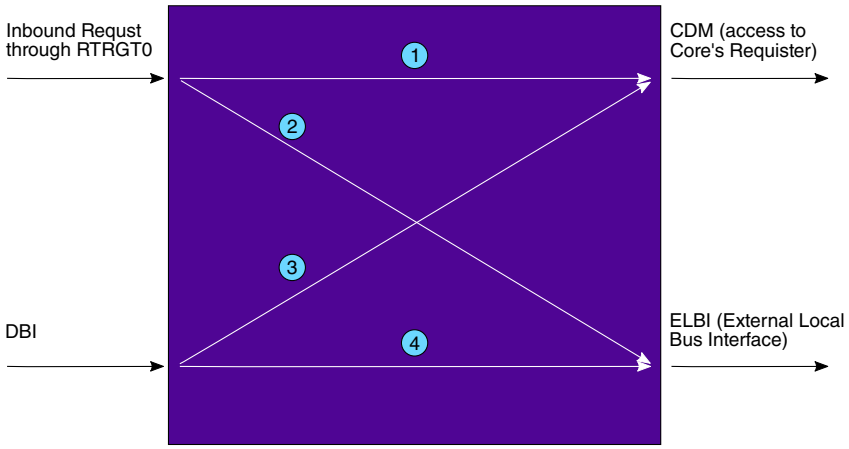


Figure 49-6. LBC Switch

NOTE

In RC mode PCIe wire access (through RTRGT0) to the CDM registers is NOT possible.

49.2.5.1.1 Simultaneous Transactions

The LBC is single-threaded and therefore, the DBI and RTRGT0 cannot use the LBC at the same time. For example, a request on the DBI will not be accepted, during a RTRGT0 transaction, until both parts of that transaction - [1] request and [2] response (completion generation) - are completed.

If the DBI and RTRGT0 present a request at the same time (regardless of the target/ destination of each request), then the LBC will grant access to the RTRGT0.

49.2.5.2 ELBI

You can connect external application registers to the ELBI. These can be accessed by PCIe request TLPs over the PCIe link or by the DBI.

NOTE

In RC mode PCIe wire access (through RTRGT0) to the ELBI is NOT possible.

49.2.5.3 CDM Register Space Layout

The core has 4096 bytes of PCI Express configuration space per function distributed as per the figure below. This address space is fully accessible from the DBI without any restrictions.

In EP mode it can be accessed from the PCIe wire using CFG requests. Under certain configurations, the Port Logic and ELBI Register spaces can also be accessed from the PCIe wire with MEM and IO requests.

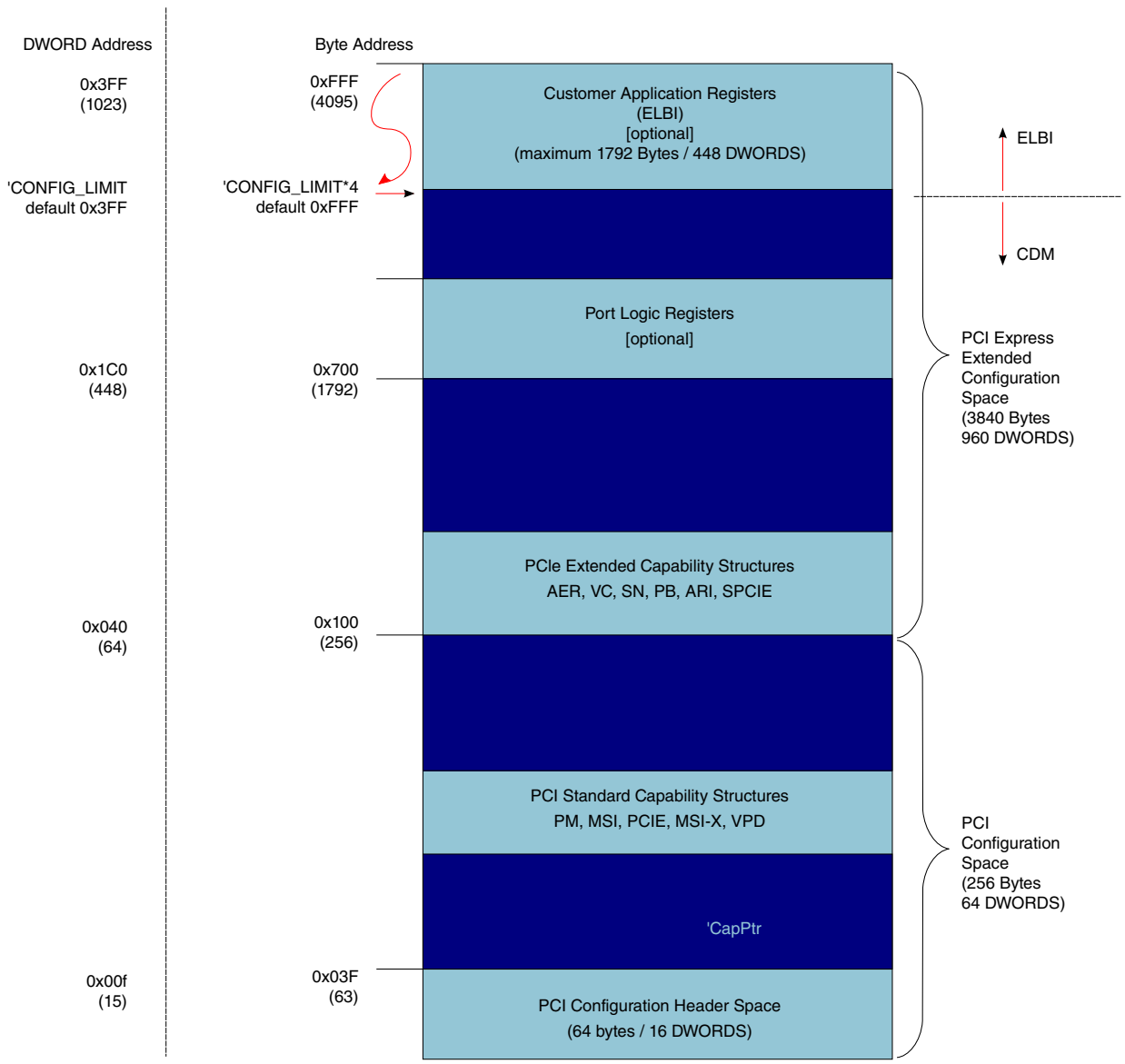


Figure 49-7. PCIe Core Configuration Space Address Map (per function)

A CFG TLP has a 6-bit Register Number Field and a 4-bit Extended Register Number field allowing 1024 DWORDS (4096 bytes) to be accessed.

49.2.5.3.1 PCI Configuration Header and Capability Registers (in CDM)

The PCI Configuration Header and Capability Registers in Figure 49-7 are PCIe core configuration registers specified by the PCI Express 3.0 Specification. Access from the PCIe wire is possible with CFG requests (in EP mode only).

These registers are fully accessible from the DBI without any restrictions.

NOTE

From the PCIe wire (through RTRGT0) in EP mode only:

You can Memory-Map the Port Logic (PL) Register Space.

You cannot Memory-Map the PCI and PCIe Configuration Register Spaces. You must always access them with a CFG request.

NOTE

In RC mode:

PCIe wire access (through RTRGT0) to the CDM registers is NOT possible.

NOTE

From the DBI:

You can access without any restriction the Port Logic (PL) Register Space.

You can access without any restriction the PCI and PCIe Configuration Register Spaces.

49.2.5.3.2 Port Logic (PL) Registers (in CDM)

The Port Logic Registers in Figure 49-7 are PCIe core configuration registers not specified by the PCI Express 3.0 Specification, but are specific to the configuration and operation of the PCIe IP core.

In EP mode, access from the PCIe wire is with CFG requests. There is no access from the PCIe wire in RC mode.These registers are fully accessible from the DBI without any restrictions.

49.2.5.3.3 Memory Mapping PL Registers

Port Logic Registers (which by default are accessed by CFG requests) can also (at the same time) be accessed by MEM requests through the use of the ENABLE_MEM_MAP_PL_REG, PL_FUNC_NUM and PL_BAR_NUM configuration parameters. These can be used to map the Port Logic Registers to any BAR of any function.

All MEM requests that match PL_BAR_NUM (when ENABLE_MEM_MAP_PL_REG=1) and whose address offset is in the range 0x700-0x8FF will be routed to the Port Logic Registers.

NOTE

The BAR corresponding to PL_BAR_NUM must also be mapped/assigned to RTRGT0.

49.2.5.4 PCIe Wire Access (EP mode)

By default[1], CFG requests are routed to RTRGT0 and then to CDM via LBC.

By default, BAR-matched MEM/IO requests are routed to RTRGT1.

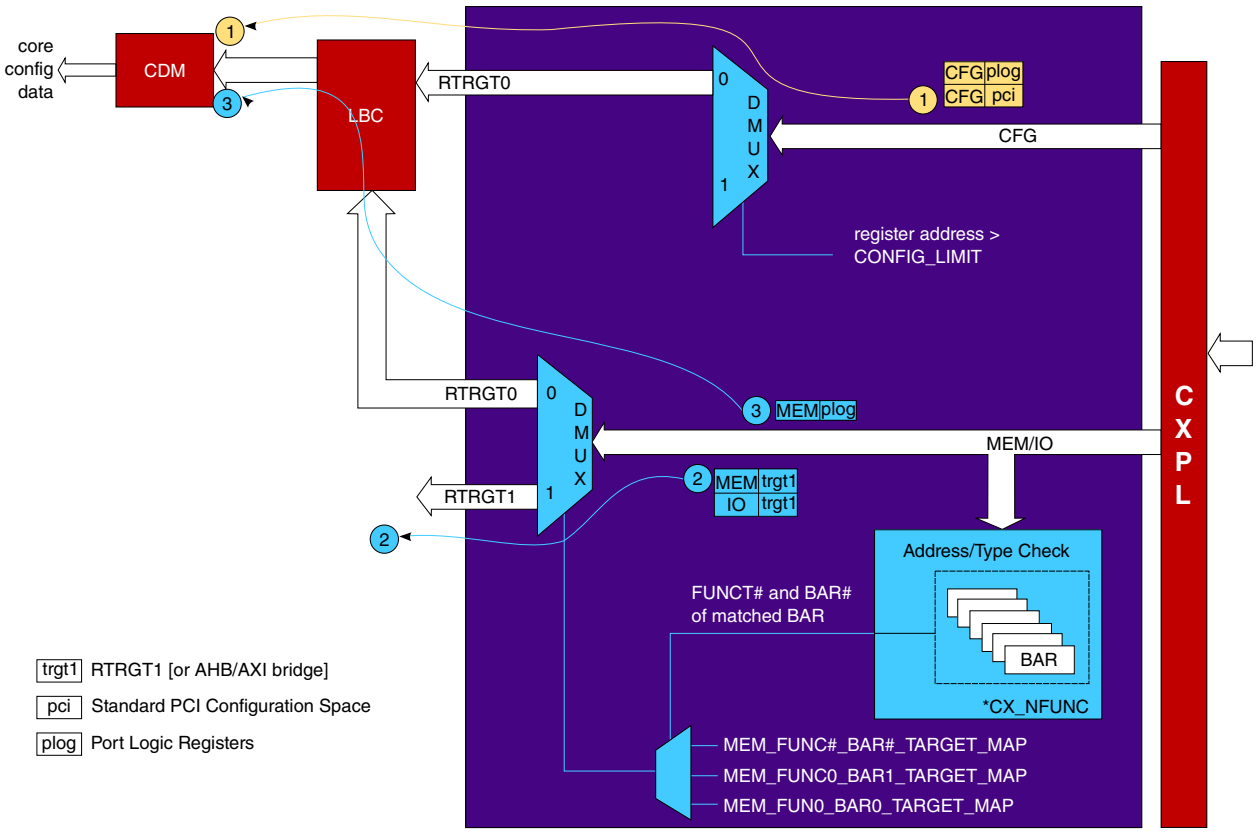


Figure 49-8. Request TLP Routing - Typical Use Model

Looking in detail at the typical routing of inbound PCIe requests (the numbers refer to the paths identified by the circled numbers in the above diagram.

CFG requests-either to Standard PCI Configuration Space or Port Logic Registers-are always routed to the CDM. This is because CONFIG_LIMIT by default is set to 0x3FF (top of the CFG register address space).

BAR-matched MEM/IO requests are routed to RTRGT1. MEM_FUNC#_BAR#_TARGET_MAP for each enabled BAR is set by default to RTRGT1.

Port Logic Registers (which by default are accessed by CFG requests-see [1] above) can also be accessed by MEM requests through the use of the ENABLE_MEM_MAP_PL_REG, PL_FUNC_NUM and PL_BAR_NUM configuration parameters. These can be used to map the Port Logic Registers to any BAR.

To see all routing possibilities, refer to Receive Routing.

49.2.5.5 PCIe Wire Access (RC mode)

To see all routing possibilities, see Receive Routing.

NOTE

In RC mode the application can access CDM registers through the DBI.

PCIe wire access (through RTRGT0) to the CDM registers is NOT possible.

49.3 Core Operations

This section describes the operations of the PCI Express core.

49.3.1 Initialization

Immediately after reset the DM core goes into either EP mode or RC mode depending on the state of the device_type input.

The internal configuration registers in the CDM assume their default reset values as listed in the following sections:

- PCIe Registers (EP mode)
- PCIe Registers (RC mode)
- PCIe Registers: Port Logic

The application must keep the app_ltssm_enable signal deasserted after reset until the application is ready to establish a Link and start receiving and transmitting TLPs. If the application needs to update configuration registers in the CDM as part of the initialization process, then the application must keep app_ltssm_enable deasserted until it has programmed all the necessary configuration registers through the DBI.

After initializing the necessary configuration registers, the application can assert app_ltssm_enable to allow the LTSSM to begin Link establishment. The LTSSM begins Link negotiation after the core_rst_n and phy_mac_phystatus signals are deasserted and app_ltssm_enable is asserted.

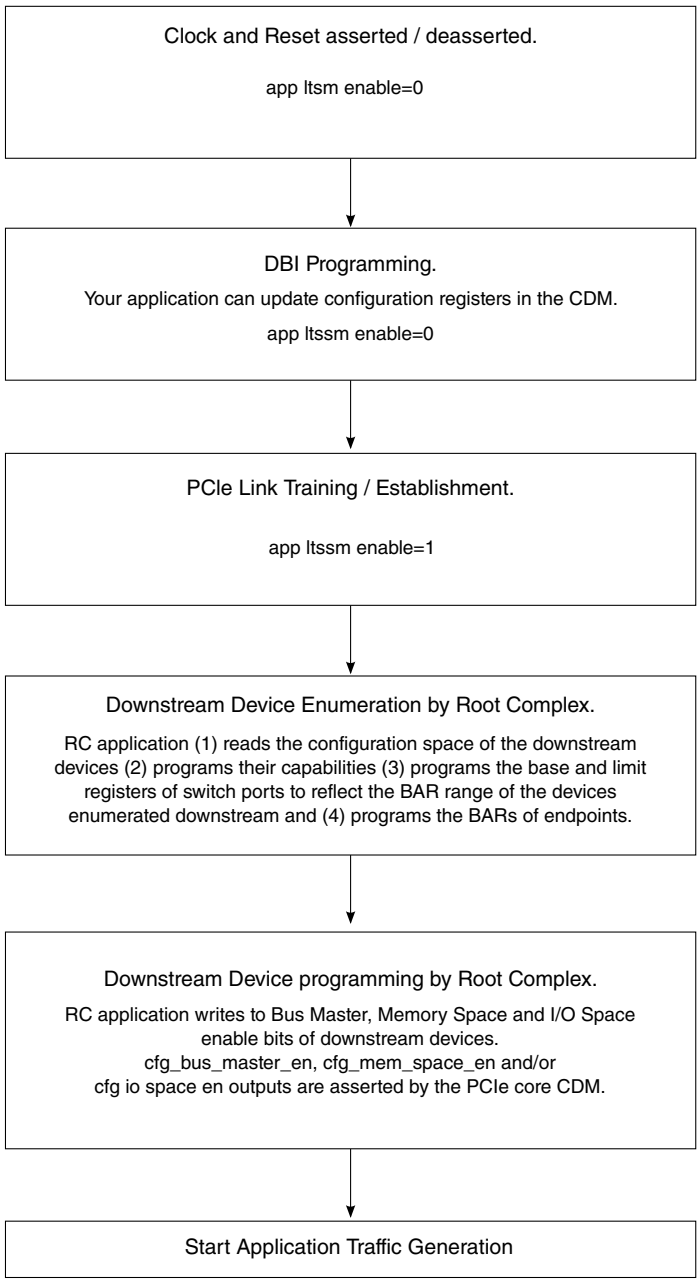


Figure 49-9. PCI Express Initialization Steps

49.3.2 Link Establishment

The core and a PCI Express compliant PHY combine to provide a complete solution for setting up and maintaining a compliant PCI Express Link.

The core implements the LTSSM function according to the *PCI Express 3.0 specification*.

In general, the process for establishing a Link is a follows:

1. Upon power-up (or directly out of reset), it is assumed that the power supply becomes stable and the PLLs reach frequency lock before the devices attempt to establish a valid Link. Once in a valid state, the SerDes either communicates a ready status to the core or simply begins transmitting and receiving valid data.
2. Per the *PCI Express 3.0 specification*, once bit and symbol synchronization are complete, the core initiates the following sequence to establish a Link (assuming a valid and properly functioning Link partner):

Receiver detection on available Lanes for the Port.

Exchange of Training Sequences to determine Link configuration (for example, Link speed, number of Lanes, and order).

Once both partners reach a valid negotiated state, the Link state is set up and the LTSSM is in L0.

1. Once Link up is achieved, the data link modules take over to manage the Link and initialize Flow Control.
2. After Flow Control initialization is complete, the data link modules signal the transaction layer modules that the link is ready to allow transmission/reception of TLP traffic.
3. During normal operation, the LTSSM and data link modules continue to manage the underlying Link integrity while data traffic is communicated across the PCI Express Link.

NOTE

The power management implementation also affects Link establishment.

49.3.3 Transmit TLP Processing

Information found here describes the flow of transmit TLPs through the core.

The topics for this section are:

- Transmit Overview
- Transmit TLP Arbitration
- Transmit Retry
- Transmit DLLP Priorities

It may helpful to first review the following sections: Common Xpress Port Logic (CXPL) and Transmit Application-Dependent Module (XADM).

49.3.3.1 Transmit Overview

Generally, all types of transmit TLPs (Posted, Non-Posted, and Completion) generated by the application travel through the core in the following flow:

1. The application presents a transaction transmission request with header information and payload (if applicable) on one of the transmit client interfaces (for example, XALI0).
2. The XADM forms the transaction into a TLP and checks the TLP against the current Flow Control credit availability. If the TLP passes the Flow Control checks and wins the arbitration with TLPs from the other the client interfaces, then the TLP goes to the CXPL.
3. The XTLH module inserts an ECRC (if applicable) and snoops/stores the necessary TLP information for Completion lookup (for Non-Posted requests only).
4. The XDLH inserts the Sequence Number and LCRC into the TLP and the retry buffer stores the TLP.
5. The XMLH inserts start and end delimiters and performs data scrambling.
6. The XMLH presents the packet to the PHY through the PIPE interface.
7. The PHY receives the packet, performs 8b10b encoding, and serialization, then sends the packet for transmission on the Link

The core does not check for TLP errors; instead it sends the TLP as presented on the XALI interface.

The native PCIe core does not check that the TLP payload size is less than the 'Maximum Payload Size' limit. However, the AXI bridge module does guarantee that this limit is not exceeded.

The native PCIe core does not check that the TLP payload size is less than CX_MAX_MTU limit. Exceeding this limit will overflow the retry buffer, resulting in data corruption. However, the AXI/AHB bridge module does guarantee that this limit is not exceeded.

49.3.3.2 Transmit TLP Arbitration

The transmit arbitration mechanisms supported by the core are as follows:

- Client-based round robin arbitration
- Strict priority client-based arbitration
- VC-based priority arbitration

To configure the transmit arbitration algorithm, use the 'Transmit Arbitration Method' configuration parameter (`CX_XADM_ARB_MODE).

Regardless of the transmit arbitration method selected, Messages (both internally-generated and Messages requested through the VMI) always have the highest priority, followed by internally-generated Completions. The priority order for all transmitted TLPs is:

1. Internally generated Messages
2. Internally-generated Completions
3. Transmit TLPs from Client0, Client1, and Client2 according to the selected arbitration method

49.3.3.2.1 Client-Based Arbitration

When you configure the core to use client-based arbitration (`CX_XADM_ARB_MODE = 1), the XADM uses round-robin arbitration between the two transmit client interfaces.

49.3.3.3 Transmit Retry

There is a Retry Buffer (RB) in the core that stores a copy of each transmitted TLP until an Ack is received. The RB consists of two buffers: retry buffer and start-of-TLP (SOT) buffer.

NOTE

The Retry Buffer does not function as a transmit queue. The core transmits TLPs immediately after they pass arbitration. The copy in the Retry Buffer is only sent in the event that the TLP must be re- transmitted.

The retry buffer is implemented with a single port RAM. The depth of the retry buffer is selected during hardware configuration either by enabling automatic buffer sizing or by setting the depth explicitly. The retry buffer width is set automatically (data bus width plus extra control bits).

The selected retry buffer size determines the size of the SOT buffer. The SOT buffer stores the starting address of each unacknowledged TLP stored in the retry buffer. The SOT buffer is implemented with a single port RAM and is indexed by the Sequence Number of the TLP whose starting address is being stored or retrieved.

The minimum depth of the SOT buffer is also a user configuration option. The selected size must allow the retry buffer to store the maximum number of shortest TLPs (3 DWORDs).

When a Nak is received or the replay timer times out, a replay is initiated. A replay is terminated by two conditions:

- When the replay of all TLPs in the retry buffer is finished, or
- An Ack DLLP is received that acknowledges all TLPs in the retry buffer

The replay timer tracks the TLP replay time. It stays at 0 when every TLP has received an Ack and starts to count when a TLP is transmitted and the LTSSM is not in the training state. The replay timer is reset to 0 when an Ack or Nak is received that acknowledges a TLP that is in the retry buffer.

49.3.3.4 Transmit DLLP Priorities

The order of priority to transmit pending DLLPs is:

1. High-priority DLLPs
2. TLPs
3. Low-priority DLLPs

49.3.4 Receive TLP Processing

The information found here describes the flow of receive TLPs through the core.

49.3.4.1 Receive Overview

It may helpful to first review the following sections:Common Xpress Port Logic (CXPL) and Receive Application-Dependent Module (RADM).

Generally, received transactions travel through the core in the following flow:

1. The PHY receives a stream of bits and aligns/forms them into 10-bit symbols (Gen1/2)
2. The PHY decodes the 10b stream into an 8b stream (Gen1/2)
3. The PHY crosses the clock domain from RX to TX and presents the stream to the PIPE.
4. The RMLH descrambles and deskews the incoming data, checks for receiver (Gen1/2) then extracts packets.
5. The RDLH strips off the LCRC and Sequence Number.
6. The RTLH strips off the ECRC (if applicable), checks for a malformed TLP, and forms a transaction across the RTLI interface to the RADM.

7. The RADM filters the transaction based on the transaction type (Posted, Non-Posted, or Completion) and the rules described in Receive Filtering.
8. Filtered transactions are sent to RADM queues.
9. Transactions residing in the RADM queues are presented to the application or locally handled by the LBC module, depending upon the filter result

49.3.4.2 Receive Filtering

The core contains a filter module that is responsible for the tasks listed here.

- Determine the status of a received TLP using filtering rules.
- Determine the destination interfaces of a received TLP based on the status from applying the filter rules.
- Signal the application for the status of the received TLP by driving signals such as DLLP abort, TLP abort and ECRC error.

The core filters and routes received TLPs according to a set of rules determined by the TLP type based on the *PCI Express Base 3.0 Specification* and user-configurable filtering options. The filtering rules for a received TLP are affected by I/O signals (run-time options), and register values (run-time options).

The application can mask some of the filtering and error handling rules by setting the corresponding bits in Symbol Timer Register and Filter Mask Register 1 and Filter Mask Register 2.

There are three types of the filtering rules in the core:

- 3.4.2.1: rules that are applicable for all TLP received
- 3.4.2.2: rules that are dependent on the type of the TLP based on PCIe specification
- 3.4.2.3: rules that are not from the PCIe specification but requested by specific applications.

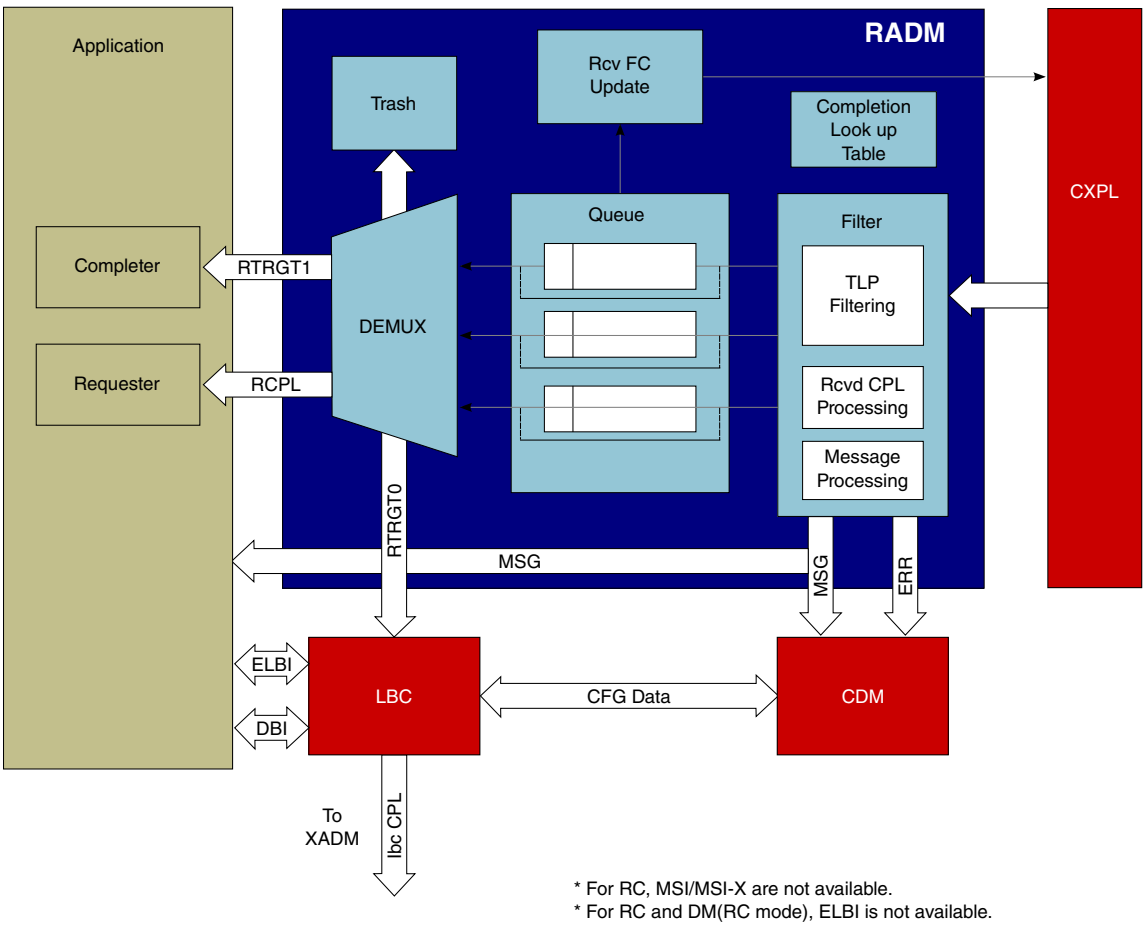


Figure 49-10. RADM Block Diagram

49.3.4.2.1 Filtering Rules Applicable for all TLPs Received

The general rules listed here apply to all incoming TLPs.

- The core discards all incoming TLPs that have an invalid Type field. This TLP is treated as a 'TLP ABORT'.
- By default, a request TLP with the poison bit set (EP=1) is considered an Unsupported Request (UR) and discarded, as the poison rule mask bit (CX_FLT_MASK_UR_POIS) is not set in the Symbol Timer Register and Filter Mask Register 1. You can control the end result of a poisoned TLP by setting this mask bit, and having the poisoned TLP sent to your application.
- A locally terminated TLP with ECRC error detected is discarded in store-and-forward mode and an ECRC error reported only when the filter mask CX_FLT_MASK_ECRC_DISCARD bit is not set. For more information see ECRC Handling.
- Filter rules have no effect on received TLPs when 'DLLP ABORT' signal is asserted.

- If a completion of a non-posted request is not received within a completion timeout period, this request will be treated as a completion timeout, and a non-advisory error will be reported.
- For messages to be accepted and decoded, the incoming Message must be one of the valid Message types with the correct payload length based on *PCIe 3.0 Specification*. Valid Messages will be decoded and passed onto the SII interface as necessary.

See Error Handling for more details.

The filtering rules for an incoming Request TLP are affected by the configuration parameters (compile-time options), I/O signals (run-time options), and register values (run-time options).

NOTE

In many cases, the standard filtering rules may be 'masked' or ignored by setting the corresponding bit in the Symbol Timer Register and Filter Mask Register 1 and Filter Mask Register 2, which take their default values from the DEFAULT_FILTER_MASK_1 and DEFAULT_FILTER_MASK_2 configuration parameters.

49.3.4.2.2 Filtering Rules Based on TLP Type Defined in PCIe Specification

PCIe TLPs are categorized as Requests and Completions. The table found here describes the filtering rules for Request and Completion TLPs and the results of the core's filter.

If a received TLP passes all of the filter rules for Request and Completion TLPs, then it is considered to have no errors, and the TLP will be routed to the destination that is configured. Details on routing are provided in Receive Routing.

Notation of filter results:

1. UR = Unsupported Request

   CA = Completer Abort

   CRS = Configuration Request Retry Status

   SU = Successful

   UC = Unexpected Completion

   MLF = Malformed

   '-' = Filtering rule does not apply to TLP type

   MA = Master Abort

TA = Target Abort

49.3.4.2.2.1 EP Mode Filtering Rules

Table 49-3. Result of Filtering Rules Applied to Request TLPs and Completion (CPL) TLPs: EP Mode

Filtering Rule	TLP Type					
	MRd IORd	MWr IOWr	CFG	MSG	CPL with UR/CA/CRS status	CPL with SU status
PowerState is not in D0.	UR	UR	SU	SU	UC	UC
Address is not within any configured Memory BAR or IO BAR if it is an IO request.	UR	UR	-	-	-	-
TLP header poison bit is set and the filter mask CX_FLT_MASK_UR_POIS bit is not set.	UR	UR	UR	UR	SU	SU
Address within a BAR that is configured to RTRGT0 and TLP DW length > 1.	CA	CA	-	-	-	-
MRd with lock and filter mask CX_FLT_MASK_LOCKED_RD_AS_UR bit is not set.	UR	-	-	-	-	-
The function number of a completer ID within a CFG request does not match an implemented function within the receiver device and the filter mask CX_FLT_MASK_UR_FUNC_MISMATCH bit is not set.	-	-	UR	-	-	-
Configuration type1 TLP request and the filter mask CX_FLT_MASK_CFG_TYPE1_REQ_AS_UR is not set.	-	-	UR	-	-	-
Application requests the core filter to return CRS by asserting signal app_req_retry_en.	-	-	CRS	-	-	-
Not Valid Message for EP device	-	-	-	UR/MLF	-	-
Illegal payload length of a message.	-	-	-	UR	-	-
Vendor MSG Type0 with filter mask CX_FLT_MASK_VENMSG0_DROP bit not set.	-	-	-	UR	-	-
Vendor MSG Type1 with r[2:0] to 3'b010 and {Bus#, Dev#, Func#} mis-match.	-	-	-	UR	-	-
TLP with ECRC error detected	CA	CA	CA	-	-	-
Requester ID mis-match	-	-	-	-	MA/TA	MLF
Requester TAG mis-match	-	-	-	-	MA/TA	MLF
TAG error (non-pad zero for reserved TAG bits	-	-	-	-	MA/TA	MLF
Byte Count Mismatch (PCIe Gen2)	-	-	-	-	MA/TA	UC/MLF

Table 49-4. Result of Filtering Rules Applied to Request TLPs and Completion (CPL) TLPs: EP Mode

Filtering Rule	TLP Type					
	MRd IORd	MWr IOWr	CFG	MSG	CPL with UR/CA/CRS status	CPL with SU status
Completion received with status of UR.	-	-	-	-	MA	-
Completion received with status of CA.	-	-	-	-	TA	-
Completion received with status of CRS	-	-	-	-	CRS	-
Completion received with CRS status and Completion is not a pending configuration request	-	-	-	-	MLF	-

A complete list of the filtering checks can be referenced at Symbol Timer Register and Filter Mask Register 1 (PCIE_PL_STRFM1) and Filter Mask Register 2 (PCIE_PL_STRFM2).

49.3.4.2.2.2 RC Mode Filtering Rules

Table 49-5. Result of Filtering Rules Applied to Request TLPs and Completion (CPL) TLPs: RC Mode

Filtering Rule	TLP Type							
	MRd	MWr	CFG[1]	IO	MSG	CPL with UR/CA status	CPL with CRS status	CPL with SU status
Address does not satisfy any of the following conditions: 1. Within any configured Memory BAR. 2. Outside of the memory range AND prefetchable memory range as determined by the corresponding Base and Limit fields in the Type-1 header. 3. The filter mask CX_FLT_MASK_UR_OUTSIDE_BAR bit is set, which treats out-of-bar TLPs as Supported Requests and indicates a special application requirement	UR	UR	-	-	-	-	-	-
Native Core (no AHB/AXI bridge): FLT_Q_ADDR_WIDTH < 64, and any upper address bit (above bit position FLT_Q_ADDR_WIDTH-1) is set to '1'	UR	UR	-	UR	-	-	-	-
With AHB/AXI bridge: MASTER_BUS_ADDR_WIDTH = 32, FLT_Q_ADDR_WIDTH > 32,	UR	UR	-	UR	-	-	-	-

Table continues on the next page...

Table 49-5. Result of Filtering Rules Applied to Request TLPs and Completion (CPL) TLPs: RC Mode (continued)

and any upper address bit (above bit position MASTER_BUS_ADDR_WIDTH-1) is set to '1'								
TLP header poison bit is set and the filter mask CX_FLT_MASK_UR_POIS bit is not set.	UR	UR	UR	UR	UR	-	-	-
MRdLk request received and filter mask CX_FLT_MASK_LOCKED_RD_AS_UR bit is set, which indicates that customer prefer to filter out the MRdLk.	UR	-	-	-	-	-	-	-
CFG Request received and the filter mask CX_FLT_MASK_RC_CFG_DISCARD is not set	-	-	UR	-	-	-	-	-
IO Request received and the filter mask CX_FLT_MASK_RC_IO_DISCARD is not set	-	-	-	UR	-	-	-	-

Table 49-6. Result of Filtering Rules Applied to Request TLPs and Completion (CPL) TLPs: RC Mode

Filtering Rule	TLP Type							
	MRd	MWr	CFG[1]	IO	MSG	CPL with UR/CA status	CPL with CRS status	CPL with SU status
Vendor MSG Type0 with filter mask CX_FLT_MASK_VENMSG0_DROP bit not set.	-	-	-	-	UR	-	-	-
Not Valid Message for RC device	-	-	-	-	UR/MLF	-	-	-
TLP with ECRC error detected	CA	CA	CA	CA	-	-	-	-
Requester ID mis-match	-	-	-	-	-	MA/TA	-	MLF
Requester TAG mis-match	-	-	-	-	-	MA/TA	-	MLF
TAG error (non-pad zero for reserved TAG bits	-	-	-	-	-	MA/TA	-	MLF
Byte Count Mismatch	-	-	-	-	-	MA/TA	-	MLF
Completion received with status of UR.	-	-	-	-	-	MA	-	-
Completion received with status of CA.	-	-	-	-	-	TA	-	-
Completion received with CRS status and Completion is not a pending configuration request.	-	-	-	-	-	-	MLF	-

1. DM (in RC mode) should not expect to receive a CFG or IO request.

49.3.4.2.3 Filtering Rules Not Defined in PCIe Specification

There are additional filtering rules that are designed to provide enhanced filter support for certain applications.

- Core to handle the received posted or non-posted requests with zero byte length. When a zero-byte request TLP is received, also called "flush" command, the core can drop[1] the zero- byte request. This is designed to support some applications that can not handle a zero-byte request. Applications can dynamically program a bit in the filter mask CX_FLT_MASK_HANDLE_FLUSH bit to turn on/off this rule. If the core is programmed to handle the flush, it will be the completer's task to return completion status.
- Core to detect oversize read request and return UR for the read request.Some applications may have a buffer limit and are not able to handle lengthy read requests. The core over-size read request detection rule can be turned on when an application can identify a maximum read request size that it can tolerate.

49.3.4.3 Receive Routing

49.3.4.3.1 EP Mode

The possible destinations of a posted or non-posted Request TLP are RTRGT1 interface, RTRGT0 interface and Core Discard (dropped[1] or terminated). By default:

- CFG requests are routed to RTRGT0 and then to CDM via LBC.
- BAR-matched MEM/IO requests are routed to RTRGT1.
- MSG requests are decoded internally, signalled on the SII interface and then terminated.

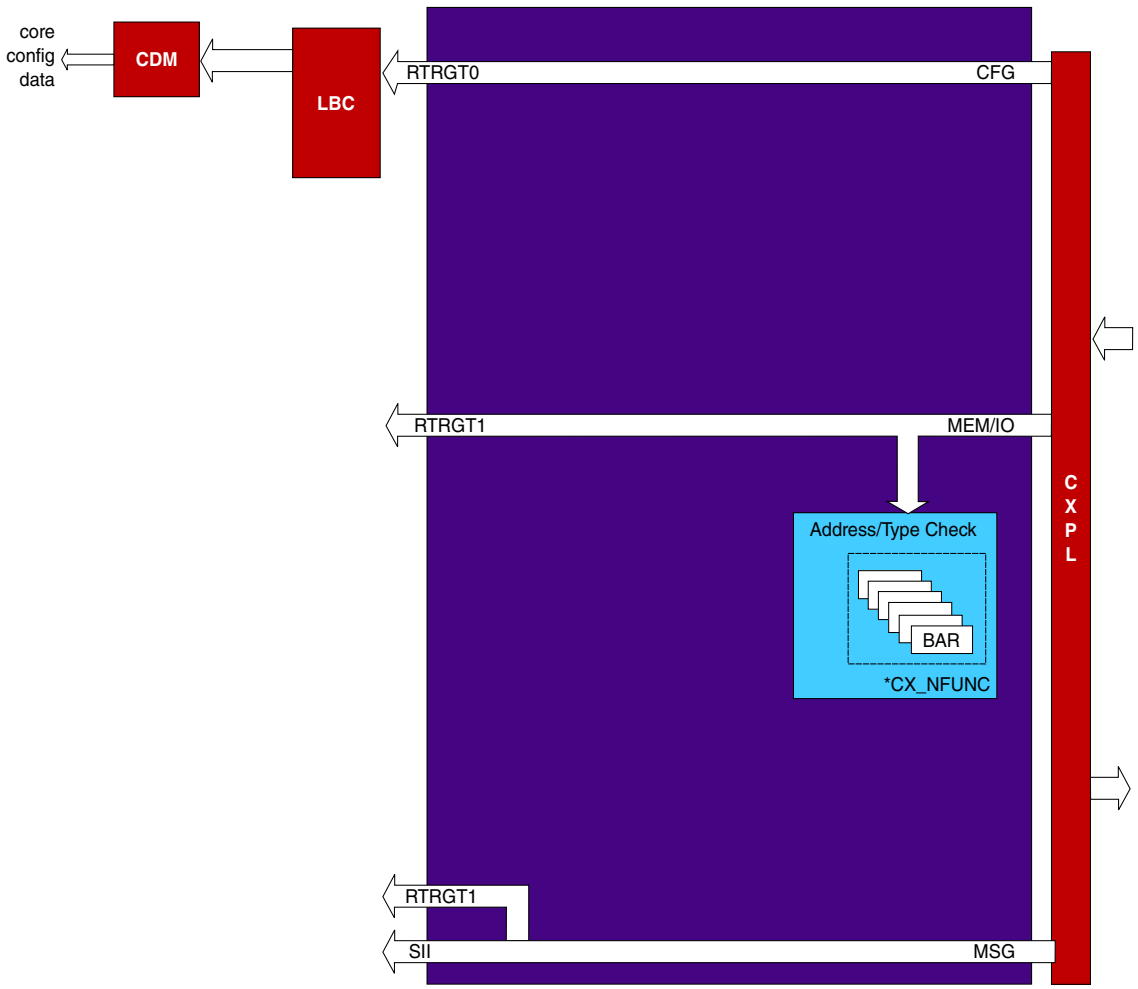


Figure 49-11. Default Request TLP Routing (assuming no TLPs with CA/CRS/UR completion status)

The possible destinations of a Completion TLP are RCPL interface, RBYP interface, RTRGT1 interface, and Core Discard.

In general, a TLP type that is configured as bypass will be sent to either the RBYP interface, or RCPL interface if it is a completion. A TLP type that is configured as a cut-through or store-forward will be sent to RTRGT1 interface.

Because the core supports three types of queue architecture (single queue, multiple queue, segmented buffer queue architecture) and three buffer modes (bypass, cut-through, store-forward mode), a configuration of the core receive queue structure will affect the destination of a received TLP.

NOTE

By default, all Configuration Requests that pass filtering go to RTRGT0 for configuration register access. However, the application can configure the core to direct certain

configuration TLPs to the RTRGT1 interface - see PCIe Wire Access (EP mode) for more details. The application is responsible for generating Completions for Configuration Requests that are routed to RTRGT1.

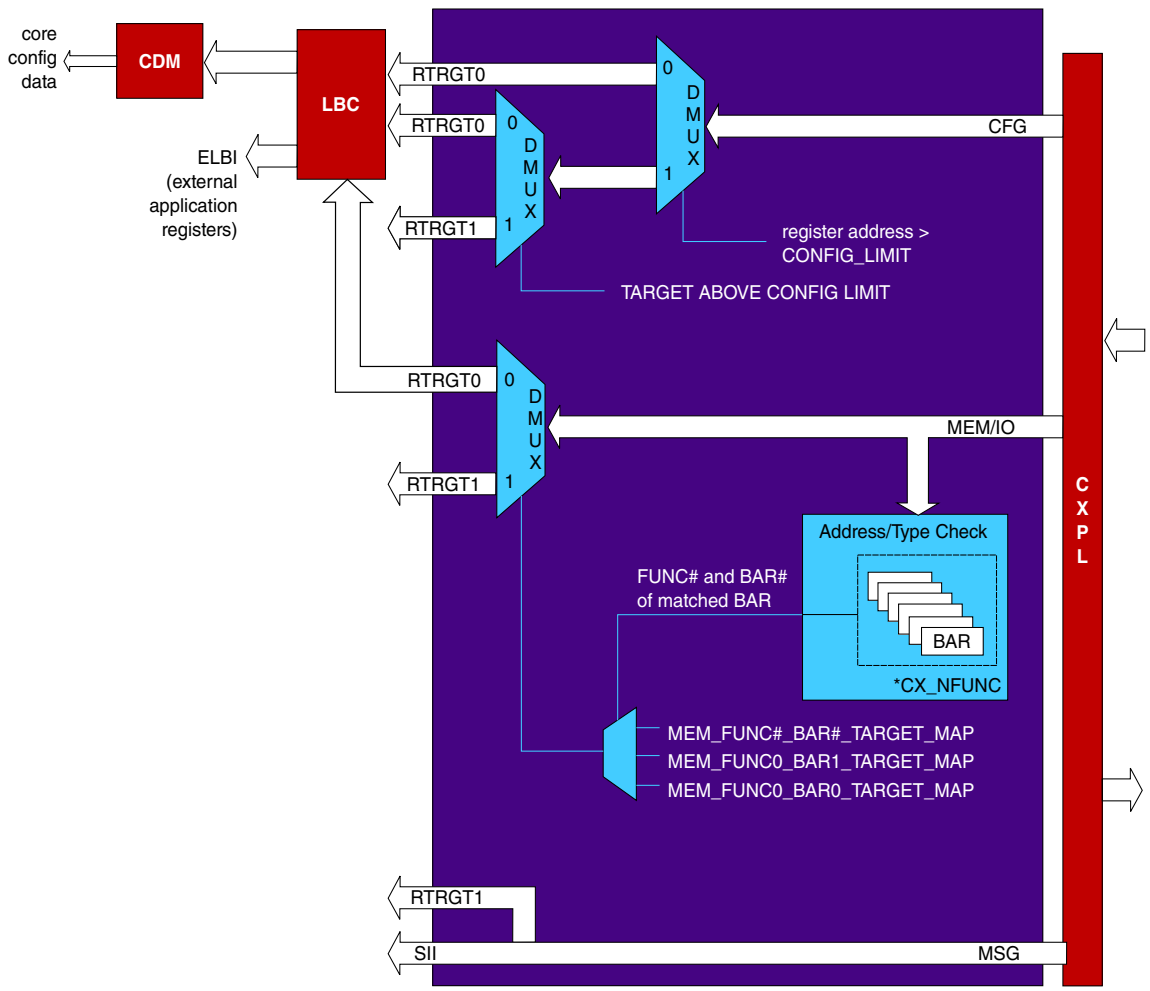


Figure 49-12. Configurable Request TLP Routing (assuming SC completion status).

49.3.4.3.2 RC mode

The possible destinations of a posted or non-posted Request TLP are RTRGT1 interface and Core Discard (dropped1 or terminated). By default:

- MEM requests outside of the memory range AND prefetchable memory range as determined by the corresponding Base and Limit fields in the Type-1 header, are routed to RTRGT1.
- MSG requests are decoded internally, signalled on the SII interface and then terminated.

- An RC does not expect to receive CFG or IO requests.
- BARs should be disabled and not used.

The possible destinations of a Completion TLP are RCPL interface, RBYP interface, RTRGT1 interface, and Core Discard.

In general, a TLP type that is configured as bypass will be sent to the RBYP interface. A TLP type that is configured as a cut-through or store-forward will be sent to RTRGT1 interface. Because the core supports three types of queue architecture (single queue, multiple queue, segmented buffer queue architecture) and three buffer modes (bypass, cut-through, store-forward mode), a configuration of the core receive queue structure will affect the destination of a received TLP.

49.3.4.3.3 ECRC Handling

The setting of the CX_FLT_MASK_ECRC_DISCARD bit (default value is 0) in the Symbol Timer Register and Filter Mask Register 1 can be used to prevent an ECRC contributing to a CA status and thereby preventing all associated downstream effects such as error handling.

By default, all incoming IO or MEM requests with UR/CA/CRS status will be dropped1/ terminated and an Advisory Non-Fatal Error is signalled. For Non Posted (NP) requests, a CPL with CA is generated. See Advisory Non-Fatal Error Messages.

If you set the DEFAULT_TARGET parameter to 'Forward' (default is 'Drop'), then all incoming IO or MEM requests with UR/CA/CRS status will not be dropped but will be forwarded to the application. Setting CX_MASK_UR_CA_4_TRGT1 (default is 0) at the same time, will suppress error reporting for TLPs (with UR/CA status) that are being routed to the application on AXI Bridge master.

A completion TLP with ECRC errors is only dropped by the RADM in the native core when CPL queue mode is store-forward and queue architecture is single or multiple queue. See Completion TLP Routing Rules for more details.

49.3.4.3.4 Request TLP Routing Rules

The next table shows the applicability of routing rules for Request TLPs, and indicates whether the destination is as stated by the rule when the conditions of the rule are met.

By default all error-free MSG requests are decoded internally, signalled on the SII interface and then terminated. To have the decoded message *also* sent to the application/ AMBA interface then see Routing of Received Messages to SII and optionally to

Application. When a MSG request is filtered with UR/CA/CRS status, the TLP is always terminated. Only MSG requests filtered with SC status, can potentially be forwarded to the application.

For a full analysis of what error conditions contribute towards an UR or CA status, see Receive Filtering. For example, an ECRC error contributes towards a CA status whereas an UR status can be generated by detecting the EP bit set in the TLP header or having an IO/MEM request not match against any of the BARs.

In many cases, the standard routing rules may be 'masked' or ignored by setting the corresponding bit in the Symbol Timer Register and Filter Mask Register 1 and îFilter Mask Register 2. For example, see Message Reception.

NOTE

RTRGT1 is the application interface and it is connected to the AXI bridge master interface.

Notation of routing results:

Yes = destination is as specified in rule when conditions of rule are met

no = destination is not as specified in rule even when conditions of rule are met

- = routing rule has no affect because it does not apply to TLP type

Table 49-7. Routing Rules for Request TLPs (EP Mode)

Routing Rule	MRd	MWr	CFG	IO	Vendor MSG Type0	Vendor MSG Type1	Other MSG
When a request is filtered with SU status, and is in BAR range, MEM_FUNC#_BAR#_TARGET_MAP parameter determines the destination.	Yes	Yes	no	Yes	-	-	-
When a request is filtered with UR/CA/CRS status, and the DEFAULT_TARGET parameter is 0, the TLP is dropped. For NP requests, a CPL is also generated.	Yes	Yes	Yes	Yes	Yes	Yes	Yes
When a request is filtered with UR/CA/CRS status, and the DEFAULT_TARGET parameter is 1, the TLP is dropped.	no	no	no	no	Yes	Yes	Yes
When a request is filtered with UR/CA/CRS status, and the DEFAULT_TARGET parameter is 1, the destination is RTRGT1 interface.	Yes	Yes	Yes	Yes	no	no	no
When a CFG request is filtered with SU status and the CFG register address is > CONFIG_LIMIT, TARGET_ABOVE_CONFIG determines the destination.	-	-	Yes	-	-	-	-

Table continues on the next page...

Table 49-7. Routing Rules for Request TLPs (EP Mode) (continued)

When a CFG request is filtered with SU status and the CFG register address is < CONFIG_LIMIT, RTRGT0 interface is the destination.	-	-	Yes	-	-	-	--
The TLP is dropped, when the filter mask CX_FLT_MASK_MSG_DROP bit is not set and the non-Vendor MSG is filtered with SU status.	-	-	-	-	no	no	Yes
The TLP is dropped, when the filter mask CX_FLT_MASK_VENMSG0_DROP bit is 0 and the VEN0 MSG is filtered with SC status.	-	-	-	-	Yes	no	no
The TLP is dropped, when the filter mask CX_FLT_MASK_VENMSG1_DROP bit is 0 and the VEN1 MSG is filtered with SC status. RTRGT1 interface is the destination when none of the previous rules are satisfied and the MSG is filtered with SU status.	- -	- -	- -	- -	no Yes	Yes Yes	no Yes

Table 49-8. Routing Rules for Request TLPs (RC mode)

Routing Rule	MRd	MWr	[1]CFG	[1]IO	Vendor MSG Type0	Vendor MSG Type1	Other MSG
When a request is filtered with SU status, and is not in BAR range, RTRGT1 is the destination.	Yes	Yes	no	Yes	-	-	-
[2]When a request is filtered with SU status, and is in BAR range, MEM_FUNC#_BAR#_TARGET_MAP parameter determines the destination.	Yes	Yes	no	no	-	-	-
When a request is filtered with UR/CA status, the TLP is dropped. For NP requests, a CPL is also generated.	Yes	Yes	Yes	Yes	Yes	Yes	Yes
The TLP is dropped, when the filter mask CX_FLT_MASK_MSG_DROP bit is 0 and the non-Vendor MSG is filtered with SU status.	-	-	-	-	no	no	Yes
The TLP is dropped, when the filter mask CX_FLT_MASK_VENMSG0_DROP bit is 0 and the VEN0 MSG is filtered with SU status.	-	-	-	-	Yes	no	no
The TLP is dropped, when the filter mask CX_FLT_MASK_VENMSG1_DROP bit is 0 and the VEN1 MSG is filtered with SU status. RTRGT1 interface is the destination when none of the previous rules are satisfied and the MSG is filtered with SU status.	- -	- -	- -	- -	no Yes	Yes Yes	no Yes

1. DM (in RC mode) should not expect to receive a CFG or IO request.

2. BARs are not normally used in RC application.

49.3.4.3.5 Completion TLP Routing Rules

The table found here shows the applicability of routing rules for Completion TLPs, and indicates whether the destination is as stated by the rule when the conditions of the rule are met.

In summary, under error conditions, CPLs are never dropped but always forwarded to the application except when:

1. 'suggested' completion or error status is any of the following:
   - UC
   - 'DLLP abort'
   - 'ECRC error'

   and

2. queue mode is store-forward and
3. queue architecture is single or multiple queue.

Table 49-9. Routing Rules for Completion TLPs.

Routing Rule	Filter Status of the Completion (CPL)	
	SC / UR / CA / CRS	UC / 'DLLP abort' / 'ECRC error'
Core Drop is the destination when queue mode is store-forward and queue architecture is single queue or multiple queue	no	yes
RCPL interface is the destination when queue mode is <any> and queue architecture is single queue or multiple queue	yes	yes
RBYP interface is the destination when queue mode is by-pass and queue architecture is segment buffer queue.	yes	no
RTRGT1 interface is the destination when queue mode is cut-through and queue architecture is segment buffer queue.	yes	no
RTRGT1 interface is the destination when queue mode is store-forward and queue architecture is segment buffer queue.	yes	no
Core Drop is the destination when queue mode is <any> and queue architecture is segment buffer queue.	no	yes

49.3.4.4 Receive Queuing

The core support three configurable queue architectures per VC: single queue, multiple queue, and segmented queue.

Each queue architecture supports three buffering modes: bypass mode, cut-through mode, and store-and-forward mode. The buffering mode is selectable for each TLP type: posted, non-posted and completion. The configurability is dependent on the queuing architecture.

The single queue architecture has one header buffer and one data buffer; and the header and data buffers are used as a single FIFO for buffering all posted, non-posted and completion TLP.

The multiple queue architecture has a single header and data buffer per posted, non-posted and completion TLP type.

The segmented queue architecture has one header buffer and one data buffer, but these two buffers are segmented by posted, non-posted and completion TLP type (versus single queue architecture).

For all queuing modes, RAM modules are either instantiated inside the top-level module of the core or connected externally, which is configurable.

49.3.4.4.1 Queuing Architecture

The queue architecture is specified by the user using the CX_RADMQ_MODE configuration parameter.

49.3.4.4.1.1 SEGMENTED-BUFFER RECEIVE QUEUE CONFIGURATION (CX_RADMQ_MODE=2)

The segmented-buffer queue architecture is designed for applications that want to enforce an ordering rule other than FIFO. This is the only queue architecture that can strictly adhere to the ordering rules of the *PCI Express Specification*. This queue architecture is relatively large in area. The segmented-buffer configuration uses a single memory module pair (header and data) for all TLP types and all VCs.

In the segmented-buffer configuration:

- The memory width is set automatically.
- The memory is divided into segments for Posted, Non-Posted, and Completion queues for each VC. The depth of each segment is set during hardware configuration,
- The operating mode is selected (bypass, cut-through, or store-and-forward) independently for each TLP type of each VC during hardware configuration. The operating mode (per TLP type and VC) can be controlled dynamically by writing to the Port Logic registers. If a TLP type is configured to be cut-through or store-and-forward, then it will be routed to the RTGT interfaces. If a TLP type is configured to be bypassed, then it will be routed to the radm_bypass interface. Once a Posted Request is configured in bypass mode, the application should not expect to send a Posted-write to the ELBI interface.

- The number of advertised credits is selected independently for each TLP type and each VC during hardware configuration. The number can be selected dynamically by writing to the Port Logic registers.
- The receive queue priority for VCs can be set to either strict priority (higher-numbered VCs have higher priority) or round robin during hardware configuration; and the priority at runtime can be set by writing to the Port Logic registers.
- The ordering rules for TLP types can be set during hardware configuration and at runtime by writing to the Port Logic registers. The choices are either strict priority (Posted first, Completion second, Non-Posted third) or priority determined according to ordering rules set forth in the PCI Express 3.0 Specification.

49.3.4.4.2 Queue Modes

The queue mode for each P/NP/CPL queue is specified using the RADM_P_QMODE_VC0 , RADM_NP_QMODE_VC0 and RADM_CPL_QMODE_VC0 configuration parameters.

It is possible to change the Queue Mode (during device setup by software) by writing to the appropriate queue control register.

BYPASS MODE (RADM_P/NP/CPL_QMODE_VC0=4)

Queues in bypass mode are completely bypassed.

In the case of aborted TLPs, the core asserts either radm_trgt1/cpl/bypass_dllp_abort or radm_trgt1/cpl/bypass_tlp_abort to the application. The application is responsible for rolling back any actions that were performed on behalf of the aborted TLP.

CUT-THROUGH MODE (RADM_P/NP/CPL_QMODE_VC0=2)

In cut-through mode, the queue presents data to the application as soon as the first data for a packet is placed into the queue.

In the case of aborted TLPs, the application is responsible for rolling back any actions that were performed on behalf of the aborted TLP. The radm_trgt1/cpl/bypass_dllp_abort and radm_trgt1/cpl/bypass_tlp_abort signals are presented to the application at the same time as the eot signal.

STORE-AND-FORWARD MODE (RADM_P/NP/CPL_QMODE_VC0=1)

In store-and-forward mode, only valid TLPs are forwarded to the application logic. Therefore, no rollback functionality is required by the application. All TLPs with radm_trgt1/cpl/bypass_dllp_abort or radm_trgt1/cpl/bypass_tlp_abort asserted will be dropped[1] by the receive queues of the core.

Flow Control credits are returned by the queue as packets are read out (even when the TLP was aborted and not presented to the application).

NOTE

Error handling by the application is dependent on the choice of queue mode. See Dependency upon Queue Architecture and Mode.

49.3.4.4.3 Order Enforcement

The ordering of TLPs within the same VC depends on the queuing architecture as follows:

- In the segmented-buffer configuration, either PCIe ordering rules or strict priority ordering are provided.
- In segmented buffer mode, you may select either strict VC priority (same as for single- and multi- buffer) or round robin.

For more information of packet ordering in relation to buffering/queue mode and architecture see:

- App Note: Order Enforcement Using the PCIe Core
- Inbound Order Enforcement for AXI Bridge

49.3.4.4.4 Queue to Port Mapping

Tables found here indicate which interfaces (ports) are used to deliver requests and completions depending on the queue architecture and buffer mode selected.

For Posted (P) and Non Posted (NP) TLP's, the destination (RTRGT1/RTRGT0) depends on the BAR setup, and if the TLP is of CFG type or not.

See Receive Routing for more details on routing for TLP's under error conditions.

QUEUE TO PORT MAPPING WHEN AHB/AXI BRIDGE PRESENT

RTRGT1 is connected to the AXI/AHB Bridge master interface (request) channel. RCPL and RBYP are connected to the AXI/AHB Bridge slave interface (response) channel. See Receive Routing for more details on routing for TLP's under error conditions.

Table 49-10. Segmented Queue Architecture (CX_RADMQ_MODE == 2) Queue to Port Mapping

Queue Mode (RADM_P/NP/CPL_QMODE_VC0)	Posted TLP Queue	Non Posted TLP Queue	CPL Queue

Table continues on the next page...

Table 49-10. Segmented Queue Architecture (CX_RADMQ_MODE == 2) Queue to Port Mapping (continued)

Store and Forward (1)	RTRGT1 / RTRGT0	RTRGT1 / RTRGT0	[1]RTRGT1
Cut Through (2)	n/a	n/a	n/a
Bypass (4)	n/a	n/a	RBYP

1. Use this feature if you want to maintain PCI Express ordering rules within the AHB/AXI Bridge module after packets have left the RADM receive queues in the native PCIe core. For more information see Ordering Enforcement Hardware Lock Feature.

49.3.5 Error Handling

An overview and additional information regarding error handling can be found here.

49.3.5.1 Error Handling Overview

Errors are classified into two levels:

- Correctable Error (CORR). This means that the PCIe core has a way of automatically handling the error. There is no loss of information. For example, Link CRC (LCRC) that is fixed by replaying the DLL.
- Uncorrectable Error (UNCORR). The PCIe core can not fix these and they are classified as:
- Fatal Error (FATAL). The link is not functioning correctly and may require a link reset.
- Non-Fatal Error (NONFATAL). The problem is not related to link operation.

The core implements the types of error handling found here.

- PCIe Baseline Capability. These reporting capabilities are a minimum set, and are required of all PCI Express devices. Error notification takes two forms:
- Messages sent to Root Complex (RC).
- Completion Status errors.

This also covers mapping of PCIe errors to legacy PCI generic error handling such as PERR# and SERR#. Many of the PCIe errors are mapped into the Status register in the PCI Compatible Configuration Space Header.

- PCIe Advanced Error Reporting (AER) Capability. Allows more sophisticated error reporting, control, masking and logging using the PCIe extended AER capability register structure.

The PCIe core supports Advisory reporting for both the Baseline and AER capabilities, which is the configurable with-holding of reporting for Non-Fatal errors (NONFATAL).

For an RC port, the reporting of most errors is internal to the root port. No external error notifications are generated. One exception to this (for example) is Unsupported Request (UR) completion status.

49.3.5.2 PCIe Baseline Capability

Reporting of errors is achieved by sending a notification to the RC (a CPL with UR/CA/CRS status for Non Posted requests, and optionally an error Msg).

The decision to send an error MSG is controlled by a complex set of associated control and status bits. The status is also logged in the Device Status Register for the following errors: Unsupported Request (UR), FATAL, NONFATAL and CORR.

MESSAGES SENT TO ROOT COMPLEX (RC).

Messages sent to the RC are of the ERR_CORR, ERR_NONFATAL, and ERR_FATAL types.

COMPLETION STATUS ERRORS.

Completion Status errors for Non Posted requests may be any of the following:

- Unsupported Request (UR)
- Configuration Request Retry Status (CRS)
- Completer Abort (CA)

REPORTING THROUGH THE DEVICE STATUS REGISTER

The PCI Express Capability Register Structure provides the following support for Baseline Error Reporting.

- Enable/disable error reporting (Device Control Register).
- Provide error status (Device Status Register) for:
- UR
- Correctable Error (CORR).
- Fatal Error (FATAL).
- Non-Fatal Error (NONFATAL).
- A method for software to force Link Retraining (Device Control Register).

49.3.5.3 Advanced Error Reporting (AER)

The decision to send an error MSG is controlled by a complex set of associated control and status bits. For more details, see the flow diagram in Error Detection. AER allows more sophisticated error reporting, control, masking and logging using the optional extended AER capability register structure. By default, AER is enabled (AER_ENABLE parameter), and may not be disabled unless you disable ECRC support in the core (CX_ECRC_ENABLE parameter).

AER REGISTERS

The AER registers are described in the Advanced Error Reporting Capability Registers section described in . All possible errors are enabled, masked and assigned a severity.

There are two sets of registers:

- Error Enable Register
- Error Severity Register
- Error Mask Register.

The Correctable set of registers handles (for example) errors arising from bad DLLPs or TLPs.

The Uncorrectable set of registers handles (for example) errors arising from UR, ECRC, Malformed TLPs, Buffer Overflow, UC, CA, Completion Timeout and Poisoned TLP.

SEVERITY PROGRAMMING

The Uncorrectable Error Severity register allows each uncorrectable error to be programmed to Fatal or Non-Fatal. The transmission of these error Messages by class (correctable, non-fatal, fatal) is enabled using the Reporting Enable fields of the Device Control register or the SERR# Enable bit in the PCI Command register.

The Uncorrectable Error Mask register and Correctable Error Mask register allows each error condition to be masked independently. If Messages for a particular class of error are not enabled by the combined settings in the Device Control register and the PCI Command register, then no Messages of that class will be sent regardless of the values for the corresponding mask register. If an individual error is masked when it is detected, its error status bit is still affected, but no error reporting Message is sent to the Root Complex, and the Header Log and First Error Pointer registers are unmodified.

49.3.5.3.1 Advisory Non-Fatal Error Messages

The PCIe core supports Advisory reporting which is the configurable with-holding of reporting for Non- Fatal errors.

- With Baseline Error Reporting, the core produces no error message.
- With AER, the core can instead, signal a non-fatal error with ERR_COR, which serves as an advisory notification to software.

It will always signal a fatal error with ERR_FATAL

UR/CA ADVISORY

The PCIe core generally sends a CPL with UR/CA status to signal a uncorrectable error for a Non-Posted Request. If the severity of the UR/CA error is non-fatal, the PCIe core will handle this case as an Advisory Non-Fatal Error.

By default, the PCIe core will signal the non-fatal error (if enabled) by sending an ERR_COR Message.

If AER is disabled (AER_ENABLE=0), the PCIe core sends no error Message for this case. Even though there was an uncorrectable error for this specific transaction, the PCIe core will handle this case as an Advisory Non-Fatal Error, since the Requester upon receiving the Completion with UR/CA Status is responsible for reporting the error (if necessary) using a Requester-specific mechanism.

UC ADVISORY

When the PCIe core receives an UC and the severity of the UC error is non-fatal, the PCIe core will handle this case as an Advisory Non-Fatal Error. By default, the PCIe core will signal the error (if enabled) by sending an ERR_COR Message.

If AER is disabled (AER_ENABLE=0), the PCIe core sends no error Message for this case.

49.3.5.4 Error Source Classification

The table found here indicates how some of the more common low level errors are classified.

Table 49-11. Possible Causes for Typical Errors

Error Type	Possible Cause
UR (Unsupported Request)	•Poisoned TLP (EP=1) •No BAR match •MRd length > max read request size
UC (Unexpected Completion)	•TAG mismatch. •Requester ID (RID) mismatch.
CPL TimeOut	Remote device hung.
CA (Completion Abort)	ECRC

Table continues on the next page...

Table 49-11. Possible Causes for Typical Errors (continued)

Malformed TLP (MLF)	Bad TLP header caused by bad link.
Buffer OverFlow	Credit miscalculation by some PCIe device.
Bad DLLP	LCRC

For a full analysis of what error conditions contribute towards an UR or CA status, see Receive Filtering.

NOTE

In many cases, the standard operation may be 'masked' or ignored by setting the corresponding bit in the Symbol Timer Register and Filter Mask Register 1, which take its default value from the DEFAULT_FILTER_MASK_1 configuration parameter.

NOTE

For example, A TLP with the poison bit set (EP=1) is considered an Unsupported Request (UR) only when the UR poison rule mask bit is not set.

49.3.5.5 Error Detection

Built into the core are all mandatory error detections, some optional error detections, and the error report mechanism based on the *PCI Express Specification*.

The core also has an option for the application to turn off the filter rules and perform its own error checking. For more details, see Receive Filtering.

The following general rules apply to all incoming TLPs:

- The core discards all incoming TLPs that have an invalid Type field. This TLP is treated as a 'TLP-ABORT'.
- A locally terminated TLP with ECRC error detected is discarded in store-and-forward mode and an ECRC error reported only when the filter mask CX_FLT_MASK_ECRC_DISCARD bit is not set.
- Filter rules have no affect on received TLP when 'DLLP-ABORT' signal is asserted.
- If a completion of a non-posted request is not received within a completion timeout period, this request will be treated as a completion timeout, and a non-advisory error will be reported. See Advanced Error Reporting (AER) for more details.
- 'DLLP-ABORT' is asserted as a result of one of two conditions:
- a. A data link layer error is detected (e.g. LCRC). A retry from a remote device will occur.

b. UC or completion with ECRC error is detected. This condition is valid only when the application has configured the core with infinite credits. Because the completion buffer of the core or application has limited resources defined for expected completions, it is necessary to avoid overflowing the completion buffer by unexpected completions. Therefore 'DLLP-ABORT' is asserted to notify the core completion buffer (if completion is in store-forward mode) or application's completion buffer to rewind their buffer pointers when a completion with ECRC error or unexpected completion is detected.

- 'TLP-ABORT' is asserted as a result of one of three conditions:
- Malformed TLP
- UC
- ECRC

The figure below provides a flow chart of the error detections for a packet received from the PCIe wire.

Signals in this figure are internal to the core (except those prefixed with radm_*)

The signals radm_cpl_dllp_abort and radm_cpl_tlp_abort are intended for use with the Completion queue configured in either bypass or cut-through mode so that the core can notify the application of errors detected as the TLP is received.

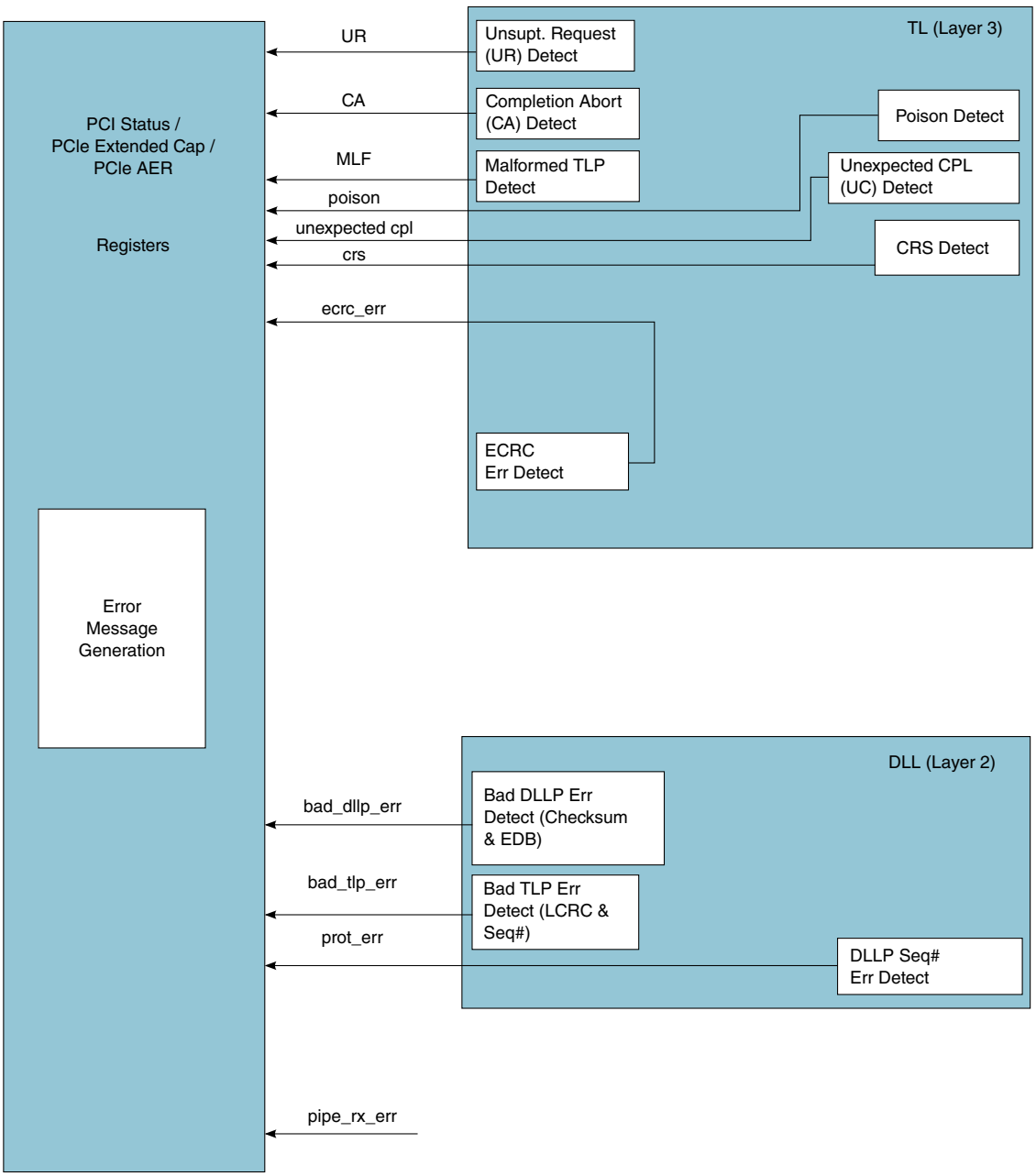


Figure 49-13. Flow Chart of Error Detections

49.3.5.6 Application Error Reporting Interface

The application may optionally generate an Error Message through either of the following methods:

- Directly at the application interface see Message Generation.

- It may instruct the PCI Express core to do so through the 'Application Error Reporting Interface'nsignals app_* .
- You must set the APP_RETURN_ERR_EN configuration parameter to enable the 'Application Error Reporting Interface'.

If you mask detection of Completion timeout errors, through setting the CPL_TIMEOUT_ERR_MASK configuration parameter, then the core will not automatically report Completion timeout errors. The application must check for Completion timeouts and report Completion timeout errors using the app_err_bus input signal.

NOTE

Your application may want to send an error message to the remote link partner if (for example) your application detected an error during the execution of an inbound/received Posted TLP, for example a MWr

49.3.5.7 Handling of General Errors with the AXI Bridge.

PCIe Advanced Error Reporting (AER) is not supported in the AXI bridge.

AER is only supported with respect to the native PCIe core. Errors detected by the AHB/AXI bridge are not reported as part of AER.

49.3.5.8 AXI

The slv_resp_err_map and mstr_resp_err_map bits are tied to 1. The corresponding CPL error will be SLVERR AXI Error Response. The corresponding CPL status error will be UR (Unsupported Request) PCIe CPL Status.

49.3.5.9 Handling of ECRC/LCRC Errors for IO/MEM with the AXI Bridge.

49.3.5.9.1 Link CRC (LCRC) - a correctable Error

When an LCRC error occurs on an inbound packet (request or completion), the core will automatically discard that packet and wait for the remote device to replay the packet.

The native core will signal (via radm_*_dllp_abort) to the AHB/AXI bridge that an LCRC error has occurred.

The AXI bridge will not signal an error to the application but will wait for the replayed packet to arrive.

49.3.5.9.2 End-to-end CRC (ECRC) - an Uncorrectable Non-Fatal Error

REQUESTS

By default, a request TLP with ECRC errors is dropped by the RADM filter in the native core and an Advisory Non-Fatal Error is signalled. For Non Posted (NP) requests, a CPL with CA is generated. See Advisory Non-Fatal Error Messages.

If you set the DEFAULT_TARGET and CX_MASK_UR_CA_4_TRGT1 parameters to 1 (default values are 0), then, a request TLP with ECRC errors will not be dropped but will be forwarded (without any error reporting) to the application on AXI bridge master.

See ECRC Handling for more details.

COMPLETIONS

A completion TLP with ECRC errors is only dropped by the RADM in the native core when CPL queue mode is store-forward and queue architecture is single or multiple queue. However, for the AHB/AXI bridge, when the queue architecture is single or multiple queue, the CPL queue mode is restricted to bypass. Therefore a completion TLP with ECRC errors is always forwarded to the AHB/AXI bridge.

When the AXI/AHB bridge receives a CPL with ECRC errors, it will not transmit it to the application through the AHB/AXI master. Instead, the bridge will wait for the subsequent 'completion timeout' (see Received Completion TLP Processing) generated by the core. It is this timeout that will cause the bridge to issue an ERROR response to the application on each beat of the burst corresponding to the original AHB/AXI request.

49.3.6 Messages

Information found here describes the processing of messages through the core.

NOTE

For a proper understanding of Messages you should be familiar with Message Request Rules of the PCI Express Base Specification.

Similar to MWr, messages (Msg/MsgD) are Posted transactions. The 8-bit 'Message Code' field defines what class of message the TLP is. Some examples of typical message classes are given in the following table.

Table 49-12. Some Message classes based on the Message code

Message Code [7:0]	Message Class	TLP Type	Note
0001_xxxx	Power Management	Msg	
0010_0xxx	Legacy PCI Interrupt	Msg	Assert/deassert for each of INT A/B/C/D.
0011_00xx	Error Signalling	Msg	ERR_CORR, ERR_NONFATAL, ERR_FATAL are encoded using 30h, 31h, 33h.
0111_11xx	Vendor Defined	Msg / MsgD	
Other classes (used by PCIe core) include Locked Transaction, Slot Power Limit.			

NOTE

Message Signalled Interrupts (MSI/MSI-X) are not messages (Msg/MsgD) but MWr TLPs. See Interrupts for more details.

49.3.6.1 Message Generation

Messages that are transmitted by the PCI Express core can potentially be derived from the following eight sources. Referring to the circled numbers in the following diagrams, outbound messages can be created either by:

the core automatically as follows:

- 'Power Management' messages.
- 'Error Signalling' messages.

or

the customer application as follows:

- Direct supply of message TLPs at AXI bridge master. An internal or external internal address translation unit (ATU) can convert IO/MEM TLPs to MSG TLPs.
- 'Locked Transaction' messages through the 'SII Message' interface [RC mode]. 'Legacy PCI Interrupt' messages through the 'SII Interrupt' interface.
- 'Error Signalling' messages through the 'SII Transmit Control' interface (app_err* I/O).

or

the host/client software as follows [RC mode]:

- Triggering the sending of 'Slot Power Limit' messages by writing to the Slot Capabilities Register.

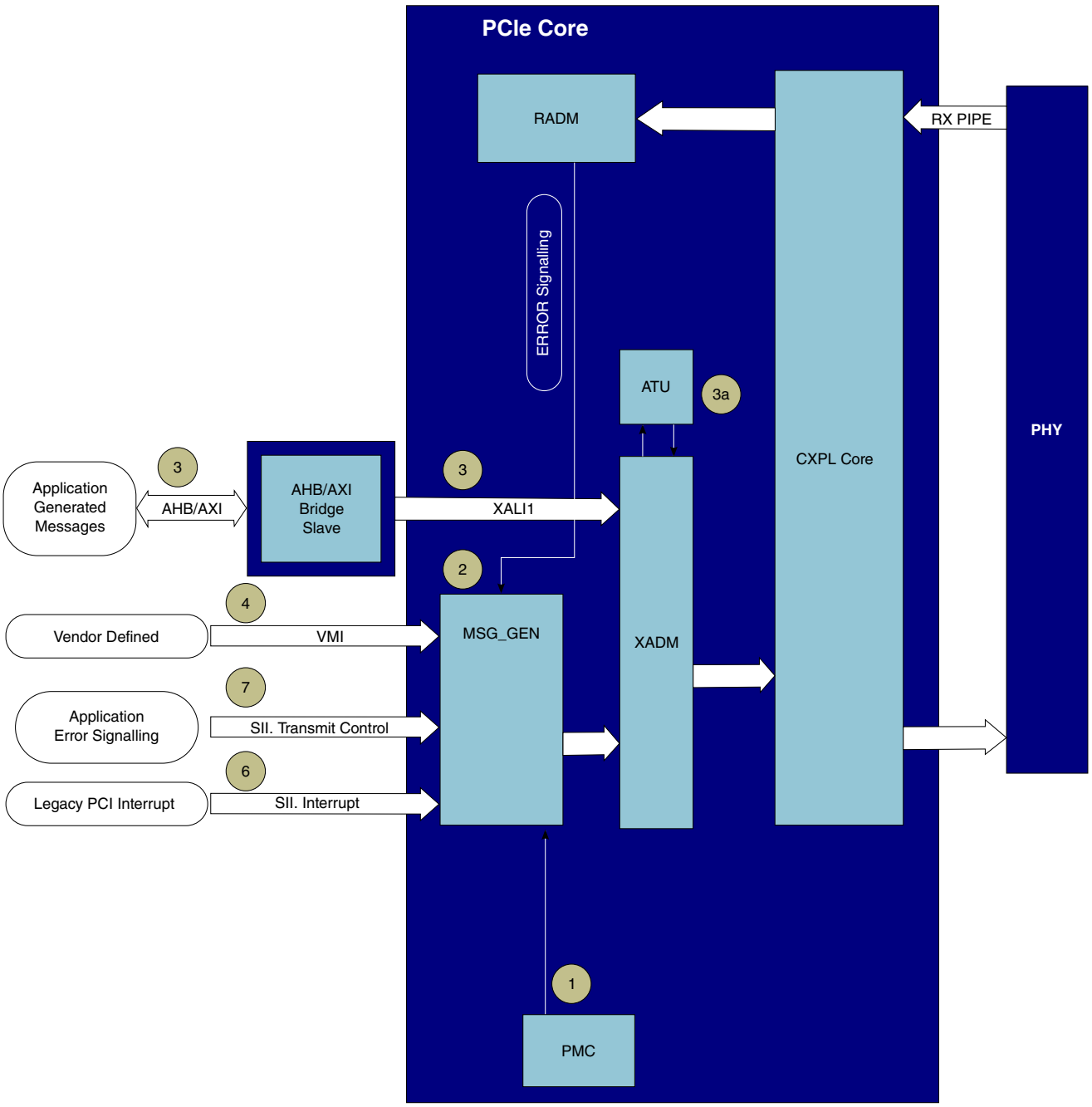


Figure 49-14. Message Transmission: EP mode

See Terms and Abbreviations for definitions of acronyms used for block and interface names.

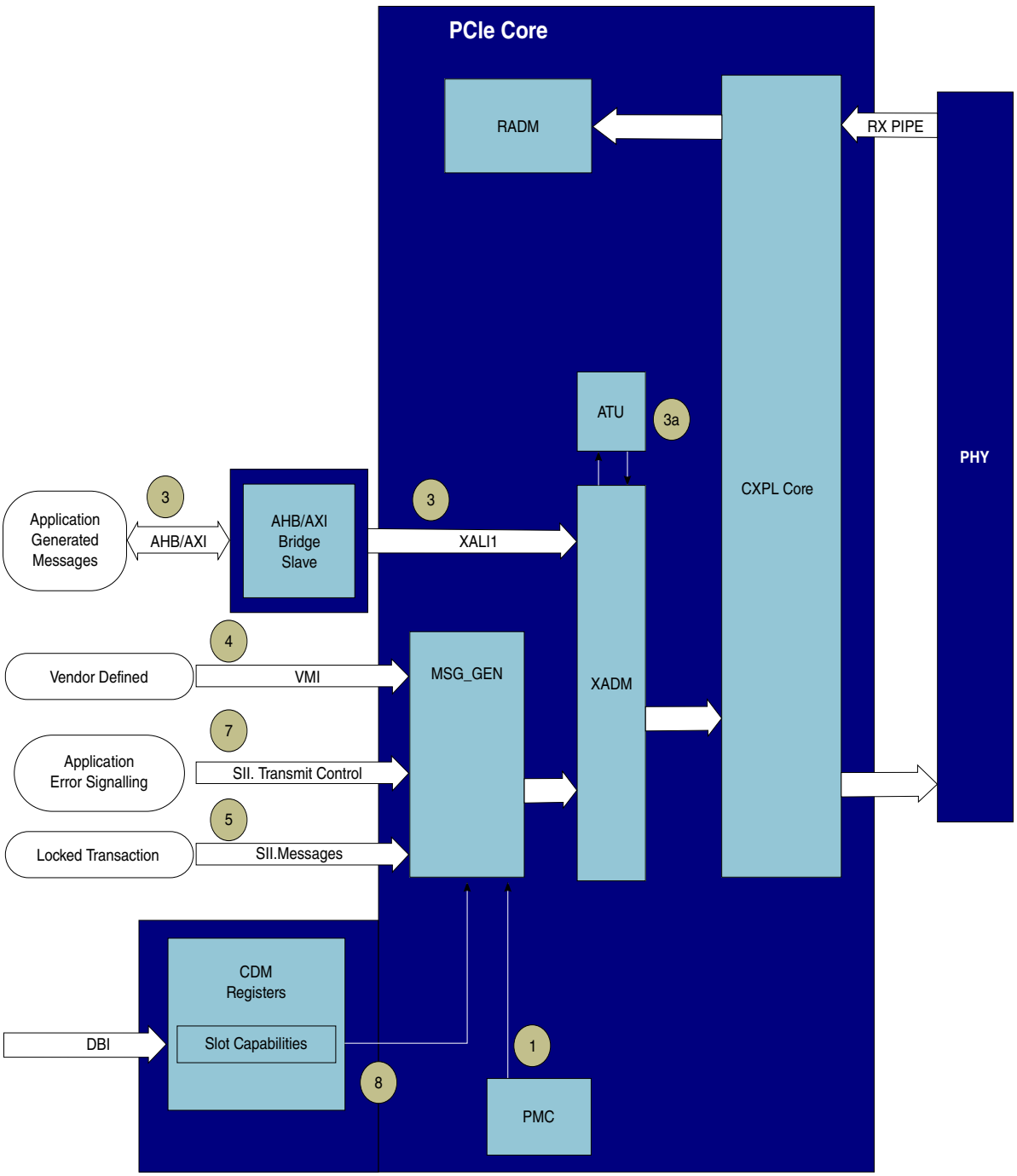


Figure 49-15. Message Transmission: RC mode

Table 49-13. Message Transmission. The 'Index' refers to the numbers in the previous diagrams

Index	Message Source (Type)	EP Mode	RC Mode

Table continues on the next page...

Table 49-13. Message Transmission. The 'Index' refers to the numbers in the previous diagrams (continued)

1	Power Management controller in the core (Msg).	PM_PME[1]	PME_Turn_Off[2] See Power Management.
2	Error Signalling inside the core (Msg).	COR_ERR / ERR_NONFATAL / ERR_FATAL. See Error Handling for more details.	n/a
3	Direct Supply of any class of Message (Msg/MsgD).	XALI0/1/2 or AHB/AXI See Application Msg/MsgD Programming Examples for details on how to generate a message at the XALI0/1/2 (or AHB/AXI) interfaces.	
3a	Indirect Supply of any class of Message (Msg/ MsgD).	See Outbound iATU Operation for more details on generating Msg/MsgD from MWr/IOWr using an internal or external address translation unit (ATU).	
5	Locked Transaction (Msg).	n/a	Unlock Message, triggered by Root Complex application logic via the app_unlock_msg pin.
6	Legacy PCI Interrupt (Msg).	'SII Interrupt' pins sys_int and dp_intx (see PCI Legacy Interrupt).	n/a
7	Error Signalling from the application (Msg).	The core generates Error Signalling Messages in response to application requests on the SII app_err* I/O . It is also possible to generate Error Messages via the client interfaces. See items 3 and 3a in this table.	
8	Slot Power Limit (Msg).	n/a	Set_Slot_Power_Limit Support Message, triggered by writing to the Slot Capabilities Register via the DBI.

1. Triggered by your EP application through the outband_pwrup_cmd or apps_pm_xmt_pme pin.

2. Triggered by your RC application through the apps_pm_xmt_turnoff pin.

3. MsgD not possible on VMI. See Vendor Defined Message (VDM) Generation.

49.3.6.1.1 Vendor Defined Message (VDM) Generation

VDMs can be generated by your application using any of the following methods (numbered 4, 3 and 3a in Table 49-13).

- Direct supply of IO/MEM TLPs AXI bridge master to be converted to VDM by either of the following.
- The internal Address Translation Unit (iATU) can convert IO/MEM TLPs to VDM TLPs.

49.3.6.1.1.1 Application Msg/MsgD Programming Examples

The tables found here enumerate the different ways your application can generate Msg and MsgD TLPs.

NOTE

xATU = "External Address Translation Unit" and iATU = "Internal Address Translation Unit (iATU)".

Table 49-14. Msg (message without payload) Generation Methods

Application Interface	Description	Application I/O Signals
AXI	Direct Supply using a Msg transaction.	slv_awmisc_info[4:0]='MSG" slv_wstrb[3:0]='0000'
	Indirect Supply (iATU) using a MWr[1] transaction. The iATU needs to be configured to translate MWr to Msg TLPs.	slv_awmisc_info[4:0]='MEM' slv_wstrb[3:0]='0000'

1. Or IOWr

NOTE

xATU = "External Address Translation" and iATU = "Internal Address Translation (iATU)".

Table 49-15. MsgD (Vendor Specific Message with payload) Generation Methods

Application Interface	Description	Application I/O Signals
AXI	Direct supply using an MsgD transaction.	slv_amisc_info[4:0]='MSG'
	Indirect Supply (xATU) using a MWr transaction.	slv_amisc_info[4:0]='MEM' xtranslated_enable =1 xtranslated_addr_in_d[8]=1 xtranslated_addr_in_d[7:0]=Message Code xtranslated_type_in_d = 10xxx
	Indirect Supply (iATU) using a MWr[1] transaction. The iATU needs to be configured to translate MWr to MsgD TLPs.	slv_amisc_info[4:0]='MEM'

1. Or IOWr

49.3.6.1.2 AHB/AXI Message Address and Size Limitations

Limitations existing in the current implementation of the AHB/AXI bridge module can be found here.

NOTE

- Vendor Messages must not be decomposed in the inbound or outbound direction.
- You must ensure that Vendor Defined Messages generated by your application or remote link partner, do not trigger decomposition as described in AXI Decomposition Rules.

Table 49-16. Processing of Inbound Messages when AHB/AXI and PCIe Core Address Widths are different

AHB/AXI Address Bus Width MASTER_BUS_ADDR_WIDTH	PCIe Core Address Bus Width FLT_Q_ADDR_WIDTH	Notes
32	64	The upper 32 bits of the data field (bytes[1] 8-11) are not forwarded by the bridge master.
64	32	The upper 32 bits of data field (bytes 8-11) will be forced to '0'.

49.3.6.2 Message Reception

The PCI Express core can receive the following types of messages. The index in the first column refers to the circled numbers in the following diagrams.

Table 49-17. Message Reception. The 'Index' refers to the numbers in the following diagrams

Index	Message Source (Type)	EP Mode	RC Mode
1	Power Management (Msg).	PME_Turn_Off See Power Management.	PM_PME PME_TO_Ack
1a	Slot Power Limit (Msg).	Set_Slot_Power_Limit Support Message.	n/a
2	Error Signalling from downstream component (Msg).	n/a	COR_ERR / ERR_NONFATAL / ERR_FATAL.
3	Vendor Defined (Msg/MsgD).		
4	Locked Transaction (Msg).	Unlock Message.	n/a
5	Legacy PCI Interrupts from downstream devices (Msg).	n/a	See PCI Legacy Interrupt.

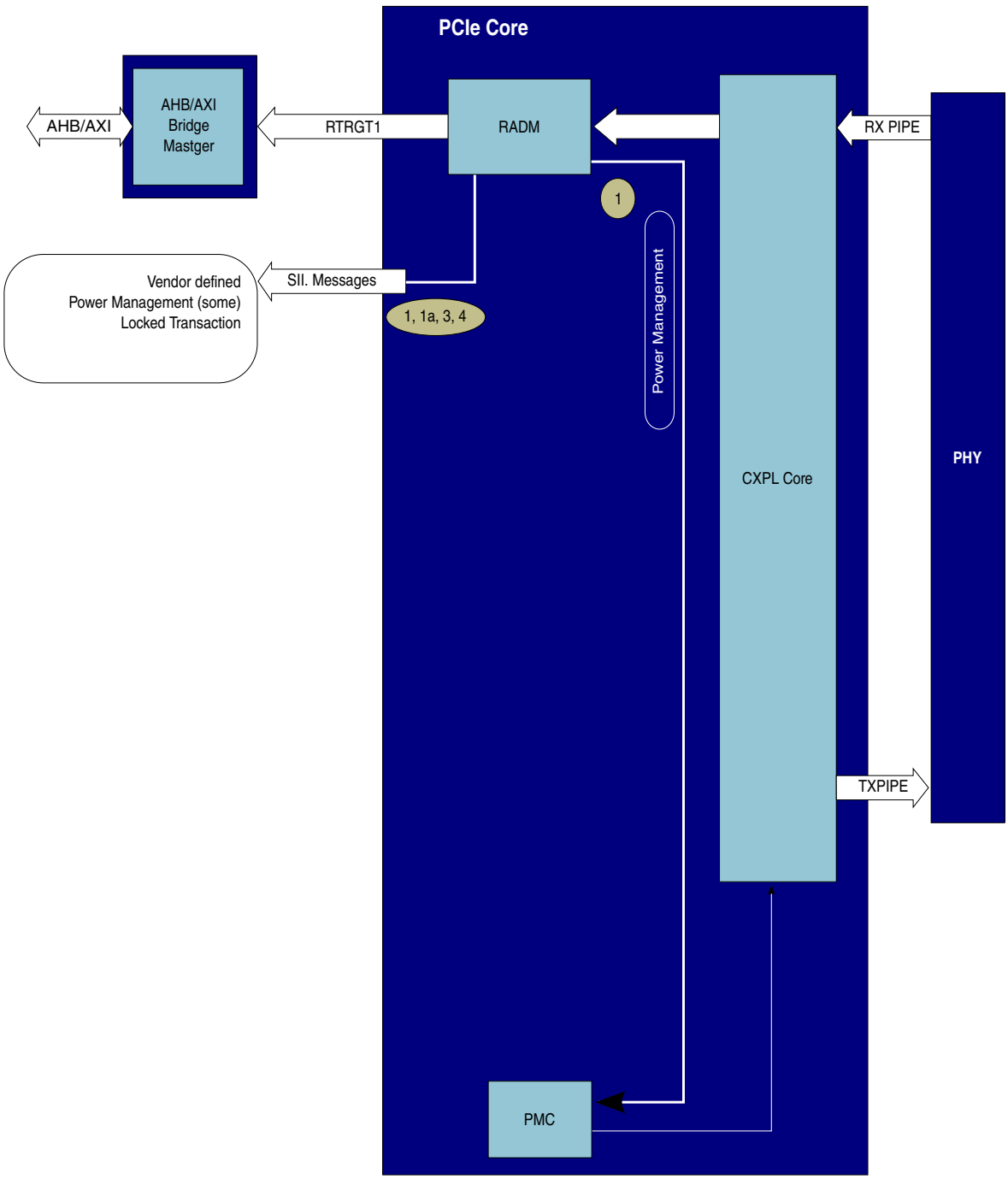


Figure 49-16. Message Reception: EP mode

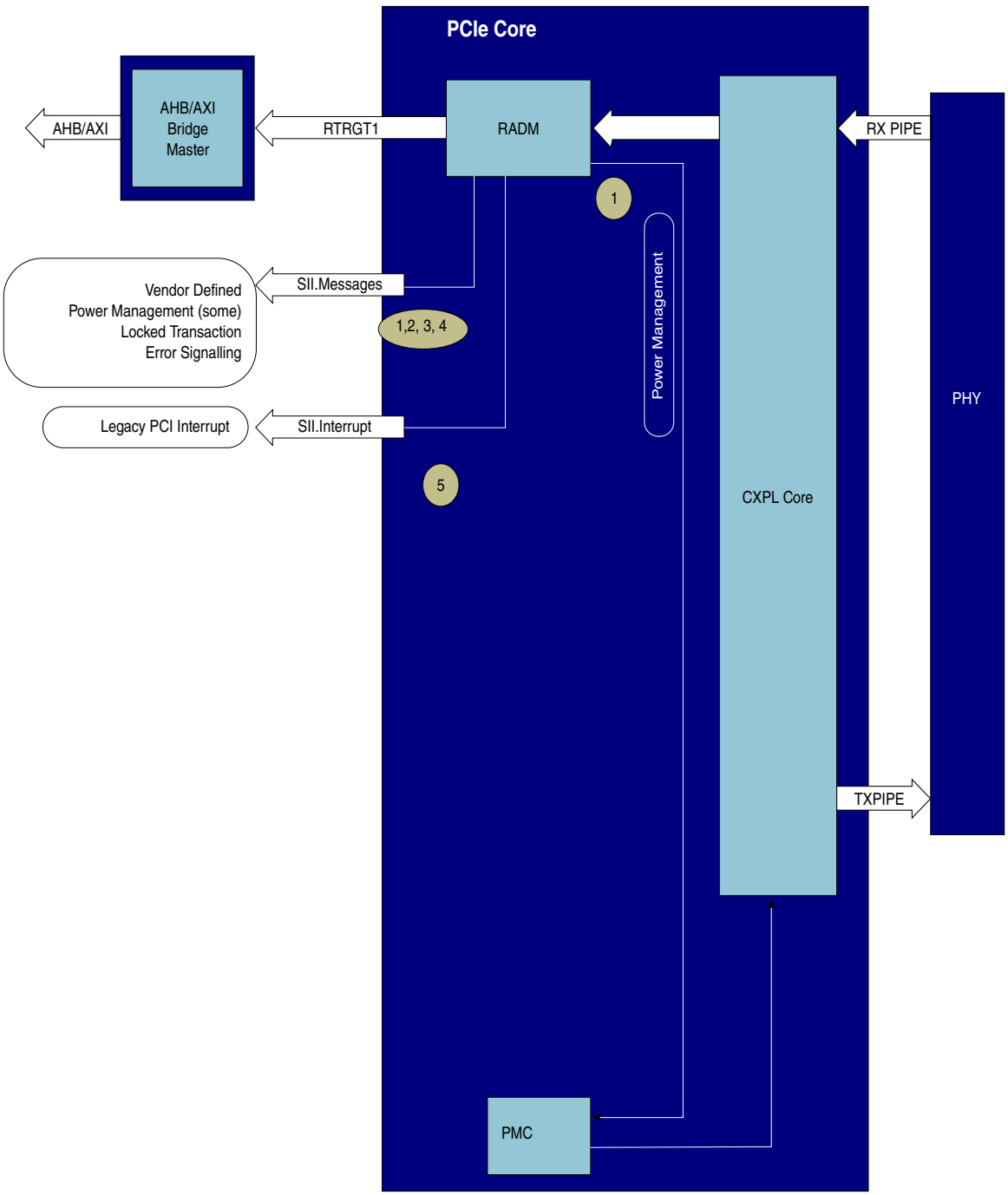


Figure 49-17. Message Reception: RC mode

See Terms and Abbreviations for definitions of acronyms used for block and interface names.

49.3.6.2.1 Message Reception IO Interfaces

The RADM filter provides a Message interface - that is grouped as part of the System Information Interface (SII) - to handle the Message TLPs received from the upstream component. The RADM filter processes the Message and decodes the header before sending it to the application logic on the SII.

49.3.6.2.2 Routing of Received Messages to SII and optionally to Application

The RADM filter processes *every* received message and decodes the header before sending it to the application logic on the System Information Interface (SII).

In addition, power management messages are processed by the PCIe core Power Management Controller (PMC).

By default, all received messages are dropped[1] (serviced internally) and *not* passed to the application on AXI bridge master.

To have all decoded messages *also* sent to the application interface (AXI bridge master), the register fields must be set to '1'. These registers allow you to override any decisions (regarding MSG routing) made at configuration time by the FLT_DROP_MSG, DEFAULT_FILTER_MASK_1 and DEFAULT_FILTER_MASK_2 configuration parameters .

Table 49-18. Controlling the Routing of Received Messages

Register	Bit	Function	Default Value
Filter Mask Register 1	29	Mask the dropping of Non-Vendor Messages 0: Drop 1: Do not drop	DEFAULT_FILTER_MASK_1[13] = ! FLT_DROP_MSG = 0
Filter Mask Register 2	0	Mask the dropping of Vendor Type 0 Messages 0: Drop[1] 1: Do not drop	DEFAULT_FILTER_MASK_2[0] = 0
Filter Mask Register 2	1	Mask the dropping of Vendor Type 1Messages 0: Drop 1: Do not drop	DEFAULT_FILTER_MASK_2[1] = 0

1. Vendor TYPE0 Messages are dropped with UR error reporting.

For the masking (of the dropping) of Vendor messages, it is not possible to differentiate between 'Vendor Message without Payload (Msg)' and 'Vendor Message with Payload (MsgD)'.

Full details of the Filter Mask Registers are at Symbol Timer Register and Filter Mask Register 1and Filter Mask Register 2.

When a MSG request is filtered with UR/CA/CRS status, the TLP is always dropped. Only MSG requests filtered with SC status, can potentially be forwarded to the application on AXI bridge master.

49.3.6.2.3 Accessing Header and Payload Fields of Received Messages

Format of Received Messages on the Application RTRGT1 Interface

In addition to normal TLP header information, the last two DWORDS of the message header (bytes[1] 8-15) are presented on radm_trgt1_addr[FLT_Q_ADDR_WIDTH-1:0] when radm_trgt1_hv is asserted.

Format of Received Messages on the Application AHB/AXI Master Interface

In addition to normal TLP header information, the last two DWORDS of the message header (bytes[1] 8-15) are presented on the master address bus as follows:

- AHB/AXI address[31:0] = 4th DWORD = bytes 12-15
- AHB/AXI address[63:32] = 3rd DWORD = bytes 8-11

The TYPE field of mstr_awmisc_info/mstr_req_misc_info indicates the type of message TLP received. The remote link partner should not use the last DWORD of a Msg/MsgD header (bytes[1] 12-15). If the last DWORD must be used, then the two lower bits (byte15) must be always set to 00b. If these two bits are not zero, then the AHB/AXI master interface (when it forwards the message packet to the AHB/AXI fabric) will initiate a burst of 8-bit or 16-bit transactions.

Messages Without Payload Restriction on AHB

Messages without Payload (Msg) cannot be forwarded to the AHB bridge module master interface, as there is no concept of 'write strobes' in AHB to support null/empty write packets. Therefore it is only possible to access (parts of) the received Msg (message without payload) over the SII.

Format of Received Messages on the SII

Not all of the message TLP fields are sent to the SII. For some messages, only an indication that it has been received is signalled. For Vendor Defined messages (Msg or MsgD), the 3rd DWORD (bytes[1] 8-11) is sent to radm_msg_payload[31:0] but the fourth DWORD (bytes 12-15) is not presented at all. Furthermore, for Vendor Defined messages with payload (MsgD), the payload is not presented on SII.

Therefore, to access this missing information (if required), it is necessary to unmask the dropping[2] of Vendor Messages and have them sent to the application interface (AXI bridge master) in order to access the payload. This can be problematic for AHB as it is not possible to differentiate between 'Vendor Message without Payload (Msg)' and 'Vendor Message with Payload (MsgD)' for the masking (of the dropping) of Vendor messages. To avoid, this problem, program the remote link partner to not send Vendor Defined messages without payload (Msg), to an AHB connected end point.

49.3.7 Interrupts

Information found here describes the processing of interrupts through the core.

49.3.7.1 Interrupts Overview

The application logic in a PCI Express Endpoint may use one of three methods to signal an interrupt:

PCI legacy interrupt

PCI includes up to four virtual interrupt wires, referred to as INTA, INTB, INTC, and INTD. These wires are shared by all the PCI devices in the system. PCI Express emulates this capability by providing Assert_INTx and Deassert_INTx Message packets sent through the PCI Express serial Link.

MSI

A PCI Express Endpoint may signal an MSI by sending a standard PCI Express Posted Write packet towards the Root Port. The packet must contain a specific address and one of up to 32 data values. The varying data values, and the address value provide more detailed identification of interrupt events than legacy interrupts.

The PCI Express Cores support optional MSI per-vector masking (PVM).

MSI-X

An MSI-X interrupt is identical to an MSI, except that an Endpoint may use one of up to 2048 address and data pairs in the MSI-X Posted Write packet. Endpoints with MSI-X capability also include application logic to mask and hold pending interrupts, as well as a memory table for the address and data pairs. The large number of address values available to each Endpoint allows MSI-X Messages to be routed to different interrupt consumers in a system, as compared to the single address available to MSI packets.

Root Ports cannot send MSI-X packets. In complex systems, MSI-X packets could be routed to devices other than the RC, including other Endpoints, based on the multiple address/data pairs available.

Support of legacy interrupts and/or MSI/MSI-X may be required for backward compatibility.You may configure your core and application logic to support all three types of interrupts. However, you may only use one of these capabilities at a time. When host software clears the MSI Enable bit, you may only use legacy interrupts. When host software sets the MSI Enable bit, you may only use MSI. If if host software enables MSI or MSI-X, legacy interrupts are automatically disabled. Functionality is undefined if both MSI and MSI-X are enabled.

49.3.7.1.1 PCI Legacy Interrupt

NOTE

Legacy Interrupt delivery are signalled using special Msg TLPs. For more details see Messages.

The PCI Express Endpoint Core provides an input pin (sys_int[NF-1:0]) per function, so that application logic can assert or deassert a legacy interrupt.

The Core automatically maps assertion/de- assertion edges on the input pin to PCI Express Assert or Deassert messages. Root Ports decode received Assert/Deassert messages into pulses. Refer to System Information Interface (SII) for additional information.

Multifunction Support

A single-function Endpoint always uses INTA. However in a multi-function Endpoint, you may choose which virtual interrupt is used for each function by setting the initial value of the Interrupt Pin configuration register. In a multi-function Endpoint, each function has its own interrupt input pin (sys_int[NF-1:0]). Each function's Interrupt Pin register determines which legacy interrupt Message the function uses (INTA, INTB, INTC, or INTD).

Ordering Considerations

You may wish to guarantee that a legacy interrupt Message is sent after a data packet. In that case, do not assert the interrupt input pin until after the data packet's header is accepted by the core's transmit client interface.

In a complex PCI Express system, which include Switches and/or multiple Virtual Channels, you cannot guarantee that the interrupt Message will arrive after a data packet, unless the data packet uses the same Traffic Class as the legacy interrupt packet (Traffic Class 0).

Deassertion of Interrupts

The application needs to deassert the virtual interrupt inputs if system software has disabled interrupts or if the PCI Express link has been placed in a low power state.

The Core does automatically send a Deassert Interrupt Message when software disables interrupts. However, the application must eventually deassert the virtual interrupt before it can send a new interrupt because the Core requires a rising edge on the virtual interrupt signal to generate a new Assert Interrupt Message.

The Core does not automatically send a Deassert Interrupt Message when the power state changes.

49.3.7.1.2 MSI

Message Signalled Interrupts (MSI) are not messages (Msg/MsgD) but Memory Write (MWr) TLPs. They are indistinguishable from normal MWr's apart from the target address used in conjunction with the MSI Capability Structure. PVM (Per Vector Masking) is supported with MSI.

The PCI Express Core automatically builds an MSI packet for your application (if MSI is enabled) whenever requested by your application logic.

A simple handshake is required. The MSI interface is used only in the DM core in EP mode.

The Core informs the application logic whether MSI interrupts are enabled, and how many MSI data vectors have been allocated by system software.

Before performing the handshake, application logic must assert Traffic Class, and MSI vector information on the Core input pins. The Core inserts the Traffic Class into the MSI packet. It also merges the MSI vector number into the MSI packet data field.

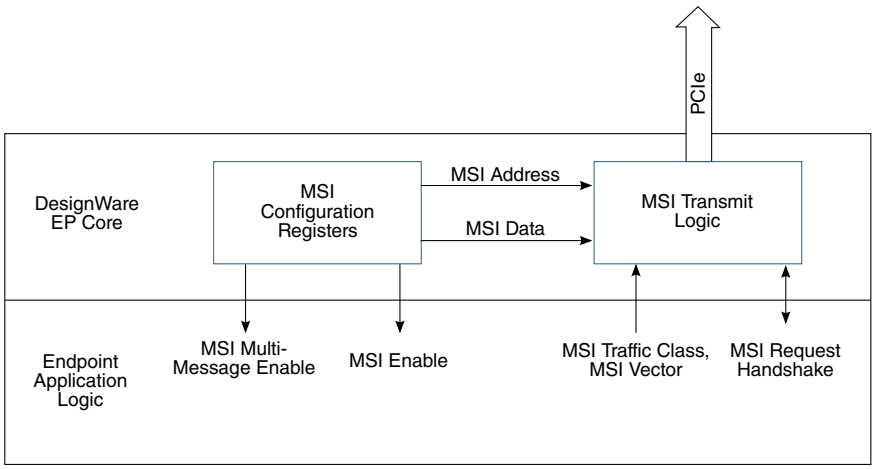


Figure 49-18. MSI Message Passing

When an Endpoint sends an MSI Message, the Message packet is routed by address in the same way as any other PCIe Posted Write packet. In most systems, these packets will be routed to the Root Port and sent to Root Port application logic. The Root Port application logic recognizes the Posted Write packet by its MSI address and handles the interrupt in hardware or software. The data field is also available in the packet to further define the interrupt.

Multifunction Support

Each function in a multi-function device has its own configuration space, and therefore, its own MSI output controls. A multi-function core adds a function number to the MSI inputs shown in the following figure.

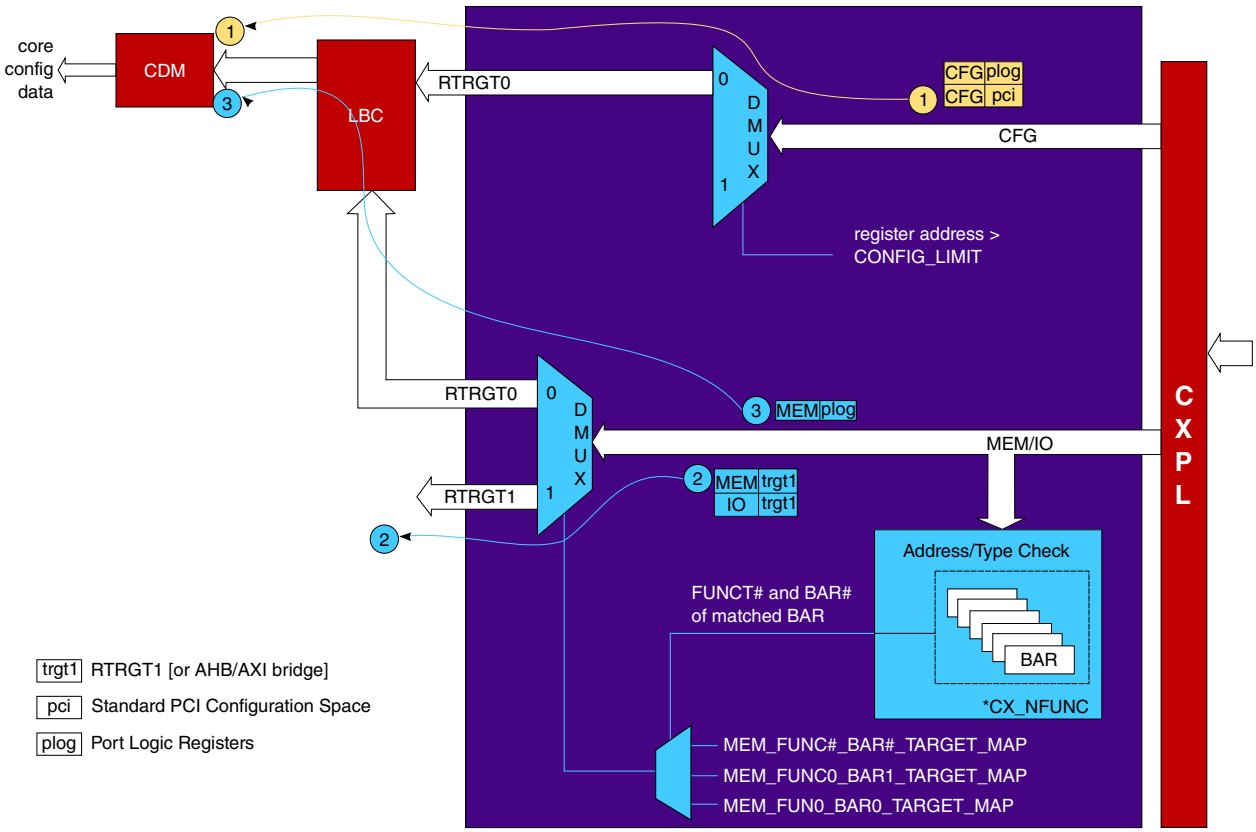


Figure 49-19. Typical use model

Ordering Considerations

You may wish to guarantee that an MSI is sent after a data packet. By setting the MSI packet's Traffic Class to the same value as an earlier data packet, you can guarantee that the MSI packet will always arrive at its destination after the data packet.

NOTE

The PCIe core sets the first byte enable (FBE) to 4'b1111 when it generates an MSI request, even though only the first two bytes of the payload are strictly valid/needed.

49.3.7.1.3 MSI-X

Message Signalled Interrupts (MSI-X) are not messages (Msg/MsgD) but Memory Write (MWr) TLPs. They are indistinguishable from normal MWr's apart from the target address used in conjunction with the MSI-X Capability Structure.

The PCI Express Core automatically builds an MSI packet for your application (if MSI-X is enabled) whenever requested by your application logic.

A simple handshake is required. The MSI-X interface is used only in the DM core in EP mode.

The following MSI interface signals are also used for MSI-X, depending on whether MSI or MSI-X is enabled: ven_msi_req, ven_msi_func_num, ven_msi_tc, and ven_msi_grant.

The address and data fields for MSI-X packets are defined in an MSI-X table located in the Endpoint's application logic, as shown in the figure below. Each entry in the table corresponds to an MSI-X packet that the application may send.

Each MSI-X table entry also includes a mask bit for that interrupt. An additional table, the pending bit array (PBA), includes a pending bit for each MSI-X table entry.

Host software sets and clears mask bits. If an interrupt's mask bit is set when the application wishes to send that MSI-X packet, the application must, instead, set the corresponding entry in the PBA.

Application logic monitors pending bits. If the corresponding mask bit is cleared, then the application sends the corresponding MSI-X packet, and clears the pending bit.

In addition, the MSI-X configuration registers in the Core include:

- Pointers to the function's MSI-X table and PBA
- Length of the function's MSI-X table
- An enable bit for MSI-X for that function
- A global mask for MSI-X

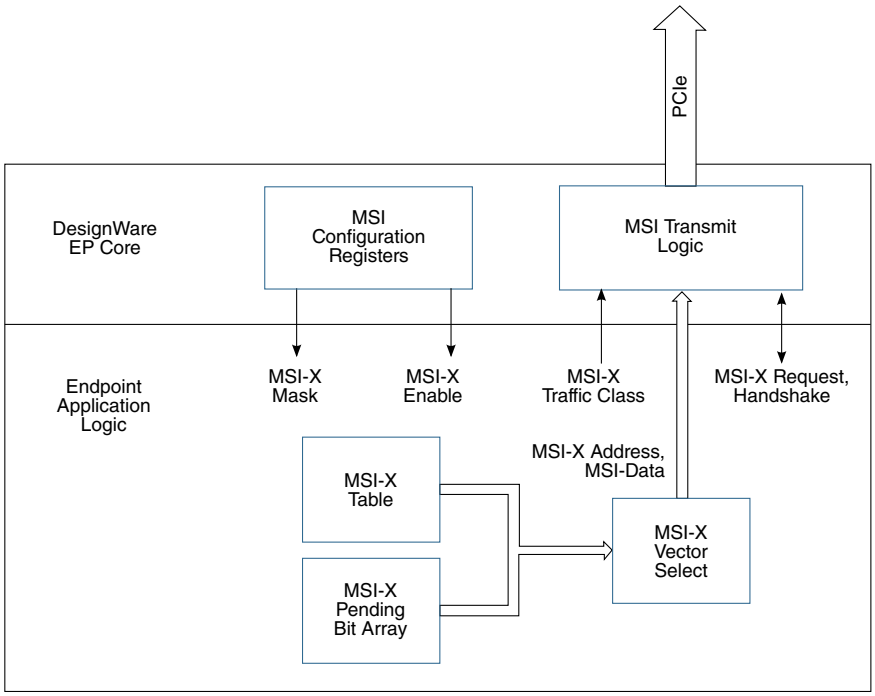


Figure 49-20. MSI-X Message Passing

Large MSI-X Tables

The MSI-X specification is written such that large MSI-X tables and PBAs may be implemented in compiled memories. However, if the host clears the function's global MSI-X mask, a search of the entire 2048 entry PBA might be necessary.

Also, in a large memory array like this, there would be no ìmemoryî of the order in which the pending bits were set. This is not an MSI-X requirement, but might be required by some systems.

49.3.7.2 Interrupts (EP Mode)

In EP mode, the core supports the Legacy PCI INTx compatible interrupt emulation mechanism, the Message Signaled Interrupts (MSI).

A function can use the Legacy PCI INTx mechanism. Different functions of the same device can use different mechanisms. For example, function 0 can use INTA while function 1 uses MSI.

Legacy PCI INTx Support

The core generates two Messages, ASSERT_INTX and DEASSERT_INTX, in response to assertion and the deassertion of the sys_int input. The Interrupt Pin register for each function determines whether the function uses INTA, INTB, INTC, or INTD.

In a single-function configuration, the core uses only INTA. An input signal, sys_int, is provided for the application to notify the core that an interrupt message should be sent. The sys_int signal is level-sensitive on a per-function basis. The rising edge transition of this signal triggers an interrupt assert message. The falling edge transition of this signal triggers an interrupt deassert message.

Message Signaled Interrupt (MSI) Support

MSI support is required for PCI Express devices. MSI-capable devices deliver interrupts by performing Memory Write transactions. The MSI is requested by application logic through the MSI interface; the core then generates the corresponding Memory Write.

49.3.7.3 Interrupts (RC Mode)

In RC mode, the core accepts ASSERT_INTX and DEASSERT_INTX Messages from the downstream component and provides the decoded output signals on the SII (for example, radm_inta_asserted and radm_inta_deasserted).

For interrupts the downstream component transmits through the MSI-X mechanism, the core passes the received Memory Write to the application on the RTRGT1 interface.

PCI Express Hot-Plug Logic Interrupt and Wakeup

In RC mode, the Hot-Plug logic supports generation of Hot-Plug interrupts on the following Hot-Plug events:

- Power Fault Detected
- MRL Sensor Changed
- Presence Detect Changed
- Command Completed
- Attention Button Pressed
- Electromechanical Interlock Status Changed
- Data Link Layer State Changed

When MSI or MSI-X mode is enabled, the core notifies the RC application of Hot-Plug events using the hp_msi output. When INTx interrupt mode is enabled, the core notifies the application of Hot-Plug events using the hp_int output.

If PME is enabled, the The Hot-Plug logic generates a Hot-Plug wake-up signal on hp_pme, triggered by the above Hot-Plug events. The RC Core does not check if the PM state is D1, D2, OR $D3_{HOT}$. It is up to the application to check the value on pm_dstate to make sure the device is in D1, D2, OR $D3_{HOT}$ upon receiving of hp_pme notification.

49.3.7.4 MSI Generation in the AXI Bridge

The standard AXI bridge simply sends MSI requests in the same manner as a memory write.

The method to send a MSI request is to have theAXI bridge send MSI requests in the same manner as a regular memory write. It is the application's responsibility to form the MSI request based on the native PCIe core's configuration. The native PCIe core CDM block contains the MSI address, enable, etc.

The application must obtain the MSI information (cfg_msi_*) from reading the MSI capability registers inthe CDM through the DBI interface, and then form the MSI request to present onto the AXI bridge slave interface.

This method should be used by an application that wants to preserve the order of Posted transfers (MemWr) requested before an MSI request.

The AXI bridge simply sends MSI requests in the same manner as a memory write. To send an MSI to the PCIe link from the AXI bridge, the application presents the MSI address and data onto the AXI slave write channel.

49.3.7.5 MSI Reception in the AHB/AXI Bridge

The standard AHB/AXI bridge receives MSI requests in the same manner as a memory write.

It cannot and does not distinguish between MSI and memory requests. The termination of an MSI request (in RC mode) must be done by the application or by using the optional MSI Controller described next.

49.3.7.5.1 AHB/AXI MSI Controller (Optional in RC mode)

The bridge provides an optional programmable MSI controller to detect and terminate inbound MSI requests in the bridge for RC and DM (RC mode) products. It is enabled by setting the CX_MSI_CTRL_ENABLE) configuration parameter.

Rather than propagating MSI MWr TLPs onto the AHB/AXI bus via the Master interface; the MSI packets are captured and terminated in the AHB/AXI Bridge and an interrupt is signaled.

The MSI Controller is programmed with an address that will be used as the system MSI address. If an inbound (received) MWr request is passed to the AHB/AXI Bridge and matches the specified MSI address as well as the conditions specified for an MSI Memory Write request, then an MSI interrupt is detected. When this Memory Write Request is about to be driven onto the AHB/AXI bridge Master Interface - it is quashed and never appears on the AHB/AXI bus.

The MSI Controller decodes the MSI MWr data payload to determine which End Point device (EP) sent the MSI and which interrupt vector it corresponds to. When a valid interrupt has been decoded, the msi_ctrl_int output is asserted. This output remains asserted when any MSI interrupt is pending. It is only deasserted when there is no MSI interrupt pending.

Features:

- MSI Interrupt Controller only enabled in RC core and DM core in RC mode based on the
- device_type input to DM. It is inactive in EP mode.
- The MSI Interrupt controller will provide support for up to eight EPs. Each supported EP will have a set of Interrupt Enable, Interrupt Mask and Interrupt Status registers.
- A maximum of 32 interrupts are supported per EP.

MSI Request Detection Criteria:

An MSI Interrupt Request is defined to occur when a Memory Write TLP that satisfies the following conditions is received by the core:

- Header Attributes bits are zero. No Snoop (NS) and Relaxed Ordering (RO) must be zero.
- Length field is 0x01 to indicate payload of one DWORD.
- First Byte Enable must be such that it is enabling the first 2 bytes (16-bits) of the payload.
- Last Byte Enable is 4'b0000.
- TLP address corresponds to system's chosen MSI Address as programmed in the MSI Controller Address Register. This register is not the MSI Lower 32 Bits Address Register which is part of the PCI Express MSI Capability Register structure.

In addition to the conditions outlined above, the Memory Write Request must also pass the receive filtering rules as outlined in Receive Filtering to be recognized as a valid MSI Interrupt Request. For example, Poisoned Bit not set in TLP header, ECRC check passed.

49.3.7.5.2 Programming and Usage Model

- The host CPU configures MSI capabilities of all Endpoints (EP) via the local DBI bus (or[1] via Config requests from the remote link partner).
- The MSI data register (MSI Data Register which is part of the PCI Express MSI Capability Register structure) of each EP is programmed as follows to allow the MSI Interrupt Controller to decode the interrupt source.

Table 49-19. MSI Data Register Programming for Use with AHB/AXI MSI Interrupt Controller

15:8	7:5	4:0
Not Used	EP Number •Allows each EP to be identified within the system, •For example, EP#5 is programmed with 3'b101.	Interrupt Vector Number •Identifies the interrupt source within each EP. •Programmed to 5'b00000. •Set by MSI generation logic to identify each interrupt source in real time. •Supports up to 32 Vectors.

- The MSI address register (MSI Lower 32 Bits Address Register which is part of the PCI Express MSI Capability Register structure) of each EP is programmed with the same message address.
- The host CPU configures the MSI Interrupt Controller via the local DBI bus (or via Config requests from the remote link partner).
- The common MSI Address that was used for the EPs is programmed into MSI Controller Address Register in the Interrupt Controller (MSI Controller Address Register).
- The Host CPU reads the MSI capabilities of each EP to determine the number of vectors enabled in each EP and uses this information to program the Interrupt Enable registers in the MSI Interrupt Controller. The Interrupt Enable register allows up to 32 MSI Interrupt Vectors to be enabled within the MSI Interrupt Controller for a given EP. It is the responsibility of the host CPU to read the contents of the "Multiple Message Enable" field in an EP's MSI capability structure and program that EP's Interrupt Enable register in the Interrupt Controller appropriately. For example, if Multiple Message Enable is 3'b100 for Endpoint N, which corresponds to 16 enabled Interrupt Vectors, then Interrupt Enable Register N in the Interrupt Controller should be programmed with 0x0000FFFF by the host CPU.

The MSI Interrupt Controller in your core is active and terminates all received MSI MemWr unless you deactivate it.

NOTE

The MSI Interrupt Controller is deactivated when:

- The core is not in RC mode (applies to DM product only). That is, device_type[3:0] is not 4'b0100.
- All of the eight Interrupt Enable Registers have a value of 0x0.

PROCESSING OF DETECTED INTERRUPTS

The MSI Controller decodes the MSI MWr data payload to determine which End Point device (EP) sent the MSI and which interrupt vector it corresponds to.

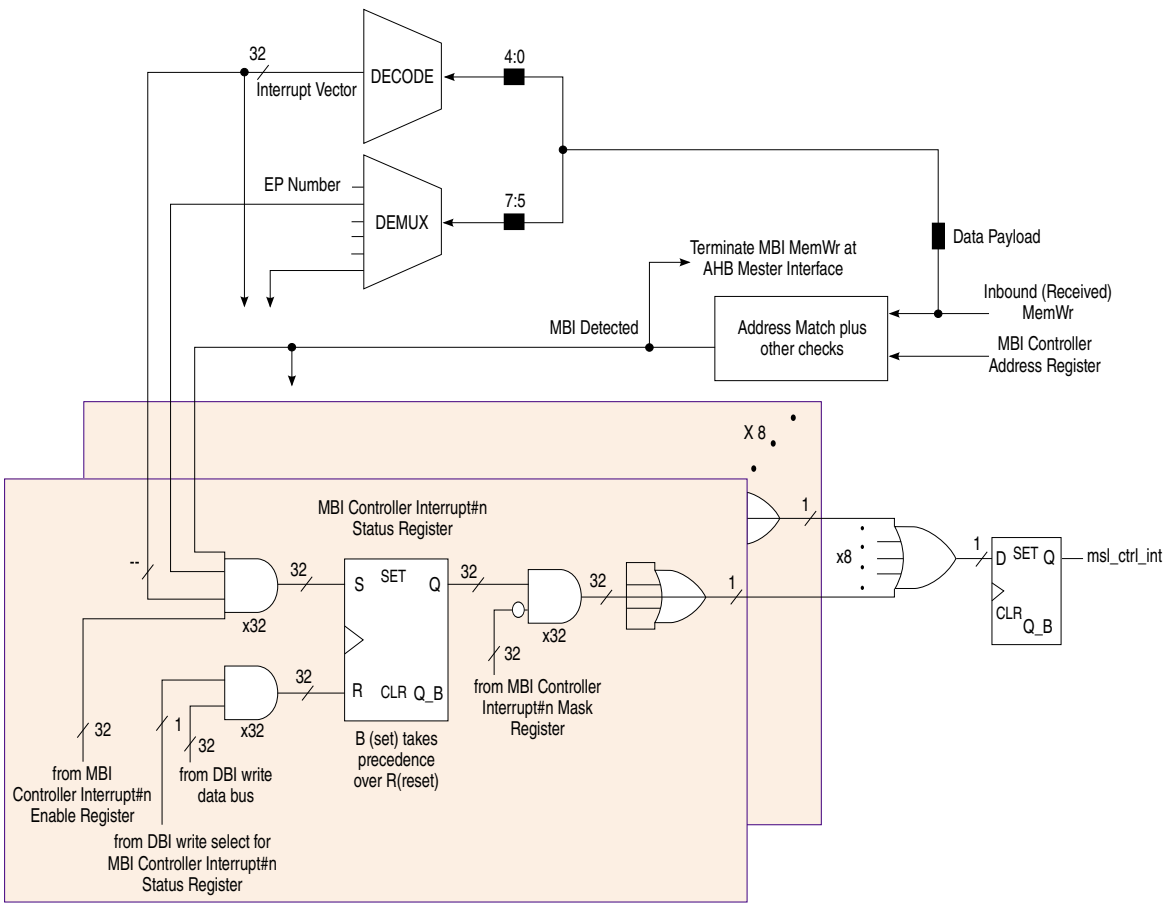


Figure 49-21. Architectural Representation of how the MSI Controller processes detected interrupts.

If the decoded Interrupt Vector is enabled and not masked then the corresponding bit is set in the Interrupt Status register (MSI Controller Interrupt#0 Status Register) and the top-level core output msi_ctrl_int is asserted. This signal remains asserted until the host CPU clears the status bit by writing a 1'b1 to the status bit. (Writing a 1'b0 has no effect). If any status bit remains set then msi_ctrl_int remains asserted. The Interrupt Status register provides a status bit for up to 32 Interrupt Vectors per Endpoint.

If the decoded Interrupt Vector is enabled but is masked then the corresponding bit is set in Interrupt Status register but the top-level core output msi_ctrl_int is not asserted.

If an MSI Interrupt Vector is received from an Endpoint but that Vector has not been enabled in the corresponding Interrupt Enable register (MSI Controller Interrupt#0 Enable Register) then no bit will get set in the Interrupt Status Register and msi_ctrl_int will not be asserted.

In addition, if no interrupts have been enabled in any of the eight Interrupt Enable Registers, then all MSI detection logic is disabled and valid MSI MWr request TLPs are not terminated in the bridge and are passed by the AHB/AXI master interface to the AHB/AXI bus.

The Interrupt Mask register (MSI Controller Interrupt#0 Mask Register) allows the Host to mask a given MSI Interrupt Vector. If a MSI Interrupt Vector is received for a masked Interrupt Vector, the corresponding bit in the Interrupt Status register will get set but msi_ctr_intl will not be asserted as the Interrupt Vector is masked. Note: This masking is local to the MSI Interrupt Controller and is not part of Per Vector Masking (PVM) in any of the downstream Endpoints.

The contents of the Interrupt Mask and Interrupt Status registers are used to drive the msi_ctrl_int output. If any status bit is set and the interrupt vector is not masked, then msi_ctrl_int is asserted HIGH.As long as any Interrupt Status bit is set and not masked, msi_ctrl_int will remain asserted.

ALTERNATIVE DATA REGISTER PROGRAMMING

The programming of the data register (MSI Data Registerwhich is part of the PCI Express MSI Capability Register structure) assumes each Endpoint enables 32 Vectors. If one or more Endpoints only support 16 or less Interrupt Vectors then it is possible to share a single set of Interrupt Enable, Interrupt Mask and Interrupt Status registers among those Endpoints.

For example, assume that Endpoint #1 enables 16 Interrupt Vectors and Endpoint #2 also enables 16 Interrupt Vectors. If the MSI Data register for Endpoint #1 is programmed to 0x20 and the MSI Data Register for Endpoint#2 is programmed to 0x30, then both Endpoint's Interrupt Status will be contained in Interrupt Status Register#1 with Endpoint#1's Vectors in the lower 16 bits and Endpoint#2's Vectors in the upper 16 bits. Manipulating the MSI Data Register format in this manner allows the Interrupt Controller to support more than 8 Endpoints using the 8 Interrupt Status registers provided.

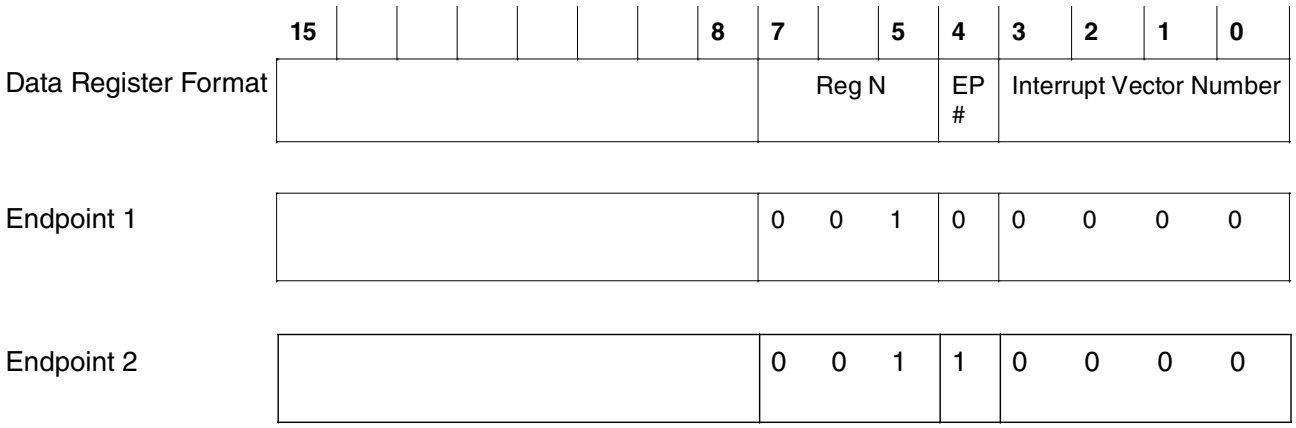


Figure 49-22. Alternative MSI Data Register Format

49.3.8 Flow Control

PCI Express implements a differentiated, credit-based Flow Control system to prevent overflows at the receiver.

In contrast to the simple XON/XOFF type flow control of the Ethernet protocol, the PCI Express Flow Control system requires that the consumer of data advertise the available buffer space for each type and priority of traffic. The Flow Control mechanism is divided into two phases: the initialization phase and the update phase. The core automatically performs both of these phases with minimal support required from the application.

The Flow Control for VC0 must be initialized following Link initialization, but prior to sending normal traffic. This initialization is performed by the Flow Control Initialization state machine. The initialization process involves exchanging information with the Link partner about the size of the receiver's buffers for each type of packet data: Posted, Non-Posted, and Completion. Header and payload buffers for each of these types are tracked and reported independently. Flow Control must be initialized for Virtual Channel 0 (VC0) before the Data Link Layer's link state moves into the DL_ACTIVE state, and normal traffic can begin flowing. Additional VCs (if any) are initialized following this, intermingled with the regular traffic already flowing on VC0. Initialization of other VCs begins when the VCs are enabled. The VC0 traffic has priority over non-VC0 flow control initialization.

The core provides a configurable solution to choose the number of credits to advertise per type and per VC. It can be configured to support multiple VCs as well as infinite credits. The core performs all required Flow Control protocol handshakes. The core currently provides a solution in which the application does not have to deal with Flow Control

updates and checks. By default, the RADM is responsible for returning Flow Control credits as data is read out of the RADM queuing structure. Additionally, the core provides optional signals to enable the application to handle Flow Control returns in an application-specific manner

The core does not return Flow Control credits for packets that have Data Link Layer errors.

49.3.9 Address Translation

This is a local address translation implementation and is not related to the PCI-SIG ATS specification support.

If address translation support is enabled for the core, it can use an address translation unit (ATU) to replace the TLP address and TLP header fields in the current TLP request header.

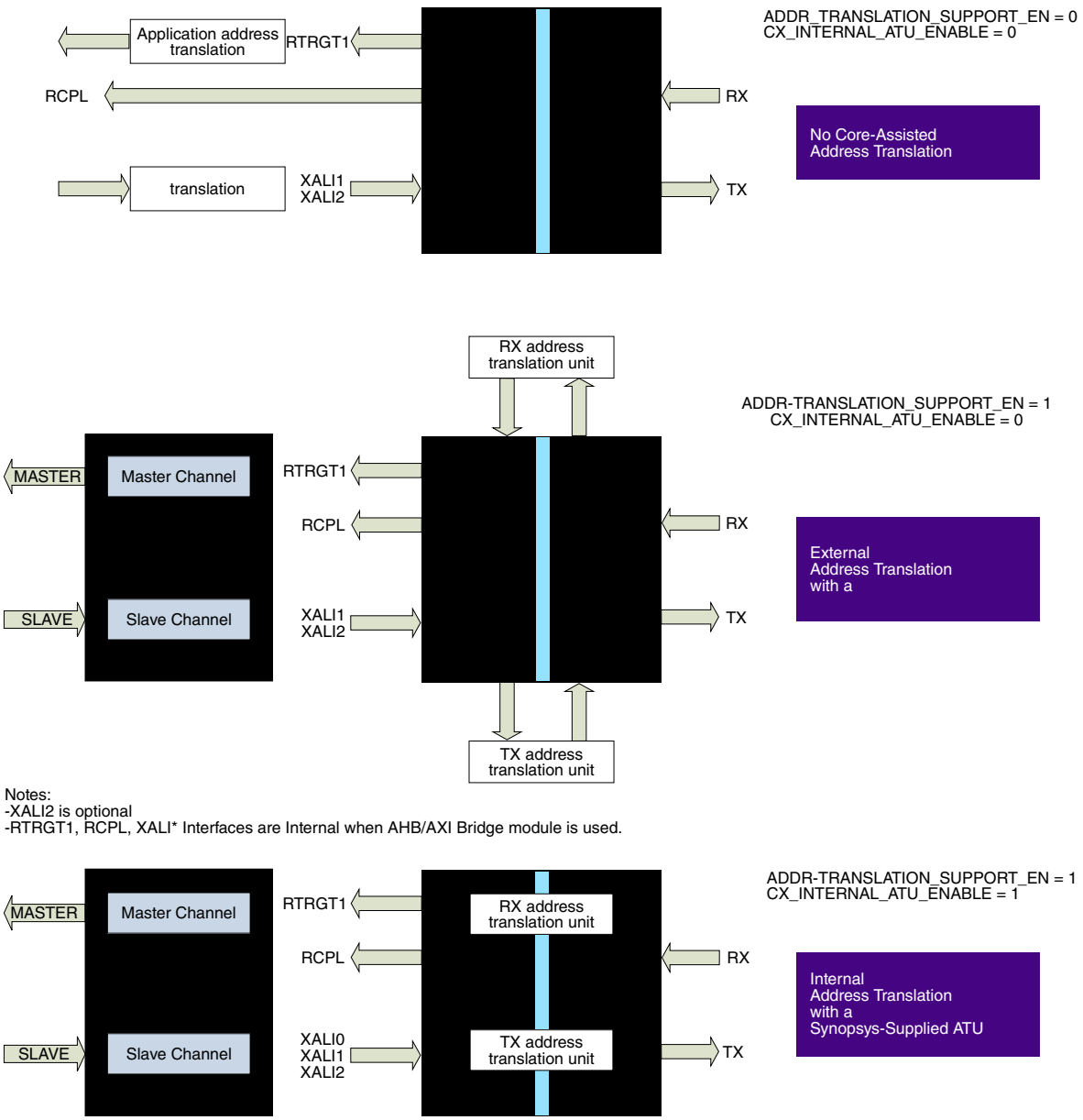


Figure 49-23. Address Translation options

Address translation is used for mapping different address ranges to different memory spaces supported by the application. A typical example will map the AMBA memory space to PCI memory space when the application has the PCIe-to-AMBA bridge in place. TYPE translation is also supported.

Without address translation, the application address is passed to/from the PCIe TLPs directly through AXI bridge interfaces.

The PCIe core supportsaddress translation.

1. Internal Address Translation (iATU) instantiates an internal ATU (iATU) inside the PCIe core. It can be configured (by software) to implement a customer-defined address translation scheme without the need for additional hardware from the customer. To enable this option:

Table 49-20. Configuration Parameters Relevant to Address Translation

Parameter Name	Parameter Label	Value Range	
CX_ATU_NUM_OUTBOUND_REGIONS	Number of Outbound Address Translation Regions	4,	
CX_ATU_NUM_INBOUND_REGIONS	Number of Inbound Address Translation Regions	4,1	
CX_ATU_MIN_REGION_SIZE	Minimum Size of Address Translation Region	64 kB	

NOTE

The easiest and recommended method of implementing address translation is to use the Internal Address Translation Unit (iATU).

49.3.9.1 Internal Address Translation (iATU)

An internal ATU (iATU) is instantiated inside the PCIe core.

It can be configured (by software) to implement a customer-defined address (and TYPE/ FORMAT) translation scheme without the need for additional external hardware.

49.3.9.1.1 Outbound (TX) Features

- Address Match Mode operation for MEM/IO/CFG/MSG TLPs. No address translation for CPL.
- Supports TYPE translation via TLP TYPE header field replacement for MEM[1] types to MSG/CFG types.
- This includes translation from Posted to Non-Posted (for example, MWr to CfgWr0).
- No TYPE translation from CPL TLPs.
- Programmable TLP header per region for the following fields for TLP field *replacement*.
- TYPE / TD / TC / AT / ATTR / MSG Code
- Function Number (Physical and Virtual).
- 4 Address Regions based on programmable registers for location and size.
- Programmable enable/disable per region.

- Automatic format (FMT) field translation between 3 DW and 4 DW for 64-bit addresses.
- Invert Address Matching Mode to translate accesses outside of a successful address match.
- ECAM Configuration Shift Mode to allow a 256 MB CFG1 space to be located anywhere in the 64-bit address space.
- Can be used to replace usage of misc sideband AXI slave bus signals.
- Supports regions from 64kB to 4 GB in size.
-

49.3.9.1.2 Inbound (RX) Features

- Address Match Mode operation for MEM/IO/CFG/MSG TLPs. No address translation for CPL. Selectable BAR Match Mode operation for IO/MEM TLPs.
- TLPs destined for RTRGT0 (internal CDM or ELBI) will not be translated.
- TLPs that are not error-free (ECRC, malformed and so on) will not be translated.
- Programmable TLP header per region for the following fields for *matching*.
- TYPE / TD / TC / AT / ATTR / MSG Code
- Function Number (Physical and Virtual).
- Up to 4 Address Regions based on programmable registers for location and size.
- Programmable enable/disable per region.
- Automatic format (FMT) field translation between 3 DW and 4 DW for 64-bit addresses.
- Invert Address Matching Mode to translate accesses outside of a successful address match.
- Configuration Shift Mode. Optimizes the memory footprint of CFG accesses destined for the AXI bridge interface in multi-function devices.
- Supports cores with and without the AHB/AXI Bridge module.
- Response Code defines the CPL completion status to return for accesses matching a region.
- Supports regions from64 kB to 4 GB in size.

49.3.9.1.3 Programming (iATU)

The iATU registers are in the PCIe cores' port logic register space (See PCIe CTRL Port Logic Memory Map/Register Definition). This may be accessed locally via the DBI interface or via PCIe Configuration accesses.

The following registers are used for programming the iATU.

Table 49-21. iATU Register Map

Byte Offset	Description

Table continues on the next page...

Table 49-21. iATU Register Map (continued)

+0x200	iATU Viewport Register
+0x204	iATU Region Control 1 Register
+0x208	iATU Region Control 2 Register
+0x20C	iATU Region Lower Base Address Register
+0x210	iATU Region Upper Base Address Register
+0x214	iATU Region Limit Address Register
+0x218	iATU Region Lower Target Address Register
+0x21C	iATU Region Upper Target Address Register

Full descriptions of each register are available at PCIe CTRL Port Logic Memory Map/ Register Definition.

49.3.9.2 Outbound iATU Operation

Information found here describes the processing of outbound requests by the iATU.

The topics for this section are:

- RID BDF Number Replacement
- iATU Outbound MSG Handling
- CFG Handling
- CFG Shift Feature
- FMT Translation
- Invert Feature
- No Address Match Result
- Writing to a MRdLk Region
- Programming Example

OVERVIEW (ADDRESS MATCH MODE)

The address field of each request MEM/IO TLP is checked to see if it falls into any of the enabled[1] address regions defined by the 'Start' and 'End' addresses. If an address match is found, then the TLP address field is modified as follows:

Address = Address - Base Address + Target Address

and the TYPE, TD, TC, AT and ATTR TLP header fields are replaced with the corresponding fields in iATU Control 1 Register.

If the application (AXI) address field matches more than one of the CX_ATU_NUM_OUTBOUND_REGIONS address regions, then the first (lowest of the numbers from 0 to CX_ATU_NUM_OUTBOUND_REGIONS-1) enabled region to be matched is used.

If there is no address match, then the address is untranslated. In the outbound direction (only), the TLP header information (for fields that are programmable) will come from the relevant fields on the AXI Slave Interface sideband busses (all fields will be 0).Since all misc_info are tied to 0, this is not recommended.

This operational mode (called Address Match Mode) is always used for outbound translation.

1. If the 'Region Enable' bit of the 'Region Control 2 Register' is '0', then that region is not used for address matching.

When the PCIe core is operating with 32-bit addresses, the operation is defined as in the following figure.

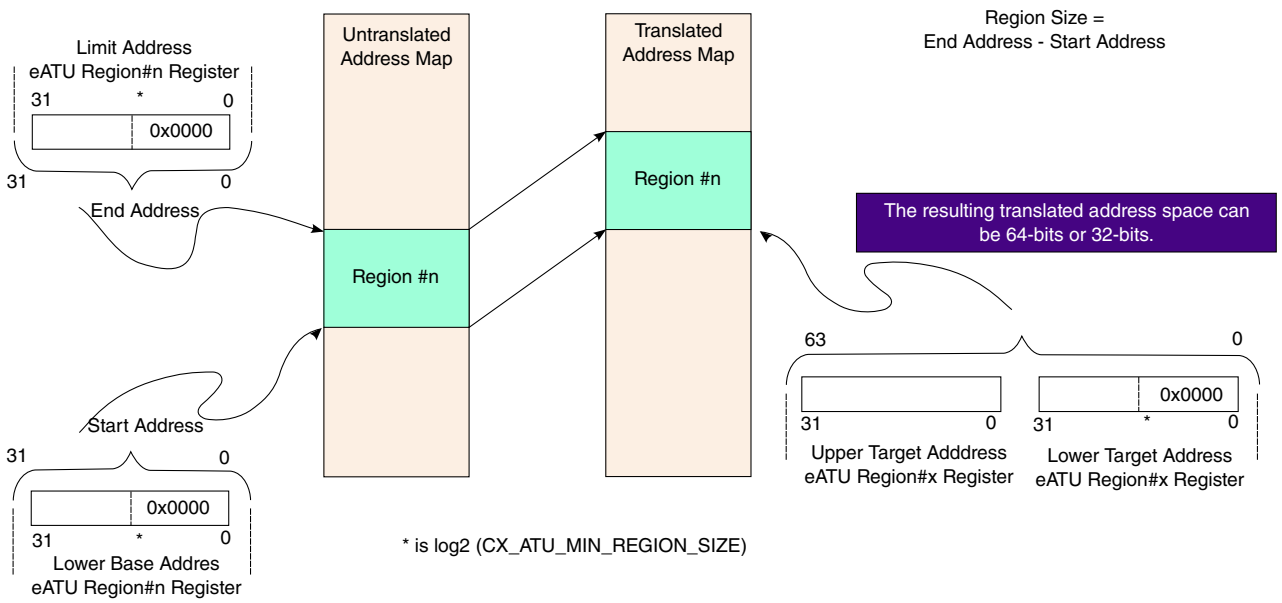


Figure 49-24. iATU Address Region Mapping: Outbound and Inbound (Address Match mode): 32-bit Address

The upper 32 bits of the Target Address Register will always form the upper 32 bits of the translated address because:

- The maximum region size is 4 GB.
- A region may not cross a 4 GB boundary.

The CX_ATU_MIN_REGION_SIZE (64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(CX_ATU_MIN_REGION_SIZE) bits are zero.

RID BDF NUMBER REPLACEMENT

When there is a successful address match on an outbound TLP, then the function number used in generating the 'Function' part of the Requester ID (RID[1]) field of the TLP is taken from the 3-bit 'Function Number' field of the iATU Control 1 Register. The value in this field must be 0x0 unless MultiFunction operation in the core is enabled (CX_NFUNC > 1).

iATU OUTBOUND MSG HANDLING

The iATU supports TYPE translation/conversion of MEM[4] TLP's to Msg/MsgD TLPs. This supports applications that are unable to generate Msg/MsgD type TLPs natively.

When there is a successful address match on an outbound MEM TLP, and the translated TLP TYPE field is 'Message' (that is, 'TYPE' field of the iATU Control 1 Register is 10xxx); then the Message Code field of the TLP is set to the value in the 'Message Code' field of the iATU Control 2 Register.

A MWr with an 'effective length of 0' (see [5]) is converted to Msg and all other MWr TLPs are converted to MsgD.

NOTE

For more information on generating Messages, see Message Generation.

For a proper understanding of Messages, you should be familiar with the PCI Express Base Specification.

MSG translation is possible with the iATU. For MSG transactions created directly by the application (as opposed to, by the iATU) you must ensure that the 3rd and 4th DWORD (In the PCI Express Base Specification) does not match any programmed iATU address region or else unintentional translation (TYPE) could occur.

RID uses the 8-bit.5-bit.3-bit PCI Bus.Device.Function (BDF) format.

The maximum size of this field is 8 bits, but the actual size depends on the number of Virtual functions (VFs) used as denoted by 2^NVF_WD Or IO.

For AXI, this takes the First and Last byte enables into account. For AXI this is through slv_wstrb. If you are just translating the address of MSG TLPs, then client0_tlp_byte_en is used to provide the Message Code.

CFG HANDLING

Outbound CFG transactions (formed by translation of IO/MEM TLPs from the application) can exist anywhere in address space -because the Routing ID or BDF, is created by the iATU from bits [31:16] of the untranslated address - and this BDF changes according on the PCIe bus topology.

In the normal untranslated transmission of CFG transactions, the PCIe core derives the CFG Bus.Device.Function (BDF) information from the address on the AXI Slave Interface address) as follows.

Table 49-22. Normal PCIe Core Outbound Derivation of BDF from XALI* or AHB/AXI Slave Interfaces

XAXI Slave Interface Bits	PCIe CFG Header Field
31:24	Bus Number
23:19	Device Number
18:16	Function Number
11:8	Extended Register Number
7:2	Register Number

The iATU supports translation (of address or type) of IO/MEM TLP's to CFG TLPs. This supports applications that are unable to generate CFG type TLPs natively.

The 16-bit Bus.Device.Function (BDF) is derived from bits [31:16] of the iATU Lower Target Address Register.

CFG translation is possible with the iATU. For CFG transactions created directly by the application (as opposed to, by the iATU) you must ensure that the Bus.Device.Function field does not match any programmed iATU address region or else unintentional translation (TYPE) could occur.

CFG SHIFT FEATURE

A expander feature (CFG Shift Feature) can be enabled by setting the 'CFG Shift' bit of the iATU Control 2 Register.

This shifts/maps the BDF - bits [27:12] of the Target Address up to bits [31:16] of the translated address. This allows all outgoing IO/MEM TLPs (that have been translated to CFG) to be mapped into any 256 MB region of the PCIe address space.

This scheme also supports the Enhanced Configuration Address Mapping (ECAM) mechanism from the *PCI Express, Revision 3.0 Base Specification.* ECAM supports the mapping (via MEM to CFG TYPE translation) from memory address space to PCI Express Configuration Space address. ECAM maps bits [27:12] of the untranslated MEM TLP to become the BDF of the resulting CFG TLP.

Table 49-23. ECAM Scheme from PCIe Specification using an 8-bit BDF Bus Number

Memory Address Bits	PCIe Configuration Space
27:20	Bus Number
19:15	Device Number
14:12	Function Number
11:8	Extended Register Number
7:2	Register Number

FMT TRANSLATION

The iATU automatically sets the TLP format (FMT) field for 3DW when it detects all zeroes in the upper 32- bits of the *translated* address. Otherwise, it sets it to 4DW when it detects a 64-bit address (that is, when there is a '1' in the upper 32-bits of the translated address). If the original address and the translated address are of different format, the iATU ensures that the TLP header size matches the translated address format.

INVERT FEATURE

Normally, an address match on an outbound TLP occurs, occurs when the untranslated address is in the region bounded by the Base Address and Limit Address.

When the Invert feature is activated, an address match occurs when the untranslated address is NOT in the region bounded by the Base Address and Limit Address.

This feature is activated by setting the 'Invert' field of the iATU Control 2 Register.

NO ADDRESS MATCH RESULT

When there is no address match, then the address is untranslated but

the TLP header information will come from the relevant fields on the AHB/AXI Slave Interface sideband busses (all fields will be 0).

WRITING TO A MRDLK REGION

When there is a successful address match for an outbound WRITE, and the TYPE header field - as replaced with the TPYE field in iATU Control 1 Register - is MRdLk, then the TYPE header field will be set to MEM (that is 00000b).

PROGRAMMING EXAMPLE

NOTE

Define Outbound Region 1 as:

IO region from 0x80000000_d000000 - 0x80000000_d000ffff (64k) mapped to 0x00010000 in PCIe IO space.

1. Setup the Viewport Register

Write 0x00000001 to Address { 0x700 + 0x200 } to set outbound region 1 as the current region

2. Setup the Region Base and Limit Address Registers

Write 0xd0000000 to Address {0x700 + 0x20C} to set the Lower Base Address. Write 0x80000000 to Address {0x700 + 0x210} to set the Upper Base Address. Write 0xd000ffff to Address {0x700 + 0x214} to set the Limit Address

3. Setup the Target Address Registers

Write 0x00010000 to Address {0x700 + 0x218} to set the Lower Target Address

Write 0x00000000 to Address {0x700 + 0x21C} to set the Upper Target Address

4. Configure the region via the Region Control 1 Register

Write 0x00000002 to Address {0x700 + 0x204} to define the type of the region to be IO.

5. Enable the region

Write 0x80000000 to Address {0x700 + 0x208} to enable the region.

49.3.9.2.1 Inbound iATU Operation

This section describe the processing of inbound requests by the iATU. The topics for this section are:

- Overview
- IO/MEM Match Modes
- CFG Handling
- Optional Matching Fields
- Response Code Feature
- iATU Inbound MSG Handling
- Fuzzy Type Match Mode
- FMT Translation
- Invert Feature
- Programming Examples

49.3.9.3 Overview (iATU)

The main difference between Inbound and Outbound iATU operation is that the TLP TYPE is never changed in the Inbound direction.

Instead, the TYPE field is used for more precise matching. Other fields may also be optionally used to further refine the matching process.

Another difference is that for MEM/IO TLPs, you can select between Address matching (as used in Outbound Operation) or BAR matching. Normally an End Point (EP) will use BAR match mode and a Root Complex (RC) will use Address mode as an RC normally has no BAR's implemented.

Lastly, for CFG0 TLPs, you can select between Routing ID matching or Accept mode.

If there is no match then the address is untranslated. In addition,

- TLPs destined for RTRGT0 (internal CDM or ELBI) will not be translated.
- TLPs that are not error-free (ECRC, malformed and so on) will not be translated.
- Address translation of all TLP types (MEM/IO/CFG/MSG) except CPL is supported in Address Match mode. In BAR Match mode only translation of IO/MEM is supported.

IO/MEM MATCH MODES

Inbound Address translation for IO/MEM TLPs will operate in one of two matching modes as determined by the 'Inbound Match Mode' field in the iATU Region Control 2 Register.

1. Address Match Mode

The operation is similar to Outbound iATU Operation.

The address field of each request TLP is checked to see if it falls into any of the enabled[1] address regions defined by the 'Start' and 'End' addresses. If an address match is found, then the TLP address field is modified as follows:

Address = Address - Base Address + Target Address

If the TLP address field matches more than one of the CX_ATU_NUM_INBOUND_REGIONS address regions, then the first (lowest of the numbers from 0 to CX_ATU_NUM_INBOUND_REGIONS-1) enabled region to be matched is used.

2. BAR Match Mode

Looking for an address match is a two-step process.

1. The address field of MEM/IO *(only)* request TLPs is checked by the standard internal *PCI Express BAR Matching Mechanism* to see if it falls into any address region defined by the enabled BAR Addresses and Masks.

2. If a matched BAR was found, then that matched BAR ID is compared by the iATU to the 'BAR Number' field in the iATU Region Control 2 Register for all enabled regions.

BAR Match Mode can only be used for MEM/IO transactions.

Address Match Mode should always be used to match MSG transactions as these will never generate a match against a BAR.

1. If the 'Region Enable' bit of the 'Region Control 2 Register' is '0', then that region is not used for address matching.

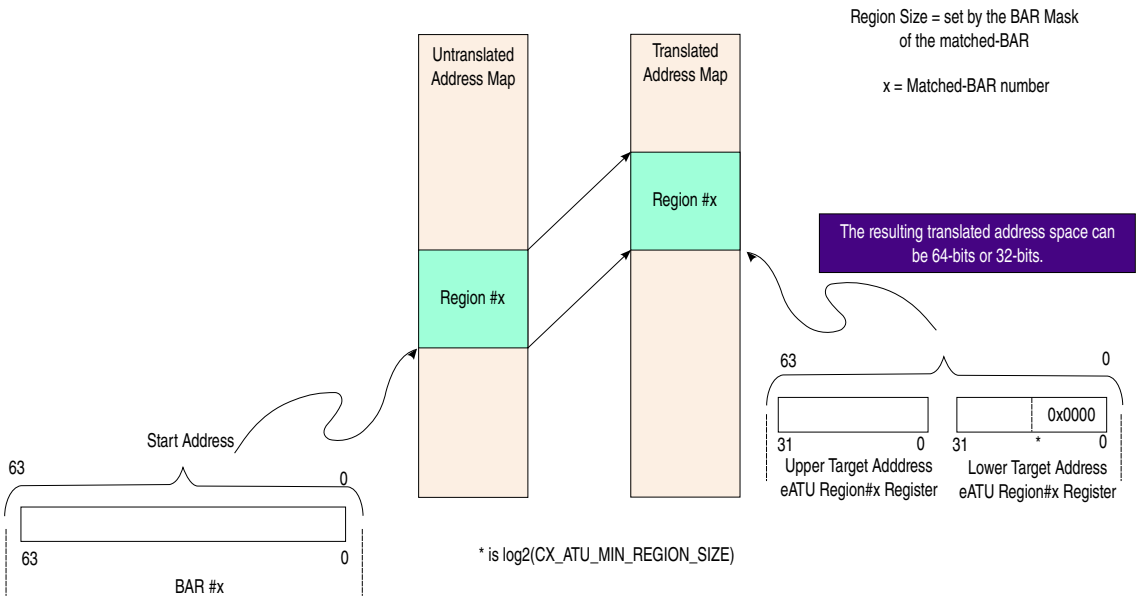


Figure 49-25. iATU Address Region Mapping: Inbound (BAR Match mode): 64-bit BAR

Normally an EP will use BAR match mode and an RC will use Address Match mode -as an RC normally has no BAR's implemented or at least must handle requests which do not match any of its BARs.

However, the user has the freedom to implement any mode in their device. For example, an EP device may use Address Match mode, but should be aware that if the address range does not match one of its BAR ranges in an EP, the device will reject the request with Unsupported Request (UR) completion status and no translation will occur.

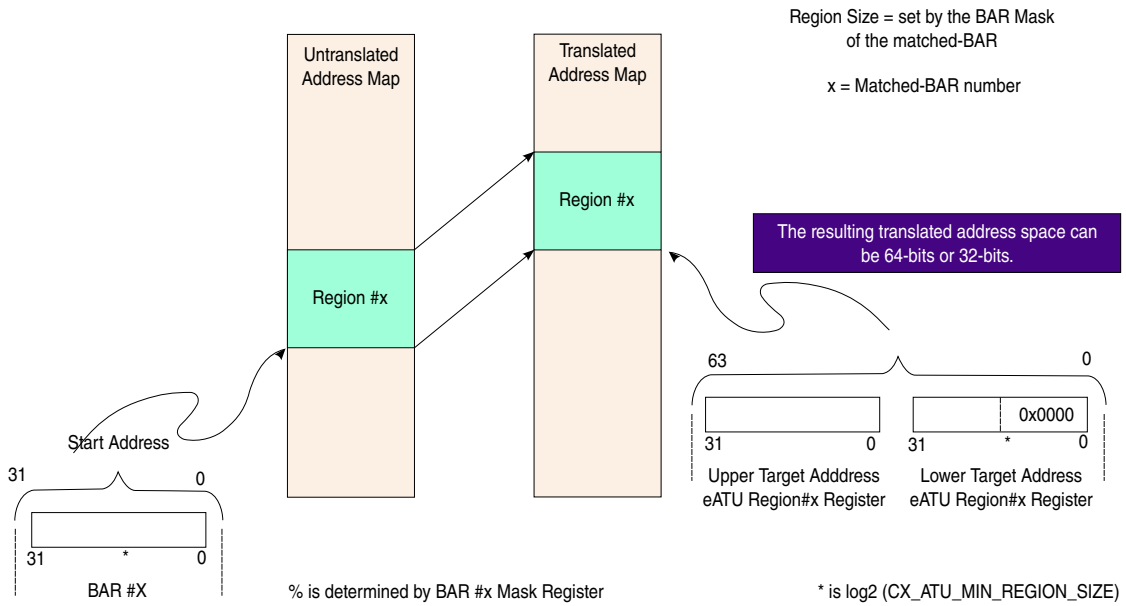


Figure 49-26. iATU Address Region Mapping: Inbound (BAR Match mode): 32-bit BAR

CFG HANDLING

CFG TLPs are normally routed to internal CDM. These will not be translated. Only CFG0 TLPs routed to AXI Bridge Master interface will be translated.

Inbound Address translation for CFG0 TLPs will operate in one of two matching modes as determined by the 'Inbound CFG0 Match Mode' field of the iATU Region Control 2 Register.

1. Routing ID Match Mode:

The operation is similar to outbound iATU operation. The Routing ID of the inbound CFG0 TLP must fall within the Base and Limit of the defined iATU region for matching to proceed. The iATU interprets the Routing ID (Bytes[1] 8 to 11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM/IO transactions.

2. Accept Mode:

CFG0 TLPs should always be accepted and processed even if the bus number does not match the current Bus number of the device. This mode follows that behavior. The Routing ID of received CFG0 TLPs will be ignored when determining a match.

CFG1 TRANSACTIONS

For CfgRd1/CfgWr1 transactions the Base and Limit Address could enclose the entire 32-bit 4G memory space with Routing ID forming the upper 16 bits. The Target Address maps these CFG transactions to anywhere in application address space.

CFG SHIFT FEATURE

Inbound CFG transactions (routed for AXI Bridge Master) can exist anywhere in address space -because the Routing ID or BDF, is processed by the PCIe core RADM filter (see Receive Application-Dependent Module (RADM)) as bits [31:16] of an address - and this BDF changes according on the PCIe bus topology.

A compressor feature (CFG Shift Feature) can be enabled by setting the CFG Shift bit of the iATU Region Control 2 Register. Bits [15:12] of the 3rd DWORD[1] of CFG TLPs are reserved. The compressor feature uses this to reduce the memory requirement.

This shifts/maps the BDF - bits [31:16] of 3rd header DWORD which would be matched against the Base and Limit Addresses - of the incoming CfgRd0/CfgWr0 down to bits [27:12] of the translated address.

OPTIONAL MATCHING FIELDS

In Address or BAR Match mode, a successful address/BAR match can be optionally gated by successful matching of the following programmable TLP header fields (per region):

- TYPE / TD / TC / AT / ATTR
- MSG Code (MSG TLP's only)
- Function Number (MEM, IO or CFG TLPs only)
- Virtual Function Number (MEM or IO TLPs only)

For each of the above fields in the iATU Region Control 1 Register (ATU Region Control 2 Register for MSG) there is an associated 'Match Enable' bit in the iATU Region Control 2 Register. Address translation will only proceed if all enabled field-matches are successful.

RESPONSE CODE FEATURE

When the 'Response Code' field of the inbound iATU Region Control 2 Register is set to a value other than 00b, it will determine the Completion Status of the CPL TLP sent in response to a successfully matched Non Posted TLP. This can be set to Unsupported Request (UR) or Completer Abort (CA). When the error response field is set to 00b, then the normal RADM (see Receive Application-Dependent Module (RADM)) filter response for this TLP will be used.

IATU INBOUND MSG HANDLING

Inbound Message (Msg/MsgD) transactions can use one of two matching modes:

Address Match Mode: The 3rd and 4th header DWORDs are treated as an address and matched against the iATU Region Base and Limit Address registers. Furthermore, for Vendor Defined messages, this allows specific vendor defined messages to be filtered

into memory at the Target Address. The Upper Base Address should be set to Bus.Device.Function (BDF) and Vendor ID. The Lower Base Address can be used as a filter for specific messages.

Vendor ID Match Mode: This mode is relevant for ID-routed Vendor Defined Messages. The iATU ignores the Routing ID (Bus, Device, Function) in bits [31:16] of the 3rd DWORD of the TLP header[1], but matches against the Vendor ID in bits [15:0] of the 3rd DWORD of the TLP header (bytes[1] 10 and 11). This allows Vendor defined messages to be filtered against specific Vendor IDs without needing to know the BDF number which may vary depending on the PCI topology.

Bits [15:0] of the Region Upper Base register should be programmed with the required Vendor ID as follows:

- Region Upper Base[15:8] = byte 10
- Region Upper Base[7:0] = byte 11

The lower Base and Limit Register should be programmed to translate TLPs based on vendor specific information in the 4th DWORD of the TLP header.

NOTE

For more information on generating Messages, see Message Generation. For a proper understanding of Messages you should be familiar with Message Request Rules of the PCI Express Base Specification.

FUZZY TYPE MATCH MODE

When enabled, the iATU relaxes the matching of the TLP TYPE field against the expected TYPE field so that

- CfgRd0 and CfgRd1 TLPs are seen as identical. Similarly with CfgWr0 and CfgWr1.
- MRd and MRdLk TLPs are seen as identical
- The Routing field of MsgD TLPs is ignored

For example, CFG0 in the TYPE field in the iATU Control 1 Register will match against an inbound CfgRd0, CfgRd1, CfgWr0 or CfgWr1 TLP.

To enable this feature, then set the 'Fuzzy Type Match Mode' bit of the iATU Region Control 2 Register.

FMT TRANSLATION

The iATU automatically sets the TLP format (FMT) field for 3DW when it detects all zeroes in the upper 32- bits of the *translated* address. Otherwise, it sets it to 4DW when it detects a 64-bit address (that is, when there is a '1' in the upper 32-bits of the translated address). If the original address and the translated address are of different format, the iATU ensures that the TLP header size matches the translated address format.

INVERT FEATURE

Normally, an address match on an outbound TLP occurs, occurs when the untranslated address is in the region bounded by the Base Address and Limit Address.

When the Invert feature is activated, an address match occurs when the untranslated address is NOT in the region bounded by the Base Address and Limit Address.

This feature is activated by setting the 'Invert' field of the iATU Region Control 2 Register.

PROGRAMMING EXAMPLES

Define Inbound Region 2 as: MEM region matching BAR4 (BAR Match mode) mapping to 0x8000000020000000 in the application memory space

1. Setup the Viewport Register
   - Write 0x80000002 to Address { 0x700 + 0x200 } to set inbound region 2as the current region
2. Setup the Target Address Registers
   - Write 0x20000000 to Address {0x700 + 0x218} to set the Lower Target Address
   - Write 0x80000000 to Address {0x700 + 0x21C} to set the Upper Target Address
3. Configure the region via the Region Control 1 Register
   - Write 0x00000000 to Address {0x700 + 0x204} to define the type of the region to be MEM.
4. Enable the region for BAR Match Mode
   - Write 0xC0000400 to Address {0x700 + 0x208} to enable the region for BAR match mode for BAR#4.

Define Inbound Region 0 as: MEM region matching TLPs with addresses in the range 0x00010000 - 0x0005ffff mapped to 0x1000000020000000 - 0x100000002004ffff in the application memory space

1. Setup the Viewport Register
   - Write 0x80000000 to Address { 0x700 + 0x200 } to set inbound region 0 as the current region
2. Setup the Region Base and Limit Address Registers
   - Write 0x00010000 to Address {0x700 + 0x20C} to set the Lower Base Address.
     Write 0x00000000 to Address {0x700 + 0x210} to set the Upper Base Address.
     Write 0x0005ffff to Address {0x700 + 0x214} to set the Limit Address

3. Setup the Target Address Registers
   - Write 0x20000000 to Address {0x700 + 0x218} to set the Lower Target Address
   - Write 0x10000000 to Address {0x700 + 0x21C} to set the Upper Target Address
4. Configure the region via the Region Control 1 Register
   - Write 0x00000000 to Address {0x700 + 0x204} to define the type of the region to be MEM.
5. Enable the region
   - Write 0x80000000 to Address {0x700 + 0x208} to enable the region in address match mode.
   - EP: Defined MEM or IO regions must be inside an enabled BAR range.
   - RC: Defined MEM or IO regions must either match a BAR or be outside of the base and limit ranges defined for the port in the Type 1 configuration header.

49.3.10 Gen2 5.0 GT/s Operation

The PCIe Express core supports all of the non-optional Gen2 5.0 GT/s features defined in the *PCI Express 3.0 Specification.*.

49.3.10.1 Overview (Gen2 5.0 GT/s)

The DWC PCIe cores support achives the PCI Express 2.0 rate, by DYNAMIC

FREQUENCY

1. When supporting Gen2 DYNAMIC FREQUENCY, the core operates at either 125 MHz at the Gen1 rate. When operating at the Gen2 rate, the core's clock frequency is changed to 250 MHz

Software configuration of Gen2 5.0 GT/s operation is available through the Gen2 Control Register.

49.3.10.2 Speed Changing

If bit 17 Directed Speed Change of the Gen2 Control Register is set to '1', then the LTSSM will initiate a speed change after the link is initialized. The default value of this register is '1'.

A PIPE signal, mac_phy_rate, is used to negotiate the Link data rate. The core drives mac_phy_rate to indicate the negotiated Link data rate. For mac_phy_rate, a value of zero indicates 2.5 Gbps, and a value of one indicates 5.0 Gbps. The PCIe core changes the rate signal and waits for a pulse on the phy_mac_phystatus signal to confirm that the PHY has accepted the requested rate.

The PCIe core uses core_clk to sample the phy_mac_phystatus signal during speed changes. The PHY generates phy_mac_phystatus based on pipe_clk. Therefore, the following restrictions are placed on DWC_pcie external logic when the PHY is Gen2 Dynamic Frequency and the PCIe core is Gen2 Dynamic Width (i.e. when core_clk is not equivalent to pipe_clk).

1. ensure that core_clk is toggling when the PHY returns phy_mac_phystatus for a speed change

OR

2. the external logic must hold phy_mac_phystaus (for the speed change) until core_clk toggles again

For other configurations, the core_clk is equivalent to the pipe_clk, and this restriction is not needed.

49.3.11 Power Management

An architectural overview of the Power Management Controller is given in "Power Management Controller (PMC)".

There are two types of power management operations:

- Software controlled PCI power management operations
- Active state power management operation (ASPM) for PCIe device only

The following figure shows the capable link state and its control conditions.

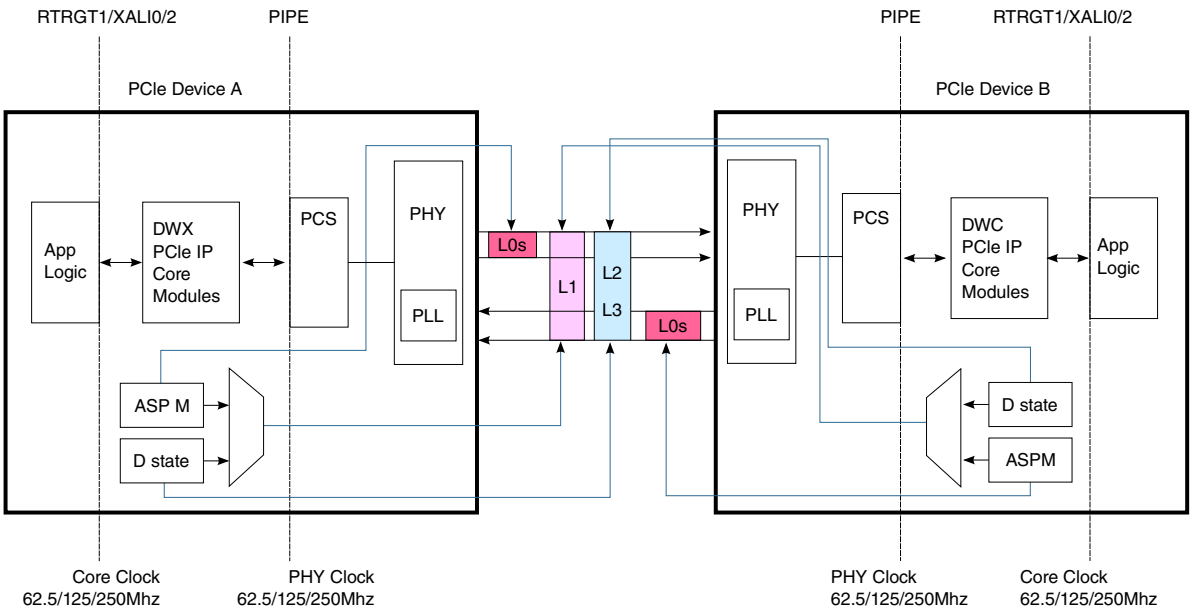


Figure 49-27. Power Management - Capable Link State and Control Conditions

The L0s link state is controlled by the ASPM L0s enter condition met state. The L1 link state is controlled either by the ASPM L1 enter condition met state, or by the D-state (D1, D2, or D3) of the PCIe device. The D- state of the PCIe device is programmable by software. The L2/L3 ready state is controlled by D-state and power turn-off event.

The power saving of links in lower power states is greater as the link state numbers get larger. The following figure shows the links states of PCIe devices and the relationships of power down states between link partners.

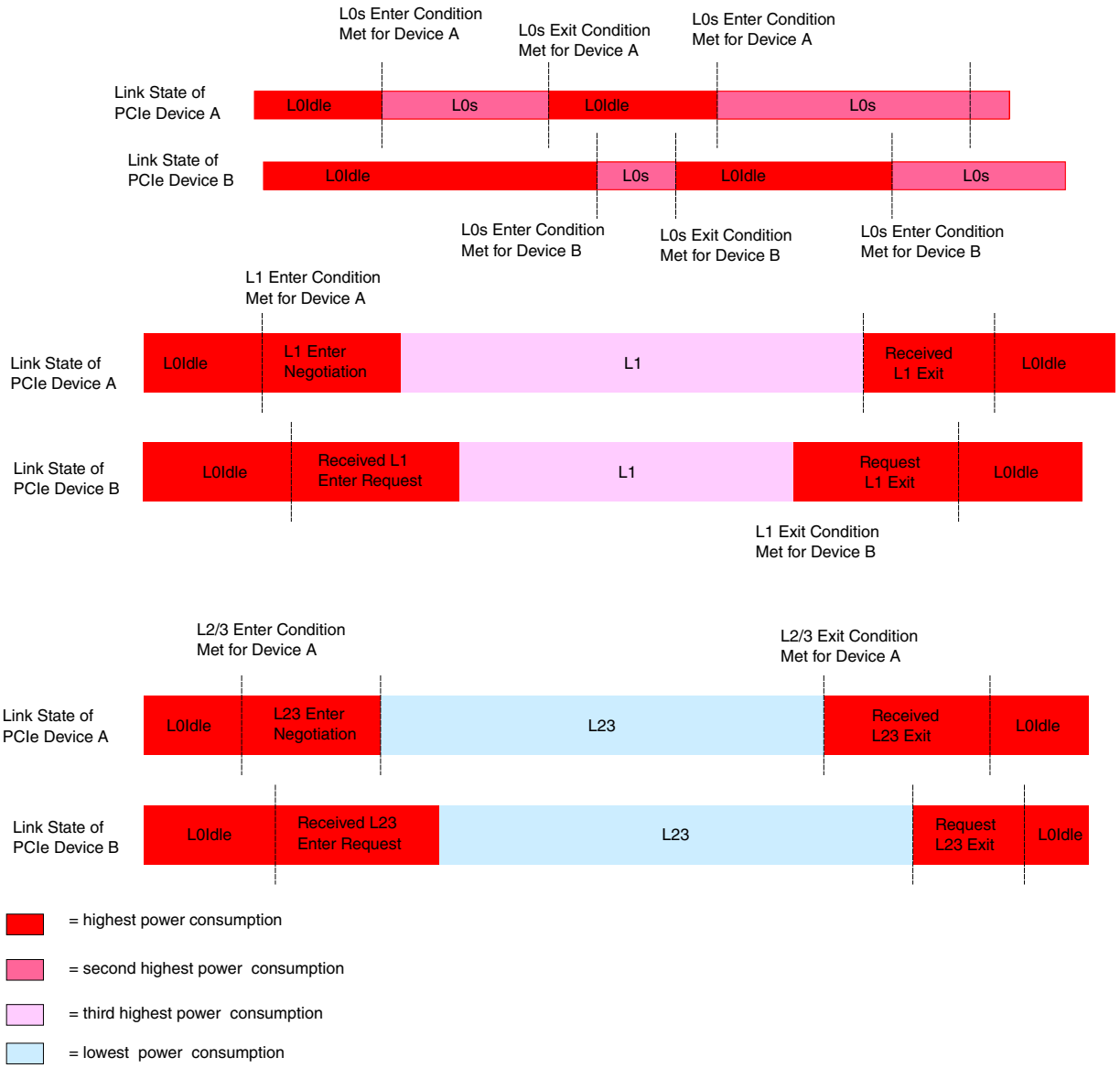


Figure 49-28. Relationships of Power Down States between Link Partners

49.3.11.1 L0s Power Down

L0s is a low power state enabled by Active State Power Management (ASPM). ASPM enabled devices can only control L0s entrance of the transmitter. The receiver L0s is controlled by the remote devices.

L0S ENTER CONDITIONS (ALL CONDITIONS MET)

- ASPM L0s is enabled.
- L0s enter conditions defined by *PCI Express Specification* for a duration of time and there is no higher stage of power down requested.
- The timeout value is controlled by the DEFAULT_L0S_ENTR_LATENCY parameter.

L0S EXIT CONDITIONS (ANY CONDITION MET)

- Any DLLP or TLP pending to be sent.
- L1 enter condition met.
- PCIe link partner request to enter into link recovery.

49.3.11.2 L1 Power Down

L1 is a power down state enabled either by ASPM or by the software controlled D1, D2 or D3 state (which is programmed by the system power management unit). L1 state is a bi-directional link power down state. Both link partners must negotiate to go to L1 state.

L1 ENTER CONDITIONS DUE TO ASPM (ALL CONDITIONS MET)

There are three scenarios that result in the L1 state:

- Scenario 1: L1 Idle Timeout From L0s

ASPM L1 and L0s are enabled.

Link state is in L0s for both transmitter and receiver of the link, and bit 30 of the Ack Frequency and L0-L1 ASPM Control Register is set to 0 (default setting) OR Link state is in L0s of transmitter and bit 30 of the Ack Frequency and L0-L1 ASPM Control Register is set to 1.

L1 enter conditions defined by PCIe spec for a duration of time and there is no higher stage of power down requested.

The timeout value is controlled by the DEFAULT_L1_ENTR_LATENCY parameter.

- Scenario 2: L1 Idle Timeout from L0

ASPM L1 is enabled and L0s is not enabled.

Link state is in L0.

L1 enter conditions defined by PCIe spec for duration of time, and there is no higher stage of power down requested.

The timeout value is controlled by the DEFAULT_L1_ENTR_LATENCY parameter.

- Scenario 3: Application Controlled

ASPM L1 is enabled.

Application request to enter L1 by asserting signal app_req_entr_l1

L1 enter conditions defined by PCIe spec is met.

NOTE

The app_req_entr_l1 is a pulse. The core latches the command and makes sure that L1 has been entered.

L1 ENTER CONDITIONS DUE TO D1/D2/D3 STATES (ALL CONDITIONS MET)

- All functions that are programmed to D1, D2 or D3 states.
- Always enter L1 when L2/L3 PM turn-off negotiation has not yet been done.

L1 EXIT CONDITIONS (ANY CONDITION MET)

- Software requests a higher stage of power down.
- Any DLLP or TLP pending to be sent.
- Application requesting exit of L1 by asserting signal app_req_exit_l1.
- Link partner requesting exit of L1.

Once L1 has exited, another L1 entry will not be initiated for 10us if the enter L1 condition is due to ASPM. If the enter L1 condition is due to lower power D-state, the core will enter L1 again after a wait time of cfg_cpl_sent_count cycles defined in PL register. This wait time to ensures the exit conditions have been served.

3.14.3 L2/L3 Power Down

The core has control over the L2 or L3 ready link state. After the L2/L3 ready is entered, the downstream device will begin preparation for the power and clock removal. After main power has been removed, the link will transition to L2 if Vaux is provided, or it will transition to L3 if no Vaux is provided. L2/L3 ready is a bi-directional link power down state.

L2/L3 ENTER CONDITIONS (ALL CONDITIONS MET)

- PME_Turn_Off/Pme_To_Ack handshake has been completed at any of D0,D1,D2,D3 states.
- Application is ready to be turned off by asserting signal app_ready_entr_l23.

L2/L3 EXIT CONDITIONS (ANY CONDITION MET)

- Device is programmed with capability to support PME and application requests wakeup by asserting the apps_pm_xmt_pme signal or by triggering a native hot-plug event when D-state is in D1, D2 or D3.
- Link partner requesting exit of L2/L3.

The core supports beacon signaling by asserting signal pm_phy_beacongen or wake when a wake-up event is initiated by a PCIe device.

49.3.12 Completion Timeout Ranges

Timeout ranges are supported as defined in the *PCI Express 3.0 Specification*.

The Device Capabilities 2 Register (Offset 24h) shows support for all ranges. The Device Control 2 Register (Offset 28h) will have a reset value equal to the default value in the spec: "0000b Default range: 50 us to 50 ms". If the default value is used then the timeout will be in "Range B: 0101b: 16ms to 55ms." This range was chosen for the default because the *PCI Express 3.0 Specification* states "It is strongly recommended that the Completion Timeout mechanism not expire in less than 10 ms." The following table illustrates the specification values versus the PCI Express core values for the ranges.

NOTE

As per the PCIe 2.1 spec, "This mechanism is intended to be activated only when there is no reasonable expectation that the Completion will be returned, and should never occur under normal operating conditions."

Table 49-24. PCIe Core Completion Timeout Ranges versus PCI Express Specification

Range	Encoding	Spec Minimum	Spec Maximum	PCIe Core Minimum	PCIe Core Maximum
Default	0000b	50µs	50ms	28ms	44ms
A	0001b	50µs	100µs	65µs	99µs
A	0010b	1ms	10ms	4.1ms	6.2ms
B	0101b	16ms	55ms	28ms	44ms
B	0110b	65ms	210ms	86ms	131ms
C	1001b	260ms	900ms	260ms	390ms
C	1010b	1s	3.5s	1.8s	2.8s
D	1101b	4s	13s	5.4s	8.2s
D	1110b	17s	64s	38s	58s

49.4 AXI Bridge Module

49.4.1 Product Overview

49.4.1.1 Overview

The PCIe Core provides an AXI Bridging capability for directly adding a PCI Express link to an AXI system fabric. This significantly reduces the time to design PCI Express into an AXI-based SOC.

The AXI Bridge Module acts as a bridge between the standard AXI interfaces and the PCIe Core native interfaces. The bridge interconnects the AXI interfaces within an AMBA-embedded system with a remote PCIe link, as either a root complex port or as an endpoint port. The bridge supports up to two AXI interfaces, one for an AXI master, and one AXI bus shared for AXI Slave and DBI bus access to the native PCIe core.

The AXI master interface enables a remote PCIe device to read and write to an AXI slave connected to the AXI bridge. The AXI slave interface enables an AXI master to read and write through the AXI bridge to a remote PCIe device. The slave DBI (see DBI Access) enables an AXI master to read and write to registers inside the native PCIe core, It is shared with the AXI slave interface.

Throughout this chapter, the terms inbound and outbound are defined with respect to the AXI fabric. That is, inbound transactions are defined as the transactions presented by the native PCIe core's AXI master interface. Outbound transactions are defined as the transactions generated by an AXI master that targets a remote PCIe device.

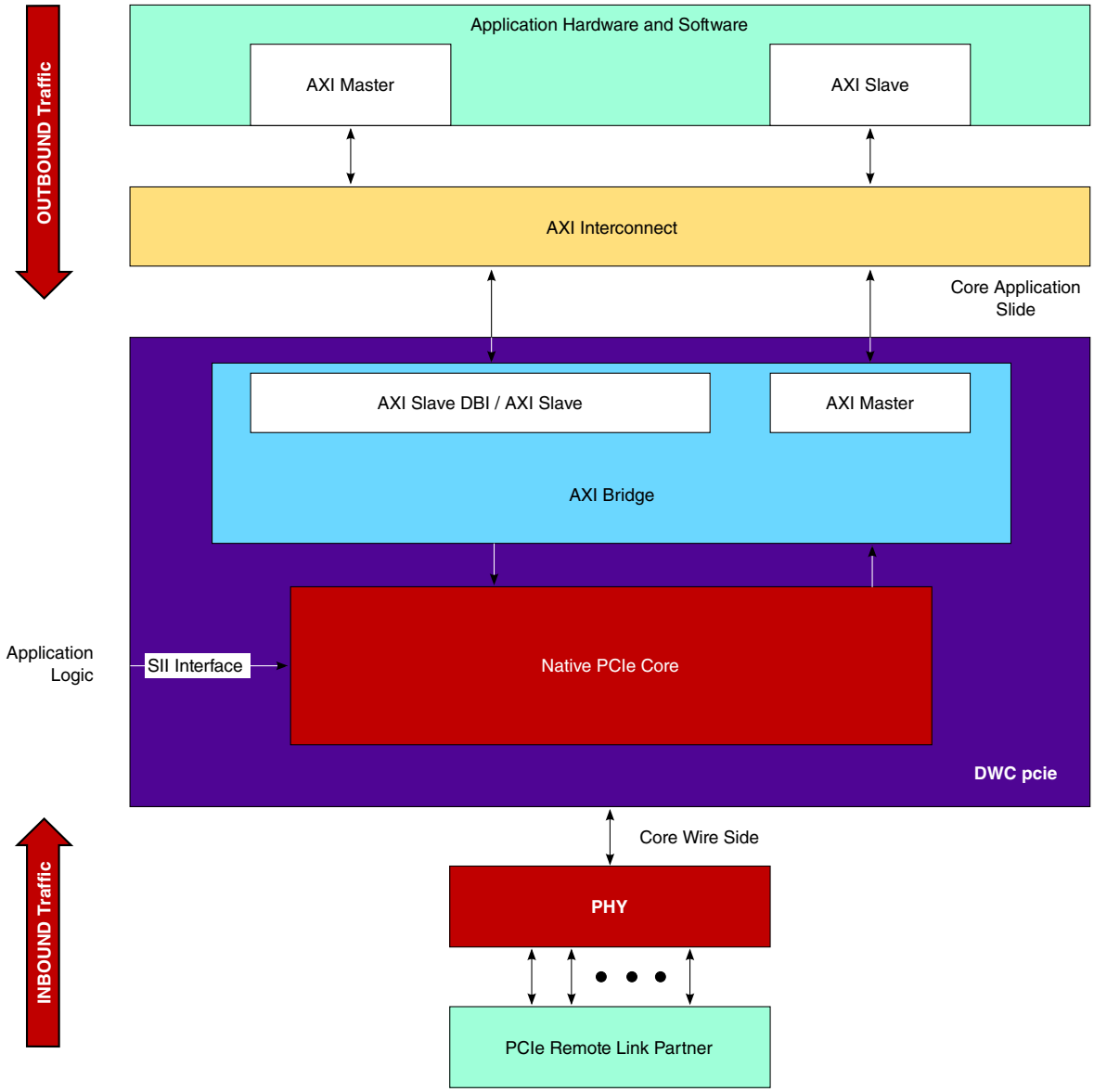


Figure 49-29. System-Level View of the DWC PCIe AXI Core

49.4.1.2 Interfaces

The figure below shows the DWC PCIe AXI core top-level interfaces.

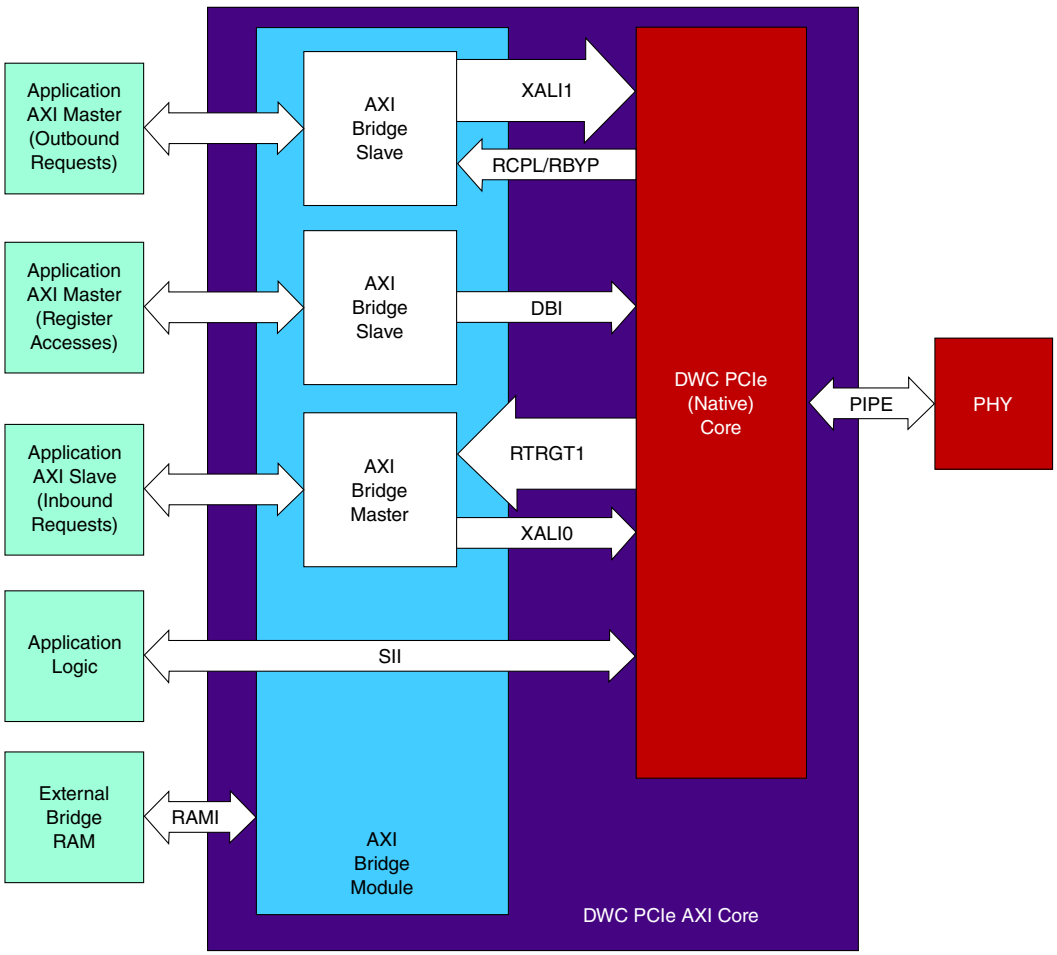


Figure 49-30. DWC PCIe AXI Core Top-Level Interfaces

For definitions of acronyms used for block and interface names, see Terms and Abbreviations.

49.4.1.3 Features List

The AXI Bridge Module supports the following features.

- AXI Master and Slave interfaces for inbound and outbound PCI Express requests.
- Multi-function support (up to 8 functions) [EP mode only].
- All types of PCI Express transactions supported through the AXI Bridge.
- A shared AXI Slave interface to access native core's CDM registers
- Programmable buffer sizes for AXI master and slave requests.

- Independent programmable user-defined clock rates for the PCI Express core, AXI master bus, AXI slave bus,.
- Programmable maximum number of outstanding inbound and outbound read requests for AXI.
- All burst-sizes supported for both AXI master and slave interfaces.
- Programmable and extended AXI burst lengths to support up to 4K read/write burst lengths over AXI master and slave interfaces.
- Little-endian operation.
- Independent maximum read request and transfer sizes between AXI and PCI Express (transfers can be split into multiple transfers).
- Response to AXI slave request combined gathering from split PCI Express completions that are received in-order.
- Response to AXI master request combined gathering from multiple AXI responses.
- PCIe legacy interrupt or MSI support.
- User-defined error mapping between PCI Express errors (UR, CA, CRS, poisoned, and ECRC error) and AXI slave response errors (SLVERR and DECERR).
- User-defined error mapping between PCI Express errors (UR, CA, CRS, poisoned, and ECRC error) and AXI master response error (DECERR_W and DECERR_R).
- Programmable byte parity check for the address and data buses throughout the bridge.
- Non-contiguous byte enables supported for inbound read/write and outbound write TLPs.
- Programmable MSI Interrupt controller to detect and terminate inbound MSI TLP's in the bridge for RC and DM (RC mode) products.

49.4.1.4 Limitations

The table below identifies the limitations encountered when using the AXI bridge with the DWC PCIe native core.

Table 49-25. AXI Bridge Limitations

AXI Bridge Limitation	Note
PCIe completions of a decomposed outbound AXI read request must be returned in-order.	Decomposition of outbound reads will not occur if your application master always generates read requests of size less than Max_Read_Request_Size. If decomposition occurs, and the remote device (or switch) is reordering completions, then corruption will occur at the Slave Response Composer in the AXI bridge. The Slave Response Composer only performs only in-order reassembly. In this case, you can set the AMBA Multiple Outbound Decomposed NP Sub-Requests Control Register to '0' to avoid corruption, at the expense of some performance.

Table continues on the next page...

Table 49-25. AXI Bridge Limitations (continued)

	See Outbound Ordering Limitation #1 for more details.
Posted and non-posted AXI writes targeting the bridge slave should use different AXI tags in order to allow the relevant B-channel responses to be received in order	See Outbound Ordering Limitation #2 for more details.
The AXI bridge only supports the CPL queue in ìStore and Forwardî mode, when a ìSegmented Queueî architecture is used. Otherwise, the CPL queue operates in Bypass mode, as it does for the ìSingleî and ìMultipleî queue architectures.	Set completions (CPL) to Bypass mode if you have not configured the core to use a segmented buffer queue architecture. See Queue to Port Mapping.
Vendor Messages must not be decomposed.	See AHB/AXI Message Address and Size Limitations for more details.
Big-endian operation is not supported	-
PCIe Advanced Error Reporting (AER) is not supported in the AXI bridge.	AER is only supported with respect to the native PCIe core. Errors detected by the bridge are not reported as part of AER.

49.4.2 Bridge Architecture

49.4.2.1 Bridge Architecture Overview

This module contains AXI master and slave protocol handlers, internal slave and master control for generic request and response interfaces, a packet composer, and a packet decomposer for response formation.

The slave and master protocol handlers support the AXI protocol conversion between an AXI transfer and a generic transfer within the bridge. The slave and master generic interface (GIF) supports the conversion of an AXI transfer to a PCIe transaction. The packet decomposer and composer support the segmentation and reassembly of a PCIe transaction.

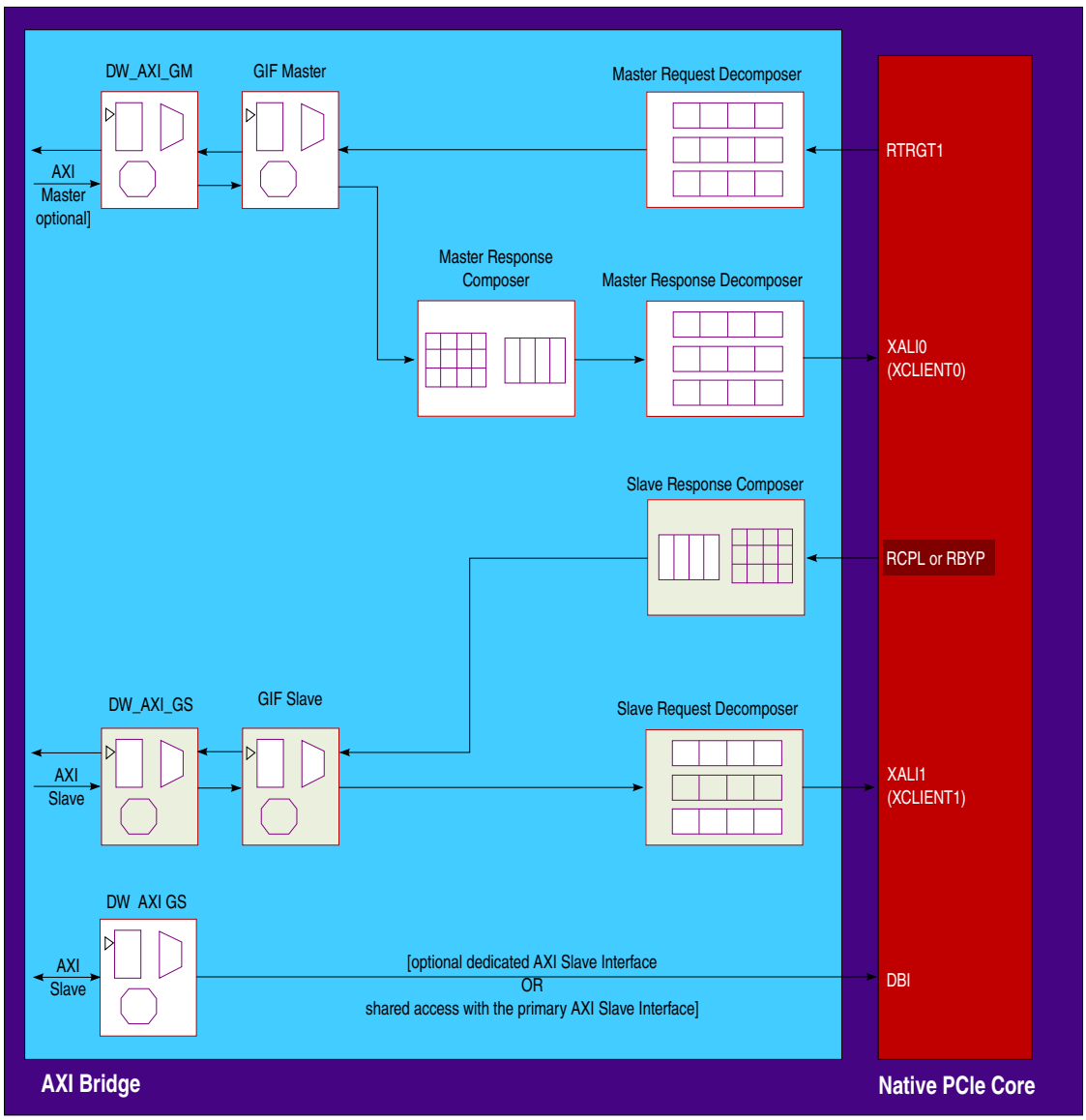


Figure 49-31. AXI Bridge Module, Block Diagram

The bridge module provides flexible buffering (Bridge Buffering) and tag management (Outbound Bridge Tag Management) to facilitate seamless bridging between the PCI Express native core and your application modules.

It has an optional master interface and a second optional slave (DBI) for accessing local registers in the native core CDM. Note also that DBI access is without the second slave (DBI). In this mode (shared DBI) the single slave is used to access both the DBI and the PCIe link.

49.4.2.1.1 Inbound Processing Module Chain

The following table lists the bridge modules involved in processing inbound requests.

Table 49-26. Bridge Modules Involved in Processing an Inbound (PCIe -> AXI) Requests

Module	Processing Step	Function
Master Request Decomposer	Request	Performs packet segmentation to satisfy the AXI fabric's "Max Burst Size" and "4KB Address Boundary" rules. These rules are discussed in Bridge Buffering.
Generic InterFace (GIF) Master	Request	The internal bridge circuitry is based on a generic protocol (GIF) and is not explicitly aware of whether it is interfacing to AXI or AHB.
Master Interface (DWC_axi_gm)	Request	This is the AXI protocol handler.
Master Response Composer	Response (Completion)	Performs packet reassembly caused by the bridge-initiated packet segmentation on the inbound request in the Master Request Decomposer.
Master Response Decomposer	Response (Completion)	Performs packet segmentation to satisfy PCI Express "Max Payload Size" and "Max Read Request Size" rules. These rules are discussed in Bridge Buffering.

49.4.2.1.2 Outbound Processing Module Chain

The following table lists the bridge modules involved in processing outbound requests.

Table 49-27. Bridge Modules Involved in Processing an Outbound (AXI -> PCIe) Requests

Module	Processing Step	Function
Slave Interface (DWC_axi_gs)	Request	This is the AXI protocol handler
Generic InterFace (GIF) Slave	Request	The internal bridge circuitry is based on a generic protocol (GIF) and is not explicitly aware of whether it is interfacing to AXI or AHB.
Slave Request Decomposer	Request	Performs packet segmentation to satisfy PCI Express ìMax Payload Sizeî and ìMax Read Request Sizeî rules. These rules are discussed in Bridge Buffering.
Slave Response Composer	Response (Completion)	Performs packet reassembly caused by the following sources of segmentation: •Bridge initiated packet segmentation on the outbound request in the Slave Request Decomposer. •Packet segmentation caused by the remote completer.

The application master request (at the AXI Slave Interface) will not - AXI protocol demands it - request more than 128 bytes and will not issue a request that involves crossing a 4K page boundary. Therefore the AXI response (completion) - composed from one or more completions that arrive in off the PCIe wire -is AXI compliant and does not need to be processed in a decomposer to enforce AXI page boundary and maximum packet size rules. Therefore there is no Slave Response Decomposer in the bridge module. In contrast, there is a Master Response Decomposer.

NOTE

CC_SLV_MTU

Also known as CC_SLV_RD_REQ_SIZE

This value is used to set memory sizes in the bridge (Slave Request Decomposer Data FIFO and Slave Response Composer).

Calculated as CC_SLV_BURST_LEN * (SLAVE_BUS_DATA_WIDTH / 8) = 16 * 64 / 8 = 128

CC_SLV_BURST_LEN is the maximum burst in beats that the application will ever issue to the bridge slave interface.

49.4.2.2 Decomposition

The bridge manages the different transfer sizes between PCIe and AXI.

The maximum payload is configured independently between the AXI and PCIe systems. Each system's transfer size will sometimes need to be converted causing a split of a request and composition of a response.

Automatic segmentation and reassembly of outbound (application to PCIe wire) packets is performed to satisfy PCI Express *Max_Read_Request_Size* and *Max_Payload_Size* rules. When a decomposition rule is triggered, the outbound TLP is broken up into two or more smaller TLPs. If the TLP type is Non Posted then additional PCIe TAGs are used from the TAG pool. Additional entries in the outbound header buffer are also used.

In a similar manner, automatic segmentation and reassembly of inbound packets is performed. The same AXI ID is used for all decomposed Non-Posted (but not for Posted; see Inbound Bridge Tag Management) packets.

The host software can program the PCIe device to support certain maximum write transfer sizes and maximum read request sizes. The native PCIe core's configuration module contains the device's *Max_Payload_Size* and *Max_Read_Request* size information, which are defined by the application software. The AXI bridge supports mismatches that occur when the AXI maximum transfer length is different than the *Max_Payload_Size* and *Max_Read_Request* size. For example, an inbound read transfer has an associated response buffer that is dependent on the remote PCIe device's *Max_Read_Request* size. An inbound write transfer has a master write buffer that is dependent on the remote PCIe device's *Max_Payload_Size*.

More detailed information on decomposition rules and effects is available at the end of the AXI bridge section at AXI Decomposition Rules.

49.4.2.3 Bridge Buffering

The bridge has additional buffering in addition to buffering already present in the native core.

The following table identifies all of the RAMs that are used in the AXI bridge for buffering purposes.

Table 49-28. Lists of Buffering Related RAMS

RAM Name	Alternative Name
Slave Request Decomposer Header and Data Queues : buffering of Outbound AXI Request	XADMX1 Decomposer Header and Data RAMs
Slave Response Composer : buffering of Inbound Responses to Outbound AXI Request	RADMX Composition RAM
Slave Response Asynchronous Clock Crossing FIFO : synchronization of Inbound Responses to Outbound AXI Request	RADMX Asynchronous RAM
Master Request Decomposer Header and Data Queues : buffering of Inbound PCIe Request	RADMX Decomposer Header and Data RAMs
Master Response Composer : buffering of Outbound Responses to Inbound PCIe Request	GM Composition RAM
Master Response Decomposer Header and Data Queues : buffering of Outbound Responses to Inbound PCIe Request	XADMX0 Decomposer Header and Data RAMs

More detailed information on buffering is available at the end of the AXI bridge section at Outbound Bridge Tag Management .

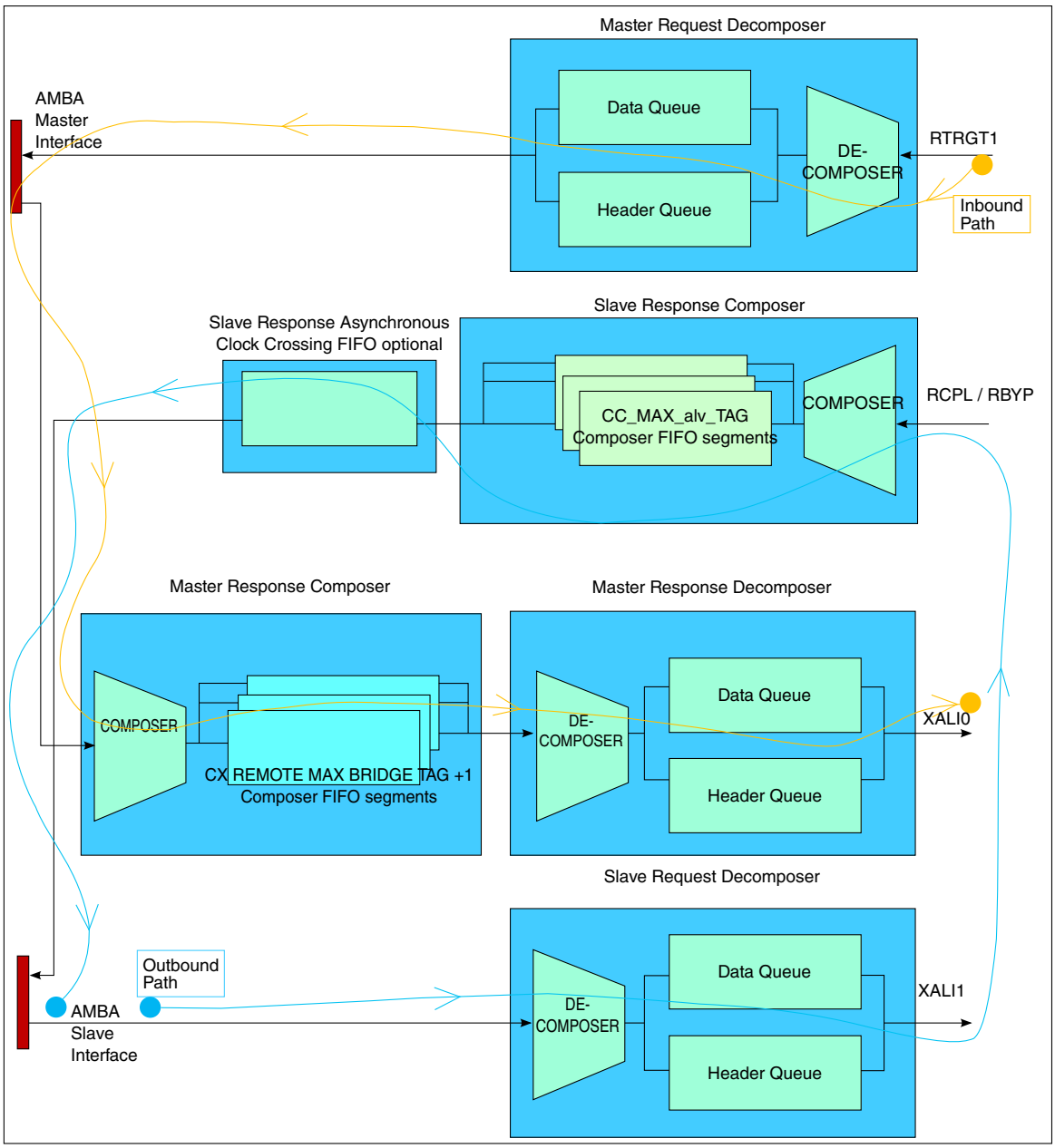


Figure 49-32. Overview of Bridge Buffering

49.4.2.4 Outbound Bridge Tag Management

The bridge transfers IDs/TAGs between PCIe and AXI.

Transactions on PCIe and AXI bus have separate IDs/TAGs and as the bridge transfers the data, it maps and keeps track of the IDs/TAGs on each side.

The diagram found here gives an overview of the outbound tag management system for Non Posted requests.

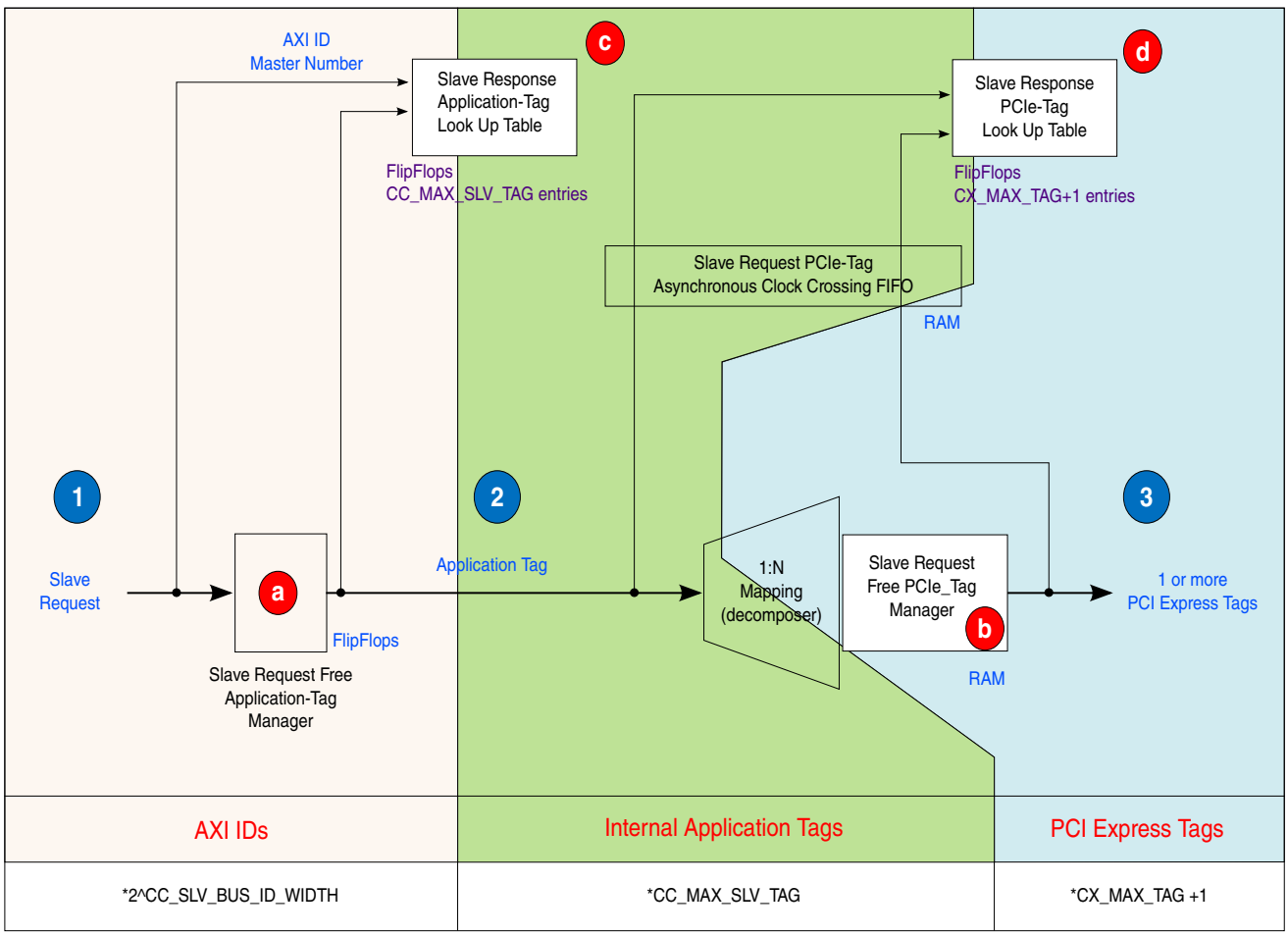


Figure 49-33. Outbound Tag Management Architecture (Non Posted requests only) - request path is shown but completion path is omitted.

When an AXI outbound Non Posted request at the slave interface is converted to a PCIe request TLP, it is necessary to:

- Select a PCIe tag from the 'free' pool of currently available PCIe tags. This is performed using a Tag Manager (see Module 'b' in figure above). When the pool is exhausted, no more PCIe request TLPs can be transmitted until the pool is replenished by tags from inbound PCIe completion TLPs. See steps 2 and 3 in figure above.
- Store the mapping from AXI request to PCIe TAG, so that when the corresponding PCIe completion TLP arrives back in the future, it can be correctly associated with

the originating request. This is achieved using a Tag Look Up Table (LUT) - see Module 'd' in figure above.

The process of mapping AXI requests IDs to PCIe tags is a two-step process - Application Tag mapping and PCIe Tag mapping (as described above).

Application Tag mapping -which is internal to the bridge -is an intermediate step which has to occur first. See steps 1 and 2 in figure above.

In the AXI protocol, it is possible to have several AXI requests with the same AXI ID. Therefore it is necessary for the bridge to differentiate between individual AXI requests to enable it to track them and their completions throughout the system.

This is necessary to aid in reassembling or recomposing (but not necessarily re-ordering, see Outbound Ordering Limitation #1 for more details) the multiple inbound completions that are returning for an original single AXI outbound request.

Multiple completions can occur due to decomposition (see Decomposition) of the original outbound request or due to the remote link partner choosing to complete a request using multiple completions.

Application tag mapping occurs via the following process:

- The bridge maps the AXI ID to an internal application tag (see Module 'a' in figure above). This application tag identifies the reserved space in the Slave Response Composer for the expected returning completions.
- The bridge store the mapping from the AXI ID to the internal application tag, so that when the corresponding PCIe completion TLP arrives back in the future, it can be correctly associated with the originating request. This is achieved using a Tag Look Up Table (LUT) - see Module 'c' in figure above.
- If decomposition occurs, each PCIe TLP is assigned a unique PCIe tag from the pool of CX_MAX_TAG (31) tags (see Module 'b' in figure above). Each of these PCIe tags is associated with the same internal application tag and this information is stored is a LUT (see Module 'c' in figure above).

If at any time, all PCIe TAGs or internal application tags are consumed, then the AXI Bridge Slave interface will not accept any new transactions and will apply back-pressure to the AXI fabric.

49.4.2.5 Inbound Bridge Tag Management

The bridge transfers IDs/TAGs between PCIe and AXI.

Transactions on PCIe and AXI bus have separate IDs/TAGs and as the bridge transfers the data, it maps and keeps track of the IDs/TAGs on each side. When an inbound PCIe request TLP is converted to an AXI request at the master interface, it is necessary to:

- Select an AXI ID from the 'free' pool of currently available AXI IDs.
  - For Non Posted requests, this is done using an ID Manager - see Module 'b' in the figure below.
  - Posted (P) requests always use ID '0'.
- When the pool is exhausted, no more requests can be launched onto the AXI fabric until the pool is replenished by IDs. An ID is released to the pool when the completion has arrived back from the application AXI master. In the case where the inbound PCIe TLP request was decomposed, then the ID is released when all the individual completions have arrived back into the master response composer.
- Store the mapping from PCIe TAG to AXI request ID to, so that when the corresponding AXI response arrives back in the future, it can be correctly associated with the originating request. This is achieved using an ID Look Up Table (LUT) - see Module 'a' in the figure below.

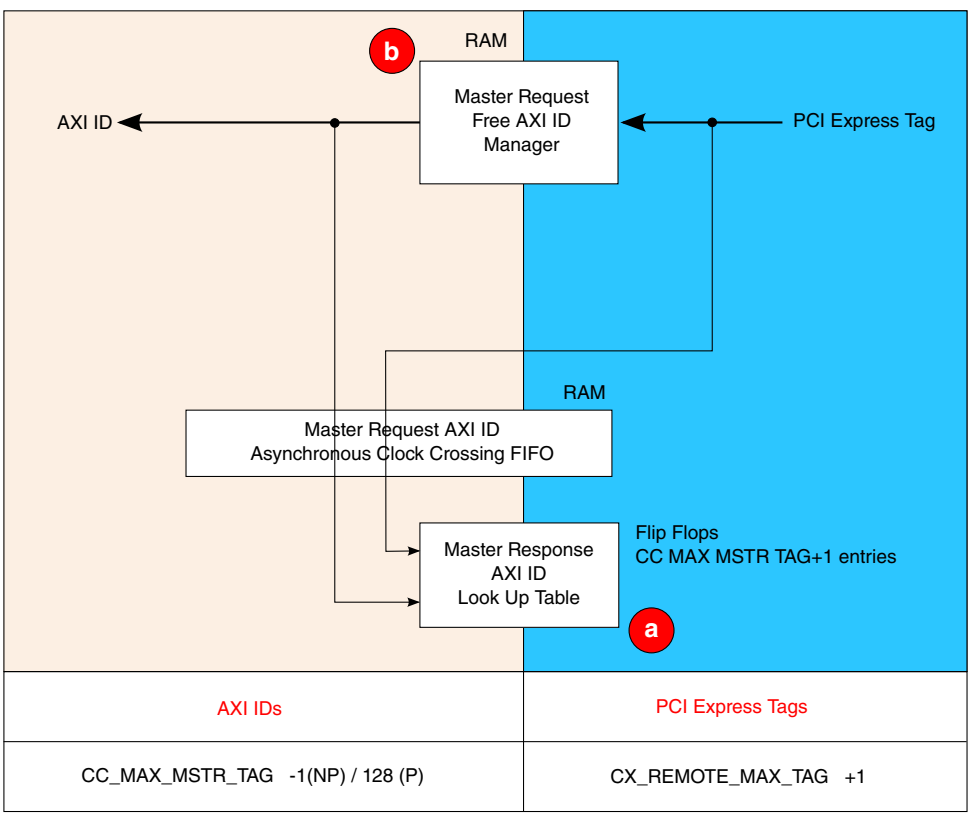


Figure 49-34. Inbound Tag Management Architecture - request path is shown but completion path is omitted.

Decomposition (Decomposition) does not affect AXI ID usage. If decomposition is occurring, then the same ID is used for each AXI request that is generated by decomposition. Each response is identified by the AXI bridge using its response ID.

Since the PCIe Posted transfer expects no response (completion) for a Posted request, the AXI bridge drops the write responses.

The AXI bridge master interface issues all Posted requests with an ID of '0'. The AXI bridge master interface issues Non Posted requests with an ID from the range 1 to CC_MAX_MSTR_TAG -1 (4-1=3).

It is possible to have CC_MAX_MSTR_TAG-1 (4-1=3). outstanding Non Posted requests and 128 Posted requests at the same time since Posted requests are always writes and use a different AXI ID signal (mstr_awid). The following memory devices are connected with inbound tag management in the bridge.

49.4.2.6 Inbound Order Enforcement for AXI Bridge

The DWC PCIe AXI bridge has the inbound buffers shown in the figure found here.

- a single FIFO queue structure (Master Request Decomposer Data and Header Queues) for inbound read/write requests
- a segmented buffer queue structure (Slave Response Composer) for inbound completions in response to outbound read requests.

There is no reordering after an inbound request has left the native core's receive queue. The request FIFO is blocking and will not allow read or write requests pass each other. This default design architecture is designed to serve inbound traffic as a FIFO (first-come, first -served) i.e. in-order service.

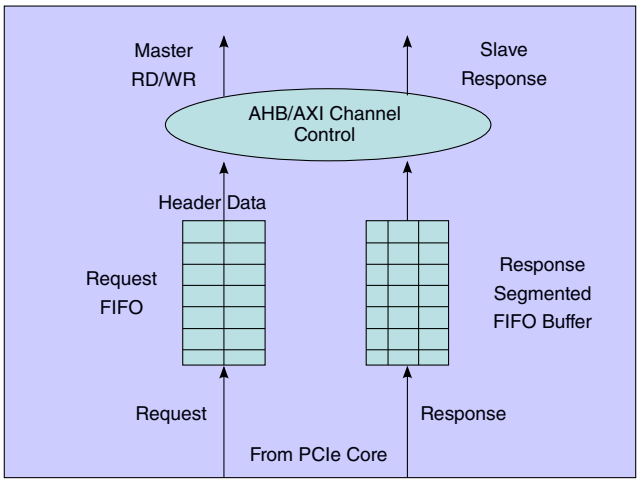


Figure 49-35. DWC PCIe AXI Bridge Inbound Traffic Queue Architectures

The native core (before the bridge) has flexible buffering (see Queue to Port Mapping and PCIe order rule enforcement is performed within the native core. See PCIe Core Inbound Order Enforcement.

49.4.2.6.1 Ordering Enforcement Hardware Lock Feature

Once a completion has been taken out of the receive queues of the DWC PCIe core, it can pass a posted request.

To enforce the PCIe Ordering rule ìcompletions should not pass posted if relaxed ordering is not setî for applications that require it, the DWC PCIe AXI bridge employs a hardware lock mechanism feature.

This feature will prevent completions from passing posted requests by requiring that posted transactions all complete through the bridge to an AXI interface before a completion can be taken out of the receive queue. This ensures that previous posted transactions are never passed by a completion. This feature is turned on when the DWC PCIe core is configured to have a receive queue in the segmented buffer mode and completion is in store-forward mode.

The feature employs packet halt signals generated by the bridge and used by the core when deciding how to unload the segmented buffer queue of the core. There is an individual halt signal for posted, non posted and completion transactions. Completion transactions are halted when there are any posted transactions in the bridge FIFO.

The limitation of the above feature is that it gives the posted transactions priority such that it can always pass through the bridge when it is available, while the completions are blocked.

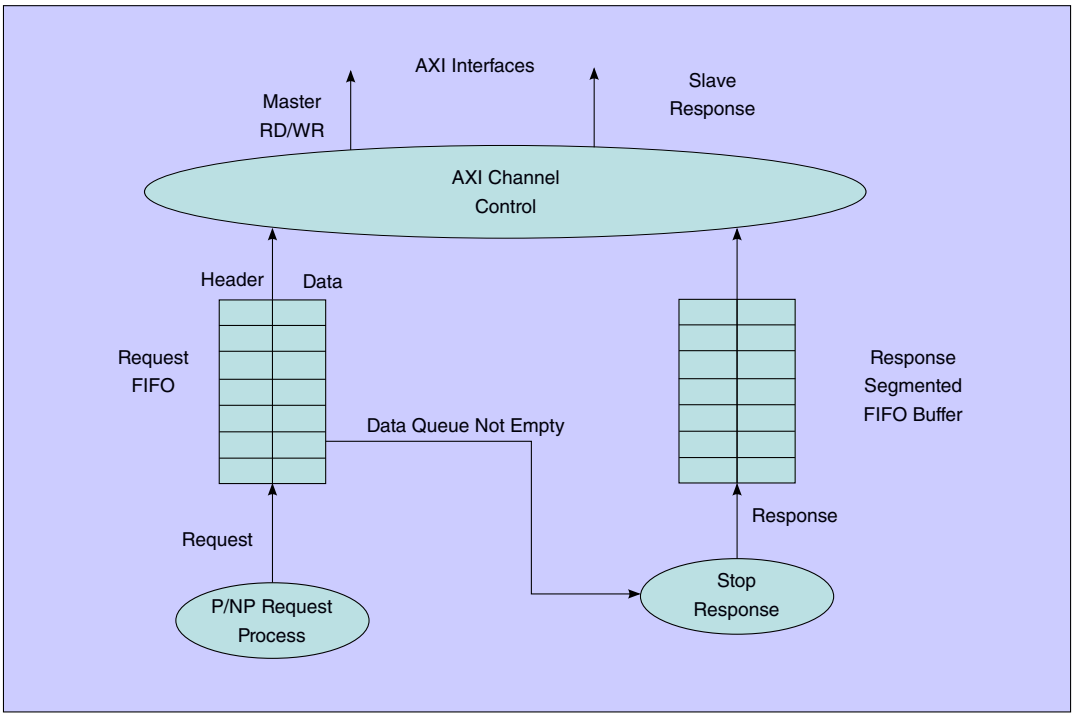


Figure 49-36. DWC PCIe AXI Bridge Inbound Traffic Order Enforcement

49.4.2.6.2 Re-Ordering Effects of AXI Fabric

The ordering rules, in typical AXI bridge configurations, apply only to the PCIe native core inbound queue and not after the inbound or outbound traffic has reached the AXI bridge. This means, for example, if a posted request followed by a completion reaches the AXI bridge, then it is possible that the completion will reach the AXI slave response channel before the posted reaches the AXI master request channel. This depends on the readiness of AXI master request channel (if wait states are asserted by the AXI bus).

The AXI bridge has independent AXI channels for outbound and inbound transfers. Therefore, a read response can be presented on the read outbound AXI channel while a write request is being presented onto the write inbound AXI channel. Since the response of an outbound read and inbound write request can be presented onto the application logic of AXI simultaneously, then the PCIe traffic order rules do not get completely enforced by most AXI bridge applications.

The AXI bridge master will generate a unique ID for each Non-Posted request (If decomposition is occurring, then the same ID is used for each AXI Non Posted request. For more details see Decomposition and Inbound Bridge Tag Management). Each response is identified by the AXI bridge using its response ID.

The AXI bridge master interface issues a Posted request (write) with a unique ID that is independent from master reads. Since the PCIe Posted transfer expects no response, the AXI bridge drops the write responses. If decomposition is occurring, then a unique ID is used for each AXI write request. For more details see Inbound Bridge Tag Management.

49.4.2.6.2.1 Inbound Ordering Limitation

Sometimes a Non-Posted transaction can pass a Posted transaction on the AXI fabric and this is a violation of the ordering rules. The AXI master interface does not wait for a write response - when it issues write transactions on the AW channel - before issuing a read transaction issued on the AR channel. As a result, a read transaction issued on the AR channel (after a write transaction on the AW channel) could complete first. This could happen if the AXI fabric delays completion of a write request (for example, due to latency in a write buffer) and a read request from the same address follows the write.

NOTE

For information on AXI ID generation see Inbound Bridge Tag Management.

49.4.2.6.3 Additional Information (ordering)

- Order enforcement through the native core is discussed in App Note: Order Enforcement Using the PCIe Core.
- Outbound Order Enforcement for AXI Bridge

The DWC PCIe AXI bridge has the following outbound buffers (see the figure below):

- a single FIFO queue structure (Slave Request Decomposer Data and Header Queues) for outbound read/write requests
- a segmented buffer queue structure (Master Response Composer) for outbound completions in response to inbound read requests.

These two queues are independently structured. Therefore, there is no order maintained (between completions and requests) once outbound requests or outbound responses have entered the bridge.

The FIFO is blocking and will not allow read or write requests pass each other. This default design architecture is designed to serve outbound traffic as a FIFO (first-come, first -served) i.e. in-order service. Outbound requests are served onto the PCIe wire in the order that they are received from an AXI master.

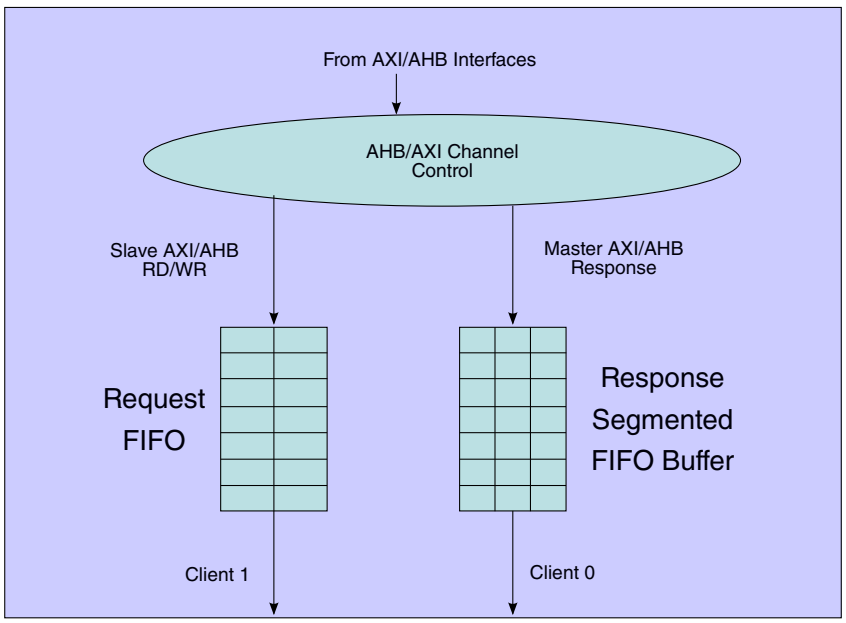


Figure 49-37. AXI Bridge Outbound Traffic Queuing Architecture

The native core (after the bridge) has no transmit buffering and requests and completions are transmitted onto the wire directly. Completions use a different interface (XALI0) to requests (XALI1) and may or may not pass the outbound requests depending on client arbitration.

49.4.2.6.4 PCIe Completion Reordering

If the returning completions (for all Non Posted outbound requests with the same AXI ID) from the remote device are returned out-of-order, the bridge Slave Response Composer will re-order them according to AXI IDs.

Only completions corresponding to outbound requests with the same AXI ID are re-ordered, while all others ones will be returned following the chronological order in with which they are received from the remote device.

49.4.2.6.5 Outbound Ordering Limitation #1

An outbound AXI request at the slave interface may be decomposed into multiple PCIe TLP requests.

See Decomposition for more information. There is a PCI Express protocol ordering rule that states that the completions associated with the different TLPs may pass each other. The current bridge Slave Response Composer design cannot handle this scenario and will not re-order the completions, resulting in data corruption. To avoid data corruption, one of the following workarounds is required.

- Prevent outbound decomposition by ensuring that your application master does not generate bursts of size greater than Max_Read_Request_Size. Otherwise, you must program or design your PCI Express system with a larger value of Max_Read_Request_Size. See Decomposition Side-Effects.
- Set AMBA Multiple Outbound Decomposed NP Sub-Requests Control Register to '0' which will disable the possibility of having multiple outstanding non-posted requests that were derived from decomposition of an AMBA request. This should only be done when decomposition is guaranteed to occur and completions are coming back out of order from the wire, as it restricts the PCIe transmit link performance by only having one outstanding non-posted request TLP (from a request that is being decomposed) on the PCIe link at any one time.

It is also a PCI Express ordering rule that a remote device which completes a single request using multiple completions (CplD) must return them in-order. The bridge can accept an interleaved stream of multiple completions (in response to multiple requests it originally transmitted), provided that all the completions for any one original request are in-order.

49.4.2.6.6 Outbound Ordering Limitation #2

Posted and non-posted AXI *writes* targeting the bridge slave must use different AXI tags in order to allow the relevant B-channel responses to be received in-order.

If this restriction is not observed, it can happen that a B-channel response to a posted (P) write (MWr/Msg) arrives before the B-channel response associated to a non posted (NP) write (for example, IOWr), even if the NP request send first.

The B-channel response to a P write is issued by the AXI bridge slave immediately after the request is accepted by the bridge slave. The B-channel response to an NP write is issued by the AXI bridge slave when the completion (CplD) arrives back from the remote link partner.

49.4.2.6.7 Additional Information (AXI bridge bandwidth)

Order enforcement through the native core is discussed in App Note: Order Enforcement Using the PCIe Core.

49.4.3 PCIe AXI Core Operations

49.4.3.1 AXI Sideband (Misc. Bus) Signals

An AXI sub-system can only convey data, address and read/write information. It has no way to propagate PCIe concepts such as TLP type, Posted/Non Posted, Function Number, TLP attributes and so on. Therefore, some method is required to map these PCIe concepts between AXI and PCIe. This is done through the Software option.

Use the Internal Address Translation (iATU) to map outbound AXI transactions from different AXI address regions to particular PCIe address regions and TLP types. For example, map all AXI transactions in the region 0x0010FD00 - 0x00FFFFFF to CFG TLPs in the PCIe address region 0xD010FD00 - 0xD0FFFFFF.

49.4.3.2 Supported AXI Transfer Type

The AXI Bridge Module is compliant with the AMBA 3.0 AXI specification.

49.4.3.3 Supported AXI Burst Operations

- For outbound transfers (accessing bridge SLAVE):
  - The AXI bridge slave supports the incremental burst type (INCR) which is used in conjunction with ARLEN and AWLEN to define any length of burst.
  - The AXI bridge slave does not support the WRAP and FIXED burst types. If your application issues these burst types to the AXI bridge slave, a SLVERR or DECERR response is not returned by the bridge slave. In this scenario, the PCIe core exhibits undefined behavior.
- For inbound transfers (using bridge MASTER):
  - The AXI bridge master used the incremental burst type (INCR) which is used in conjunction with ARLEN and WLEN to define any length of burst.
    - The parameter CC_MSTR_BURST_LEN controls the maximum length burst that will be generated by the master interface.
  - The AXI bridge master does not use the WRAP and FIXED burst types.

49.4.3.4 I/O and CFG Transaction Handling over AXI Bridge

I/O and CFG-type inbound and outbound transfers are fully supported by the AXI bridge. I/O and CFG transactions always have the following characteristics:

Table 49-29. IO and CFG Characteristics

A payload of length equal to one DWORD (four bytes, 32 bits).
First byte enable (FBE) can be any value including '0000'.

Table continues on the next page...

Table 49-29. IO and CFG Characteristics (continued)

A payload of length equal to one DWORD (four bytes, 32 bits).
Last byte enable (LBE) is always '0000'.

49.4.3.4.1 Outbound I/O and CFG Transaction Handling

The following conditions must be satisfied for all outbound CFG and I/O requests presented by the application to the bridge slave interface:

- The outbound IO/CFG transactions should be DWORD aligned. Therefore, slv_a*addr[1:0]=00b and slv_a*size=2.
- The outbound AXI -> PCIe transfer must complete in a single AXI transfer without the need for a bus burst or a series of bus transfers.
- The maximum burst length for outbound IO and CFG transfers is one, corresponding to one DWORD.

There is one methods that can be used to signal to the PCIe core that the application wants to send an IO or CFG transfer.

49.4.3.4.1.1 Method I: Address Translation Method of Sending an Outbound IO or CFG Transfer

The optional internal address translation unit (iATU) in the native core can change the TYPE of an outbound request from MEM to IO or CFG by matching the address of that request to a configured address range as set by the application. See Internal Address Translation (iATU).

NOTE

You should use the ìInternal Address Translation (iATU)î instead of the Slave Request Sideband Bus (slv_a*misc) - to transmit CFG or IO requests.

It is also possible to do the same (for READs only) using a customer defined external address translation unit.

49.4.3.4.2 Inbound I/O and CFG Transaction Handling

The inbound PCIe -> AXI transfer completes in a single transfer without the need for a burst or a series of transfers except during:

- NCBE transfer. e.g. FBE = 1011 | 0101 | 1001 | 1101.
- Unaligned[1] transfer. e.g. FBE = 1110 | 1100 | 1000
- Narrow transfer. e.g. FBE = 1110 | 1100 | 1000 | 0111 | 0011 | 0001

1.For an unaligned (i.e. non-DWORD aligned) inbound PCIe -> AXI transfer (FBE = 1110 | 1100 | 1000), the AXI transfer still completes in one bus beat since LBE = 0000.

49.4.4 Additional AXI Reference Material

49.4.4.1 AXI Decomposition Rules

The host software can program the PCIe device to support certain maximum write transfer sizes and maximum read request sizes.

The native PCIe core's configuration module contains the device's *Max_Payload_Size* and *Max_Read_Request* size information, which are defined by the application software. The AXI bridge supports mismatches that occur when the AXI maximum transfer length is different than the *Max_Payload_Size* and *Max_Read_Request* size. For example, an inbound read transfer has an associated response buffer that is dependent on the remote PCIe device's *Max_Read_Request* size. An inbound write transfer has a master write buffer that is dependent on the remote PCIe device's *Max_Payload_Size*.

49.4.4.1.1 Outbound Decomposition

Automatic segmentation and reassembly of outbound (application to PCIe wire) packets is performed to satisfy PCI Express *Max_Read_Request_Size* and *Max_Payload_Size* rules. When a decomposition rule is triggered, the outbound TLP is broken up into two or more smaller TLPs. If the TLP type is Non Posted then additional PCIe TAGs are used from the TAG pool. Additional entries in the outbound header buffer are also used.

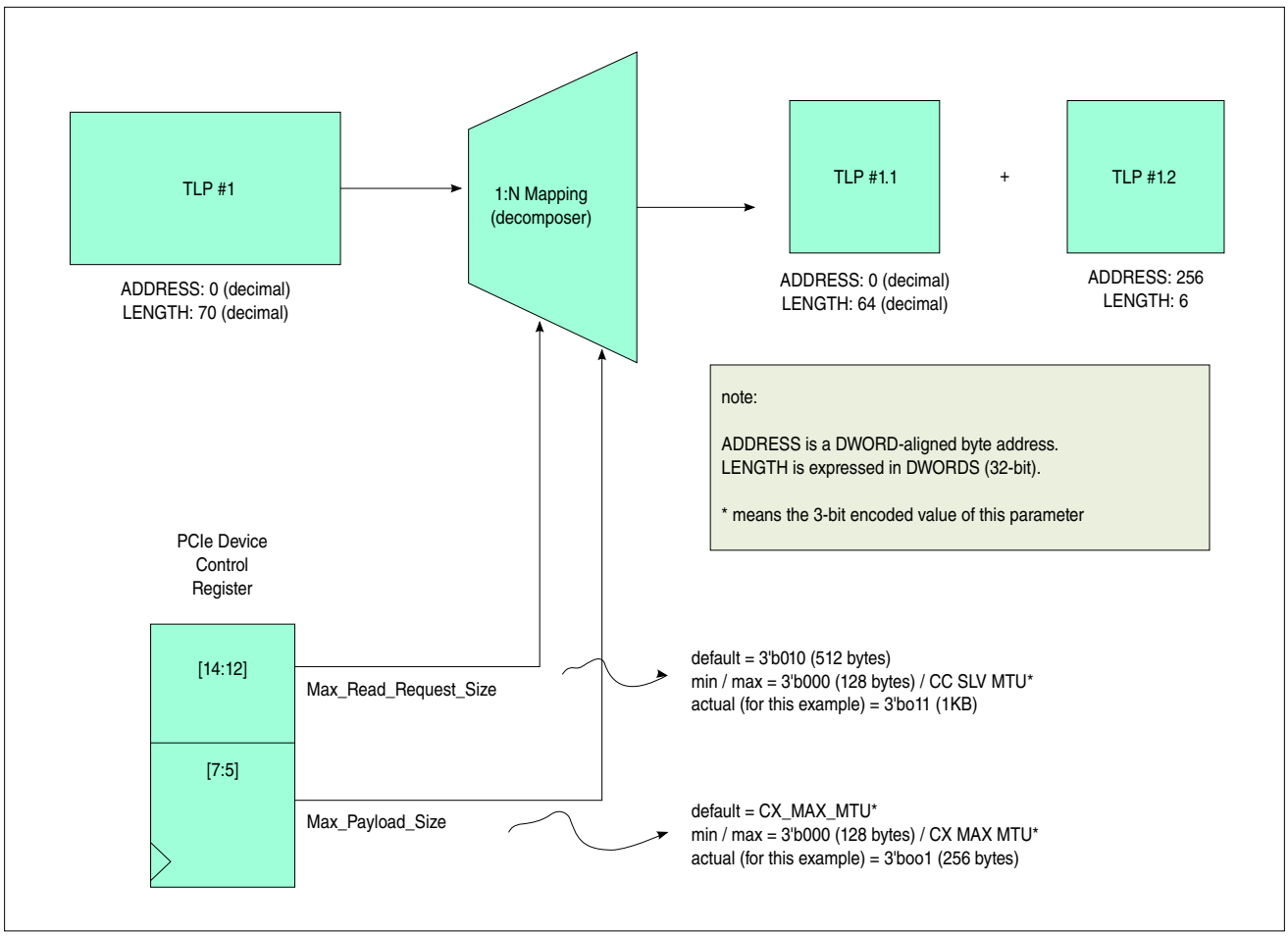


Figure 49-38. Outbound Decomposition Example of MemWr with Max Payload Size set to 256 bytes (64 DWORDs)

The Slave Request Decomposer module uses the following rules to determine when decomposing takes place.

Table 49-30. Outbound Decomposition Rules

TLP Request Type	Decompose[1] When
Read	TLP length >= PCIe Max_Read_Request_Size
Write	TLP length >= PCIe Max_Payload_Size

1. To avoid possible decomposition when the address is not DWORD aligned, you must ensure that TLP length is not only > Max_Read_Request_Size but also >= Max_Read_Request_Size.

The AXI page boundary is 4K, and so the application master request (at the AXI Slave Interface) will not issue a request that involves crossing a 4K PCIe page boundary - AXI protocol demands it.

NOTE

CX_MAX_MTU - 128

The largest packet payload (Maximum Transfer Unit) that the device will support. Specified in bytes and not DWORDS.

This is distinct from the maximum operating payload (Max_Payload_Size) which may be set by software.

49.4.4.1.1.1 Decomposition Side-Effects

Decomposition degrades the PCIe link performance as because it increases the amount of TLP header overhead and uses up extra PCIe TAGS.

Decomposition uses up extra header FIFO locations which will reduce the bridges bus-offloading ability in some cases.

49.4.4.1.1.2 Reducing Outbound Decomposition

To reduce decomposition on the outbound path, observe the following guidelines.

For Write Requests:

- Ensure that your application master does not generate bursts of size greater than or equal to Max Payload Size. In this context 'burst size' refers to the total number of bytes requested, or
- Program your PCI Express system with a larger value of Max_Payload_Size without exceeding CX_MAX_MTU (128).

For Read Requests:

- Ensure that your application master does not generate bursts of size greater than or equal to Max_Read_Request Size or,
- Program your PCI Express system with a larger value of Max_Read_Request without exceeding CC_SLV_MTU (128).

49.4.4.1.2 Inbound Decomposition

In a similar manner, automatic segmentation and reassembly of inbound packets is performed. The same AXI ID is used for all decomposed packets. The figure below illustrates the rules that are used in the Master Request Decomposer module to determine when decomposing takes place.

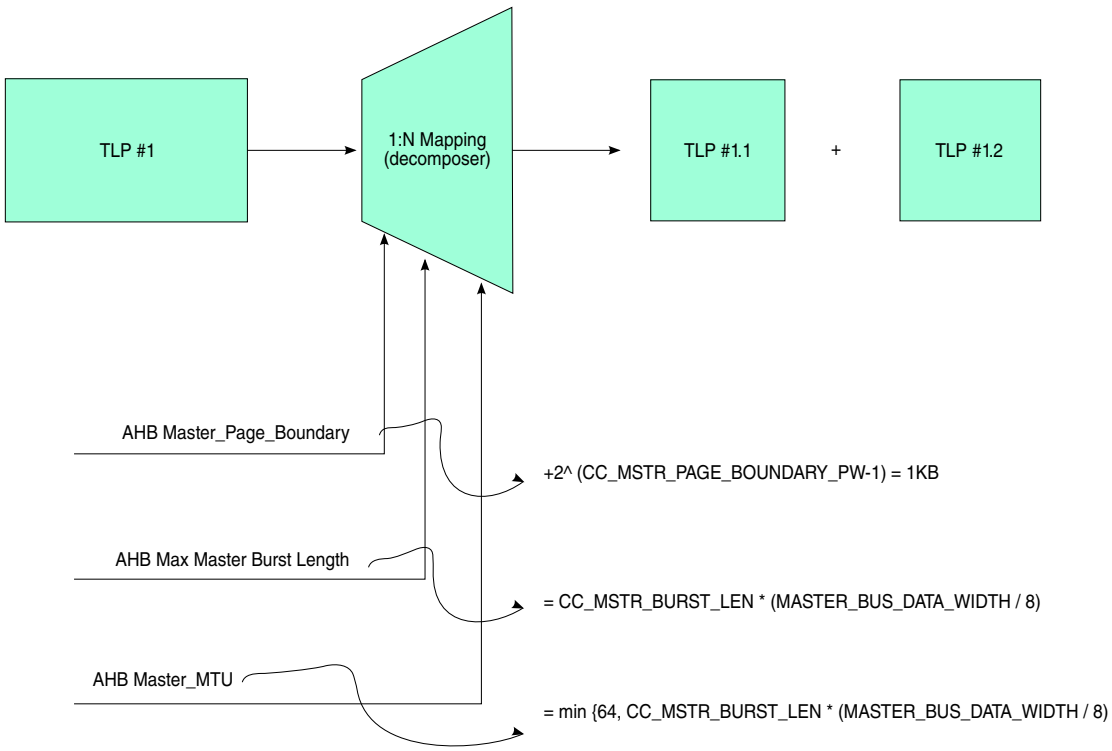


Figure 49-39. Inbound Decomposition

CC_MSTR_BURST_LEN (16) is the maximum burst in bus beats that the bridge will ever issue at the bridge master interface. It is set by the user.

The following rules are used in the Master Request Decomposer module to determine when decomposing takes place.

Table 49-31. Inbound Decomposition Rules

TLP Request Type	Decompose When
Read	TLP length (bytes) > AXI Master_Max_Read_Request_Size (see Figure 49-39)
Write	TLP length (bytes) > AXI Master_MTU (see Figure 49-39)
Read or Write	TLP address + TLP length crosses an AXI Master_Page_Boundary (see Figure 49-39)

The AXI *Master_Page_Boundary* specifies an address page boundary of 4K. The AXI bridge will ensure that the inbound request will not cross the selected page boundary when driven onto AXI master interface. .

49.4.4.1.2.1 Reducing Inbound Decomposition

To reduce decomposition on the inbound path, observe the following guidelines. For Read Requests:

- Ensure that the remote link partner does not issue a read request TLP with a requested length greater than the AXI *Master_Max_Read_Request Size* (see Figure 49-39).

or

- Design your application (and configure the bridge) to handle longer AXI bursts. That is, increase the value of the configuration parameter CC_MSTR_BURST_LEN or increase the AXI master data bus width.

For Write Requests:

- Ensure that the remote link partner does not issue a write request TLP with a payload greater than the AXI *Master_MTU* (see Figure 49-39).

or

- Design your application (and configure the bridge) to handle longer AXI bursts. That is, increase the value of the configuration parameter CC_MSTR_BURST_LEN or increase the AXI master data bus width.

For All Requests:

- Program or design your PCI Express system so that a TLP address plus the TLP length does not cross the AXI *Master_Page_Boundary* (see Figure 49-39).

.

49.5 App Note: Order Enforcement Using the PCIe Core

49.5.1 PCIe Ordering Rule Overview

This application note is written for applications that require certain ordering rules within PCIe traffic. There are many ordering rules according to the PCIe specification. The following is a general description from the PCIe 2.1. base spec.

PCIe 2.1 base specification defines the ordering requirements for PCI Express transactions. The ordering rules defined in the spec apply within a single traffic class (TC). There is no ordering requirement among transactions with different TC levels.

Root Complexes that support peer-to-peer operation and Switches must enforce these transaction ordering rules for all forwarded traffic. These forwarding devices should not forward traffic from one virtual channel to another.

Basic ordering rules are summarized as follows:

- Posted is permitted to pass posted transaction only if the relaxed order bit is set.
- Non-posted is permitted to pass or to be blocked by non-posted.
- Completion with different ID is permitted to pass or be blocked by other completions.
- Posted must be allowed to pass non-posted to avoid deadlock.
- Posted is permitted to be blocked by completions.
- Non-posted can never pass posted.
- Non-posted is permitted to pass or be blocked by completions.
- Completions can pass posted only if the relaxed order bit is set.
- Completions must be allowed to pass non-posted to avoid deadlock.

In general, ordering rules are used based on a specific device's functionality. Root Complexes with peer-to- peer support and switch devices must enforce the above rules. For endpoint devices, it is device-specific. For example, it is possible that an endpoint device is allowed to have completions passing posted and non- posted requests. Because most of the above rules are considered as "permitted or blocked", it is legal to have "permitted" or "blocked" be determined by the architecture of the PCIe core and AXI bridge. The DWC PCIe core and AXI bridge IP is designed such that the order rules may be enforced strictly or less strictly, based on the requirements of the application.

49.5.2 PCIe Core Inbound Order Enforcement

There are three receive queue architectures within the DWC PCIe core. These queue structures will determine different order enforcement based on different receive queue architectures.

If the core is configured with all transactions in bypass mode, then order enforcement should be performed by the application logic. The core will not be responsible for implementing PCIe order enforcement. Since it is an all bypass configuration, all transactions are presented onto the application interface in the order that they arrived.

Below are descriptions for different queueing architectures that are implemented in the PCIe IP core. Three different queue modes are configurable and are designed to suit different applications.

49.5.2.1 Single queue

In Single queue mode, a single header and data queue are populated in the DWC PCIe core for all transactions received. Completions can be bypassed if desired, based on the application. Single queue mode implements strong ordering (in order delivery). All transactions of the same or different traffic class are enqueued and dequeued in the order the transactions were received. For example, if a non-posted transaction is followed by a posted transaction, then the application interface (the DWC PCIe core's target1 interface) will have a non-posted transaction being presented, followed by a posted transaction. Even if the non-posted transaction is blocked by the application (halt), the posted transaction will not be allowed to bypass.

This queue mode has followed all ordering rules above except for rule #4 and rule #9 if completion is not in bypass. What this queuing mode implements is that posted transactions will be blocked by another posted, non-posted will be blocked by another non-posted, etc. That is, we have selected ìblockî as the default of the above ordering rules. According to the spec., these are permitted except for rule #4, and possibly rule #9, if the completion is not in bypass. Rule #4 and rule #9 exist to avoid deadlocks caused by credit starvation and are applicable only when the TLP will be forwarded to another PCI/ PCIe link. Because of these violations, this queue mode is not appropriate for a Root Complex (with peer-to-peer support) or for switch designs. This is designed for endpoint devices that have no order rule requirement at the transaction forwarding between the DWC PCIe core receive queue and the core application interface. The advantage of this queue mode is that it is simple, resulting in an area and power advantage.

49.5.2.2 Multiple queue

In Multiple queue mode, a header and data queue per type of transaction are populated in the DWC PCIe core. All transactions are enqueued in-order. All posted are enqueued into posted header/data queue. All non-posted are enqueued into a header/data non-posted queue and all completions are enqueued into a header/data completion queue. In multiple VC cases, each VC will have its own set of queues.

This queue mode has the same ordering as for single-queue mode. The user has further flexibility in setting which TLPs are bypassed and which TLPs are added to the ordering queue, but these additional modes offer little practical value in the normal case. Since it shares the same limitations as singe-queue mode, this queue mode is not likely appropriate for a Root Complex with peer-to-peer support or for switch designs This queue has some gate count advantage over the full order support provided by the segmented buffer; however, this advantage is generally minor, especially for multi-VC systems.

49.5.2.3 Segmented buffer queue

A single header/data queue are populated in the DWC PCIe core for this queue mode. Transactions of different types are queued into a segmented buffer; each TLP type into a separate segment in the buffer. The transaction's order information is also queued into the segmented buffer. All posted requests are enqueued into the posted segment, all non-posted requests are queued into the non-posted segment and all completions are enqueued into the completion segment. Each traffic class also has its own dedicated per- type segments.

All PCIe order enforcement rules are strictly followed. All transactions are normally served in the order of reception, if the application has no blocking (halt) of the transactions received. If the application does have blocking conditions , then the order rules are enforced for transactions being read out of the queue. For example, if a non-posted request is received followed by a posted request, and the application could not take any more non-posted requests, then the posted will be allowed to pass the non-posted request and the posted transaction will be presented onto the application interface. In a second example, a posted request is received followed by a non-posted request, and the application can not take any more posted requests, then the non-posted transaction will be blocked even if the application can take the next non-posted transaction. Only when there are no previously received posted requests will the non-posted transaction be presented onto the application interface.

Since the order information is stored along with each transaction, full PCIe order enforcement can be accomplished through this mode of operation. This is the difference from the single and multiple queue architectures.

Segmented buffer queue mode is designed for Switches and Root Complexes with peer-to-peer support. The primary advantage of this mode is the full support of PCIe ordering rules. In addition, this mode offers a reduced RAM requirement when compared with multi-queue mode, especially for multi-VC designs.

The figure below represents the various queue architectures of the DWC PCIe IP core inbound receiving traffic. There are three different queue modes. As described above, the ordering rules are implemented differently, based on different queue architecture.

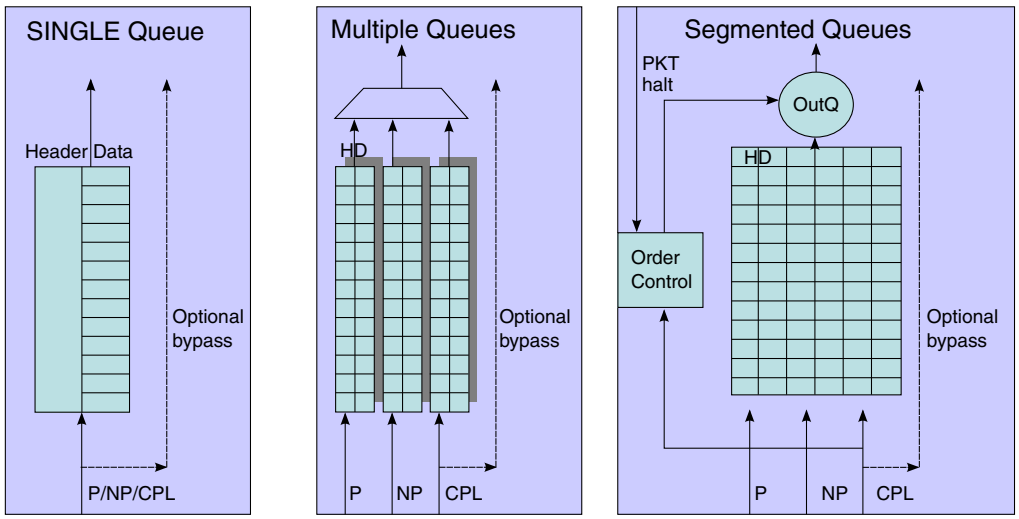


Figure 49-40. DWC PCIe Core Inbound Traffic Queue Architectures

49.5.3 PCIe Core Outbound Order Enforcement

The PCIe Core has a basic transmit architecture such that all transactions are presented outbound onto the PCIe wire based on the order the application presents them onto the PCIe core transmit interfaces.

There are up to three application outbound transmit interfaces (client0, client1 and optionally, client2). All client interfaces are served default as round robin arbitration if credit is available, regardless of the type of transaction. For example, if a posted transaction is presented onto client1 followed by a completion transaction, and if credits permit, then the posted transaction will be transmitted onto the wire before the completion. If the credit is not available, then the completion can pass the posted and be sent onto the wire.

It is the responsibility of the application to make appropriate use of the three client interfaces. An application that has three independent threads of traffic can use the three client interfaces such that the PCIe core will do the arbitration for the outbound transmission. For example, if the application has one source of outbound read transactions and one source of outbound write transactions, and they are independent sources, then client0 and client1 interfaces of the PCIe core should be used to perform the outbound transmission of the transactions. If the read and write are related, then one single interface client0 should be used and it is up to the application to maintain the desired order.

Most applications have independent threads of inbound reads such that its completion is not related to the outbound reads and writes. With these, an independent application interface such as client2 should be used for outbound completions.

In general, there is no transmit queuing in the PCIe core. Therefore, there is no order enforcement among outbound transactions within the transmit queue. The PCIe core has a cut-through transmit architecture where all outbound transaction order rules are enforced by the application. The PCIe core transmits TLPs in the order they are accepted by the core.

Endpoint devices can use two or three client interfaces for posted, non-posted and completion outbound traffic. The two or three client interfaces are determined by whether there is a dependency between the posted and non-posted traffic.

Root Complex and Switch devices will likely use one client interface where the order enforcement is performed by the application logic, where the outbound transfer queue is near to the transmit interface. The credit information is an output of the PCIe core, where it can be used as an input for taking outbound transactions out of the application's queue.

The figure below shows the outbound traffic queue architecture of the PCIe core. As shown in the figure, there is no transmit queue implemented in the core. There is just a retry buffer for replaying PCIe traffic as required by the PCIe base specification. Therefore, outbound TLPs will pass directly from the client interface through the core and to the wire in the same order they were accepted from the application.

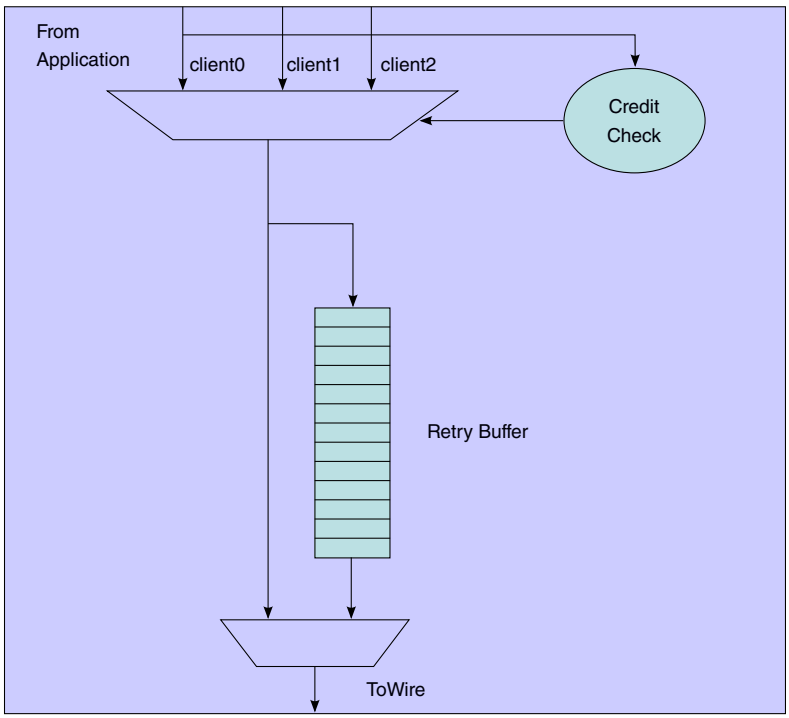


Figure 49-41. PCIe Core Outbound Traffic Queue Architectures

49.5.4 PCIe AHB/AXI Bridge Order Enforcement

Order enforcement through the AXI/AHB bridges is discussed in the following sections:

- Inbound Order Enforcement for AXI Bridge
- Additional Information (ordering)

49.5.5 Additional Information

Additional information is available at:

- Queuing Architecture.
- Queue Modes.
- Order Enforcement.

49.6 App Note: Calculating Gen1 PCI Express and AXI Bridge Throughput

This application note defines the throughput calculation (primarily) with respect to the PCI Express core in Gen1 2.5 GT/s mode.

NOTE

The max transfer size is 128bits

49.6.1 PCI Express Throughput

PCI Express bandwidth is 2.5 Gb/s (gigabits per second), per lane, when operating at the Gen 1 data rate. Because data is encoded using 8b10b encoding, the effective maximum throughput is 250 MB/s (megabytes per second), per lane (and overall throughput since we have 1 lane) , calculated as follows:

2.5 Gb * 8b/10b = 2 Gb * 1B/8b = 250 MB/sec per lane

49.6.2 Effective Throughput

The effective throughput is the payload throughput once all PCIe protocol's overheads have been factored out. The key protocol features are:

- 8b10b encoding at the physical layer. This takes away 20% of the raw bandwidth.
- Acknowledge and flow control update packets at the data link layer (DLLPs). This takes away 1% to 5% of the remaining bandwidth.
- Packet overhead at the transaction layer. Your design choices have a great effect here:
  - Small packets may take away 75% of the bandwidth!
  - Large packets may take away as little as 1%.

49.6.2.1 Effective Throughput Calculation

The table below identifies the TLP package.

Table 49-32. TLP Packet (with associated Data Link Layer overhead bytes)

STP 1 Byte	SEQ 2 Bytes	TLP Header 12/16 Bytes	Data Payload	ECRC 4 Bytes	LCRC 4 Bytes	END 1 Byte

49.6.2.1.1 Packet Level: (Start and End, Link CRC, Header)

The table below summarizes throughput calculations based on a payload size from 16 to 4K bytes.

Table 49-33. Effective Throughput

	Payload Bytes	Header Bytes	ECRC Bytes	PHY and Data Link Layer Bytes	Calculation	Percent Throughput	Note
Worst Packet	16	16	4	8	16/44	36%	Has ECRC.
Typical Packet	128	16	0	8	128/152	84%	No ECRC.
Typical Packet	256	16	0	8	256/280	91%	No ECRC.
Typical Packet	512	16	0	8	512/536	96%	No ECRC.
Best Packet	4096	12	0	8	4096/4116	99%	No ECRC.

NOTE

A 128-byte payload size yields about 67% of the net throughput. Increasing the payload size to 512 bytes increases the net throughput to 92%. Increasing the payload size from 512 bytes to 4096 bytes only contributes an increase of 8% in the net throughput, and the storage requirements are more than doubled. In addition, a large payload size may have an impact on performance due to re- transmission of TLPs. Therefore, 256-byte and 512-byte payload sizes are the most popular choices. Most chipsets support 128 -byte or 256-byte payload sizes, with 512 bytes gaining in popularity.

49.6.2.1.2 Link Layer: (Flow Control and ACK/NAK DLLPs)

DLLPs should be sent as often as required to avoid a negative impact on the TLP throughput. The core sends a pending DLLP if there is no competing TLP traffic.

Sending ACK/NAK and Update FC is controlled by timers. The core provides flexibility to fine-tune these as required. The default setup is minimal flow control (one per time-out) and minimal ACK/NAK device latencies. The link partners retry buffer (payload) size may require a change in the ACK/NAK/Update FC time-out to obtain optimal system latencies.

The core provides a feature to accumulate up to 255 ACKs before issuing an ACK. This feature offers-fine tuning of ACK frequency impact. The core transmits ACKs at fixed intervals, by default.

The following table identifies the DLLP package.

Table 49-34. DLLP Package

STP	Type	Data	16b CRC	END
1 Byte	1 Byte	4 Bytes	2 Bytes	1 Byte

The following table summarizes the throughput calculations.

Table 49-35. Effective Throughput

	Bytes	Result	Percent Throughput
Worst Packet	One ACK plus one FC per 128 (8 packets of 16) bytes of data => 2 DLLPs per 8 data packets	(8*44) / (8*44 + 2*8)	95%
Typical Packet	One ACK plus one FC per 1.4 (256 byte payload) bytes of data => 2 DLLPs per 1.4 packets	1.4*280 / (1.4*280 + 2*8)	96%
Best Packet	One ACK plus one FC per 4096 bytes of data => 2 DLLP per data packet	4116 / (4116 + 2*8)	99%

49.6.2.2 Other Factors Impacting Throughput

Replay buffer and receiver queue sizing should be optimal to avoid a harmful impact of larger ACK/NAK or Update FC latencies by the link partner. The core does a very effective autosizing of the buffers taking into consideration the impact of these parameters.

Effective Throughput (ET) = Raw Throughput * (NL x 2.5 Gb)

Consideration of lane width and payload size results in the following:

Table 49-36. Consideration of lane width and payload size

Factor	Throughput
8b10b	*(80%)
Link Packet	*(95% to 99%)
Data Payload	*(36% to 99%)

The table below identifies lane width and payload vs. throughput.

Table 49-37. Lane Width and Payload vs. Throughput

	Real Throughput Gb/s vs. Data Payload			
Lane Width	16	128	256	4096
x1	0.5	1.7	1.7	2.0

49.7 PCIe Registers (EP mode)

Information found here describes the core implementation of the PCI Express configuration space in the core in EP mode.

49.7.1 Register Space Layout (EP mode)

The core has 4096 bytes of PCI Express configuration space per function.

For each function, the PCI Express configuration space is divided into:

- 256 bytes of PCI Configuration Space, containing:
- 64 bytes of PCI 3.0 Compatible Configuration Space Header (type 0 in EP mode)
- PCI Standard Capabilities Structures linked list, which can start anywhere after offset 0x40
- 3840 bytes of PCI Express Extended Configuration Space (which starts at offset 0x100), containing:
- PCI Express Extended Capabilities Structures linked list, which starts at offset 0x100
- Port Logic registers (vendor-specific registers), which start at offset 0x700. The Port Logic registers have specific pre-defined usages, mostly for test purposes, and can optionally be removed from the core hardware configuration. The usage of the Port Logic registers is the same in both EP mode and RC mode. See PCIe Registers: Port Logic for details.

The figure below shows the EP mode layout of the core configuration space.

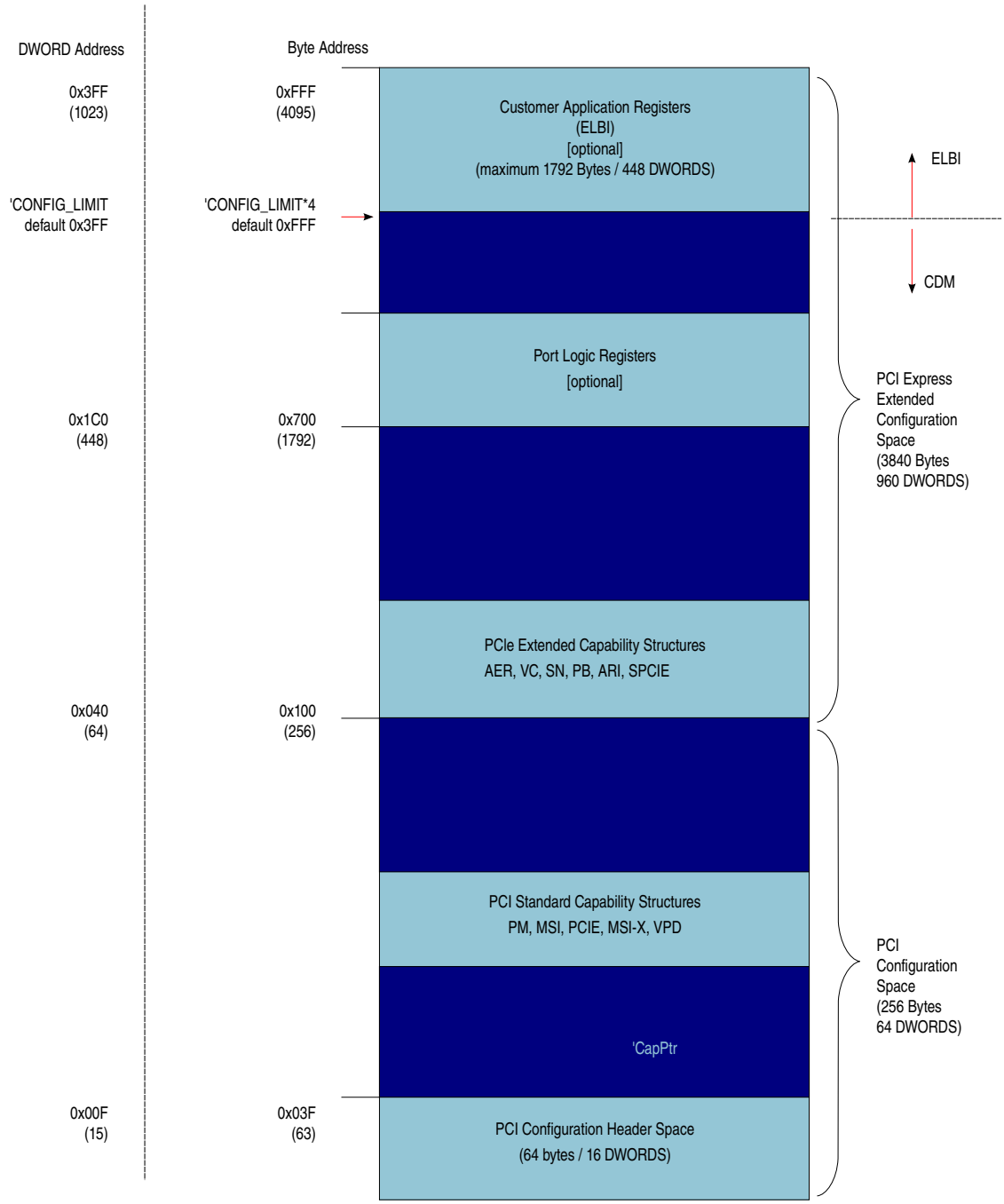


Figure 49-42. Core Configuration Space Layout (EP Mode)

49.7.2 PF Register Maps

Capability configuration registers are in structures (groups) identified by a Capability ID.

Groups are linked together as in PCI. Register locations within a group are specified, but the starting location of each group must be found by traversing the linked list. There are two linked lists of register groups: PCI Compatible Capability registers and PCI Express Extended Capability registers. PCI Compatible Capability register groups begin at the configuration address stored in the capability pointer register at 0x34. PCI Express Extended Capability register groups begin at address 0x100.

49.7.2.1 PF PCI Configuration Space Header - Type 0

The table below shows the layout of the Type 0 Configuration Space Header. Most PCI-compatible register fields have the same software interpretation in PCI 3.0 and PCI Express.

Table 49-38. PF PCI Configuration Space Header - Type 0

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0x00	Device ID		Vendor ID	
0x04	Status Register		Command Register	
0x08	Class Code			Revision ID
0x0C	BIST(0x00)	Header Type	Latency Timer	Cache Line Size
0x10	Base Address Register 0			
0x14	Base Address Register 1			
0x18	Base Address Register 2			
0x1C	Base Address Register 3			
0x20	Base Address Register 4			
0x24	Base Address Register 5			
0x28	CardBus CIS Pointer			
0x2C	Subsystem ID		Subsystem Vendor ID	
0x30	Expansion ROM Base Address			
0x34	Reserved			CapPtr
0x38	Reserved			
0x3C	Max_Latency[1]	Min_Grant[1]	Interrupt Pin	Interrupt Line

1. The Max_Latency and Min_Grant registers do not apply to PCI Express and are read-only registers with values hardwired to 0x00.

49.7.2.2 PF PCI Standard Capability Structures Register Maps

The Capability Pointer register in the PCI-compatible header register points to the next item in the linked list of capabilities, which, by default, is the PCI Power Management capabilities register space.

NOTE

Even though there is an unique standard capabilities linked lists provided per function, specific capabilities cannot be enabled/disabled on a per-function basis; for example, MSI-X capability is enabled/disabled for all functions (PF) at the same time through the coreConsultant GUI.

NOTE

Each function (PF) may have a different configuration of that capability structure once it is enabled, although some features/settings are common across all functions.

The following tables list the capabilities supported by the core and their respective default address offsets and next capability pointers and provides the default values of the default address offsets.

Table 49-39. PF Configuration Structure: Starting Addresses and Next Capability Pointers

Start Address Offset	Item	Next Pointer

Table 49-40. PF Default Values of Parameters that Define Starting Addresses

Parameter	Default Value
CFG_PM_CAP	8'h40
CFG_MSI_CAP	8'h50
CFG_PCIE_CAP	8'h70

The following tables show the PCI Standard Capability Structures.

Table 49-41. PF Power Management Capability Structure

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
CFG_PM_CAP	Power Management Capabilities (PMC)		Next Capability Pointer (PM_NEXT_PTR)	Capability ID (0x01)
+0x4	Data	PMCSR_BSE Bridge Extensions	Power Management Control Status Register (PMCSR)	

Table 49-42. PF MSI Capability Structure

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
+0x4	MSI Lower 32-bit Address Register			
+0x8	MSI Upper 32-bit Address Register			
+0xC	Reserved		MSI Data	

Table continues on the next page...

Table 49-42. PF MSI Capability Structure (continued)

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
+0x10	Mask Bits Register			
+0x14	Pending Bits Register			

Table 49-43. PF PCI Express Capability Structure

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
+0x4	Device Capabilities			
+0x8	Device Status		Device Control	
+0xC	Link Capabilities			
+0x10	Link Status		Link Control	
+0x24	Device Capabilities 2			
+0x28			Device Control 2	
+0x2C	Link Capabilities 2			
+0x30	Link Status 2		Link Control 2	

49.7.2.3 PF PCI Express Extended Capability Register Maps

The PCI Express Extended Capabilities registers are located in device configuration space at offsets 0x100 or higher. As with PCI Standard Capability Structures, the PCI Express Extended Capability structures are allocated using a linked list with a similar method and format to those of PCI.

NOTE

Even though there is a unique extended capabilities linked list provided per function, specific capabilities cannot be enabled/disabled on a per function basis.

NOTE

Each function (PF) may have a different configuration of that capability structure once it is enabled, although some features/settings are common across all functions.

The Advanced Error Reporting (AER) Capability and Virtual Channel (VC) Capability are optional extended capabilities that may be implemented by PCI Express devices supporting advanced error control and reporting and multiple VCs, respectively. The Advanced Error Reporting Capability is required when the device supports ECRC generation/checking. The Virtual Channel Capability is required for any device that supports multiple VCs and/or multiple Traffic Classes (TCs).

The Next Capability Pointer register in the PCI Express Extended Capability Structures Register Maps points to the next item in the linked list of capabilities, which, by default, is the Advanced Error Reporting (AER) Capability register space.

The table below lists the extended capabilities supported by the core and their respective default address offsets and next capability pointers.

Table 49-44. PF Configuration Structure: Starting Addresses and Next Capability Pointers

Start Address Offset	Item	Next Pointer

The following tables outline the PCI Express Extended Capabilities structures.

Table 49-45. PF Advanced Error Reporting (AER) Capability Structure

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
+0x4	Uncorrectable Error Status Register			
+0x8	Uncorrectable Error Mask Register			
+0xC	Uncorrectable Error Severity Register			
+0x10	Correctable Error Status Register			
+0x14	Correctable Error Mask Register			
+0x18	Advanced Error Capabilities and Control Register			
+0x1C through +0x28	Header Log Registers			

Table 49-46. PF Virtual Channel Capability Structure

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
+0x4	Port VC Capability Register 1			
+0x8	Port Capability Register 2			
+0xC	Port VC Status Register		Port VC Control Register	
+0x10	VC Resource Capability Register (0)			
+0x14	VC Resource Control Register (0)			
+0x18	VC Resource Status Register (0)		RsvdP	
0x10+(N*0x0C)	VC Resource Capability Register (N)[1]			
0x14+(N*0x0C)	VC Resource Control Register (N)			
0x18+(N*0x0C)	VC Resource Status Register (N)		RsvdP	

1. There is one VC Resource Capability/Control/Status Register N set for each configured VC (in addition to VC0).

1. Depends on the number of VCs

49.7.3 VF Register Maps

Similar to PFs, VFs also have Type 0 Configuration Space Header, a linked list of PCI Compatible Capabilities and a linked list of PCI Express Extended Capabilities. The following PCI Compatible Capabilities are supported for VFs: PCI Express, PM (optional), MSI (optional), MSI-X. The following PCI Express Extended Capabilities are supported for VFs: ARI, AER (optional).

49.7.3.1 VF PCI Configuration Space Header - Type 0

The table below shows the layout of the Type 0 Configuration Space Header for Virtual Functions.

Table 49-47. VF PCI Configuration Space Header - Type 0

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0x00	Device ID		Vendor ID	
0x04	Status Register		Command Register	
0x08	Class Code			Revision ID
0x0C	BIST(0x00)	Header Type	Latency Timer	Cache Line Size
0x10	Base Address Register 0			
0x14	Base Address Register 1			
0x18	Base Address Register 2			
0x1C	Base Address Register 3			
0x20	Base Address Register 4			
0x24	Base Address Register 5			
0x28	CardBus CIS Pointer			
0x2C	Subsystem ID		Subsystem Vendor ID	
0x30	Expansion ROM Base Address			
0x34	Reserved			CapPtr
0x38	Reserved			
0x3C	Max_Latency[1]	Min_Grant[1]	Interrupt Pin	Interrupt Line

1. The Max_Latency and Min_Grant registers do not apply to PCI Express and are read-only registers with values hardwired to 0x00.

49.7.3.2 VF PCI Standard Capability Structures Register Maps

NOTE

All VFs have the same linked list of capabilities.

The Capability Pointer register in the Type 0 Header points to the next item in the linked list of capabilities, which, by default, is the PCI Express Capability.

The tables below list the VF standard capabilities supported by the core and their respective default address offsets and next capability pointers and provides the default values of the default address offsets.

Table 49-48. VF Configuration Structure: Starting Addresses and Next Capability Pointers

Start Address Offset	Item	Next Pointer

Table 49-49. VF Default Values of Parameters that Define Starting Addresses

Parameter	Default Value
`CFG_PCIE_CAP	8'h70

The table below shows the VF PCI Express Capability Structure.

Table 49-50. VF PCI Express Capability Structure

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
+0x4	Device Capabilities			
+0x8	Device Status		Device Control	
+0xC	Link Capabilities			
+0x10	Link Status		Link Control	
+0x24	Device Capabilities 2			
+0x28			Device Control 2	
+0x30	Link Status 2		Link Control 2	

49.7.4 Accessing Configuration Registers

The application can access the configuration space through the DBI. Bits [11:0] of the DBI address bus select the target register. Host software accesses the configuration registers through PCI Express Configuration Requests.

The Attribute (Attr) column in each register description indicates the read/write access for the register or bit. The table below defines the read/write attribute abbreviations that are used in the register and bit descriptions throughout this chapter.

Table 49-51. Configuration Register Bit-Field Types

Attribute	Description
HwInit	Hardware Initialized HwInit bits are controlled by core hardware and are read-only (RO) by host system software. These bits can only be reset with cold reset. They are not modified by an FLR, or by a warm or hot reset.
RO	Read-Only Register bits are read-only and cannot be altered by software. Register bits are permitted to be initialized by core hardware.
RW	Read-Write Register bits are read-write and may be read and written normally by the host and the application. Writing from the application side (if any) requires careful synchronization with the host software.
RW1C	Read-Only Status/Write-1-to-Clear Status Register bits indicate status when read. A set bit indicates a status event may is cleared by writing a 1. Writing 0 to RW1C bits has no effect. Writing from the application side (if any) requires careful synchronization with host software.
ROS	Sticky Read-Only Register bits are read-only and cannot be altered by host or application software, except as noted. Registers are not initialized or modified by a hot reset or FLR. A few bits designated as very sticky are not cleared by any type of core reset when auxiliary power is supplied and enabled.
RWS	Sticky Read-Write Register bits are read-write and are set or cleared by host or application software to the desired state. Bits are not initialized or modified by a hot reset or FLR. A few bits designated very sticky are not cleared by any type of core reset when auxiliary power is supplied and enabled.
RW1CS	Sticky Read-Only Status/Write-1-to-Clear Status Register bits indicate status when read. A set bit indicates a status event which is cleared by writing a 1, except as noted. Writing 0 to RW1CS bits has no effect. Bits are not initialized or modified by a hot reset or FLR. A few bits designated very sticky are not cleared by any type of core reset when auxiliary power is supplied and enabled.
RsvdP	Reserved and Preserved Reserved for future RW implementations. Registers are read-only and return zero when read. Software must preserve the value read when writing to other bits in the same register.
RsvdZ	Reserved and Zero Reserved for future RW1C implementations. Registers are read-only and return zero when read. Software must write 0 to these bits when writing to other bits in the same register.
PF	Physical Function Indicates that this Virtual Function (VF) register bit inherits it's value from the corresponding parent Physical Function.

Most of the these registers are in the non-sticky reset domain. Non-sticky registers should be reset after a cold, warm or hot reset. .

49.8 PCIe Registers (RC mode)

Information found here describes the core implementation of the PCI Express configuration space when in RC mode.

Register definition applies to all cases unless otherwise specified.

49.8.1 Register Space Layout

The core has 4096 bytes of PCI Express configuration space per function.

For each function, the PCI Express configuration space is divided into:

- 256 bytes of PCI Configuration Space, containing:
- 64 bytes of PCI 3.0 Compatible Configuration Space Header (type 1)
- PCI Standard Capabilities Structures linked list, which can start anywhere after offset 0x40
- 3840 bytes of PCI Express Extended Configuration Space (which starts at offset 0x100), containing:
- PCI Express Extended Capabilities Structures linked list, which starts at offset 0x100
- Port Logic registers (vendor-specific registers), which start at offset 0x700. The Port Logic registers have specific pre-defined usages, mostly for test purposes, and can optionally be removed from the core hardware configuration. The usage of the Port Logic registers is the same in both EP mode and RC mode. See PCIe Registers: Port Logic for details.

The figure below shows the layout of the core configuration space.

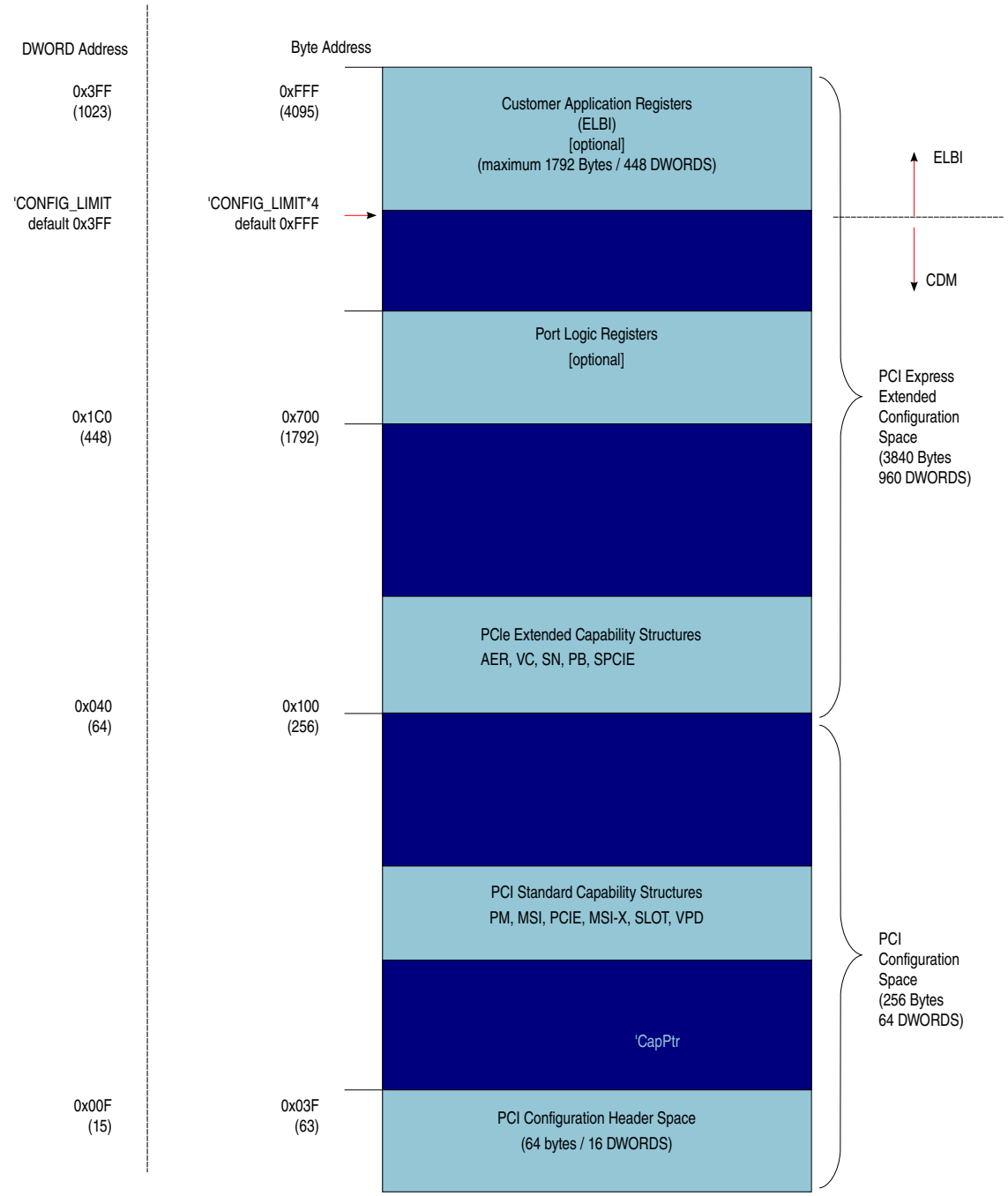


Figure 49-43. Core Configuration Space Layout: (RC Mode)

49.8.2 Register Maps

Configuration registers are in structures (groups) identified by a Capability ID. Groups are linked together as in PCI.

Register locations within a group are specified, but the starting location of each group must be found by traversing the linked list. There are two linked lists of register groups: PCI-compatible base registers and PCI Express Extended Capability registers. PCI-compatible base register groups begin at configuration address stored in capability pointer register at 0x34. PCI Express Extended Capability register groups begin at address 0x100.

Table below shows the configuration field register definitions for PCI Express Type 1 Configuration Space header. Most PCI-compatible register fields have the same software interpretation in PCI 3.0 and PCI Express.

Table 49-52. PCI Configuration Space Header - Type 1

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0x00	Device ID		Vendor ID	
0x04	Status Register		Command Register	
0x08	Class Code			Revision ID
0x0C	BIST(0x00)	Header Type	Latency Timer	Cache Line Size
0x10	Base Address Register 0			
0x14	Base Address Register 1			
0x18	Secondary Latency Timer	Subordinate Bus Number	Secondary Bus Number	Primary Bus Number
0x1C	Secondary Status		I/O Limit	I/O Base
0x20	Memory Limit		Memory Base	
0x24	Prefetchable Memory Limit		Prefetchable Memory Base	
0x28	Prefetchable Base Upper 32 Bits			
0x2C	Prefetchable Limit 32 Upper Bits			
0x30	I/O Limit Upper 16 Bits		I/O Base Upper 16 Bits	
0x34	Reserved			CapPtr
0x38	Expansion ROM Base Address			
0x3C	Bridge Control		Interrupt Pin	Interrupt Line

49.8.2.1 PCI Standard Capability Structures Register Maps

The Capability Pointer register in the PCI-compatible header register points to the next item in the linked list of capabilities, which, by default, is the PCI Power Management capabilities register space.

Even though there is an unique standard capabilities linked lists provided per function, specific capabilities cannot be enabled/disabled on a per-function basis; for example, MSI-X capability is enabled/disabled for all functions at the same time through the coreConsultant GUI.

Each function may have a different configuration of that capability structure once it is enabled, although some features/settings are common across all functions.

Table below lists the capabilities supported by the core and their respective default address offsets and next capability pointers. Table 49-54 provides the default values of the default address offsets.

Table 49-53. Configuration Structure: Starting Addresses and Next Capability Pointers

Start Address Offset	Item	Next Pointer

Table 49-54. Default Values of Parameters that Define Starting Addresses

Parameter	Default Value
CFG_PM_CAP	8'h40
CFG_MSI_CAP	8'h50
CFG_PCIE_CAP	8'h70

The following tables show the PCI Standard Capability structures.

Table 49-55. Power Management Capability Structure

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
+0x4	Data	PMCSR_BSE Bridge Extensions	Power Management Control Status Register (PMCSR)	

Table 49-56. MSI Capability Structure

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
+0x4	MSI Lower 32-bit Address Register			
+0x8	MSI Upper 32-bit Address Register			
+0xC			MSI Data	

Table 49-57. PCI Express Capability Structure

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
+0x4	Device Capabilities			
+0x8	Device Status		Device Control	
+0xC	Link Capabilities			

Table continues on the next page...

Table 49-57. PCI Express Capability Structure (continued)

+0x10	Link Status	Link Control
+0x14	Slot Capabilities	
+0x18	Slot Status	Slot Control
+0x1C	Root Capabilities	Root Control
+0x20	Root Status	
+0x24	Device Capabilities 2	
+0x28	Device Status 2	Device Control 2
+0x2C	Link Capabilities 2	
+0x30	Link Status 2	Link Control 2
+0x34	Slot Capabilities 2	
+0x38	Slot Status 2	Slot Control 2
1. Slot Capabilities apply only to downstream ports; for example, RC.		

49.8.2.2 PCI Express Extended Capability Register Maps

The PCI Express Extended Capabilities registers are located in device configuration space at offsets 0x100 or higher. As with PCI Capabilities, the PCI Express Extended Capability structures are allocated using a linked list with a similar method and format to those of PCI.

Even though there is a unique extended capabilities linked list provided per function, specific capabilities cannot be enabled/disabled on a per function basis.

Each function may have a different configuration of that capability structure once it is enabled, although some features/settings are common across all functions.

The Advanced Error Reporting Capability and Virtual Channel Capability are optional extended capabilities that may be implemented by PCI Express devices supporting advanced error control and reporting and multiple VCs, respectively. The Advanced Error Reporting Capability is required when the device supports ECRC generation/checking. The Virtual Channel Capability is required for any device that supports multiple VCs and/or multiple Traffic Classes (TCs).

The Next Capability Pointer register in the PCI Express Extended Capability Structures Register Maps points to the next item in the linked list of capabilities, which, by default, is the Advanced Error Reporting (AER) Capability register space.

Table below lists the capabilities supported by the core and their respective default address offsets and next capability pointers.

Table 49-58. Configuration Structure: Starting Addresses and Next Capability Pointers

Start Address Offset	Item	Next Pointer

The following tables outline the PCI Express Extended Capabilities structures.

Table 49-59. Advanced Error Reporting Capability Structure

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
+0x4	Uncorrectable Error Status Register			
+0x8	Uncorrectable Error Mask Register			
+0xC	Uncorrectable Error Severity Register			
+0x10	Correctable Error Status Register			
+0x14	Correctable Error Mask Register			
+0x18	Advanced Error Capabilities and Control Register			
+0x1C through +0x28	Header Log Registers			
+0x2C	Root Error Command Register			
+0x30	Root Error Status Register			
+0x34	Error Source Identification Register			

Table 49-60. Virtual Channel Capability Structure

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
+0x4	Port VC Capability Register 1			
+0x8	Port Capability Register 2			
+0xC	Port VC Status Register		Port VC Control Register	
+0x10	VC Resource Capability Register (0)			
+0x14	VC Resource Control Register (0)			
+0x18	VC Resource Status Register (0)		RsvdP	
0x10+(N*0x0C)	VC Resource Capability Register (N)1			
0x14+(N*0x0C)	VC Resource Control Register (N)			
0x18+(N*0x0C)	VC Resource Status Register (N)		RsvdP	

1. There is one VC Resource Capability/Control/Status Register N set for each configured VC (in addition to VC0).

49.8.3 Accessing Configuration Registers (Configuration)

The application can access the configuration space through the DBI. Bits [11:0] of the DBI address bus select the target register.

The Attribute column in each register description indicates the read/write access for the register or bit. Table below defines the read/write attribute abbreviations that are used in the register and bit descriptions throughout this chapter. The definitions match the PCI Express 3.0 Specification; in the case of a Root Port, all accesses are through the DBI so the host-access information does not apply.

Table 49-61. Configuration Register Bit-Field Types

Attribute	Description
HwInit	Hardware Initialized HwInit bits are controlled by core hardware and are read-only (RO) by host system software. Some HwInit bits are writable from the application through the DBI (if `CX_DBI_RO_WR_EN = 1), as indicated in the register descriptions. If `CX_DBI_RO_WR_EN = 0, none of the HwInit bits are writable through the DBI. These bits can only be reset with cold reset. They are not modified by an FLR, or by a warm or hot reset.
RO	Read-Only Register bits are read-only and cannot be altered by software. Register bits are permitted to be initialized by core hardware. Some RO bits are writable from the application through the DBI (if `CX_DBI_RO_WR_EN = 1), as indicated in the register descriptions. If `CX_DBI_RO_WR_EN = 0, none of the RO bits are writable through the DBI.
RW	Read-Write Register bits are read-write and may be read and written normally by the host and the application. Writing from the application side (if any) requires careful synchronization with the host software.
RW1C	Read-Only Status/Write-1-to-Clear Status Register bits indicate status when read. A set bit indicates a status event may is cleared by writing a 1. Writing 0 to RW1C bits has no effect. Writing from the application side (if any) requires careful synchronization with host software.
ROS	Sticky Read-Only Register bits are read-only and cannot be altered by host or application software, except as noted. Registers are not initialized or modified by a hot reset or FLR. A few bits designated as very sticky are not cleared by any type of core reset when auxiliary power is supplied and enabled.
RWS	Sticky Read-Write Register bits are read-write and are set or cleared by host or application software to the desired state. Bits are not initialized or modified by a hot reset or FLR. A few bits designated very sticky are not cleared by any type of core reset when auxiliary power is supplied and enabled.
RW1CS	Sticky Read-Only Status/Write-1-to-Clear Status Register bits indicate status when read. A set bit indicates a status event which is cleared by writing a 1, except as noted. Writing 0 to RW1CS bits has no effect.

Table continues on the next page...

Table 49-61. Configuration Register Bit-Field Types (continued)

	Bits are not initialized or modified by a hot reset or FLR. A few bits designated very sticky are not cleared by any type of core reset when auxiliary power is supplied and enabled.
RsvdP	Reserved and Preserved Reserved for future RW implementations. Registers are read-only and return zero when read. Software must preserve the value read when writing to other bits in the same register.
RsvdZ	Reserved and Zero Reserved for future RW1C implementations. Registers are read-only and return zero when read. Software must write 0 to these bits when writing to other bits in the same register.
PF	Physical Function Indicates that this Virtual Function (VF) register bit inherits it's value from the corresponding parent Physical Function.

Some RO bits are writable from the application through the DBI

Most of the these registers are in the non-sticky reset domain. Non-sticky registers should be reset after a cold, warm or hot reset.

49.9 PCIe Registers: Port Logic

Information found here describes the Port Logic register map and the usage of each Port Logic register.

49.9.1 Overview (Port Logic)

The Port Logic (PL) registers are vendor-specific registers and are used mainly for configuration of PCIe core implementation features, status reporting and testing.

The usage of the Port Logic registers is the same in both EP and RC modes and in the Switch. The Port Logic registers reside in the application register section of the configuration space starting at address 0x700.

49.9.2 Non-Standard Addressing of the iATU Port Logic Registers

See PCIe CTRL Port Logic Memory Map/Register Definition iATU Registers are programmed through an index (Viewport) register to reduce the memory footprint in the PCI Express Extended Configuration Space.

49.9.3 Accessing Configuration Registers (Port Logic Registers)

The application can access the port logic configuration space through the DBI. Bits [11:0] of the DBI address bus select the target register. Host software accesses the configuration registers through PCI Express Configuration Requests.

The Attribute (Attr) column in each register description indicates the read/write access for the register or bit. The table below defines the read/write attribute abbreviations that are used in the register and bit descriptions throughout this chapter.

Table 49-62. Configuration Register Bit-Field Types

Attribute	Description
HwInit	Hardware Initialized HwInit bits are controlled by core hardware and are read-only (RO) by host system software. None of the HwInit bits are writable through the DBI. These bits can only be reset with cold reset. They are not modified by an FLR, or by a warm or hot reset.
RO	Read-Only Register bits are read-only and cannot be altered by software. Register bits are permitted to be initialized by core hardware. None of the RO bits are writable through the DBI.
RW	Read-Write Register bits are read-write and may be read and written normally by the host and the application. Writing from the application side (if any) requires careful synchronization with the host software.
RW1C	Read-Only Status/Write-1-to-Clear Status Register bits indicate status when read. A set bit indicates a status event may is cleared by writing a 1. Writing 0 to RW1C bits has no effect. Writing from the application side (if any) requires careful synchronization with host software.
ROS	Sticky Read-Only Register bits are read-only and cannot be altered by host or application software, except as noted. Registers are not initialized or modified by a hot reset or FLR. A few bits designated as very sticky are not cleared by any type of core reset when auxiliary power is supplied and enabled.
RWS	Sticky Read-Write Register bits are read-write and are set or cleared by host or application software to the desired state.

Table continues on the next page...

Table 49-62. Configuration Register Bit-Field Types (continued)

Attribute	Description
	Bits are not initialized or modified by a hot reset or FLR. A few bits designated very sticky are not cleared by any type of core reset when auxiliary power is supplied and enabled.
RW1CS	Sticky Read-Only Status/Write-1-to-Clear Status Register bits indicate status when read. A set bit indicates a status event which is cleared by writing a 1, except as noted. Writing 0 to RW1CS bits has no effect. Bits are not initialized or modified by a hot reset or FLR. A few bits designated very sticky are not cleared by any type of core reset when auxiliary power is supplied and enabled.
RsvdP	Reserved and Preserved Reserved for future RW implementations. Registers are read-only and return zero when read. Software must preserve the value read when writing to other bits in the same register.
RsvdZ	Reserved and Zero Reserved for future RW1C implementations. Registers are read-only and return zero when read. Software must write 0 to these bits when writing to other bits in the same register.
PF	Physical Function Indicates that this Virtual Function (VF) register bit inherits it's value from the corresponding parent Physical Function.

Some of the Port Logic registers are in the sticky reset domain. Sticky registers should not be reset after a warm or hot reset.

49.10 PCIe CTRL EP Mode Memory Map/Register Definition

PCIE_EP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1FF_C000	Device ID and Vendor ID Register (PCIE_EP_DeviceID)	32	R	ABCD _16C3h	49.10.1/ 4278
1FF_C004	Command and Status Register (PCIE_EP_Command)	32	R/W	0000_0000h	49.10.2/ 4278
1FF_C00C	BIST Register (PCIE_EP_BIST)	32	R/W	0000_0000h	49.10.3/ 4281
1FF_C010	Base Address 0 (PCIE_EP_BAR0)	32	R	0000_000Ch	49.10.4/ 4282

Table continues on the next page...

PCIE_EP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1FF_C010	BAR 0 Mask Register (PCIE_EP_MASK0)	32	R	0000_000Ch	49.10.5/ 4284
1FF_C014	BAR 1 Mask Register (PCIE_EP_MASK1)	32	R	0000_0000h	49.10.6/ 4287
1FF_C018	BAR 2 Mask Register (PCIE_EP_MASK2)	32	R	0000_0008h	49.10.7/ 4288
1FF_C01C	BAR 3 Mask Register (PCIE_EP_MASK3)	32	R	0000_0000h	49.10.8/ 4289
1FF_C028	CardBus CIS Pointer Register (PCIE_EP_CISP)	32	R	0000_0000h	49.10.9/ 4290
1FF_C02C	Subsystem ID and Subsystem Vendor ID Register (PCIE_EP_SSID)	32	R	0000_0000h	49.10.10/ 4290
1FF_C030	Expansion ROM Base Address Register (PCIE_EP_EROMBAR)	32	R/W	0000_0000h	49.10.11/ 4291
1FF_C030	Expansion ROM BAR Mask Register (PCIE_EP_EROMMASK)	32	R/W	0000_0000h	49.10.12/ 4292
1FF_C034	Capability Pointer Register (PCIE_EP_CAPPR)	32	R	0000_0040h	49.10.13/ 4293
1FF_C03C	Interrupt Line and Pin Register (PCIE_EP_ILR)	32	R/W	0000_01FFh	49.10.14/ 4293
1FF_C100	AER Capability Header (PCIE_EP_AER)	32	R/W	0000_0000h	49.10.15/ 4294
1FF_C104	Uncorrectable Error Status Register (PCIE_EP_UESR)	32	R/W	0000_0000h	49.10.16/ 4295
1FF_C108	Uncorrectable Error Mask Register (PCIE_EP_UEMR)	32	R/W	0000_0000h	49.10.17/ 4297
1FF_C10C	Uncorrectable Error Severity Register (PCIE_EP_UESevR)	32	R/W	000C_2031h	49.10.18/ 4299
1FF_C110	Correctable Error Status Register (PCIE_EP_CESR)	32	R/W	0000_0000h	49.10.19/ 4301
1FF_C114	Correctable Error Mask Register (PCIE_EP_CEMR)	32	R/W	0000_0000h	49.10.20/ 4302
1FF_C118	Advanced Capabilities and Control Register (PCIE_EP_ACCR)	32	R/W	0000_00A0h	49.10.21/ 4304
1FF_C11C	Header Log Register (PCIE_EP_HLR)	32	R	0000_0000h	49.10.22/ 4305
1FF_C140	VC Extended Capability Header (PCIE_EP_VCECHR)	32	R	0000_0012h	49.10.23/ 4305
1FF_C144	Port VC Capability Register 1 (PCIE_EP_PVCCR1)	32	R	0000_0000h	49.10.24/ 4306
1FF_C148	Port VC Capability Register 2 (PCIE_EP_PVCCR2)	32	R	0000_0000h	49.10.25/ 4307
1FF_C14C	Port VC Control and Status Register (PCIE_EP_PVCCSR)	32	R/W	0000_0000h	49.10.26/ 4308

Table continues on the next page...

PCIE_EP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1FF_C150	VC Resource Capability Register n (PCIE_EP_VCRCR)	32	R	0000_0000h	49.10.27/ 4310
1FF_C154	VC Resource Control Register n (PCIE_EP_VCRConR)	32	R/W	0000_0000h	49.10.28/ 4312
1FF_C158	VC Resource Status Register n (PCIE_EP_VCRSR)	32	R	0000_0000h	49.10.29/ 4314

49.10.1 Device ID and Vendor ID Register (PCIE_EP_DeviceID)

Offset: 0x00

The default values of both Device ID and Vendor ID are hardware configuration parameters. The application can overwrite the default values of both Device ID and Vendor ID through the DBI.

Address: 1FF_C000h base + 0h offset = 1FF_C000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																Reserved															
Reset	1	0	1	0	1	0	1	1	1	1	0	0	1	1	0	1	0	0	0	1	0	1	1	0	1	1	0	0	0	0	1	1

PCIE_EP_DeviceID field descriptions

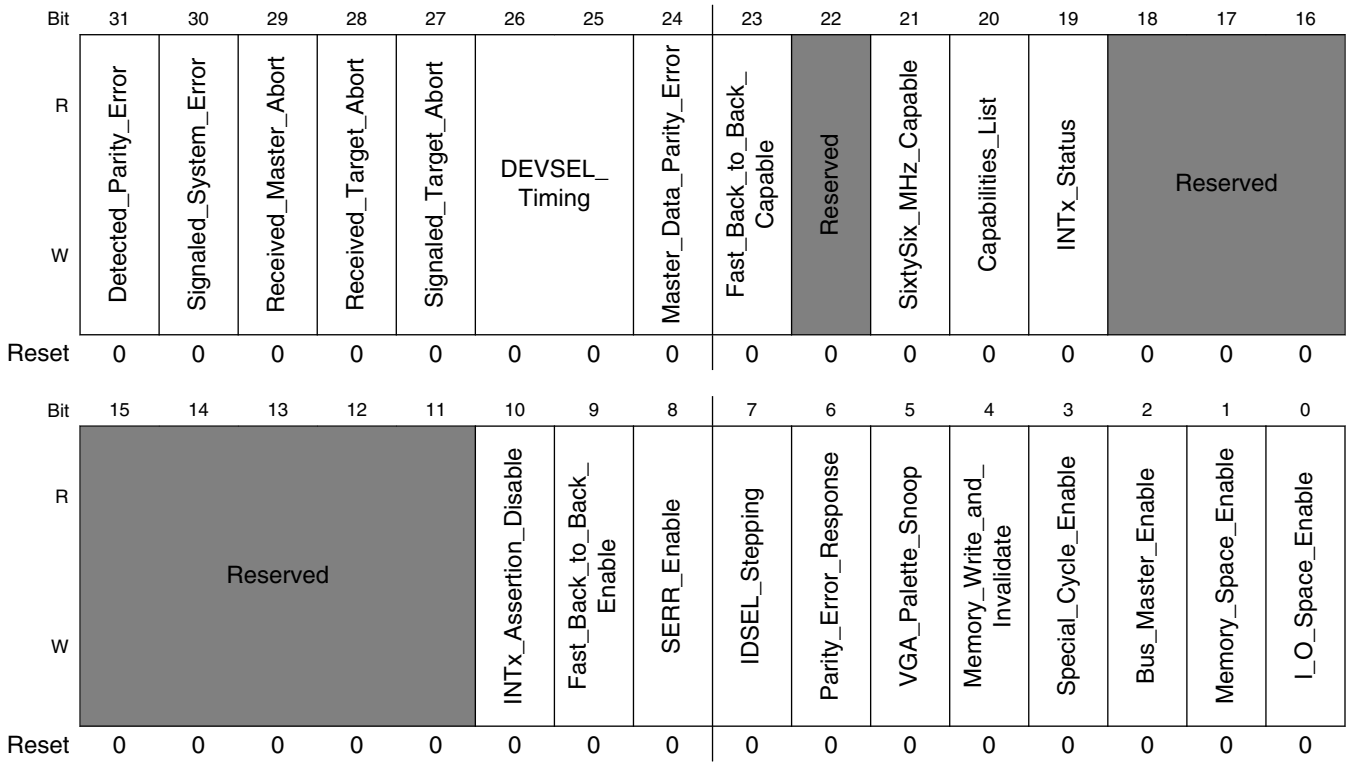
Field	Description
31–16 -	This field is reserved. Reserved
-	This field is reserved. Reserved

49.10.2 Command and Status Register (PCIE_EP_Command)

Offset: 0x04

Bytes: 0-1

Address: 1FF_C000h base + 4h offset = 1FF_C004h



PCIE_EP_Command field descriptions

Field	Description
31 Detected_Parity_Error	Detected Parity Error
30 Signaled_System_Error	Signaled System Error
29 Received_Master_Abort	Received Master Abort
28 Received_Target_Abort	Received Target Abort
27 Signaled_Target_Abort	Signaled Target Abort
26–25 DEVSEL_Timing	DEVSEL Timing Not applicable for PCI Express. Hardwired to 0.
24 Master_Data_Parity_Error	Master Data Parity Error
23 Fast_Back_to_Back_Capable	Fast Back-to-Back Capable Not applicable for PCI Express. Hardwired to 0.

Table continues on the next page...

PCIE_EP_Command field descriptions (continued)

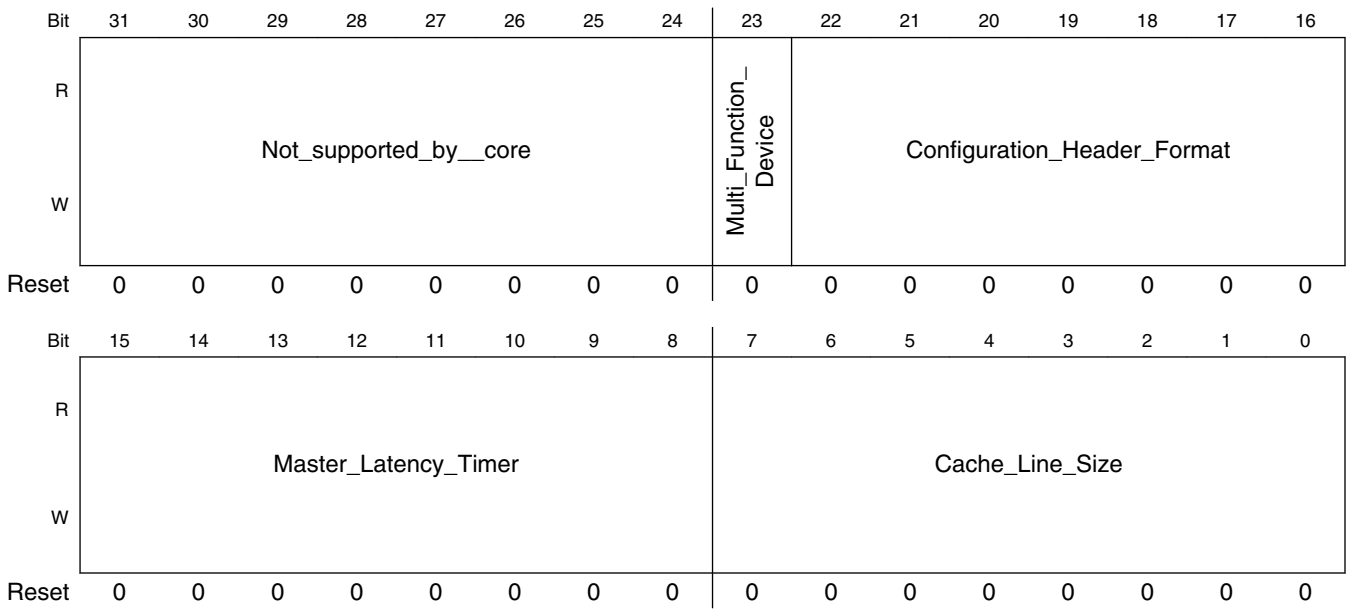
Field	Description
22 -	This field is reserved. Reserved
21 SixtySix_MHz_Capable	66 MHz Capable Not applicable for PCI Express. Hardwired to 0.
20 Capabilities_List	Capabilities List Indicates presence of an extended capability item. Hardwired to 1.
19 INTx_Status	INTx Status
18–16 -	This field is reserved. Reserved
15–11 -	This field is reserved. Reserved
10 INTx_Assertion_Disable	INTx Assertion Disable
9 Fast_Back_to_Back_Enable	Fast Back-to-Back Enable Not applicable for PCI Express. Must be hardwired to 0.
8 SERR_Enable	SERR# Enable
7 IDSEL_Stepping	IDSEL Stepping/Wait Cycle Control Not applicable for PCI Express. Must be hardwired to 0
6 Parity_Error_Response	Parity Error Response
5 VGA_Palette_Snoop	VGA Palette Snoop Not applicable for PCI Express. Must be hardwired to 0.
4 Memory_Write_and_Invalidate	Memory Write and Invalidate Not applicable for PCI Express. Must be hardwired to 0.
3 Special_Cycle_Enable	Special Cycle Enable Not applicable for PCI Express. Must be hardwired to 0.
2 Bus_Master_Enable	Bus Master Enable
1 Memory_Space_Enable	Memory Space Enable
0 I_O_Space_Enable	I/O Space Enable

49.10.3 BIST Register (PCIE_EP_BIST)

Offset: 0x0C

Byte: 0

Address: 1FF_C000h base + Ch offset = 1FF_C00Ch



PCIE_EP_BIST field descriptions

Field	Description
31–24 Not_supported_by__core	The BIST register functions are not supported by the core. All 8 bits of the BIST register are hardwired to 0.
23 Multi_Function_Device	Multi Function Device The default value is 0 for a single function device (`CX_NFUNC = 1) or 1 for a multi-function device (`CX_NFUNC != 1). The Multi Function Device bit is writable through the DBI.
22–16 Configuration_Header_Format	Configuration Header Format Hardwired to 0 for type 0.
15–8 Master_Latency_Timer	Master Latency Timer Not applicable for PCI Express, hardwired to 0.
Cache_Line_Size	Cache Line Size The Cache Line Size register is RW for legacy compatibility purposes and is not applicable to PCI Express device functionality. Writing to the Cache Line Size register does not impact functionality of the core.

49.10.4 Base Address 0 (PCIE_EP_BAR0)

Offset: 0x10-0x24

The core provides three pairs of 32-bit BARs for each implemented function. Each pair (BARs 0 and 1, BARs 2 and 3, BARs 4 and 5) can be configured as follows:

- One 64-bit BAR: For example, BARs 0 and 1 are combined to form a single 64-bit BAR.
- Two 32-bit BARs: For example, BARs 0 and 1 are two independent 32-bit BARs.
- One 32-bit BAR: For example, BAR 0 is a 32-bit BAR and BAR 1 is either disabled or removed from the core altogether to reduce gate count.

In addition, you can configure each BAR to have its incoming Requests routed to either:

- RTRGT1
-

The following sections describe how to set up the BAR types and sizes by programming values into the base address registers. For more information about routing Requests to either RTRGT1 on a BAR-by- BAR basis, see Receive Filtering.

The contents of the six BARs determine the BAR configuration. The reset values of the BARs are determined by hardware configuration options.

At runtime, application software can overwrite the BAR contents to reconfigure the BARs (unless the affected BAR is removed during hardware configuration). Application software must observe the rules listed below when writing to the BARs.

The rules for BAR configuration are the same for all three pairs. Using BARs 0 and 1 as the example pair, the rules for BAR configuration are:

- Any pair (for example, BARs 0 and 1) can be configured as one 64-bit BAR, two 32-bit BARs, or one 32-bit BAR.
- BAR pairs cannot overlap to form a 64-bit BAR. For example, you cannot combine BARs 1 and 2 to form a 64-bit BAR.
-
- An I/O BAR must be a 32-bit BAR and cannot be prefetchable.
- If the device is configured as a PCI Express Endpoint (not a Legacy Endpoint), then any memory that is configured as prefetchable must be a 64-bit memory BAR.
- If BAR 0 is configured as a 64-bit BAR:

- BAR 1 is the upper 32 bits of the combined 64-bit BAR formed by BARs 0 and 1. Therefore, BAR 1 must be disabled and cannot be configured independently.

- BAR 0 must be a memory BAR and can be either prefetchable or non-prefetchable.
- The contents of the BAR 0 Mask register determine the number of writable bits in the 64-bit BAR, subject to the restrictions described in BAR Mask Registers . The BAR 1 Mask register contains the upper 32 bits of the BAR 0 Mask value.
- BAR 0 can be disabled by writing 0 to bit 0 of the BAR 0 Mask register

- If BAR 0 is configured as a 32-bit BAR:

- You can configure BAR 1 as an independent 32-bit BAR
- BAR 0 can be configured as a memory BAR or an I/O BAR.
- The contents of the BAR 0 Mask register determine the number of writable bits in the 32-bit BAR 0, subject to the restrictions described in BAR Mask Registers.
- BAR 0 can be disabled by writing 0 to bit 0 of the BAR 0 Mask register

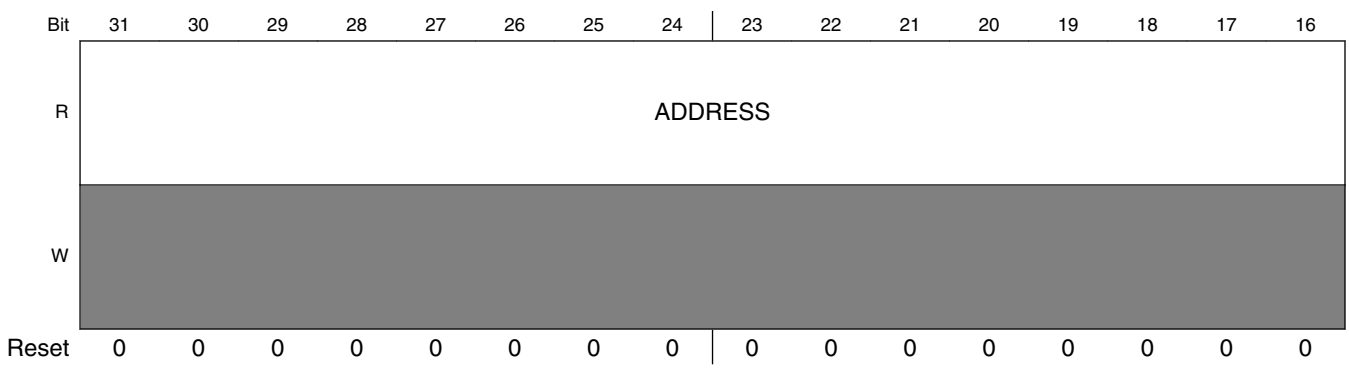
- When BAR 0 is configured as a 32-bit BAR, BAR 1 is available as an independent 32-bit BAR according to the following rules:

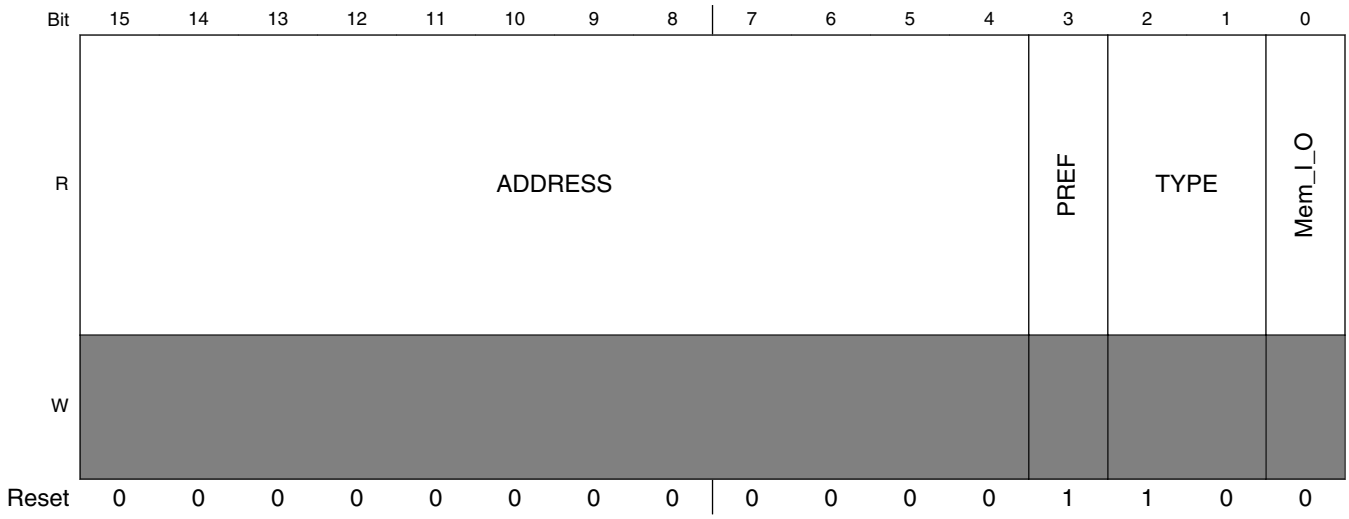
- BAR 1 can be configured as a memory BAR or an I/O BAR.
- The contents of the BAR 1 Mask register determine the number of writable bits in the 32-bit BAR 1, subject to the restrictions described in BAR Mask Registers.
-
-

The same rules apply for pairs 2/3 and 4/5.

Offset: 0x10 (if included in the core hardware configuration)

Address: 1FF_C000h base + 10h offset = 1FF_C010h





PCIE_EP_BAR0 field descriptions

Field	Description
31–4 ADDRESS	BAR 0 base address bits (for a 64-bit BAR, the remaining upper address bits are in BAR 1). The BAR 0 Mask value determines which address bits are masked.
3 PREF	If BAR 0 is an I/O BAR, bit 3 is the second least significant bit of the base address. Bits [3:0] are writable through the DBI. If BAR 0 is a memory BAR, bit 3 indicates if the memory region is prefetchable: 0 = Non-prefetchable 1 = Prefetchable
2–1 TYPE	If BAR 0 is an I/O BAR, bit 2 the least significant bit of the base address and bit 1 is 0. Bits [3:0] are writable through the DBI. If BAR 0 is a memory BAR, bits [2:1] determine the BAR type: 00 = 32-bit BAR 10 = 64-bit BAR
0 Mem_I_O	Bits [3:0] are writable through the DBI. 0 = BAR 0 is a memory BAR 1 = BAR 0 is an I/O BAR

49.10.5 BAR 0 Mask Register (PCIE_EP_MASK0)

The BAR masks are used for indicating the amount of memory each BAR requests from host software. The application logic can overwrite the default values via the DBI.

The BAR Mask registers determine which bits in each BAR are non-writable by host software, which determines the size of the address space claimed by each BAR.

The BAR Mask values indicate the range of low-order bits in each implemented BAR not to use for address matching. The BAR Mask value also indicates the range of low-order bits in the BAR that cannot be written from the host. The application can write to all BAR bits to allow setting of memory, I/O, and other standard BAR options.

To disable any BAR, the application can write a 0 to bit 0 of the corresponding BAR Mask register. To change the BAR Mask value for a disabled BAR, the application must first enable the BAR by writing 1 to bit 0. After enabling the BAR, the application can then write a new value to the BAR Mask register.

The BAR Mask registers are accessible through the same address as the corresponding BAR registers, but requires dbi_cs2 assertions instead i.e. bit address 12 must be set. The BAR Mask registers are writable only, not readable.

If the BAR Mask value for a BAR is less than that required for the BAR type, the core automatically uses the minimum value for the BAR type:

BAR bits [11:0] are always masked for a memory BAR. The core requires each memory BAR to claim at least 4 KB.

The PCI Express Base Specification states that the minimum memory address range requested by a BAR is 128 bytes. In the PCI Local Bus Specification, Rev 3.0 it is recommended that devices that need less than 4 KB of Address Space should still consume 4 KB of address space in order to minimize the number of bits in the address decoder. A Memory BAR size of 256 bytes can be achieved by using a DBI2 write to BAR Mask.

BAR bits [7:0] are always masked for an I/O BAR. The core requires each I/O BAR to claim at least 256 bytes.

The PCI Local Bus Specification, Rev 3.0 allows I/O BARs to consume between 4 bytes and 256 bytes of address space. The core only permits I/O BARs to consume 256 bytes of address space. This restriction is used in order to minimize the number of bits in the address decoder.

The aperture of the BAR is actually the larger of the written sizeor the system page size (set by the operating system). In the case where the system page size is larger than the requested bar size, the BAR is actually sized to the system page size . This means that when the OS writes all 1s then reads back to determine the size of the BAR, the OS will see the BAR size to be the system page size. The application logic, most likely, will only have the original requested amount of physical memory. A transaction will receive a UR if the transaction is from the RC and it targets an address that is within the range of the allocated system page size but above the implemented application memory.

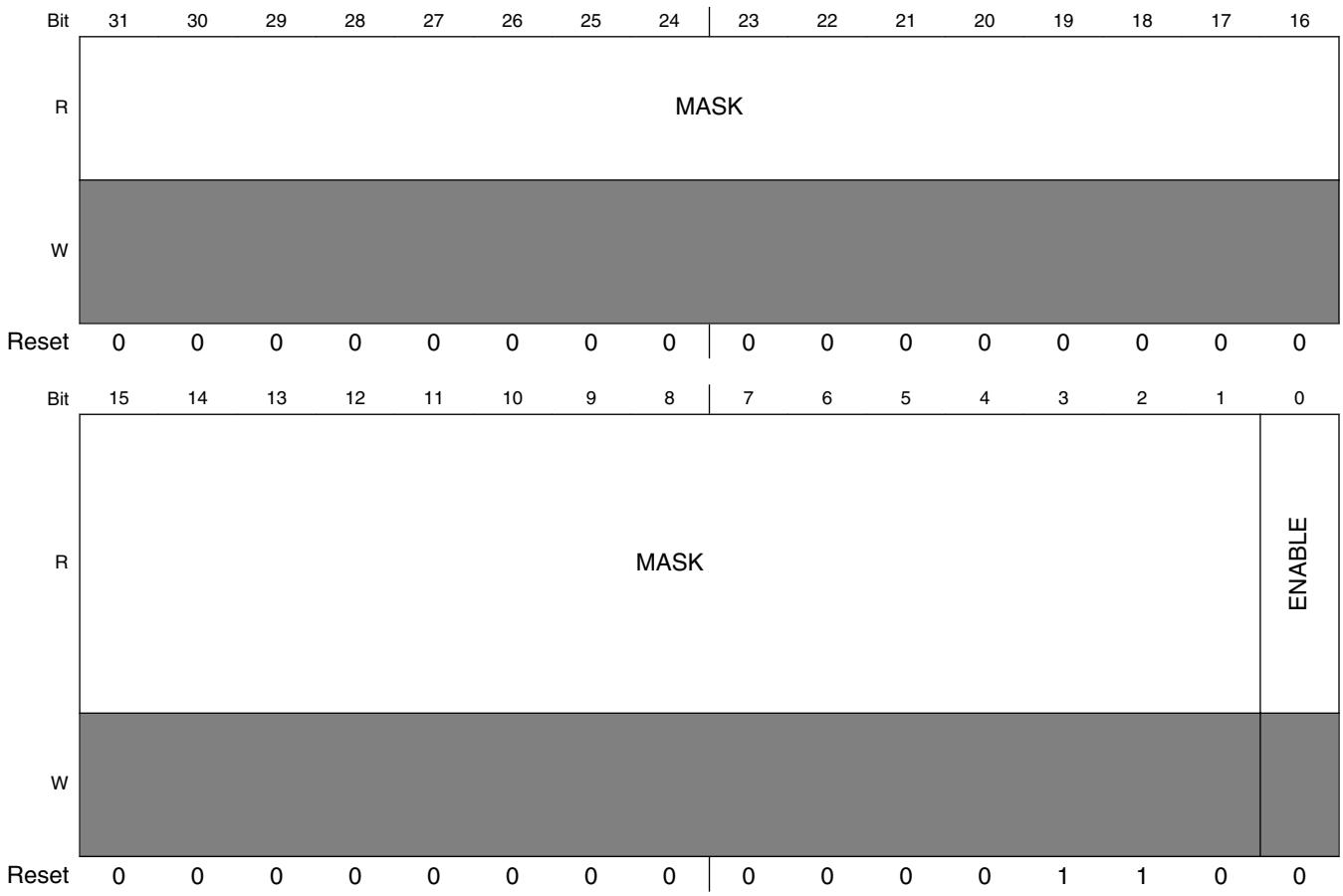
The figure below shows an example configuration of the six BARs and their corresponding BAR Mask registers. The example configuration includes:

- One 64-bit memory BAR (non-prefetchable)
- One 32-bit memory BAR (non-prefetchable)
- One 32-bit I/O BAR

Figure 49-44. Example Base Address Register Configuration

Offset: 0x10 (same as Base Address Register 0, but requires dbi_cs2 for write access)

Address: 1FF_C000h base + 10h offset = 1FF_C010h



PCIE_EP_MASK0 field descriptions

Field	Description
31–1 MASK	Indicates which BAR 0 bits to mask (make non- writable) from host software, which, in turn, determines the size of the BAR. For example, writing 0xFFF to the BAR 0 Mask register claims a 4096- byte BAR by masking bits 11:0 of the BAR from writing by host software. The BAR 1 Mask register contains the upper bits of the BAR 0 Mask. The BAR 0 Mask register is invisible to host software and not readable from the application. • the BAR 0 Mask register is writable through the DBI. • •
0 ENABLE	Bit 0 is interpreted as BAR Enable when writing to the BAR Mask register rather than as a mask bit because bit 0 of a BAR is always masked from writing by host software.

Table continues on the next page...

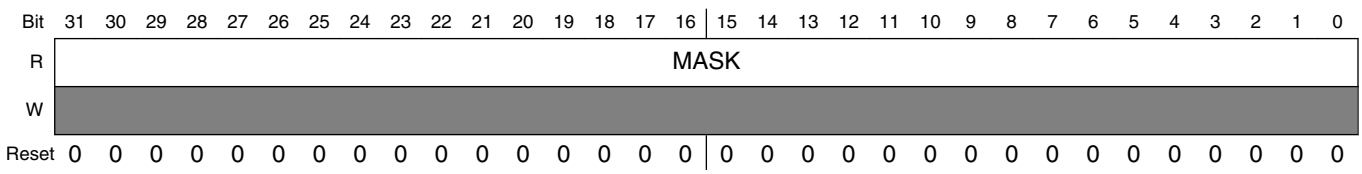
PCIE_EP_MASK0 field descriptions (continued)

Field	Description
	BAR 0 Enable 0 BAR 0 is disabled 1 BAR 0 is enabled

49.10.6 BAR 1 Mask Register (PCIE_EP_MASK1)

Offset: 0x14 (same as Base Address Register 1, but requires dbi_cs2 for write access)

Address: 1FF_C000h base + 14h offset = 1FF_C014h



PCIE_EP_MASK1 field descriptions

Field	Description
MASK	• Bits [31:1]: BAR 1 Mask value, interpreted the same way as BAR 0 Mask. Default value is `BAR1_MASK_N. • Bit 0: BAR 1 Enable (0 = BAR 1 is disabled; 1= BAR 1 is enabled). Default value is `BAR1_ENABLED_N. • `BAR1_MASK_WRITABLE_N controls application write access to the BAR 1 Mask register.  • Bits [31:0] are the upper bits of the BAR 0 Mask value. • `BAR0_MASK_WRITABLE_N controls application write access to the full 64-bit BAR 0 Mask register.

49.10.7 BAR 2 Mask Register (PCIE_EP_MASK2)

Offset: 0x18 (same as Base Address Register 2, but requires dbi_cs2 for write access)

Address: 1FF_C000h base + 18h offset = 1FF_C018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BAR2_MASK_N															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BAR2_MASK_N															BAR2_ENABLED_N
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

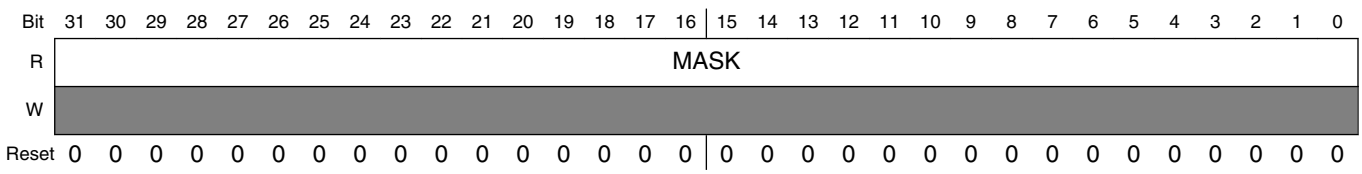
PCIE_EP_MASK2 field descriptions

Field	Description
31–1 BAR2_MASK_N	Indicates which BAR 2 bits to mask (make non- writable) from host software, which, in turn, determines the size of the BAR. For example, writing 0xFFF to the BAR 2 Mask register claims a 4096-byte BAR by masking bits 11:0 of the BAR from writing by host software. The BAR 2 Mask register is invisible to host software and not readable from the application. • the BAR 2 Mask register is writable through the DBI. •
0 BAR2_ ENABLED_N	Bit 0 is interpreted as BAR Enable when writing to the BAR Mask register rather than as a mask bit because bit 0 of a BAR is always masked from writing by host software. BAR 2 Enable 0 BAR 2 is disabled 1 BAR 2 is enabled

49.10.8 BAR 3 Mask Register (PCIE_EP_MASK3)

Offset: 0x1C (same as Base Address Register 3, but requires dbi_cs2 for write access)

Address: 1FF_C000h base + 1Ch offset = 1FF_C01Ch



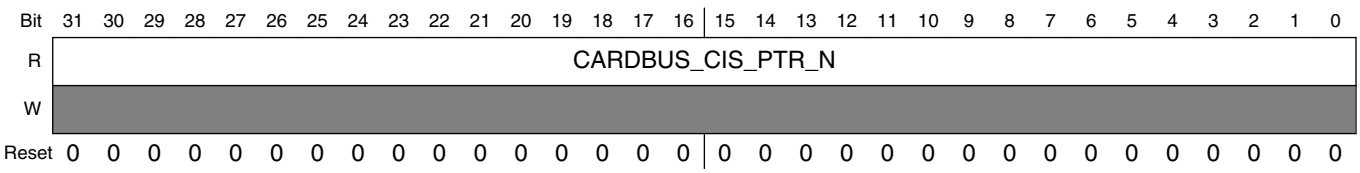
PCIE_EP_MASK3 field descriptions

Field	Description
MASK	• Bits [31:1]: BAR 3 Mask value, interpreted the same way as BAR 2 Mask. Default value is `BAR3_MASK_N. • Bit 0: BAR 3 Enable (0 = BAR 3 is disabled; 1= BAR 3 is enabled). Default value is `BAR3_ENABLED_N. • `BAR3_MASK_WRITABLE_N controls application can not write access to the BAR 3 Mask register. • Bits [31:0] are the upper bits of the BAR 2 Mask value. • `BAR2_MASK_WRITABLE_N controls application write access to the full 64-bit BAR 2 Mask register.

49.10.9 CardBus CIS Pointer Register (PCIE_EP_CISP)

Offset: 0x28

Address: 1FF_C000h base + 28h offset = 1FF_C028h



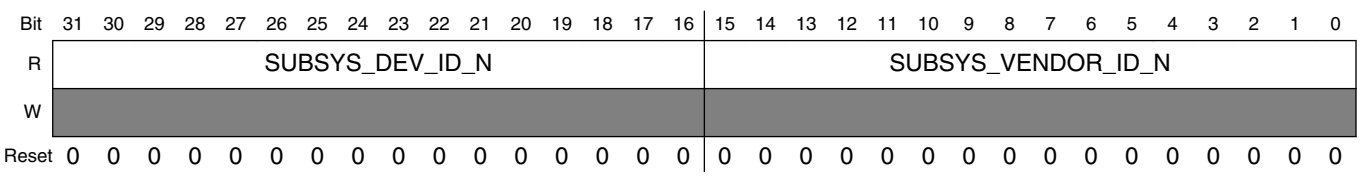
PCIE_EP_CISP field descriptions

Field	Description
CARDBUS_CIS_PTR_N	CardBus CIS Pointer Optional, writable through the DBI.

49.10.10 Subsystem ID and Subsystem Vendor ID Register (PCIE_EP_SSID)

Offset: 0x2C

Address: 1FF_C000h base + 2Ch offset = 1FF_C02Ch



PCIE_EP_SSID field descriptions

Field	Description
31–16 SUBSYS_DEV_ID_N	Subsystem ID Writable through the DBI.
SUBSYS_VENDOR_ID_N	Subsystem Vendor ID Writable through the DBI.

49.10.11 Expansion ROM Base Address Register (PCIE_EP_EROMBAR)

Offset: 0x30

Address: 1FF_C000h base + 30h offset = 1FF_C030h



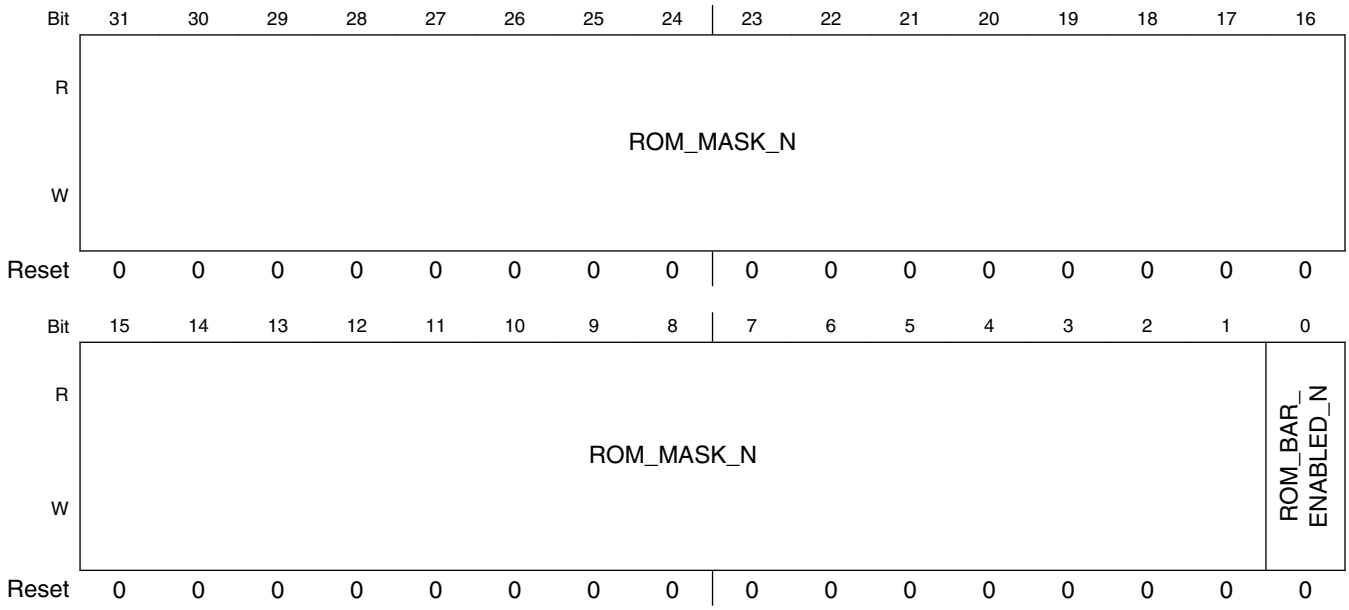
PCIE_EP_EROMBAR field descriptions

Field	Description
31–11 ADDRESS	Expansion ROM Address
10–1 -	This field is reserved. Reserved
0 ENABLE	Expansion ROM Enable

49.10.12 Expansion ROM BAR Mask Register (PCIE_EP_EROMMASK)

Offset: 0x30 (same as the Expansion ROM BAR, but requires dbi_cs2 for write access)

Address: 1FF_C000h base + 30h offset = 1FF_C030h



PCIE_EP_EROMMASK field descriptions

Field	Description
31–1 ROM_MASK_N	Indicates which Expansion ROM BAR bits to mask (make non-writable) from host software, which, in turn, determines the size of the BAR. For example, writing 0xFFF to the Expansion ROM BAR Mask register claims a 4096-byte BAR by masking bits 11:0 of the BAR from writing by host software. The maximum value is 0xFFFFFF because the maximum space that can be claimed by an Expansion ROM BAR is 16 MB. The Expansion ROM BAR Mask register is invisible to host software and not readable from the application. Application access depends on the value of • the Expansion ROM BAR Mask register is writable through the DBI. •
0 ROM_BAR_ ENABLED_N	Expansion ROM BAR Enable 0 Expansion ROM BAR is disabled 1 Expansion ROM BAR is enabled

49.10.13 Capability Pointer Register (PCIE_EP_CAPPR)

Offset: 0x34

Byte: 0

Address: 1FF_C000h base + 34h offset = 1FF_C034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																								CFG_NEXT_PTR							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

PCIE_EP_CAPPR field descriptions

Field	Description
31–8 -	This field is reserved. Reserved
CFG_NEXT_PTR	First Capability Pointer. See PF PCI Standard Capability Structures Register Maps for more information.

49.10.14 Interrupt Line and Pin Register (PCIE_EP_ILR)

Offset: 0x3C

Byte: 0

Address: 1FF_C000h base + 3Ch offset = 1FF_C03Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																INT_PIN_MAPPING_N								INTERRUPT_LINE							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1

PCIE_EP_ILR field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15–8 INT_PIN_ MAPPING_N	Interrupt Pin Identifies the legacy interrupt Message that the device (or device function) uses. In a single-function configuration, the core only uses INTA. The Interrupt Pin register is writable through the DBI. Valid values are: 0x00 The device (or function) does not use legacy interrupt

Table continues on the next page...

PCIE_EP_ILR field descriptions (continued)

Field	Description
	0x01 The device (or function) uses INTA 0x02 The device (or function) uses INTB 0x03 The device (or function) uses INTC 0x04 The device (or function) uses INTD
INTERRUPT_ LINE	Interrupt Line Value in this register is system architecture specific. POST software will write the routing information into this register as it initializes and configures the system.

49.10.15 AER Capability Header (PCIE_EP_AER)

The core implements the following PCI Express Extended Capabilities registers:

? Advanced Error Reporting Capability register set

? Virtual Channel Capability register set -

Address: 0x100

Address: 1FF_C000h base + 100h offset = 1FF_C100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Next_Capability_Offset												Capability_ Version				PCI_Express_Extended_Capability_ID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

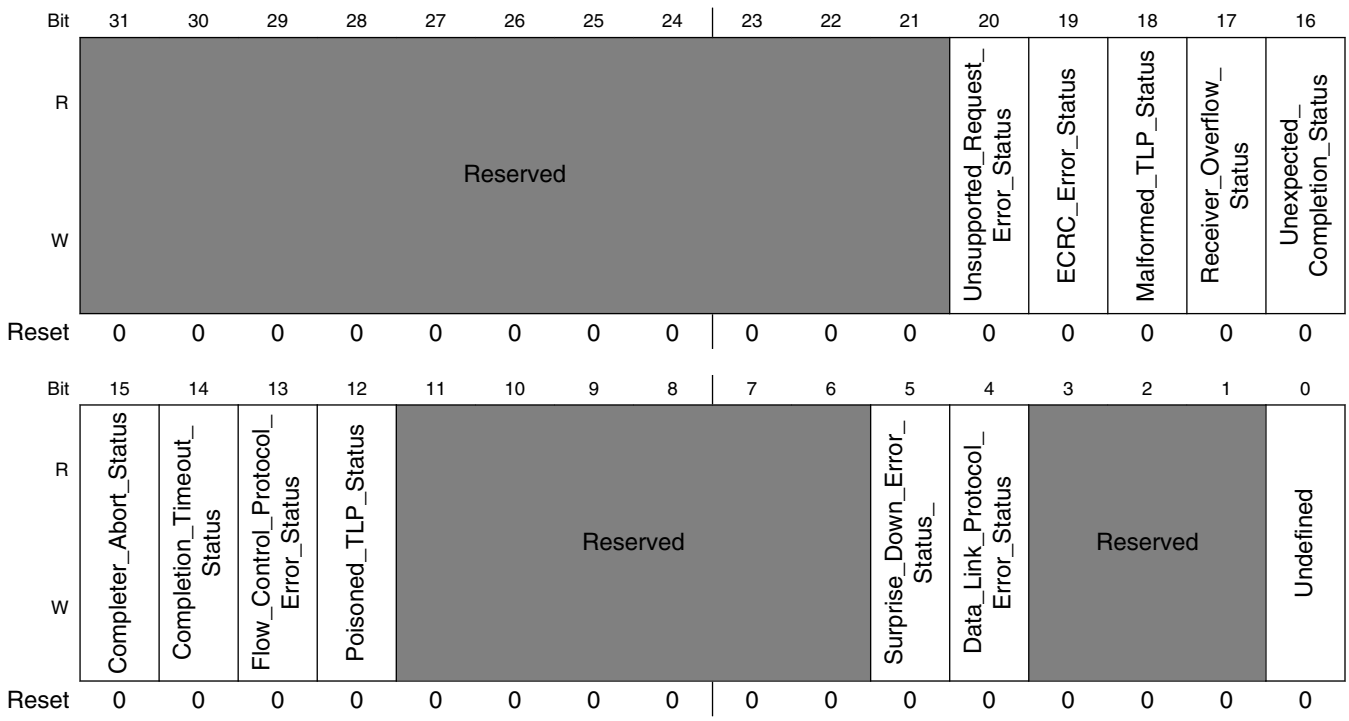
PCIE_EP_AER field descriptions

Field	Description
31–20 Next_Capability_ Offset	Next Capability Offset
19–16 Capability_ Version	Capability Version
PCI_Express_ Extended_ Capability_ID	PCI Express Extended Capability ID Value is 0x1 for Advanced Error Reporting.

49.10.16 Uncorrectable Error Status Register (PCIE_EP_UESR)

Offset: 0x04

Address: 1FF_C000h base + 104h offset = 1FF_C104h



PCIE_EP_UESR field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 Unsupported_Request_Error_Status	Unsupported Request Error Status
19 ECRC_Error_Status	ECRC Error Status
18 Malformed_TLP_Status	Malformed TLP Status
17 Receiver_Overflow_Status	Receiver Overflow Status

Table continues on the next page...

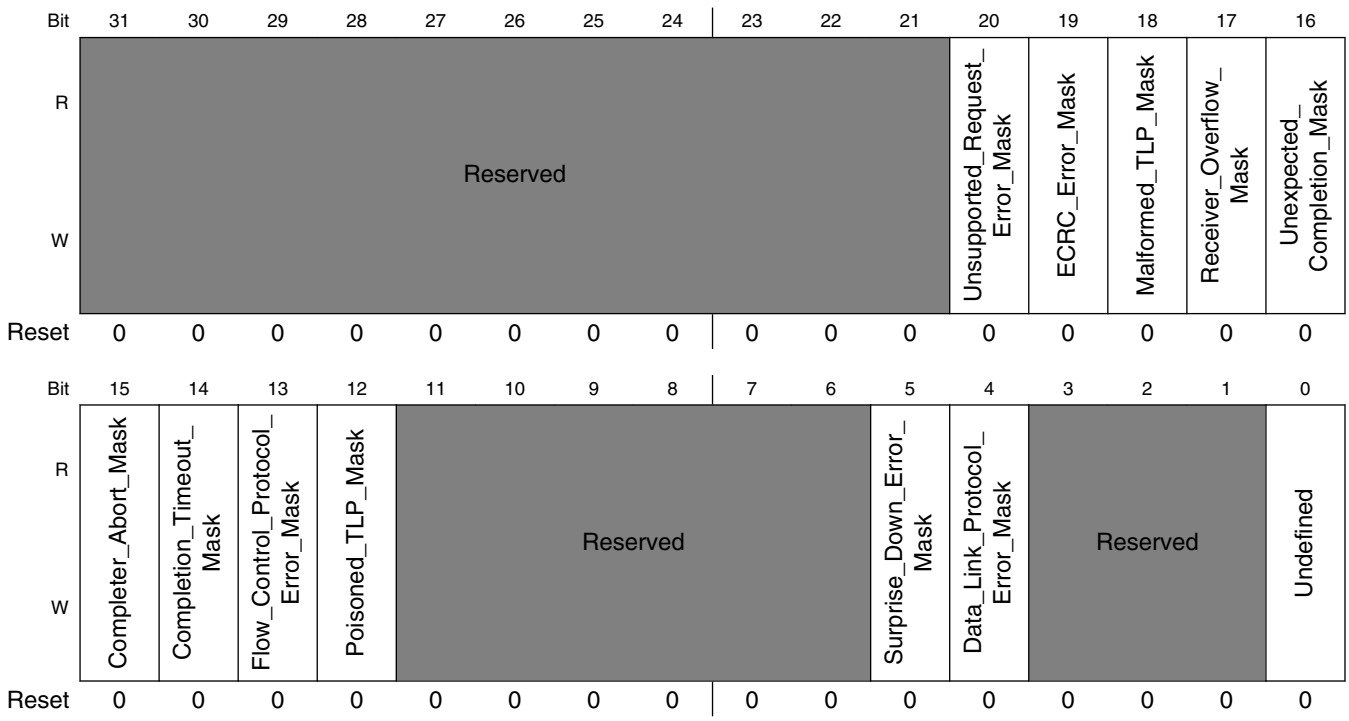
PCIE_EP_UESR field descriptions (continued)

Field	Description
16 Unexpected_ Completion_ Status	Unexpected Completion Status
15 Completer_ Abort_Status	Completer Abort Status
14 Completion_ Timeout_Status	Completion Timeout Status
13 Flow_Control_ Protocol_Error_ Status	Flow Control Protocol Error Status
12 Poisoned_TLP_ Status	Poisoned TLP Status
11–6 -	This field is reserved. Reserved
5 Surprise_Down_ Error_Status_	Surprise Down Error Status (not supported)
4 Data_Link_ Protocol_Error_ Status	Data Link Protocol Error Status
3–1 -	This field is reserved. Reserved
0 Undefined	Undefined for PCI Express 1.1 (Was Training Error Status for PCI Express 1.0a)

49.10.17 Uncorrectable Error Mask Register (PCIE_EP_UEMR)

Offset: 0x08

Address: 1FF_C000h base + 108h offset = 1FF_C108h



PCIE_EP_UEMR field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 Unsupported_Request_Error_Mask	Unsupported Request Error Mask
19 ECRC_Error_Mask	ECRC Error Mask
18 Malformed_TLP_Mask	Malformed TLP Mask
17 Receiver_Overflow_Mask	Receiver Overflow Mask

Table continues on the next page...

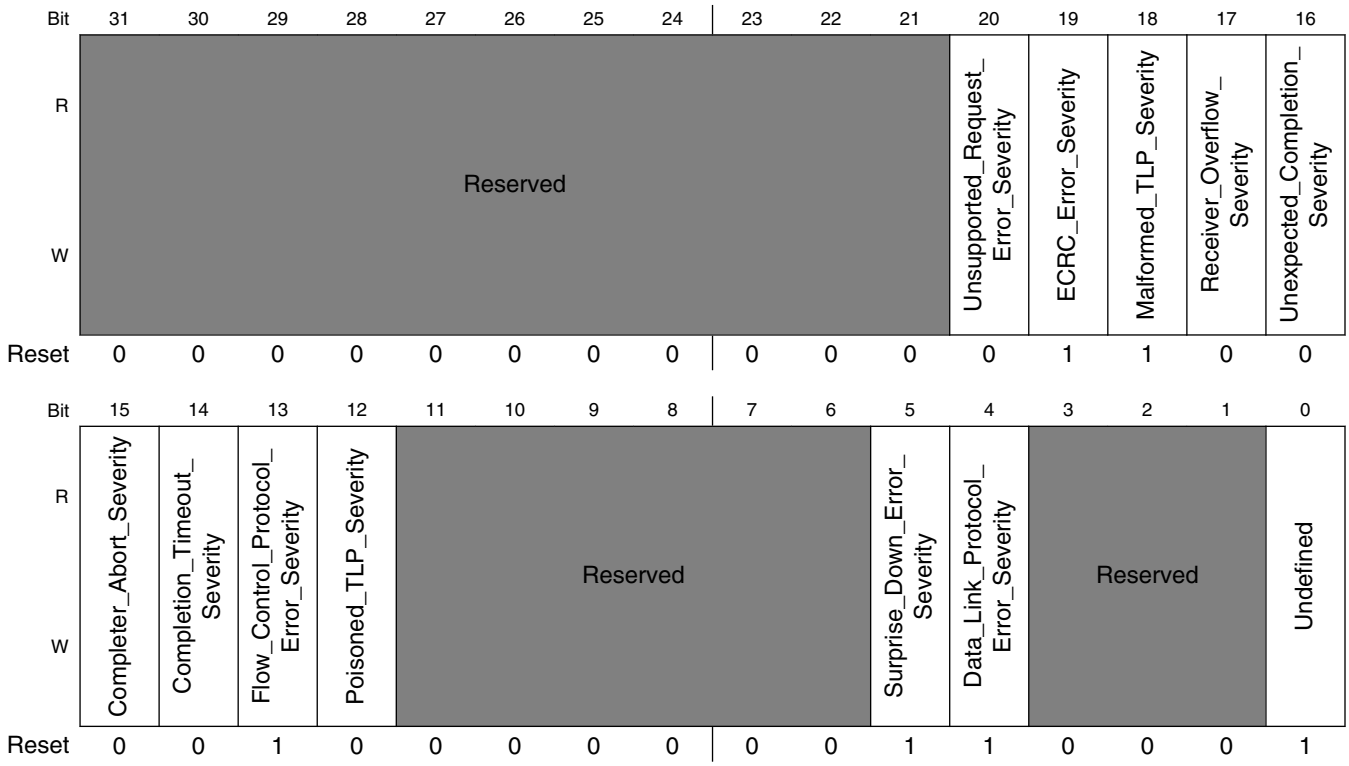
PCIE_EP_UEMR field descriptions (continued)

Field	Description
16 Unexpected_ Completion_ Mask	Unexpected Completion Mask
15 Completer_ Abort_Mask	Completer Abort Mask
14 Completion_ Timeout_Mask	Completion Timeout Mask
13 Flow_Control_ Protocol_Error_ Mask	Flow Control Protocol Error Mask
12 Poisoned_TLP_ Mask	Poisoned TLP Mask
11–6 -	This field is reserved. Reserved
5 Surprise_Down_ Error_Mask	Surprise Down Error Mask (not supported)
4 Data_Link_ Protocol_Error_ Mask	Data Link Protocol Error Mask
3–1 -	This field is reserved. Reserved
0 Undefined	Undefined for PCI Express 1.1 (Was Training Error Mask for PCI Express 1.0a)

49.10.18 Uncorrectable Error Severity Register (PCIE_EP_UESevR)

Offset: 0x0C

Address: 1FF_C000h base + 10Ch offset = 1FF_C10Ch



PCIE_EP_UESevR field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 Unsupported_Request_Error_Severity	Unsupported Request Error Severity
19 ECRC_Error_Severity	ECRC Error Severity
18 Malformed_TLP_Severity	Malformed TLP Severity

Table continues on the next page...

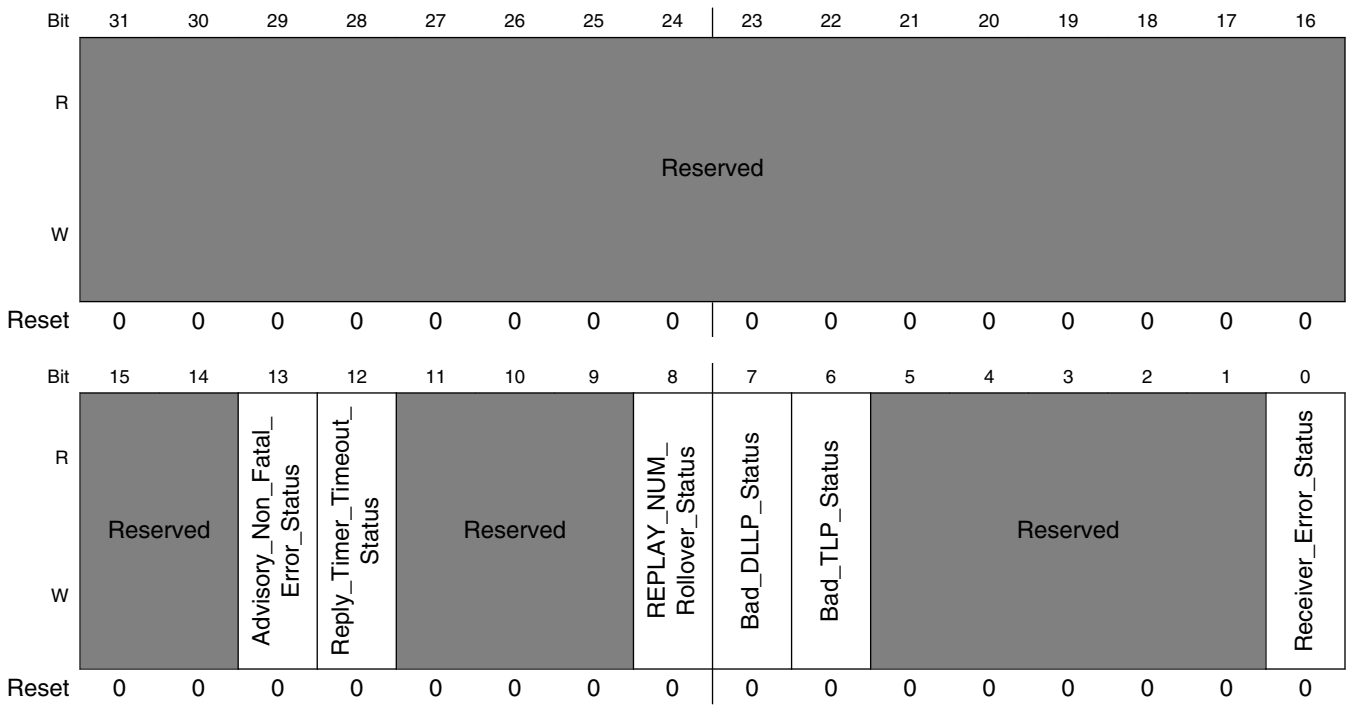
PCIE_EP_UESevR field descriptions (continued)

Field	Description
17 Receiver_ Overflow_ Severity	Receiver Overflow Severity
16 Unexpected_ Completion_ Severity	Unexpected Completion Severity
15 Completer_ Abort_Severity	Completer Abort Severity
14 Completion_ Timeout_Severity	Completion Timeout Severity
13 Flow_Control_ Protocol_Error_ Severity	Flow Control Protocol Error Severity
12 Poisoned_TLP_ Severity	Poisoned TLP Severity
11–6 -	This field is reserved. Reserved
5 Surprise_Down_ Error_Severity	Surprise Down Error Severity (not supported)
4 Data_Link_ Protocol_Error_ Severity	Data Link Protocol Error Severity
3–1 -	This field is reserved. Reserved
0 Undefined	Undefined for PCI Express 1.1 (Was Training Error Severity for PCI Express 1.0a)

49.10.19 Correctable Error Status Register (PCIE_EP_CESR)

Offset: 0x10

Address: 1FF_C000h base + 110h offset = 1FF_C110h



PCIE_EP_CESR field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13 Advisory_Non_Fatal_Error_Status	Advisory Non-Fatal Error Status
12 Reply_Timer_Timeout_Status	Reply Timer Timeout Status
11–9 -	This field is reserved. Reserved
8 REPLAY_NUM_Rollover_Status	REPLAY_NUM Rollover Status
7 Bad_DLLP_Status	Bad DLLP Status

Table continues on the next page...

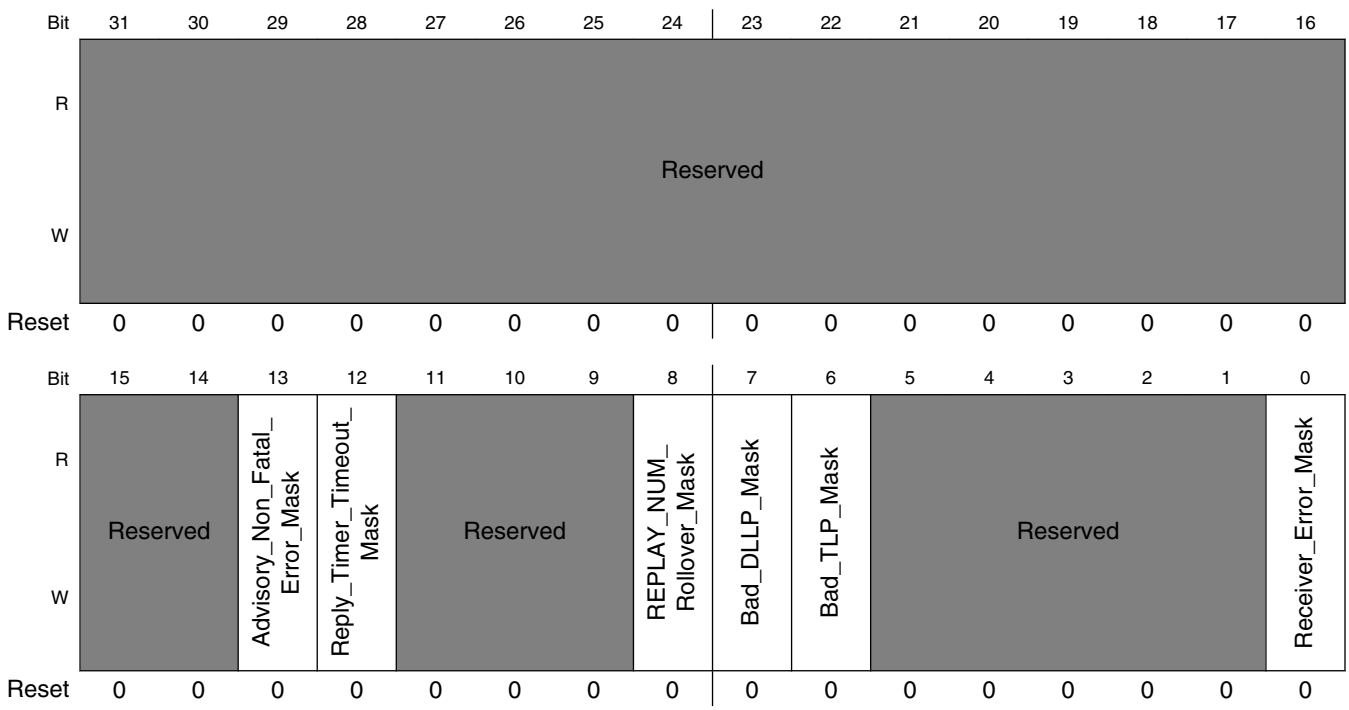
PCIE_EP_CESR field descriptions (continued)

Field	Description
6 Bad_TLP_Status	Bad TLP Status
5–1 -	This field is reserved. Reserved
0 Receiver_Error_Status	Receiver Error Status

49.10.20 Correctable Error Mask Register (PCIE_EP_CEMR)

Offset: 0x14

Address: 1FF_C000h base + 114h offset = 1FF_C114h



PCIE_EP_CEMR field descriptions

Field	Description
31–14 -	This field is reserved. Reserved

Table continues on the next page...

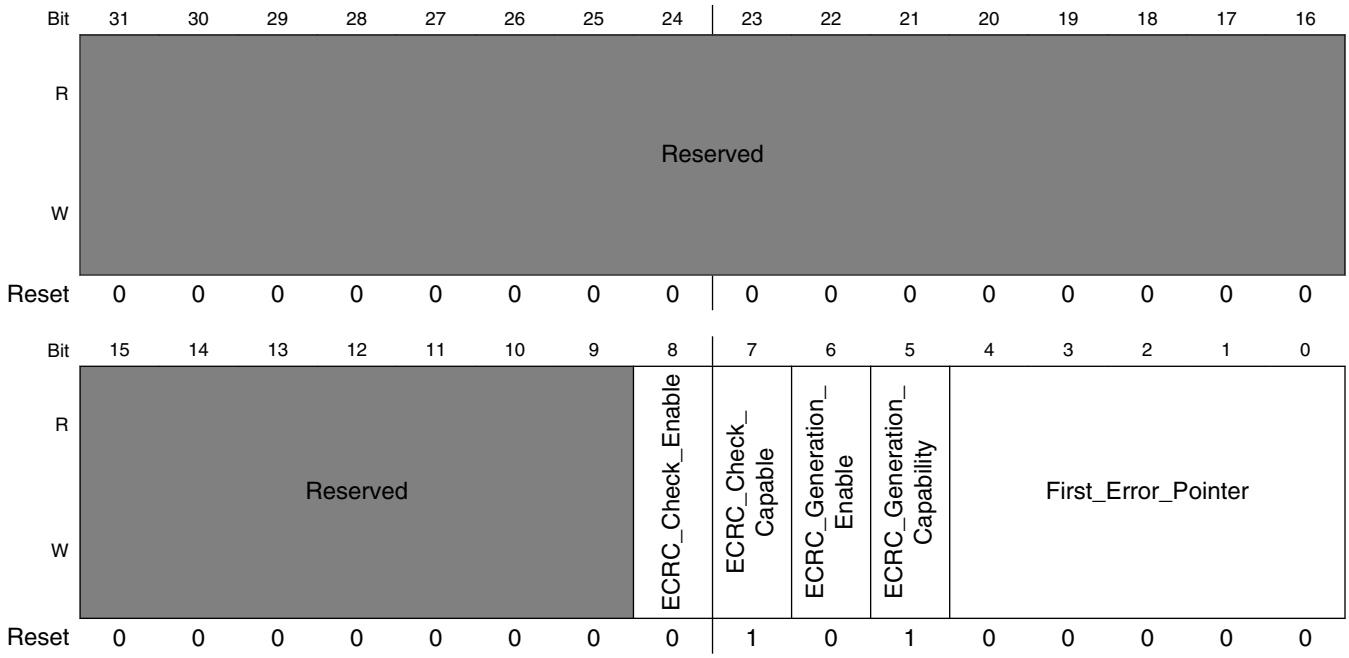
PCIE_EP_CEMR field descriptions (continued)

Field	Description
13 Advisory_Non_Fatal_Error_Mask	Advisory Non-Fatal Error Mask
12 Reply_Timer_Timeout_Mask	Reply Timer Timeout Mask
11–9 -	This field is reserved. Reserved
8 REPLAY_NUM_Rollover_Mask	REPLAY_NUM Rollover Mask
7 Bad_DLLP_Mask	Bad DLLP Mask
6 Bad_TLP_Mask	Bad TLP Mask
5–1 -	This field is reserved. Reserved
0 Receiver_Error_Mask	Receiver Error Mask

49.10.21 Advanced Capabilities and Control Register (PCIE_EP_ACCR)

Offset: 0x18

Address: 1FF_C000h base + 118h offset = 1FF_C118h



PCIE_EP_ACCR field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 ECRC_Check_ Enable	ECRC Check Enable
7 ECRC_Check_ Capable	ECRC Check Capable
6 ECRC_ Generation_ Enable	ECRC Generation Enable
5 ECRC_ Generation_ Capability	ECRC Generation Capability
First_Error_ Pointer	First Error Pointer

49.10.22 Header Log Register (PCIE_EP_HLR)

Offset: 0x1C

The Header Log registers collect the header for the TLP corresponding to a detected error. See the PCI Express 3.0 Specification for details. Each of the Header Log registers is type ROS; the default reset value of each Header Log register is 0x00000000.

Address: 1FF_C000h base + 11Ch offset = 1FF_C11Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_EP_HLR field descriptions

Field	Description
-	Header Log Register (nth DWORD)

49.10.23 VC Extended Capability Header (PCIE_EP_VCECHR)

Offset: 0x140

Address: 1FF_C000h base + 140h offset = 1FF_C140h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Next_Capability_Offset												Capability_ Version				Extended_Capability															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

PCIE_EP_VCECHR field descriptions

Field	Description
31–20 Next_Capability_ Offset	Next Capability Offset
19–16 Capability_ Version	Capability Version
Extended_ Capability	PCI Express Extended Capability The default value is 0x2 for VC Capability.

49.10.24 Port VC Capability Register 1 (PCIE_EP_PVCCR1)

Offset: 0x140 + 0x4

Address: 1FF_C000h base + 144h offset = 1FF_C144h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				Port_ Arbitration_ Table_Entry_ Size		Reference_ Clock		Reserved	Low_Priority_ Extended_VC_ Count			Reserved	Extended_VC_ Count		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

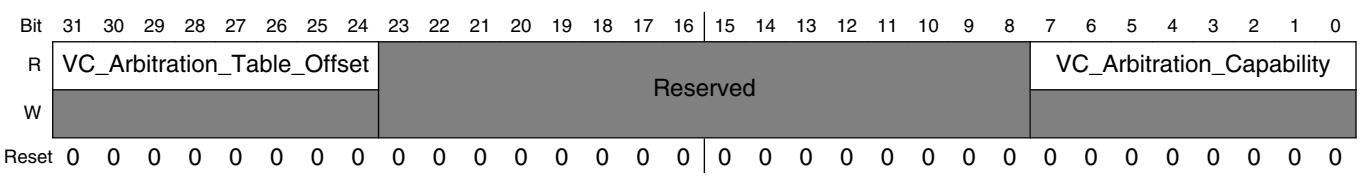
PCIE_EP_PVCCR1 field descriptions

Field	Description
31–12 -	This field is reserved. Reserved
11–10 Port_Arbitration_Table_Entry_Size	Port Arbitration Table Entry Size
9–8 Reference_Clock	Reference Clock
7 -	This field is reserved. Reserved
6–4 Low_Priority_Extended_VC_Count	Low Priority Extended VC Count, writable through the DBI
3 -	This field is reserved. Reserved
Extended_VC_Count	Extended VC Count The default value is the one less than the number of VCs that

49.10.25 Port VC Capability Register 2 (PCIE_EP_PVCCR2)

Offset: 0x140 + 0x8

Address: 1FF_C000h base + 148h offset = 1FF_C148h



PCIE_EP_PVCCR2 field descriptions

Field	Description
31–24 VC_Arbitration_Table_Offset	VC Arbitration Table Offset (not supported) The default value is 0x00 (no arbitration table present).
23–8 -	This field is reserved. Reserved
VC_Arbitration_Capability	VC Arbitration Capability Indicates which VC arbitration mode(s) the device supports, writable through the DBI: •Bit 0: Device supports hardware fixed arbitration scheme. For the core, the scheme is 16-phase weighted round robin (WRR). •Bit 1: Device supports 32-phase WRR

Table continues on the next page...

PCIE_EP_PVCCR2 field descriptions (continued)

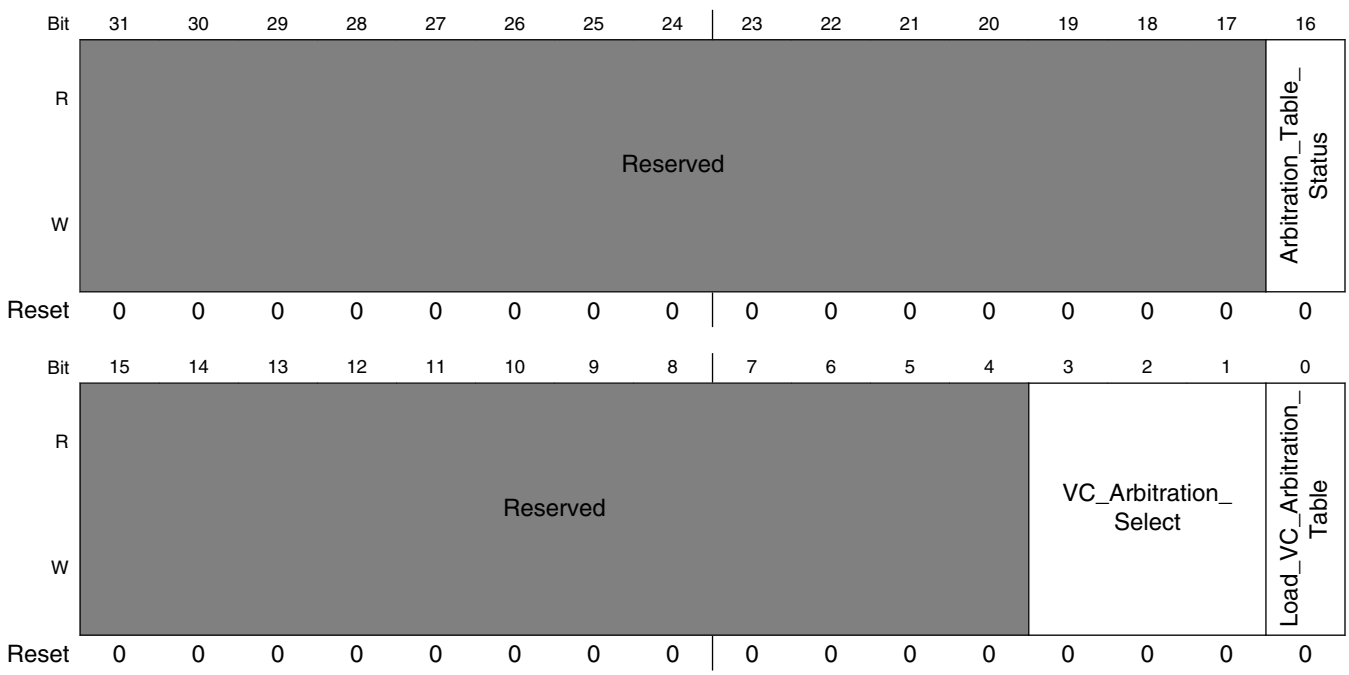
Field	Description
	•Bit 2: Device supports 64-phase WRR •Bit 3: Device supports 128-phase WRR •Bits 4-7: Reserved

49.10.26 Port VC Control and Status Register (PCIE_EP_PVCCSR)

Offset: 0x140 + 0xC

Bytes: 0-1

Address: 1FF_C000h base + 14Ch offset = 1FF_C14Ch



PCIE_EP_PVCCSR field descriptions

Field	Description
31–17 -	This field is reserved. Reserved
16 Arbitration_Table_Status	Arbitration Table Status
15–4 -	This field is reserved. Reserved

Table continues on the next page...

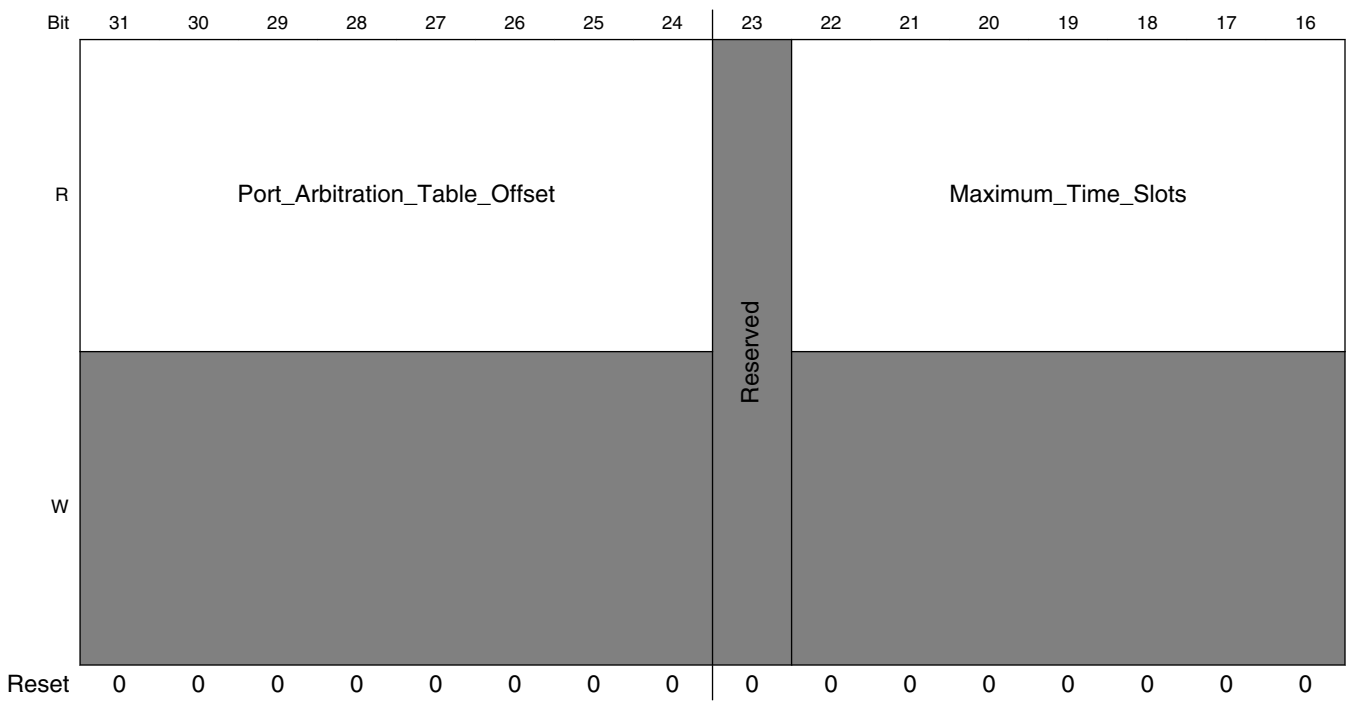
PCIE_EP_PVCCSR field descriptions (continued)

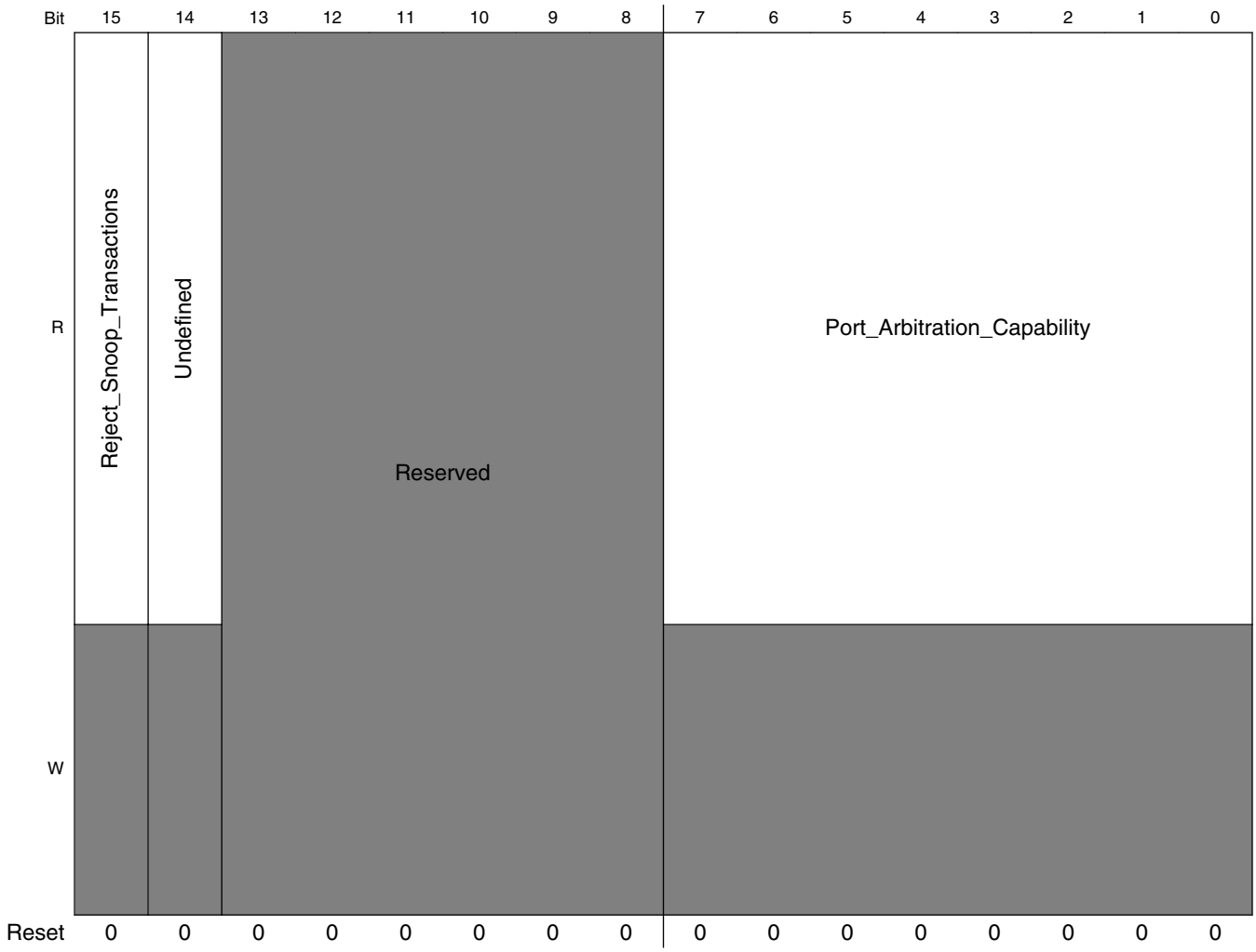
Field	Description
3–1 VC_Arbitration_ Select	VC Arbitration Select
0 Load_VC_ Arbitration_Table	Load VC Arbitration Table

49.10.27 VC Resource Capability Register n (PCIE_EP_VCRCR)

Offset: 0x140 + 0x10

Address: 1FF_C000h base + 150h offset = 1FF_C150h





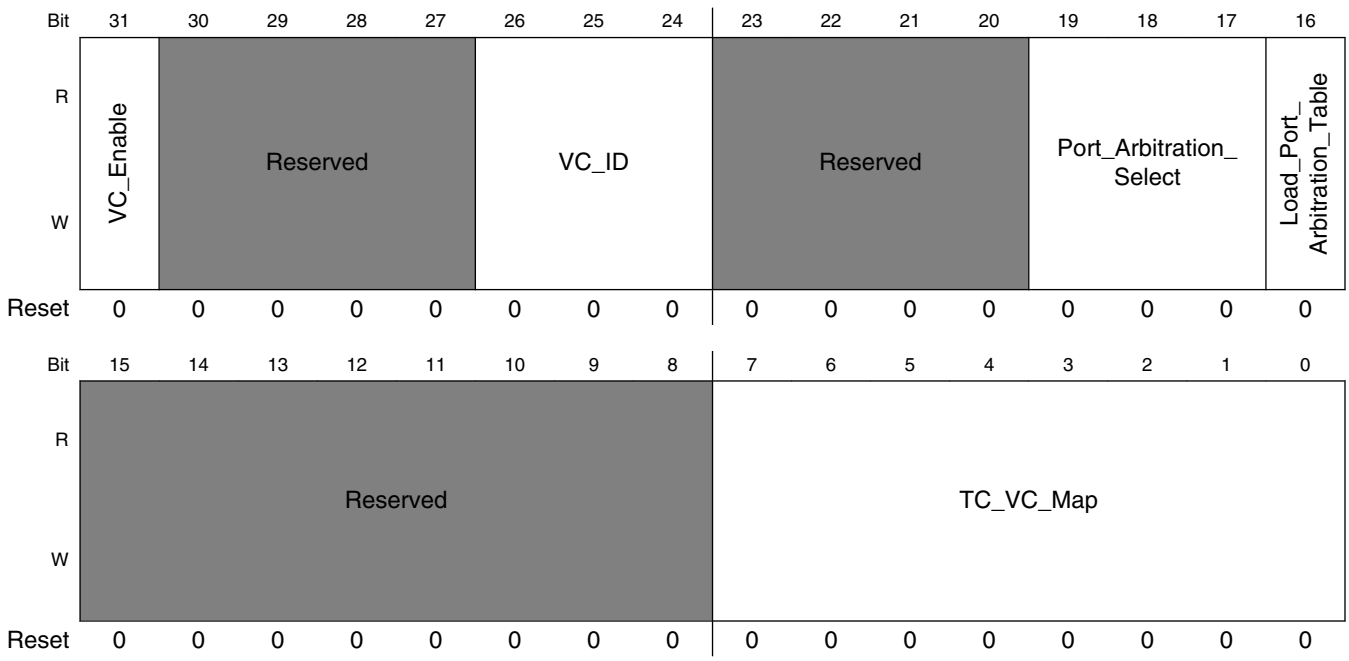
PCIE_EP_VCRCR field descriptions

Field	Description
31–24 Port_Arbitration_Table_Offset	Port Arbitration Table Offset
23 -	This field is reserved. Reserved
22–16 Maximum_Time_Slots	Maximum Time Slots
15 Reject_Snoop_Transactions	Reject Snoop Transactions
14 Undefined	Undefined for PCI Express 1.1 (Was Advanced Packet Switching for PCI Express 1.0a)
13–8 -	This field is reserved. Reserved
Port_Arbitration_Capability	Port Arbitration Capability

49.10.28 VC Resource Control Register n (PCIE_EP_VCRConR)

Offset: 0x140 + 0x14

Address: 1FF_C000h base + 154h offset = 1FF_C154h



PCIE_EP_VCRConR field descriptions

Field	Description
31 VC_Enable	VC Enable Hardwired to 1 for the first VC.
30–27 -	This field is reserved. Reserved
26–24 VC_ID	VC ID Hardwired to 0 for VC0.
23–20 -	This field is reserved. Reserved
19–17 Port_Arbitration_ Select	Port Arbitration Select
16 Load_Port_ Arbitration_Table	Load Port Arbitration Table
15–8 -	This field is reserved. Reserved
TC_VC_Map	TC/VC Map

Table continues on the next page...

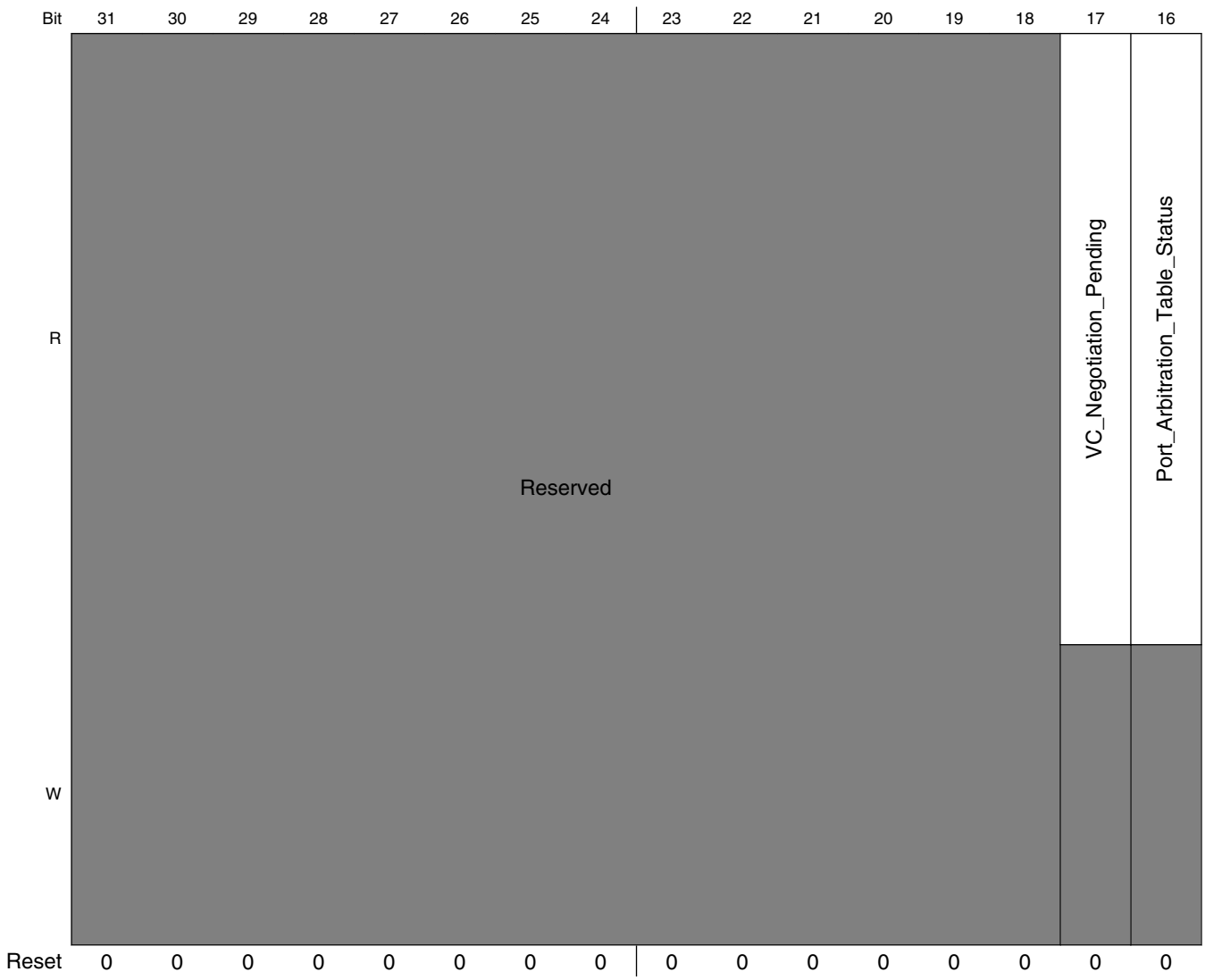
PCIE_EP_VCRConR field descriptions (continued)

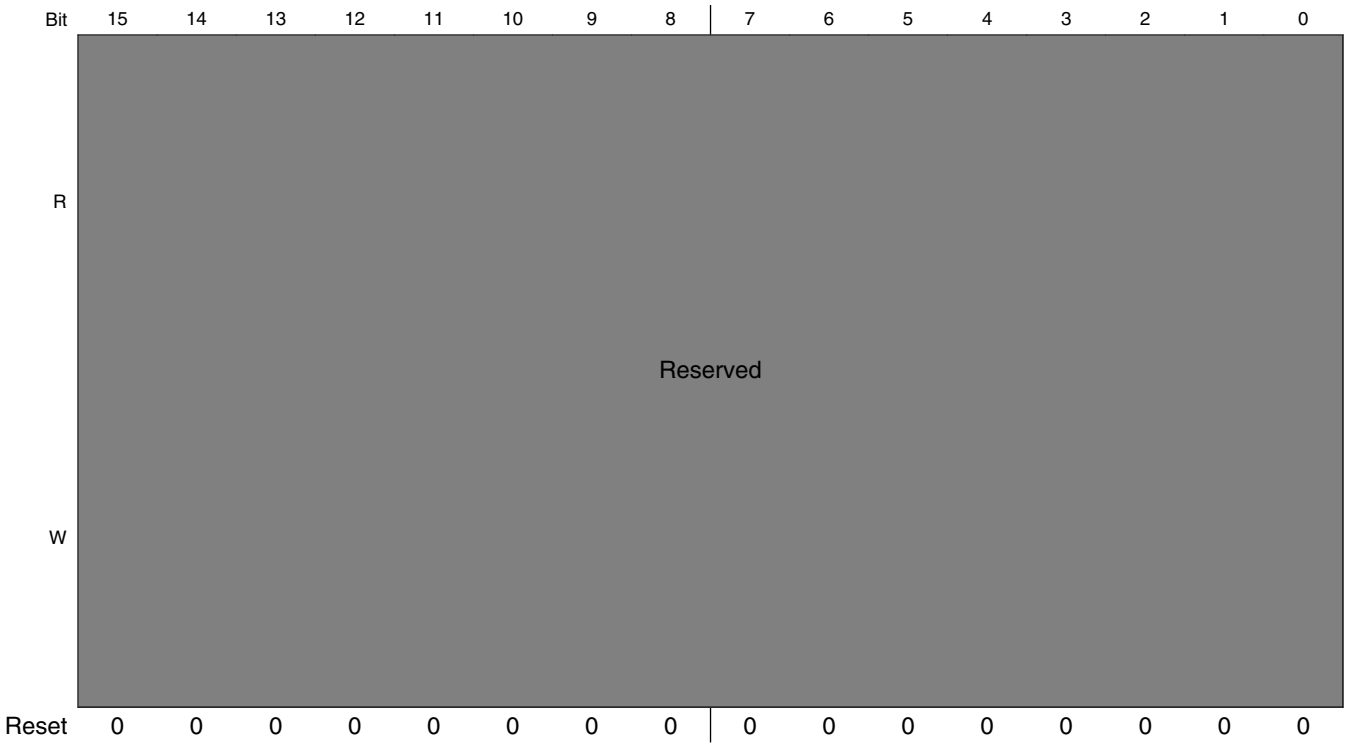
Field	Description
	Bit 0 is hardwired to 1; bits 7:1 are RW.

49.10.29 VC Resource Status Register n (PCIE_EP_VCRSR)

Offset: 0x140 + 0x18

Address: 1FF_C000h base + 158h offset = 1FF_C158h





PCIE_EP_VCRSR field descriptions

Field	Description
31–18 -	This field is reserved. Reserved
17 VC_Negotiation_Pending	VC Negotiation Pending
16 Port_Arbitration_Table_Status	Port Arbitration Table Status
-	This field is reserved. Reserved

49.11 PCIe CTRL RC Mode Memory Map/Register Definition

PCIE_RC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1FF_C000	Device ID and Vendor ID Register (PCIE_RC_DeviceID)	32	R	ABCD _16C3h	49.11.1/ 4318
1FF_C004	Command and Status Register (PCIE_RC_Command)	32	R/W	0000_0000h	49.11.2/ 4318

Table continues on the next page...

PCIE_RC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1FF_C008	Revision ID and Class Code Register (PCIE_RC_RevID)	32	R	0000_0000h	49.11.3/ 4321
1FF_C00C	BIST Register (PCIE_RC_BIST)	32	R/W	0000_0000h	49.11.4/ 4321
1FF_C010	Base Address 0 (PCIE_RC_BAR0)	32	R	0000_000Ch	49.11.5/ 4322
1FF_C014	Base Address 1 (PCIE_RC_BAR1)	32	R	0000_0000h	49.11.6/ 4325
1FF_C018	Bus Number Registers (PCIE_RC_BNR)	32	R	0000_0000h	49.11.7/ 4325
1FF_C01C	I/O Base Limit Secondary Status Register (PCIE_RC_IOBLSSR)	32	R/W	0000_0000h	49.11.8/ 4327
1FF_C020	Memory Base and Memory Limit Register (PCIE_RC_MEM_BLR)	32	R/W	0000_0000h	49.11.9/ 4329
1FF_C024	Prefetchable Memory Base and Limit Register (PCIE_RC_PREF_MEM_BLR)	32	R/W	0000_0000h	49.11.10/ 4330
1FF_C028	Prefetchable Base Upper 32 Bits Register (PCIE_RC_PREF_BASE_U32)	32	R/W	0000_0000h	49.11.11/ 4330
1FF_C02C	Prefetchable Limit Upper 32 Bits Register (PCIE_RC_PREF_LIM_U32)	32	R/W	0000_0000h	49.11.12/ 4331
1FF_C030	I/O Base and Limit Upper 16 Bits Register (PCIE_RC_IO_BASE_LIM_U16)	32	R/W	0000_0000h	49.11.13/ 4331
1FF_C034	Capability Pointer Register (PCIE_RC_CAPPR)	32	R	0000_0040h	49.11.14/ 4332
1FF_C038	Expansion ROM Base Address Register (PCIE_RC_EROMBAR)	32	R/W	0000_0000h	49.11.15/ 4332
1FF_C038	Expansion ROM BAR Mask Register (PCIE_RC_EROMMASK)	32	R/W	0000_0000h	49.11.16/ 4333
1FF_C040	Power Management Capability Register (PCIE_RC_PMCR)	32	R	DBC3_5001h	49.11.17/ 4334
1FF_C044	Power Management Control and Status Register (PCIE_RC_PMCSR)	32	R/W	0000_0004h	49.11.18/ 4337
1FF_C070	PCI Express Capability ID Register (PCIE_RC_CIDR)	32	R	0000_0000h	49.11.19/ 4338
1FF_C074	Device Capabilities Register (PCIE_RC_DCR)	32	R/W	0000_0000h	49.11.20/ 4341
1FF_C078	Device Control Register (PCIE_RC_DConR)	32	R/W	0000_0000h	49.11.21/ 4343
1FF_C07C	Link Capabilities Register (PCIE_RC_LCR)	32	R	0000_0000h	49.11.22/ 4345
1FF_C080	Link Control and Status Register (PCIE_RC_LCSR)	32	R/W	0000_0000h	49.11.23/ 4348
1FF_C084	Slot Capabilities Register (PCIE_RC_SCR)	32	R	0000_0000h	49.11.24/ 4350

Table continues on the next page...

PCIE_RC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1FF_C088	Slot Control and Status Register (PCIE_RC_SCSR)	32	R/W	0000_0000h	49.11.25/ 4353
1FF_C08C	Root Control and Capabilities Register (PCIE_RC_RCCR)	32	R/W	0000_0000h	49.11.26/ 4356
1FF_C090	Root Status Register (PCIE_RC_RSR)	32	w1c	0000_0000h	49.11.27/ 4358
1FF_C094	Device Capabilities 2 Register (PCIE_RC_DCR2)	32	R	0000_001Fh	49.11.28/ 4359
1FF_C098	Device Control and Status 2 Register (PCIE_RC_DCSR2)	32	R/W	0000_0000h	49.11.29/ 4361
1FF_C09C	Link Capabilities 2 Register (PCIE_RC_LCR2)	32	R	0000_0000h	49.11.30/ 4363
1FF_C0A0	Link Control and Status 2 Register (PCIE_RC_LCSR2)	32	R/W	0000_0000h	49.11.31/ 4365
1FF_C100	AER Capability Header (PCIE_RC_AER)	32	R/W	0000_0000h	49.11.32/ 4367
1FF_C104	Uncorrectable Error Status Register (PCIE_RC_UESR)	32	R/W	0000_0000h	49.11.33/ 4368
1FF_C108	Uncorrectable Error Mask Register (PCIE_RC_UEMR)	32	R/W	0000_0000h	49.11.34/ 4370
1FF_C10C	Uncorrectable Error Severity Register (PCIE_RC_UESevR)	32	R/W	000C_2031h	49.11.35/ 4372
1FF_C110	Correctable Error Status Register (PCIE_RC_CESR)	32	R/W	0000_0000h	49.11.36/ 4374
1FF_C114	Correctable Error Mask Register (PCIE_RC_CEMR)	32	R/W	0000_0000h	49.11.37/ 4375
1FF_C118	Advanced Capabilities and Control Register (PCIE_RC_ACCR)	32	R/W	0000_00A0h	49.11.38/ 4377
1FF_C11C	Header Log Register (PCIE_RC_HLR)	32	R	0000_0000h	49.11.39/ 4378
1FF_C12C	Root Error Command Register (PCIE_RC_RECR)	32	R/W	0000_0000h	49.11.40/ 4379
1FF_C130	Root Error Status Register (PCIE_RC_RESR)	32	R/W	0000_0000h	49.11.41/ 4380
1FF_C134	Error Source Identification Register (PCIE_RC_ESIR)	32	R	0000_0000h	49.11.42/ 4381
1FF_C140	VC Extended Capability Header (PCIE_RC_VCECHR)	32	R	0000_0012h	49.11.43/ 4382
1FF_C144	Port VC Capability Register 1 (PCIE_RC_PVCCR1)	32	R	0000_0000h	49.11.44/ 4383
1FF_C148	Port VC Capability Register 2 (PCIE_RC_PVCCR2)	32	R	0000_0000h	49.11.45/ 4384
1FF_C14C	Port VC Control and Status Register (PCIE_RC_PVCCSR)	32	R/W	0000_0000h	49.11.46/ 4385

Table continues on the next page...

PCIE_RC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1FF_C150	VC Resource Capability Register n (PCIE_RC_VCRCR)	32	R	0000_0000h	49.11.47/ 4387
1FF_C154	VC Resource Control Register n (PCIE_RC_VCRConR)	32	R/W	0000_0000h	49.11.48/ 4389
1FF_C158	VC Resource Status Register n (PCIE_RC_VCRSR)	32	R	0000_0000h	49.11.49/ 4391

49.11.1 Device ID and Vendor ID Register (PCIE_RC_DeviceID)

Offset: 0x00

The default values of both Device ID and Vendor ID are hardware configuration parameters. The application can overwrite the default values of both Device ID and Vendor ID through the DBI.

Address: 1FF_C000h base + 0h offset = 1FF_C000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																Reserved															
Reset	1	0	1	0	1	0	1	1	1	1	0	0	1	1	0	1	0	0	0	1	0	1	1	0	1	1	0	0	0	0	1	1

PCIE_RC_DeviceID field descriptions

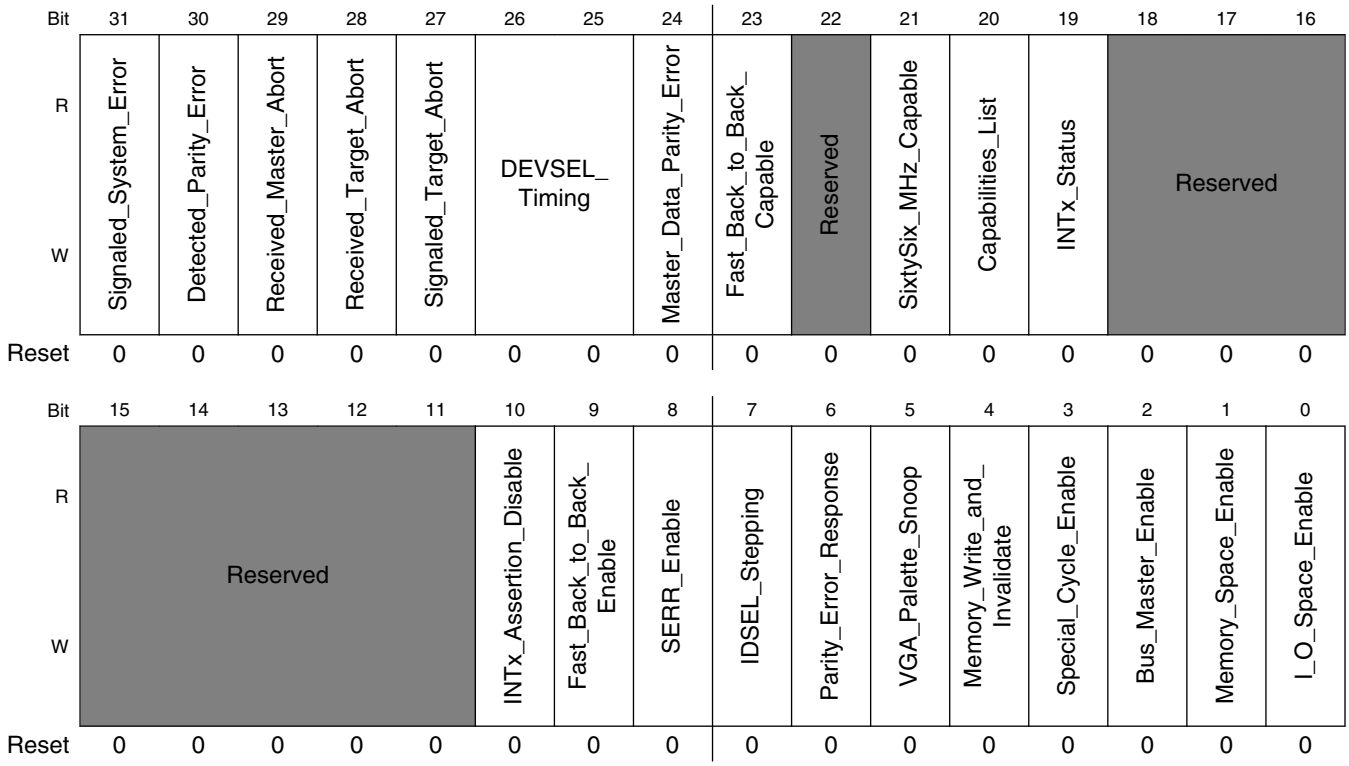
Field	Description
31–16 -	This field is reserved. Reserved
-	This field is reserved. Reserved

49.11.2 Command and Status Register (PCIE_RC_Command)

Offset: 0x04

Bytes: 0-1

Address: 1FF_C000h base + 4h offset = 1FF_C004h



PCIE_RC_Command field descriptions

Field	Description
31 Signaled_ System_Error	Signaled System Error
30 Detected_Parity_ Error	Detected Parity Error
29 Received_ Master_Abort	Received Master Abort
28 Received_ Target_Abort	Received Target Abort
27 Signaled_Target_ Abort	Signaled Target Abort
26–25 DEVSEL_Timing	DEVSEL Timing Not applicable for PCI Express. Hardwired to 0.
24 Master_Data_ Parity_Error	Master Data Parity Error
23 Fast_Back_to_ Back_Capable	Fast Back-to-Back Capable Not applicable for PCI Express. Hardwired to 0.

Table continues on the next page...

PCIE_RC_Command field descriptions (continued)

Field	Description
22 -	This field is reserved. Reserved
21 SixtySix_MHz_Capable	66 MHz Capable Not applicable for PCI Express. Hardwired to 0.
20 Capabilities_List	Capabilities List Indicates presence of an extended capability item. Hardwired to 1.
19 INTx_Status	INTx Status
18–16 -	This field is reserved. Reserved
15–11 -	This field is reserved. Reserved
10 INTx_Assertion_Disable	INTx Assertion Disable
9 Fast_Back_to_Back_Enable	Fast Back-to-Back Enable Not applicable for PCI Express. Must be hardwired to 0.
8 SERR_Enable	SERR# Enable
7 IDSEL_Stepping	IDSEL Stepping/Wait Cycle Control Not applicable for PCI Express. Must be hardwired to 0
6 Parity_Error_Response	Parity Error Response
5 VGA_Palette_Snoop	VGA Palette Snoop Not applicable for PCI Express. Must be hardwired to 0.
4 Memory_Write_and_Invalidate	Memory Write and Invalidate Not applicable for PCI Express. Must be hardwired to 0.
3 Special_Cycle_Enable	Special Cycle Enable Not applicable for PCI Express. Must be hardwired to 0.
2 Bus_Master_Enable	Bus Master Enable
1 Memory_Space_Enable	Memory Space Enable
0 I_O_Space_Enable	I/O Space Enable

49.11.3 Revision ID and Class Code Register (PCIE_RC_RevID)

Offset: 0x08

Byte: 0

Address: 1FF_C000h base + 8h offset = 1FF_C008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BASE_CLASS_CODE_N								SUB_CLASS_CODE_N								IF_CODE_N								CX_REVISION_ID_N							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_RC_RevID field descriptions

Field	Description
31–24 BASE_CLASS_CODE_N	Base Class Code, writable through the DBI
23–16 SUB_CLASS_CODE_N	Subclass Code, writable through the DBI
15–8 IF_CODE_N	Programming Interface, writable through the DBI
CX_REVISION_ID_N	Revision ID, writable through the DBI

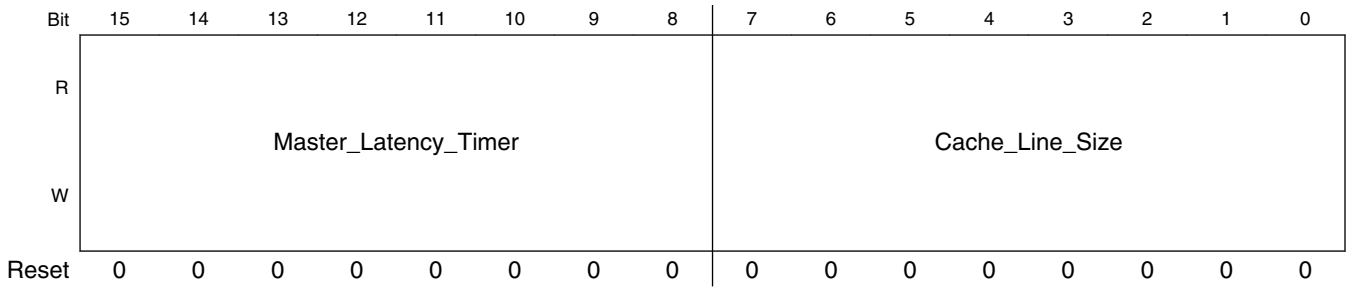
49.11.4 BIST Register (PCIE_RC_BIST)

Offset: 0x0C

Byte: 0

Address: 1FF_C000h base + Ch offset = 1FF_C00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Not_supported_by__core								Multi_Function_Device	Configuration_Header_Format						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



PCIE_RC_BIST field descriptions

Field	Description
31–24 Not_supported_by__core	The BIST register functions are not supported by the core. All 8 bits of the BIST register are hardwired to 0.
23 Multi_Function_Device	Multi Function Device The default value is 0 for a single function device (`CX_NFUNC = 1) or 1 for a multi-function device (`CX_NFUNC != 1). The Multi Function Device bit is writable through the DBI.
22–16 Configuration_Header_Format	Configuration Header Format Hardwired to 0 for type 0.
15–8 Master_Latency_Timer	Master Latency Timer Not applicable for PCI Express, hardwired to 0.
Cache_Line_Size	Cache Line Size The Cache Line Size register is RW for legacy compatibility purposes and is not applicable to PCI Express device functionality. Writing to the Cache Line Size register does not impact functionality of the core.

49.11.5 Base Address 0 (PCIE_RC_BAR0)

Offset: 0x10-0x24

The core provides three pairs of 32-bit BARs for each implemented function. Each pair (BARs 0 and 1, BARs 2 and 3, BARs 4 and 5) can be configured as follows:

- One 64-bit BAR: For example, BARs 0 and 1 are combined to form a single 64-bit BAR.
- Two 32-bit BARs: For example, BARs 0 and 1 are two independent 32-bit BARs.
- One 32-bit BAR: For example, BAR 0 is a 32-bit BAR and BAR 1 is either disabled or removed from the core altogether to reduce gate count.

In addition, you can configure each BAR to have its incoming Requests routed to either:

- RTRGT1
-

The following sections describe how to set up the BAR types and sizes by programming values into the base address registers. For more information about routing Requests to either RTRGT1 on a BAR-by- BAR basis, see Receive Filtering.

The contents of the six BARs determine the BAR configuration. The reset values of the BARs are determined by hardware configuration options.

At runtime, application software can overwrite the BAR contents to reconfigure the BARs (unless the affected BAR is removed during hardware configuration). Application software must observe the rules listed below when writing to the BARs.

The rules for BAR configuration are the same for all three pairs. Using BARs 0 and 1 as the example pair, the rules for BAR configuration are:

- Any pair (for example, BARs 0 and 1) can be configured as one 64-bit BAR, two 32-bit BARs, or one 32-bit BAR.
- BAR pairs cannot overlap to form a 64-bit BAR. For example, you cannot combine BARs 1 and 2 to form a 64-bit BAR.
-
- An I/O BAR must be a 32-bit BAR and cannot be prefetchable.
- If the device is configured as a PCI Express Endpoint (not a Legacy Endpoint), then any memory that is configured as prefetchable must be a 64-bit memory BAR.
- If BAR 0 is configured as a 64-bit BAR:

- BAR 1 is the upper 32 bits of the combined 64-bit BAR formed by BARs 0 and 1. Therefore, BAR 1 must be disabled and cannot be configured independently.
- BAR 0 must be a memory BAR and can be either prefetchable or non-prefetchable.
- The contents of the BAR 0 Mask register determine the number of writable bits in the 64-bit BAR, subject to the restrictions described in BAR Mask Register. The BAR 1 Mask register contains the upper 32 bits of the BAR 0 Mask value.
- BAR 0 can be disabled by writing 0 to bit 0 of the BAR 0 Mask register

- If BAR 0 is configured as a 32-bit BAR:

- You can configure BAR 1 as an independent 32-bit BAR
- BAR 0 can be configured as a memory BAR or an I/O BAR.
- The contents of the BAR 0 Mask register determine the number of writable bits in the 32-bit BAR 0, subject to the restrictions described in BAR Mask Registers.
- BAR 0 can be disabled by writing 0 to bit 0 of the BAR 0 Mask register

- When BAR 0 is configured as a 32-bit BAR, BAR 1 is available as an independent 32-bit BAR according to the following rules:

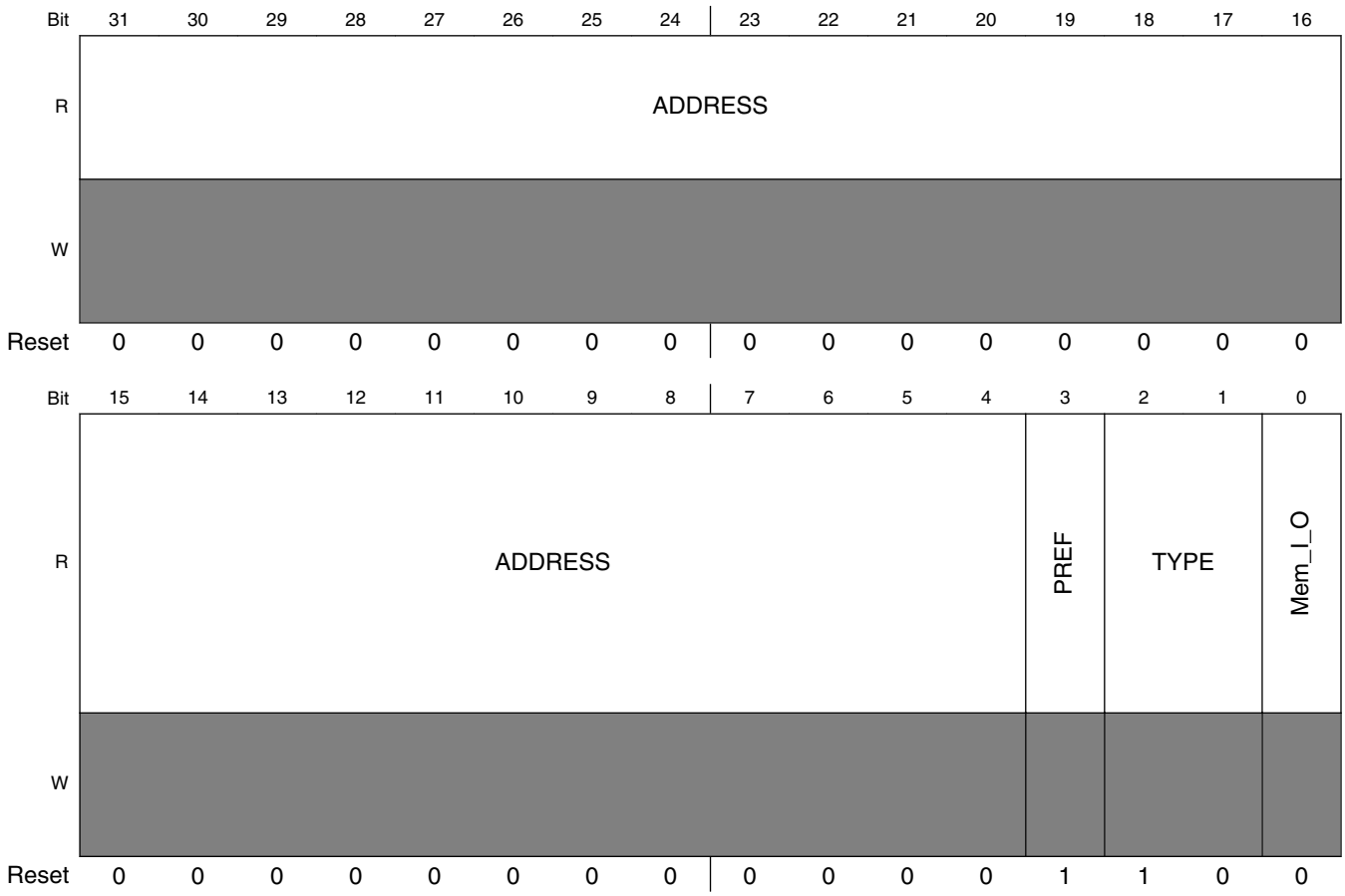
- BAR 1 can be configured as a memory BAR or an I/O BAR.

- The contents of the BAR 1 Mask register determine the number of writable bits in the 32-bit BAR 1, subject to the restrictions described in BAR Mask Registers.
-
-

The same rules apply for pairs 2/3 and 4/5.

Offset: 0x10 (if included in the core hardware configuration)

Address: 1FF_C000h base + 10h offset = 1FF_C010h



PCIE_RC_BAR0 field descriptions

Field	Description
31–4 ADDRESS	BAR 0 base address bits (for a 64-bit BAR, the remaining upper address bits are in BAR 1). The BAR 0 Mask value determines which address bits are masked.
3 PREF	If BAR 0 is an I/O BAR, bit 3 is the second least significant bit of the base address. Bits [3:0] are writable through the DBI. If BAR 0 is a memory BAR, bit 3 indicates if the memory region is prefetchable: 0 = Non-prefetchable 1 = Prefetchable

Table continues on the next page...

PCIE_RC_BAR0 field descriptions (continued)

Field	Description
2–1 TYPE	If BAR 0 is an I/O BAR, bit 2 the least significant bit of the base address and bit 1 is 0. Bits [3:0] are writable through the DBI. If BAR 0 is a memory BAR, bits [2:1] determine the BAR type: 00 = 32-bit BAR 10 = 64-bit BAR
0 Mem_I_O	Bits [3:0] are writable through the DBI. 0 = BAR 0 is a memory BAR 1 = BAR 0 is an I/O BAR

49.11.6 Base Address 1 (PCIE_RC_BAR1)

Address: 0x14

Address: 1FF_C000h base + 14h offset = 1FF_C014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADDRESS																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_RC_BAR1 field descriptions

Field	Description
ADDRESS	BAR 1 contains the upper 32 bits of the BAR 0 base address (bits [63:32]).

49.11.7 Bus Number Registers (PCIE_RC_BNR)

Address: 1FF_C000h base + 18h offset = 1FF_C018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SECONDARY_LAT_TMR								SUBORD_BUS_NUM								SECONDARY_BUS_NUM								PRIMARY_BUS_NUM							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

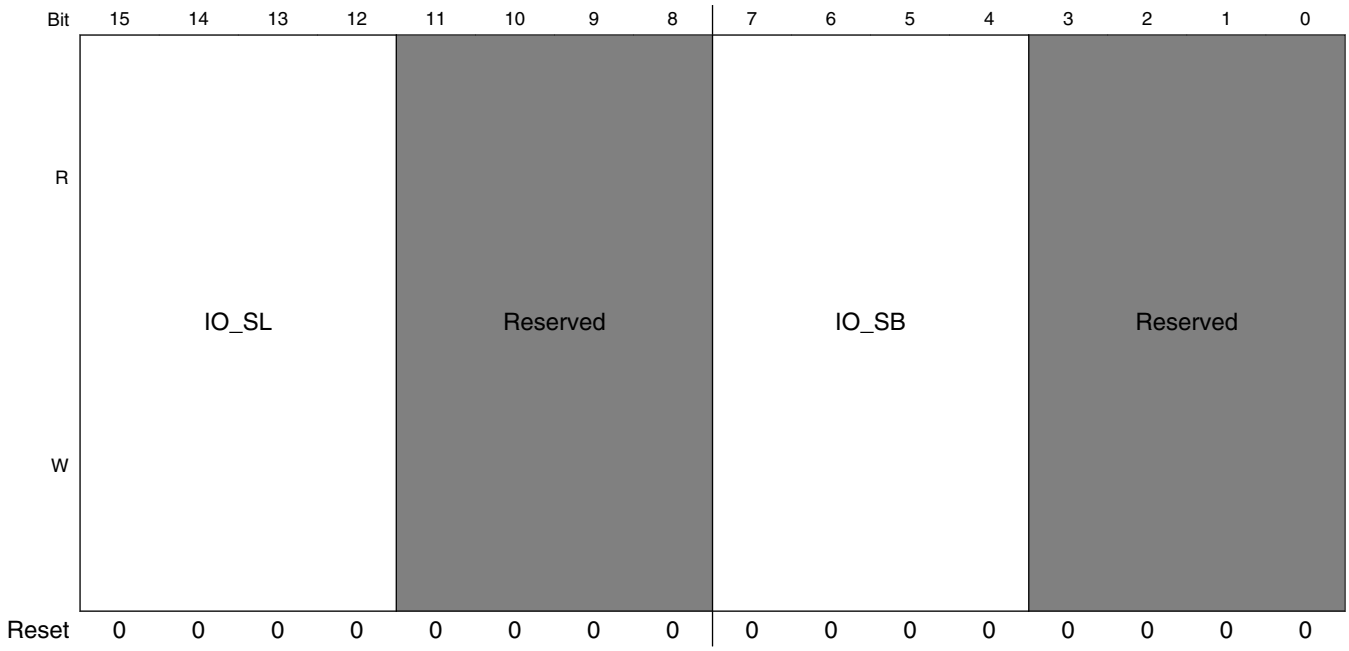
PCIE_RC_BNR field descriptions

Field	Description
31–24 SECONDARY_LAT_TMR	Secondary latency timer.
23–16 SUBORD_BUS_NUM	Subordinate bus number.
15–8 SECONDARY_BUS_NUM	Secondary bus number.
PRIMARY_BUS_NUM	Primary bus number.

49.11.8 I/O Base Limit Secondary Status Register (PCIE_RC_IOBLSSR)

Address: 1FF_C000h base + 1Ch offset = 1FF_C01Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DET_PARITY_ERR	RX_SYS_ERR	RX_MASTER_ABORT	RX_TARGET_ABORT	SIG_TARGET_ABORT	Reserved		MSTR_DAT_PARITY_ERR	FAST_B2B_CAP	Reserved	CAP_66M	Reserved				
W	w1c	w1c	w1c	w1c	w1c			w1c								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



PCIE_RC_IOBLSSR field descriptions

Field	Description
31 DET_PARITY_ERR	Detected Parity Error.
30 RX_SYS_ERR	Received System Error.
29 RX_MASTER_ABORT	Received Master Abort.
28 RX_TARGET_ABORT	Received Target Abort.
27 SIG_TARGET_ABORT	Signaled Target Abort.
26–25 -	This field is reserved. Reserved.
24 MSTR_DAT_PARITY_ERR	Master Data Parity Error.
23 FAST_B2B_CAP	Reserved.
22 -	This field is reserved. Reserved.
21 CAP_66M	66 MHz Capable. Not applicable to PCI Express, hardwired to 0.
20–16 -	This field is reserved. Reserved.

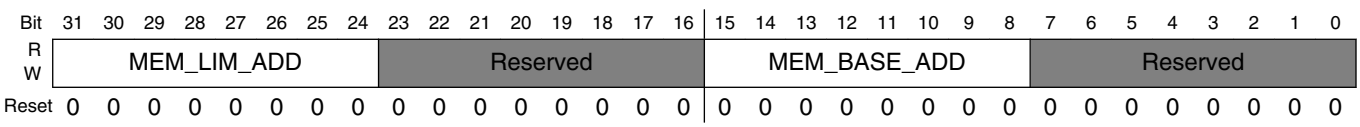
Table continues on the next page...

PCIE_RC_IOBLSSR field descriptions (continued)

Field	Description
15–12 IO_SL	I/O Space Limit.
11–8 -	This field is reserved. Reserved.
7–4 IO_SB	I/O Space Base.
-	This field is reserved. Reserved.

49.11.9 Memory Base and Memory Limit Register (PCIE_RC_MEM_BLR)

Address: 1FF_C000h base + 20h offset = 1FF_C020h



PCIE_RC_MEM_BLR field descriptions

Field	Description
31–24 MEM_LIM_ADD	Memory Limit Address.
23–16 -	This field is reserved. Reserved.
15–8 MEM_BASE_ ADD	Memory Base Address.
-	This field is reserved. Reserved.

49.11.10 Prefetchable Memory Base and Limit Register (PCIE_RC_PREF_MEM_BLR)

Address: 1FF_C000h base + 24h offset = 1FF_C024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER12_END_ADD												Reserved				UPPER12_START_ADD												Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_RC_PREF_MEM_BLR field descriptions

Field	Description
31–20 UPPER12_END_ADD	Upper 12 bits of 32-bit Prefetchable Memory End Address.
19–16 -	This field is reserved. Reserved.
15–4 UPPER12_START_ADD	Upper 12 bits of 32-bit Prefetchable Memory Start Address.
-	This field is reserved. Reserved.

49.11.11 Prefetchable Base Upper 32 Bits Register (PCIE_RC_PREF_BASE_U32)

Prefetchable Base Upper 32 Bits Register

Address: 1FF_C000h base + 28h offset = 1FF_C028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER32_BASE_PREF_MEM_ADD																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_RC_PREF_BASE_U32 field descriptions

Field	Description
UPPER32_BASE_PREF_MEM_ADD	Upper 32 Bits of Base Address of Prefetchable Memory Space. Used only when 64-bit prefetchable memory addressing is enabled.

49.11.12 Prefetchable Limit Upper 32 Bits Register (PCIE_RC_PREF_LIM_U32)

Prefetchable Limit Upper 32 Bits Register

Address: 1FF_C000h base + 2Ch offset = 1FF_C02Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER32_LIM_PREF_MEM_ADD																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_RC_PREF_LIM_U32 field descriptions

Field	Description
UPPER32_LIM_PREF_MEM_ADD	Upper 32 Bits of Limit Address of Prefetchable Memory Space. Used only when 64-bit prefetchable memory addressing is enabled.

49.11.13 I/O Base and Limit Upper 16 Bits Register (PCIE_RC_IO_BASE_LIM_U16)

I/O Base and Limit Upper 16 Bits Register

Address: 1FF_C000h base + 30h offset = 1FF_C030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER16_IO_LIM																UPPER16_IO_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_RC_IO_BASE_LIM_U16 field descriptions

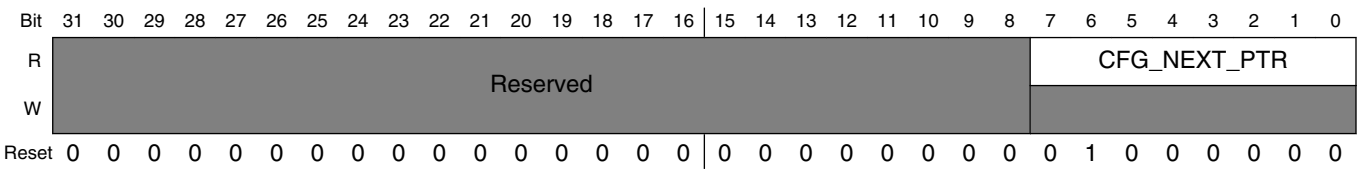
Field	Description
31–16 UPPER16_IO_LIM	Upper 16 Bits of I/O Limit (if 32-bit I/O decoding is supported for devices on the secondary side).
UPPER16_IO_BASE	Upper 16 Bits of I/O Base (if 32-bit I/O decoding is supported for devices on the secondary side).

49.11.14 Capability Pointer Register (PCIE_RC_CAPPR)

Offset: 0x34

Byte: 0

Address: 1FF_C000h base + 34h offset = 1FF_C034h



PCIE_RC_CAPPR field descriptions

Field	Description
31–8 -	This field is reserved. Reserved
CFG_NEXT_PTR	First Capability Pointer.

49.11.15 Expansion ROM Base Address Register (PCIE_RC_EROMBAR)

Offset: 0x38

Address: 1FF_C000h base + 38h offset = 1FF_C038h



PCIE_RC_EROMBAR field descriptions

Field	Description
31–11 ADDRESS	Expansion ROM Address
10–1 -	This field is reserved. Reserved
0 ENABLE	Expansion ROM Enable

49.11.16 Expansion ROM BAR Mask Register (PCIE_RC_EROMMASK)

Offset: 0x38 (same as the Expansion ROM BAR, but requires dbi_cs2 for write access)

Address: 1FF_C000h base + 38h offset = 1FF_C038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ROM_MASK_N															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ROM_MASK_N															ROM_BAR_ENABLED_N
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_RC_EROMMASK field descriptions

Field	Description
31–1 ROM_MASK_N	Indicates which Expansion ROM BAR bits to mask (make non-writable) from host software, which, in turn, determines the size of the BAR. For example, writing 0xFFF to the Expansion ROM BAR Mask register claims a 4096-byte BAR by masking bits 11:0 of the BAR from writing by host software. The maximum value is 0xFFFFFF because the maximum space that can be claimed by an Expansion ROM BAR is 16 MB. The Expansion ROM BAR Mask register is invisible to host software and not readable from the application. Application access depends on the value of

Table continues on the next page...

PCIE_RC_EROMMASK field descriptions (continued)

Field	Description
	• the Expansion ROM BAR Mask register is writable through the DBI. •
0 ROM_BAR_ ENABLED_N	Expansion ROM BAR Enable 0 Expansion ROM BAR is disabled 1 Expansion ROM BAR is enabled

49.11.17 Power Management Capability Register (PCIE_RC_PMCR)

The core implements power management capabilities. The Capability Pointer field in the configuration header points to the PCI Power Management registers as the first extended capability by default.

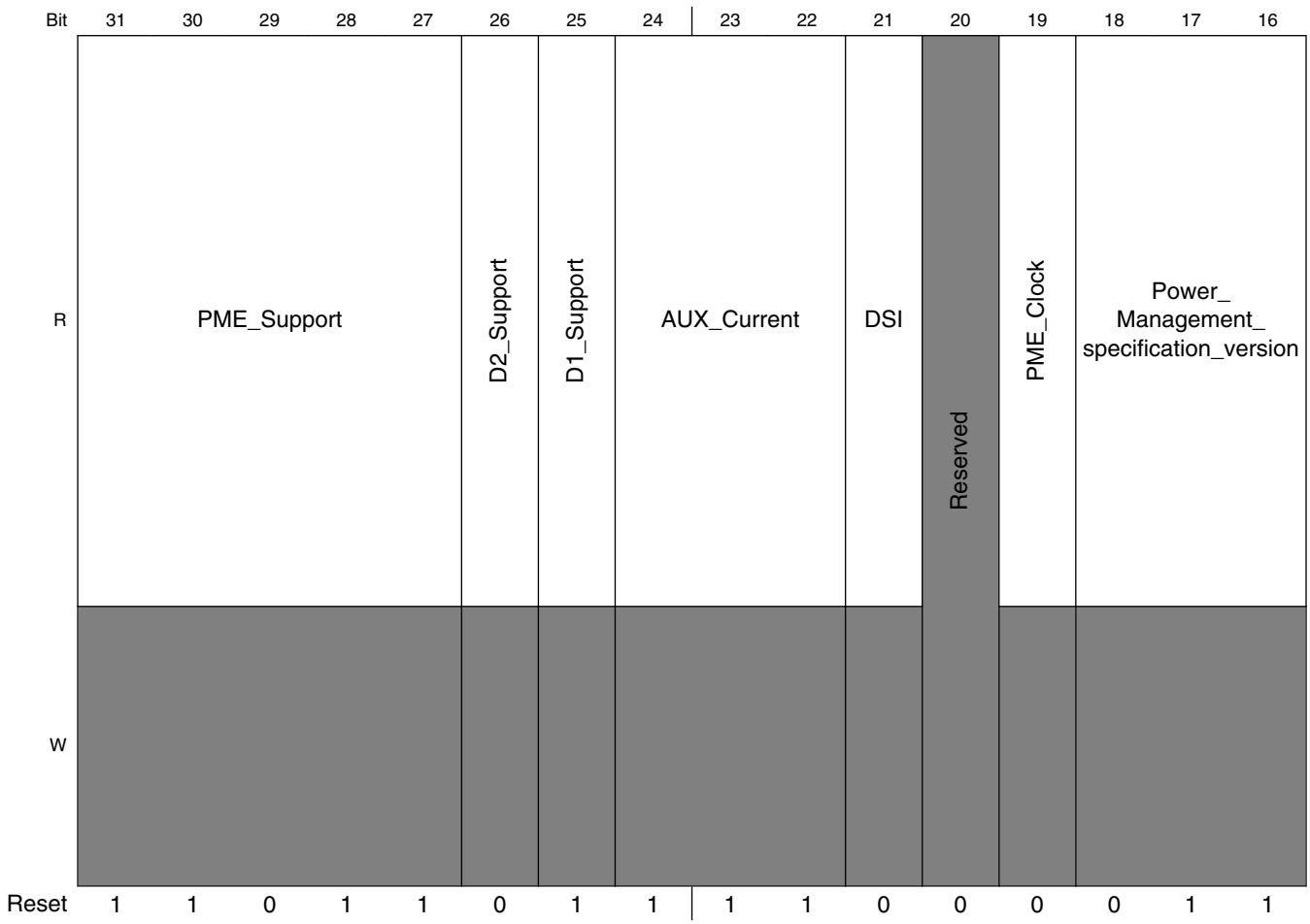
The extent of the power management implementation in the core includes:

- Power Management register space
- Link state information (provided to both the application logic and PHY interfaces)
- Power management-ready clock and reset implementation

The following sections describe the PCI Power Management registers implemented in the core. See the *PCI Power Management specification* and the *PCI Express 3.0 Specification* for more details.

Offset: `CFG_PM_CAP

Address: 1FF_C000h base + 40h offset = 1FF_C040h



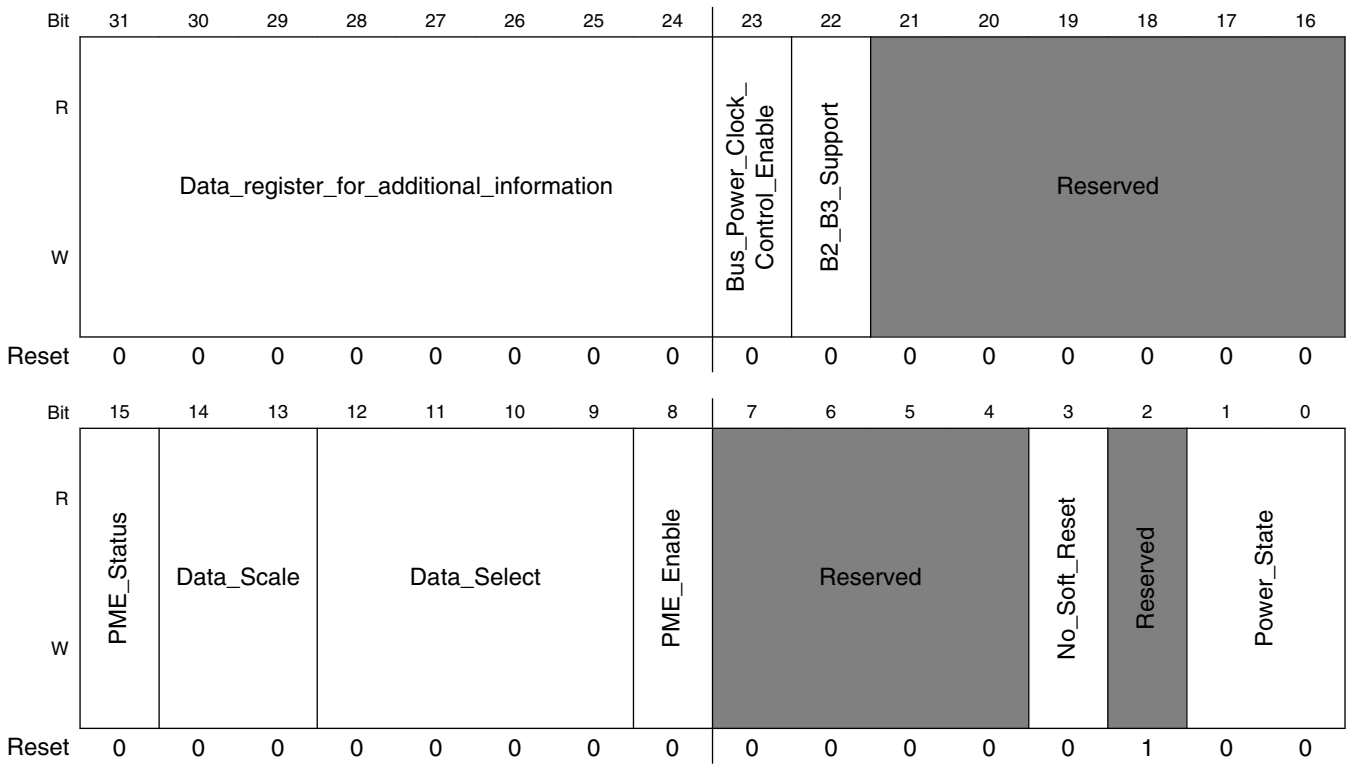
PCIE_RC_PMCR field descriptions

Field	Description
31–27 PME_Support	PME_Support Identifies the power states from which the core can generate PME Messages. A value of 0 for any bit indicates that the device (or function) is not capable of generating PME Messages while in that power state: • Bit 11: If set, PME Messages can be generated from D0 • Bit 12: If set, PME Messages can be generated from D1 • Bit 13: If set, PME Messages can be generated from D2 • Bit 14: If set, PME Messages can be generated from $D3_{hot}$ • Bit 15: If set, PME Messages can be generated from $D3_{cold}$ The PME_Support field is writable through the DBI.
26 D2_Support	D2 Support, writable through the DBI
25 D1_Support	D1 Support, writable through the DBI
24–22 AUX_Current	AUX Current, writable through the DBI
21 DSI	Device Specific Initialization (DSI), writable through the DBI
20 -	This field is reserved. Reserved
19 PME_Clock	PME Clock, hardwired to 0
18–16 Power_ Management_ specification_ version	Power Management specification version, writable through the DBI
15–8 Next_Capability_ Pointer	Next Capability Pointer See and .
Power_ Management_ Capability_ID	Power Management Capability ID

49.11.18 Power Management Control and Status Register (PCIE_RC_PMCSR)

Offset: `CFG_PM_CAP + 0x04

Address: 1FF_C000h base + 44h offset = 1FF_C044h



PCIE_RC_PMCSR field descriptions

Field	Description
31–24 Data_register_for_additional_information	Data register for additional information (not supported)
23 Bus_Power_Clock_Control_Enable	Bus Power/Clock Control Enable, hardwired to 0
22 B2_B3_Support	B2/B3 Support, hardwired to 0
21–16 -	This field is reserved. Reserved
15 PME_Status	PME Status

Table continues on the next page...

PCIE_RC_PMCSR field descriptions (continued)

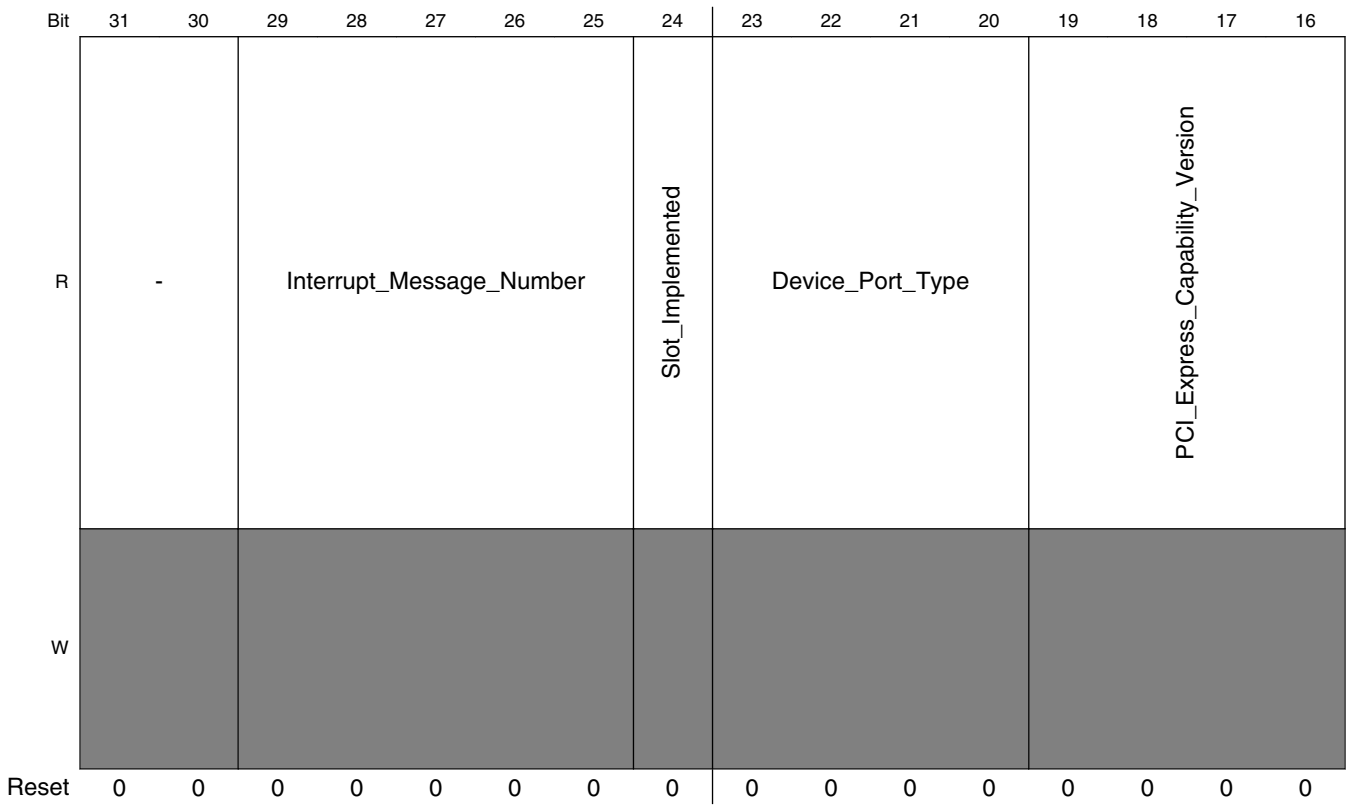
Field	Description
	Indicates if a previously enabled PME event occurred or not.
14–13 Data_Scale	Data Scale (not supported)
12–9 Data_Select	Data Select (not supported)
8 PME_Enable	PME Enable (sticky bit) A value of 1 indicates that the device is enabled to generate PME.
7–4 -	This field is reserved. Reserved
3 No_Soft_Reset	No Soft Reset, writable through the DBI
2 -	This field is reserved. Reserved
Power_State	Power State The written value is ignored if the specific state is not supported. Controls the device power state: 00 D0 01 D1 10 D2 11 D3

49.11.19 PCI Express Capability ID Register (PCIE_RC_CIDR)

The core implements the PCI Express Capability Structure as defined in the PCI Express 3.0 Specification.

Offset: CFG_PCIE_CAP + 0x00

Address: 1FF_C000h base + 70h offset = 1FF_C070h



PCIE_RC_CIDR field descriptions

Field	Description
31–30 -	RsvdP

Table continues on the next page...

PCIE_RC_CIDR field descriptions (continued)

Field	Description
29–25 Interrupt_ Message_ Number	Interrupt Message Number Updated by hardware, writable through the DBI.
24 Slot_ Implemented	Slot Implemented, writable through the DBI
23–20 Device_Port_ Type	Device/Port Type Indicates the specific type of this PCI Express Function. Supported encodings for RC and DM(RC mode) are: •4'b0100: Root Port of PCI Express Root Complex **NOTE:** Note: All other encodings (including those for PCI/PCI-X bridges and RC Integrated Endpoint) are NOT supported.
19–16 PCI_Express_ Capability_ Version	PCI Express Capability Version
15–8 Next_Capability_ Pointer	Next Capability Pointer
PCI_Express_ Capability_ID	PCI Express Capability ID

49.11.20 Device Capabilities Register (PCIE_RC_DCR)

Offset: `CFG_PCIE_CAP + 0x04

Address: 1FF_C000h base + 74h offset = 1FF_C074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				Captured_ Slot_Power_ Limit_Scale		Captured_Slot_Power_Limit_Value								Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Role_Based_Error_ Reporting	Reserved	Reserved	Reserved	Endpoint_L1_ Acceptable_Latency			Endpoint_L0s_ Acceptable_Latency			Extended_Tag_Field_ Supported	Phantom_ Function_ Supported		Max_Payload_Size_ Supported		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_RC_DCR field descriptions

Field	Description
31–28 -	This field is reserved. Reserved
27–26 Captured_Slot_ Power_Limit_ Scale	Captured Slot Power Limit Scale Upstream port only.
25–18 Captured_Slot_ Power_Limit_ Value	Captured Slot Power Limit Value Upstream port only.
17–16 -	This field is reserved. Reserved
15 Role_Based_ Error_Reporting	Role-Based Error Reporting, writable through the DBI. Required to be set for device compliant to 1.1 spec and later.

Table continues on the next page...

PCIE_RC_DCR field descriptions (continued)

Field	Description
14 -	This field is reserved. Reserved Undefined since PCI Express 1.1 (Was Power Indicator Present for PCI Express 1.0a)
13 -	This field is reserved. Reserved Undefined since PCI Express 1.1 (Was Attention Indicator Present for PCI Express 1.0a)
12 -	This field is reserved. Reserved Undefined since PCI Express 1.1 (Was Attention Button Present for PCI Express 1.0a)
11–9 Endpoint_L1_Acceptable_Latency	Endpoint L1 Acceptable Latency Must be 0x0 for non-Endpoint devices.
8–6 Endpoint_L0s_Acceptable_Latency	Endpoint L0s Acceptable Latency Must be 0x0 for non-Endpoint devices.
5 Extended_Tag_Field_Supported	Extended Tag Field Supported This bit is writable through the DBI. However, if the core supports only 5 bits of TAG, then the application must not write a 1 to this field because the hardware to support more than 32 tags are not implemented.
4–3 Phantom_Function_Supported	Phantom Function Supported This field is writable through the DBI. However, Phantom Function is not supported. Therefore, the application must not write any value other than 0x0 to this field.
Max_Payload_Size_Supported	Max_Payload_Size Supported, writable through the DBI

49.11.21 Device Control Register (PCIE_RC_DConR)

Offset: `CFG_PCIE_CAP + 0x08

Address: 1FF_C000h base + 78h offset = 1FF_C078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved										Transaction_Pending	Aux_Power_Detected	Unsupported_Request_Detected	Fatal_Error_Detected	Non_Fatal_Error_detected	Correctable_Error_Detected
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Max_Read_Request_Size			Enable_No_Snoop	AUX_Power_PM_Enable	Phantom_Function_Enable	Extended_Tag_Field_Enable	Max_Payload_Size			Enable_Relaxed_Ordering	Unsupported_Request_Reporting_Enable	Fatal_Error_Reporting_Enable	Non_Fatal_Error_Reporting_Enable	Correctable_Error_Reporting_Enable
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_RC_DConR field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21 Transaction_Pending	Transaction Pending Hard-wired to 0.
20 Aux_Power_Detected	Aux Power Detected From sys_aux_pwr_det input port.
19 Unsupported_Request_Detected	Unsupported Request Detected Errors are logged in this register regardless of whether error reporting is enabled in the Device Control register.
18 Fatal_Error_Detected	Fatal Error Detected Errors are logged in this register regardless of whether error reporting is enabled in the Device Control register.

Table continues on the next page...

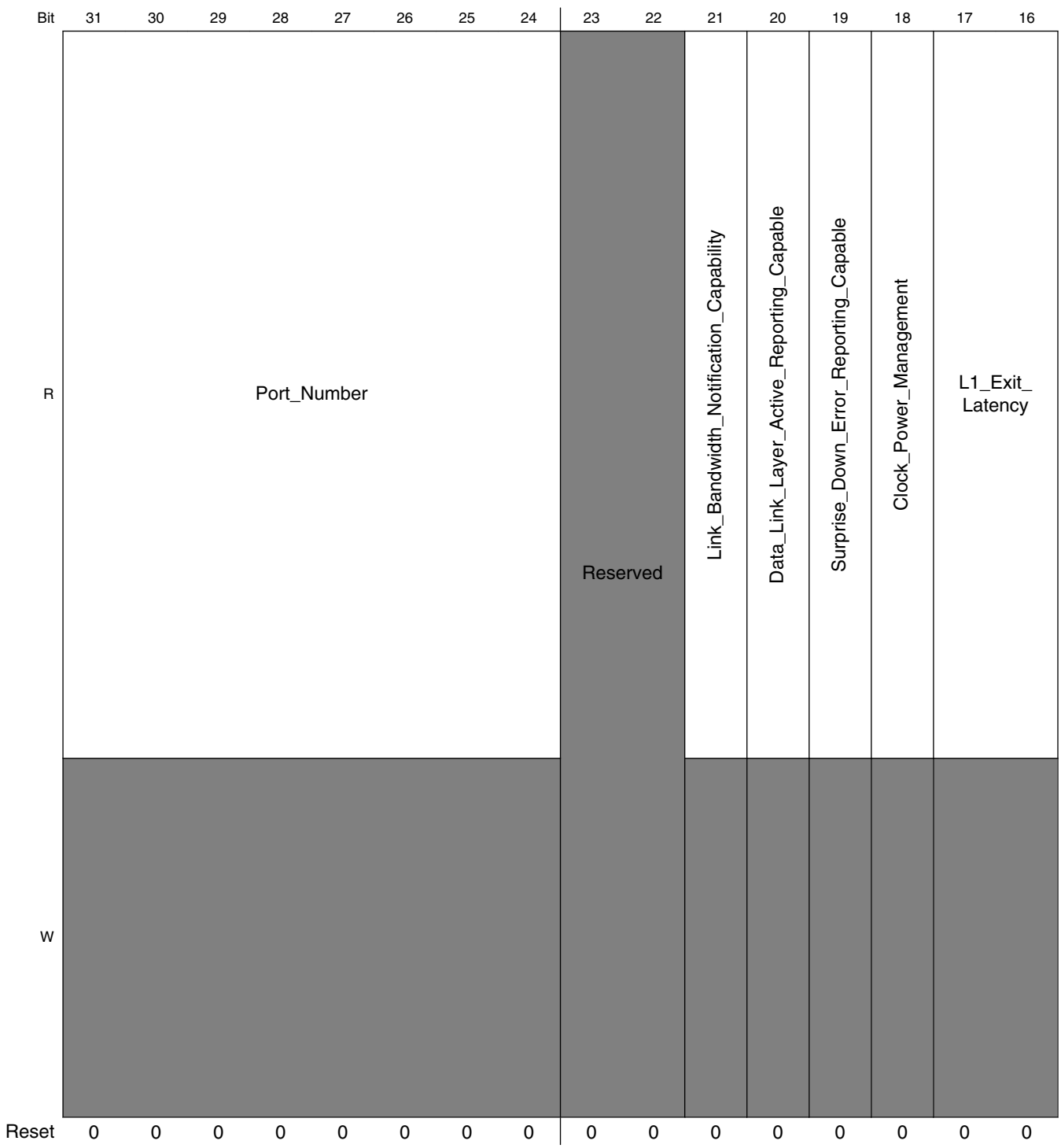
PCIE_RC_DConR field descriptions (continued)

Field	Description
17 Non_Fatal_ Error_detected	Non-Fatal Error detected Errors are logged in this register regardless of whether error reporting is enabled in the Device Control register.
16 Correctable_ Error_Detected	Correctable Error Detected Errors are logged in this register regardless of whether error reporting is enabled in the Device Control register.
15 -	This field is reserved. Reserved
14–12 Max_Read_ Request_Size	Max_Read_Request_Size
11 Enable_No_ Snoop	Enable No Snoop
10 AUX_Power_ PM_Enable	AUX Power PM Enable
9 Phantom_ Function_Enable	Phantom Function Enable
8 Extended_Tag_ Field_Enable	Extended Tag Field Enable
7–5 Max_Payload_ Size	Max_Payload_Size
4 Enable_Relaxed_ Ordering	Enable Relaxed Ordering
3 Unsupported_ Request_ Reporting_ Enable	Unsupported Request Reporting Enable
2 Fatal_Error_ Reporting_ Enable	Fatal Error Reporting Enable
1 Non_Fatal_ Error_Reporting_ Enable	Non-Fatal Error Reporting Enable
0 Correctable_ Error_Reporting_ Enable	Correctable Error Reporting Enable

49.11.22 Link Capabilities Register (PCIE_RC_LCR)

Offset: `CFG_PCIE_CAP + 0x0C

Address: 1FF_C000h base + 7Ch offset = 1FF_C07Ch



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	L1_Exit_Latency	L0s_Exit_Latency			Active_State_Link_PM_Support		Maximum_Link_Width						Max_Link_Speeds			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_RC_LCR field descriptions

Field	Description
31–24 Port_Number	Port Number
23–22 -	This field is reserved. Reserved
21 Link_Bandwidth_Notification_Capability	Link Bandwidth Notification Capability Hardwired to 1 for Downstream Ports and 0 for Upstream Ports.
20 Data_Link_Layer_Active_Reporting_Capable	Data Link Layer Active Reporting Capable Hardwired to 1 for Downstream Ports and 0 for Upstream Ports.
19 Surprise_Down_Error_Reporting_Capable	Surprise Down Error Reporting Capable Not supported, hardwired to 0x0.

Table continues on the next page...

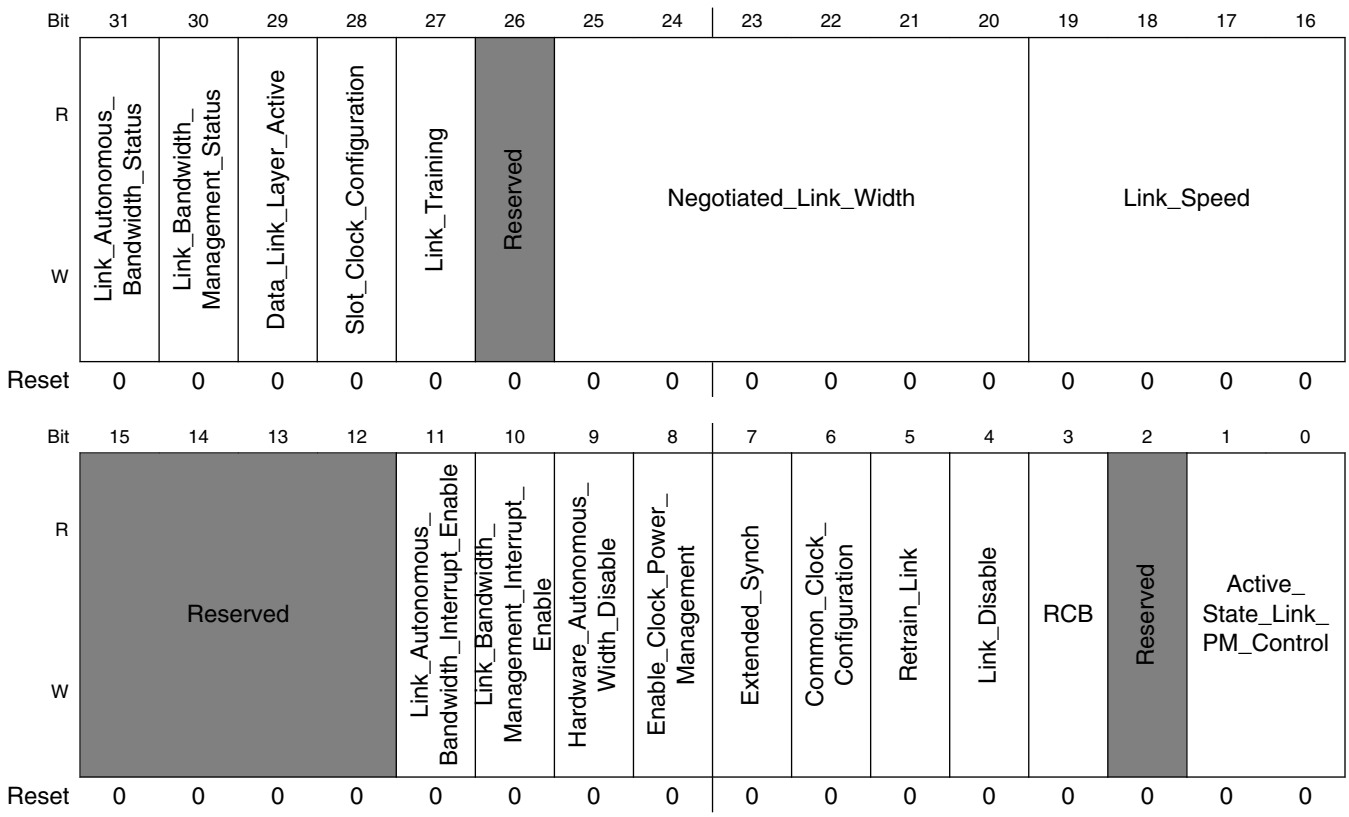
PCIE_RC_LCR field descriptions (continued)

Field	Description
18 Clock_Power_Management	Clock Power Management Component can tolerate the removal of refclk via CLKREQ# (if supported). Hardwired to 0 for downstream ports. Writable through the DBI.
17–15 L1_Exit_Latency	L1 Exit Latency Writable through the DBI.
14–12 L0s_Exit_Latency	L0s Exit Latency Writable through the DBI.
11–10 Active_State_Link_PM_Support	Active State Link PM Support The default value is the value you specify during core configuration, writable through the DBI.
9–4 Maximum_Link_Width	Maximum Link Width Writable through the DBI.
Max_Link_Speeds	Max Link Speeds Indicates the supported maximum Link speeds of the associated Port. The encoding is the binary value of the bit location in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the maximum Link speed. This field is writable through the DBI. 0001 Gen1 2.5 GT/s 0010 Gen2 5.0 GT/s 0100 Reserved

49.11.23 Link Control and Status Register (PCIE_RC_LCSR)

Offset: `CFG_PCIE_CAP + 0x10

Address: 1FF_C000h base + 80h offset = 1FF_C080h



PCIE_RC_LCSR field descriptions

Field	Description
31 Link_ Autonomous_ Bandwidth_ Status	Link Autonomous Bandwidth Status This bit is set by hardware to indicate that hardware has autonomously changed Link speed or width, without the Port transitioning through DL_Down status, for reasons other than to attempt to correct unreliable Link operation. This bit must be set if the Physical Layer reports a speed or a width change was initiated by the Downstream component that was indicated as an autonomous change. **NOTE:** This bit is not applicable and is reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges.
30 Link_Bandwidth_ Management_ Status	Link Bandwidth Management Status This bit is set by hardware to indicate that either of the following has occurred without the Port transitioning through DL_Down status: •A Link retraining has completed following a write of 1b to the Retrain Link bit.

Table continues on the next page...

PCIE_RC_LCSR field descriptions (continued)

Field	Description
	•Hardware has changed Link speed or width to attempt to correct unreliable Link operation, either through an LTSSM timeout or a higher level process. This bit must be set if the Physical Layer reports a speed or width change was initiated by the Downstream component that was not indicated as an autonomous change. **NOTE:** : This bit is set following any write of 1b to the Retrain Link bit, including when the Link is in the process of retraining for some other reason. **NOTE:** This bit is not applicable and is reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges.
29 Data_Link_Layer_Active	Data Link Layer Active This bit must be implemented if the corresponding Data Link Layer Link Active Reporting capability bit is implemented. Otherwise, this bit must be hardwired to 0b.
28 Slot_Clock_Configuration	Slot Clock Configuration Indicates that the component uses the same physical reference clock that the platform provides on the connector. The default value is the value you select during hardware configuration, writable through the DBI.
27 Link_Training	Link Training This bit is not applicable and is reserved for Endpoints, PCI Express to PCI/PCI-X bridges.
26 -	This field is reserved. Reserved Undefined for PCI Express 1.1 (Was Training Error for PCI Express 1.0a)
25–20 Negotiated_Link_Width	Negotiated Link Width Set automatically by hardware after Link initialization. The value is undefined when link is not up.
19–16 Link_Speed	Link Speed Indicates the negotiated Link speed. The encoding is the binary value of the bit location in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the current Link speed. Possible values are: 0001 Gen1 2.5 GT/s 0010 Gen2 5.0 GT/s
15–12 -	This field is reserved. Reserved
11 Link_Autonomous_Bandwidth_Interrupt_Enable	Link Autonomous Bandwidth Interrupt Enable When set, this bit enables the generation of an interrupt to indicate that the Link Autonomous Bandwidth Status bit has been set. **NOTE:** This bit is not applicable and is reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges.
10 Link_Bandwidth_Management_Interrupt_Enable	Link Bandwidth Management Interrupt Enable When set, this bit enables the generation of an interrupt to indicate that the Link Bandwidth Management Status bit has been set. **NOTE:** This bit is not applicable and is reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges.

Table continues on the next page...

PCIE_RC_LCSR field descriptions (continued)

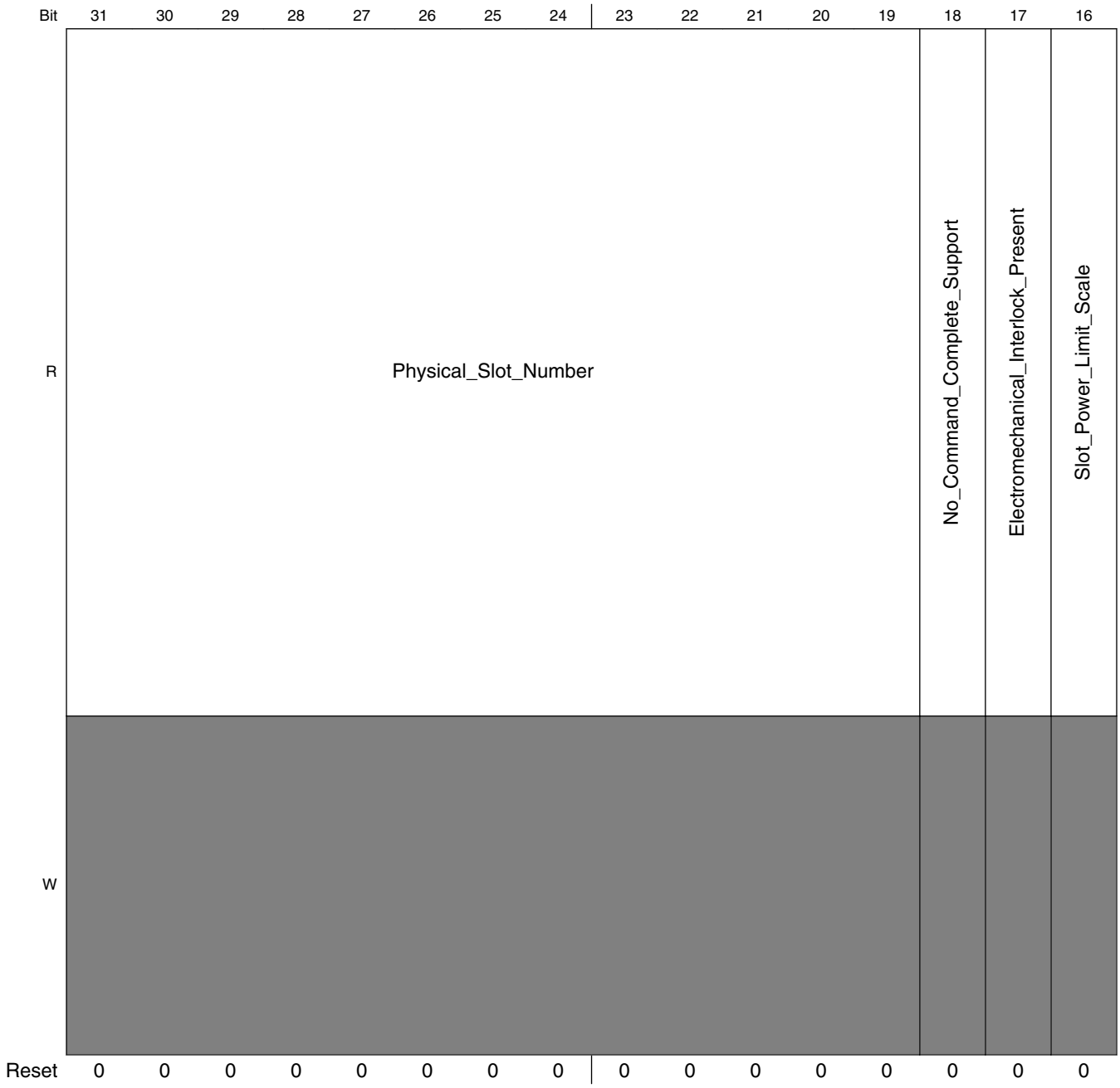
Field	Description
9 Hardware_ Autonomous_ Width_Disable	Hardware Autonomous Width Disable Not supported, hardwired to 0.
8 Enable_Clock_ Power_ Management	Enable Clock Power Management Hardwired to 0 if Clock Power Management is disabled in the Link Capabilities register.
7 Extended_Synch	Extended Synch
6 Common_Clock_ Configuration	Common Clock Configuration
5 Retrain_Link	Retrain Link This bit is reserved for PCI Express-to-PCI/PCI-X bridges.
4 Link_Disable	Link Disable This bit is reserved for PCI Express-to-PCI/PCI-X bridges.
3 RCB	Read Completion Boundary (RCB) RC: Writable through DBI
2 -	This field is reserved. Reserved
Active_State_ Link_PM_Control	Active State Link PM Control

49.11.24 Slot Capabilities Register (PCIE_RC_SCR)

This section applies only to Downstream Ports (for example, RC.

Offset: `CFG_PCIE_CAP + 0x14

Address: 1FF_C000h base + 84h offset = 1FF_C084h



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Slot_Power_Limit_Scale	Slot_Power_Limit_Value								Hot_Plug_Capable	Hot_Plug_Surprise	Power_Indicator_Present	Attention_Indicator_Present_1	MRL_Sensor_Present	Power_Controller_Present_0	Attention_Indicator_Present
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_RC_SCR field descriptions

Field	Description
31–19 Physical_Slot_Number	Physical Slot Number, writable through the DBI
18 No_Command_ Complete_Support	No Command Complete Support, writable through the DBI
17 Electromechanical_ Interlock_Present	Electromechanical Interlock Present, writable through the DBI

Table continues on the next page...

PCIE_RC_SCR field descriptions (continued)

Field	Description
16–15 Slot_Power_Limit_Scale	Slot Power Limit Scale, writable through the DBI
14–7 Slot_Power_Limit_Value	Slot Power Limit Value, writable through the DBI
6 Hot_Plug_Capable	Hot-Plug Capable, writable through the DBI
5 Hot_Plug_Surprise	Hot-Plug Surprise, writable through the DBI
4 Power_Indicator_Present	Power Indicator Present, writable through the DBI
3 Attention_Indicator_Present_1	Attention Indicator Present, writable through the DBI
2 MRL_Sensor_Present	MRL Sensor Present, writable through the DBI
1 Power_Controller_Present_0	Power Controller Present, writable through the DBI
0 Attention_Indicator_Present	Attention Indicator Present, writable through the DBI

49.11.25 Slot Control and Status Register (PCIE_RC_SCSR)

This section applies only to Downstream Ports (for example, RC.

Offset: `CFG_PCIE_CAP + 0x18

Address: 1FF_C000h base + 88h offset = 1FF_C088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved							Data_Link_Layer_State_Changed	Electromechanical_Interlock_Status	Presence_Detect_State	MRL_Sensor_State	Command_Completed	Presence_Detect_Changed	MRL_Sensor_Changed	Power_Fault_Detected	Attention_Button_Pressed
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			Data_Link_Layer_State_Changed_Enable	Electromechanical_Interlock_Control	Power_Controller_Control	Power_Indicator_Control		Attention_Indicator_Control		Hot_Plug_Interrupt_Enable	Command_Completed_Interrupt_Enable	Presence_Detect_Changed_Enable	MRL_Sensor_Changed_Enable	Power_Fault_Detected_Enable	Attention_Button_Pressed_Enable
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_RC_SCSR field descriptions

Field	Description
31–25 -	This field is reserved. Reserved
24 Data_Link_Layer_State_Changed	Data Link Layer State Changed
23 Electromechanical_Interlock_Status	Electromechanical Interlock Status
22 Presence_Detect_State	Presence Detect State
21 MRL_Sensor_State	MRL Sensor State
20 Command_Completed	Command Completed
19 Presence_Detect_Changed	Presence Detect Changed
18 MRL_Sensor_Changed	MRL Sensor Changed
17 Power_Fault_Detected	Power Fault Detected
16 Attention_Button_Pressed	Attention Button Pressed
15–13 -	This field is reserved. Reserved
12 Data_Link_Layer_State_Changed_Enable	Data Link Layer State Changed Enable
11 Electromechanical_Interlock_Control	Electromechanical Interlock Control

Table continues on the next page...

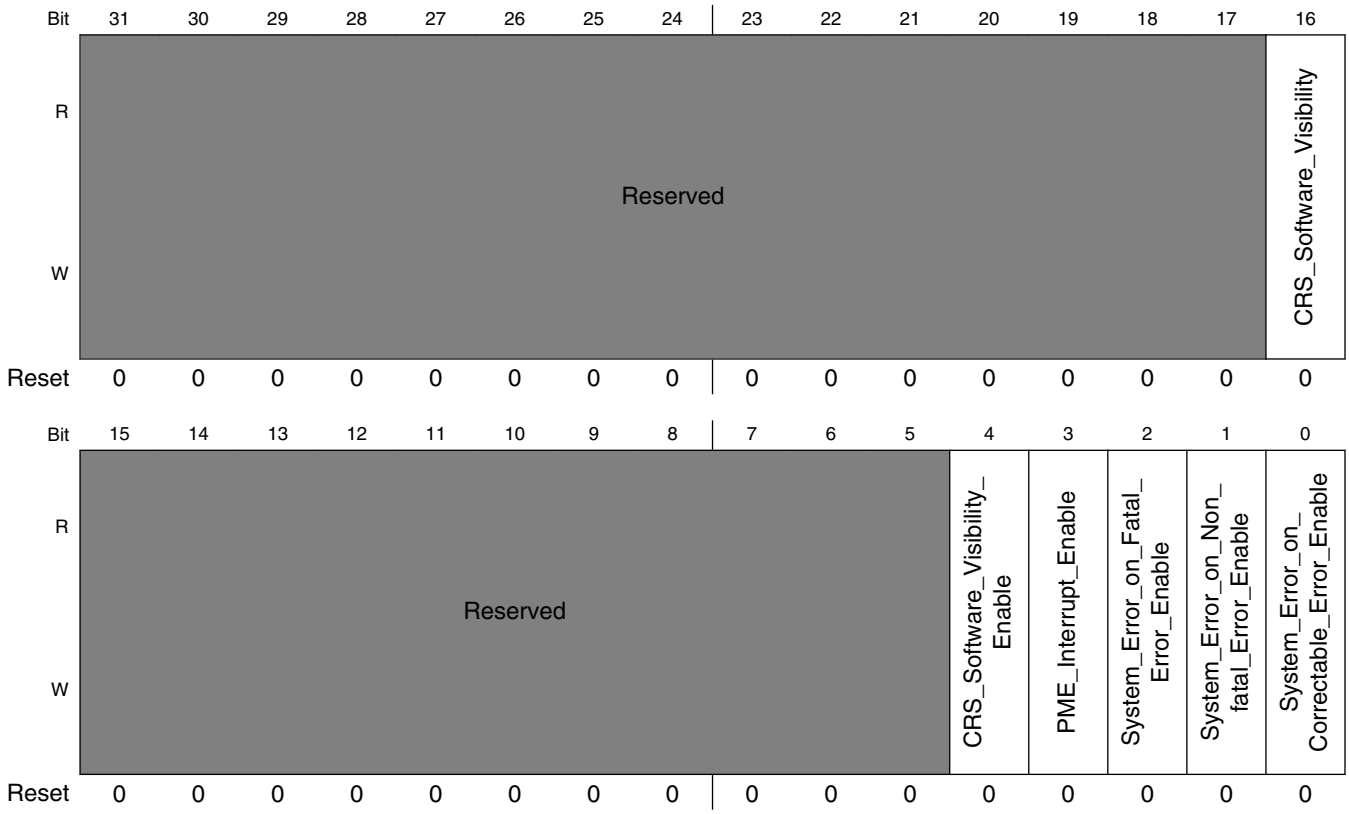
PCIE_RC_SCSR field descriptions (continued)

Field	Description
10 Power_Controller_ Control	Power Controller Control
9–8 Power_Indicator_ Control	Power Indicator Control
7–6 Attention_Indicator_ Control	Attention Indicator Control
5 Hot_Plug_Interrupt_ Enable	Hot-Plug Interrupt Enable
4 Command_ Completed_Interrupt_ Enable	Command Completed Interrupt Enable
3 Presence_Detect_ Changed_Enable	Presence Detect Changed Enable
2 MRL_Sensor_ Changed_Enable	MRL Sensor Changed Enable
1 Power_Fault_ Detected_Enable	Power Fault Detected Enable
0 Attention_Button_ Pressed_Enable	Attention Button Pressed Enable

49.11.26 Root Control and Capabilities Register (PCIE_RC_RCCR)

Offset: `CFG_PCIE_CAP + 0x1C

Address: 1FF_C000h base + 8Ch offset = 1FF_C08Ch



PCIE_RC_RCCR field descriptions

Field	Description
31–17 -	This field is reserved. Reserved
16 CRS_Software_Visibility	CRS Software Visibility Not supported, hardwired to 0x0.
15–5 -	This field is reserved. Reserved
4 CRS_Software_Visibility_Enable	CRS Software Visibility Enable Not supported, hardwired to 0x0.

Table continues on the next page...

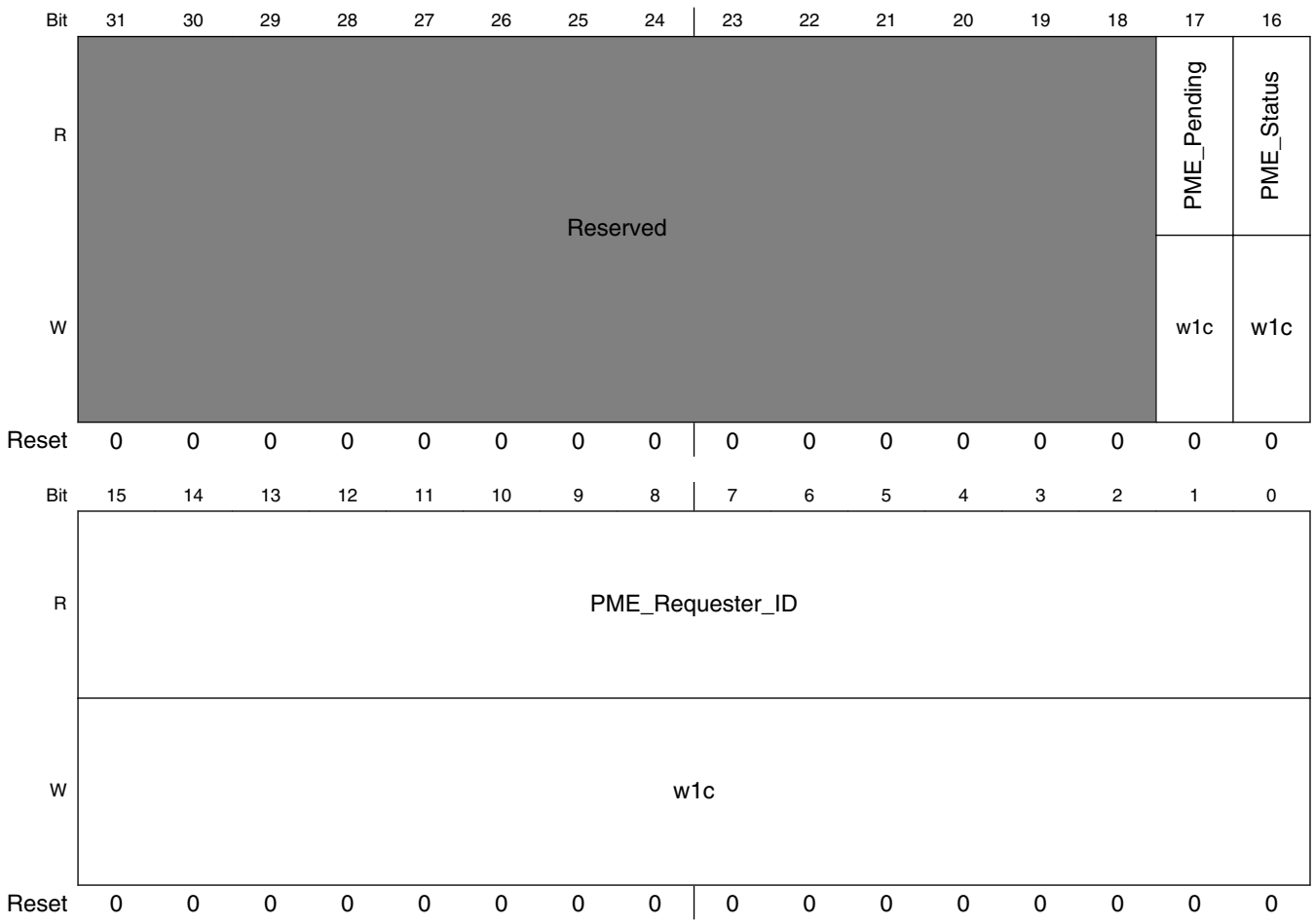
PCIE_RC_RCCR field descriptions (continued)

Field	Description
3 PME_Interrupt_Enable	PME Interrupt Enable
2 System_Error_on_Fatal_Error_Enable	System Error on Fatal Error Enable
1 System_Error_on_Non_fatal_Error_Enable	System Error on Non-fatal Error Enable
0 System_Error_on_Correctable_Error_Enable	System Error on Correctable Error Enable

49.11.27 Root Status Register (PCIE_RC_RSR)

Offset: `CFG_PCIE_CAP + 0x20

Address: 1FF_C000h base + 90h offset = 1FF_C090h



PCIE_RC_RSR field descriptions

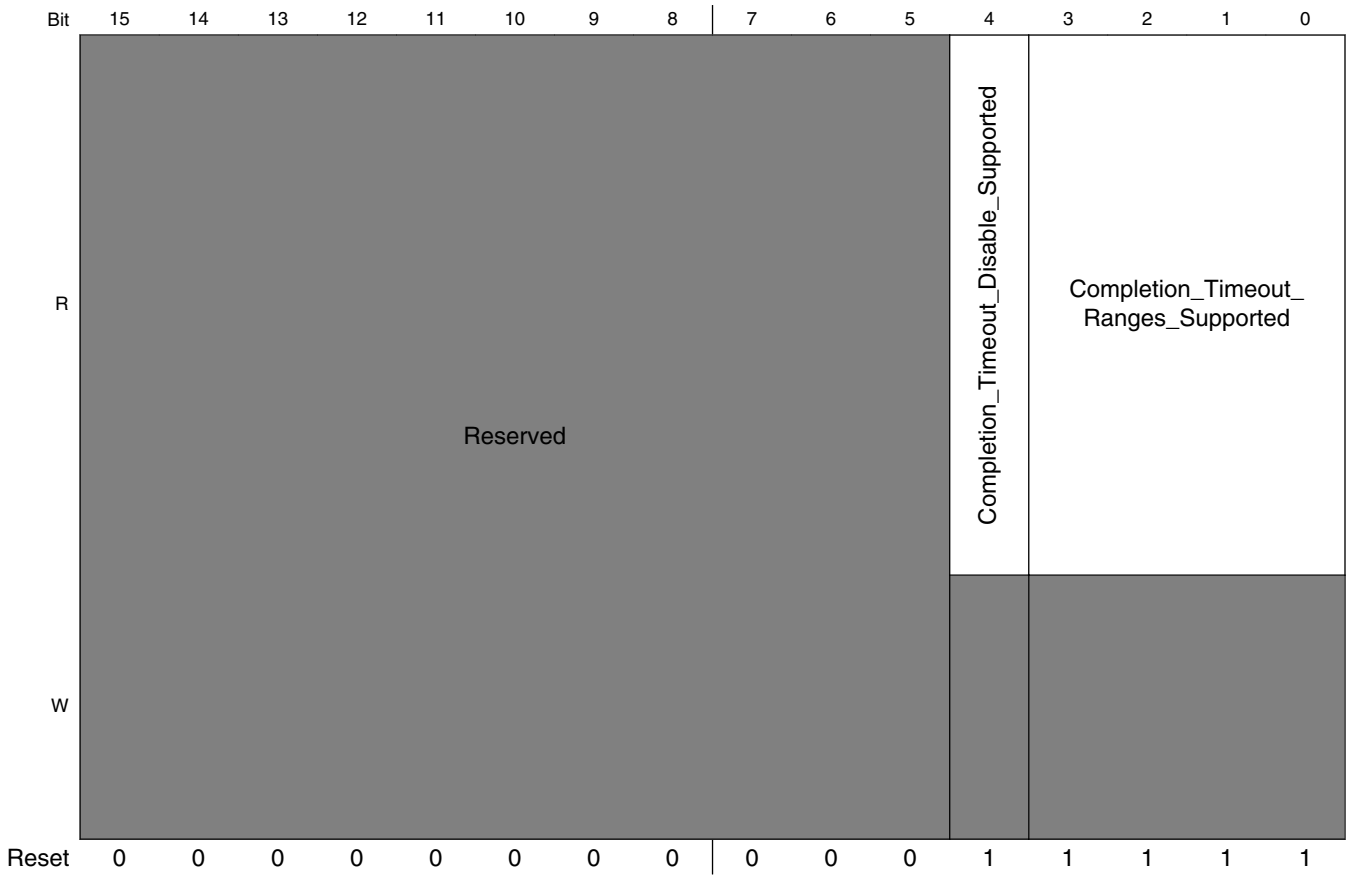
Field	Description
31–18 -	This field is reserved. Reserved
17 PME_Pending	PME Pending
16 PME_Status	PME Status
PME_Requester_ID	PME Requester ID

49.11.28 Device Capabilities 2 Register (PCIE_RC_DCR2)

Offset: `CFG_PCIE_CAP + 0x24

Address: 1FF_C000h base + 94h offset = 1FF_C094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



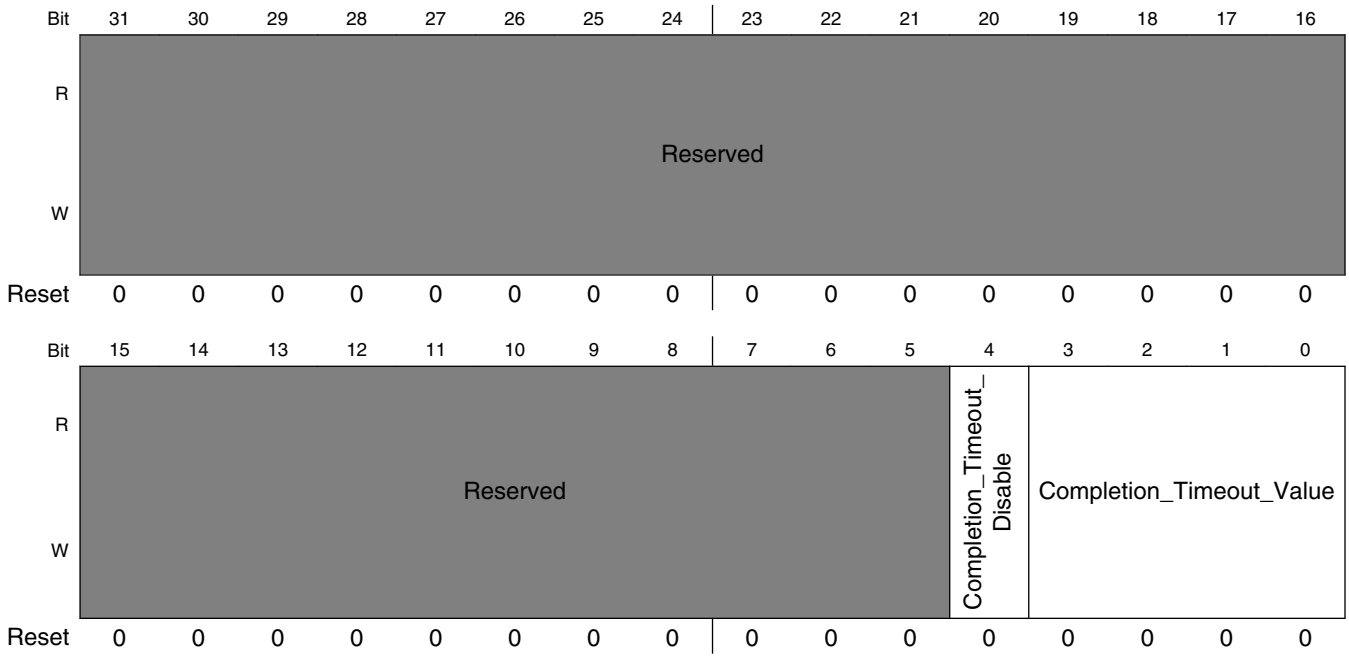
PCIE_RC_DCR2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 Completion_ Timeout_ Disable_ Supported	Completion Timeout Disable Supported
Completion_ Timeout_ Ranges_ Supported	Completion Timeout Ranges Supported This field is applicable only to Root Ports, Endpoints that issue Requests on their own behalf, and PCI Express to PCI/PCI-X Bridges that take ownership of Requests issued on PCI Express. the default value is 0xf (A, B, C and D ranges supported)

49.11.29 Device Control and Status 2 Register (PCIE_RC_DCSR2)

Offset: `CFG_PCIE_CAP + 0x28

Address: 1FF_C000h base + 98h offset = 1FF_C098h



PCIE_RC_DCSR2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 Completion_ Timeout_Disable	Completion Timeout Disable
Completion_ Timeout_Value	Completion Timeout Value If the default range is chosen, the core will have a timeout in the range of 16ms to 55ms. . following encodings apply: Values not defined below are reserved. 0000 Default range: 50 ìs to 50 ms 0001 50 ìs to 100 ìs 0010 1 ms to 10 ms 0101 16 ms to 55 ms 0110 65 ms to 210 ms

Table continues on the next page...

PCIE_RC_DCSR2 field descriptions (continued)

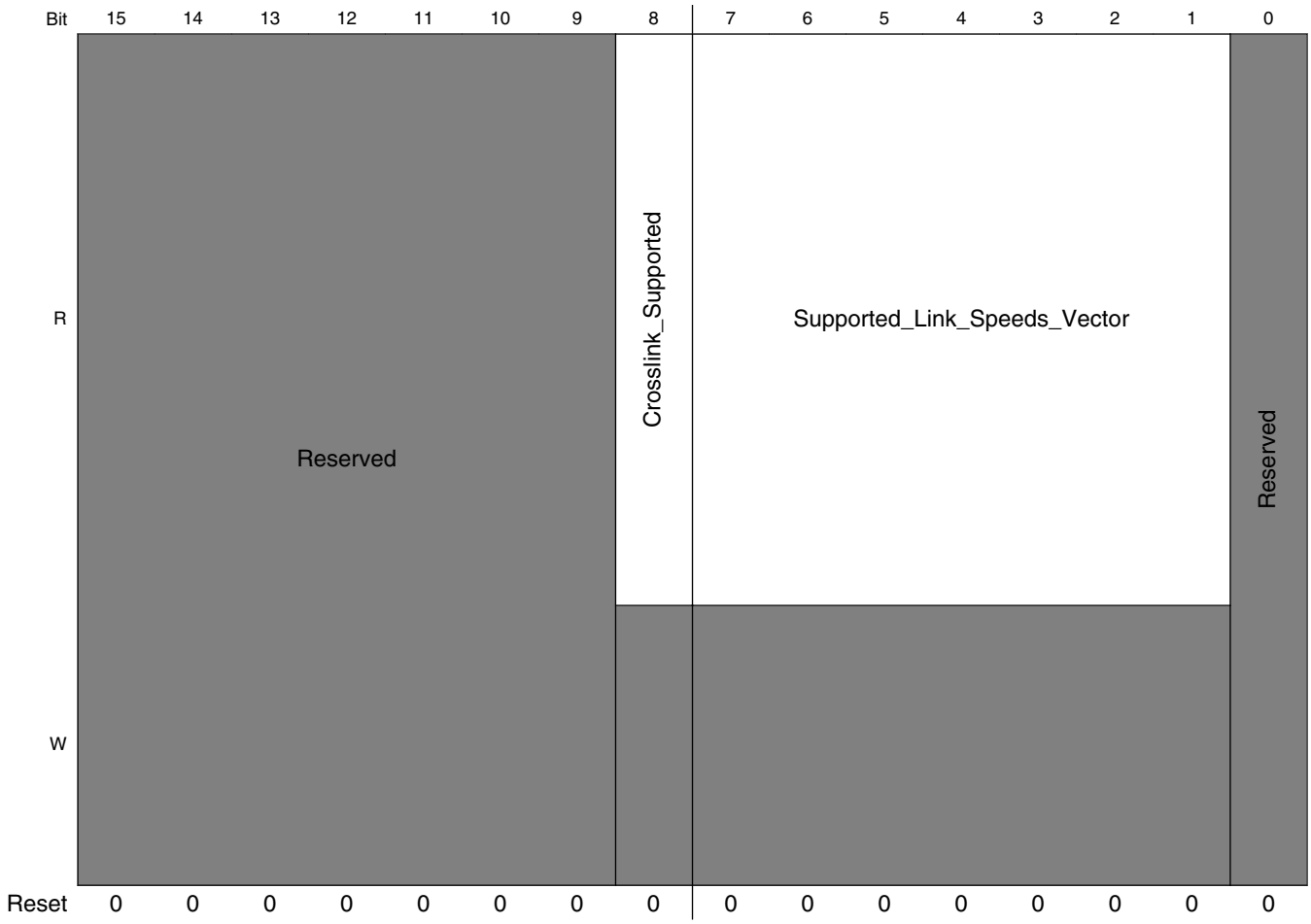
Field	Description
	1001 260 ms to 900 ms 1010 1 s to 3.5 s 1101 4 s to 13 s 1110 17 s to 64 s

49.11.30 Link Capabilities 2 Register (PCIE_RC_LCR2)

Offset: `CFG_PCIE_CAP + 0x2C

Address: 1FF_C000h base + 9Ch offset = 1FF_C09Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



PCIE_RC_LCR2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 Crosslink_ Supported	Crosslink Supported
7–1 Supported_Link_ Speeds_Vector	Supported Link Speeds Vector Indicates the supported Link speeds of the associated Port. For each bit, a value of 1b indicates that the corresponding Link speed is supported; otherwise, the Link speed is not supported. Bit definitions are: Bit 1 2.5 GT/s Bit 2 5.0 GT/s Reserved Bits 7:4 reserved This field is writable through the DBI.
0 -	This field is reserved. Reserved

49.11.31 Link Control and Status 2 Register (PCIE_RC_LCSR2)

Offset: `CFG_PCIE_CAP + 30

Address: 1FF_C000h base + A0h offset = 1FF_C0A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved										Link_Equalization_Request	Equalization_Phase_3_Successful	Equalization_Phase_2_Successful	Equalization_Phase_1_Successful	Equalization_Complete	Current_Deemphasis_Level
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Compliance_Pre_set_Deemphasis				Compliance_SOS	Enter_Modified_Compliance	Transmit_Margin			Selectable_Deemphasis	Hardware_Autonomous_Speed_Disable	Enter_Compliance	Target_Link_Speed			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_RC_LCSR2 field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21 Link_ Equalization_ Request	Link Equalization Request
20 Equalization_ Phase_3_ Successful	Equalization Phase 3 Successful
19 Equalization_ Phase_2_ Successful	Equalization Phase 2 Successful

Table continues on the next page...

PCIE_RC_LCSR2 field descriptions (continued)

Field	Description
18 Equalization_ Phase_1_ Successful	Equalization Phase 1 Successful
17 Equalization_ Complete	Equalization Complete
16 Current_ Deemphasis_ Level	Current De-emphasis Level
15–12 Compliance_ Pre_set_ Deemphasis	Compliance Pre-set/ De-emphasis
11 Compliance_ SOS	Compliance SOS When set to 1b, the LTSSM is required to send SKP Ordered Sets periodically in between the (modified) compliance patterns. GT/s speed are permitted to hardwire this bit to 0b. **NOTE:** When the Link is operating at 2.5 GT/s, the setting of this bit has no effect. Components that support only 2.5
10 Enter_Modified_ Compliance	Enter Modified Compliance When this bit is set to 1b, the device transmits modified compliance pattern if the LTSSM enters Polling. Compliance state.
9–7 Transmit_Margin	Transmit Margin This field is reset to 000b on entry to the LTSSM Polling. Compliance substate. Components that support only the 2.5 GT/s speed are permitted to hard-wire this bit to 0b. When operating in 5.0 GT/s mode with full swing, the de-emphasis ratio must be maintained within +/- 1 dB from the specification-defined operational value (either -3.5 or -6 dB). This field controls the value of the non-de-emphasized voltage level at the Transmitter pins: 000 800-1200 mV for full swing 400-600 mV for half- swing 001-010 values must be monotonic with a non-zero slope 011 200-400 mV for full-swing and 100-200 mV for halfswing 100-111 reserved
6 Selectable_ Deemphasis	Selectable De-emphasis When the Link is operating at 2.5 GT/s speed, the setting of this bit has no effect. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. Default value is implementation-specific, unless a specific value is required for a selected form factor or platform. When the Link is operating at 5.0 GT/s speed, selects the level of de-emphasis: 1 -3.5 dB 0 -6 dB

Table continues on the next page...

PCIE_RC_LCSR2 field descriptions (continued)

Field	Description
5 Hardware_ Autonomous_ Speed_Disable	Hardware Autonomous Speed Disable When cfg_hw_auto_sp_dis signal is asserted, the application must disable hardware from changing the Link speed for device-specific reasons other than attempting to correct unreliable Link operation by reducing Link speed. Initial transition to the highest supported common link speed is not blocked by this signal.
4 Enter_ Compliance	Enter Compliance Software is permitted to force a link to enter Compliance mode at the speed indicated in the Target Link Speed field by setting this bit to 1b in both components on a link and then initiating a hot reset on the link. The default value of this field following Fundamental Reset is 0b.
Target_Link_ Speed	Target Link Speed For Downstream ports, this field sets an upper limit on link operational speed by restricting the values advertised by the upstream component in its training sequences: The encoding is the binary value of the bit in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the desired target Link speed. **NOTE:** If a value is written to this field that does not correspond to a speed included in the Supported Link Speeds field, the result is undefined. **NOTE:** The default value of this field is the highest link speed supported by the component (as reported in the Max Link Speed field of the Link Capabilities Register) unless the corresponding platform / form factor requires a different default value. **NOTE:** Components that support only the 2.5 GT/s speed are permitted to hardwire this field to 0000b. All other encodings are reserved. 0000001 Gen1 2.5 GT/s 0000010 Gen2 5.0 GT/s

49.11.32 AER Capability Header (PCIE_RC_AER)

The core implements the following PCI Express Extended Capabilities registers:

- Advanced Error Reporting Capability register set
- Virtual Channel Capability register set

Address: 0x100

Address: 1FF_C000h base + 100h offset = 1FF_C100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Next_Capability_Offset												Capability_ Version				PCI_Express_Extended_Capability_ID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

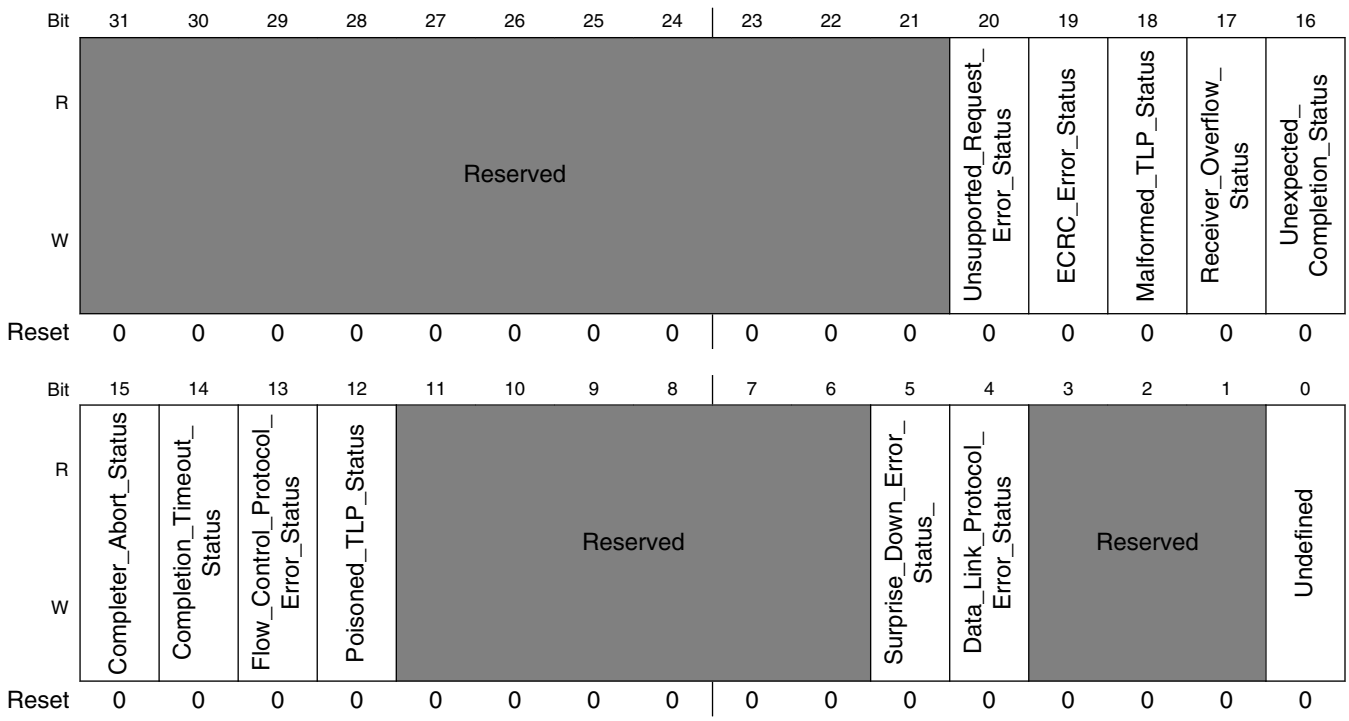
PCIE_RC_AER field descriptions

Field	Description
31–20 Next_Capability_Offset	Next Capability Offset
19–16 Capability_Version	Capability Version
PCI_Express_Extended_Capability_ID	PCI Express Extended Capability ID Value is 0x1 for Advanced Error Reporting.

49.11.33 Uncorrectable Error Status Register (PCIE_RC_UESR)

Offset: 0x04

Address: 1FF_C000h base + 104h offset = 1FF_C104h



PCIE_RC_UESR field descriptions

Field	Description
31–21 -	This field is reserved. Reserved

Table continues on the next page...

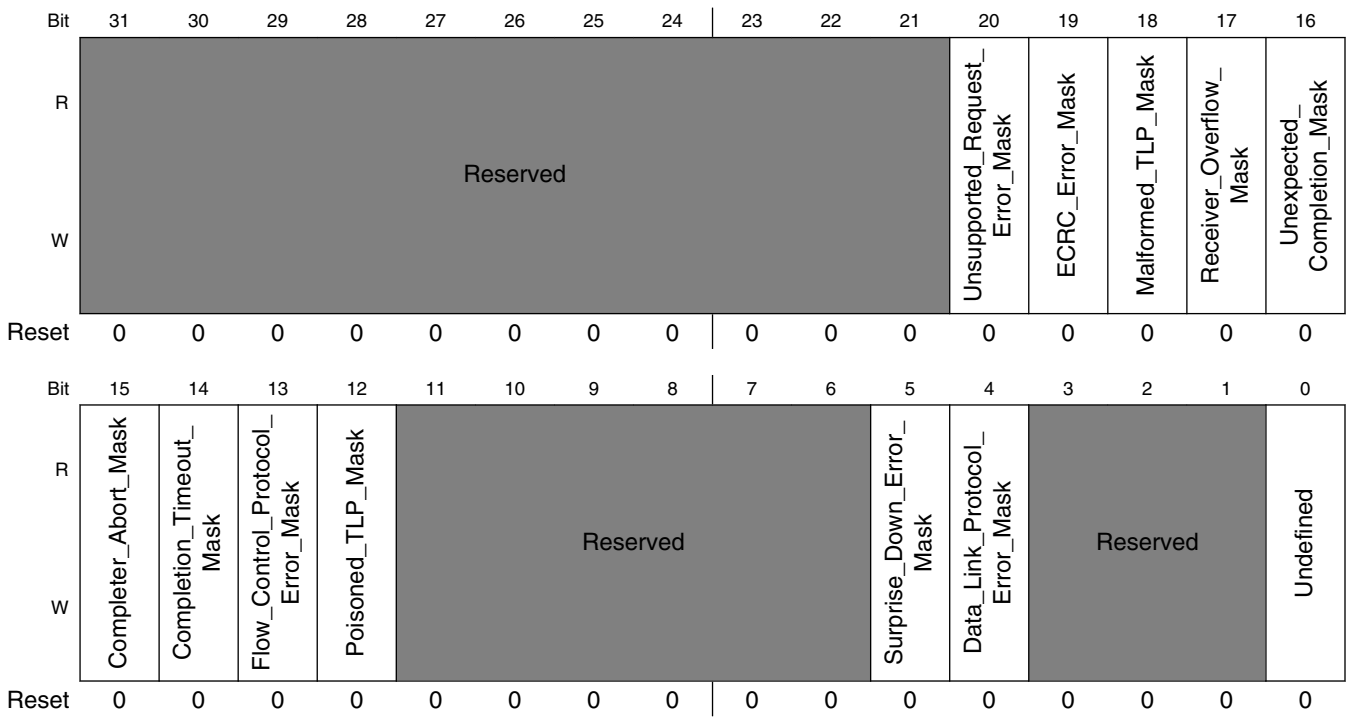
PCIE_RC_UESR field descriptions (continued)

Field	Description
20 Unsupported_ Request_Error_ Status	Unsupported Request Error Status
19 ECRC_Error_ Status	ECRC Error Status
18 Malformed_TLP_ Status	Malformed TLP Status
17 Receiver_ Overflow_Status	Receiver Overflow Status
16 Unexpected_ Completion_ Status	Unexpected Completion Status
15 Completer_ Abort_Status	Completer Abort Status
14 Completion_ Timeout_Status	Completion Timeout Status
13 Flow_Control_ Protocol_Error_ Status	Flow Control Protocol Error Status
12 Poisoned_TLP_ Status	Poisoned TLP Status
11–6 -	This field is reserved. Reserved
5 Surprise_Down_ Error_Status_	Surprise Down Error Status (not supported)
4 Data_Link_ Protocol_Error_ Status	Data Link Protocol Error Status
3–1 -	This field is reserved. Reserved
0 Undefined	Undefined for PCI Express 1.1 (Was Training Error Status for PCI Express 1.0a)

49.11.34 Uncorrectable Error Mask Register (PCIE_RC_UEMR)

Offset: 0x08

Address: 1FF_C000h base + 108h offset = 1FF_C108h



PCIE_RC_UEMR field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 Unsupported_Request_Error_Mask	Unsupported Request Error Mask
19 ECRC_Error_Mask	ECRC Error Mask
18 Malformed_TLP_Mask	Malformed TLP Mask
17 Receiver_Overflow_Mask	Receiver Overflow Mask

Table continues on the next page...

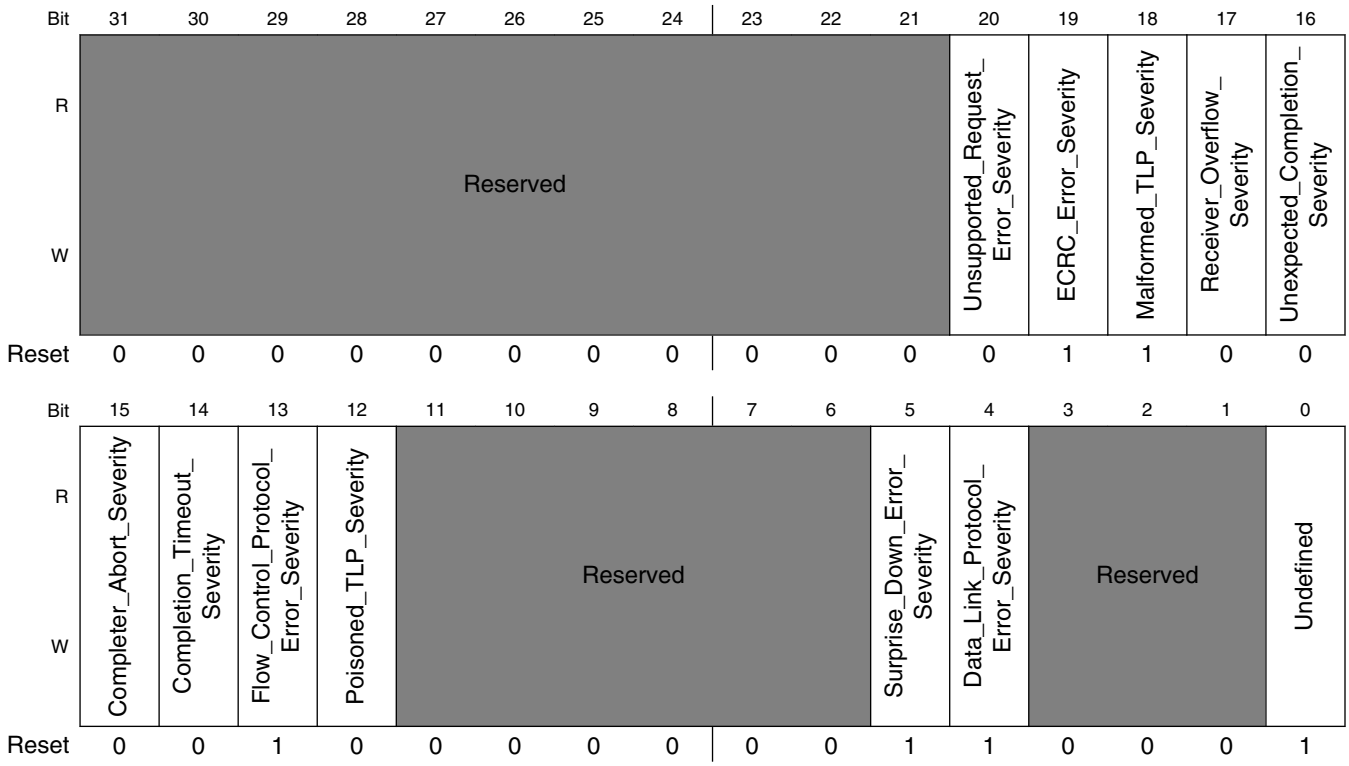
PCIE_RC_UEMR field descriptions (continued)

Field	Description
16 Unexpected_ Completion_ Mask	Unexpected Completion Mask
15 Completer_ Abort_Mask	Completer Abort Mask
14 Completion_ Timeout_Mask	Completion Timeout Mask
13 Flow_Control_ Protocol_Error_ Mask	Flow Control Protocol Error Mask
12 Poisoned_TLP_ Mask	Poisoned TLP Mask
11–6 -	This field is reserved. Reserved
5 Surprise_Down_ Error_Mask	Surprise Down Error Mask (not supported)
4 Data_Link_ Protocol_Error_ Mask	Data Link Protocol Error Mask
3–1 -	This field is reserved. Reserved
0 Undefined	Undefined for PCI Express 1.1 (Was Training Error Mask for PCI Express 1.0a)

49.11.35 Uncorrectable Error Severity Register (PCIE_RC_UESevR)

Offset: 0x0C

Address: 1FF_C000h base + 10Ch offset = 1FF_C10Ch



PCIE_RC_UESevR field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 Unsupported_Request_Error_Severity	Unsupported Request Error Severity
19 ECRC_Error_Severity	ECRC Error Severity
18 Malformed_TLP_Severity	Malformed TLP Severity

Table continues on the next page...

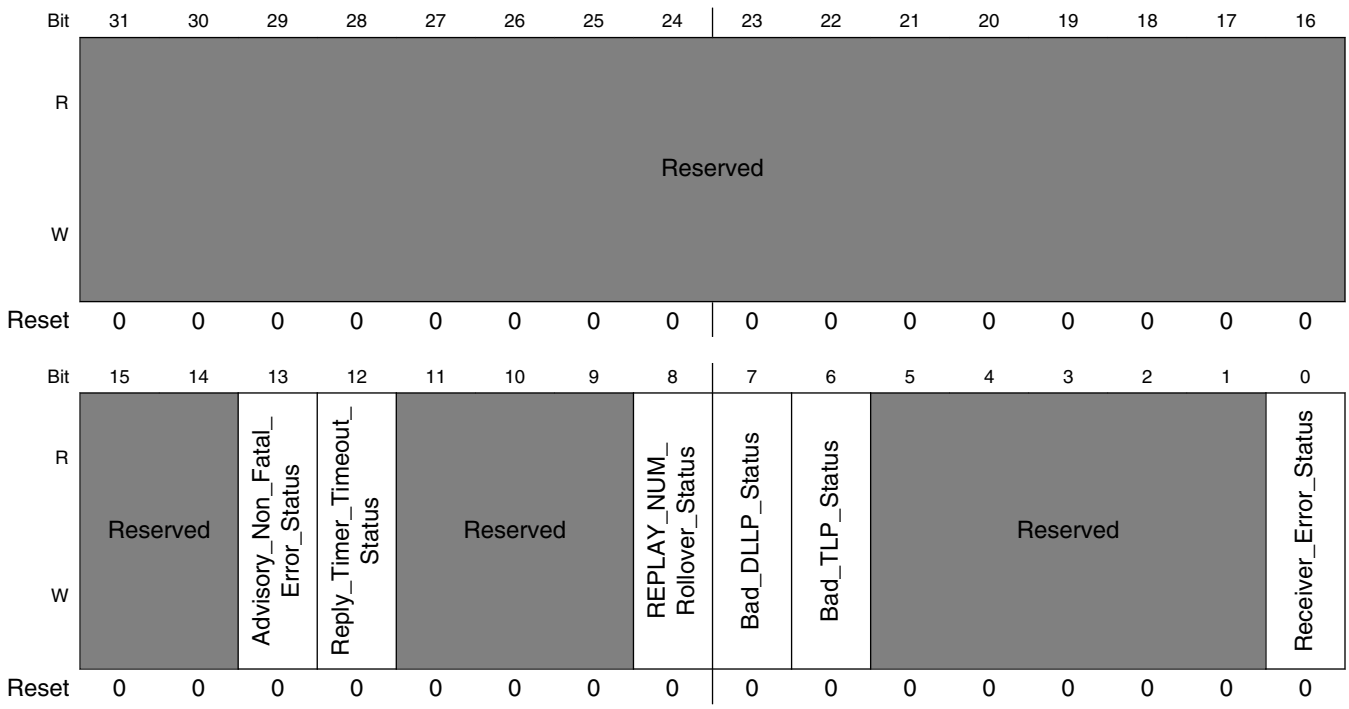
PCIE_RC_UESevR field descriptions (continued)

Field	Description
17 Receiver_ Overflow_ Severity	Receiver Overflow Severity
16 Unexpected_ Completion_ Severity	Unexpected Completion Severity
15 Completer_ Abort_Severity	Completer Abort Severity
14 Completion_ Timeout_Severity	Completion Timeout Severity
13 Flow_Control_ Protocol_Error_ Severity	Flow Control Protocol Error Severity
12 Poisoned_TLP_ Severity	Poisoned TLP Severity
11–6 -	This field is reserved. Reserved
5 Surprise_Down_ Error_Severity	Surprise Down Error Severity (not supported)
4 Data_Link_ Protocol_Error_ Severity	Data Link Protocol Error Severity
3–1 -	This field is reserved. Reserved
0 Undefined	Undefined for PCI Express 1.1 (Was Training Error Severity for PCI Express 1.0a)

49.11.36 Correctable Error Status Register (PCIE_RC_CESR)

Offset: 0x10

Address: 1FF_C000h base + 110h offset = 1FF_C110h



PCIE_RC_CESR field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13 Advisory_Non_Fatal_Error_Status	Advisory Non-Fatal Error Status
12 Reply_Timer_Timeout_Status	Reply Timer Timeout Status
11–9 -	This field is reserved. Reserved
8 REPLAY_NUM_Rollover_Status	REPLAY_NUM Rollover Status
7 Bad_DLLP_Status	Bad DLLP Status

Table continues on the next page...

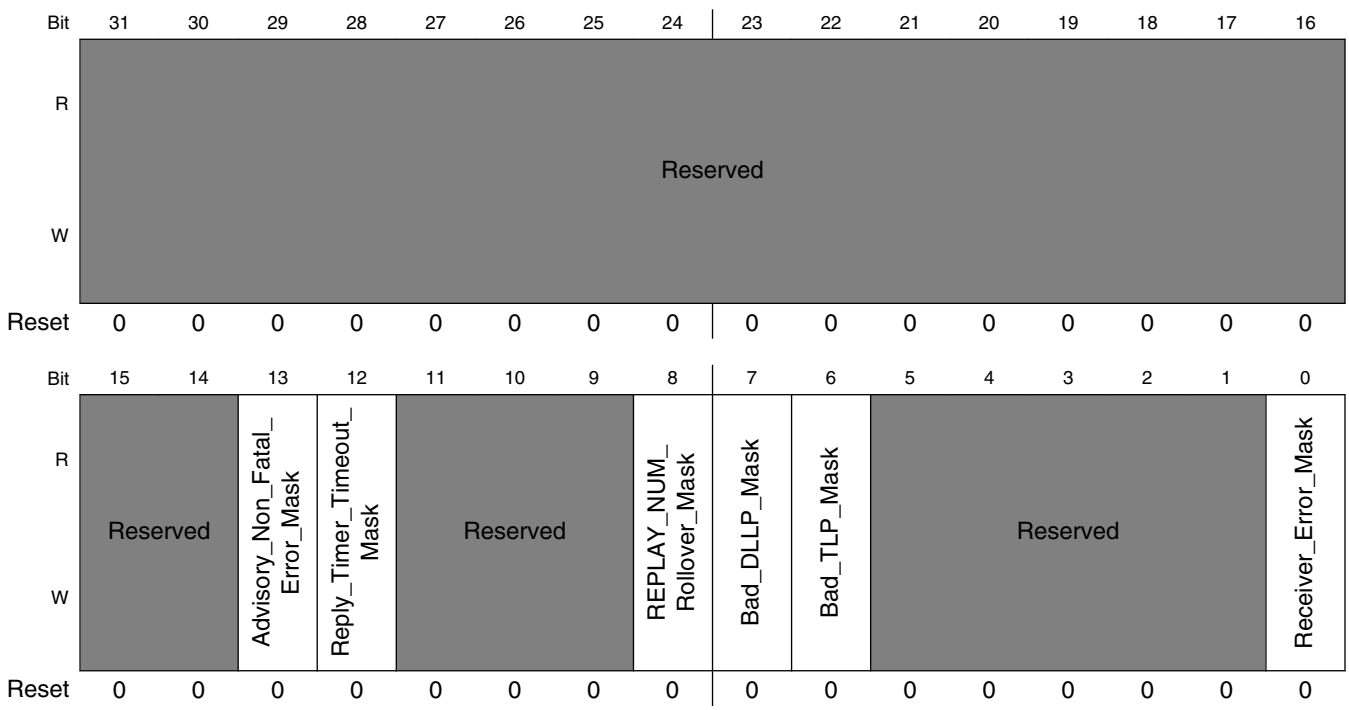
PCIE_RC_CESR field descriptions (continued)

Field	Description
6 Bad_TLP_Status	Bad TLP Status
5–1 -	This field is reserved. Reserved
0 Receiver_Error_Status	Receiver Error Status

49.11.37 Correctable Error Mask Register (PCIE_RC_CEMR)

Offset: 0x14

Address: 1FF_C000h base + 114h offset = 1FF_C114h



PCIE_RC_CEMR field descriptions

Field	Description
31–14 -	This field is reserved. Reserved

Table continues on the next page...

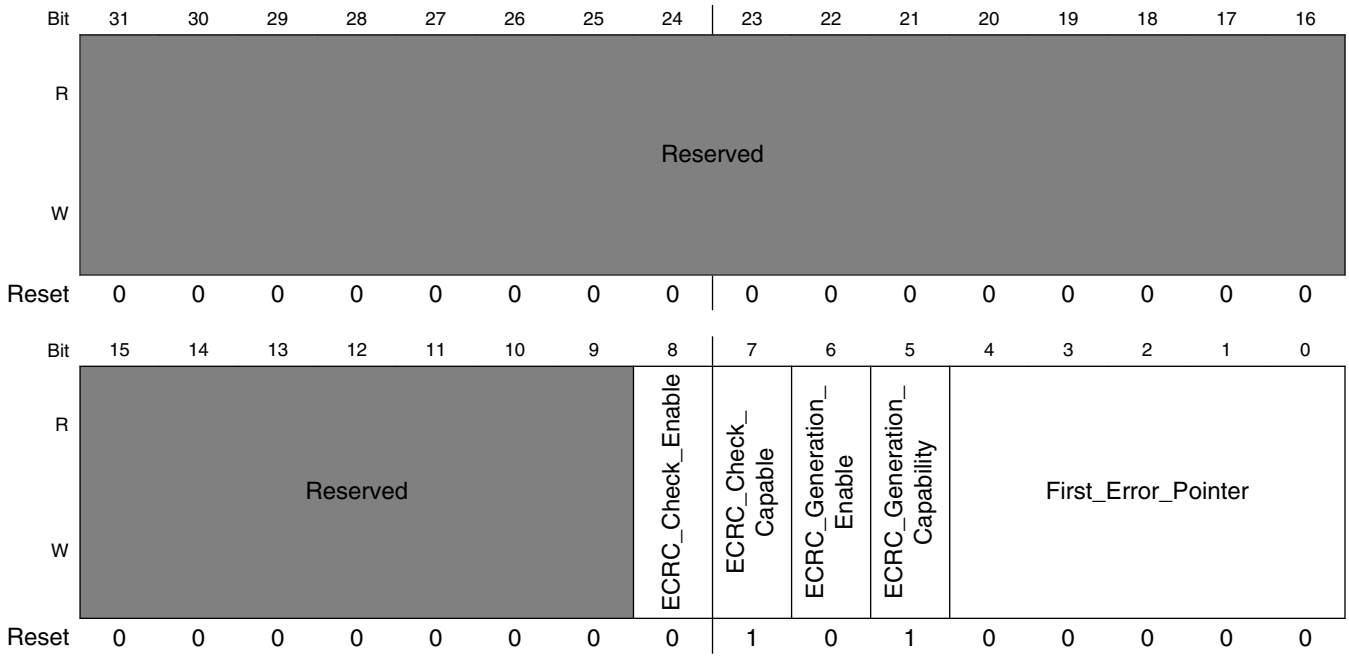
PCIE_RC_CEMR field descriptions (continued)

Field	Description
13 Advisory_Non_Fatal_Error_Mask	Advisory Non-Fatal Error Mask
12 Reply_Timer_Timeout_Mask	Reply Timer Timeout Mask
11–9 -	This field is reserved. Reserved
8 REPLAY_NUM_Rollover_Mask	REPLAY_NUM Rollover Mask
7 Bad_DLLP_Mask	Bad DLLP Mask
6 Bad_TLP_Mask	Bad TLP Mask
5–1 -	This field is reserved. Reserved
0 Receiver_Error_Mask	Receiver Error Mask

49.11.38 Advanced Capabilities and Control Register (PCIE_RC_ACCR)

Offset: 0x18

Address: 1FF_C000h base + 118h offset = 1FF_C118h



PCIE_RC_ACCR field descriptions

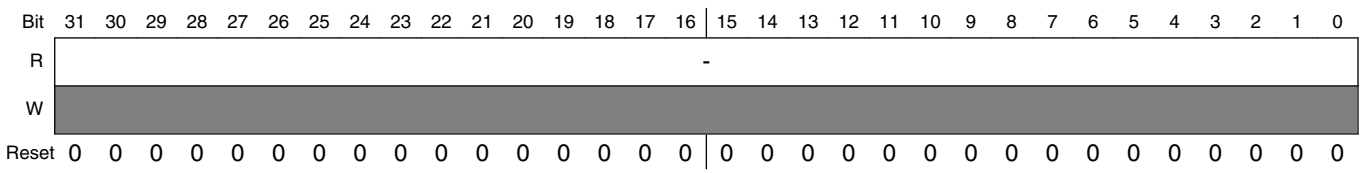
Field	Description
31–9 -	This field is reserved. Reserved
8 ECRC_Check_ Enable	ECRC Check Enable
7 ECRC_Check_ Capable	ECRC Check Capable
6 ECRC_ Generation_ Enable	ECRC Generation Enable
5 ECRC_ Generation_ Capability	ECRC Generation Capability
First_Error_ Pointer	First Error Pointer

49.11.39 Header Log Register (PCIE_RC_HLR)

Offset: 0x1C

The Header Log registers collect the header for the TLP corresponding to a detected error. See the PCI Express 3.0 Specification for details. Each of the Header Log registers is type ROS; the default reset value of each Header Log register is 0x00000000.

Address: 1FF_C000h base + 11Ch offset = 1FF_C11Ch



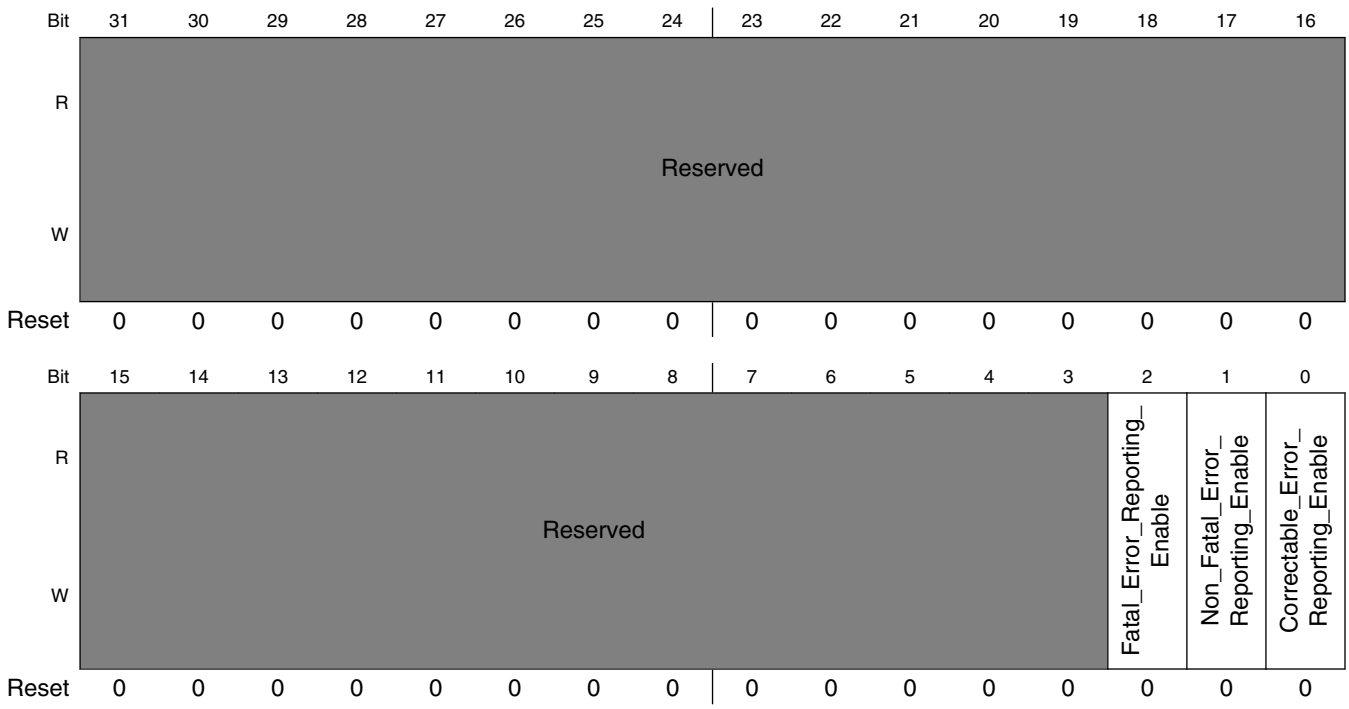
PCIE_RC_HLR field descriptions

Field	Description
-	Header Log Register (nth DWORD)

49.11.40 Root Error Command Register (PCIE_RC_RECR)

Offset: 0x100 + 0x2C

Address: 1FF_C000h base + 12Ch offset = 1FF_C12Ch



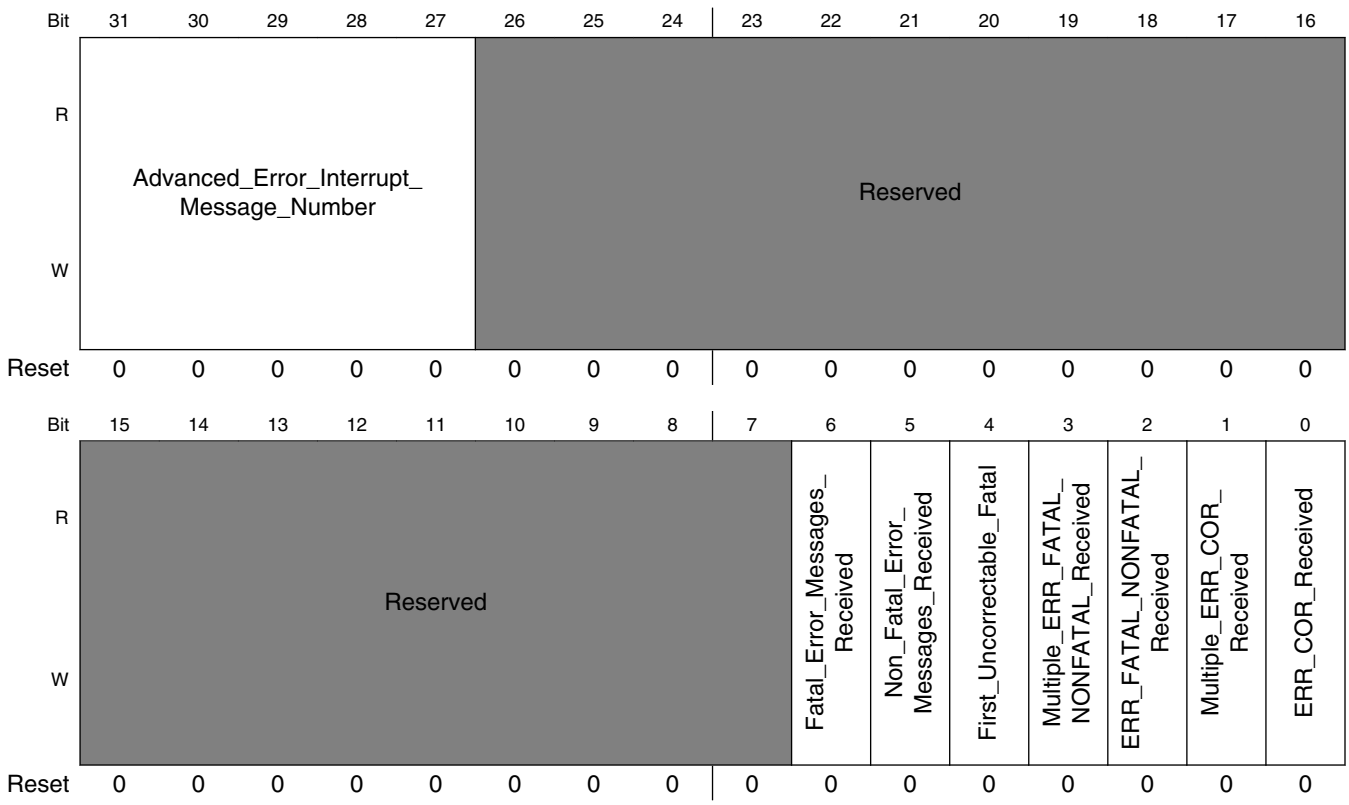
PCIE_RC_RECR field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
2 Fatal_Error_ Reporting_ Enable	Fatal Error Reporting Enable
1 Non_Fatal_ Error_Reporting_ Enable	Non-Fatal Error Reporting Enable
0 Correctable_ Error_Reporting_ Enable	Correctable Error Reporting Enable

49.11.41 Root Error Status Register (PCIE_RC_RESR)

Offset: 0x100 + 0x30

Address: 1FF_C000h base + 130h offset = 1FF_C130h



PCIE_RC_RESR field descriptions

Field	Description
31–27 Advanced_Error_ Interrupt_ Message_ Number	Advanced Error Interrupt Message Number, writable through the DBI
26–7 -	This field is reserved. Reserved
6 Fatal_Error_ Messages_ Received	Fatal Error Messages Received
5 Non_Fatal_Error_ Messages_ Received	Non-Fatal Error Messages Received

Table continues on the next page...

PCIE_RC_RESR field descriptions (continued)

Field	Description
4 First_ Uncorrectable_ Fatal	First Uncorrectable Fatal
3 Multiple_ERR_ FATAL_ NONFATAL_ Received	Multiple ERR_FATAL/NONFATAL Received
2 ERR_FATAL_ NONFATAL_ Received	ERR_FATAL/NONFATAL Received
1 Multiple_ERR_ COR_Received	Multiple ERR_COR Received
0 ERR_COR_ Received	ERR_COR Received

49.11.42 Error Source Identification Register (PCIE_RC_ESIR)

Offset: 0x100 + 0x34

Address: 1FF_C000h base + 134h offset = 1FF_C134h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_FATAL_NONFATAL_SID																ERR_COR_SID															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_RC_ESIR field descriptions

Field	Description
31–16 ERR_FATAL_ NONFATAL_SID	ERR_FATAL/NONFATAL Source Identification
ERR_COR_SID	ERR_COR Source Identification

49.11.43 VC Extended Capability Header (PCIE_RC_VCECHR)

Offset: 0x140

Address: 1FF_C000h base + 140h offset = 1FF_C140h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Next_Capability_Offset												Capability_ Version				Extended_Capability															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

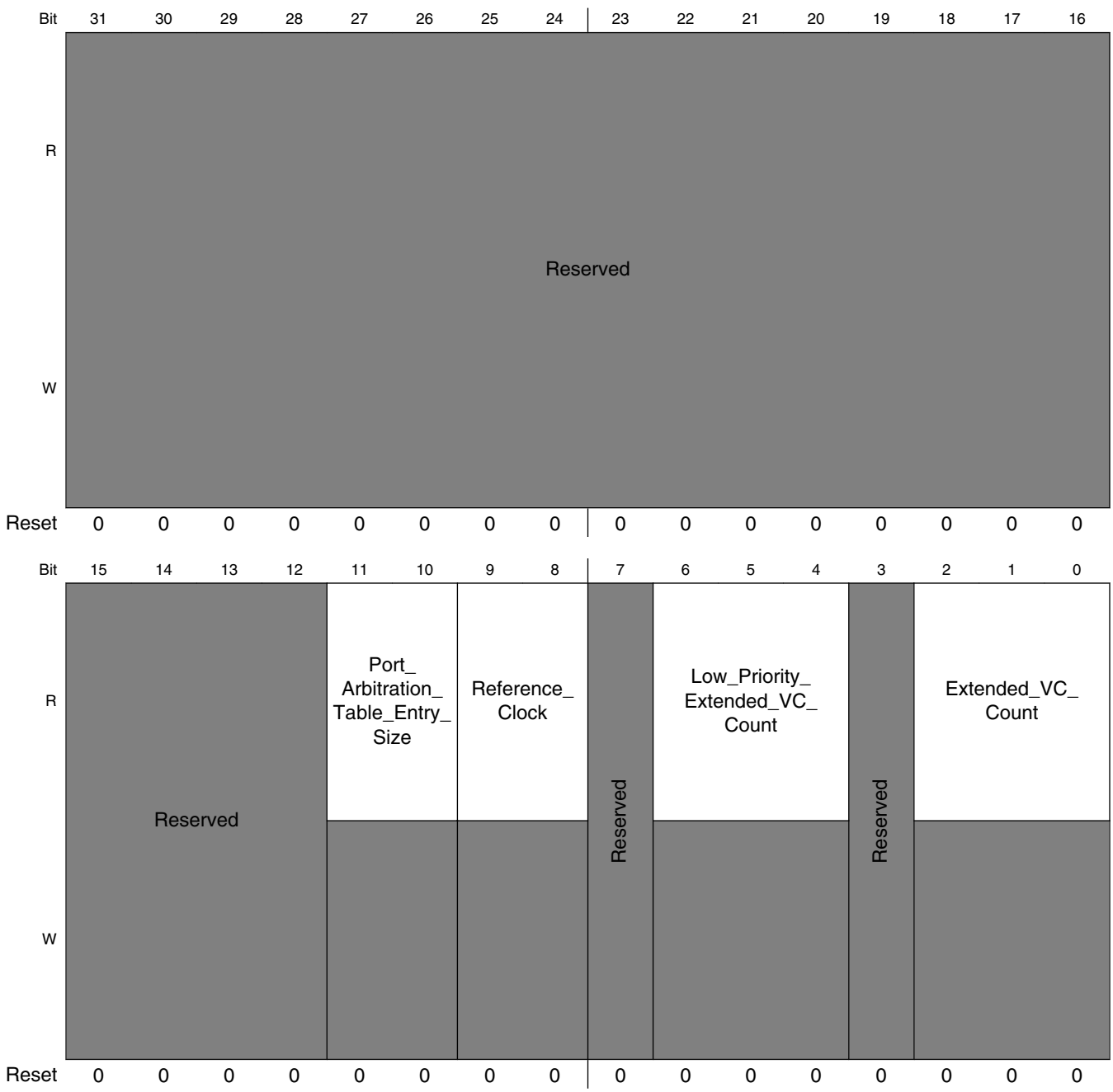
PCIE_RC_VCECHR field descriptions

Field	Description
31–20 Next_Capability_ Offset	Next Capability Offset
19–16 Capability_ Version	Capability Version
Extended_ Capability	PCI Express Extended Capability The default value is 0x2 for VC Capability.

49.11.44 Port VC Capability Register 1 (PCIE_RC_PVCCR1)

Offset: 0x140 + 0x4

Address: 1FF_C000h base + 144h offset = 1FF_C144h



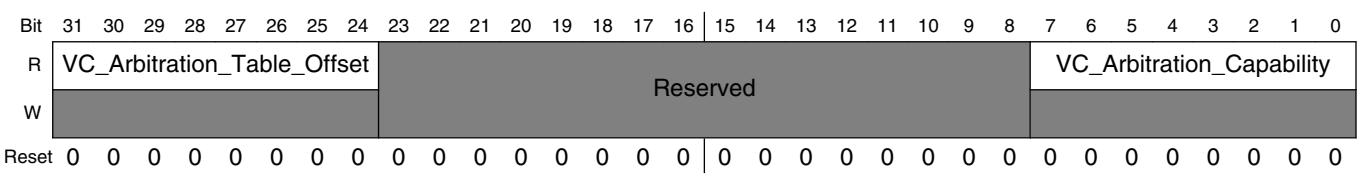
PCIE_RC_PVCCR1 field descriptions

Field	Description
31–12 -	This field is reserved. Reserved
11–10 Port_Arbitration_Table_Entry_Size	Port Arbitration Table Entry Size
9–8 Reference_Clock	Reference Clock
7 -	This field is reserved. Reserved
6–4 Low_Priority_Extended_VC_Count	Low Priority Extended VC Count, writable through the DBI
3 -	This field is reserved. Reserved
Extended_VC_Count	Extended VC Count The default value is the one less than the number of VCs that

49.11.45 Port VC Capability Register 2 (PCIE_RC_PVCCR2)

Offset: 0x140 + 0x8

Address: 1FF_C000h base + 148h offset = 1FF_C148h



PCIE_RC_PVCCR2 field descriptions

Field	Description
31–24 VC_Arbitration_Table_Offset	VC Arbitration Table Offset (not supported) The default value is 0x00 (no arbitration table present).
23–8 -	This field is reserved. Reserved
VC_Arbitration_Capability	VC Arbitration Capability Indicates which VC arbitration mode(s) the device supports, writable through the DBI: •Bit 0: Device supports hardware fixed arbitration scheme. For the core, the scheme is 16-phase weighted round robin (WRR). •Bit 1: Device supports 32-phase WRR

Table continues on the next page...

PCIE_RC_PVCCR2 field descriptions (continued)

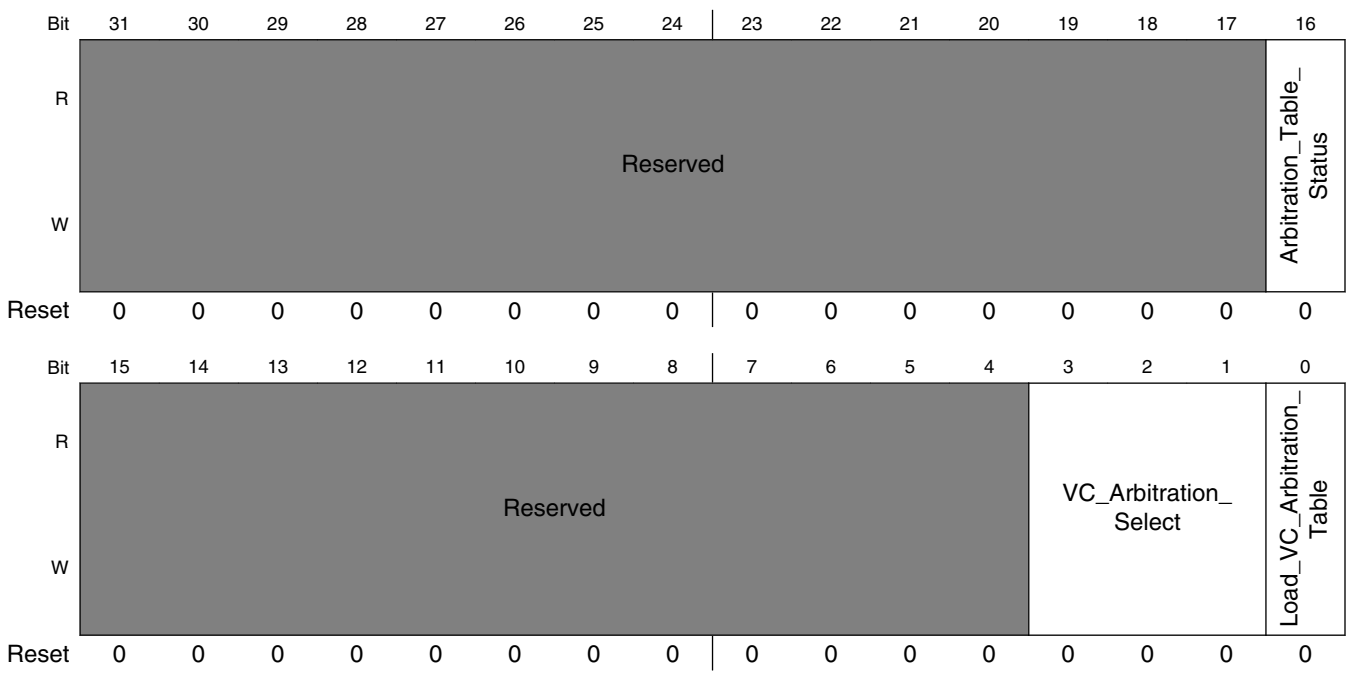
Field	Description
	•Bit 2: Device supports 64-phase WRR •Bit 3: Device supports 128-phase WRR •Bits 4-7: Reserved

49.11.46 Port VC Control and Status Register (PCIE_RC_PVCCSR)

Offset: 0x140 + 0xC

Bytes: 0-1

Address: 1FF_C000h base + 14Ch offset = 1FF_C14Ch



PCIE_RC_PVCCSR field descriptions

Field	Description
31–17 -	This field is reserved. Reserved
16 Arbitration_Table_Status	Arbitration Table Status
15–4 -	This field is reserved. Reserved

Table continues on the next page...

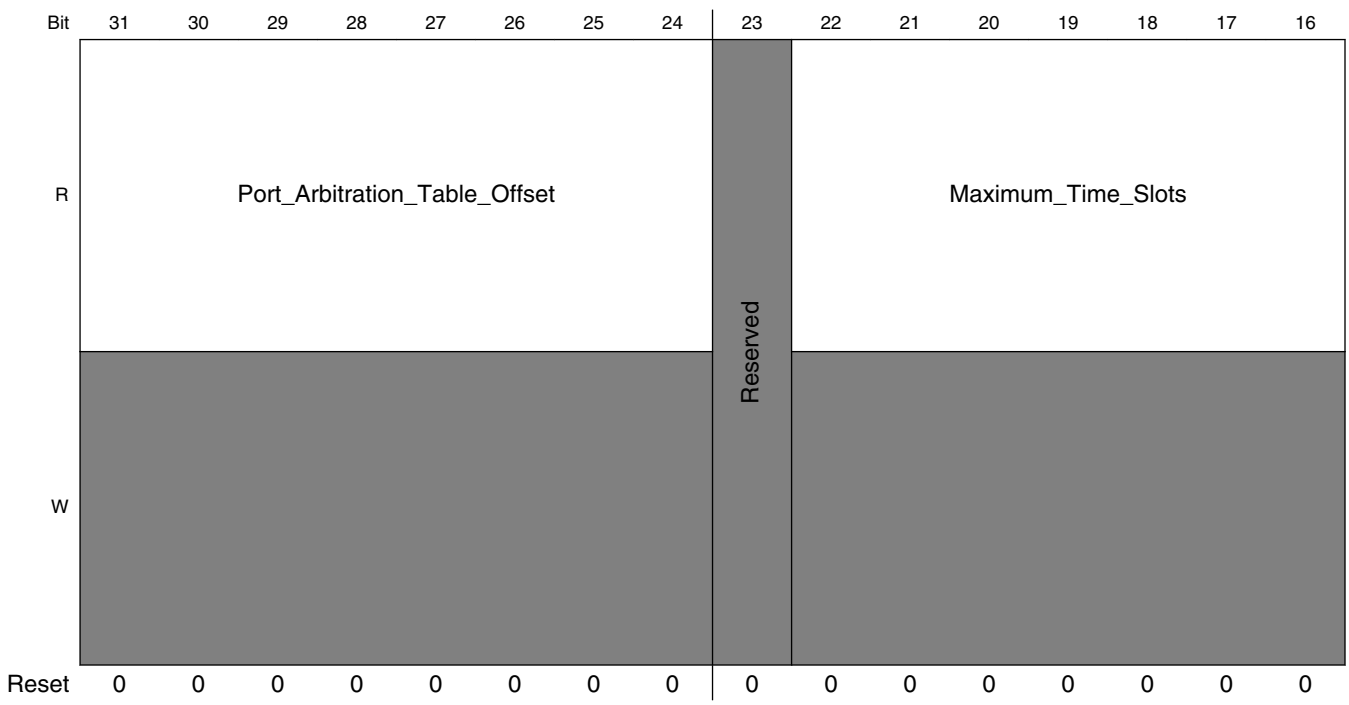
PCIE_RC_PVCCSR field descriptions (continued)

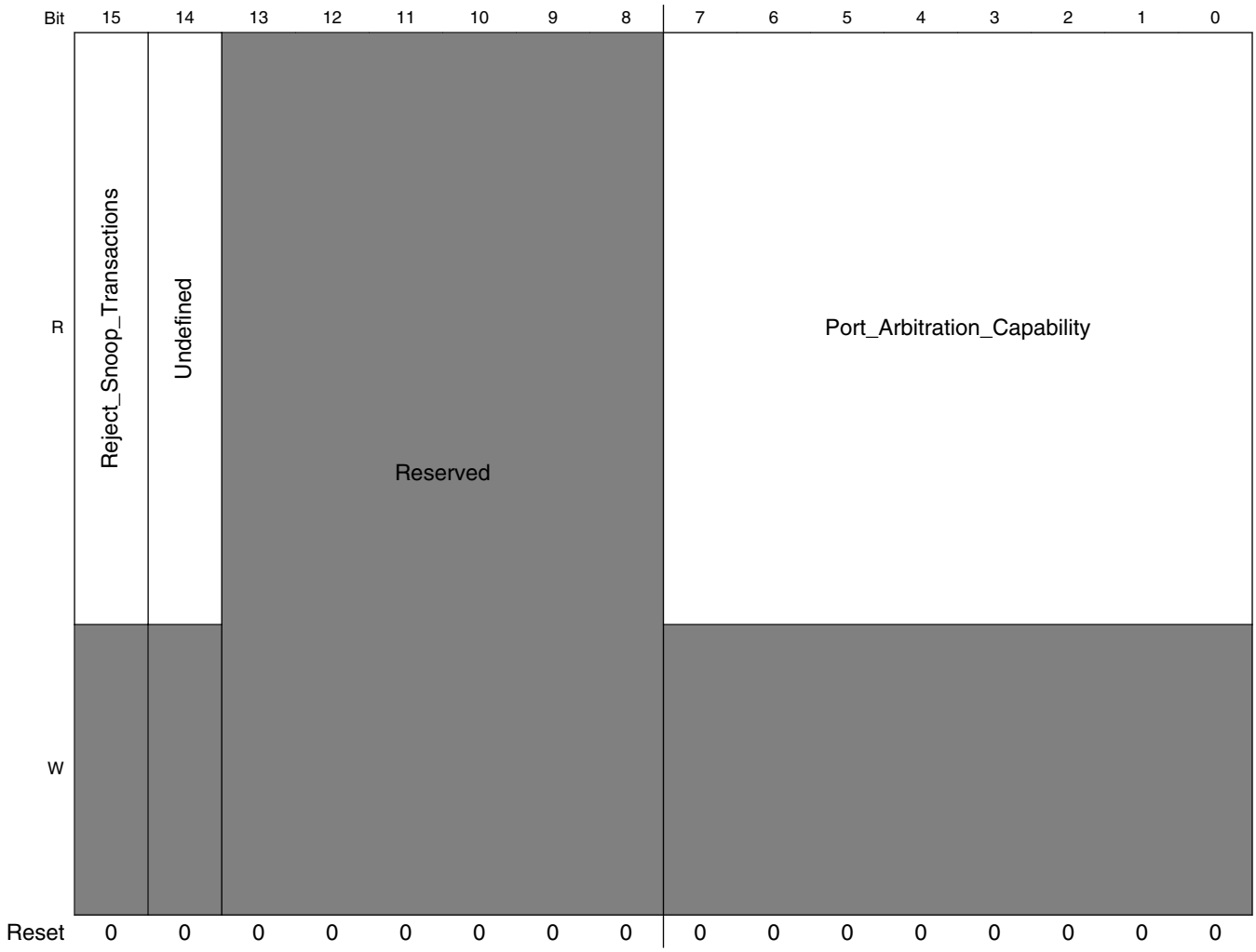
Field	Description
3–1 VC_Arbitration_ Select	VC Arbitration Select
0 Load_VC_ Arbitration_Table	Load VC Arbitration Table

49.11.47 VC Resource Capability Register n (PCIE_RC_VCRCR)

Offset: 0x140 + 0x10

Address: 1FF_C000h base + 150h offset = 1FF_C150h





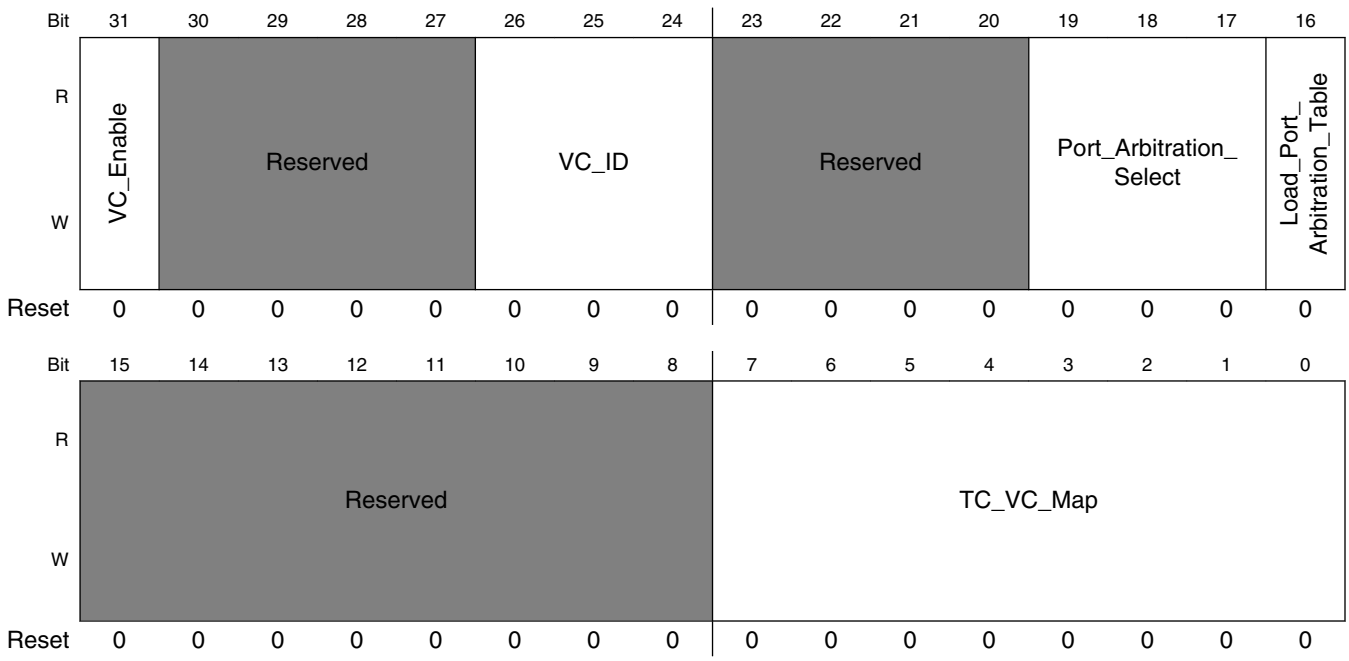
PCIE_RC_VCRCR field descriptions

Field	Description
31–24 Port_Arbitration_Table_Offset	Port Arbitration Table Offset
23 -	This field is reserved. Reserved
22–16 Maximum_Time_Slots	Maximum Time Slots
15 Reject_Snoop_Transactions	Reject Snoop Transactions
14 Undefined	Undefined for PCI Express 1.1 (Was Advanced Packet Switching for PCI Express 1.0a)
13–8 -	This field is reserved. Reserved
Port_Arbitration_Capability	Port Arbitration Capability

49.11.48 VC Resource Control Register n (PCIE_RC_VCRConR)

Offset: 0x140 + 0x14

Address: 1FF_C000h base + 154h offset = 1FF_C154h



PCIE_RC_VCRConR field descriptions

Field	Description
31 VC_Enable	VC Enable Hardwired to 1 for the first VC.
30–27 -	This field is reserved. Reserved
26–24 VC_ID	VC ID Hardwired to 0 for VC0.
23–20 -	This field is reserved. Reserved
19–17 Port_Arbitration_ Select	Port Arbitration Select
16 Load_Port_ Arbitration_Table	Load Port Arbitration Table
15–8 -	This field is reserved. Reserved
TC_VC_Map	TC/VC Map

Table continues on the next page...

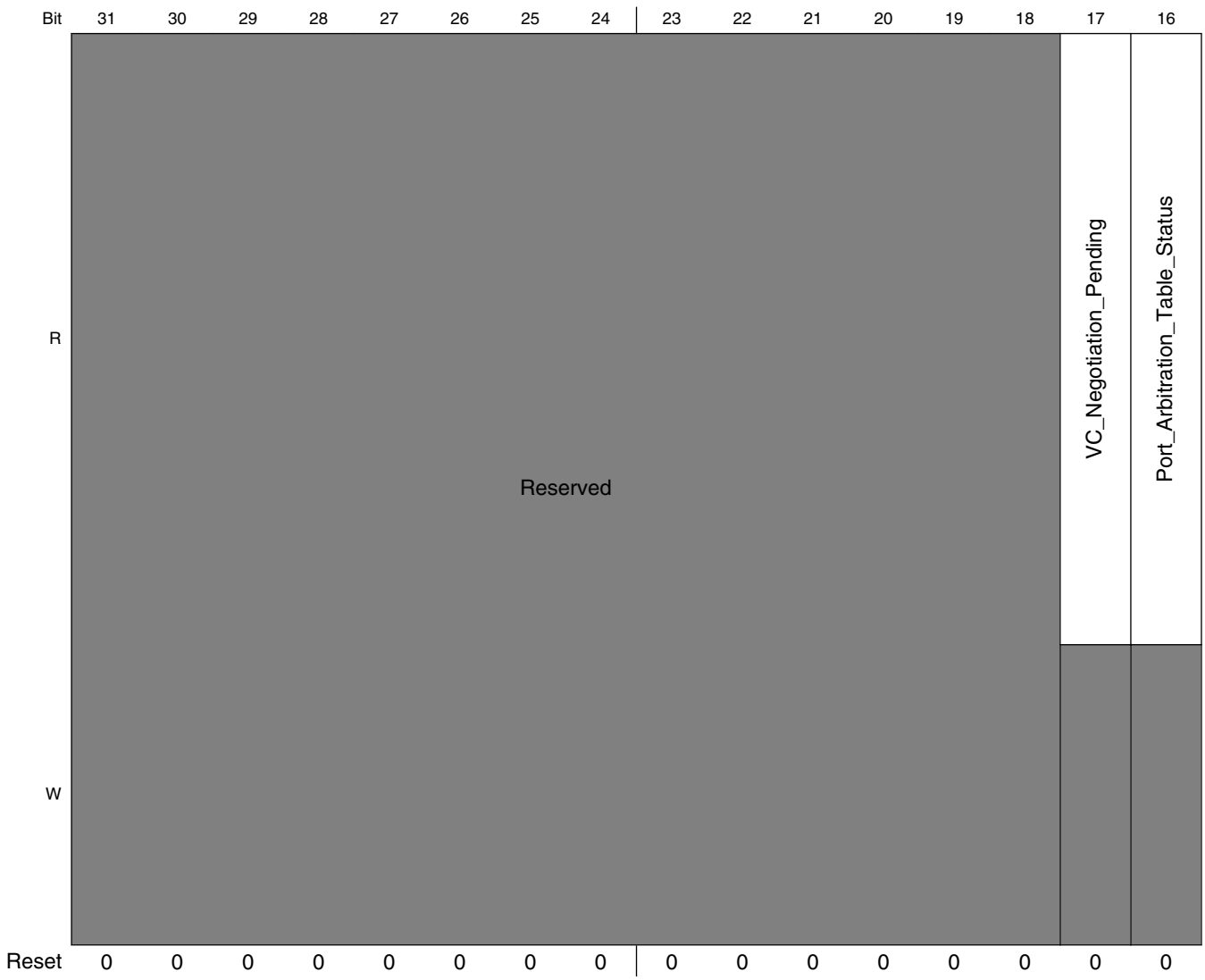
PCIE_RC_VCRConR field descriptions (continued)

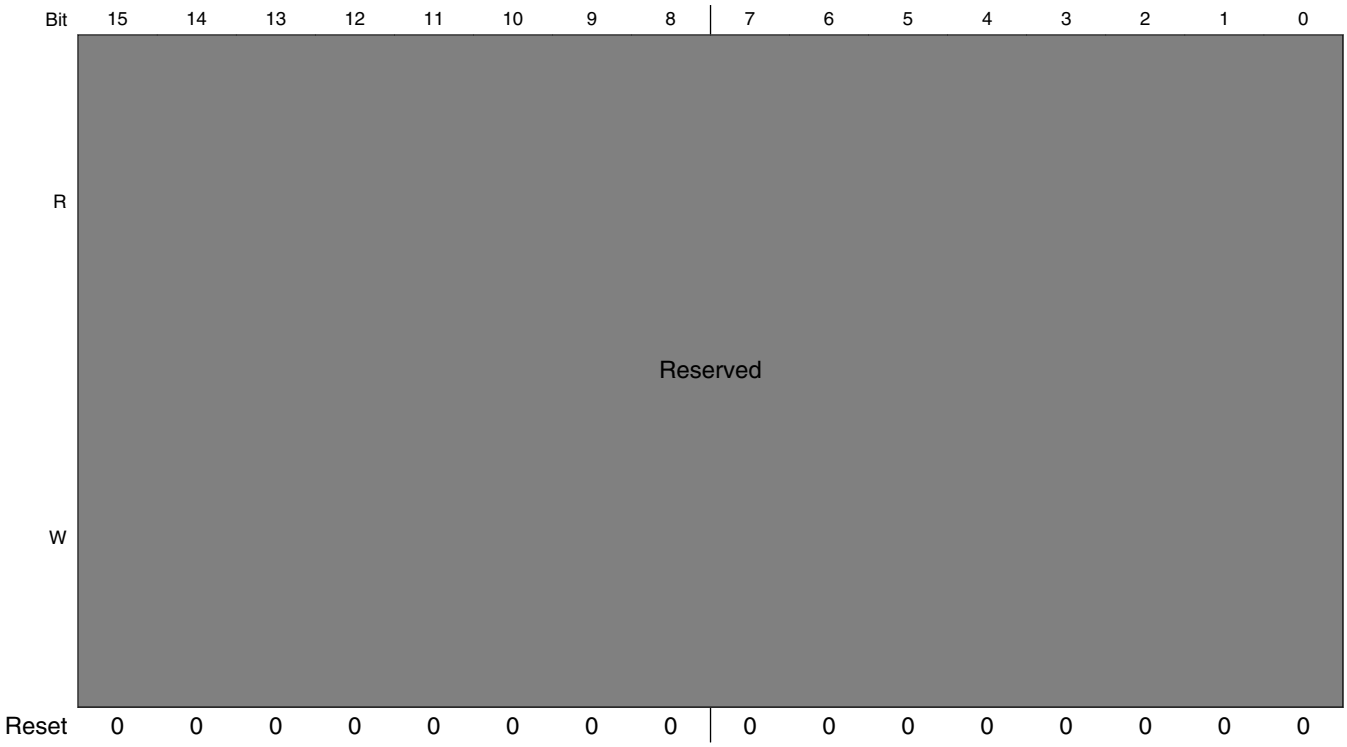
Field	Description
	Bit 0 is hardwired to 1; bits 7:1 are RW.

49.11.49 VC Resource Status Register n (PCIE_RC_VCRSR)

Offset: 0x140 + 0x18

Address: 1FF_C000h base + 158h offset = 1FF_C158h





PCIE_RC_VCRSR field descriptions

Field	Description
31–18 -	This field is reserved. Reserved
17 VC_Negotiation_Pending	VC Negotiation Pending
16 Port_Arbitration_Table_Status	Port Arbitration Table Status
-	This field is reserved. Reserved

49.12 PCIe CTRL Port Logic Memory Map/Register Definition

PCIE_PL memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1FF_C700	Ack Latency Timer and Replay Timer Register (PCIE_PL_ALTRTR)	32	R/W	03B4_3677h	49.12.1/ 4398
1FF_C704	Vendor Specific DLLP Register (PCIE_PL_VSDR)	32	R/W	FFFF_FFFFh	49.12.2/ 4398

Table continues on the next page...

PCIE_PL memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1FF_C708	Port Force Link Register (PCIE_PL_PFLR)	32	R/W	0700_0004h	49.12.3/ 4399
1FF_C70C	Ack Frequency and L0-L1 ASPM Control Register (PCIE_PL_AFLACR)	32	R/W	1B2C_2C00h	49.12.4/ 4400
1FF_C710	Port Link Control Register (PCIE_PL_PLCR)	32	R/W	0001_0020h	49.12.5/ 4402
1FF_C714	Lane Skew Register (PCIE_PL_LSR)	32	R/W	0000_0000h	49.12.6/ 4404
1FF_C718	Symbol Number Register (PCIE_PL_SNR)	32	R/W	0000_830Ah	49.12.7/ 4405
1FF_C71C	Symbol Timer Register and Filter Mask Register 1 (PCIE_PL_STRFM1)	32	R/W	0000_0640h	49.12.8/ 4406
1FF_C720	Filter Mask Register 2 (PCIE_PL_STRFM2)	32	R/W	0000_0000h	49.12.9/ 4410
1FF_C724	AMBA Multiple Outbound Decomposed NP Sub-Requests Control Register (PCIE_PL_AMODNPSR)	32	R/W	0000_0001h	49.12.10/ 4411
1FF_C728	Debug Register 0 (PCIE_PL_DEBUG0)	32	R	0000_0000h	49.12.11/ 4412
1FF_C72C	Debug Register 1 (PCIE_PL_DEBUG1)	32	R	0000_0000h	49.12.12/ 4412
1FF_C730	Transmit Posted FC Credit Status Register (PCIE_PL_TPFCSR)	32	R	0000_0000h	49.12.13/ 4413
1FF_C734	Transmit Non-Posted FC Credit Status Register (PCIE_PL_TNFCSR)	32	R	0000_0000h	49.12.14/ 4414
1FF_C738	Transmit Completion FC Credit Status Register (PCIE_PL_TCFCSR)	32	R	0000_0000h	49.12.15/ 4415
1FF_C73C	Queue Status Register (PCIE_PL_QSR)	32	R	0000_0000h	49.12.16/ 4416
1FF_C740	VC Transmit Arbitration Register 1 (PCIE_PL_VCTAR1)	32	R	0000_000Fh	49.12.17/ 4418
1FF_C744	VC Transmit Arbitration Register 2 (PCIE_PL_VCTAR2)	32	R	0000_0000h	49.12.18/ 4419
1FF_C748	VC0 Posted Receive Queue Control (PCIE_PL_VC0PRQC)	32	R/W	0010_C019h	49.12.19/ 4420
1FF_C74C	VC0 Non-Posted Receive Queue Control (PCIE_PL_VC0NRQC)	32	R/W	0020_0000h	49.12.20/ 4422
1FF_C750	VC0 Completion Receive Queue Control (PCIE_PL_VC0CRQC)	32	R/W	0080_0000h	49.12.21/ 4423
1FF_C754	VCn Posted Receive Queue Control (PCIE_PL_VC1PRQC)	32	R/W	0020_0000h	49.12.22/ 4424
1FF_C758	VCn Non-Posted Receive Queue Control (PCIE_PL_VC1NRQC)	32	R/W	0020_0000h	49.12.23/ 4426
1FF_C75C	VCn Completion Receive Queue Control (PCIE_PL_VC1CRQC)	32	R/W	0080_0000h	49.12.24/ 4427

Table continues on the next page...

PCIE_PL memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1FF_C760	VCn Posted Receive Queue Control (PCIE_PL_VC2PRQC)	32	R/W	0020_0000h	49.12.22/ 4424
1FF_C764	VCn Non-Posted Receive Queue Control (PCIE_PL_VC2NRQC)	32	R/W	0020_0000h	49.12.23/ 4426
1FF_C768	VCn Completion Receive Queue Control (PCIE_PL_VC2CRQC)	32	R/W	0080_0000h	49.12.24/ 4427
1FF_C76C	VCn Posted Receive Queue Control (PCIE_PL_VC3PRQC)	32	R/W	0020_0000h	49.12.22/ 4424
1FF_C770	VCn Non-Posted Receive Queue Control (PCIE_PL_VC3NRQC)	32	R/W	0020_0000h	49.12.23/ 4426
1FF_C774	VCn Completion Receive Queue Control (PCIE_PL_VC3CRQC)	32	R/W	0080_0000h	49.12.24/ 4427
1FF_C778	VCn Posted Receive Queue Control (PCIE_PL_VC4PRQC)	32	R/W	0020_0000h	49.12.22/ 4424
1FF_C77C	VCn Non-Posted Receive Queue Control (PCIE_PL_VC4NRQC)	32	R/W	0020_0000h	49.12.23/ 4426
1FF_C780	VCn Completion Receive Queue Control (PCIE_PL_VC4CRQC)	32	R/W	0080_0000h	49.12.24/ 4427
1FF_C784	VCn Posted Receive Queue Control (PCIE_PL_VC5PRQC)	32	R/W	0020_0000h	49.12.22/ 4424
1FF_C788	VCn Non-Posted Receive Queue Control (PCIE_PL_VC5NRQC)	32	R/W	0020_0000h	49.12.23/ 4426
1FF_C78C	VCn Completion Receive Queue Control (PCIE_PL_VC5CRQC)	32	R/W	0080_0000h	49.12.24/ 4427
1FF_C790	VCn Posted Receive Queue Control (PCIE_PL_VC6PRQC)	32	R/W	0020_0000h	49.12.22/ 4424
1FF_C794	VCn Non-Posted Receive Queue Control (PCIE_PL_VC6NRQC)	32	R/W	0020_0000h	49.12.23/ 4426
1FF_C798	VCn Completion Receive Queue Control (PCIE_PL_VC6CRQC)	32	R/W	0080_0000h	49.12.24/ 4427
1FF_C79C	VCn Posted Receive Queue Control (PCIE_PL_VC7PRQC)	32	R/W	0020_0000h	49.12.22/ 4424
1FF_C7A0	VCn Non-Posted Receive Queue Control (PCIE_PL_VC7NRQC)	32	R/W	0020_0000h	49.12.23/ 4426
1FF_C7A4	VCn Completion Receive Queue Control (PCIE_PL_VC7CRQC)	32	R/W	0080_0000h	49.12.24/ 4427
1FF_C7A8	VC0 Posted Buffer Depth (PCIE_PL_VC0PBD)	32	R	000D_0065h	49.12.25/ 4428
1FF_C7AC	VC0 Non-Posted Buffer Depth (PCIE_PL_VC0NPBD)	32	R	000D_000Dh	49.12.26/ 4429
1FF_C7B0	VC0 Completion Buffer Depth (PCIE_PL_VC0CBD)	32	R	0003_0003h	49.12.27/ 4430
1FF_C7B4	VCn Posted Buffer Depth (PCIE_PL_VC1PBD)	32	R	0000_0000h	49.12.28/ 4431

Table continues on the next page...

PCIE_PL memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1FF_C7B8	VCn Non-Posted Buffer Depth (PCIE_PL_VC1NPBD)	32	R	0000_0000h	49.12.29/ 4432
1FF_C7BC	VCn Completion Buffer Depth (PCIE_PL_VC1CBD)	32	R	0000_0000h	49.12.30/ 4433
1FF_C7C0	VCn Posted Buffer Depth (PCIE_PL_VC2PBD)	32	R	0000_0000h	49.12.28/ 4431
1FF_C7C4	VCn Non-Posted Buffer Depth (PCIE_PL_VC2NPBD)	32	R	0000_0000h	49.12.29/ 4432
1FF_C7C8	VCn Completion Buffer Depth (PCIE_PL_VC2CBD)	32	R	0000_0000h	49.12.30/ 4433
1FF_C7CC	VCn Posted Buffer Depth (PCIE_PL_VC3PBD)	32	R	0000_0000h	49.12.28/ 4431
1FF_C7D0	VCn Non-Posted Buffer Depth (PCIE_PL_VC3NPBD)	32	R	0000_0000h	49.12.29/ 4432
1FF_C7D4	VCn Completion Buffer Depth (PCIE_PL_VC3CBD)	32	R	0000_0000h	49.12.30/ 4433
1FF_C7D8	VCn Posted Buffer Depth (PCIE_PL_VC4PBD)	32	R	0000_0000h	49.12.28/ 4431
1FF_C7DC	VCn Non-Posted Buffer Depth (PCIE_PL_VC4NPBD)	32	R	0000_0000h	49.12.29/ 4432
1FF_C7E0	VCn Completion Buffer Depth (PCIE_PL_VC4CBD)	32	R	0000_0000h	49.12.30/ 4433
1FF_C7E4	VCn Posted Buffer Depth (PCIE_PL_VC5PBD)	32	R	0000_0000h	49.12.28/ 4431
1FF_C7E8	VCn Non-Posted Buffer Depth (PCIE_PL_VC5NPBD)	32	R	0000_0000h	49.12.29/ 4432
1FF_C7EC	VCn Completion Buffer Depth (PCIE_PL_VC5CBD)	32	R	0000_0000h	49.12.30/ 4433
1FF_C7F0	VCn Posted Buffer Depth (PCIE_PL_VC6PBD)	32	R	0000_0000h	49.12.28/ 4431
1FF_C7F4	VCn Non-Posted Buffer Depth (PCIE_PL_VC6NPBD)	32	R	0000_0000h	49.12.29/ 4432
1FF_C7F8	VCn Completion Buffer Depth (PCIE_PL_VC6CBD)	32	R	0000_0000h	49.12.30/ 4433
1FF_C7FC	VCn Posted Buffer Depth (PCIE_PL_VC7PBD)	32	R	0000_0000h	49.12.28/ 4431
1FF_C800	VCn Non-Posted Buffer Depth (PCIE_PL_VC7NPBD)	32	R	0000_0000h	49.12.29/ 4432
1FF_C804	VCn Completion Buffer Depth (PCIE_PL_VC7CBD)	32	R	0000_0000h	49.12.30/ 4433
1FF_C80C	Gen2 Control Register (PCIE_PL_G2CR)	32	R/W	0000_0001h	49.12.31/ 4433
1FF_C810	PHY Status (PCIE_PL_PHY_STATUS)	32	R	0000_0000h	49.12.32/ 4435

Table continues on the next page...

PCIE_PL memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1FF_C814	PHY Control (PCIE_PL_PHY_CTRL)	32	R/W	0000_0000h	49.12.33/ 4435
1FF_C818	Master Response Composer Control Register 0 (PCIE_PL_MRCCR0)	32	R/W	0000_0302h	49.12.34/ 4436
1FF_C81C	Master Response Composer Control Register 1 (PCIE_PL_MRCCR1)	32	R/W	0000_0000h	49.12.35/ 4437
1FF_C820	MSI Controller Address (PCIE_PL_MSICA)	32	R/W	0000_0000h	49.12.36/ 4438
1FF_C824	MSI Controller Upper Address (PCIE_PL_MSICUA)	32	R/W	0000_0000h	49.12.37/ 4438
1FF_C828	MSI Controller Interrupt n Enable (PCIE_PL_MSICI0_ENB)	32	R/W	0000_0000h	49.12.38/ 4439
1FF_C82C	MSI Controller Interrupt n Mask (PCIE_PL_MSICI0_MASK)	32	R/W	0000_0000h	49.12.39/ 4439
1FF_C830	MSI Controller Interrupt n Status (PCIE_PL_MSICI0_STATUS)	32	R/W	0000_0000h	49.12.40/ 4440
1FF_C834	MSI Controller Interrupt n Enable (PCIE_PL_MSICI1_ENB)	32	R/W	0000_0000h	49.12.38/ 4439
1FF_C838	MSI Controller Interrupt n Mask (PCIE_PL_MSICI1_MASK)	32	R/W	0000_0000h	49.12.39/ 4439
1FF_C83C	MSI Controller Interrupt n Status (PCIE_PL_MSICI1_STATUS)	32	R/W	0000_0000h	49.12.40/ 4440
1FF_C840	MSI Controller Interrupt n Enable (PCIE_PL_MSICI2_ENB)	32	R/W	0000_0000h	49.12.38/ 4439
1FF_C844	MSI Controller Interrupt n Mask (PCIE_PL_MSICI2_MASK)	32	R/W	0000_0000h	49.12.39/ 4439
1FF_C848	MSI Controller Interrupt n Status (PCIE_PL_MSICI2_STATUS)	32	R/W	0000_0000h	49.12.40/ 4440
1FF_C84C	MSI Controller Interrupt n Enable (PCIE_PL_MSICI3_ENB)	32	R/W	0000_0000h	49.12.38/ 4439
1FF_C850	MSI Controller Interrupt n Mask (PCIE_PL_MSICI3_MASK)	32	R/W	0000_0000h	49.12.39/ 4439
1FF_C854	MSI Controller Interrupt n Status (PCIE_PL_MSICI3_STATUS)	32	R/W	0000_0000h	49.12.40/ 4440
1FF_C858	MSI Controller Interrupt n Enable (PCIE_PL_MSICI4_ENB)	32	R/W	0000_0000h	49.12.38/ 4439
1FF_C85C	MSI Controller Interrupt n Mask (PCIE_PL_MSICI4_MASK)	32	R/W	0000_0000h	49.12.39/ 4439
1FF_C860	MSI Controller Interrupt n Status (PCIE_PL_MSICI4_STATUS)	32	R/W	0000_0000h	49.12.40/ 4440
1FF_C864	MSI Controller Interrupt n Enable (PCIE_PL_MSICI5_ENB)	32	R/W	0000_0000h	49.12.38/ 4439
1FF_C868	MSI Controller Interrupt n Mask (PCIE_PL_MSICI5_MASK)	32	R/W	0000_0000h	49.12.39/ 4439

Table continues on the next page...

PCIE_PL memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1FF_C86C	MSI Controller Interrupt n Status (PCIE_PL_MSICI5_STATUS)	32	R/W	0000_0000h	49.12.40/ 4440
1FF_C870	MSI Controller Interrupt n Enable (PCIE_PL_MSICI6_ENB)	32	R/W	0000_0000h	49.12.38/ 4439
1FF_C874	MSI Controller Interrupt n Mask (PCIE_PL_MSICI6_MASK)	32	R/W	0000_0000h	49.12.39/ 4439
1FF_C878	MSI Controller Interrupt n Status (PCIE_PL_MSICI6_STATUS)	32	R/W	0000_0000h	49.12.40/ 4440
1FF_C87C	MSI Controller Interrupt n Enable (PCIE_PL_MSICI7_ENB)	32	R/W	0000_0000h	49.12.38/ 4439
1FF_C880	MSI Controller Interrupt n Mask (PCIE_PL_MSICI7_MASK)	32	R/W	0000_0000h	49.12.39/ 4439
1FF_C884	MSI Controller Interrupt n Status (PCIE_PL_MSICI7_STATUS)	32	R/W	0000_0000h	49.12.40/ 4440
1FF_C888	MSI Controller General Purpose IO Register (PCIE_PL_MSICGPIO)	32	R/W	0000_0000h	49.12.41/ 4440
1FF_C900	iATU Viewport Register (PCIE_PL_iATUVR)	32	R/W	0000_0000h	49.12.42/ 4441
1FF_C904	iATU Region Control 1 Register (PCIE_PL_iATURC1)	32	R/W	0000_0000h	49.12.43/ 4442
1FF_C908	iATU Region Control 2 Register (PCIE_PL_iATURC2)	32	R/W	0000_0000h	49.12.44/ 4444
1FF_C90C	iATU Region Lower Base Address Register (PCIE_PL_iATURLBA)	32	R/W	0000_0000h	49.12.45/ 4447
1FF_C910	iATU Region Upper Base Address Register (PCIE_PL_iATURUBA)	32	R/W	0000_0000h	49.12.46/ 4448
1FF_C914	iATU Region Limit Address Register (PCIE_PL_iATURLA)	32	R/W	0000_FFFFh	49.12.47/ 4449
1FF_C918	iATU Region Lower Target Address Register (PCIE_PL_iATURLTA)	32	R/W	0000_0000h	49.12.48/ 4449
1FF_C91C	iATU Region Upper Target Address Register (PCIE_PL_iATURUTA)	32	R/W	0000_0000h	49.12.49/ 4450

49.12.1 Ack Latency Timer and Replay Timer Register (PCIE_PL_ALTRTR)

Offset: 0x700

Address: 1FF_C000h base + 700h offset = 1FF_C700h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Replay_Time_Limit																Round_Trip_Latency_Time_Limit															
Reset	0	0	0	0	0	0	1	1	1	0	1	1	0	1	0	0	0	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1

PCIE_PL_ALTRTR field descriptions

Field	Description
31–16 Replay_Time_Limit	Replay Time Limit The replay timer expires when it reaches this limit. The core initiates a replay upon reception of a Nak or when the replay timer expires. The default value is 4143 / 2 = 2071.. The default is then updated based on the Negotiated Link Width and Max_Payload_Size. **NOTE:** If operating at 5 Gb/s, then an additional 51 / 2 = 26 is added. This is for additional internal processing for received TLPs and transmitted DLLPs.
Round_Trip_Latency_Time_Limit	Round Trip Latency Time Limit The Ack/Nak latency timer expires when it reaches this limit. The default value is 12429 / 2 = 5215. The default is then updated based on the Negotiated Link Width and Max_Payload_Size. **Note**: If operating at 5 Gb/s, then an additional 153 / 2 = 76 is added. This is for additional internal processing for received TLPs and transmitted DLLPs.

49.12.2 Vendor Specific DLLP Register (PCIE_PL_VSDR)

Offset: 0x700 + 0x4

Address: 1FF_C000h base + 704h offset = 1FF_C704h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Vendor_Specific_DLLP																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

PCIE_PL_VSDR field descriptions

Field	Description
Vendor_Specific_DLLP	Vendor Specific DLLP Register Used to send a specific PCI Express DLLP. The application writes the 8-bit DLLP Type and 24-bits of Payload data into this register, then sets bit 0 of Port Link Control Register (PCIE_PL_PLCR) to send the DLLP.

49.12.3 Port Force Link Register (PCIE_PL_PFLR)

Offset: 0x700 + 0x8

Address: 1FF_C000h base + 708h offset = 1FF_C708h



PCIE_PL_PFLR field descriptions

Field	Description
31–24 Low_Power_Entrance_Count	Low Power Entrance Count The Power Management state will wait for this many clock cycles for the associated completion of a CfgWr to D-state register to go low-power. This register is intended for applications that do not let the core handle a completion for configuration request to the PMCSCR register. **Note**: Only used in the DM core (in EP mode), EP core, and the upstream port of a Switch.
23–22 -	This field is reserved. Reserved
21–16 Link_State	Link State The Link state that the core will be forced to when bit 15 (Force Link) is set. State encoding is defined in xmlh_ltssm.v.
15 Force_Link	Force Link Forces the Link to the state specified by the Link State field. The Force Link pulse will trigger Link re-negotiation.

Table continues on the next page...

PCIE_PL_PFLR field descriptions (continued)

Field	Description
	* Reading from this self-clearing register field always returns a 0.
14–8 -	This field is reserved. Reserved
Link_Number	Link Number Not used for Endpoint

49.12.4 Ack Frequency and L0-L1 ASPM Control Register (PCIE_PL_AFLACR)

Offset: 0x700 + 0xC

Address: 1FF_C000h base + 70Ch offset = 1FF_C70Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	Enter_ASPM_L1	L1_Entrance_ Latency			L0s_Entrance_ Latency			Common_Clock_N_FTS							
W																
Reset	0	0	0	1	1	0	1	1	0	0	1	0	1	1	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	N_FTS								Ack_Frequency							
W																
Reset	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0

PCIE_PL_AFLACR field descriptions

Field	Description
31 -	This field is reserved. Reserved
30 Enter_ASPM_L1	Enter ASPM L1 without receive in L0s. Allow core to enter ASPM L1 even when link partner did not go to L0s (receive is not in L0s).

Table continues on the next page...

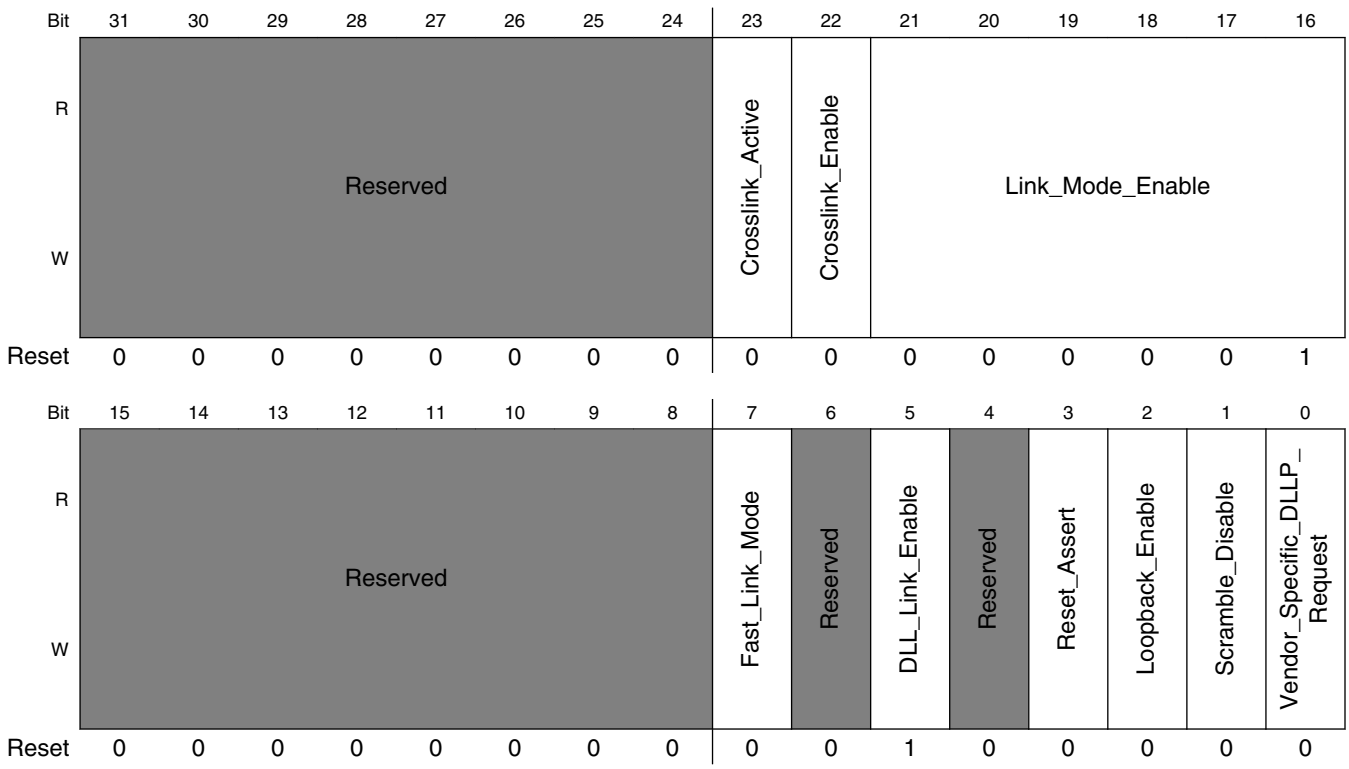
PCIE_PL_AFLACR field descriptions (continued)

Field	Description
	When not set, core goes to ASPM L1 only after idle period during which both receive and transmit are in L0s.
29–27 L1_Entrance_ Latency	L1 Entrance Latency Values correspond to: 000 1 ìs 001 2 ìs 010 4 ìs 011 8 ìs 100 16 ìs 101 32 ìs 110 64 ìs 111 64 ìs
26–24 L0s_Entrance_ Latency	L0s Entrance Latency Values correspond to: 000 1 ìs 001 2 ìs 010 3 ìs 011 4 ìs 100 5 ìs 101 6 ìs 110 7 ìs 111 7 ìs
23–16 Common_Clock_ N_FTS	Common Clock N_FTS This is the N_FTS when common clock is used. The number of Fast Training Sequence ordered sets to be transmitted when transitioning from L0s to L0. The maximum number of FTS ordered-sets that a component can request is 255. This field is writable only if the parameters are selected as follows during configuration of the core; CX_NFTS != CX_COMM_NFTS DEFAULT_L0S_EXIT_LATENCY != DEFAULT_COMM_L0S_EXIT_LATENCY DEFAULT_L1_EXIT_LATENCY != DEFAULT_COMM_L1_EXIT_LATENCY otherwise, it will be hard coded to the value of the CX_COMM_NFTS configuration parameter. **Note**: The core does not support a value of zero; a value of zero can cause the LTSSM to go into the recovery state when exiting from L0s.
15–8 N_FTS	N_FTS The number of Fast Training Sequence ordered sets to be transmitted when transitioning from L0s to L0. The maximum number of FTS ordered-sets that a component can request is 255. **Note**: The core does not support a value of zero; a value of zero can cause the LTSSM to go into the recovery state when exiting from L0s.
Ack_Frequency	Ack Frequency The core accumulates the number of pending Ack's specified here (up to 255) before sending an Ack DLLP see Link Layer: (Flow Control and ACK/NAK DLLPs) for more details.

49.12.5 Port Link Control Register (PCIE_PL_PLCR)

Offset: 0x700 + 0x10

Address: 1FF_C000h base + 710h offset = 1FF_C710h



PCIE_PL_PLCR field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 Crosslink_Active	Crosslink Active. Indicates a change from upstream to downstream or downstream to upstream. Same as the xmlh_crosslink_active output.
22 Crosslink_Enable	Crosslink Enable
21–16 Link_Mode_ Enable	Link Mode Enable The default value is the number of Lanes supported in the version of the core you are using. 000001 x1 000011 x2 000111 x4 001111 x8

Table continues on the next page...

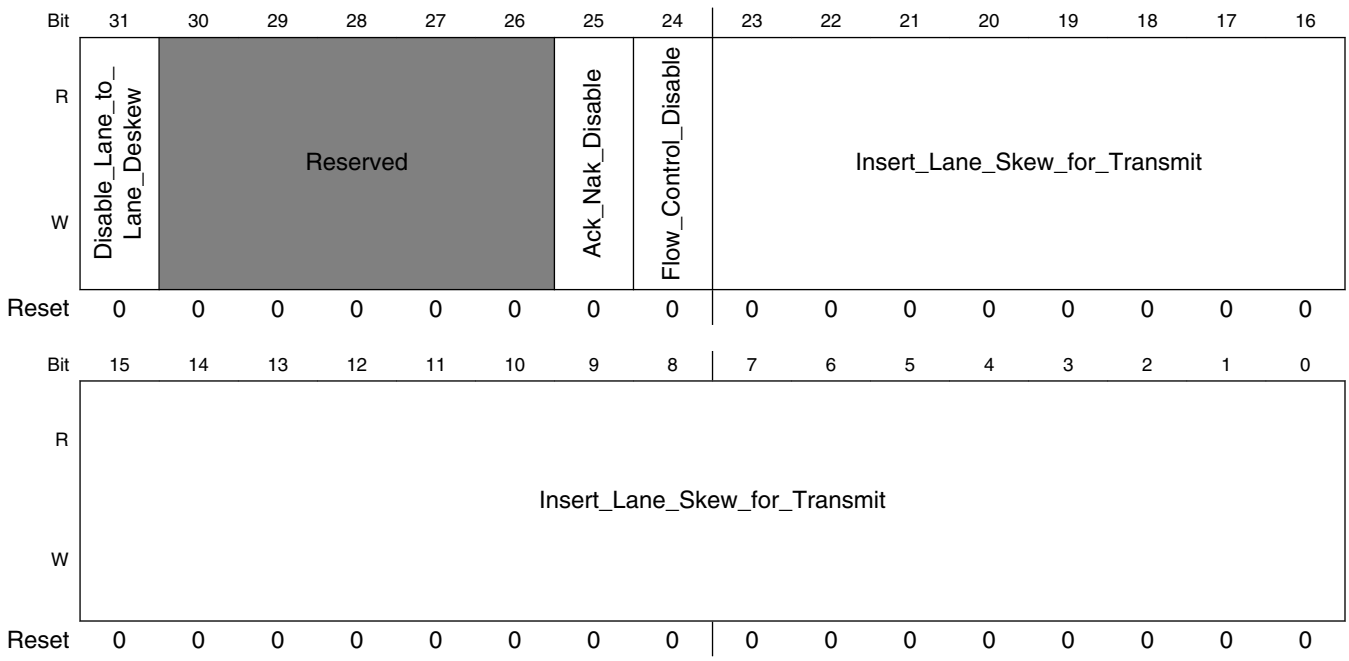
PCIE_PL_PLCR field descriptions (continued)

Field	Description
	011111 x16 111111 x32 (not supported)
15–8 -	This field is reserved. Reserved
7 Fast_Link_Mode	Fast Link Mode Sets all internal timers to Fast Mode for speeding up simulation. Forces the LTSSM training (link initialization) to use shorter time-outs and to link up faster. The scaling factor is 1024 for all internal timers. **Note**: Fast Link Mode can also be activated by setting the diag_ctrl_bus[2] pin to '1'.
6 -	This field is reserved. Reserved
5 DLL_Link_Enable	DLL Link Enable Enables Link initialization. If DLL Link Enable = 0, the core does not transmit InitFC DLLPs and does not establish a Link.
4 -	This field is reserved. Reserved
3 Reset_Assert	Reset Assert Triggers a recovery and forces the LTSSM to the Hot Reset state (downstream port only).
2 Loopback_ Enable	Loopback Enable Turns on loopback.
1 Scramble_ Disable	Scramble Disable Turns off data scrambling.
0 Vendor_Specific_ DLLP_Request	Vendor Specific DLLP Request When software writes a '1' to this bit, the core transmits the DLLP contained in the Vendor Specific DLLP Register (PCIE_PL_VSDR). * Reading from this self-clearing register field always returns a 0.

49.12.6 Lane Skew Register (PCIE_PL_LSR)

Offset: 0x700 + 0x14

Address: 1FF_C000h base + 714h offset = 1FF_C714h



PCIE_PL_LSR field descriptions

Field	Description
31 Disable_Lane_to_Lane_Deskew	Disable Lane-to-Lane Deskew Causes the core to disable the internal Lane-to-Lane deskew logic.
30–26 -	This field is reserved. Reserved
25 Ack_Nak_Disable	Ack/Nak Disable Prevents the core from sending Ack and Nak DLLPs.
24 Flow_Control_Disable	Flow Control Disable Prevents the core from sending FC DLLPs.
Insert_Lane_Skew_for_Transmit	Insert Lane Skew for Transmit (not supported for x16) Optional feature that causes the core to insert skew between Lanes for test purposes. There are three bits per Lane. The value is in units of one symbol time. For example, the value 010b for a Lane forces a skew of two symbol times for that Lane. The maximum skew value for any Lane is 5 symbol times.

49.12.7 Symbol Number Register (PCIE_PL_SNR)

Offset: 0x700 + 0x18

Address: 1FF_C000h base + 718h offset = 1FF_C718h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Configuration_ Requests			Timer_Modifier_for_Flow_Control_ Watchdog_Timer					Timer_Modifier_for_Ack_Nak_ Latency_Timer					Timer_Modifier_for_ Replay_Timer		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Timer_ Modifier_for_ Replay_ Timer		Reserved			Number_of_SKP_ Symbols			Reserved				Number_of_TS_Symbols			
Reset	1	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0

PCIE_PL_SNR field descriptions

Field	Description
31–29 Configuration_ Requests	Configuration Requests targeted at function numbers above this value will be returned with UR (unsupported request).
28–24 Timer_Modifier_ for_Flow_ Control_ Watchdog_Timer	Timer Modifier for Flow Control Watchdog Timer Increases the timer value for the Flow Control watchdog timer, in increments of 16 clock cycles.
23–19 Timer_Modifier_ for_Ack_Nak_ Latency_Timer	Timer Modifier for Ack/Nak Latency Timer Increases the timer value for the Ack/Nak latency timer, in increments of 64 clock cycles.
18–14 Timer_Modifier_ for_Replay_Timer	Timer Modifier for Replay Timer Increases the timer value for the replay timer, in increments of 64 clock cycles.
13–11 -	This field is reserved. Reserved
10–8 Number_of_SKP_ Symbols	Number of SKP Symbols
7–4 -	This field is reserved. Reserved
Number_of_TS_ Symbols	Number of TS Symbols Sets the number of TS identifier symbols that are sent in TS1 and TS2 ordered sets.

49.12.8 Symbol Timer Register and Filter Mask Register 1 (PCIE_PL_STRFM1)

Offset: 0x700 + 0x1C

Table 49-63. Filter Mask 1: Mask RADM Filtering and Error Handling Rules

Bits	Name	Name
31	CX_FLT_MASK_RC_CFG_DISCARD	CX_FLT_MASK_RC_CFG_DISCARD 0 For RADM RC filter to not allow CFG transaction being received 1 For RADM RC filter to allow CFG transaction being received
30	CX_FLT_MASK_RC_IO_DISCARD	CX_FLT_MASK_RC_IO_DISCARD 0 For RADM RC filter to not allow IO transaction being received 1 For RADM RC filter to allow IO transaction being received
29	CX_FLT_MASK_MSG_DROP	CX_FLT_MASK_MSG_DROP 0 Drop MSG TLP (except for Vendor MSG) 1 Do not Drop MSG (except for Vendor MSG)
28	CX_FLT_MASK_CPL_ECRC_DISCARD	CX_FLT_MASK_CPL_ECRC_DISCARD 0 Discard TLPs with ECRC errors for CPL type 1 Allow TLPs with ECRC errors to be passed up for CPL type
27	CX_FLT_MASK_ECRC_DISCARD	CX_FLT_MASK_ECRC_DISCARD 0 Discard TLPs with ECRC errors 1 Allow TLPs with ECRC errors to be passed up
26	CX_FLT_MASK_CPL_LEN_MATCH	CX_FLT_MASK_CPL_LEN_MATCH 0 Enforce length match for received CPL TLPs; a violation results in cpl_abort, and possibly AER of unexp_cpl_err 1 MASK length match for received CPL TLPs
25	CX_FLT_MASK_CPL_ATTR_MATCH	CX_FLT_MASK_CPL_ATTR_MATCH 0 Enforce attribute match for received CPL TLPs; a violation results in a malformed TLP error, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca

Table continues on the next page...

Table 49-63. Filter Mask 1: Mask RADM Filtering and Error Handling Rules (continued)

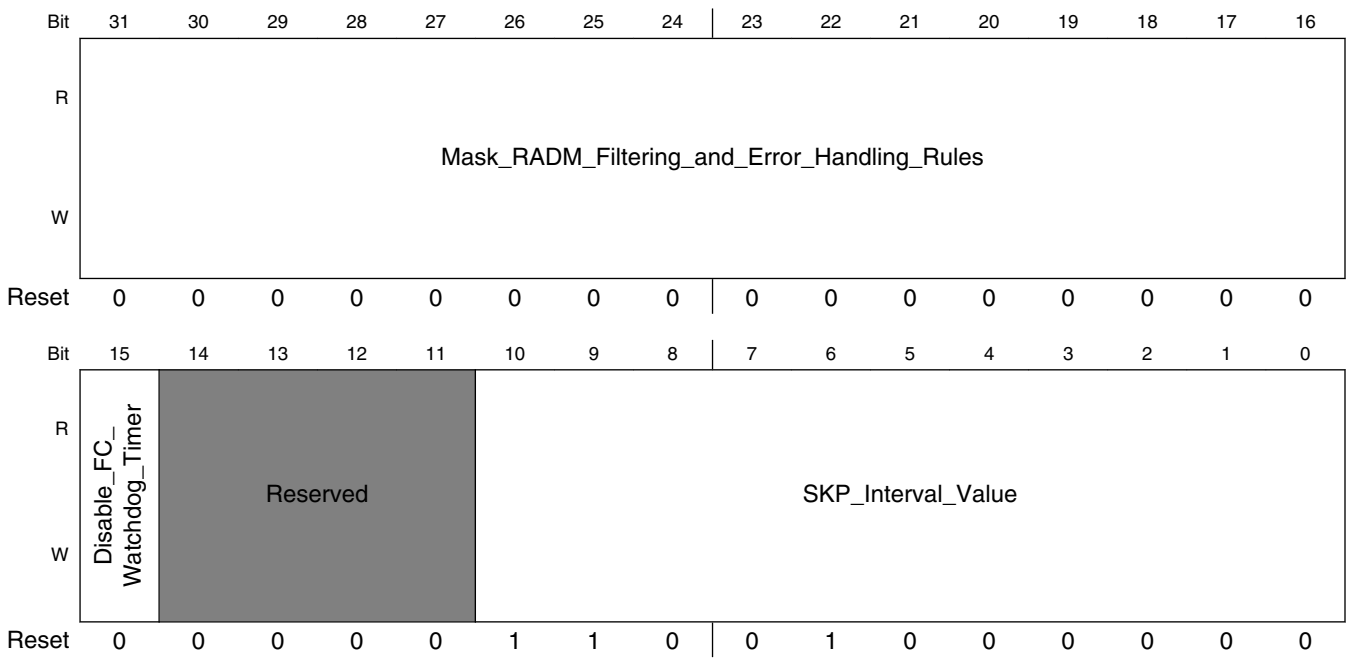
Bits	Name	Name
		1 Mask attribute match for received CPL TLPs
24	CX_FLT_MASK_CPL_TC_MATCH	CX_FLT_MASK_CPL_TC_MATCH 0 Enforce Traffic Class match for received CPL TLPs; a violation results in a malformed TLP error, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca 1 Mask Traffic Class match for received CPL TLPs
23	CX_FLT_MASK_CPL_FUNC_MATCH	CX_FLT_MASK_CPL_FUNC_MATCH 0 Enforce function match for received CPL TLPs; a violation results in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca 1 Mask function match for received CPL TLPs
22	CX_FLT_MASK_CPL_REQID_MATCH	CX_FLT_MASK_CPL_REQID_MATCH 0 Enforce Req. Id match for received CPL TLPs; a violation result in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca 1 Mask Req. Id match for received CPL TLPs
21	CX_FLT_MASK_CPL_TAGERR_MATCH	CX_FLT_MASK_CPL_TAGERR_MATCH 0 Enforce Tag Error Rules for received CPL TLPs; a violation result in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca 1 Mask Tag Error Rules for received CPL TLPs
20	CX_FLT_MASK_LOCKED_RD_AS_UR	CX_FLT_MASK_LOCKED_RD_AS_UR 0 Treat locked Read TLPs as UR for EP; Supported for RC 1 Treat locked Read TLPs as Supported for EP; UR for RC
19	CX_FLT_MASK_CFG_TYPE1_RE_AS_UR	CX_FLT_MASK_CFG_TYPE1_RE_AS_UR 0 Treat CFG type1 TLPs as UR for EP; Supported for RC 1 Treat CFG type1 TLPs as Supported for EP; UR for RC.
18	CX_FLT_MASK_UR_OUTSIDE_BAR	CX_FLT_MASK_UR_OUTSIDE_BAR 0 Treat out-of-bar TLPs as UR

Table continues on the next page...

Table 49-63. Filter Mask 1: Mask RADM Filtering and Error Handling Rules (continued)

Bits	Name	Name
		1 Treat out-of-bar TLPs as Supported Requests
17	CX_FLT_MASK_UR_POIS	CX_FLT_MASK_UR_POIS 0 Treat poisoned TLPs as UR 1 Treat poisoned TLPs as Supported Requests
16	CX_FLT_MASK_UR_FUNC_MISMATCH	CX_FLT_MASK_UR_FUNC_MISMATCH 0 Treat Function MisMatched TLPs as UR 1 Treat Function MisMatched TLPs as Supported

Address: 1FF_C000h base + 71Ch offset = 1FF_C71Ch



PCIE_PL_STRFM1 field descriptions

Field	Description
31–16 Mask_RADM_Filtering_and_Error_Handling_Rules	Mask RADM Filtering and Error Handling Rules: Mask 1 There are several mask bits to turn off the filtering and error handling rules In each case, 0 applies the associated filtering rule and 1 masks the associated filtering rule. A more detailed description for these bits is provided in Table 5-348. [31]: Mask filtering of received Configuration Requests (RC mode only) [30]: Mask filtering of received I/O Requests (RC mode only)

Table continues on the next page...

PCIE_PL_STRFM1 field descriptions (continued)

Field	Description
	[29]: Send Message TLPs to the application on RTRGT1 and send decoded Message on the SII (1) or send decoded Message on the SII, then drop the Message TLPs (0). The default value for this bit is the inverse of `FLT_DROP_MSG. That is, if `FLT_DROP_MSG = 1, then the default value of this bit is 0 (drop Message TLPs). Note that this bit only controls message TLPs other than Vendor MSGs. Vendor MSGs are controlled by Filter Mask Register 2 (PCIE_PL_STRFM2), bits [1:0]. [28]: Mask ECRC error filtering for Completions [27]: Mask ECRC error filtering [26]: Mask Length mismatch error for received Completions [25]: Mask Attributes mismatch error for received Completions [24]: Mask Traffic Class mismatch error for received Completions [23]: Mask function mismatch error for received Completions [23]: Mask Requester ID mismatch error for received Completions [21]: Mask Tag error rules for received Completions [20]: Mask Locked Request filtering [19]: Mask Type 1 Configuration Request filtering [18]: Mask BAR match filtering [17]: Mask poisoned TLP filtering [16]: Mask function mismatch filtering for incoming Requests
15 Disable_FC_ Watchdog_Timer	Disable FC Watchdog Timer
14–11 -	This field is reserved. Reserved
SKP_Interval_ Value	SKP Interval Value The number of symbol times to wait between transmitting SKP ordered sets. Note that the core actually waits the number of symbol times in this register plus 1 between transmitting SKP ordered sets. The application must program this register accordingly. For example, if 1536 we're programmed into this register (in a 250MHz core), then the core will actually transmit Skp ordered sets once every 1537 symbol times. Also, the value programmed to this register is actually clock ticks and not symbol times. In a 125MHz core, programming the value programmed to this register should be scaled down by a factor of 2 (since 1 clock tick=2 symbol times in this case).

49.12.9 Filter Mask Register 2 (PCIE_PL_STRFM2)

Offset: 0x700 + 0x20

Address: 1FF_C000h base + 720h offset = 1FF_C720h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Mask_RADM_Filtering_and_Error_Handling_Rules																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

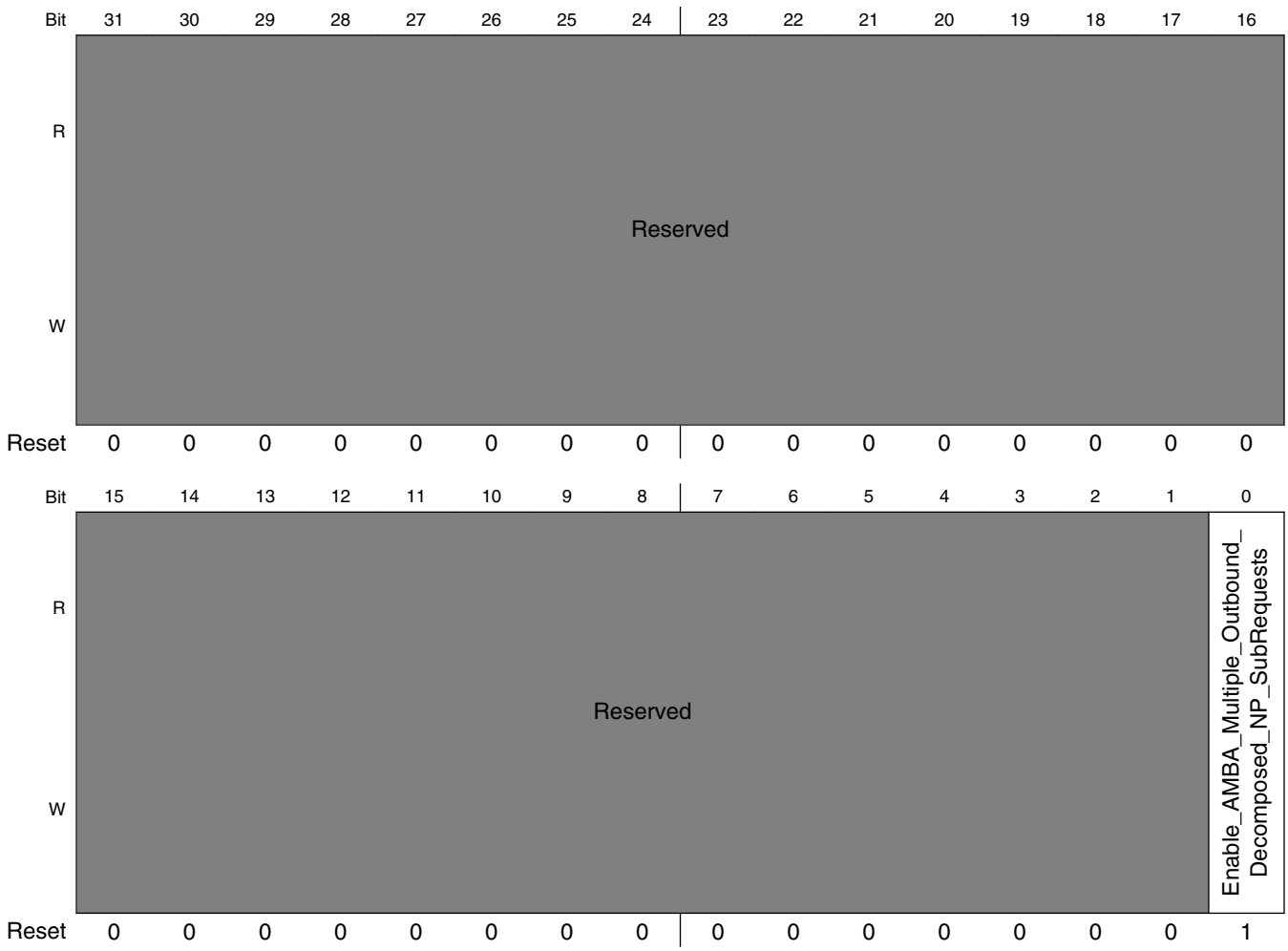
PCIE_PL_STRFM2 field descriptions

Field	Description
Mask_RADM_ Filtering_and_ Error_Handling_ Rules	Mask RADM Filtering and Error Handling Rules: Mask 2 It defaults to 0x0 There are several mask bits used to turn off the filtering and error handling rules . •[31:4]: Reserved •[3]: `CX_FLT_MASK_HANDLE_FLUSH - 0: Disable Core Filter to handle flush request - 1: Enable Core Filter to handle flush request •[2]: `CX_FLT_MASK_DABORT_4UCPL - 0: Enable DLLP abort for unexpected CPL - 1: Do not enable DLLP abort for unexpected CPL •[1]: `CX_FLT_MASK_VENMSG1_DROP - 0: Vendor MSG Type 1 dropped silently - 1: Vendor MSG Type 1 not dropped •[0]: `CX_FLT_MASK_VENMSG0_DROP - 0: Vendor MSG Type 0 dropped with UR error reporting - 1: Vendor MSG Type 0 not dropped

49.12.10 AMBA Multiple Outbound Decomposed NP Sub-Requests Control Register (PCIE_PL_AMODNPSR)

Offset: 0x700 + 0x24

Address: 1FF_C000h base + 724h offset = 1FF_C724h



PCIE_PL_AMODNPSR field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 Enable_AMBA_Multiple_Outbound_Decomposed_	Enable AMBA Multiple Outbound Decomposed NP Sub- Requests. This bit when set to '0' disables the possibility of having multiple outstanding non-posted requests that were derived from decomposition of an outbound AMBA request. See Supported AXI Burst Operations for more details.

Table continues on the next page...

PCIE_PL_AMODNPSR field descriptions (continued)

Field	Description
NP_ SubRequests	You should not clear this register unless your application master is requesting an amount of read data greater than Max_Read_Request_Size, and the remote device (or switch) is reordering completions that have different tags

49.12.11 Debug Register 0 (PCIE_PL_DEBUG0)

Offset: 0x700 + 0x28

[31:28]: rmlh_ts_link_ctrl Link control bits advertised by link partner

[27]: rmlh_ts_lane_num_is_k23 Currently receiving k237 (PAD) in place of lane number

[26]: rmlh_ts_link_num_is_k23 Currently receiving k237 (PAD) in place of link number

[25]: rmlh_rcvd_idle[0] Receiver is receiving logical idle

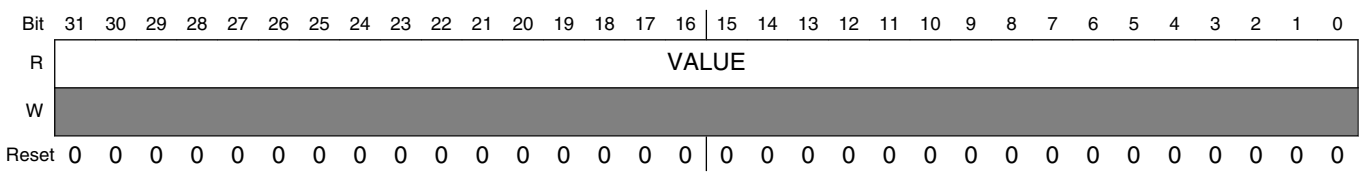
[24]: rmlh_rcvd_idle[1] 2nd symbol is also idle (16bit PHY interface only)

[23:8]: mac_phy_txdata PIPE Transmit data

[7:6]: mac_phy_txdatak PIPE transmit K indication

[5:0]: xmlh_ltssm_state LTSSM current state. See source for encodings

Address: 1FF_C000h base + 728h offset = 1FF_C728h



PCIE_PL_DEBUG0 field descriptions

Field	Description
VALUE	The value on cxpl_debug_info[31:0].

49.12.12 Debug Register 1 (PCIE_PL_DEBUG1)

Offset: 0x700 + 0x2C

[63]: xmlh_scrambler_disable Scrambling disabled for the link

[62]: xmlh_link_disable LTSSM in DISABLE state. Link inoperable

[61]: xmlh_link_in_training LTSSM performing link training

[60]: xmlh_rcvr_revrs_pol_en LTSSM testing for polarity reversal

[59]: xmlh_training_rst_n LTSSM-negotiated link reset

[58:55]: 0000b Constant/reserved

[54]: mac_phy_txdetectrx_loop PIPE receiver detect/loopback request

[53]: mac_phy_txelecidle[0] PIPE transmit electrical idle request

[52]: mac_phy_txcompliance[0] PIPE transmit compliance request

[51]: app_init_rst Application request to initiate training reset

[50:48]: 000b Constant/reserved

[47:40]: rmlh_ts_link_num Link number advertised/confirmed by link partner

[39:38]: 00b Constant/reserved

[37]: xmtbyte_skip_sent A skip ordered set has been transmitted

[36]: xmlh_link_up LTSSM reports PHY link up

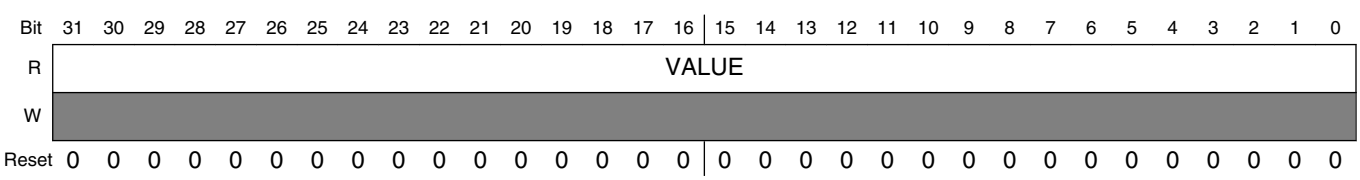
[35]: rmlh_inskip_rcv Receiver reports skip reception

[34]: rmlh_ts1_rcvd TS1 training sequence received (pulse)

[33]: rmlh_ts2_rcvd TS2 training sequence received (pulse)

[32]: rmlh_rcvd_lane_rev Receiver detected lane reversal

Address: 1FF_C000h base + 72Ch offset = 1FF_C72Ch



PCIE_PL_DEBUG1 field descriptions

Field	Description
VALUE	The value on cxpl_debug_info[63:32].

49.12.13 Transmit Posted FC Credit Status Register (PCIE_PL_TPFCSR)

Offset: 0x700 + 0x30

*Default value depends on the number of advertised credits for header and data {12'b0, xtlh_xadm_ph_cdts, xtlh_xadm_pd_cdts}; If the number of advertised completion credits (both header and data) are infinite, then the default would be {12'b0, 8'hFF, 12'hFFF}.

Address: 1FF_C000h base + 730h offset = 1FF_C730h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved												Transmit_Posted_Header_FC_Credits			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Transmit_Posted_Header_FC_Credits				Transmit_Posted_Data_FC_Credits											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_TPFCSR field descriptions

Field	Description
31–20 -	This field is reserved. Reserved
19–12 Transmit_Posted_Header_FC_Credits	Transmit Posted Header FC Credits The Posted Header credits advertised by the receiver at the other end of the Link, updated with each UpdateFC DLLP.
Transmit_Posted_Data_FC_Credits	Transmit Posted Data FC Credits The Posted Data credits advertised by the receiver at the other end of the Link, updated with each UpdateFC DLLP.

49.12.14 Transmit Non-Posted FC Credit Status Register (PCIE_PL_TNFCSR)

Offset: 0x700 + 0x34

*Default value depends on the number of advertised credits for header and data {12'b0, xtlh_xadm_nph_cdts, xtlh_xadm_npd_cdts}; If the number of advertised completion credits (both header and data) are infinite, then the default would be {12'b0, 8'hFF, 12'hFFF}.

Address: 1FF_C000h base + 734h offset = 1FF_C734h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												Transmit_Non_Posted_Header_FC_Credits				Transmit_Non_Posted_Data_FC_Credits															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_TNFCSR field descriptions

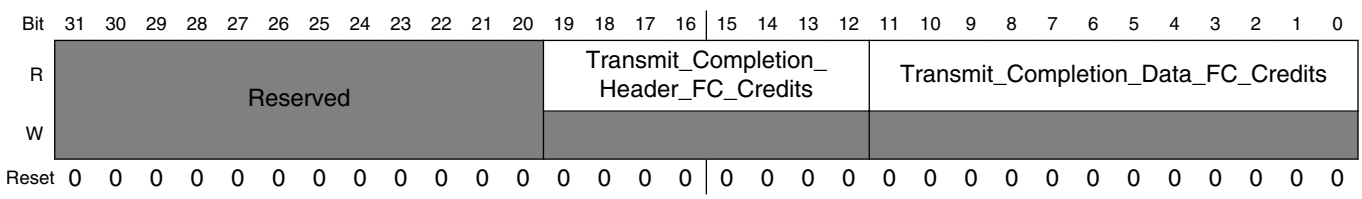
Field	Description
31–20 -	This field is reserved. Reserved
19–12 Transmit_Non_Posted_Header_FC_Credits	Transmit Non-Posted Header FC Credits The Non-Posted Header credits advertised by the receiver at the other end of the Link, updated with each UpdateFC DLLP.
Transmit_Non_Posted_Data_FC_Credits	Transmit Non-Posted Data FC Credits The Non-Posted Data credits advertised by the receiver at the other end of the Link, updated with each UpdateFC DLLP.

49.12.15 Transmit Completion FC Credit Status Register (PCIE_PL_TCFCSR)

Offset: 0x700 + 0x38

*Default value depends on the number of advertised credits for header and data {12'b0, xtlh_xadm_cplh_cdts, xtlh_xadm_cpld_cdts}; If the number of advertised completion credits (both header and data) are infinite, then the default would be {12'b0, 8'hFF, 12'hFFF}.

Address: 1FF_C000h base + 738h offset = 1FF_C738h



PCIE_PL_TCFCSR field descriptions

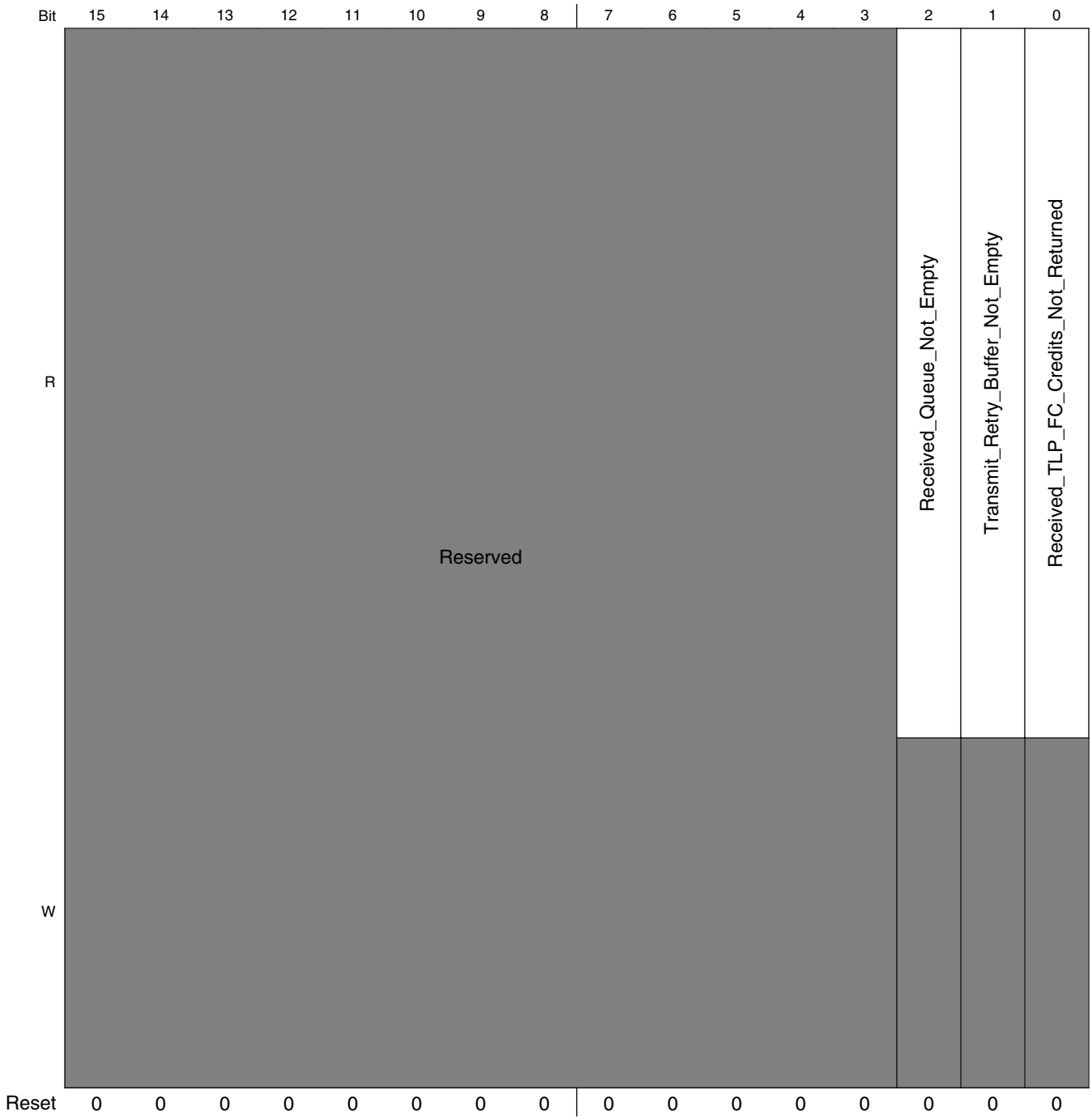
Field	Description
31–20 -	This field is reserved. Reserved
19–12 Transmit_Completion_Header_FC_Credits	Transmit Completion Header FC Credits The Completion Header credits advertised by the receiver at the other end of the Link, updated with each UpdateFC DLLP.
Transmit_Completion_Data_FC_Credits	Transmit Completion Data FC Credits The Completion Data credits advertised by the receiver at the other end of the Link, updated with each UpdateFC DLLP.

49.12.16 Queue Status Register (PCIE_PL_QSR)

Offset: 0x700 + 0x3C

Address: 1FF_C000h base + 73Ch offset = 1FF_C73Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



PCIE_PL_QSR field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
2 Received_ Queue_Not_ Empty	Received Queue Not Empty Indicates there is data in one or more of the receive buffers.

Table continues on the next page...

PCIE_PL_QSR field descriptions (continued)

Field	Description
1 Transmit_Retry_Buffer_Not_Empty	Transmit Retry Buffer Not Empty Indicates that there is data in the transmit retry buffer.
0 Received_TLP_FC_Credits_Not_Returned	Received TLP FC Credits Not Returned Indicates that the core has sent a TLP but has not yet received an UpdateFC DLLP indicating that the credits for that TLP have been restored by the receiver at the other end of the Link. **Note**: This bit is for simulation only and will always be synthesized as 0.

49.12.17 VC Transmit Arbitration Register 1 (PCIE_PL_VCTAR1)

Offset: 0x700 + 0x40

VC Transmit Arbitration Registers 1 and 2 specify the weights assigned to VC0-VC7 to be used for WRR transmit arbitration for VCs in the LPVC group. The following rules and restrictions apply regarding the values programmed in VC Transmit Arbitration Registers 1 and 2:

- There are 8 bits allocated for each weight value.
- No weight value for a VC in the LPVC group can be less than 1.
- No weight value can be greater than the number of phases in the selected arbitration scheme.
- The sum of the weights assigned to all VCs in the LPVC group must equal the number of phases in the selected arbitration scheme. For example, if 64-phase WRR arbitration is selected, the total of all WRR Weight values for all VCs in the LPVC group must equal 64.

Each of the VC numbers listed in the bit field table is a VC ID, not the VC structure number. VC Transmit Arbitration Registers 1 and 2 are hardwired to the default values set by the configuration parameters listed in the Default columns of the bit field table.

Address: 1FF_C000h base + 740h offset = 1FF_C740h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WRR_Weight_for_VC3								WRR_Weight_for_VC2								WRR_Weight_for_VC1								WRR_Weight_for_VC0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

PCIE_PL_VCTAR1 field descriptions

Field	Description
31–24 WRR_Weight_ for_VC3	WRR Weight for VC3
23–16 WRR_Weight_ for_VC2	WRR Weight for VC2
15–8 WRR_Weight_ for_VC1	WRR Weight for VC1
WRR_Weight_ for_VC0	WRR Weight for VC0

49.12.18 VC Transmit Arbitration Register 2 (PCIE_PL_VCTAR2)

Offset: 0x700 + 0x44

VC Transmit Arbitration Registers 1 and 2 specify the weights assigned to VC0-VC7 to be used for WRR transmit arbitration for VCs in the LPVC group. The following rules and restrictions apply regarding the values programmed in VC Transmit Arbitration Registers 1 and 2:

- There are 8 bits allocated for each weight value.
- No weight value for a VC in the LPVC group can be less than 1.
- No weight value can be greater than the number of phases in the selected arbitration scheme.
- The sum of the weights assigned to all VCs in the LPVC group must equal the number of phases in the selected arbitration scheme. For example, if 64-phase WRR arbitration is selected, the total of all WRR Weight values for all VCs in the LPVC group must equal 64.

Each of the VC numbers listed in the bit field table is a VC ID, not the VC structure number. VC Transmit Arbitration Registers 1 and 2 are hardwired to the default values set by the configuration parameters listed in the Default columns of the bit field table.

Address: 1FF_C000h base + 744h offset = 1FF_C744h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WRR_Weight_for_VC7								WRR_Weight_for_VC6								WRR_Weight_for_VC5								WRR_Weight_for_VC4							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_VCTAR2 field descriptions

Field	Description
31–24 WRR_Weight_ for_VC7	WRR Weight for VC7
23–16 WRR_Weight_ for_VC6	WRR Weight for VC6
15–8 WRR_Weight_ for_VC5	WRR Weight for VC5
WRR_Weight_ for_VC4	WRR Weight for VC4

49.12.19 VC0 Posted Receive Queue Control (PCIE_PL_VC0PRQC)

NOTE

The data and header credits fields of the Receive Queue Control registers are used in all receive buffer configurations.

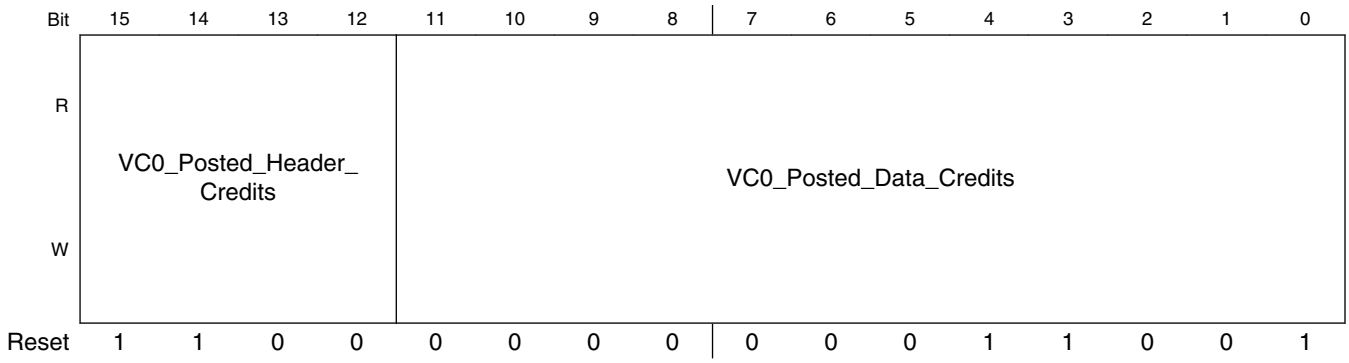
NOTE

All other fields of the Receive Queue Control and Depth registers are used only in the segmented- buffer configuration.

Offset: 0x700 + 0x48

Address: 1FF_C000h base + 748h offset = 1FF_C748h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VC_Ordering_for_ Receive_Queues	TLP_Type_Ordering_for_ VC0	Reserved						VC0_Posted_TLP_ Queue_Mode			Reserved	VC0_Posted_Header_ Credits			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0



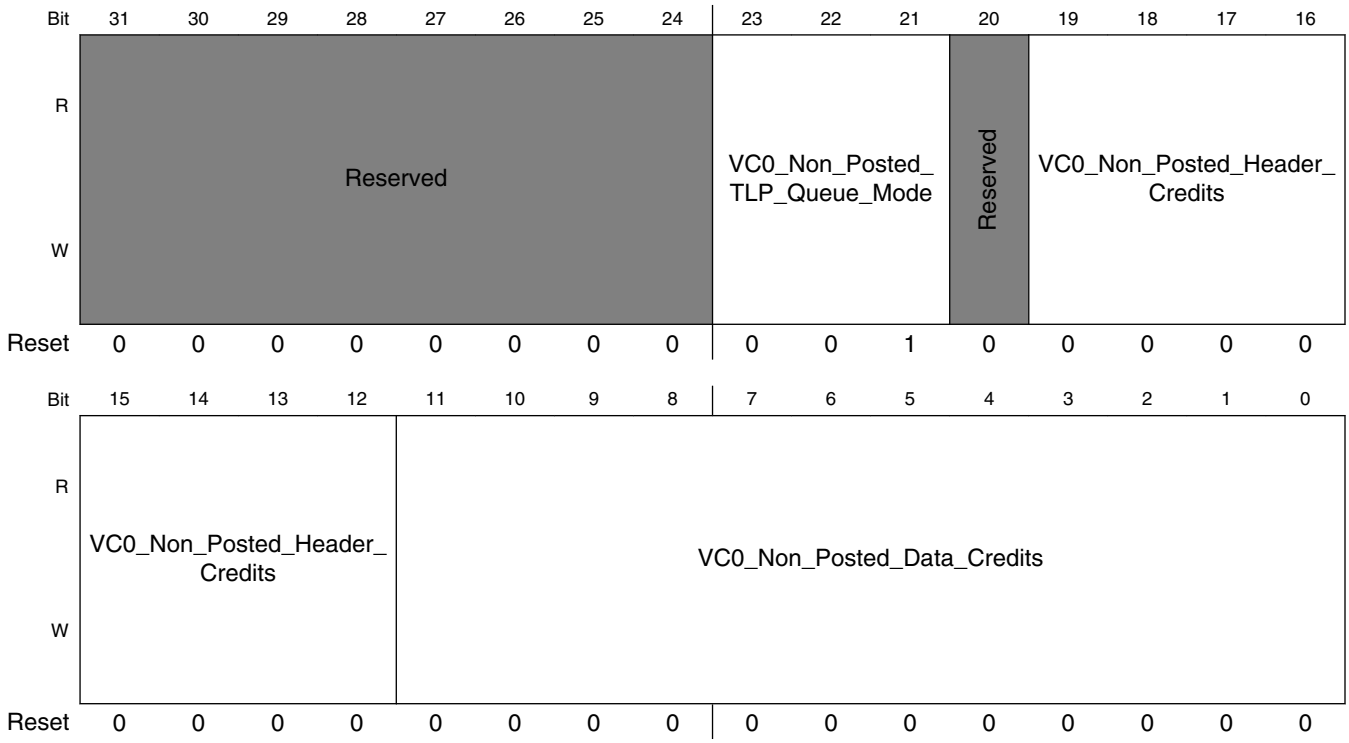
PCIE_PL_VC0PRQC field descriptions

Field	Description
31 VC_Ordering_for_Receive_Queues	VC Ordering for Receive Queues Determines the VC ordering rule for the receive queues, used only in the segmented-buffer configuration, writable through the DBI: 1 Strict ordering, higher numbered VCs have higher priority 0 Round robin
30 TLP_Type_Ordering_for_VC0	TLP Type Ordering for VC0 Determines the TLP type ordering rule for VC0 receive queues, used only in the segmented-buffer configuration, writable through the DBI: 1 Ordering of received TLPs follows the rules in PCI Express 3.0 Specification. 0 Strict ordering for received TLPs: Posted, then Completion, then Non-Posted
29–24 -	This field is reserved. Reserved
23–21 VC0_Posted_TLP_Queue_Mode	VC0 Posted TLP Queue Mode The operating mode of the Posted receive queue for VC0, used only in the segmented-buffer configuration, writable through the DBI. Only one bit can be set at a time: • Bit 23: Bypass • Bit 22: Cut-through • Bit 21: Store-and-forward
20 -	This field is reserved. Reserved
19–12 VC0_Posted_Header_Credits	VC0 Posted Header Credits The number of initial Posted header credits for VC0, used for all receive queue buffer configurations. This field is not writable through the DBI
VC0_Posted_Data_Credits	VC0 Posted Data Credits The number of initial Posted data credits for VC0, used for all receive queue buffer configurations. This field is not writable through the DBI

49.12.20 VC0 Non-Posted Receive Queue Control (PCIE_PL_VC0NRQC)

Offset: 0x700 + 0x4C

Address: 1FF_C000h base + 74Ch offset = 1FF_C74Ch



PCIE_PL_VC0NRQC field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–21 VC0_Non_Posted_TLP_Queue_Mode	VC0 Non-Posted TLP Queue Mode The operating mode of the Non-Posted receive queue for VC0, used only in the segmented-buffer configuration, writable through the DBI. Only one bit can be set at a time: • Bit 23: Bypass • Bit 22: Cut-through • Bit 21: Store-and-forward
20 -	This field is reserved. Reserved
19–12 VC0_Non_Posted_Header_Credits	VC0 Non-Posted Header Credits The number of initial Non-Posted header credits for VC0, used for all receive queue buffer configurations. This field is not writable through the DBI

Table continues on the next page...

PCIE_PL_VC0NRQC field descriptions (continued)

Field	Description
VC0_Non_ Posted_Data_ Credits	VC0 Non-Posted Data Credits The number of initial Non-Posted data credits for VC0, used for all receive queue buffer configurations. This field is not writable through the DBI

49.12.21 VC0 Completion Receive Queue Control (PCIE_PL_VC0CRQC)

Offset: 0x700 + 0x50

Address: 1FF_C000h base + 750h offset = 1FF_C750h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								VC0_Completion_ TLP_Queue_Mode			Reserved	VC0_Completion_Header_ Credits			
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VC0_Completion_Header_ Credits				VC0_Completion_Data_Credits											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_VC0CRQC field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–21 VC0_ Completion_ TLP_Queue_ Mode	VC0 Completion TLP Queue Mode The operating mode of the Completion receive queue for VC0, used only in the segmented-buffer configuration, writable through the DBI. Only one bit can be set at a time: • Bit 23: Bypass

Table continues on the next page...

PCIE_PL_VC0CRQC field descriptions (continued)

Field	Description
	• Bit 22: Cut-through • Bit 21: Store-and-forward
20 -	This field is reserved. Reserved
19–12 VC0_ Completion_ Header_Credits	VC0 Completion Header Credits The number of initial Completion header credits for VC0, used for all receive queue buffer configurations. This field is not writable through the DBI
VC0_ Completion_ Data_Credits	VC0 Completion Data Credits The number of initial Completion data credits for VC0, used for all receive queue buffer configurations. This field is not writable through the DBI.

49.12.22 VCn Posted Receive Queue Control (PCIE_PL_VC*n*PRQC)

Offset: 0x700 + 0x48 + C*n (n=[1:7])

Address: 1FF_C000h base + 754h offset + (12d × i), where i=0d to 6d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	TLP_Type_Ordering_for_VCn	Reserved						VC1_Posted_TLP_Queue_Mode			Reserved	VCn_Posted_Header_Credits			
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VCn_Posted_Header_Credits				VCn_Posted_Data_Credits											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_VC*n*PRQC field descriptions

Field	Description
31 -	This field is reserved. Reserved
30 TLP_Type_Ordering_for_VCn	TLP Type Ordering for VCn Determines the TLP type ordering rule for VCn receive queues, used only in the segmented-buffer configuration, writable through the DBI: • 1: Ordering of received TLPs follows the rules in *PCI Express Base 3.0 Specification* • 0: Strict ordering for received TLPs: Posted, then Completion, then Non-Posted
29–24 -	This field is reserved. Reserved
23–21 VC1_Posted_TLP_Queue_Mode	VCn Posted TLP Queue Mode The operating mode of the Posted receive queue for VCn, used only in the segmented-buffer configuration, writable through the DBI. Only one bit can be set at a time: • Bit 23: Bypass • Bit 22: Cut-through • Bit 21: Store-and-forward
20 -	This field is reserved. Reserved
19–12 VCn_Posted_Header_Credits	VCn Posted Header Credits The number of initial Posted header credits for VCn, used for all receive queue buffer configurations. This field is not writable through the DBI
VCn_Posted_Data_Credits	VCn Posted Data Credits The number of initial Posted data credits for VCn, used for all receive queue buffer configurations. This field is not writable through the DBI

49.12.23 VCn Non-Posted Receive Queue Control (PCIE_PL_VC*n*NRQC)

Offset: 0x700 + 0x4C + C*n (n=[1:7])

Address: 1FF_C000h base + 758h offset + (12d × i), where i=0d to 6d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								VCn_Non_Posted_TLP_Queue_Mode			Reserved	VCn_Non_Posted_Header_Credits			
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VCn_Non_Posted_Header_Credits				VCn_Non_Posted_Data_Credits											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_VC*n*NRQC field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–21 VCn_Non_Posted_TLP_Queue_Mode	VCn Non-Posted TLP Queue Mode The operating mode of the Non-Posted receive queue for VCn, used only in the segmented-buffer configuration, writable through the DBI. Only one bit can be set at a time: • Bit 23: Bypass • Bit 22: Cut-through • Bit 21: Store-and-forward
20 -	This field is reserved. Reserved
19–12 VCn_Non_Posted_Header_Credits	VCn Non-Posted Header Credits The number of initial Non-Posted header credits for VCn, used for all receive queue buffer configurations. This field is not writable through the DBId.

Table continues on the next page...

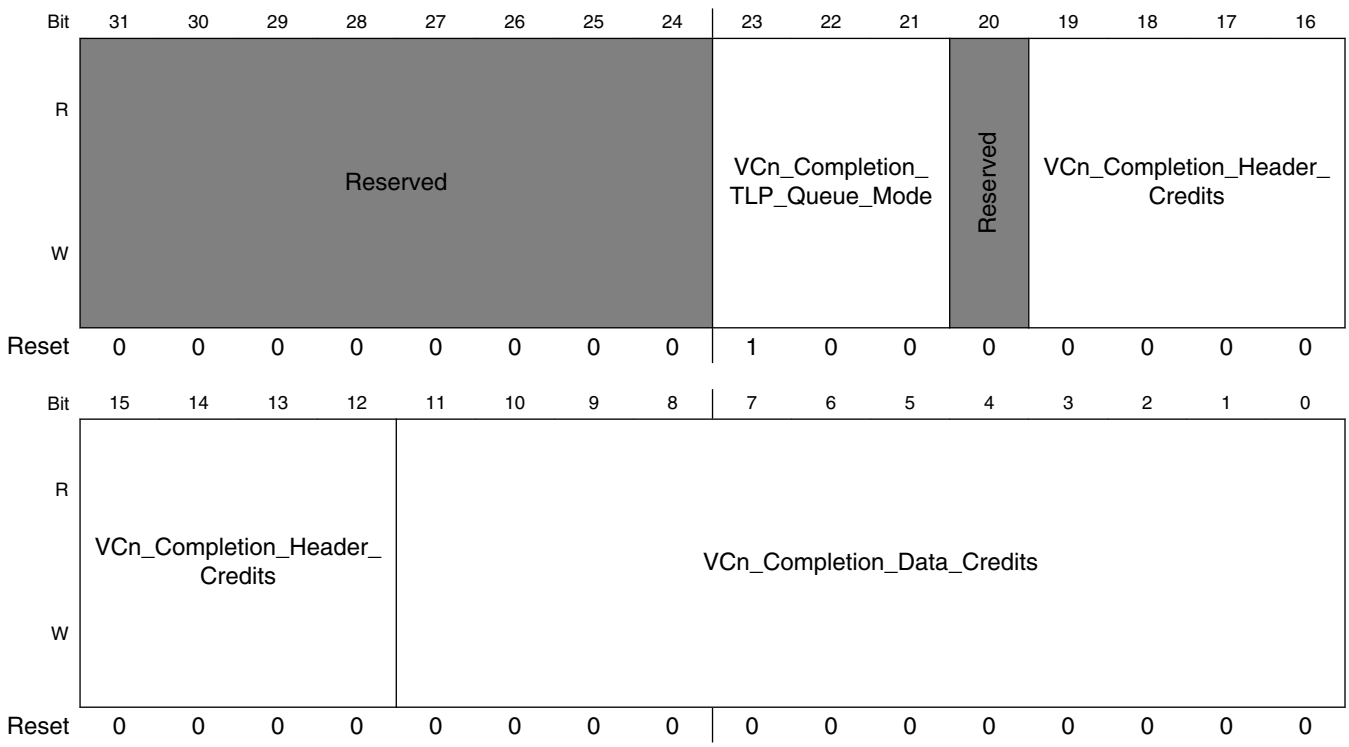
PCIE_PL_VC*n*NRQC field descriptions (continued)

Field	Description
VCn_Non_Posted_Data_Credits	VCn Non-Posted Data Credits The number of initial Non-Posted data credits for VCn, used for all receive queue buffer configurations. This field is not writable through the DBI

49.12.24 VCn Completion Receive Queue Control (PCIE_PL_VC*n*CRQC)

Offset: 0x700 + 0x50 + C*n (n=[1:7])

Address: 1FF_C000h base + 75Ch offset + (12d × i), where i=0d to 6d



PCIE_PL_VC*n*CRQC field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–21 VCn_Completion_TLP_Queue_Mode	VCn Completion TLP Queue Mode The operating mode of the Completion receive queue for VCn, used only in the segmented-buffer configuration, writable through the DBI. Only one bit can be set at a time: • Bit 23: Bypass

Table continues on the next page...

PCIE_PL_VC*n*CRQC field descriptions (continued)

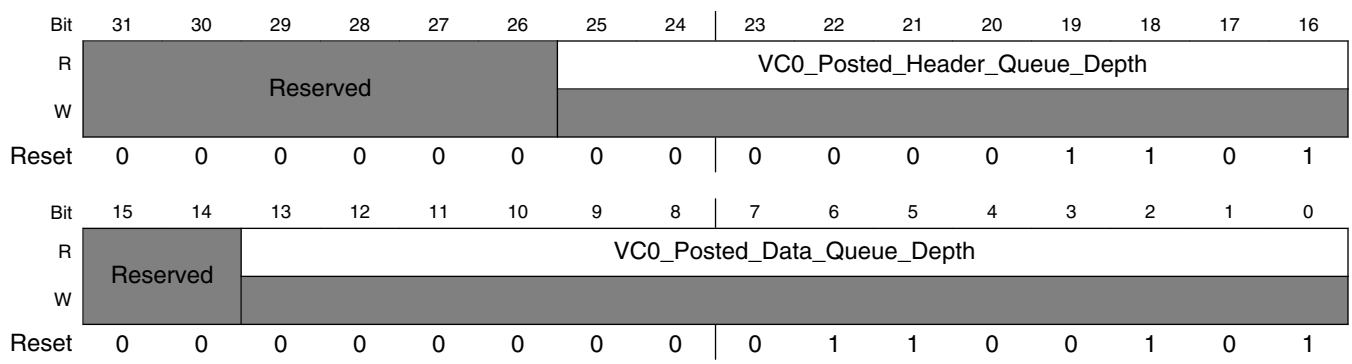
Field	Description
	• Bit 22: Cut-through • Bit 21: Store-and-forward
20 -	This field is reserved. Reserved
19–12 VCn_ Completion_ Header_Credits	VCn Completion Header Credits The number of initial Completion header credits for VCn, used for all receive queue buffer configurations. This field is not writable through the DBI
VCn_ Completion_ Data_Credits	VCn Completion Data Credits The number of initial Completion data credits for VCn, used for all receive queue buffer configurations. This field is not writable through the DBI

49.12.25 VC0 Posted Buffer Depth (PCIE_PL_VC0PBD)

- The Buffer Depth registers are used only in the segmented-buffer configuration.
- Writing to these registers is not possible (through the DBI)

Offset: 0x700 + 0xA8

Address: 1FF_C000h base + 7A8h offset = 1FF_C7A8h



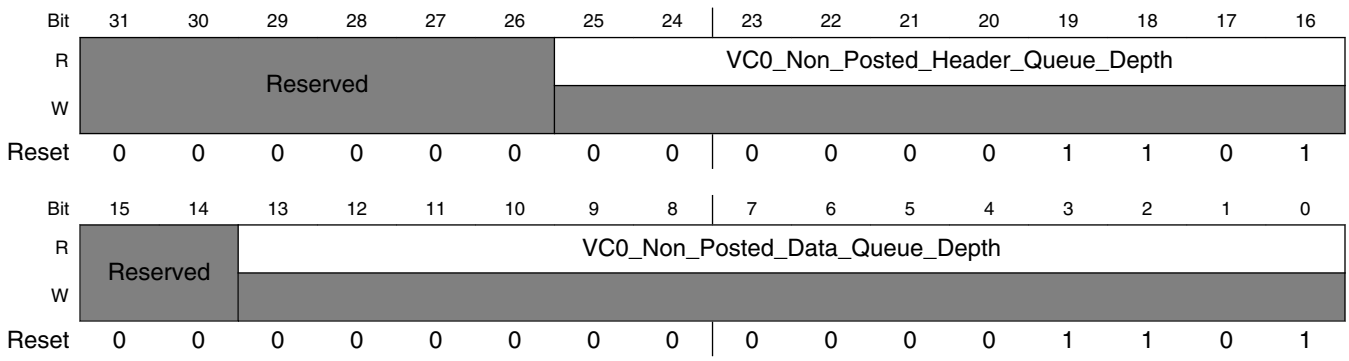
PCIE_PL_VC0PBD field descriptions

Field	Description
31–26 -	This field is reserved. Reserved
25–16 VC0_Posted_ Header_Queue_ Depth	VC0 Posted Header Queue Depth Sets the number of entries in the Posted header queue for VC0 when using the segmented-buffer configuration.Not writable through the DBI
15–14 -	This field is reserved. Reserved
VC0_Posted_ Data_Queue_ Depth	VC0 Posted Data Queue Depth Sets the number of entries in the Posted data queue for VC0 when using the segmented-buffer configuration.Not writable through the DBI

49.12.26 VC0 Non-Posted Buffer Depth (PCIE_PL_VC0NPBD)

Offset: 0x700 + 0xAC

Address: 1FF_C000h base + 7ACh offset = 1FF_C7ACh



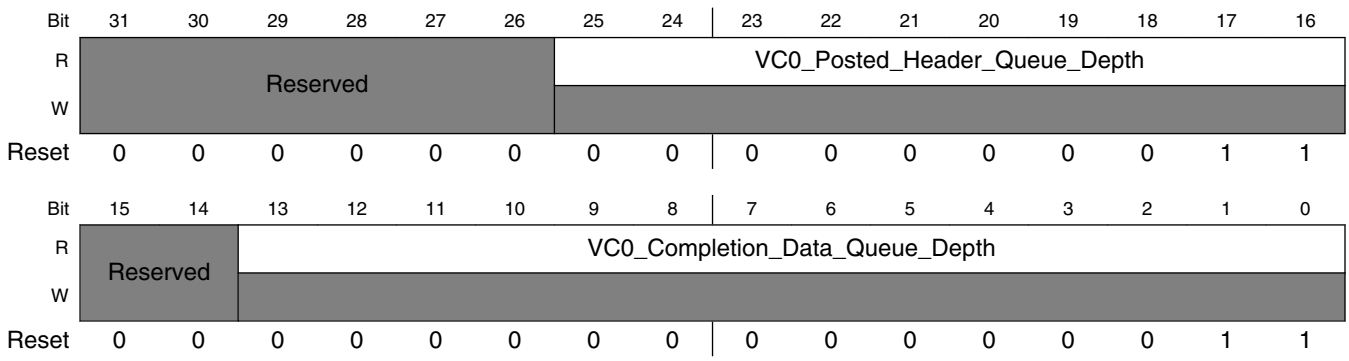
PCIE_PL_VC0NPBD field descriptions

Field	Description
31–26 -	This field is reserved. Reserved
25–16 VC0_Non_Posted_Header_Queue_Depth	VC0 Non-Posted Header Queue Depth Sets the number of entries in the Non-Posted header queue for VC0 when using the segmented-buffer configuration.Not writable through the DBI
15–14 -	This field is reserved. Reserved
VC0_Non_Posted_Data_Queue_Depth	VC0 Non-Posted Data Queue Depth Sets the number of entries in the Non-Posted data queue for VC0 when using the segmented-buffer configuration. Not writable through the DBI

49.12.27 VC0 Completion Buffer Depth (PCIE_PL_VC0CBD)

Offset: 0x700 + 0xB0

Address: 1FF_C000h base + 7B0h offset = 1FF_C7B0h



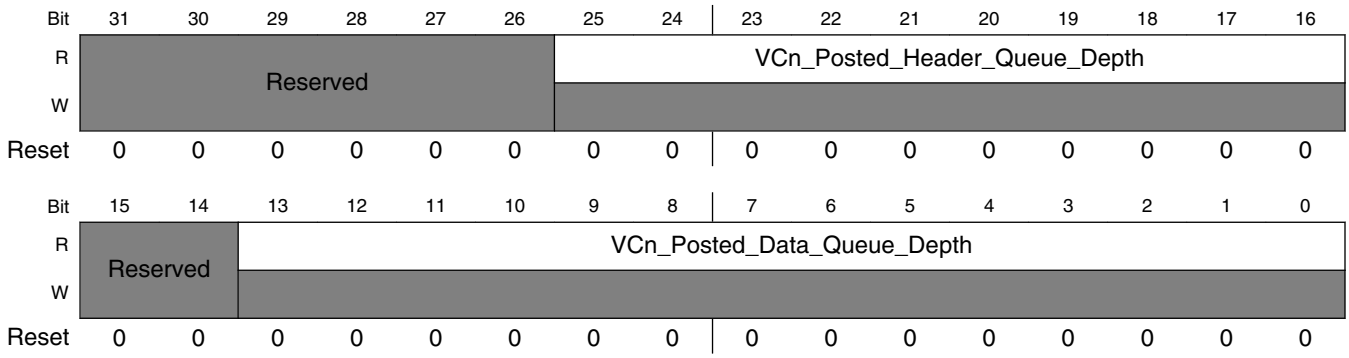
PCIE_PL_VC0CBD field descriptions

Field	Description
31–26 -	This field is reserved. Reserved
25–16 VC0_Posted_Header_Queue_Depth	VC0 Posted Header Queue Depth Sets the number of entries in the Completion header queue for VC0 when using the segmented-buffer configuration.Not writable through the DBI
15–14 -	This field is reserved. Reserved
VC0_Completion_Data_Queue_Depth	VC0 Completion Data Queue Depth Sets the number of entries in the Completion data queue for VC0 when using the segmented-buffer configuration. Not writable through the DBI

49.12.28 VCn Posted Buffer Depth (PCIE_PL_VC*n*PBD)

Offset: 0x700 + 0xA8 + C*n (n=[1:7])

Address: 1FF_C000h base + 7B4h offset + (12d × i), where i=0d to 6d



PCIE_PL_VC*n*PBD field descriptions

Field	Description
31–26 -	This field is reserved. Reserved
25–16 VCn_Posted_Header_Queue_Depth	VCn Posted Header Queue Depth Sets the number of entries in the Posted header queue for VCn when using the segmented-buffer configuration.Not writable through the DBI
15–14 -	This field is reserved. Reserved
VCn_Posted_Data_Queue_Depth	VCn Posted Data Queue Depth Sets the number of entries in the Posted data queue for VCn when using the segmented-buffer configuration.Not writable through the DBI

49.12.29 VCn Non-Posted Buffer Depth (PCIE_PL_VC*n*NPBD)

Offset: 0x700 + 0xAC + C*n (n=[1:7])

Address: 1FF_C000h base + 7B8h offset + (12d × i), where i=0d to 6d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved						VCn_Non_Posted_Header_Queue_Depth									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		VCn_Non_Posted_Data_Queue_Depth													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

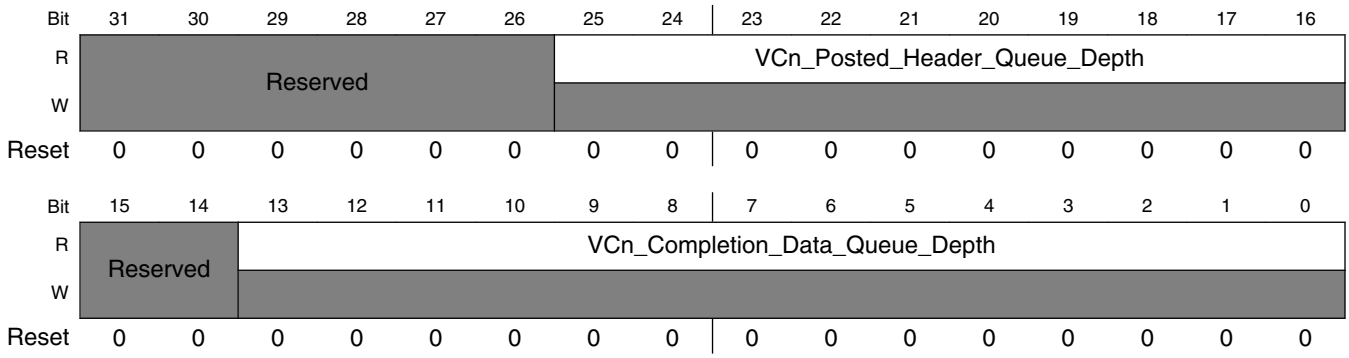
PCIE_PL_VC*n*NPBD field descriptions

Field	Description
31–26 -	This field is reserved. Reserved
25–16 VCn_Non_Posted_Header_Queue_Depth	VCn Non-Posted Header Queue Depth Sets the number of entries in the Non-Posted header queue for VCn when using the segmented-buffer configuration. Not writable through the DBI
15–14 -	This field is reserved. Reserved
VCn_Non_Posted_Data_Queue_Depth	VCn Non-Posted Data Queue Depth Sets the number of entries in the Non-Posted data queue for VCn when using the segmented-buffer configuration. Not writable through the DBI

49.12.30 VCn Completion Buffer Depth (PCIE_PL_VC*n*CBD)

Offset: 0x700 + 0xB0 + C*n (n=[1:7])

Address: 1FF_C000h base + 7BCh offset + (12d × i), where i=0d to 6d



PCIE_PL_VC*n*CBD field descriptions

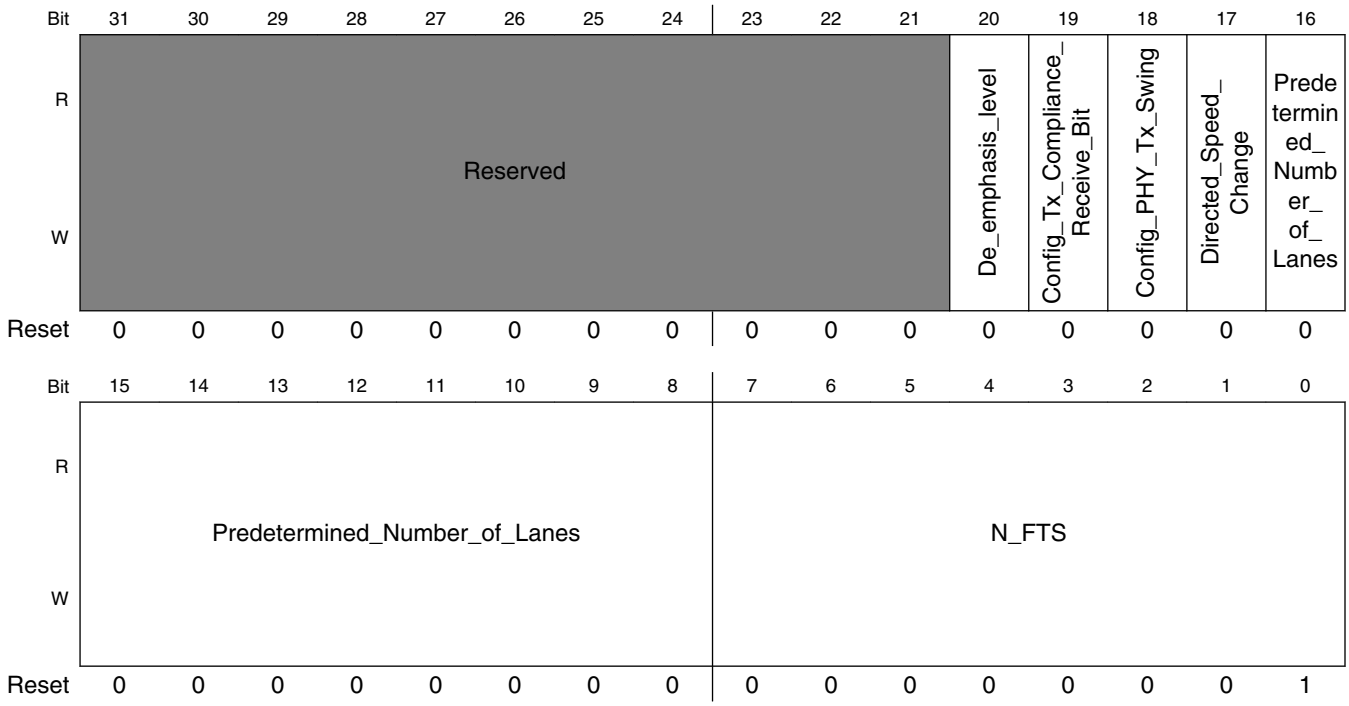
Field	Description
31–26 -	This field is reserved. Reserved
25–16 VCn_Posted_Header_Queue_Depth	VCn Posted Header Queue Depth Sets the number of entries in the Completion header queue for VCn when using the segmented-buffer configuration. Not writable through the DBI
15–14 -	This field is reserved. Reserved
VCn_Completion_Data_Queue_Depth	VCn Completion Data Queue Depth Sets the number of entries in the Completion data queue for VCn when using the segmented-buffer configuration. Not writable through the DBI

49.12.31 Gen2 Control Register (PCIE_PL_G2CR)

The Port Logic Gen2 Control Register controls features specific to data rates greater than 2.5 GT/s. The "Lane Enable" field is an exception in that it applies regardless of the data rate.

Offset: 0x700 + 0x10C

Address: 1FF_C000h base + 80Ch offset = 1FF_C80Ch



PCIE_PL_G2CR field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 De_emphasis_level	Used to set the de-emphasis level for upstream ports.
19 Config_Tx_Compliance_Receive_Bit	Config Tx Compliance Receive Bit When set to 1, signals LTSSM to transmit TS ordered sets with the compliance receive bit assert (equal to 1).
18 Config_PHY_Tx_Swing	Config PHY Tx Swing Indicates the voltage level the PHY should drive. When set to 1, indicates Full Swing. When set to 0, indicates Low Swing
17 Directed_Speed_Change	Directed Speed Change Indicates to the LTSSM whether or not to initiate a speed change to Gen2
16–8 Predetermined_Number_of_Lanes	Predetermined Number of Lanes Used to limit the effective link width to ignore "broken" lanes that detect a receiver. Indicates the number of lanes to check for exit from Electrical Idle in POLLING.ACTIVE and L2.IDLE. It is possible that the LTSSM may detect a Receiver on a 'bad' or 'broken' lane during the Detect Sub-state. However, it is also possible that such a lane may also fail to exit Electrical Idle and therefore prevent a valid link from being configured. Encoding is as follows: 0x01 = 1 lane

Table continues on the next page...

PCIE_PL_G2CR field descriptions (continued)

Field	Description
N_FTS	Sets the Number of Fast Training Sequences (N_FTS) that the core advertises as its N_FTS during Gen2 Link training. This value is used to inform the Link partner about the PHY's ability to recover synchronization after a low power state. The number should be provided by the PHY vendor. **NOTE: Note**: Do not set N_FTS to zero; doing so can cause the LTSSM to go into the recovery state when exiting from L0s.

49.12.32 PHY Status (PCIE_PL_PHY_STATUS)

Offset: 0x700 + 0x110

Address: 1FF_C000h base + 810h offset = 1FF_C810h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PHY_Status																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_PHY_STATUS field descriptions

Field	Description
PHY_Status	PHY Status Data received directly from the phy_cfg_status bus.

49.12.33 PHY Control (PCIE_PL_PHY_CTRL)

Offset: 0x700 + 0x114

Address: 1FF_C000h base + 814h offset = 1FF_C814h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PHY_Control																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_PHY_CTRL field descriptions

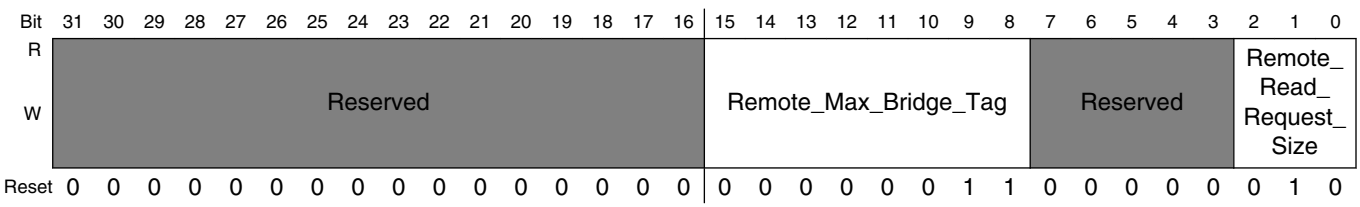
Field	Description
PHY_Control	PHY Control Data sent directly to the cfg_phy_control bus.

49.12.34 Master Response Composer Control Register 0 (PCIE_PL_MRCCR0)

You must not modify these registers for AHB configurations, as this feature is only supported for AXI.

Offset: 0x700 + 0x118

Address: 1FF_C000h base + 818h offset = 1FF_C818h



PCIE_PL_MRCCR0 field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15–8 Remote_Max_ Bridge_Tag	Remote Max Bridge Tag Specifies the maximum number (-1) of Non-Posted AMBA requests outstanding at one time issued from the bridge master. Excludes any internally created TLP's as a result of decomposition. The core will automatically derive this when bits[2:0] (Remote Read Request Size) are written to. It is s saturated in core at CX_REMOTE_MAX_TAG since the TRGT_CPL_LUT has only this many entries. Therefore it is important that the core is initially sized (at configuration time pre-silicon) with the true maximum value of CX_REMOTE_MAX_TAG to take advantage of the ability to dynamically increase remote_max_bridge_tag from CX_REMOTE_MAX_BRIDGE_TAG to any new value up to a maximum of CX_REMOTE_MAX_TAG
7–3 -	This field is reserved. Reserved
Remote_Read_ Request_Size	Remote Read Request Size Specifies the largest amount of data (bytes) that will ever be requested (via an inbound MemRd TLP) by a remote device. Must never be programmed with a value that exceeds the value represented by the configuration parameter CX_REMOTE_RD_REQ_SIZE as the Master Response Composer RAM in the AXI bridge is sized using CX_REMOTE_RD_REQ_SIZE. Must only be programmed with the values 3'b000 to 3'b101. Any other value has the same effect as writing a value of 3'b000. Encoding is as follows: 000 128 001 256 010 512

Table continues on the next page...

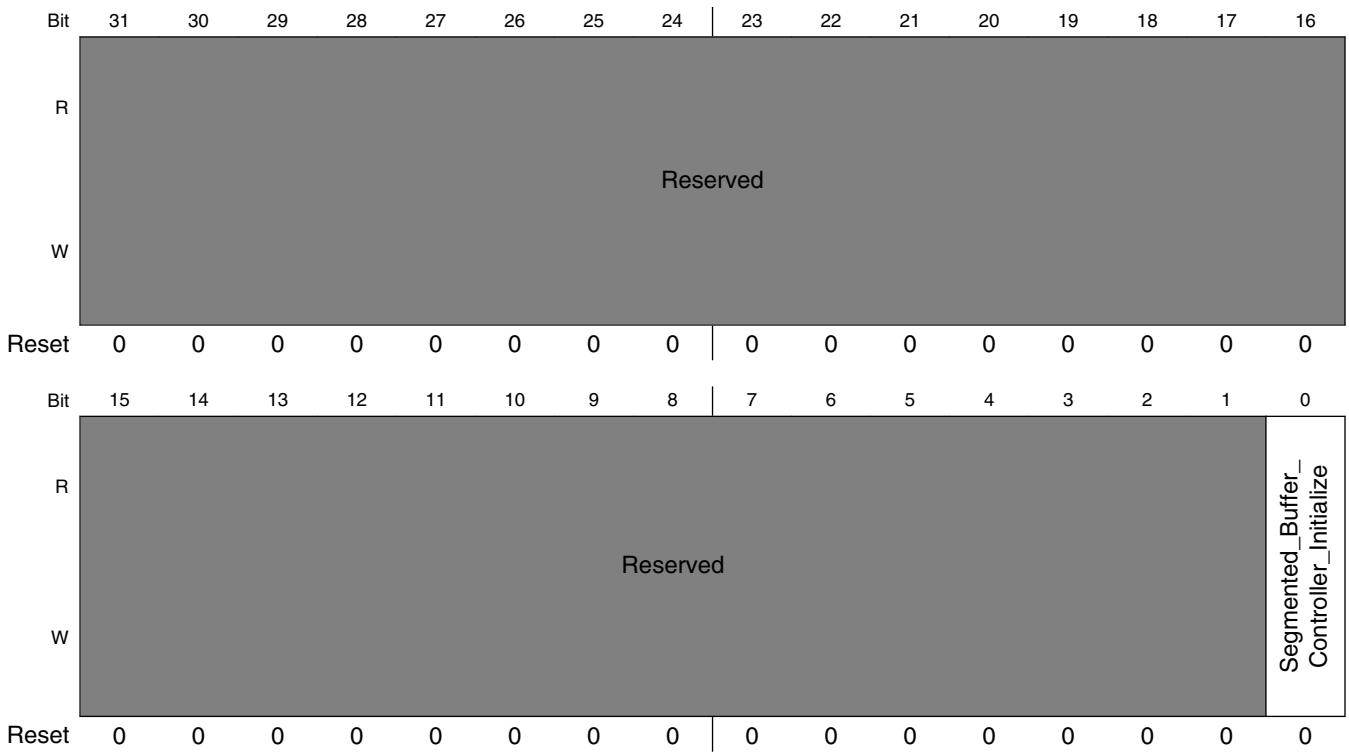
PCIE_PL_MRCCR0 field descriptions (continued)

Field	Description
	011 1024 100 2048 101 4096 default: 128

49.12.35 Master Response Composer Control Register 1 (PCIE_PL_MRCCR1)

Offset: 0x700 + 0x11C

Address: 1FF_C000h base + 81Ch offset = 1FF_C81Ch



PCIE_PL_MRCCR1 field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 Segmented_ Buffer_ Controller_ Initialize	Segmented Buffer Controller Initialize. Writing '1' to this (self-clearing register) causes any changes in the Master Response Composer Control Register 0 to take place in the bridge hardware. The sbc_init register triggers the initialization of the segmented buffer controller (DWC_sbc).

Table continues on the next page...

PCIE_PL_MRCCR1 field descriptions (continued)

Field	Description
	When sbc_init is written to, the segmented buffer controller (DWC_sbc) samples cfg_remote_max_bridge_tag and starts the internal finite state machine (FSM). * Reading from this self-clearing register field always returns a 0.

49.12.36 MSI Controller Address (PCIE_PL_MSICA)

See AHB/AXI MSI Controller (Optional in RC mode).

These registers are not part of the PCI Express MSI Capability Register structure which is detailed at MSI Capability Register Details.

Offset: 0x700 + 0x120

Address: 1FF_C000h base + 820h offset = 1FF_C820h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_Controller_Address																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_MSICA field descriptions

Field	Description
MSI_Controller_Address	MSI Controller Address System specified address for MSI memory write transaction termination. Within the AHB/AXI Bridge, every received Memory Write Request is examined to see if it targets the MSI Address that has been specified in the MSI Controller Address Register and also to see if it satisfies the definition of an MSI Interrupt Request. If these conditions are satisfied the Memory Write Request is marked as an MSI Request.

49.12.37 MSI Controller Upper Address (PCIE_PL_MSICUA)

Offset: 0x700 + 0x124

Address: 1FF_C000h base + 824h offset = 1FF_C824h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_Controller_Upper_Address																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_MSICUA field descriptions

Field	Description
MSI_Controller_ Upper_Address	MSI Controller Upper Address System specified upper address for MSI memory write transaction termination. Allows functions to support a 64- bit MSI address.

49.12.38 MSI Controller Interrupt n Enable (PCIE_PL_MSICI*n*_ENB)

Offset: 0x700 + 0x128 + C*n (n=[0:7])

Address: 1FF_C000h base + 828h offset + (12d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_Interruptn_Enable																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_MSICI*n*_ENB field descriptions

Field	Description
MSI_Interruptn_ Enable	MSI Interrupt#n Enable Specifies which interrupts are enabled. If an MSI is received from a disabled interrupt, no status bit gets set in MSI Controller Interrupt Status Register. Each bit corresponds to a single MSI Interrupt Vector.

49.12.39 MSI Controller Interrupt n Mask (PCIE_PL_MSICI*n*_MASK)

Offset: 0x700 + 0x12C + C*n (n=[0:7])

Address: 1FF_C000h base + 82Ch offset + (12d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_Interruptn_Mask																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_MSICI*n*_MASK field descriptions

Field	Description
MSI_Interruptn_ Mask	MSI Interrupt#n Mask

PCIE_PL_MSICI*n*_MASK field descriptions (continued)

Field	Description
	Allows enabled interrupts to be masked. If an MSI is received for a masked interrupt, the corresponding status bit gets set in the Interrupt Status Register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector.

49.12.40 MSI Controller Interrupt n Status (PCIE_PL_MSICI*n*_STATUS)

Offset: 0x700 + 0x130 + C*n (n=[0:7])

Address: 1FF_C000h base + 830h offset + (12d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_Interruptn_Status																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_MSICI*n*_STATUS field descriptions

Field	Description
MSI_Interruptn_Status	MSI Interrupt#n Status If an MSI is detected for EP#n, one bit in this register is set. The decoding of the data payload of the MSI Memory Write Request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

49.12.41 MSI Controller General Purpose IO Register (PCIE_PL_MSICGPIO)

Offset: 0x700 + 0x188

Address: 1FF_C000h base + 888h offset = 1FF_C888h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_GPIO_Register																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_MSICGPIO field descriptions

Field	Description
MSI_GPIO_Register	MSI GPIO Register The contents of this register drives the top-level output msi_ctrl_io[31:0]

49.12.42 iATU Viewport Register (PCIE_PL_iATUVR)

See Internal Address Translation (iATU) for more information on iATU operation.

The iATU registers are programmed through an index (Viewport) register to reduce the footprint in the PCI Express Extended Configuration Space. The size of the required port logic space does not depend on the number of regions defined as the Viewport register is used to select which memory region is being accessed. There are 28 bytes of register space implemented *per address region* per direction. The number of address regions that are remapped by the iATU is 4 for inbound and 4 for outbound. However, only 32 bytes of the PCIe Extended Configuration Space Address Map is used.

Offset: 0x700 + 0x200

The viewport register has a "Region Direction" bit to determine whether an inbound or outbound region is being accessed and a "Region Index" field to determine which region to program/read when accessing the other address translation registers in the iATU Register Map below..

As an example, to access the Control, Base, Limit and Target registers for Outbound region number 4:

- Write 0x00000004 to Address {0x700 + 0x200} to index the Outbound Address Region #4
- Then proceed to write to any of the other registers in the iATU Register Map below.

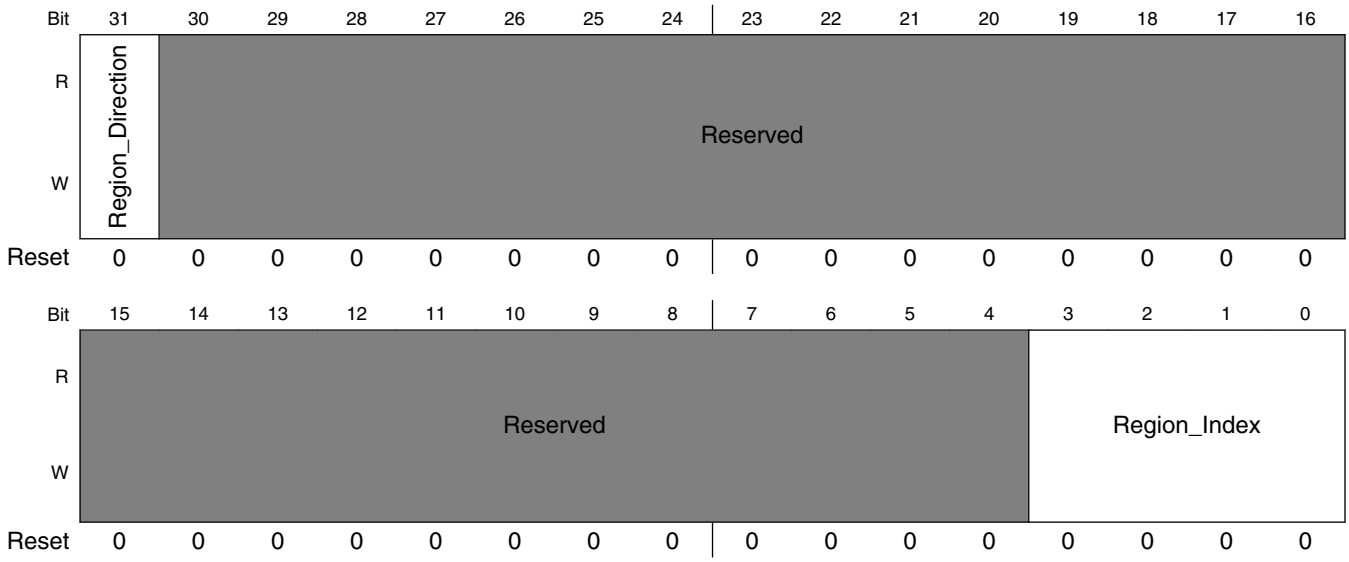
Table 49-64. iATU Register Map

Byte Offset	Description
+0x200	iATU Viewport Register
+0x204	iATU Region Control 1 Register
+0x208	iATU Region Control 2 Register
+0x20C	iATU Region Lower Base Address Register
+0x210	iATU Region Upper Base Address Register
+0x214	iATU Region Limit Address Register
+0x218	iATU Region Lower Target Address Register
+0x21C	iATU Region Upper Target Address Register

NOTE

Since AXI core is async to the core_clk, the iATU registers may not be updated while operations are in progress on the AXI Bridge Slave interface.

Address: 1FF_C000h base + 900h offset = 1FF_C900h



PCIE_PL_iATUVR field descriptions

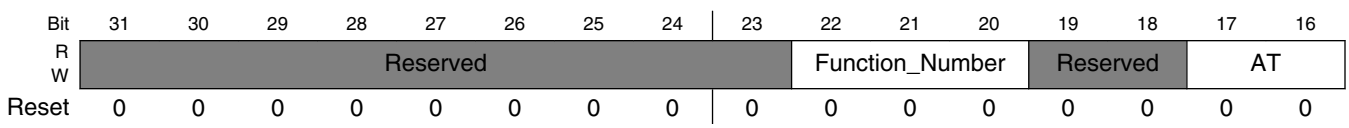
Field	Description
31 Region_Direction	Region Direction Defines the region being accessed as either 0 Outbound 1 Inbound
30–4 -	This field is reserved. Reserved
Region_Index	Region Index Defines which region is being accessed when writing to the control, base, limit and target registers. Must not be set to a number greater than CX_ATU_NUM_OUTBOUND_REGIONS - 1 when an outbound region is being accessed. Must not be set to a value greater than 3 since there are 4 regions for both inbound or outbound (4 each).

49.12.43 iATU Region Control 1 Register (PCIE_PL_iATURC1)

Offset: 0x700 + 0x204

footnote 1 - If all other enabled field-matches are successful

Address: 1FF_C000h base + 904h offset = 1FF_C904h



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					ATTR		TD	TC			TYPE				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_iATURC1 field descriptions

Field	Description
31–23 -	This field is reserved. Reserved
22–20 Function_ Number	Function Number **Outbound**: When the address of an outbound TLP is matched to this region, then the function number used in generating the 'Function' part of the Requester ID (RID) field of the TLP is taken from this 3-bit register. The value in this register must be 0x0 **Inbound MEM/IO**: When the Address and BAR matching logic in the core indicate that a MEM/IO transaction matches a BAR in the function corresponding to this value, then address translation will proceed[1]. This check is only performed if the ìFunction Number Match Enable bit of the ìiATU Control 2 Register is set. **Inbound CFG0/CFG1**: When the destination function number as specified in the routing ID of the TLP header matches the function, then address translation will proceed1. This check is only performed if the ìFunction Number Match Enable bit of the ìiATU Control 2 Register is set.
19–18 -	This field is reserved. Reserved
17–16 AT	AT **Outbound**: When the address of an outbound TLP is matched to this region, then the AT field of the TLP is changed to the value in this register. Only valid when the ATS_ENABLE configuration parameter is 1. **Inbound**: When the TYPE field of an inbound TLP is matched to this value, then address translation will proceed[1]. This check is only performed if the “AT Match Enable” bit of the “iATU Control 2 Register” is set. Only valid when the ATS_ENABLE configuration parameter is 1.
15–11 -	This field is reserved. Reserved
10–9 ATTR	ATTR **Outbound**: When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. **Inbound**: When the ATTR field of an inbound TLP is matched to this value, then address translation will proceed[1]. This check is only performed if the ìATTR Match Enable bit of the ìiATU Control 2 Register is set.
8 TD	TD **Outbound**: When the address of an outbound TLP is matched to this region, then the TD field of the TLP is changed to the value in this register. **Inbound**: When the TD field of an inbound TLP is matched to this value, then address translation will proceed1. This check is only performed if the ìTD Match Enable bit of the ìiATU Control 2 Register is set.
7–5 TC	TC **Outbound**: When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. **Inbound**: When the TC field of an inbound TLP is matched to this value, then address translation will proceed[1]. This check is only performed if the ìTC Match Enable bit of the ìiATU Control 2 Register is set.

Table continues on the next page...

PCIE_PL_iATURC1 field descriptions (continued)

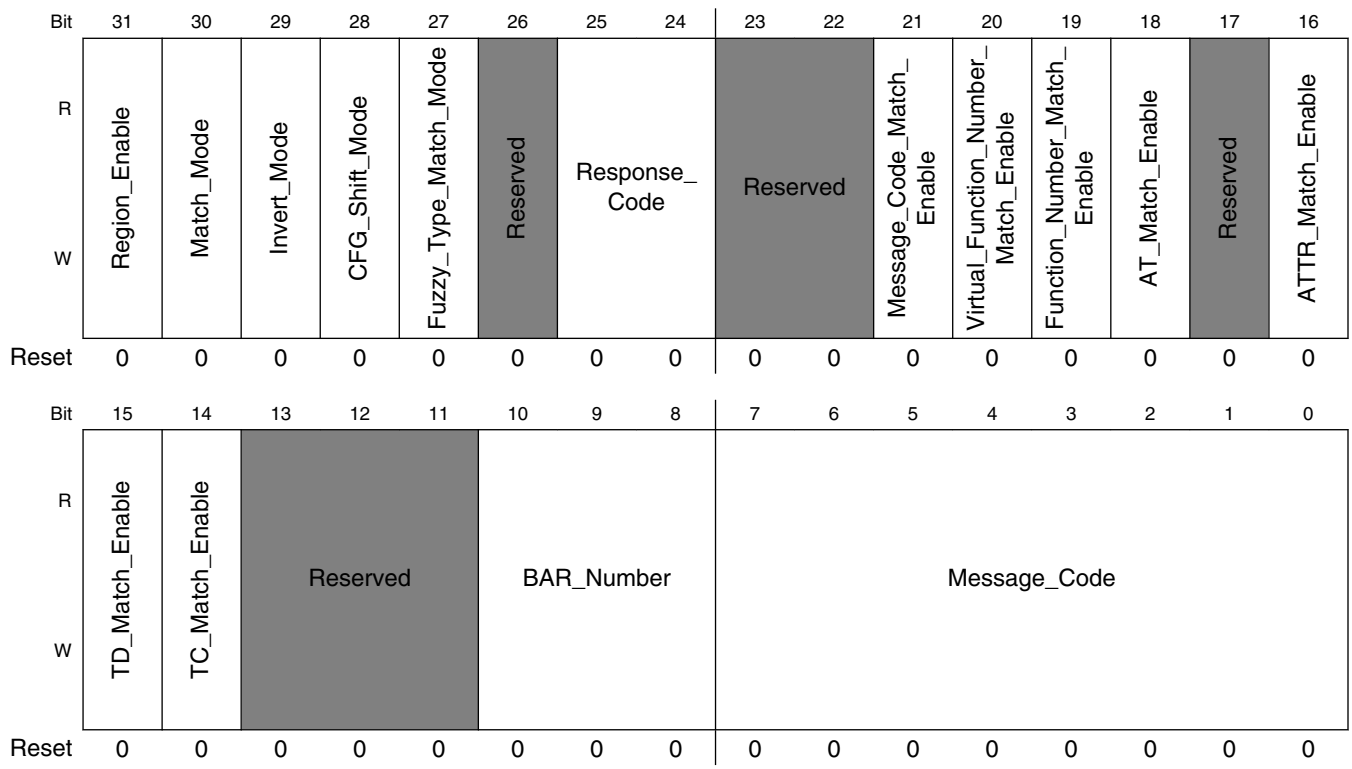
Field	Description
TYPE	TYPE **Outbound**: When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. **Inbound**: When the TYPE field of an inbound TLP is matched to this value, then address translation will proceed[2].

1. 1
2. 1

49.12.44 iATU Region Control 2 Register (PCIE_PL_iATURC2)

Offset: 0x700 + 0x208

Address: 1FF_C000h base + 908h offset = 1FF_C908h



PCIE_PL_iATURC2 field descriptions

Field	Description
31 Region_Enable	Region Enable **Outbound / Inbound**: This bit must be set to '1' for address translation to take place.

Table continues on the next page...

PCIE_PL_iATURC2 field descriptions (continued)

Field	Description
30 Match_Mode	Match Mode **Outbound**: Not used. **Inbound MEM/IO**: Determines Inbound matching mode for MEM/IO TLPs. • **0**: Address Mode. The iATU operates using addresses as in the Outbound direction. The Region Base and Limit Registers must be setup. • **1**: BAR Mode. BAR matching is used. The 'BAR Number' field is relevant. **Inbound CFG0**: Determines Inbound matching mode for CFG0 TLPs. • **0**: Routing ID match mode. The iATU interprets the Routing ID (Bytes 8 to 11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM/IO transactions. The Routing ID of the TLP must be within the base and limit of the iATU region for matching to proceed. • **1**: Accept Mode. The iATU accepts all CFG0 transactions as address matches. The routing ID in the CFG0 TLP is ignored. This is useful as all received CFG0 TLPs should be processed regardless of the Bus number. **Inbound MSG/MSGD**: Determines Inbound matching mode for MSG/MSGD TLPs. • **0**: Address Mode. The iATU treats the 3rd DWORD and 4th DWORD of the inbound MSG/MSGD TLP as an address and it is matched against the Region Base and Limit Registers. • **1**: Vendor ID match mode. This mode is relevant for ID-routed Vendor Defined Messages. The iATU ignores the Routing ID(Bus, Device, Function) in bits [31:16] of the 3rd DWORD of the TLP header, but matches against the Vendor ID in bits [15:0] of the 3rd DWORD of the TLP header. Bits [15:0] of the Region Upper Base register should be programmed with the required Vendor ID. The lower Base and Limit Register should be programmed to translate TLPs based on vendor specific information in the 4th DWORD of the TLP header.
29 Invert_Mode	Invert Mode **Outbound / Inbound**: When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address).
28 CFG_Shift_Mode	CFG Shift Mode This is useful for CFG transactions where the PCIe configuration mechanism maps bits [27:12] of the address to the bus/device and function number. This allows a CFG configuration space to be located in any 256MB window of the application memory space using a 28-bit effective address. **Outbound**: Shifts bits [27:12] of the untranslated address to form bits [31:16] of the translated address. **Inbound**: Shifts bits [31:16] of the untranslated address to form bits [27:12] of the translated address.
27 Fuzzy_Type_ Match_Mode	Fuzzy Type Match Mode **Outbound**: Not used. **Inbound**: When enabled, the iATU relaxes the matching of the TLP TYPE field against the expected TYPE field so that • CfgRd0 and CfgRd1 TLPs are seen as identical. Similarly with CfgWr0 and CfgWr1. • MRd and MRdLk TLPs are seen as identical • The Routing field of Msg/MsgD TLPs is ignored For example, CFG0 in the TYPE field in the iATU Region Control 1 Register (PCIE_PL_iATURC1) will match against an inbound CfgRd0, CfgRd1, CfgWr0 or CfgWr1 TLP.
26 -	This field is reserved. Reserved

Table continues on the next page...

PCIE_PL_iATURC2 field descriptions (continued)

Field	Description
25–24 Response_Code	Response Code **Outbound**: Not used. **Inbound**: Defines the type of response to give for accesses matching this region. This overrides the normal RADM filter response. 00 - Normal RADM filter response is used. 01 - Unsupported Request (UR) 10 - Completer Abort (CA) 11 - Not used / undefined / reserved.
23–22 -	This field is reserved. Reserved
21 Message_Code_Match_Enable	Message Code Match Enable **Outbound**: Not used. **Inbound**: Ensures that a successful Message Code TLP field comparison match occurs in MSG transactions (see Message Code field of the iATU Control 1 Register in PCIe CTRL Memory Map/Register Definition) for address translation to proceed.
20 Virtual_Function_Number_Match_Enable	Virtual Function Number Match Enable **Outbound**: Not used. **Inbound**: Ensures that a successful Virtual Function Number TLP field comparison match (see Virtual Function Number field of the iATU Control 1 Register in PCIe CTRL Memory Map/Register Definition occurs (in **MEM/IO** transactions) for address translation to proceed. **Note**: This bit must not be set at the same time as 'Function Number Match Enable'.
19 Function_Number_Match_Enable	Function Number Match Enable **Outbound**: Not used. **Inbound**: Ensures that a successful Function Number TLP field comparison match (see Function Number field of the iATU Control 1 Register in PCIe CTRL Memory Map/Register Definition) occurs (in **MEM/IO** and **CFG0/CFG1** transactions) for address translation to proceed. **Note**: This bit must not be set at the same time as 'Virtual Function Number Match Enable'.
18 AT_Match_Enable	AT Match Enable **Outbound**: Not used. **Inbound**: Ensures that a successful AT TLP field comparison match (see AT field of the iATU Control 1 Register in PCIe CTRL Memory Map/Register Definition) occurs for address translation to proceed. Only valid when the ATS_RX_ENABLE configuration parameter is 1.
17 -	This field is reserved. Reserved
16 ATTR_Match_Enable	ATTR Match Enable **Outbound**: Not used. **Inbound**: Ensures that a successful ATTR TLP field comparison match (see ATTR field of the PCIe CTRL Memory Map/Register Definition) occurs for address translation to proceed.
15 TD_Match_Enable	TD Match Enable **Outbound**: Not used. **Inbound**: Ensures that a successful TD TLP field comparison match (see TD field of the PCIe CTRL Memory Map/Register Definition) occurs for address translation to proceed.

Table continues on the next page...

PCIE_PL_iATURC2 field descriptions (continued)

Field	Description
14 TC_Match_ Enable	TC Match Enable **Outbound**: Not used. **Inbound**: Ensures that a successful TC TLP field comparison match (see TC field of the PCIe CTRL Memory Map/Register Definition) occurs for address translation to proceed.
13–11 -	This field is reserved. Reserved
10–8 BAR_Number	BAR Number **Outbound**: Not used. **Inbound**: When the BAR number of an inbound **MEM** or **IO** TLP - that is matched by the normal internal BAR address matching mechanism - is the same as this field, address translation will proceed[1].This check is only performed if the ìMatch Mode bit of the iATU Region Control 2 Register (PCIE_PL_iATURC2) is set. IO translation would require either 00100b or 00101b in the inbound TLP TYPE; the BAR Number set in the range 000b - 101b and that BAR configured as an IO BAR. 000b - BAR#0 001b - BAR#1 010b - BAR#2 011b - BAR#3 100b - BAR#4 101b - BAR#5 110b - ROM 111b - reserved
Message_Code	Message Code **Outbound**: When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the Message field of the TLP is changed to the value in this register. **Inbound**: When the TYPE field of an inbound **Msg/MsgD** TLP is matched to this value, then address translation will proceed[1]. This check is only performed if the ìFunction Message Code Match Enable bit of the iATU Region Control 2 Register (PCIE_PL_iATURC2) is set.

49.12.45 iATU Region Lower Base Address Register (PCIE_PL_iATURLBA)

The CX_ATU_MIN_REGION_SIZE configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(CX_ATU_MIN_REGION_SIZE) bits are zero.

Offset: 0x700 + 0x20C

Address: 1FF_C000h base + 90Ch offset = 1FF_C90Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Address_upper																Address_lower															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_iATURLBA field descriptions

Field	Description
31–16 Address_upper	Forms bits [31:16] of the start address of the address region to be translated.
Address_lower	Forms bits [15:0] of the start address of the address region to be translated. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe core.

49.12.46 iATU Region Upper Base Address Register (PCIE_PL_iATURUBA)

Offset: 0x700 + 0x210

Address: 1FF_C000h base + 910h offset = 1FF_C910h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Address																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_iATURUBA field descriptions

Field	Description
Address	**Outbound / Inbound**: Forms bits [63:32] of the start (and end) address of the address region to be translated. **Outbound**: In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect.

49.12.47 iATU Region Limit Address Register (PCIE_PL_iATURLA)

Offset: 0x700 + 0x214

Address: 1FF_C000h base + 914h offset = 1FF_C914h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Address_upper																Address_lower															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

PCIE_PL_iATURLA field descriptions

Field	Description
31–16 Address_upper	Forms bits [31:16] of the end address of the address region to be translated.
Address_lower	Forms bits [15:0] of the end address of the address region to be translated. The end address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe core.

49.12.48 iATU Region Lower Target Address Register (PCIE_PL_iATURLTA)

Offset: 0x700 + 0x218

Address: 1FF_C000h base + 918h offset = 1FF_C918h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Address_upper																Address_lower															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_iATURLTA field descriptions

Field	Description
31–16 Address_upper	Forms bits [31:16] of the of the new address of the translated region.
Address_lower	Forms bits [15:0] of the start address of the new address of the translated region. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe core.

49.12.49 iATU Region Upper Target Address Register (PCIE_PL_iATURUTA)

Offset: 0x700 + 0x21C

Address: 1FF_C000h base + 91Ch offset = 1FF_C91Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Address																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PL_iATURUTA field descriptions

Field	Description
Address	**Outbound / Inbound**: Forms bits [63:32] of the start address of the new address of the translated region. **Inbound**: In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect.

Chapter 50
PCI Express PHY (PCIe_PHY)

50.1 Overview

PCIe 2.0 PHY is a complete mixed-signal semiconductor intellectual property (IP) solution, designed for single-chip integration into computer applications.

The PCIe 2.0 PHY supports both the 5 Gbp/s data rate of the PCI Express Gen 2.0 specifications as well as being backwards compatible to the 2.5Gb/s Gen 1.1 specification.

This chapter provides an introduction to the PCIe 2.0 PHY and its features.

50.2 Applications

Designed for low power, the PCIe 2.0 PHY allows designers to introduce competitive products using the latest generation of the PCI Express standard.

50.3 PCIe2 PHY Features

50.3.1 Standards Compliance

The PCIe2 PHY is fully compliant with all of the required features of the following standards:

- PCI Express Base Specification, Revision 2.0 (including legacy 2.5-Gbps support
- 5.0 Gbps data rate
- PCI Express Base Specification, Revision 1.1
- 2.5Gbps data rate

50.3.2 PHY Features

- 5 Gbps data transmission rate
- Integrated PHY includes transmitter, receiver, PLL, digital core, and ESD.
- Programmable RX equalization
- Designed for excellent performance margin and receiver sensitivity
- Robust PHY architecture tolerates wide process, voltage and temperature variations
- Low-jitter PLL technology with excellent supply isolation
- IEEE 1149.6 (JTAG) boundary scan
- Built-in Self-Test (BIST) features for production, at-speed, testing on any digital tester
- 5Gb/s PCIe Gen 2 and 2.5Gb/s PCIe Gen 1.1 test modes supported
- Advanced built-in diagnostics including on-chip sampling scope for easy debug
- Visibility & controllability of hard macro functionality thru programmable registers in the design
- Over-rides on all ASIC side inputs for easy debug
- Access register space thru simple 16 bit parallel interface
- Access register space thru JTAG

50.4 External Signals

The following table describes the external signals of PCIE:

Table 50-1. PCIE External Signals

Signal	Description	Pad	Mode	Direction
PCIE_RX_N	PCIe negative receive signal	PCIE_RX_N	No muxing	I
PCIE_RX_P	PCIe positive receive signal	PCIE_RX_P	No muxing	I
PCIE_TX_N	PCIe negative transmit signal	PCIE_TX_N	No muxing	O
PCIE_TX_P	PCIe positive transmit signal	PCIE_TX_P	No muxing	O

50.5 Functional Description

This section provides a complete functional description of the block.

50.5.1 Clocks and Resets

50.5.1.1 Reference Clock Enables

To enable to lowest possible power state, there is an enable on the reference clock input buffer. When totally powered down and prior to removing the reference clock, can be de-asserted to completely shut down the PHY.

- To enable the reference clock buffer in the PCIe2 PHY, the signal phy_ref_ssp_en must be asserted after the reference clock is up and stable and prior to de-asserting phy_reset.

50.5.1.2 Reference Clock Frequency Selection

The MPLL in the PCIe2 PHY has the ability to multiply the reference clock by integer and non-integer values, allowing for a wide range input reference clock frequencies.

Based on the incoming reference clock frequency, the MPLL controls must be set as shown in the table below to get proper 5Gb/s operation. With these MPLL settings, the reference clock output phy_ref_output_clk will provide the frequencies shown in the right-most column the table below.

NOTE

Reference clock jitter is greater than the specification requirement when the clock source is the internal ENET PLL.

Table 50-2. Reference Clock Frequency Selection

Reference Clock (MHz)	Required Multiplication	ref_clkdiv2	mpll_multiplier[6:0]	phy_ref_output_clk
100	25	0	0011001	100 MHz
125	40	1	0101000	62.5 MHz
200	25	1	0011001	100 MHz

50.5.1.3 Spread Spectrum Clocking

The PCIe2 PHY uses the PCIe reference clock which may or may not be spread. Since the PCIe specification does not allow independently spread clocks, the ability of the PHY to add a spread to a fixed frequency reference clock must be disabled.

50.5.2 Termination Resistance Tuning

The PHY uses an external resistor to calibrate the termination impedances of the high speed inputs and outputs of the PHY.

50.5.3 Control Register Access

The CR port is a simple 16bit data/16bit address parallel port that is provided for on-chip access to the control registers inside the PCIe2 PHY.

While access to these registers is NOT required for normal operation of the PHY, this interface is included for users that want to access some of the diagnostic features of the PHY during normal operation or over-ride some of the basic PHY control signals.

This interface is completely asynchronous using a hand shake between phy_cr_cap_addr, phy_cr_cap_data, phy_cr_read, and phy_cr_write input commands with phy_cr_ack acknowledgements and phy_cr_data_out outputs from the PHY. The CR port access is broken down into Address, Read, and Write transactions.

The details of the Control Registers themselves is detailed Control Registers.

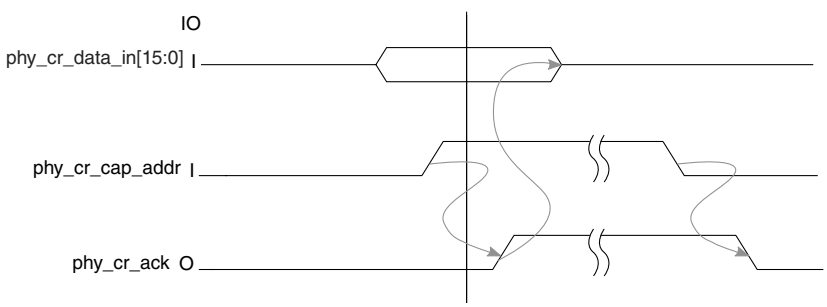


Figure 50-1. CR Bus Address Capture Transaction

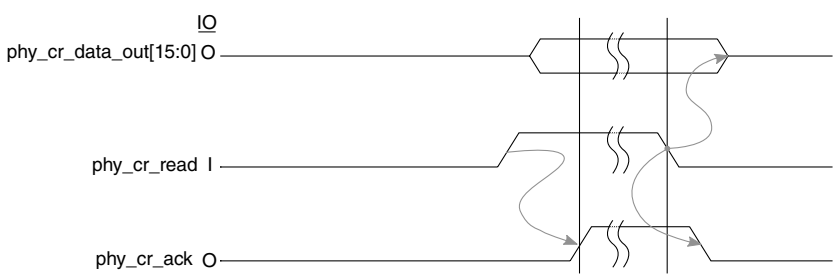


Figure 50-2. CR Bus Read Transaction

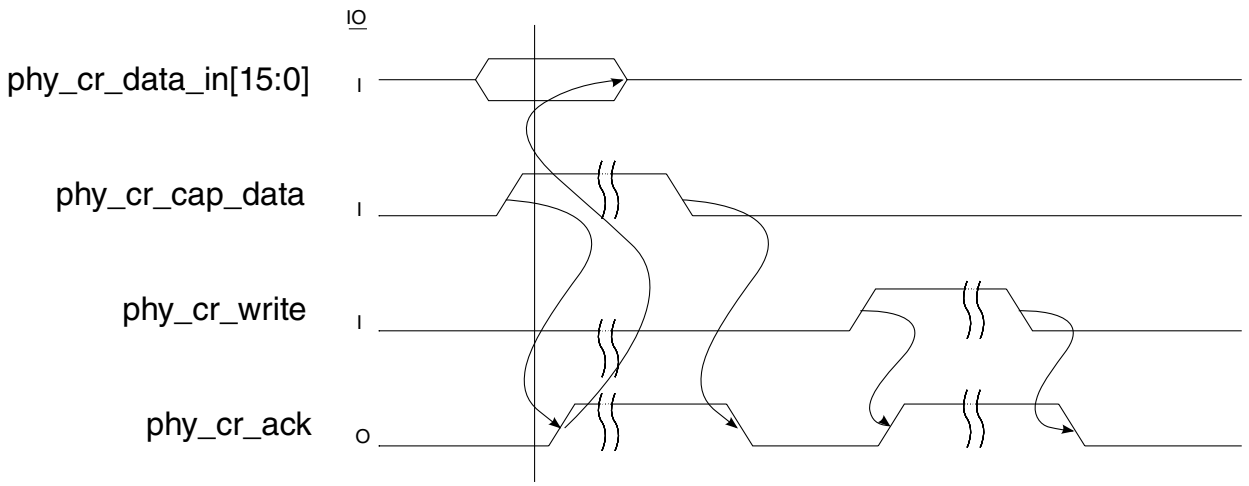


Figure 50-3. CR Bus Write Transaction

NOTE

In all cases, phy_cr_ack will de-assert when the command (phy_cr_cap_addr, phy_cr_cap_data, phy_cr_read, cr_write) is removed.

If the Control Register port is not going to be used, tie the input signals as follows,

- phy_cr_cap_addr =1'b0
- phy_cr_cap_data =1'b0
- phy_cr_data_in = 16'b0
- phy_cr_read = 1'b0
- phy_cr_write = 1'b0
- No connect (leave floating) the following:
- phy_cr_data_out[15:0]
- phy_cr_ack.

50.6 System Operation

50.6.1 Powering Up and Powering Down

50.6.1.1 Power Up Requirements

The PCIe2 PHY has three power supplies: vp, vptx0, and vph.

vp and vptx0 are both 1.1 V supplies and, while separate pins are used for noise isolation purposes, they both should be connected to the same supply on the board. The vph is the 2.5 V supply. The PCIe2 PHYcan have the 2.5 V and the 1.1 V supply power up in any sequence as long as the low supply comes up within tens of milliseconds after the high supply comes up.

50.6.1.2 Power-On Reset

While the power supplies can be brought up in any sequence as long as they come up at relatively the same time, it is the responsibility of the ASIC to hold the phy_reset pin high until the supplies are within the specified range in Table 6-1 and the reference clock is up and stable.

The timing of Power-on-Reset is shown in the figure below.

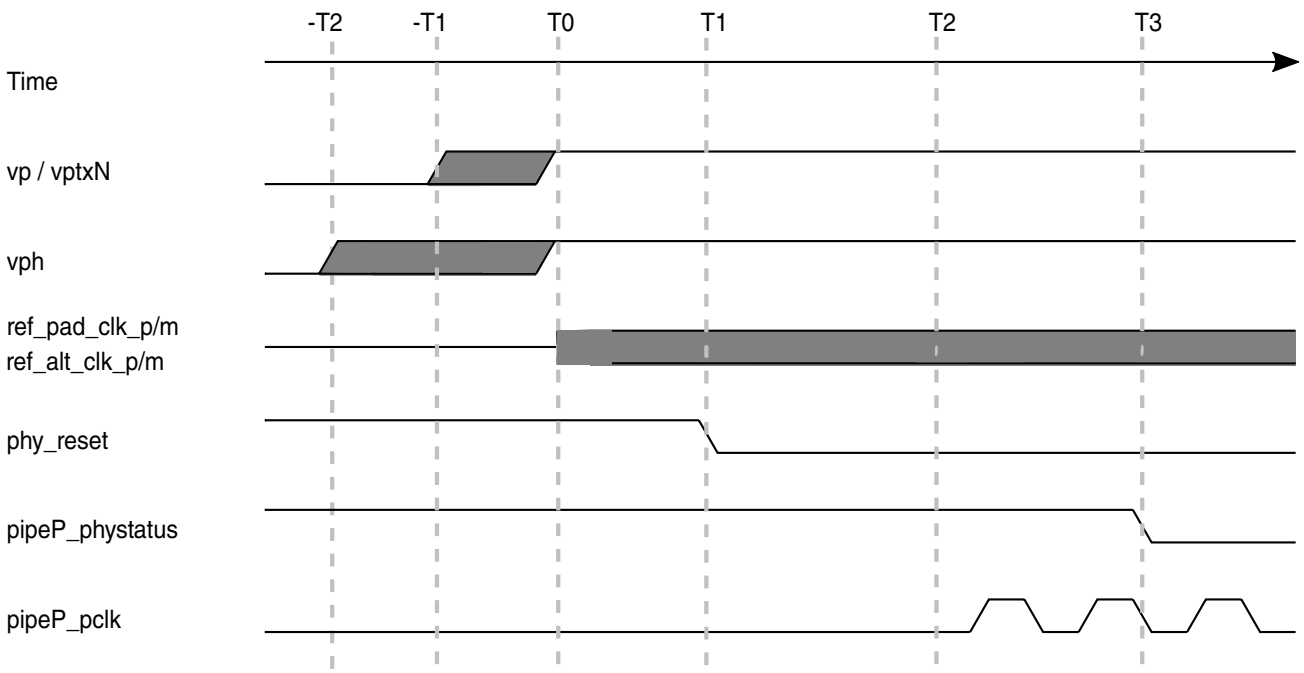


Figure 50-4. POR Timing

Table 50-3. TSMC 40LP 1.1/2.5-V

Timing Parameter	Description	Value
RefClk	Differential reference clock input into the PHY. It will be equal to pins ref_alt_clk_p, ref_alt_clk_m or ref_pad_clk_m, ref_pad_clk_p depending on the setting of ref_use_pad	-
-T2 to -T1	Maximum time vph is applied before or after vp / vptxN	< 10's of ms

Table continues on the next page...

Table 50-3. TSMC 40LP 1.1/2.5-V (continued)

Timing Parameter	Description	Value
T0	Time when supplies are within specified limits and the reference clock is up and stable	-
T1	Time after T0 when ASIC can remove phy_reset	> 10 µs
T3 to T1	Time after T1 when pipeP_status de-asserts to indicate that pipeP_pclk is guaranteed to be up and stable	> 165 µs

50.6.1.3 Power Supply Sequencing when the PCIe 2.0 PHY is Not Used

When the PCIe2 PHY is not being used, either both supplies can be left floating and all outputs of the PHY ignored, or power up both supplies and assert the phy_test_powerdown signal (1'b1) to put the PHY in its lowest power state.

50.6.1.4 Power Down Requirements

There are no power down requirements.

50.7 Control Memory Map/Register Definition

NOTE

PCIe PHY registers are only accessible by the corresponding controller (PCIE_PHY_CTRL_R and PCIE_PHY_STS_R) or in debug through the JTAG port. PCIe PHY is not memory mapped to processor address space, so the absolute addresses shown is the relative address and is not valid.

PCIE_PHY memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Register ID Low 16 bits (PCIE_PHY_IDCODE_LO)	16	R	0000h	50.7.1/4461
1	Register ID High 16 bits (PCIE_PHY_IDCODE_HI)	16	R	0000h	50.7.2/4461
2	Debug Register (PCIE_PHY_DEBUG)	16	R/W	000Ah	50.7.3/4462
3	Debug Register (PCIE_PHY_RTUNE_DEBUG)	16	R/W	0000h	50.7.4/4462

Table continues on the next page...

PCIE_PHY memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4	PCIE_PHY_RTUNE_STAT	16	R/W	0000h	50.7.5/4463
5	PCIE_PHY_SS_PHASE	16	R/W	0000h	50.7.6/4463
6	PCIE_PHY_SS_FREQ	16	R/W	3327h	50.7.7/4464
10	PCIE_PHY_ATEOVRD	16	R/W	0000h	50.7.8/4464
11	PCIE_PHY_MPLL_OVRD_IN_LO	16	R/W	004Ch	50.7.9/4465
11	PCIE_PHY_MPLL_OVRD_IN_HI	16	R/W	004Ch	50.7.10/ 4466
13	PCIE_PHY_SSC_OVRD_IN	16	R/W	0000h	50.7.11/ 4467
14	PCIE_PHY_BS_OVRD_IN	16	R/W	0000h	50.7.12/ 4468
15	PCIE_PHY_LEVEL_OVRD_IN	16	R/W	0000h	50.7.13/ 4469
16	PCIE_PHY_SUP_OVRD_OUT	16	R/W	0101h	50.7.14/ 4469
17	PCIE_PHY_MPLL_ASIC_IN	16	R	0000h	50.7.15/ 4470
18	PCIE_PHY_BS_ASIC_IN	16	R	0000h	50.7.16/ 4471
19	PCIE_PHY_LEVEL_ASIC_IN	16	R	0000h	50.7.17/ 4472
1A	PCIE_PHY_SSC_ASIC_IN	16	R	0000h	50.7.18/ 4473
1B	PCIE_PHY_SUP_ASIC_OUT	16	R	0000h	50.7.19/ 4473
1C	PCIE_PHY_ATEOVRD_STATUS	16	R	0000h	50.7.20/ 4474
20	PCIE_PHY_SCOPE_ENABLES	16	R/W	0000h	50.7.21/ 4475
21	PCIE_PHY_SCOPE_SAMPLES	16	R/W	0100h	50.7.22/ 4476
22	PCIE_PHY_SCOPE_COUNT	16	R/W	FFFFh	50.7.23/ 4476
23	PCIE_PHY_SCOPE_CTL	16	R/W	0000h	50.7.24/ 4477
24	PCIE_PHY_SCOPE_MASK_000	16	R/W	0000h	50.7.25/ 4477
25	PCIE_PHY_SCOPE_MASK_001	16	R/W	0000h	50.7.25/ 4477
26	PCIE_PHY_SCOPE_MASK_010	16	R/W	0000h	50.7.25/ 4477
27	PCIE_PHY_SCOPE_MASK_011	16	R/W	0000h	50.7.25/ 4477

Table continues on the next page...

PCIE_PHY memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
28	PCIE_PHY_SCOPE_MASK_100	16	R/W	0000h	50.7.25/ 4477
29	PCIE_PHY_SCOPE_MASK_101	16	R/W	0000h	50.7.25/ 4477
2A	PCIE_PHY_SCOPE_MASK_110	16	R/W	0000h	50.7.25/ 4477
2B	PCIE_PHY_SCOPE_MASK_111	16	R/W	0000h	50.7.25/ 4477
30	PCIE_PHY_MPLL_LOOP_CTL	16	R/W	00C0h	50.7.26/ 4478
32	PCIE_PHY_MPLL_ATB_MEAS2	16	R/W	0000h	50.7.27/ 4478
33	PCIE_PHY_MPLL_OVR	16	R/W	0000h	50.7.28/ 4479
34	PCIE_PHY_RTUNE_RTUNE_CTRL	16	R/W	0000h	50.7.29/ 4480
1000	PCIE_PHY_TX_OVRD_IN_LO	16	R/W	0000h	50.7.30/ 4481
1001	PCIE_PHY_TX_OVRD_IN_HI	16	R/W	0000h	50.7.31/ 4483
1003	PCIE_PHY_TX_OVRD_DRV_LO	16	R/W	0000h	50.7.32/ 4484
1004	PCIE_PHY_TX_OVRD_OUT	16	R/W	0000h	50.7.33/ 4484
1005	PCIE_PHY_RX_OVRD_IN_LO	16	R/W	0000h	50.7.34/ 4485
1006	PCIE_PHY_RX_OVRD_IN_HI	16	R/W	0000h	50.7.35/ 4486
1007	PCIE_PHY_RX_OVRD_OUT	16	R/W	0000h	50.7.36/ 4487
1008	PCIE_PHY_TX_ASIC_IN	16	R	0000h	50.7.37/ 4488
1009	PCIE_PHY_TX_ASIC_DRV_LO	16	R	0000h	50.7.38/ 4489
100A	PCIE_PHY_TX_ASIC_DRV_HI	16	R	0000h	50.7.39/ 4490
100B	PCIE_PHY_TX_ASIC_OUT	16	R	0000h	50.7.40/ 4490
100C	PCIE_PHY_RX_ASIC_IN	16	R	0000h	50.7.41/ 4491
100D	PCIE_PHY_RX_ASIC_OUT	16	R	0000h	50.7.42/ 4492
1011	PCIE_PHY_TX_VMD_FSM_TX_VCM_0	16	R	0000h	50.7.43/ 4493

Table continues on the next page...

PCIE_PHY memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1012	PCIE_PHY_TX_VMD_FSM_TX_VCM_1	16	R	0000h	50.7.44/ 4493
1013	PCIE_PHY_TX_VMD_FSM_TX_VCM_DEBUG_IN	16	R/W	0000h	50.7.45/ 4494
1014	PCIE_PHY_TX_VMD_FSM_TX_VCM_DEBUG_OUT	16	R	0000h	50.7.46/ 4495
1015	PCIE_PHY_TX_LBERT_CTL	16	R/W	0000h	50.7.47/ 4495
1016	PCIE_PHY_RX_LBERT_CTL	16	R/W	0000h	50.7.48/ 4496
1017	PCIE_PHY_RX_LBERT_ERR	16	R/W	0000h	50.7.49/ 4497
1018	PCIE_PHY_RX_SCOPE_CTL	16	R/W	0000h	50.7.50/ 4497
1019	PCIE_PHY_RX_SCOPE_PHASE	16	R/W	0000h	50.7.51/ 4498
101A	PCIE_PHY_RX_DPLL_FREQ	16	R/W	0000h	50.7.52/ 4498
101B	PCIE_PHY_RX_CDR_CTL	16	R/W	000Fh	50.7.53/ 4499
101C	PCIE_PHY_RX_CDR_CDR_FSM_DEBUG	16	R	0000h	50.7.54/ 4500
101D	PCIE_PHY_RX_CDR_LOCK_VEC_OVRD	16	R/W	8000h	50.7.55/ 4501
101E	PCIE_PHY_RX_CDR_LOCK_VEC	16	R	0000h	50.7.56/ 4502
101F	PCIE_PHY_RX_CDR_ADAP_FSM	16	R	0000h	50.7.57/ 4502
1020	PCIE_PHY_RX_ATB0	16	R/W	0000h	50.7.58/ 4503
1021	PCIE_PHY_RX_ATB1	16	R/W	0000h	50.7.59/ 4504
1022	PCIE_PHY_RX_ENPWR0	16	R/W	0000h	50.7.60/ 4504
1023	PCIE_PHY_RX_PMIX_PHASE	16	R/W	0000h	50.7.61/ 4505
1024	PCIE_PHY_RX_ENPWR1	16	R/W	0000h	50.7.62/ 4506
1025	PCIE_PHY_RX_ENPWR2	16	R/W	0000h	50.7.63/ 4507
1026	PCIE_PHY_RX_SCOPE	16	R/W	0000h	50.7.64/ 4508
102B	PCIE_PHY_TX_TXDRV_CNTRL	16	R/W	0000h	50.7.65/ 4509

Table continues on the next page...

PCIE_PHY memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
102C	PCIE_PHY_TX_POWER_CTL	16	R/W	0000h	50.7.66/ 4510
102D	PCIE_PHY_TX_ALT_BLOCK	16	R/W	0000h	50.7.67/ 4511
102E	PCIE_PHY_TX_ALT_AND_LOOPBACK	16	R/W	0000h	50.7.68/ 4512
102F	PCIE_PHY_TX_TX_ATB_REG	16	R/W	0000h	50.7.69/ 4513

50.7.1 Register ID Low 16 bits (PCIE_PHY_IDCODE_LO)

Address: 0h base + 0h offset = 0h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	IDCODE_LO															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_IDCODE_LO field descriptions

Field	Description
IDCODE_LO	Data

50.7.2 Register ID High 16 bits (PCIE_PHY_IDCODE_HI)

Address: 0h base + 1h offset = 1h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	IDCODE_HI															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_IDCODE_HI field descriptions

Field	Description
IDCODE_HI	Data

50.7.3 Debug Register (PCIE_PHY_DEBUG)

Address: 0h base + 2h offset = 2h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	Reserved										DTB_SEL		TX_VREF_SEL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

PCIE_PHY_DEBUG field descriptions

Field	Description
15–6 -	This field is reserved. Reserved
6–5 DTB_SEL	Description: The lane DTB's are ORed together with the support DTB signals selected with the following encodings. 00 None 01 reset_ctl DTB output 10 Scope DTB output 11 rtune DTB output
TX_VREF_SEL	-

50.7.4 Debug Register (PCIE_PHY_RTUNE_DEBUG)

Address: 0h base + 3h offset = 3h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved	VALUE						
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	VALUE			TYPE		SET_VAL	MAN_TUNE	FLIP_COMP
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RTUNE_DEBUG field descriptions

Field	Description
15 -	This field is reserved. Reserved
14–5 VALUE	Value to use when triggering SET_VAL field. Only the 6 LSB's are used when setting Rx cal or Tx cal values.
4–3 TYPE	Type of manual tuning or register read/write to execute. 00 ADC, or read/write rt_value 01 Rx tune, or read/write rx_cal_val (only 6 bits)

Table continues on the next page...

PCIE_PHY_RTUNE_DEBUG field descriptions (continued)

Field	Description
	10 Tx tune, or read/write tx_cal_val (only 6 bits) 11 Resref detect (no affect when triggering SET_VAL fi
2 SET_VAL	Sets value. Write a 1 to manually write the register specified by the TYPE field to the value in the VALUE field.
1 MAN_TUNE	Write a 1 to perform a manual tuning specified by the TYPE field. Starting a manual tune while a tune is currently running can cause unpredictable results. For use only when you know what the part is doing (with respect to resistor tuning). **NOTE:** Write a 1 to perform an operation. Subsequent writes with the bit set will trigger the operation. No need to clear (0) the bit between writes.
0 FLIP_COMP	Inverts Analog Comparator Output.

50.7.5 PCIE_PHY_RTUNE_STAT

Address: 0h base + 4h offset = 4h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Reserved								STAT							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_RTUNE_STAT field descriptions

Field	Description
15–10 -	This field is reserved. Reserved
STAT	Current value of the register specifed by the RTUNE_DEBUG[TYPE] field.

50.7.6 PCIE_PHY_SS_PHASE

Address: 0h base + 5h offset = 5h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	Reserved								DTHR							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_SS_PHASE field descriptions

Field	Description
15–10 -	This field is reserved. Reserved
DTHR	Current value of the register specifed by the RTUNE_DEBUG[TYPE] field.

50.7.7 PCIE_PHY_SS_FREQ

Address: 0h base + 6h offset = 6h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved	FREQ_ OVRD	FREQ_PK					
Reset	0	0	1	1	0	0	1	1

Bit	7	6	5	4	3	2	1	0
Read Write	FREQ_PK	FREQ_CNT_INIT						
Reset	0	0	1	0	0	1	1	1

PCIE_PHY_SS_FREQ field descriptions

Field	Description
15 -	This field is reserved. Reserved
14 FREQ_OVRD	Frequency register override. Spread spectrum clocking must be enabled to read from or write to this register. **NOTE:** Must be set for PHASE writes to stick.
13–7 FREQ_PK	Peak frequency value (for changing direction). Spread spectrum clocking must be enabled to read from or write to this register.
FREQ_CNT_INIT	Initial frequency counter value. Spread spectrum clocking must be enabled to read from or write to this register.

50.7.8 PCIE_PHY_ATEOVRD

Address: 0h base + 10h offset = 10h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				ateovrd_en	ref_usb2_en	ref_clkdiv2	Reserved
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_ATEOVRD field descriptions

Field	Description
15–4 -	This field is reserved. Reserved
3 ateovrd_en	Override enable for ATE signals.

Table continues on the next page...

PCIE_PHY_ATEOVRD field descriptions (continued)

Field	Description
2 ref_usb2_en	Override value for HSPHY ref_clk enable.
1 ref_clkdiv2	Override value for SSP ref_clk prescaler.
0 -	This field is reserved. Reserved

50.7.9 PCIE_PHY_MPLL_OVRD_IN_LO

Address: 0h base + 11h offset = 11h

Bit	15	14	13	12	11	10	9	8
Read Write	RES_ACK_IN_OVRD	RES_ACK_IN	RES_REQ_IN_OVRD	RES_REQ_IN	RTUNE_REQ_OVRD	RTUNE_REQ	MPLL_MULTIPLIER_OVRD	MPLL_MULTIPLIER
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	MPLL_MULTIPLIER						MPLL_EN_OVRD	MPLL_EN
Reset	0	1	0	0	1	1	0	0

PCIE_PHY_MPLL_OVRD_IN_LO field descriptions

Field	Description
15 RES_ACK_IN_OVRD	Override enable for res_ack_in.
14 RES_ACK_IN	Override value for res_ack_in.
13 RES_REQ_IN_OVRD	Override enable for res_req_in.
12 RES_REQ_IN	Override value for res_req_in.
11 RTUNE_REQ_OVRD	Override enable for rtune_req.
10 RTUNE_REQ	Override value for rtune_req.
9 MPLL_MULTIPLIER_OVRD	Override enable for mpll_multiplier.

Table continues on the next page...

PCIE_PHY_MPLL_OVRD_IN_LO field descriptions (continued)

Field	Description
8–2 MPLL_ MULTIPLIER	Override value for mpll_multiplier.
1 MPLL_EN_ OVRD	Override enable for mpll_en.
0 MPLL_EN	Override value for mpll_en.

50.7.10 PCIE_PHY_MPLL_OVRD_IN_HI

Address: 0h base + 11h offset = 11h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved					MPLL_RST	FSEL_OVR	FSEL
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	FSEL		MPLL_ WORD_ CLK_EN_ OVRD	MPLL_ WORD_ CLK_EN	MPLL_ DWORD_ CLK_EN_ OVRD	MPLL_ DWORD_ CLK_EN	MPLL_ QWORD_ CLK_EN_ OVRD	MPLL_ QWORD_ CLK_EN
Reset	0	1	0	0	1	1	0	0

PCIE_PHY_MPLL_OVRD_IN_HI field descriptions

Field	Description
15–11 -	This field is reserved. Reserved.
10 MPLL_RST	Resets the MPLL state machine. Writing the register with this bit set will reset the MPLL power-up/down FSM, regardless of the current state of the register bit.
9 FSEL_OVR	Override enable for fsel[2:0].
8–6 FSEL	: Override value for fsel[2:0].
5 MPLL_WORD_ CLK_EN_OVRD	Override enable for mpll_word_clk_en.
4 MPLL_WORD_ CLK_EN	Override value for mpll_word_clk_en.
3 MPLL_DWORD_ CLK_EN_OVRD	Override enable for mpll_dword_clk_en.

Table continues on the next page...

PCIE_PHY_MPLL_OVRD_IN_HI field descriptions (continued)

Field	Description
2 MPLL_DWORD_CLK_EN	Override value for mpll_dword_clk_en.
1 MPLL_QWORD_CLK_EN_OVRD	Override enable for mpll_qword_clk_en.
0 MPLL_QWORD_CLK_EN	Override value for mpll_qword_clk_en.

50.7.11 PCIE_PHY_SSC_OVRD_IN

Address: 0h base + 13h offset = 13h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Reserved				SSC_OVRD_IN_EN	SSC_EN	SSC_RANGE		SSC_REF_CLK_SEL							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_SSC_OVRD_IN field descriptions

Field	Description
15–12 -	This field is reserved. Reserved
11 SSC_OVRD_IN_EN	Override enable for Spread Spectrum generator.
10 SSC_EN	Override value for SSC enable.
9–8 SSC_RANGE	Override value for SSC modulation range.
SSC_REF_CLK_SEL	Override value for reference clock scaling.

50.7.12 PCIE_PHY_BS_OVRD_IN

Address: 0h base + 14h offset = 14h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved				EN	INVERT	INIT	HIGHZ
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	CLAMP	EXTEST_ AC	EXTEST	PRELOAD	UPDATE_ DR	CAPTURE_ DR	SHIFT_DR	IN
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_BS_OVRD_IN field descriptions

Field	Description
15–12 -	This field is reserved. Reserved.
11 EN	Enables override values for all inputs controlled by this register.
10 INVERT	Override value for bs_invert.
9 INIT	Override value for bs_init.
8 HIGHZ	Override value for bs_highz.
7 CLAMP	Override value for bs_clamp.
6 EXTEST_AC	Override value for bs_extest_ac.
5 EXTEST	Override value for bs_extest.
4 PRELOAD	Override value for bs_preload.
3 UPDATE_DR	Override value for bs_update_dr.
2 CAPTURE_DR	Override value for bs_capture_dr
1 SHIFT_DR	Override value for bs_shift_dr.
0 IN	Override value for bs_shift_dr.

50.7.13 PCIE_PHY_LEVEL_OVRD_IN

Address: 0h base + 15h offset = 15h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	Reserved					EN	ACJT_LEVEL					LOS_LEVEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_LEVEL_OVRD_IN field descriptions

Field	Description
15–11 -	This field is reserved. Reserved.
10 EN	Enables override values for all inputs controlled by this register.
9–5 ACJT_LEVEL	Override value for acjt_level.
LOS_LEVEL	Override value for los_level.

50.7.14 PCIE_PHY_SUP_OVRD_OUT

Address: 0h base + 16h offset = 16h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved						MPLL_ STATE_ OVRD	MPLL_ STATE
Reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Read Write	BS_OUT_ OVRD	BS_OUT	RTUNE_ ACK_OVRD	RTUNE_ ACK	RES_REQ_ OUT_OVRD	RES_REQ_ OUT	RES_ACK_ OUT_OVRD	RES_ACK_ OUT
Reset	0	0	0	0	0	0	0	1

PCIE_PHY_SUP_OVRD_OUT field descriptions

Field	Description
15–10 -	This field is reserved. Reserved.
9 MPLL_STATE_ OVRD	Override enable for mpll_state output.
8 MPLL_STATE	Override value for mpll_state output.
7 BS_OUT_OVRD	Override enable for bs_out output.

Table continues on the next page...

PCIE_PHY_SUP_OVRD_OUT field descriptions (continued)

Field	Description
6 BS_OUT	Override value for bs_out output.
5 RTUNE_ACK_ OVRD	Override enable for rtune_ack output.
4 RTUNE_ACK	Override value for rtune_ack output.
3 RES_REQ_ OUT_OVRD	Override enable for res_req_out output.
2 RES_REQ_OUT	Override value for res_req_out output.
1 RES_ACK_OUT_ OVRD	Override enable for res_ack_out output.
0 RES_ACK_OUT	Override value for res_ack_out output.

50.7.15 PCIE_PHY_MPLL_ASIC_IN

Address: 0h base + 17h offset = 17h

Bit	15	14	13	12	11	10	9	8
Read	Reserved		MPLL_ WORD_ CLK_EN	MPLL_ DWORD_ CLK_EN	MPLL_ QWORD_ CLK_EN	RES_ACK_ IN	RES_REQ_ IN	RTUNE_ REQ
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	MPLL_MULTIPLIER							MPLL_EN
Write								
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_MPLL_ASIC_IN field descriptions

Field	Description
15–14 -	This field is reserved. Reserved.
13 MPLL_WORD_ CLK_EN	Value from ASIC for mpll_word_clk_en.

Table continues on the next page...

PCIE_PHY_MPLL_ASIC_IN field descriptions (continued)

Field	Description
12 MPLL_DWORD_CLK_EN	Value from ASIC for mpll_dword_clk_en.
11 MPLL_QWORD_CLK_EN	Value from ASIC for mpll_qword_clk_en.
10 RES_ACK_IN	Value from ASIC for res_ack_in.
9 RES_REQ_IN	Value from ASIC for res_req_in.
8 RTUNE_REQ	Value from ASIC for rtune_req.
7–1 MPLL_MULTIPLIER	Value from ASIC for mpll_multiplier.
0 MPLL_EN	Value from ASIC for mpll_en.

50.7.16 PCIE_PHY_BS_ASIC_IN

Address: 0h base + 18h offset = 18h

Bit	15	14	13	12	11	10	9	8
Read	Reserved					INVERT	INIT	HIGHZ
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	CLAMP	EXTEST_AC	EXTEST	PRELOAD	UPDATE_DR	CAPTURE_DR	SHIFT_DR	IN
Write								
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_BS_ASIC_IN field descriptions

Field	Description
15–11 -	This field is reserved. Reserved.
10 INVERT	Value from ASIC for bs_invert.
9 INIT	Value from ASIC for bs_init.
8 HIGHZ	Value from ASIC for bs_highz.

Table continues on the next page...

PCIE_PHY_BS_ASIC_IN field descriptions (continued)

Field	Description
7 CLAMP	Value from ASIC for bs_clamp.
6 EXTEST_AC	Value from ASIC for bs_extest_ac.
5 EXTEST	Value from ASIC for bs_extest.
4 PRELOAD	Value from ASIC for bs_preload.
3 UPDATE_DR	Value from ASIC for bs_update_dr.
2 CAPTURE_DR	Value from ASIC for bs_capture_dr.
1 SHIFT_DR	Value from ASIC for bs_shift_dr.
0 IN	Value from ASIC for bs_in.

50.7.17 PCIE_PHY_LEVEL_ASIC_IN

Address: 0h base + 19h offset = 19h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Reserved						ACJT_LEVEL					LOS_LEVEL				
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_LEVEL_ASIC_IN field descriptions

Field	Description
15–10 -	This field is reserved. Reserved.
9–5 ACJT_LEVEL	Value from ASIC for acjt_level.
LOS_LEVEL	Value from ASIC for los_level.

50.7.18 PCIE_PHY_SSC_ASIC_IN

Address: 0h base + 1Ah offset = 1Ah

Bit	15	14	13	12	11	10	9	8
Read	Reserved		SS_EN	SSC_RANGE		SSC_REF_CLK_SEL		
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	SSC_REF_CLK_SEL						FSEL	
Write								
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_SSC_ASIC_IN field descriptions

Field	Description
15–14 -	This field is reserved. Reserved.
13 SS_EN	Value from ASIC for ssc_en.
12–11 SSC_RANGE	Value from ASIC for ssc_range.
10–3 SSC_REF_CLK_SEL	Value from ASIC for ssc_ref_clk_sel
FSEL	Value from ASIC for fsel.

50.7.19 PCIE_PHY_SUP_ASIC_OUT

Address: 0h base + 1Bh offset = 1Bh

Bit	15	14	13	12	11	10	9	8
Read	Reserved							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	Reserved			MPLL_STATE	BS_OUT	RTUNE_ACK	RES_REQ_OUT	RES_ACK_OUT
Write								
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_SUP_ASIC_OUT field descriptions

Field	Description
15–5 -	This field is reserved. Reserved.
4 MPLL_STATE	Value from PHY for mpll_state output.
3 BS_OUT	Value from PHY for bs_out output.
2 RTUNE_ACK	Value from PHY for rtune_ack output.
1 RES_REQ_OUT	Value from PHY for res_req_out output.
0 RES_ACK_OUT	Value from PHY for res_ack_out output.

50.7.20 PCIE_PHY_ATEOVRD_STATUS

Address: 0h base + 1Ch offset = 1Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Reserved							REF_SSP_EN	REF_USE_PAD	PHY_RESET_IN	REF_CLKDIV2_IN	REF_USB2_EN_IN	ATEOVRD_EN	PHY_RESET_OUT	REF_CLKDIV2_OUT	REF_USB2_EN_OUT
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_ATEOVRD_STATUS field descriptions

Field	Description
15–9 -	This field is reserved. Reserved.
8 REF_SSP_EN	Value from ASIC for ref_ssp_en.
7 REF_USE_PAD	Value from ASIC for ref_use_pad
6 PHY_RESET_IN	Value from ASIC for phy_reset
5 REF_CLKDIV2_IN	Value from ASIC for ref_clkdiv2.
4 REF_USB2_EN_IN	Value from ASIC for ref_usb2_en.
3 ATEOVRD_EN	When set, values from ATEOVRD register are sent to PHY.
2 PHY_RESET_OUT	Value from ATEOVRD for phy_reset.
1 REF_CLKDIV2_OUT	Value from ATEOVRD for ref_clkdiv2.
0 REF_USB2_EN_OUT	Value from ATEOVRD for ref_usb2_en.

50.7.21 PCIE_PHY_SCOPE_ENABLES

Address: 0h base + 20h offset = 20h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Reserved													MASK_SATURATION_MODE	MASK_EN	XOR_EN
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_SCOPE_ENABLES field descriptions

Field	Description
15–3 -	This field is reserved. Reserved.

Table continues on the next page...

PCIE_PHY_SCOPE_ENABLES field descriptions (continued)

Field	Description
2 MASK_ SATURATION_ MODE	Method of mask saturation. 1 Saturates when the first mask_counter reaches sample_limit. 0 Saturates when all mask_counters have reached sample_limit.
1 MASK_EN	Enables scope_mask input for tracking count values. Clears registers when deasserted.
0 XOR_EN	Uses scope_xor input for count values.

50.7.22 PCIE_PHY_SCOPE_SAMPLES

Address: 0h base + 21h offset = 21h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SAMPLES															
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

PCIE_PHY_SCOPE_SAMPLES field descriptions

Field	Description
SAMPLES	Number of samples to count.

50.7.23 PCIE_PHY_SCOPE_COUNT

Address: 0h base + 22h offset = 22h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	COUNT															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

PCIE_PHY_SCOPE_COUNT field descriptions

Field	Description
COUNT	A write to this register starts the counting process. The value of FFFF indicates counting still in progress. If in MASK mode, asserting MASK_EN also starts the counting

50.7.24 PCIE_PHY_SCOPE_CTL

Address: 0h base + 23h offset = 23h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved						COUNT	MASK_ SATURATI ON
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_SCOPE_CTL field descriptions

Field	Description
15–2 -	This field is reserved. Reserved.
1 COUNT	A write to this register starts the counting process. The value of FFFF indicates counting still in progress. If in MASK mode, asserting MASK_EN also starts the counting
0 MASK_ SATURATION	When asserted, mask registers have saturated.

50.7.25 PCIE_PHY_SCOPE_MASK*n*

Address: 0h base + 24h offset + (1d × i), where i=0d to 7d

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	MASK_VAL_n															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_SCOPE_MASK*n* field descriptions

Field	Description
MASK_VAL_n	Starting count value of mask register. Scope must be enabled to read from or write to this register.

50.7.26 PCIE_PHY_MPLL_LOOP_CTL

Address: 0h base + 30h offset = 30h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	PROP_CNTRL				INT_CNTRL		VBF_SF	VMB
Reset	1	1	0	0	0	0	0	0

PCIE_PHY_MPLL_LOOP_CTL field descriptions

Field	Description
15–8 -	This field is reserved. Reserved.
7–4 PROP_CNTRL	Charge pump proportional current setting.
3–2 INT_CNTRL	Charge pump integrating current setting.
1 VBF_SF	Measures MPLL VBF_SF (RC filtered gate voltage for VPSF source follower).
0 VMB	Measures MPLL master bias voltage.

50.7.27 PCIE_PHY_MPLL_ATB_MEAS2

Address: 0h base + 32h offset = 32h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	Reserved								IVCO_FILT	VCNTRL_M	VCNTRL_P	ATB_SENSE_SEL	MEAS_TEMP	FRC_PMIX_VPMIX	EN_MPMIX_VPMIX	EN_MPMIX_TST
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_MPLL_ATB_MEAS2 field descriptions

Field	Description
15–8 -	This field is reserved. Reserved.
7 IVCO_FILT	Puts filtered version of ivco on atb_s_p

Table continues on the next page...

PCIE_PHY_MPLL_ATB_MEAS2 field descriptions (continued)

Field	Description
6 VCNTRL_M	Puts dcc output vcntrl_p on atb_s_m
5 VCNTRL_P	Puts dcc output vcntrl_m on atb_s_p
4 ATB_SENSE_SEL	connects internal atb sense bus to external bus
3 MEAS_TEMP	Instructs POR block to measure the temperature.
2 FRC_PMIX_VPMIX	Forces mpll_pmix_vreg to use atb_s_m as its input instead of vbg.
1 EN_MPMIX_VPMIX	Puts vreg_pmix on atb_s_p.
0 EN_MPMIX_TST	Enables XOR gate to test linearity of MPLL phase mixer.

50.7.28 PCIE_PHY_MPLL_OVR

Address: 0h base + 33h offset = 33h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	PWRON_LCL	EN_PWRON_LCL	GS_LCL	EN_GS_LCL	RST_LCL	EN_RST_LCL	PMIX_CLK_SEL_LCL	EN_PMIX_CLK_SEL_LCL
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_MPLL_OVR field descriptions

Field	Description
15–8 -	This field is reserved. Reserved.

Table continues on the next page...

PCIE_PHY_MPLL_OVR field descriptions (continued)

Field	Description
7 PWRON_LCL	local power_on value
6 EN_PWRON_LCL	Enables local control of power_on
5 GS_LCL	local gear_shift value
4 EN_GS_LCL	Enables local control of gear_shift
3 RST_LCL	local Reset value
2 EN_RST_LCL	enable local control of reset
1 PMIX_CLK_SEL_LCL	local pmix_clk_sel value
0 EN_PMIX_CLK_SEL_LCL	enable local control of pmix_clk_sel

50.7.29 PCIE_PHY_RTUNE_RTUNE_CTRL

Address: 0h base + 34h offset = 34h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	RT_PWroN_FRC_ON	X4_FRC_OFF	RT_DAC_MODE		RT_DAC_CHOP	RT_ATB	RT_SEL_ATBP	RT_SEL_ATBF
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RTUNE_RTUNE_CTRL field descriptions

Field	Description
15–8 -	This field is reserved. Reserved.
7 RT_PWroN_FRC_ON	When set, forces RTUNE block to be on
6 X4_FRC_OFF	When set, do not multiply test current by 4
5–4 RT_DAC_MODE	Margin DAC mode control bits

Table continues on the next page...

PCIE_PHY_RTUNE_RTUNE_CTRL field descriptions (continued)

Field	Description
	00 powerdown 01 DAC drives atb_s_p/m directly 10 DAC drives atb_s_p/m to the RX in margining mode 11 illegal state
3 RT_DAC_CHOP	Margin DAC chop control bit
2 RT_ATB	RTUNE ATB mode control bit 1 RTUNE performs ADC on ATB input 0 not accessing ATB
1 RT_SEL_ATBP	RTUNE ATB sense input select bit 1 atb_s_p 0 atb_s_m
0 RT_SEL_ATBF	RTUNE ATB input slect bit 1 atb_fm 0 atb_s_p/m

50.7.30 PCIE_PHY_TX_OVRD_IN_LO

Address: 0h base + 1000h offset = 1000h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved		TX_DETECT_RX_REQ_OVRD	TX_DETECT_RX_REQ	TX_BEACON_EN_OVRD	TX_BEACON_EN	TX_CM_EN_OVRD	TX_CM_EN
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	TX_EN_OVRD	TX_EN	TX_DATA_EN_OVRD	TX_DATA_EN	TX_INVERT_OVRD	TX_INVERT	TX_LOOPBK_EN_OVRD	LOOPBK_EN
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_TX_OVRD_IN_LO field descriptions

Field	Description
15–14 -	This field is reserved. Reserved.

Table continues on the next page...

PCIE_PHY_TX_OVRD_IN_LO field descriptions (continued)

Field	Description
13 TX_DETECT_ RX_REQ_OVRD	Override enable for tx_detect_rx_req
12 TX_DETECT_ RX_REQ	Override value for tx_detect_rx_req
11 TX_BEACON_ EN_OVRD	Override enable for tx_beacon_en
10 TX_BEACON_ EN	Override value for tx_beacon_en
9 TX_CM_EN_ OVRD	Override enable for tx_cm_en
8 TX_CM_EN	Override value for tx_cm_en
7 TX_EN_OVRD	Override enable for tx_en
6 TX_EN	Override value for tx_en
5 TX_DATA_EN_ OVRD	Override enable for tx_data_en
4 TX_DATA_EN	Override value for tx_data_en
3 TX_INVERT_ OVRD	Override enable for tx_invert
2 TX_INVERT	Override value for tx_invert
1 TX_LOOPBK_ EN_OVRD	Override enable for loopbk_en
0 LOOPBK_EN	Override value for loopbk_en

50.7.31 PCIE_PHY_TX_OVRD_IN_HI

Address: 0h base + 1001h offset = 1001h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	TX_ RESET_ OVRD	TX_RESET	TX_ NYQUIST_ DATA	TX_CLK_ OUT_EN_ OVRD	TX_CLK_ OUT_EN	TX_RATE_ OVRD	TX_RATE	
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_TX_OVRD_IN_HI field descriptions

Field	Description
15–8 -	This field is reserved. Reserved.
7 TX_RESET_ OVRD	Override enable for tx_reset
6 TX_RESET	Override value for tx_reset
5 TX_NYQUIST_ DATA	Override incoming data to nyquist
4 TX_CLK_OUT_ EN_OVRD	Override enable for tx_clk_out_en.
3 TX_CLK_OUT_ EN	Override incomming tx_clk_out_en.
2 TX_RATE_ OVRD	Override enable for tx_rate.
TX_RATE	Override incomming tx lane rate.

50.7.32 PCIE_PHY_TX_OVRD_DRV_LO

Address: 0h base + 1003h offset = 1003h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved	EN	PREEMPH					
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	PREEMPH	AMPLITUDE						
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_TX_OVRD_DRV_LO field descriptions

Field	Description
15 -	This field is reserved. Reserved.
14 EN	Enables override values for all inputs controlled by this register
13–7 PREEMPH	Override value for transmit preemphasis
AMPLITUDE	Override value for transmit amplitude.

50.7.33 PCIE_PHY_TX_OVRD_OUT

Address: 0h base + 1004h offset = 1004h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	TX_STATE_ OVRD	TX_STATE	TX_CM_ STATE_ OVRD	TX_CM_ STATE	TX_ DETECT_ RX_ACK_ OVRD	TX_ DETECT_ RX_ACK	DETECT_ RX_RES_ OVRD	DETECT_ RX_RES
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_TX_OVRD_OUT field descriptions

Field	Description
15–8 -	This field is reserved. Reserved.
7 TX_STATE_ OVRD	Override enable for tx_state

Table continues on the next page...

PCIE_PHY_TX_OVRD_OUT field descriptions (continued)

Field	Description
6 TX_STATE	Override value for tx_state
5 TX_CM_STATE_OVRD	Override enable for tx_cm_state
4 TX_CM_STATE	Override value for tx_cm_state
3 TX_DETECT_RX_ACK_OVRD	Override enable for tx_detect_rx_ack
2 TX_DETECT_RX_ACK	Override value for tx_detect_rx_ack
1 DETECT_RX_RES_OVRD	Override enable for tx_detect_rx_res
0 DETECT_RX_RES	Override value for tx_detect_rx_res

50.7.34 PCIE_PHY_RX_OVRD_IN_LO

Address: 0h base + 1005h offset = 1005h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Reserved		RX_LOS_EN_OVRD	RX_LOS_EN	RX_TERM_EN_OVRD	RX_TERM_EN	RX_BIT_SHIFT_OVRD	RX_BIT_SHIFT	RX_ALIGN_EN_OVRD	RX_ALIGN_EN	RX_DATA_EN_OVRD	RX_DATA_EN	RX_PLL_EN_OVRD	RX_PLL_EN	RX_INVERT_OVRD	RX_INVERT
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_RX_OVRD_IN_LO field descriptions

Field	Description
15–14 -	This field is reserved. Reserved.
13 RX_LOS_EN_OVRD	Override enable for rx_los_en
12 RX_LOS_EN	Override value for rx_los_en
11 RX_TERM_EN_OVRD	Override enable for rx_term_en

Table continues on the next page...

PCIE_PHY_RX_OVRD_IN_LO field descriptions (continued)

Field	Description
10 RX_TERM_EN	Override value for rx_term_en
9 RX_BIT_SHIFT_OVRD	Override enable for rx_bit_shift
8 RX_BIT_SHIFT	Override value for rx_bit_shift
7 RX_ALIGN_EN_OVRD	Override enable for rx_align_en
6 RX_ALIGN_EN	Override value for rx_align_en
5 RX_DATA_EN_OVRD	Override enable for rx_data_en
4 RX_DATA_EN	Override value for rx_data_en
3 RX_PLL_EN_OVRD	Override enable for rx_pll_en
2 RX_PLL_EN	Override value for rx_pll_en
1 RX_INVERT_OVRD	Override enable for rx_invert
0 RX_INVERT	Override value for rx_invert

50.7.35 PCIE_PHY_RX_OVRD_IN_HI

Address: 0h base + 1006h offset = 1006h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved		RX_RESET_OVRD	RX_RESET	RX_EQ_OVRD	RX_EQ		
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	RX_EQ_EN_OVRD	RX_EQ_EN	RX_LOS_FILTER_OVRD	RX_LOS_FILTER		RX_RATE_OVRD	RX_RATE	
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RX_OVRD_IN_HI field descriptions

Field	Description
15–14 -	This field is reserved. Reserved.
13 RX_RESET_ OVRD	Override enable for rx_reset
12 RX_RESET	Override value for rx_reset
11 RX_EQ_OVRD	Override enable for rx_eq
10–8 RX_EQ	Override value for rx_eq
7 RX_EQ_EN_ OVRD	Override enable for rx_eq_en
6 RX_EQ_EN	Override value for rx_eq_en
5 RX_LOS_ FILTER_OVRD	Override enable for rx_los_filter
4–3 RX_LOS_ FILTER	Override value for rx_los_filter
2 RX_RATE_ OVRD	Override enable for rx_rate
RX_RATE	Override value for rx_rate

50.7.36 PCIE_PHY_RX_OVRD_OUT

Address: 0h base + 1007h offset = 1007h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved	ZERO_ DATA	LOS_OVRD	LOS	PLL_ STATE_ OVRD	PLL_STATE	VALID_ OVRD	VALID
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RX_OVRD_OUT field descriptions

Field	Description
15–7 -	This field is reserved. Reserved.
6 ZERO_DATA	Override data output to all zeros
5 LOS_OVRD	Override value for rx_los
4 LOS	Override value for rx_los
3 PLL_STATE_ OVRD	Override enable for rx_pll_state
2 PLL_STATE	Override value for rx_pll_state
1 VALID_OVRD	Override enable for rx_valid
0 VALID	Override value for rx_valid

50.7.37 PCIE_PHY_TX_ASIC_IN

Address: 0h base + 1008h offset = 1008h

Bit	15	14	13	12	11	10	9	8
Read	Reserved					TX_CLK_ OUT_EN	DETECT_ RX_REQ	BEACON_ EN
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	CM_EN	TX_EN	DATA_EN	TX_RESET	INVERT	LOOPBK_ EN	TX_RATE	
Write								
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_TX_ASIC_IN field descriptions

Field	Description
15–11 -	This field is reserved. Reserved.
10 TX_CLK_OUT_ EN	Value from ASIC for tx_clk_out_en

Table continues on the next page...

PCIE_PHY_TX_ASIC_IN field descriptions (continued)

Field	Description
9 DETECT_RX_ REQ	Value from ASIC for tx_detect_rx_req
8 BEACON_EN	Value from ASIC for tx_beacon_en
7 CM_EN	Value from ASIC for tx_cm_en
6 TX_EN	Value from ASIC for tx_en
5 DATA_EN	Value from ASIC for tx_data_en
4 TX_RESET	Value from ASIC for tx_reset
3 INVERT	Value from ASIC for tx_invert
2 LOOPBK_EN	Value from ASIC for loopbk_en
TX_RATE	Value from ASIC for tx_rate

50.7.38 PCIE_PHY_TX_ASIC_DRV_LO

Address: 0h base + 1009h offset = 1009h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Reserved		PREEMPH							AMPLITUDE						
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_TX_ASIC_DRV_LO field descriptions

Field	Description
15–14 -	This field is reserved. Reserved.
13–7 PREEMPH	Value from ASIC for tx_preemph
AMPLITUDE	Value from ASIC for tx_amplitude

50.7.39 PCIE_PHY_TX_ASIC_DRV_HI

Address: 0h base + 100Ah offset = 100Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Reserved											TERM_OFFSET				
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_TX_ASIC_DRV_HI field descriptions

Field	Description
15–5 -	This field is reserved. Reserved.
TERM_OFFSET	Value from ASIC for tx_term_offset

50.7.40 PCIE_PHY_TX_ASIC_OUT

Address: 0h base + 100Bh offset = 100Bh

Bit	15	14	13	12	11	10	9	8
Read	Reserved							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	Reserved			STATE	CM_STATE	DETECT_ RX_ACK	0	DETECT_ RX_RES
Write								
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_TX_ASIC_OUT field descriptions

Field	Description
15–5 -	This field is reserved. Reserved.
4 STATE	Value from PHY for tx_state
3 CM_STATE	Value from PHY for tx_cm_state
2 DETECT_RX_ ACK	Value from PHY for tx_detect_rx_ack

Table continues on the next page...

PCIE_PHY_TX_ASIC_OUT field descriptions (continued)

Field	Description
1 Reserved	This read-only field is reserved and always has the value 0.
0 DETECT_RX_RES	Value from PHY for tx_detect_rx_res

50.7.41 PCIE_PHY_RX_ASIC_IN

Address: 0h base + 100Ch offset = 100Ch

Bit	15	14	13	12	11	10	9	8
Read	RX_EQ_EN	RX_EQ			LOS_FILTER		LOS_EN	TERM_EN
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	CLK_SHIFT	ALIGN_EN	DATA_EN	PLL_EN	RX_RESET	INVERT	RX_RATE	
Write								
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RX_ASIC_IN field descriptions

Field	Description
15 RX_EQ_EN	Value from ASIC for rx_eq_en
14–12 RX_EQ	Value from ASIC for rx_eq
11–10 LOS_FILTER	Value from ASIC for rx_los_filter
9 LOS_EN	Value from ASIC for rx_los_en
8 TERM_EN	Value from ASIC for rx_term_en
7 CLK_SHIFT	Value from ASIC for rx_bit_shift
6 ALIGN_EN	Value from ASIC for rx_align_en
5 DATA_EN	Value from ASIC for rx_data_en
4 PLL_EN	Value from ASIC for rx_pll_en
3 RX_RESET	Value from ASIC for rx_reset

Table continues on the next page...

PCIE_PHY_RX_ASIC_IN field descriptions (continued)

Field	Description
2 INVERT	Value from ASIC for rx_invert
RX_RATE	Value from ASIC for rx_rate

50.7.42 PCIE_PHY_RX_ASIC_OUT

Address: 0h base + 100Dh offset = 100Dh

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	Reserved					LOS	PLL_STATE	VALID
Write								
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RX_ASIC_OUT field descriptions

Field	Description
15–3 -	This field is reserved. Reserved.
2 LOS	Value from PHY for rx_los
1 PLL_STATE	Value from PHY for rx_pll_state
0 VALID	Value from PHY for rx_valid

50.7.43 PCIE_PHY_TX_VMD_FSM_TX_VCM_0

Address: 0h base + 1011h offset = 1011h

Bit	15	14	13	12	11	10	9	8
Read	Reserved	DONE	N_USE					
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	N_USE	N_TRISTATE						
Write								
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_TX_VMD_FSM_TX_VCM_0 field descriptions

Field	Description
15 -	This field is reserved. Reserved.
14 DONE	Configuration is done
13–7 N_USE	Value from VMD for legs to use
N_TRISTATE	Value from VMD for number of tristate legs.

50.7.44 PCIE_PHY_TX_VMD_FSM_TX_VCM_1

Address: 0h base + 1012h offset = 1012h

Bit	15	14	13	12	11	10	9	8
Read	FIXED_ DONE	TRA_DONE	N_FIXED					
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	N_FIXED	N_TRAILER						
Write								
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_TX_VMD_FSM_TX_VCM_1 field descriptions

Field	Description
15 FIXED_DONE	N_FIXED Multiplication has completed.
14 TRA_DONE	N_TRAILER Multiplication has completed.
13–7 N_FIXED	Value from VMD for number of fixed driver legs.
N_TRAILER	Value from VMD for number of trailer legs.

50.7.45 PCIE_PHY_TX_VMD_FSM_TX_VCM_DEBUG_IN

Address: 0h base + 1013h offset = 1013h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

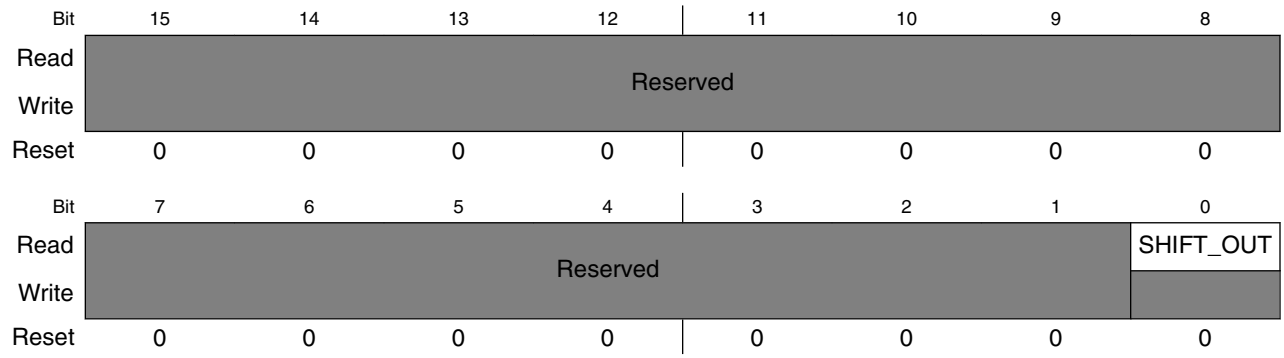
Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				CONFIG_ OVRD	CONFIG_ LOAD	CONFIG_ CLK	CONFIG_ DATA
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_TX_VMD_FSM_TX_VCM_DEBUG_IN field descriptions

Field	Description
15–4 -	This field is reserved. Reserved
3 CONFIG_OVRD	Override the Voltage Mode Driver Configuration FSM and access the shift chain directly.
2 CONFIG_LOAD	Override value for the Voltage Mode Driver Configuration FSM's config load.
1 CONFIG_CLK	Override value for the Voltage Mode Driver Configuration FSM's config clk.
0 CONFIG_DATA	Override value for the Voltage Mode Driver Configuration FSM's config data.

50.7.46 PCIE_PHY_TX_VMD_FSM_TX_VCM_DEBUG_OUT

Address: 0h base + 1014h offset = 1014h

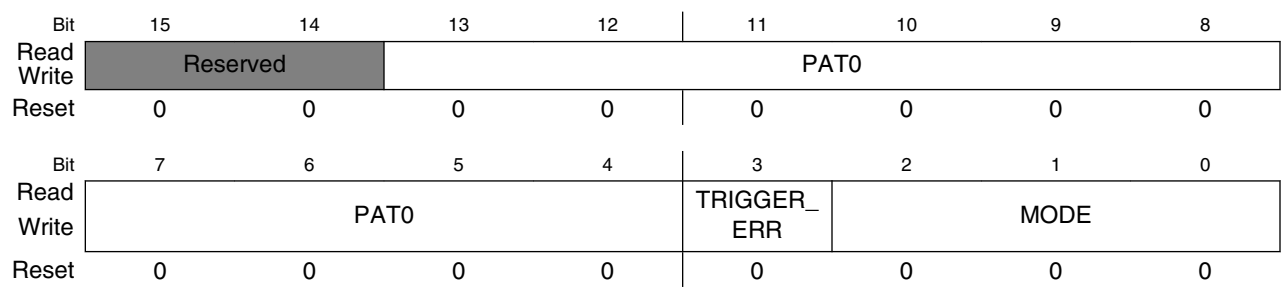


PCIE_PHY_TX_VMD_FSM_TX_VCM_DEBUG_OUT field descriptions

Field	Description
15–1 -	This field is reserved. Reserved
0 SHIFT_OUT	Current value from TX_ANAs configuration shift register.

50.7.47 PCIE_PHY_TX_LBERT_CTL

Address: 0h base + 1015h offset = 1015h



PCIE_PHY_TX_LBERT_CTL field descriptions

Field	Description
15–14 -	This field is reserved. Reserved
13–4 PAT0	Pattern for modes 3-5
3 TRIGGER_ERR	Insert a single error into a lsb Any write of a 1 to this bit will insert an error
MODE	Pattern to generate When changing modes, you must first change to disabled.

Table continues on the next page...

PCIE_PHY_TX_LBERT_CTL field descriptions (continued)

Field	Description
	6 DC-balanced word (PAT0) 5 Fixed word (PAT0) 4 lfsr7. X^7 + X^6 + 1 3 lfsr15. X^15 + X^14 + 1 2 lfsr23. X^23 + X^18 + 1 1 lfsr31. X^31 + X^28 + 1 0 Disabled

50.7.48 PCIE_PHY_RX_LBERT_CTL

Address: 0h base + 1016h offset = 1016h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved				SYNC	MODE		
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RX_LBERT_CTL field descriptions

Field	Description
15–4 -	This field is reserved. Reserved
3 SYNC	Synchronize pattern matcher LFSR with incoming data A write of a one to this bit will reset the error counter and start a synchronization of the PM. There is no need to write this back to zero to run normally.
MODE	Pattern to match When changing modes, you must first change to disabled. 7 d[n] = 6 d[n] = !d[n-10] 5 d[n] = d[n-10] 4 lfsr7 : X^7 + X^6 + 1 3 lfsr15: X^15 + X^14 + 1 2 lfsr23. X^23 + X^18 + 1 1 lfsr31. X^31 + X^28 + 1 0 Disabled

50.7.49 PCIE_PHY_RX_LBERT_ERR

Address: 0h base + 1017h offset = 1017h

Bit	15	14	13	12	11	10	9	8
Read Write	OV14	COUNT						
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	COUNT							
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RX_LBERT_ERR field descriptions

Field	Description
15 OV14	If active, multiply COUNT by 128. If OV14=1 and COUNT=2^15-1, signals overflow of counter
COUNT	A read of this register, or a sync of the PM resets the error count. Current error count If OV14 field is active, then multiply count by 128

50.7.50 PCIE_PHY_RX_SCOPE_CTL

Address: 0h base + 1018h offset = 1018h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	Reserved													MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_RX_SCOPE_CTL field descriptions

Field	Description
15–3 -	This field is reserved. Reserved
MODE	Sampling mode of counters. **NOTE:** WORD is 20 bits. 0 Off 1 Sample data every WORD *(1 + DELAY) bits 2 Sample data every WORD *(1 + DELAY) + 1 bits 3 Sample data every WORD *(1 + DELAY) + 2 bits 4 Sample data every clk and assert XOR and MASK increment

50.7.51 PCIE_PHY_RX_SCOPE_PHASE

Address: 0h base + 1019h offset = 1019h

Bit	15	14	13	12	11	10	9	8
Read / Write	Reserved	BASE					SCOPE_DELAY	
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read / Write	SCOPE_SEL	UPDATE	SAMPLE_PHASE					
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RX_SCOPE_PHASE field descriptions

Field	Description
15 -	This field is reserved. Reserved
14–10 BASE	which bit to sample when MODE = 1 or 4
9–8 SCOPE_DELAY	How many clocks to delay the analog scope_data.
7 SCOPE_SEL	Select sampling mode. 0 Before AFE sampling 1 After AFE sampling
6 UPDATE	Update Sampling phase. Write a 1.
SAMPLE_PHASE	Sampling Phase

50.7.52 PCIE_PHY_RX_DPLL_FREQ

Address: 0h base + 101Ah offset = 101Ah

Bit	15	14	13	12	11	10	9	8
Read / Write	Reserved			VAL				
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read / Write	VAL							DTHR
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RX_DPLL_FREQ field descriptions

Field	Description
15–13 -	This field is reserved. Reserved
12–1 VAL	Freq is 1.526*VAL ppm from the reference When mpll_slow is set, the ppm is half the eqn above
0 DTHR	Bits below the useful resolution

50.7.53 PCIE_PHY_RX_CDR_CTL

Address: 0h base + 101Bh offset = 101Bh

Bit	15	14	13	12	11	10	9	8
Read Write	DTB_SEL				ALWAYS_ REALIGN	FAST_ START	FRUG_VALUE	
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	PHUG_VALUE		OVRD_ DPLL_GAIN	PHDET_ POL	PHDET_EDGE		PHDET_EN	
Reset	0	0	0	0	1	1	1	1

PCIE_PHY_RX_CDR_CTL field descriptions

Field	Description
15–12 DTB_SEL	Select to drive various signals onto the DTB. 0 disabled 1 pll_ana_rst,pll_count from rx_pwr_ctl 2 com_good_high/low from aligner 3 com_bad_high/low from aligner 4 shift_in_prog,ana_odd_data from aligner 5 Low bits of XAUI align FSM state
11 ALWAYS_ REALIGN	realign on any misaligned comma
10 FAST_START	decrease startup steps by 50%
9–8 FRUG_VALUE	override value for FRUG
7–6 PHUG_VALUE	override value for PHUG
5 OVRD_DPLL_ GAIN	Override PHUG and FRUG values
4 PHDET_POL	Reverse polarity of phase error

Table continues on the next page...

PCIE_PHY_RX_CDR_CTL field descriptions (continued)

Field	Description
3–2 PHDET_EDGE	Edges to use for phase detection. 11 Use both edges 10 Use rising edges only 01 Use falling edges only 00 Ignore all edges
PHDET_EN	Enables phase detecter. top bit is odd slicers, bottom is even

50.7.54 PCIE_PHY_RX_CDR_CDR_FSM_DEBUG

Address: 0h base + 101Ch offset = 101Ch

Bit	15	14	13	12	11	10	9	8
Read	adap_rx_eq			rx_eq_ctr			rx_ana_eq	
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	rx_ana_eq	adap_rx_ valid	cdr_en_ adap	cdr_en_eq	aligned	cdr_rx_valid	cdr_timeout	cdr_en
Write								
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RX_CDR_CDR_FSM_DEBUG field descriptions

Field	Description
15–13 adap_rx_eq	Equalization setting from adaptation FSM.
12–10 rx_eq_ctr	Initial centre point from equalization FSM.
9–7 rx_ana_eq	Equalization setting to Analog.
6 adap_rx_valid	Adaptation has completed and locked
5 cdr_en_adap	Adapatation loop is enabling the CDR.
4 cdr_en_eq	Equalization loop is enabling the CDR.
3 aligned	Datapath is bit-aligned.
2 cdr_rx_valid	CDR has locked to incoming data stream.

Table continues on the next page...

PCIE_PHY_RX_CDR_CDR_FSM_DEBUG field descriptions (continued)

Field	Description
1 cdr_timeout	CDR has not locked to datastream and has timed-out.
0 cdr_en	CDR has been enabled.

50.7.55 PCIE_PHY_RX_CDR_LOCK_VEC_OVRD

Address: 0h base + 101Dh offset = 101Dh

Bit	15	14	13	12	11	10	9	8
Read Write	adap_ctr_level					adap_ polarity	lock_vector_ ovrd	lock_vector_ en
Reset	1	0	0	0	0	0	0	0

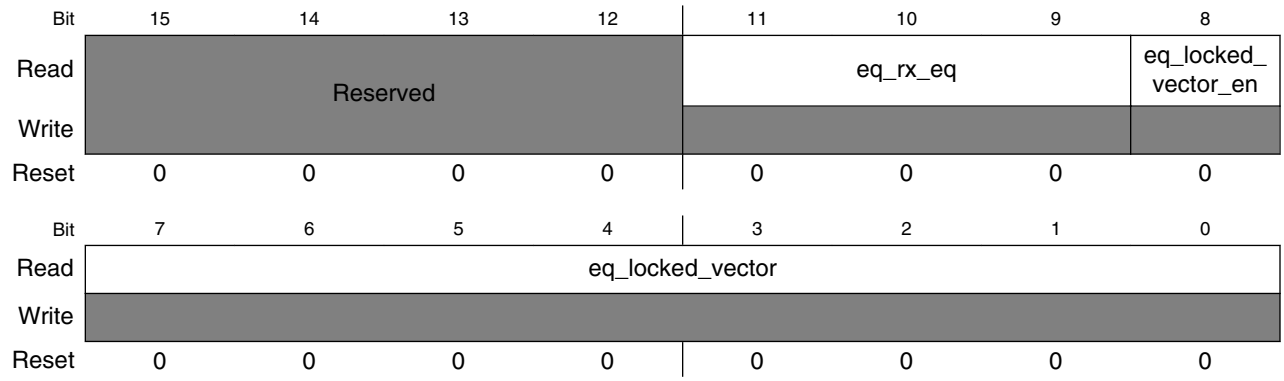
Bit	7	6	5	4	3	2	1	0
Read Write	lock_vector							
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RX_CDR_LOCK_VEC_OVRD field descriptions

Field	Description
15–11 adap_ctr_level	Amount of earlies that increment the adaptation counter (times 16).
10 adap_polarity	If asserted invert default adaptation adjustment for equalization. IF early decrease equalization. Normal mode is to decrease.
9 lock_vector_ovrd	Override enable for the rx_eq outputs.
8 lock_vector_en	Override value for the locked_vector output completion.
lock_vector	Override value for the locked_vector.

50.7.56 PCIE_PHY_RX_CDR_LOCK_VEC

Address: 0h base + 101Eh offset = 101Eh

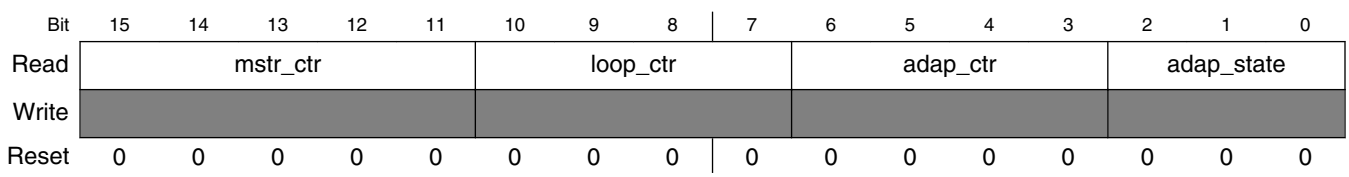


PCIE_PHY_RX_CDR_LOCK_VEC field descriptions

Field	Description
15–12 -	This field is reserved. Reserved
11–9 eq_rx_eq	Equalization setting from the Equalization Loop.
8 eq_locked_ vector_en	Equalization locked vector has been filled.
eq_locked_vector	Results of equalization loop.

50.7.57 PCIE_PHY_RX_CDR_ADAP_FSM

Address: 0h base + 101Fh offset = 101Fh



PCIE_PHY_RX_CDR_ADAP_FSM field descriptions

Field	Description
15–11 mstr_ctr	Master count register.
10–7 loop_ctr	Loop count register.
6–3 adap_ctr	Adaptation count register.

Table continues on the next page...

PCIE_PHY_RX_CDR_ADAP_FSM field descriptions (continued)

Field	Description
adap_state	Adaptation State. 000 ADAP_RESET 001 ADAP_LOCK 010 ADAP_SUFF 011 ADAP_LOOP 100 ADAP_MSTR 101 ADAP_DONE

50.7.58 PCIE_PHY_RX_ATB0

Address: 0h base + 1020h offset = 1020h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Reserved								EN_ATB	EN_ATB	EN_MARG	EN_ATB_RM_F	EN_ATB_RM_S	EN_ATB_RP_F	EN_ATB_RP_S	EN_ATB_VOFF
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_RX_ATB0 field descriptions

Field	Description
15–8 -	This field is reserved. Reserved
7 EN_ATB	Enables ATB sensing and forcing on internal Rx nodes.
6 EN_ATB	Enables margining mode in receiver; requires atb_f_m to be high-Z!.
5 EN_MARG	Enables atb_force_p on negative-side termination resistor.
4 EN_ATB_RM_F	Enables atb_sense_m on negative-side termination resistor.
3 EN_ATB_RM_S	Enables atb_force_p on positive-side termination resistor.
2 EN_ATB_RP_F	Enables atb_sense_p on positive-side termination resistor.
1 EN_ATB_RP_S	Puts rxafe outputs vo_p on atb_s_p and vo_m on atb_s_m.
0 EN_ATB_VOFF	Puts rxafe voff_p on atb_s_p and voff_m on atb_s_m.

50.7.59 PCIE_PHY_RX_ATB1

Address: 0h base + 1021h offset = 1021h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	VLOS_MAX	VLOS_MIN	EN_ATB_ VLOS	EN_ATB_ VRF	MEAS_GD	MEAS_VP	EN_VLOS_ USB3	NC0
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RX_ATB1 field descriptions

Field	Description
15–8 -	This field is reserved. Reserved
7 VLOS_MAX	Sets LOS reference voltage. (VLOS_MAX, VLOS_MIN): (1,1): None (1,0): Maximum (0,1): Minimum (0,0): Nominal
6 VLOS_MIN	Sets LOS reference voltage. (VLOS_MAX, VLOS_MIN): (1,1): None (1,0): Maximum (0,1): Minimum (0,0): Nominal
5 EN_ATB_VLOS	Enables sensing of LOS reference voltage on atb_sense_p.
4 EN_ATB_VRF	Enables sensing of vref_rx on atb_sense_p.
3 MEAS_GD	Enables sensing of local gd in Rx; ties gd to atb_sense_m.
2 MEAS_VP	Enables sensing of local vp in Rx; ties vp to atb_sense_p.
1 EN_VLOS_USB3	Enables LOS levels to be those for USB3; otherwise, PCI Express levels.
0 NC0	Enables/disables Rx termination resistor.

50.7.60 PCIE_PHY_RX_ENPWR0

Address: 0h base + 1022h offset = 1022h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	CTL_ RXPWRON	LCL_ RXPWRON	CTL_EN_ LOS	LCL_EN_ LOS	CTL_RXCK	LCL_RXCK	CTL_ACJT	LCL_ACJT
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RX_ENPWR0 field descriptions

Field	Description
15–8 -	This field is reserved. Reserved
7 CTL_RXPWRON	Enables override of Rx block power.
6 LCL_RXPWRON	Enables/disables Rx slicers.
5 CTL_EN_LOS	Enables override of LOS block state.
4 LCL_EN_LOS	Enables/disables LOS block.
3 CTL_RXCK	Enables override of Rx clock circuit state.
2 LCL_RXCK	Enables/disables en_rx_clock (Rx clock enable).
1 CTL_ACJT	Enables override of ACJTAG block state.
0 LCL_ACJT	Enables/disables ACJTAG block.

50.7.61 PCIE_PHY_RX_PMIX_PHASE

Address: 0h base + 1023h offset = 1023h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	Reserved								PHASE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_RX_PMIX_PHASE field descriptions

Field	Description
15–8 -	This field is reserved. Reserved
PHASE	Write to bits 8-1 of the Phase Select register in the phase mixer.

50.7.62 PCIE_PHY_RX_ENPWR1

Address: 0h base + 1024h offset = 1024h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	LCL_ RXTERM	CTL_ RXTERM	LCL_BST			CTL_BST	LCL_ PHASE_ REG_RST	CTL_ PHASE_ REG_RST
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RX_ENPWR1 field descriptions

Field	Description
15–8 -	This field is reserved. Reserved
7 LCL_RXTERM	Enables/disables Rx termination.
6 CTL_RXTERM	Enables override of rx_term_en.
5–3 LCL_BST	Rx boost (equalization) value
2 CTL_BST	Enables override of Rx boost (equalization) value.
1 LCL_PHASE_ REG_RST	Reset Phase register.
0 CTL_PHASE_ REG_RST	Enables override of Phase register reset.

50.7.63 PCIE_PHY_RX_ENPWR2

Address: 0h base + 1025h offset = 1025h

Bit	15	14	13	12	11	10	9	8
Read	Reserved							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	EN_RXPMIX_TST	EN_RXPMIX_VPMIX	EN_RXPMIX_VRX	EN_RXPMIX_VOSC	EN_RXPMIX_FRC_VPMIX	RX_SCOPE_ATB_2	RX_SCOPE_ATB_1	RX_SCOPE_ATB_0
Write								
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RX_ENPWR2 field descriptions

Field	Description
15–8 -	This field is reserved. Reserved
7 EN_RXPMIX_TST	Enables XOR gate to test linearity of Rx phase mixer using atb_s_p and atb_s_m.
6 EN_RXPMIX_VPMIX	Puts vreg_pmix on atb_s_p.
5 EN_RXPMIX_VRX	Puts vreg_rx on atb_s_p.
4 EN_RXPMIX_VOSC	Puts vreg_vosc on atb_s_p.
3 EN_RXPMIX_FRC_VPMIX	Instructs rx_pmix_vreg_pmix to use atb_s_m as a reference instead of vbg.
2 RX_SCOPE_ATB_2	Puts XOR of Rx scope PMIX input and output on atb_s_p.
1 RX_SCOPE_ATB_1	Puts Rx scope regulated VP on atb_s_p.
0 RX_SCOPE_ATB_0	Instructs Rx scope regulated VP to use atb_f_p as reference instead of VP.

50.7.64 PCIE_PHY_RX_SCOPE

Address: 0h base + 1026h offset = 1026h

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	NC_SCOPE_2			RX_ SCOPE_ SLEW	RX_ SCOPE_ FDIV20	NC_SCOPE_3		
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_RX_SCOPE field descriptions

Field	Description
15–8 -	This field is reserved. Reserved
7–5 NC_SCOPE_2	NC
4 RX_SCOPE_ SLEW	Sets high for low Rx clock frqeuencies (625 MHz) for Rx scope to work correctly.
3 RX_SCOPE_ FDIV20	Divides scope output clock by 20 instead of 10.
NC_SCOPE_3	NC

50.7.65 PCIE_PHY_TX_TXDRV_CNTRL

Address: 0h base + 102Bh offset = 102Bh

Bit	15	14	13	12	11	10	9	8
Read	Reserved							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	PULL_DN_REG	PULL_UP_REG	OVRD_PULL_UP	VCM_HOLD_REG	OVRD_VCM_HOLD	Reserved	Reserved	Reserved
Write								
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_TX_TXDRV_CNTRL field descriptions

Field	Description
15–8 -	This field is reserved. Reserved
7 PULL_DN_REG	Register bit that causes the calibrated Tx bits to pull down in common mode fashion. If pull_dn_reg and tx_pull_up are both high, then pull_dn_reg wins (takes precedence").
6 PULL_UP_REG	Register override for tx_pull_up; selected when ovrd_pull_up is high; causes calibrated TX bits to pull up in common mode fashion, unless pull_dn_reg is high.
5 OVRD_PULL_UP	Selects loval value of pull_up_reg instead of tx_pull_up.
4 VCM_HOLD_REG	Register override for tx_vcm_hold; selected when ovrd_vcm_hold is high; controls the TX common mode hold circuitry.
3 OVRD_VCM_HOLD	Selects local value of vcm_hold_reg instead of tx_vcm_hold to control state of TX common mode hold circuitry.
2 NOCONN_8	This field is reserved. Reserved
1 NOCONN_7	This field is reserved. Reserved
0 NOCONN_6	This field is reserved. Reserved

50.7.66 PCIE_PHY_TX_POWER_CTL

Address: 0h base + 102Ch offset = 102Ch

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	OVRD_EN	SERIAL_ EN_REG	CLK_EN_ REG	DATA_EN_ REG	REFGEN_ EN_REG	TX_DIV_ CLK_EN	REFGEN_ PDN_REG	Reserved
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_TX_POWER_CTL field descriptions

Field	Description
15–8 -	This field is reserved. Reserved
7 OVRD_EN	Enables local overrides for all signals in this register.
6 SERIAL_EN_ REG	Value for tx_serial_en when OVRD_EN is 1.
5 CLK_EN_REG	Value for tx_clk_en when OVRD_EN is 1.
4 DATA_EN_REG	Value for tx_data_en when OVRD_EN is 1.
3 REFGEN_EN_ REG	Register override value for tx_refgen_en; turns on the pmos_bias refgen block and the rxdetect comparators.
2 TX_DIV_CLK_ EN	Enables the div clock that is output from the Tx to the undersampler, more appropriately called tx_sampler_clk_en; this clock is output after the optional divide-by-2/ 4; tx_clk_en must be high to output a clock.
1 REFGEN_PDN_ REG	Value for refgen_pwdn when OVRD_EN is 1.
0 NOCONN_5	This field is reserved. Reserved

50.7.67 PCIE_PHY_TX_ALT_BLOCK

Address: 0h base + 102Dh offset = 102Dh

Bit	15	14	13	12	11	10	9	8
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	EN_ALT_BUS	DRV_SOURCE_REG		JTAG_DATA_REG	ALT_OSC_VP	ALT_OSC_VPH	ALT_OSC_VPHREG	OVRD_ALT_BUS
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_TX_ALT_BLOCK field descriptions

Field	Description
15–8 -	This field is reserved. Reserved
7 EN_ALT_BUS	Enables the Tx for alt bus mode, powers up the pmos_bias block, and so on; required if manually running the alt bus features.
6–5 DRV_SOURCE_REG	Value for tx_data_source when OVRD_ALT_BUS is 1 11 JTAG data common mode for test 10 LFPS oscillator differential 01 JTAG data differential 00 Serializer data or alt oscillator vp/vph/vphreg/vptx if selected
4 JTAG_DATA_REG	Value for jtag_data when OVRD_ALT_BUS is 1.
3 ALT_OSC_VP	Enables and connects the vp oscillator to the transmit pins; must set drv_source_reg bus correctly.
2 ALT_OSC_VPH	Enables and connects the vph oscillator to the transmit pins; must set drv_source_reg bus correctly.
1 ALT_OSC_VPHREG	Enables and connects the vphreg oscillator to the transmit pins; must set drv_source_reg bus correctly.
0 OVRD_ALT_BUS	Enables local overrides for alt-bus control signals.

50.7.68 PCIE_PHY_TX_ALT_AND_LOOPBACK

Address: 0h base + 102Eh offset = 102Eh

Bit	15	14	13	12	11	10	9	8
Read	Reserved							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	OVRD_TX_LB	TX_LB_EN_REG	ALT_VPTX_OSC	Reserved	Reserved	Reserved	Reserved	Reserved
Write								
Reset	0	0	0	0	0	0	0	0

PCIE_PHY_TX_ALT_AND_LOOPBACK field descriptions

Field	Description
15–8 -	This field is reserved. Reserved
7 OVRD_TX_LB	Enables the override of the tx_lb_en pin.
6 TX_LB_EN_REG	Value of the tx_lb_en pin when OVRD_TX_LB is enabled.
5 ALT_VPTX_OSC	Enables and connects the vptx oscillator to the transmit pins; must set drv_source_reg bus correctly.
4 NOCONN_04	This field is reserved. Reserved
3 NOCONN_03	This field is reserved. Reserved
2 NOCONN_02	This field is reserved. Reserved
1 NOCONN_01	This field is reserved. Reserved
0 NOCONN_00	This field is reserved. Reserved

50.7.69 PCIE_PHY_TX_TX_ATB_REG

Address: 0h base + 102Fh offset = 102Fh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Reserved								ATB_PBIAS	ATB_VCM_REP	ATB_RXDETREF	ATB_TXFP	ATB_TXFM	ATB_TXSP	ATB_TXSM	ATB_VCM
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_TX_TX_ATB_REG field descriptions

Field	Description
15–8 -	This field is reserved. Reserved
7 ATB_PBIAS	Connects real pmos_bias voltage for Tx PMOS driver pull-up path to atb_s_p and local ground at the pmos_bias block to atb_s_m.
6 ATB_VCM_REP	Connects common mode replica voltage in pmos_bias block to atb_s_p and local ground to atb_s_m.
5 ATB_RXDETREF	Connects Rx detect block reference voltage to atb_s_p and local ground to atb_s_m.
4 ATB_TXFP	Connects tx_p to atb_f_p.
3 ATB_TXFM	Connects tx_m to atb_f_m.
2 ATB_TXSP	Connects tx_p to atb_s_p.
1 ATB_TXSM	Connects tx_m to atb_s_m.
0 ATB_VCM	Connects tx_p/tx_m common mode voltage onto atb_s_p and local ground onto atb_s_m.

Chapter 51
Power Management Unit (PMU)

51.1 Overview

The power management unit (PMU) is designed to simplify the external power interface. The power system can be split into the input power sources and their characteristics, the integrated power transforming and controlling elements, and the final load interconnection and requirements.

A typical power system uses the PMU is depicted in the following diagram.

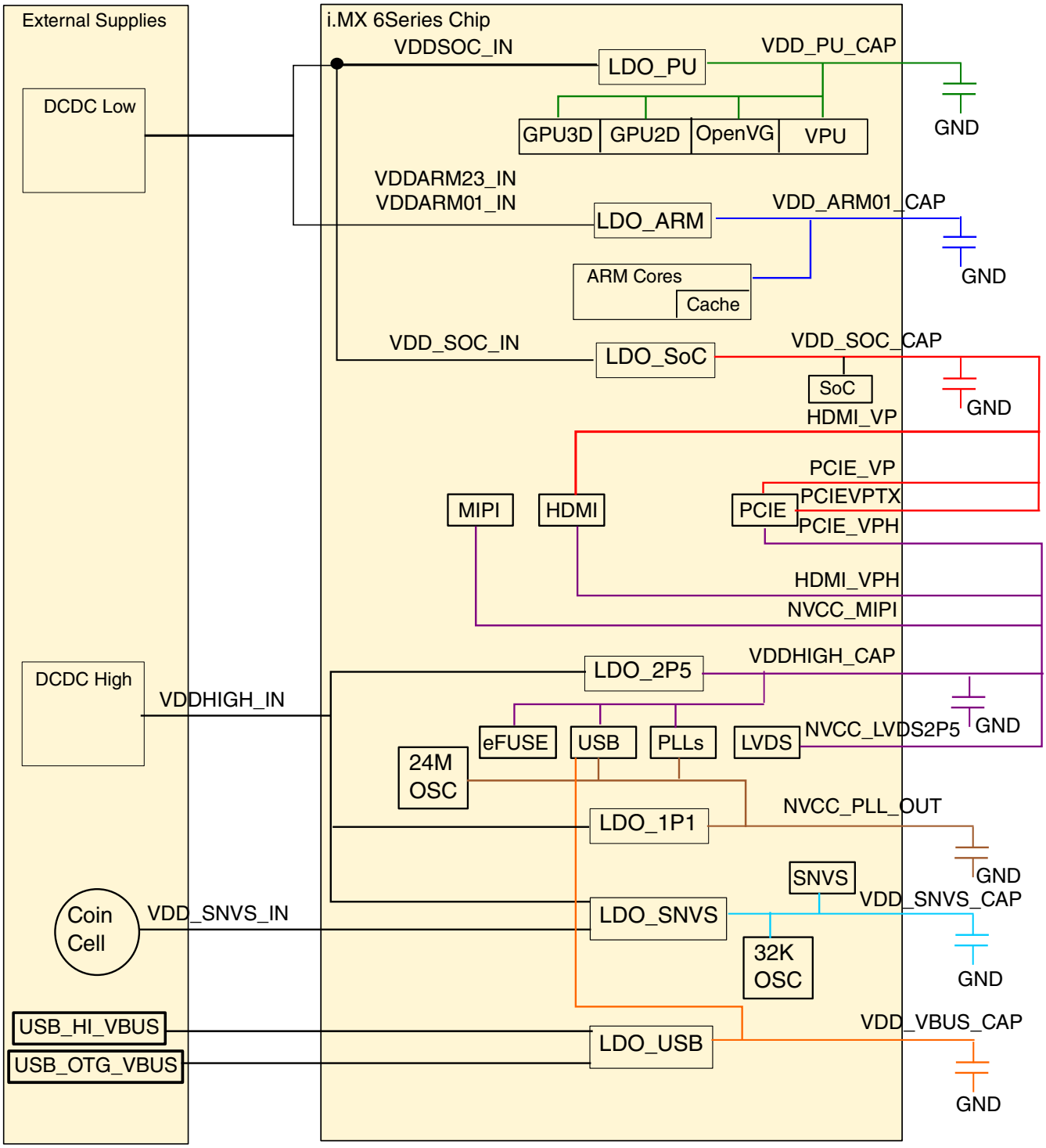


Figure 51-1. Power system overview

Using seven LDO regulators, the number of external supplies is greatly reduced. Not counting the backup coin and USB inputs, the number of external supplies is reduced to two. Missing from this external supply total is the number of necessary external supplies to power the desired memory interface; that number varies depending on the type of external memory selected. Other supplies may also be necessary to supply the voltage to the different I/O power segments if their I/O voltages have to be different from what is provided above.

51.2 Digital LDO Regulators

The PMU has three digital LDO regulators. They are referred to as "digital" because of the logic loads they drive, not because of their construction. These regulators have three basic modes that are unique to the digital regulators.

- External Bypass—The input and output of the regulator are shorted externally to the SOC. If operating in this configuration, the internal bypass mode must be enabled before attempting high-frequency/high-power operation.
- Power Gate—The regulation FET is switched off fully, limiting the current draw from the supply. The analog part of the regulator is powered down, limiting the power consumption. The output voltage falls to a level at which the residual leakage of the power FET balances with the leakage of the load. (TARG = 0x00)
- Analog regulation mode—The regulation FET is controlled such that the output voltage of the regulator equals the programmed target voltage. The target voltage is fully programmable in 25mV steps.

These modes allow the regulators to implement voltage scaling and power gating and allow bypass. With the bypass feature, all of the accuracy and control requirements can be shifted to the external supply source if capable and desired.

These digital regulators also feature brownout detection which is helpful when supplies are starting to collapse. The voltage value where brownout is signaled is programmable as an offset from the programmed target voltage. The controls are located in the PMU_MISC2 register. The core is interrupted on a brownout.

The three digital regulators are known as LDO_ARM, LDO_PU, and LDO_SOC. As shown in the power system overview figure, the Arm regulator powers the Arm cores. The LDO_PU powers the GPU, VPU, and display portions of the chip. The SOC regulator powers the rest of the digital logic on the chip. All regulators support generous programming ranges in 25mV steps. It is possible to program voltages above the process limit for the chip, thus causing permanent damage. Likewise, it is possible to program the voltage so low that the chip cannot continue to operate or even retain state without clocks. Care should be taken with these settings.

Care must be taken when raising the output voltage of the regulator rapidly. This can cause large currents to flow into the output cap of the regulator up to the limits of the input supply. When the input supply capability is exceeded, this can cause an input supply dip that may affect other regulators on the same supply. Therefore, the rate of voltage change on the output of the regulator should be limited. When powering up the regulator, the integrated current limiter controls the ramp rate. This limiter is only effective when transitioning from the off state of the regulator (bypassed or power gated).

However, in a DVFS situation, the same high rate of change can occur if the target voltage is raised rapidly by software. To limit the rate of change, the hardware controlling the regulator effects a piecewise linear ramp by stepping the output voltage in 25mV steps until the desired output voltage is reached. The slope of the ramp is controlled by the time spent at each 25mV step and is controlled by the step time field in the PMU_MISC2 register. The same situation is not a problem when the output voltage is dropped as the load pulls down the output cap. As a result, any reduction in the programmed regulator target voltage is immediately effective with the actual supply voltage falling at a rate controlled by the load on the regulator.

51.3 Analog LDO Regulators

There are two analog regulators described here.

51.3.1 LDO 1P1

The LDO_1P1 module on the chip implements a programmable linear-regulator function from a higher analog supply voltage (2.8 V–3.3 V) to produce a nominal 1.1 V output voltage.

The output of the regulator can be programmed in 25 mV steps from 0.8 V to 1.4 V. The regulator has been designed to be stable with a minimum external low-ESR decoupling capacitance, though the actual capacitance required should be determined by the application. A programmable brownout detector is included in the regulator which can be used by the system to determine when the load capability of the regulator is being exceeded, so the necessary steps can be taken.

Current limiting can be enabled by setting the PMU_REG_1P1[ENABLE_ILIMIT] bit to allow for in-rush current requirements during startup if needed. Active pulldown can also be enabled by setting the PMU_REG_1P1[ENABLE_PULLDOWN] bit for systems requiring this feature.

51.3.2 LDO 2P5

The LDO_2P5 module on the chip implements a programmable linear-regulator function from a higher analog supply voltage (2.8V-3.3V) to produce a nominal 2.5V output voltage.

The output of the regulator can be programmed in 25mV steps from 2.0V to 2.75V. The regulator has been designed to be stable with a minimum external low-ESR decoupling capacitance, though the actual capacitance required should be determined by the application. A programmable brown-out detector is included in the regulator which can be used by the system to determine when the load capability of the regulator is being exceeded to take the necessary steps.

Current-limiting can be enabled by setting the REG_PMU_2P5[ENABLE_ILIMIT] bit to allow for in-rush current requirements during start-up if needed. Active-pulldown can also be enabled by setting the REG_PMU_2P5[ENABLE_PULLDOWN] bit for systems requiring this feature.

51.3.3 Low Power Operation

The 2.5 V LDO includes an alternate, self-biased, low-precision, weak regulator which can be enabled for applications needing to keep the 2.5 V output voltage alive during low-power modes where the main regulator and its associated global bandgap reference module are disabled.

The output of this weak regulator is not programmable and is a function of its input power supply as well as load current. The low-power mode is enabled by setting high the PMU_REG_2P5[ENABLE_WEAK_LINREG] bit of the regulator. It is recommended that the following sequence be followed to enable this mode:

1. Throttle down the 2.5 V attached load to its low-power maintain state.
2. Disable the main 2.5 V regulator driver by clearing the PMU_REG_2P5[ENABLE_LINREG] bit.
3. Enable the weak 2.5 V regulator by setting the PMU_REG_2P5[ENABLE_WEAK_LINREG] bit.

To go back to full-power operation, reverse the steps outlined above. Note that the external decoupling cap is supporting the power supply between steps 2 and 3. Therefore step 3 should happen appropriately in time relative to the discharge of the supporting capacitor.

51.4 USB LDO Regulator

The USB_LDO module on the chip implements a programmable linear-regulator function from the USB VBUS voltages (typically 5 V) to produce a nominal 3.0 V output voltage.

The output of the regulator can be programmed in 25 mV steps, from 2.625V to 3.4 V . The regulator has been designed to be stable with a minimum external low-ESR decoupling capacitor of 4.7 μF, though the actual capacitance required should be determined by the application. A programmable brownout detector is included in the regulator which can be used by the system to determine when the load capability of the regulator is being exceeded, so the necessary steps can be taken. This regulator has a built-in power mux which allows the user to choose to run the regulator from either VBUS supply when both are present. If only one of the VBUS voltages is present, then the regulator automatically selects this supply. Current limit is also included to help the system meet in-rush current targets.

Upon attachment of VBUS, this regulator starts up in a low-power, self-preservation mode to prevent over-voltage conditions on the chip. It is expected that the user transition to full regulation by enabling the regulator and disabling the in-rush current limits via its control registers. Upon VBUS removal, it is further expected that the regulator controls are returned to their reset state.

51.5 SNVS Regulator

The SNVS regulator takes the SNVS_IN supply and generates the SNVS_CAP supply, which powers the real time clock and SNVS blocks.

If VDDHIGH_IN is present, then the SNVS_IN supply is internally shorted to the VDDHIGH_IN supply to allow coin cell recharging if necessary. The output voltage is roughly one third of SNVS_IN.

VDDHIGH_IN can be separated from SNVS_IN by setting the PMU_MISC0[discon_high_snvs] bit.

51.6 Power Modes

51.6.1 Reverse Well Biasing

The reverse well biasing module on the chip includes a self-clocked/self-regulating charge-pump circuit to generate a negative bias voltage for the floating PWELL, and a low-power regulator to generate a positive bias voltage for the NWELL of digital logic cells on the SOC power domain.

Static leakage reduction can be achieved through the use of these reverse well bias voltages. Typical power consumption of the module is 50 µA when driving a 10-nF purely capacitive load.

51.7 PMU Memory Map/Register Definition

The register definitions that affect the behavior of the digital LDO regulators follow.

NOTE

Some of the registers are collections of bits that affect multiple components on the chip. Those that are not pertinent to this chapter have comments in the related register bitfields.

If a full description is desired, please consult the full register programming reference in the related block.

PMU memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_8110	Regulator 1P1 Register (PMU_REG_1P1)	32	R/W	0000_1073h	51.7.1/4523
20C_8114	Regulator 1P1 Register (PMU_REG_1P1_SET)	32	R/W	0000_1073h	51.7.1/4523
20C_8118	Regulator 1P1 Register (PMU_REG_1P1_CLR)	32	R/W	0000_1073h	51.7.1/4523
20C_811C	Regulator 1P1 Register (PMU_REG_1P1_TOG)	32	R/W	0000_1073h	51.7.1/4523
20C_8120	Regulator 3P0 Register (PMU_REG_3P0)	32	R/W	0000_0F74h	51.7.2/4526
20C_8124	Regulator 3P0 Register (PMU_REG_3P0_SET)	32	R/W	0000_0F74h	51.7.2/4526
20C_8128	Regulator 3P0 Register (PMU_REG_3P0_CLR)	32	R/W	0000_0F74h	51.7.2/4526
20C_812C	Regulator 3P0 Register (PMU_REG_3P0_TOG)	32	R/W	0000_0F74h	51.7.2/4526
20C_8130	Regulator 2P5 Register (PMU_REG_2P5)	32	R/W	0000_5071h	51.7.3/4528
20C_8134	Regulator 2P5 Register (PMU_REG_2P5_SET)	32	R/W	0000_5071h	51.7.3/4528
20C_8138	Regulator 2P5 Register (PMU_REG_2P5_CLR)	32	R/W	0000_5071h	51.7.3/4528
20C_813C	Regulator 2P5 Register (PMU_REG_2P5_TOG)	32	R/W	0000_5071h	51.7.3/4528
20C_8140	Digital Regulator Core Register (PMU_REG_CORE)	32	R/W	0040_2010h	51.7.4/4530
20C_8144	Digital Regulator Core Register (PMU_REG_CORE_SET)	32	R/W	0040_2010h	51.7.4/4530

Table continues on the next page...

PMU memory map (continued)

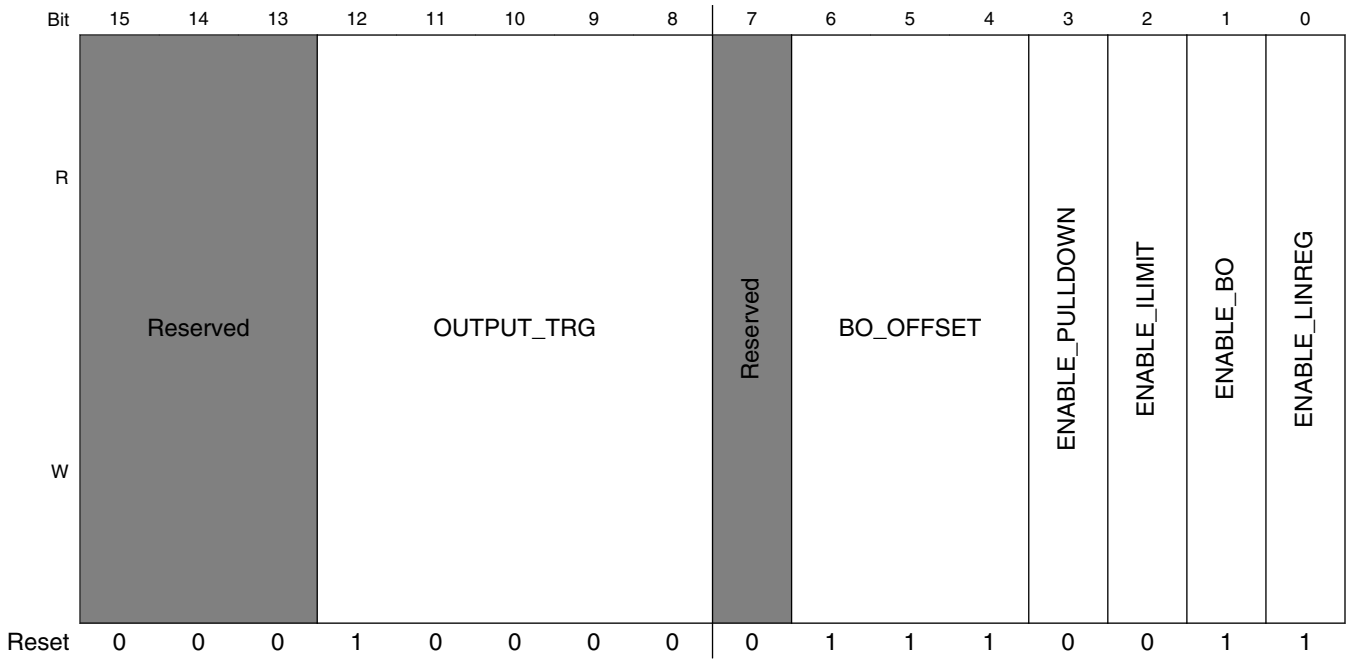
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_8148	Digital Regulator Core Register (PMU_REG_CORE_CLR)	32	R/W	0040_2010h	51.7.4/4530
20C_814C	Digital Regulator Core Register (PMU_REG_CORE_TOG)	32	R/W	0040_2010h	51.7.4/4530
20C_8150	Miscellaneous Register 0 (PMU_MISC0)	32	R/W	0400_0000h	51.7.5/4533
20C_8154	Miscellaneous Register 0 (PMU_MISC0_SET)	32	R/W	0400_0000h	51.7.5/4533
20C_8158	Miscellaneous Register 0 (PMU_MISC0_CLR)	32	R/W	0400_0000h	51.7.5/4533
20C_815C	Miscellaneous Register 0 (PMU_MISC0_TOG)	32	R/W	0400_0000h	51.7.5/4533
20C_8160	Miscellaneous Register 1 (PMU_MISC1)	32	R/W	0000_0000h	51.7.6/4536
20C_8164	Miscellaneous Register 1 (PMU_MISC1_SET)	32	R/W	0000_0000h	51.7.6/4536
20C_8168	Miscellaneous Register 1 (PMU_MISC1_CLR)	32	R/W	0000_0000h	51.7.6/4536
20C_816C	Miscellaneous Register 1 (PMU_MISC1_TOG)	32	R/W	0000_0000h	51.7.6/4536
20C_8170	Miscellaneous Control Register (PMU_MISC2)	32	R/W	0027_2727h	51.7.7/4538
20C_8174	Miscellaneous Control Register (PMU_MISC2_SET)	32	R/W	0027_2727h	51.7.7/4538
20C_8178	Miscellaneous Control Register (PMU_MISC2_CLR)	32	R/W	0027_2727h	51.7.7/4538
20C_817C	Miscellaneous Control Register (PMU_MISC2_TOG)	32	R/W	0027_2727h	51.7.7/4538

51.7.1 Regulator 1P1 Register (PMU_REG_1P1*n*)

This register defines the control and status bits for the 1.1V regulator. This regulator is designed to power the digital portions of the analog cells.

Address: 20C_8000h base + 110h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved														OK_VDD1P1	BO_VDD1P1
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



PMU_REG_1P1*n* field descriptions

Field	Description
31–18 -	This field is reserved.
17 OK_VDD1P1	Status bit that signals when the regulator output is ok. 1 = regulator output > brownout target
16 BO_VDD1P1	Status bit that signals when a brownout is detected on the regulator output.
15–13 -	This field is reserved.
12–8 OUTPUT_TRG	Control bits to adjust the regulator output voltage. Each LSB is worth 25mV. Programming examples are detailed below. Other output target voltages may be interpolated from these examples. Choices must be in this range: 0x1b >= output_trg >= 0x04 **NOTE:** There may be reduced chip functionality or reliability at the extremes of the programming range. 0x04 0.8V 0x10 1.1V 0x1b 1.375V
7 -	This field is reserved.
6–4 BO_OFFSET	Control bits to adjust the regulator brownout offset voltage in 25mV steps. The reset brown-offset is 175mV below the programmed target code. Brownout target = OUTPUT_TRG - BO_OFFSET. Some steps may be irrelevant because of input supply limitations or load operation.
3 ENABLE_ PULLDOWN	Control bit to enable the pull-down circuitry in the regulator
2 ENABLE_ILIMIT	Control bit to enable the current-limit circuitry in the regulator.

Table continues on the next page...

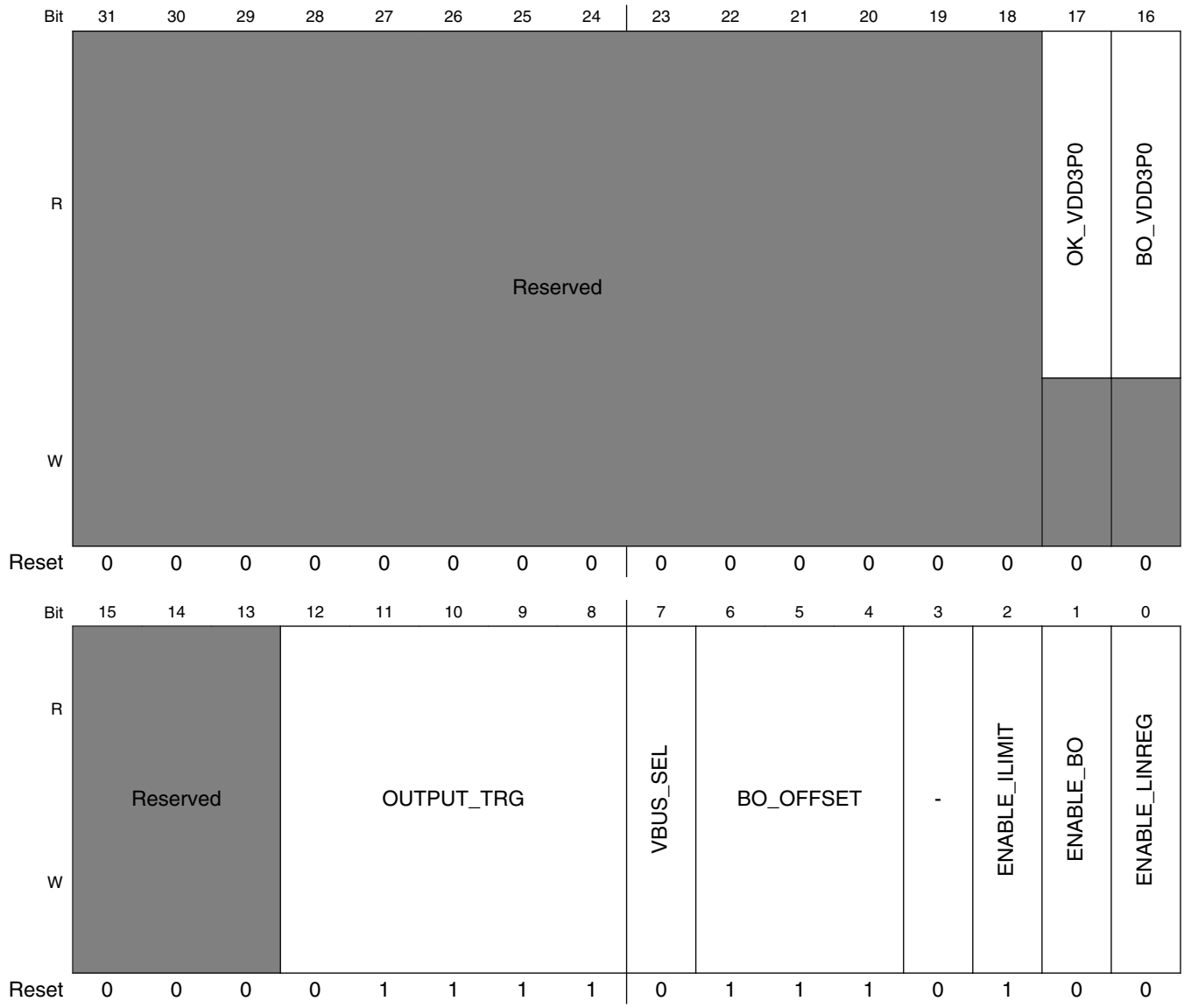
PMU_REG_1P1*n* field descriptions (continued)

Field	Description
1 ENABLE_BO	Control bit to enable the brownout circuitry in the regulator.
0 ENABLE_ LINREG	Control bit to enable the regulator output.

51.7.2 Regulator 3P0 Register (PMU_REG_3P0*n*)

This register defines the control and status bits for the 3.0V regulator powered by the host USB VBUS pin.

Address: 20C_8000h base + 120h offset + (4d × i), where i=0d to 3d



PMU_REG_3P0*n* field descriptions

Field	Description
31–18 -	This field is reserved.

Table continues on the next page...

PMU_REG_3P0*n* field descriptions (continued)

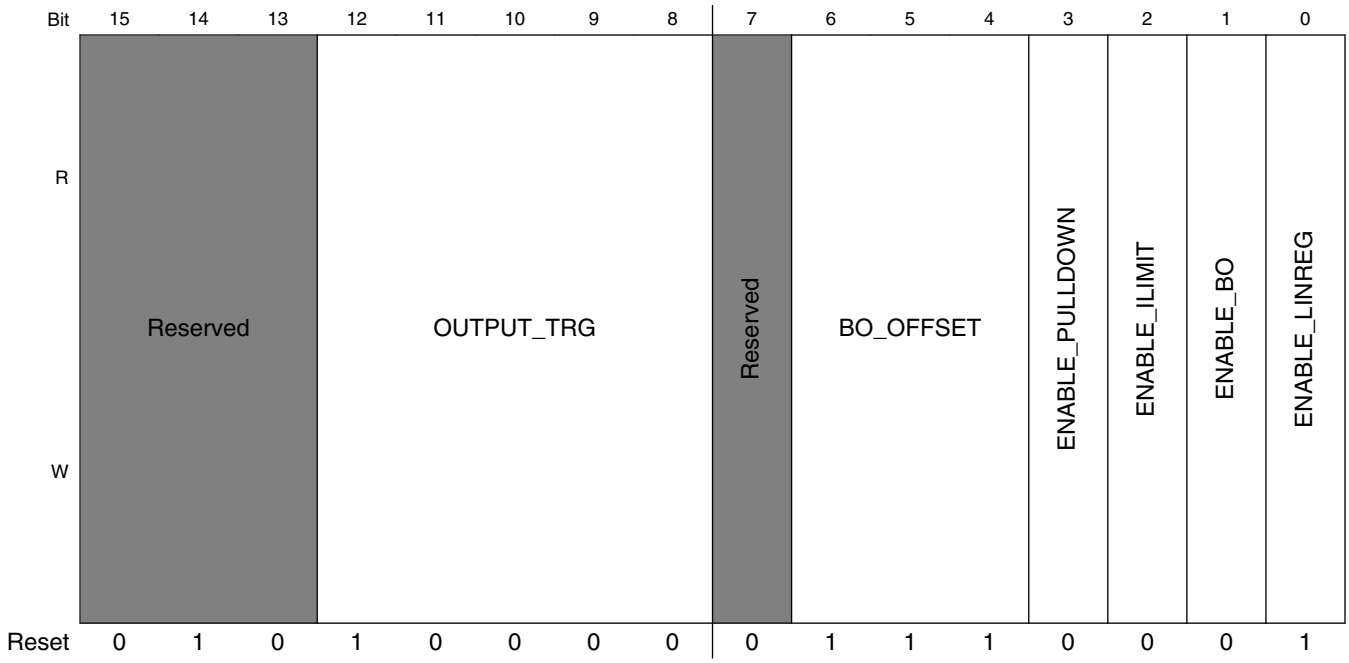
Field	Description
17 OK_VDD3P0	Status bit that signals when the regulator output is ok. 1 = regulator output > brownout target
16 BO_VDD3P0	Status bit that signals when a brownout is detected on the regulator output.
15–13 -	This field is reserved.
12–8 OUTPUT_TRG	Control bits to adjust the regulator output voltage. Each LSB is worth 25mV. Programming examples are detailed below. Other output target voltages may be interpolated from these examples. **NOTE:** There may be reduced chip functionality or reliability at the extremes of the programming range. 0x00 2.625V 0x0f 3.000V 0x1f 3.400V
7 VBUS_SEL	Select input voltage source for LDO_3P0 from either USB_H1_VBUS or USB_OTG_VBUS. If only one of the two VBUS voltages is present, it will automatically be selected. 0 **USB_H1_VBUS** — Utilize VBUS H1 for power 1 **USB_OTG_VBUS** — Utilize VBUS OTG for power
6–4 BO_OFFSET	Control bits to adjust the regulator brownout offset voltage in 25mV steps. The reset brown-offset is 175mV below the programmed target code. Brownout target = OUTPUT_TRG - BO_OFFSET. Some steps may not be relevant because of input supply limitations or load operation.
3 -	Reserved
2 ENABLE_ILIMIT	Control bit to enable the current-limit circuitry in the regulator.
1 ENABLE_BO	Control bit to enable the brownout circuitry in the regulator.
0 ENABLE_ LINREG	Control bit to enable the regulator output to be set by the programmed target voltage setting and internal bandgap reference.

51.7.3 Regulator 2P5 Register (PMU_REG_2P5*n*)

This register defines the control and status bits for the 2.5V regulator.

Address: 20C_8000h base + 130h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved													ENABLE_WEAK_LINREG	OK_VDD2P5	BO_VDD2P5
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



PMU_REG_2P5*n* field descriptions

Field	Description
31–19 -	This field is reserved.
18 ENABLE_ WEAK_LINREG	Enables the weak 2p5 regulator. This low power regulator is used when the main 2p5 regulator is disabled to keep the 2.5V output roughly at 2.5V. Scales directly with the value of VDDHIGH_IN.
17 OK_VDD2P5	Status bit that signals when the regulator output is ok. 1 = regulator output > brownout target
16 BO_VDD2P5	Status bit that signals when a brownout is detected on the regulator output.
15–13 -	This field is reserved.
12–8 OUTPUT_TRG	Control bits to adjust the regulator output voltage. Each LSB is worth 25mV. Programming examples are detailed below. Other output target voltages may be interpolated from these examples. **NOTE:** There may be reduced chip functionality or reliability at the extremes of the programming range. 0x00 2.10V 0x10 2.50V 0x1f 2.875V
7 -	This field is reserved.
6–4 BO_OFFSET	Control bits to adjust the regulator brownout offset voltage in 25mV steps. The reset brown-offset is 175mV below the programmed target code. Brownout target = OUTPUT_TRG - BO_OFFSET. Some steps may be irrelevant because of input supply limitations or load operation.
3 ENABLE_ PULLDOWN	Control bit to enable the pull-down circuitry in the regulator

Table continues on the next page...

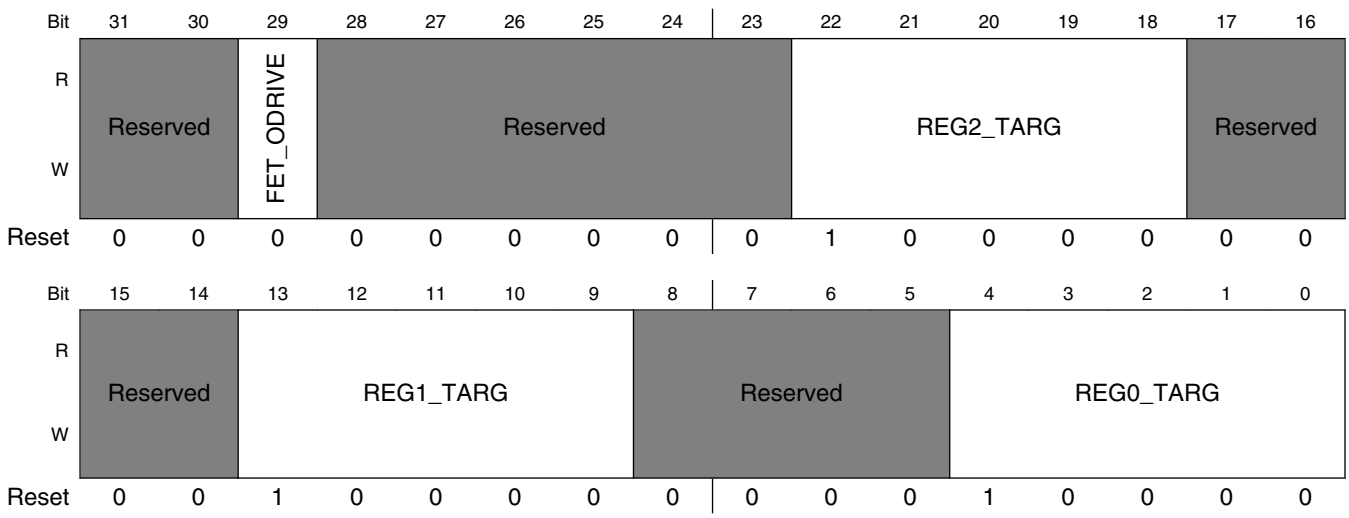
PMU_REG_2P5*n* field descriptions (continued)

Field	Description
2 ENABLE_ILIMIT	Control bit to enable the current-limit circuitry in the regulator.
1 ENABLE_BO	Control bit to enable the brownout circuitry in the regulator.
0 ENABLE_ LINREG	Control bit to enable the regulator output.

51.7.4 Digital Regulator Core Register (PMU_REG_CORE*n*)

This register defines the function of the digital regulators

Address: 20C_8000h base + 140h offset + (4d × i), where i=0d to 3d



PMU_REG_CORE*n* field descriptions

Field	Description
31–30 -	This field is reserved.
29 FET_ODRIVE	If set, increases the gate drive on power gating FETs to reduce leakage in the off state. Care must be taken to apply this bit only when the input supply voltage to the power FET is less than 1.1V. **NOTE:** This bit should only be used in low-power modes where the external input supply voltage is nominally 0.9V.
28–23 -	This field is reserved. Reserved
22–18 REG2_TARG	

Table continues on the next page...

PMU_REG_CORE*n* field descriptions (continued)

Field	Description
	This field defines the target voltage for the SOC power domain. Single-bit increments reflect 25mV core voltage steps. Some steps may not be relevant because of input supply limitations or load operation. **NOTE:** This register is capable of programming an over-voltage condition on the device. Consult the datasheet Operating Ranges table for the allowed voltages. 00000 Power gated off 00001 Target core voltage = 0.725V 00010 Target core voltage = 0.750V 00011 Target core voltage = 0.775V ... 10000 Target core voltage = 1.100V ... 11110 Target core voltage = 1.450V 11111 Power FET switched full on. No regulation.
17–14 -	This field is reserved.
13–9 REG1_TARG	This field defines the target voltage for the VPU/GPU power domain. Single-bit increments reflect 25mV core voltage steps. Some steps may not be relevant because of input supply limitations or load operation. **NOTE:** This register is capable of programming an over-voltage condition on the device. Consult the datasheet Operating Ranges table for the allowed voltages. 00000 Power gated off 00001 Target core voltage = 0.725V 00010 Target core voltage = 0.750V 00011 Target core voltage = 0.775V ... 10000 Target core voltage = 1.100V ... 11110 Target core voltage = 1.450V 11111 Power FET switched full on. No regulation.
8–5 -	This field is reserved.
REG0_TARG	This field defines the target voltage for the Arm core power domain. Single-bit increments reflect 25mV core voltage steps. Some steps may not be relevant because of input supply limitations or load operation. **NOTE:** This register is capable of programming an over-voltage condition on the device. Consult the datasheet Operating Ranges table for the allowed voltages. 00000 Power gated off 00001 Target core voltage = 0.725V 00010 Target core voltage = 0.750V 00011 Target core voltage = 0.775V ... 10000 Target core voltage = 1.100V ...

Table continues on the next page...

PMU_REG_CORE*n* field descriptions (continued)

Field	Description
	11110 Target core voltage = 1.450V 11111 Power FET switched full on. No regulation.

51.7.5 Miscellaneous Register 0 (PMU_MISC0*n*)

This register defines the control and status bits for miscellaneous analog blocks.

Address: 20C_8000h base + 150h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			CLKGATE_DELAY			CLKGATE_CTRL	Reserved					WBCP_VPW_THRESH		OSC_XTALOK_EN	OSC_XTALOK
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OSC_I		discon_high_snvs	STOP_MODE_CONFIG	Reserved				REFTOP_VBGUP	REFTOP_VBGADJ			REFTOP_SELFBIASOFF	Reserved		REFTOP_PWD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMU_MISC0*n* field descriptions

Field	Description
31–29 -	This field is reserved.
28–26 CLKGATE_ DELAY	This field specifies the delay between powering up the XTAL 24MHz clock and releasing the clock to the digital logic inside the analog block. **NOTE:** Do not change the field during a low power event. This is not a field that the user would normally need to modify. **NOTE:** Not related to PMU. 000 0.5ms 001 1.0ms 010 2.0ms 011 3.0ms 100 4.0ms 101 5.0ms 110 6.0ms 111 7.0ms
25 CLKGATE_CTRL	This bit allows disabling the clock gate (always ungated) for the xtal 24MHz clock that clocks the digital logic in the analog block. **NOTE:** Do not change the field during a low power event. This is not a field that the user would normally need to modify. **NOTE:** Not related to PMU. 0 **ALLOW_AUTO_GATE** — Allow the logic to automatically gate the clock when the XTAL is powered down. 1 **NO_AUTO_GATE** — Prevent the logic from ever gating off the clock.
24–20 -	This field is reserved. Always set to zero.
19–18 WBCP_VPW_ THRESH	This signal alters the voltage that the pwell is charged pumped to. 00 **NOMINAL_BIAS** — Nominal output pwell bias voltage. 01 **PLUS_25MV** — Increase pwell output voltage by 25mV. 10 **MINUS_25MV** — Decrease pwell output pwell voltage by 25mV. 11 **MINUS_50MV** — Decrease pwell output pwell voltage by 50mV.
17 OSC_XTALOK_ EN	This bit enables the detector that signals when the 24MHz crystal oscillator is stable. **NOTE:** Not related to PMU, Clocking content
16 OSC_XTALOK	Status bit that signals that the output of the 24-MHz crystal oscillator is stable. Generated from a timer and active detection of the actual frequency. **NOTE:** Not related to PMU, clocking content.
15–14 OSC_I	This field determines the bias current in the 24MHz oscillator. The aim is to start up with the highest bias current, which can be decreased after startup if it is determined to be acceptable. **NOTE:** Not related to PMU.

Table continues on the next page...

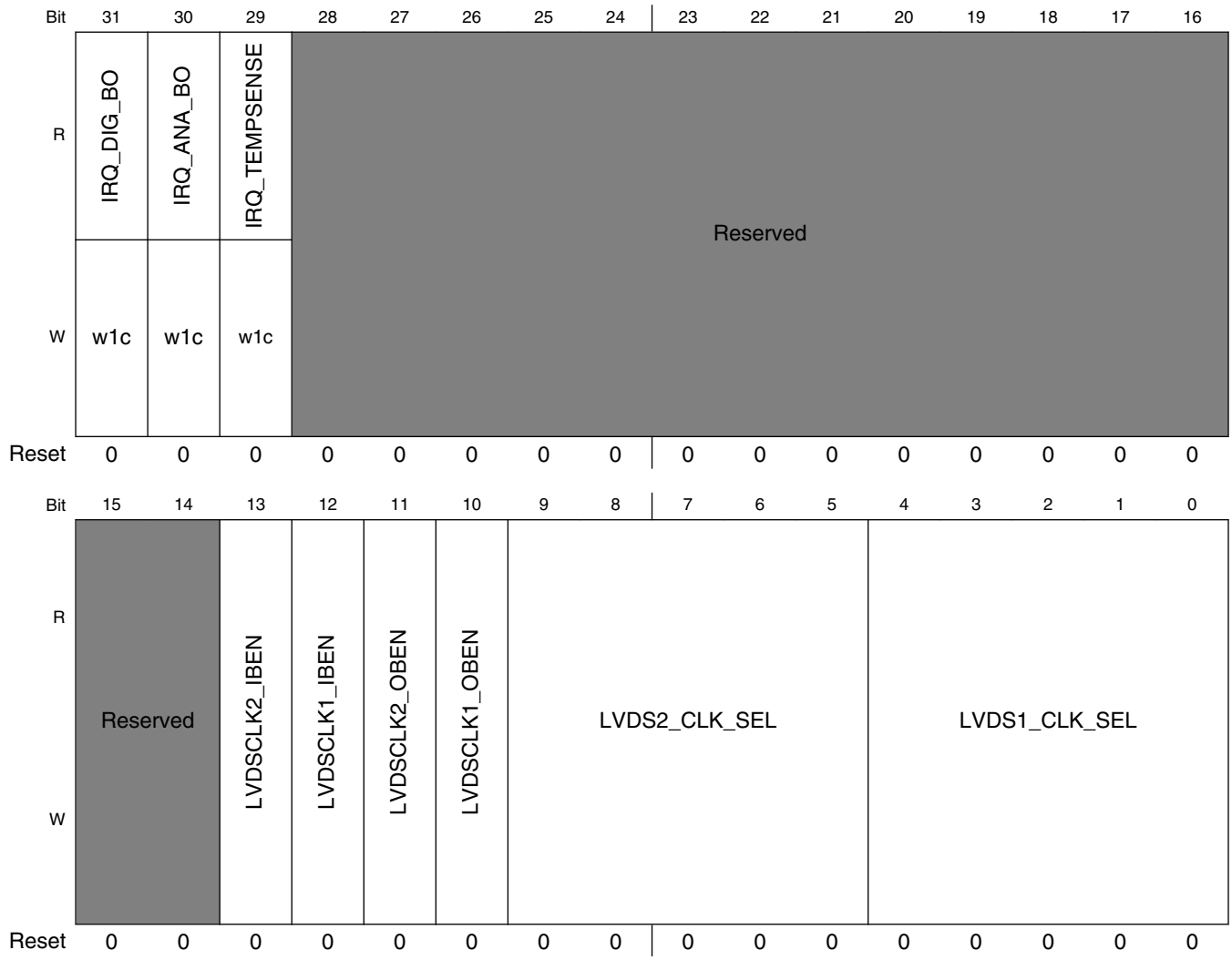
PMU_MISC0*n* field descriptions (continued)

Field	Description
	00 **NOMINAL** — Nominal 01 **MINUS_12_5_PERCENT** — Decrease current by 12.5% 10 **MINUS_25_PERCENT** — Decrease current by 25.0% 11 **MINUS_37_5_PERCENT** — Decrease current by 37.5%
13 discon_high_ snvs	This bit forces the short between VDDHIGH_IN and VSNVS_IN to open when asserted. This is useful in power cases where SNVS_IN > VDDHIGH_IN.
12 STOP_MODE_ CONFIG	Configure the analog behavior in stop mode. 0x0 **DEEP** — Deep Stop Mode - 0x0 All analog except RTC powered down on Stop mode assertion 0x1 **LIGHT** — Light Stop Mode - 0x1 All the analog domain except the LDO_1P1, LDO_2P5, and PLL3 is powered down on STOP mode assertion. If required the CCM can be configured not to power down the oscillator (XTALOSC). PLL3 can be disabled with register settings if desired.
11–8 -	This field is reserved. Reserved
7 REFTOP_ VBGUP	Status bit that signals the analog bandgap voltage is up and stable. 1 - Stable.
6–4 REFTOP_ VBGADJ	000 Nominal VBG 001 VBG+0.78% 010 VBG+1.56% 011 VBG+2.34% 100 VBG-0.78% 101 VBG-1.56% 110 VBG-2.34% 111 VBG-3.12%
3 REFTOP_ SELFBIASOFF	Control bit to disable the self-bias circuit in the analog bandgap. The self-bias circuit is used by the bandgap during startup. This bit should be set after the bandgap has stabilized and is necessary for best noise performance of analog blocks using the outputs of the bandgap. **NOTE:** Value should be returned to zero before removing vddhigh_in or asserting bit 0 of this register (REFTOP_PWD) to assure proper restart of the circuit. 0 Uses coarse bias currents for startup 1 Uses bandgap-based bias currents for best performance.
2–1 -	This field is reserved.
0 REFTOP_PWD	Control bit to power-down the analog bandgap reference circuitry. **NOTE:** A note of caution, the bandgap is necessary for correct operation of most of the LDO, pll, and other analog functions on the die.

51.7.6 Miscellaneous Register 1 (PMU_MISC1*n*)

This register defines the control and status bits for miscellaneous analog blocks. The LVDS1 and LVDS2 controls below control the behavior of the anaclk1/1b and anaclk2/2b LVDS IO's.

Address: 20C_8000h base + 160h offset + (4d × i), where i=0d to 3d



PMU_MISC1*n* field descriptions

Field	Description
31 IRQ_DIG_BO	This status bit is set to one when when any of the digital regulator brownout interrupts assert. Check the regulator status bits to discover which regulator interrupt asserted.
30 IRQ_ANA_BO	This status bit is set to one when when any of the analog regulator brownout interrupts assert. Check the regulator status bits to discover which regulator interrupt asserted.

Table continues on the next page...

PMU_MISC1*n* field descriptions (continued)

Field	Description
29 IRQ_ TEMPSENSE	This status bit is set to one when when the temperature sensor interrupt asserts. **NOTE:** Not related to PMU, Temperature Monitor content.
28–14 -	This field is reserved.
13 LVDSCLK2_ IBEN	This enables the LVDS input buffer for anaclk2/2b. Do not enable input and output buffers simultaneously. **NOTE:** Not related to PMU.
12 LVDSCLK1_ IBEN	This enables the LVDS input buffer for anaclk1/1b. Do not enable input and output buffers simultaneously. **NOTE:** Not related to PMU, Clocking content.
11 LVDSCLK2_ OBEN	This enables the LVDS output buffer for anaclk2/2b. Do not enable input and output buffers simultaneously. **NOTE:** Not related to PMU.
10 LVDSCLK1_ OBEN	This enables the LVDS output buffer for anaclk1/1b. Do not enable input and output buffers simultaneously. **NOTE:** Not related to PMU, clocking content.
9–5 LVDS2_CLK_ SEL	This field selects the clk to be routed to anaclk2/2b. **NOTE:** Not related to PMU. 00000 **ARM_PLL** — ARM PLL 00001 **SYS_PLL** — System PLL 00010 **PFD4** — ref_pfd4_clk == pll2_pfd0_clk 00011 **PFD5** — ref_pfd5_clk == pll2_pfd1_clk 00100 **PFD6** — ref_pfd6_clk == pll2_pfd2_clk 00101 **PFD7** — ref_pfd7_clk == pll2_pfd3_clk 00110 **AUDIO_PLL** — Audio PLL 00111 **VIDEO_PLL** — Video PLL 01000 **MLB_PLL** — MLB PLL 01001 **ETHERNET_REF** — ethernet ref clock (ENET_PLL) 01010 **PCIE_REF** — PCIe ref clock (125M) 01100 **USB1_PLL** — USB1 PLL clock 01101 **USB2_PLL** — USB2 PLL clock 01110 **PFD0** — ref_pfd0_clk == pll3_pfd0_clk 01111 **PFD1** — ref_pfd1_clk == pll3_pfd1_clk 10000 **PFD2** — ref_pfd2_clk == pll3_pfd2_clk 10001 **PFD3** — ref_pfd3_clk == pll3_pfd3_clk 10010 **XTAL** — xtal (24M) 10011 **LVDS1** — LVDS1 (loopback) 10100 **LVDS2** — LVDS2 (not useful) 10101 to 11111 **-** — ref_pfd7_clk == pll2_pfd3_clk

Table continues on the next page...

PMU_MISC1*n* field descriptions (continued)

Field	Description
LVDS1_CLK_SEL	This field selects the clk to be routed to anaclk1/1b. **NOTE:** Not related to PMU. 00000 **ARM_PLL** — ARM PLL 00001 **SYS_PLL** — System PLL 00010 **PFD4** — ref_pfd4_clk == pll2_pfd0_clk 00011 **PFD5** — ref_pfd5_clk == pll2_pfd1_clk 00100 **PFD6** — ref_pfd6_clk == pll2_pfd2_clk 00101 **PFD7** — ref_pfd7_clk == pll2_pfd3_clk 00110 **AUDIO_PLL** — Audio PLL 00111 **VIDEO_PLL** — Video PLL 01000 **MLB_PLL** — MLB PLL 01001 **ETHERNET_REF** — ethernet ref clock (ENET_PLL) 01010 **PCIE_REF** — PCIe ref clock (125M) 01011 **REF_100M** — REF_100M clock 01100 **USB1_PLL** — USB1 PLL clock 01101 **USB2_PLL** — USB2 PLL clock 01110 **PFD0** — ref_pfd0_clk == pll3_pfd0_clk 01111 **PFD1** — ref_pfd1_clk == pll3_pfd1_clk 10000 **PFD2** — ref_pfd2_clk == pll3_pfd2_clk 10001 **PFD3** — ref_pfd3_clk == pll3_pfd3_clk 10010 **XTAL** — xtal (24M) 10011 **LVDS1** — LVDS1 (loopback) 10100 **LVDS2** — LVDS2 (not useful) 10101 to 11111 **-** — ref_pfd7_clk == pll2_pfd3_clk

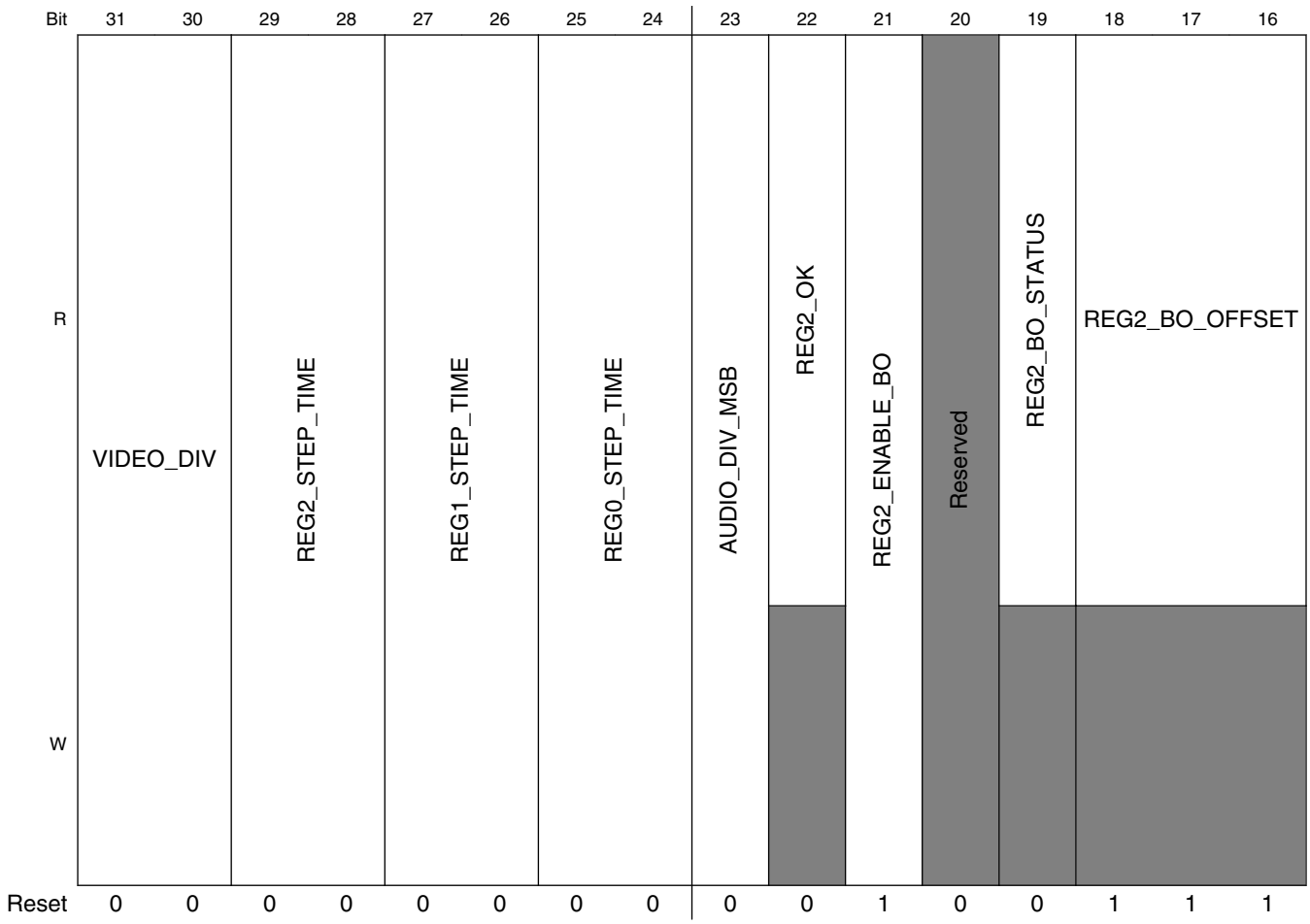
51.7.7 Miscellaneous Control Register (PMU_MISC2*n*)

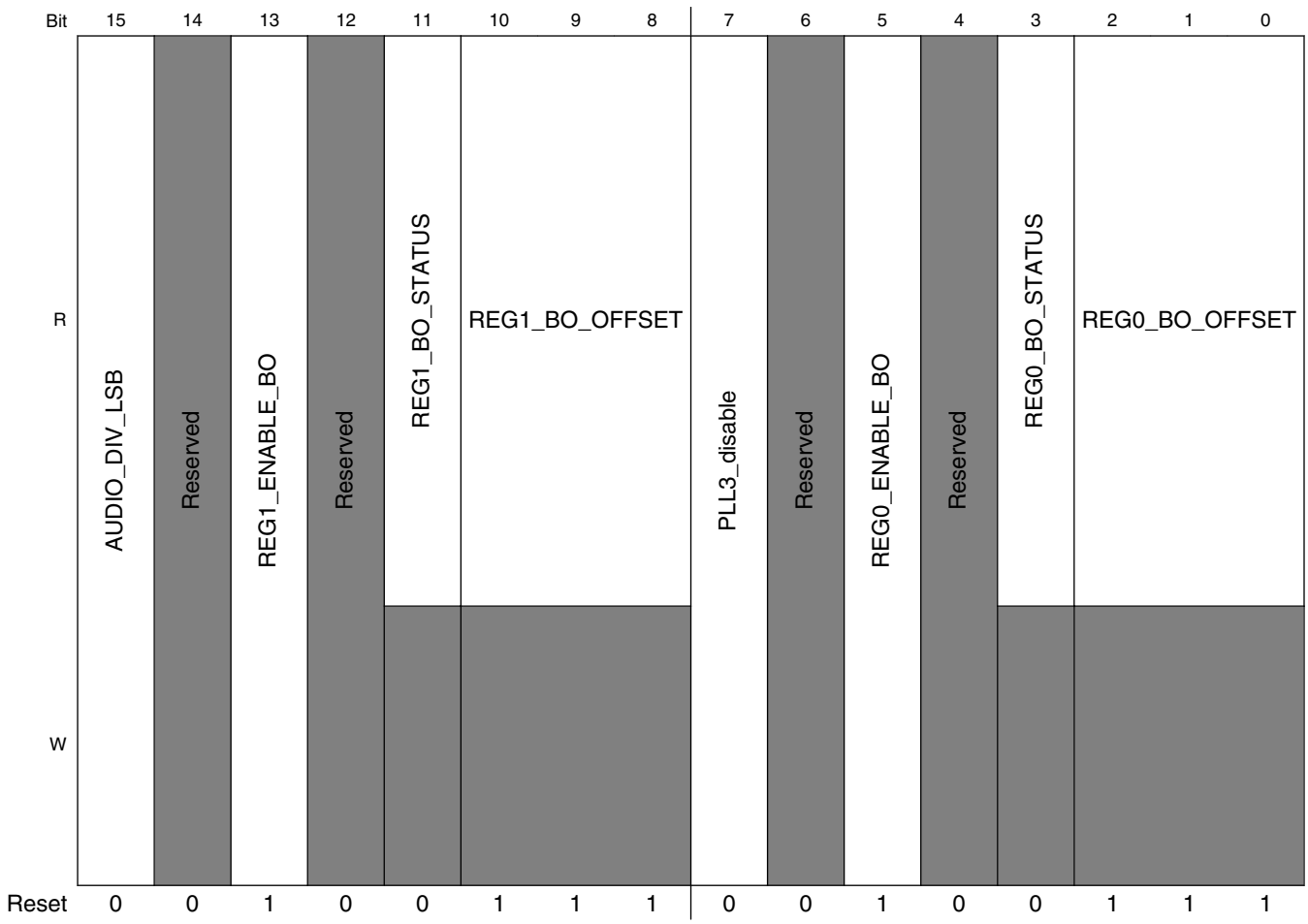
This register defines the control for miscellaneous PMU Analog blocks.

NOTE

This register is shared with CCM.

Address: 20C_8000h base + 170h offset + (4d × i), where i=0d to 3d





PMU_MISC2*n* field descriptions

Field	Description
31–30 VIDEO_DIV	Post-divider for video. The output clock of the video PLL should be gated prior to changing this divider to prevent glitches. This divider is feed by PLL_VIDEOn[POST_DIV_SELECT] to achieve division ratios of /1, /2, /4, /8, and /16. **NOTE:** Not related to PMU. See Clock Controller Module (CCM) for more information. 00 divide by 1 (Default) 01 divide by 2 10 divide by 1 11 divide by 4
29–28 REG2_STEP_TIME	Number of clock periods (24MHz clock). 00 **64_CLOCKS** — 64 01 **128_CLOCKS** — 128 10 **256_CLOCKS** — 256 11 **512_CLOCKS** — 512
27–26 REG1_STEP_TIME	Number of clock periods (24MHz clock). 00 **64_CLOCKS** — 64 01 **128_CLOCKS** — 128

Table continues on the next page...

PMU_MISC2*n* field descriptions (continued)

Field	Description
	10 **256_CLOCKS** — 256 11 **512_CLOCKS** — 512
25–24 REG0_STEP_TIME	Number of clock periods (24MHz clock). 00 **64_CLOCKS** — 64 01 **128_CLOCKS** — 128 10 **256_CLOCKS** — 256 11 **512_CLOCKS** — 512
23 AUDIO_DIV_MSB	MSB of Post-divider for Audio PLL. The output clock of the video PLL should be gated prior to changing this divider to prevent glitches. This divider is feed by PLL_AUDIOn[POST_DIV_SELECT] to achieve division ratios of /1, /2, /4, /8, and /16. **NOTE:** MSB bit value pertains to the first bit, please program the LSB bit (bit 15) as well to change divider value **NOTE:** Not related to PMU. See Clock Controller Module (CCM) for more information. 00 divide by 1 (Default) 01 divide by 2 10 divide by 1 11 divide by 4
22 REG2_OK	Signals that the voltage is above the brownout level for the SOC supply. 1 = regulator output > brownout_target
21 REG2_ENABLE_BO	Enables the brownout detection.
20 -	This field is reserved.
19 REG2_BO_STATUS	Reg2 brownout status bit.
18–16 REG2_BO_OFFSET	This field defines the brown out voltage offset for the xPU power domain. IRQ_DIG_BO is also asserted. Single-bit increments reflect 25mV brownout voltage steps. The reset brown-offset is 175mV below the programmed target code. Brownout target = OUTPUT_TRG - BO_OFFSET. Some steps may be irrelevant because of input supply limitations or load operation. 100 Brownout offset = 0.100V 111 Brownout offset = 0.175V
15 AUDIO_DIV_LSB	LSB of Post-divider for Audio PLL. The output clock of the video PLL should be gated prior to changing this divider to prevent glitches. This divider is feed by PLL_AUDIOn[POST_DIV_SELECT] to achieve division ratios of /1, /2, /4, /8, and /16. **NOTE:** LSB bit value pertains to the last bit, please program the MSB bit (bit 23) as well, to change divider value **NOTE:** Not related to PMU. See Clock Controller Module (CCM) for more information. 00 divide by 1 (Default) 01 divide by 2 10 divide by 1 11 divide by 4

Table continues on the next page...

PMU_MISC2*n* field descriptions (continued)

Field	Description
14 - Reserved	This field is reserved.
13 REG1_ENABLE_BO	Enables the brownout detection.
12 -	This field is reserved.
11 REG1_BO_STATUS	Reg1 brownout status bit. 1 Brownout, supply is below target minus brownout offset.
10–8 REG1_BO_OFFSET	This field defines the brown out voltage offset for the xPU power domain. IRQ_DIG_BO is also asserted. Single-bit increments reflect 25mV brownout voltage steps. The reset brown-offset is 175mV below the programmed target code. Brownout target = OUTPUT_TRG - BO_OFFSET. Some steps may be irrelevant because of input supply limitations or load operation. 100 Brownout offset = 0.100V 111 Brownout offset = 0.175V
7 PLL3_disable	Default value of "0". Should be set to "1" to turn off the USB-PLL(PLL3) in run mode. **NOTE:** Not related to PMU. See Clock Controller Module (CCM) for more information.
6 -	This field is reserved.
5 REG0_ENABLE_BO	Enables the brownout detection.
4 -	This field is reserved.
3 REG0_BO_STATUS	Reg0 brownout status bit. 1 Brownout, supply is below target minus brownout offset.
REG0_BO_OFFSET	This field defines the brown out voltage offset for the CORE power domain. IRQ_DIG_BO is also asserted. Single-bit increments reflect 25mV brownout voltage steps. Some steps may be irrelevant because of input supply limitations or load operation. 100 Brownout offset = 0.100V 111 Brownout offset = 0.175V

Chapter 52
Pulse Width Modulation (PWM)

52.1 Overview

The Pulse Width Modulation (PWM) has a 16-bit counter, and is optimized to generate sound from stored sample audio images and it can also generate tones. It uses 16-bit resolution and a 4 x 16 data FIFO.

This section presents an overview of the PWM. A block diagram of the PWM module is shown in the figure below.

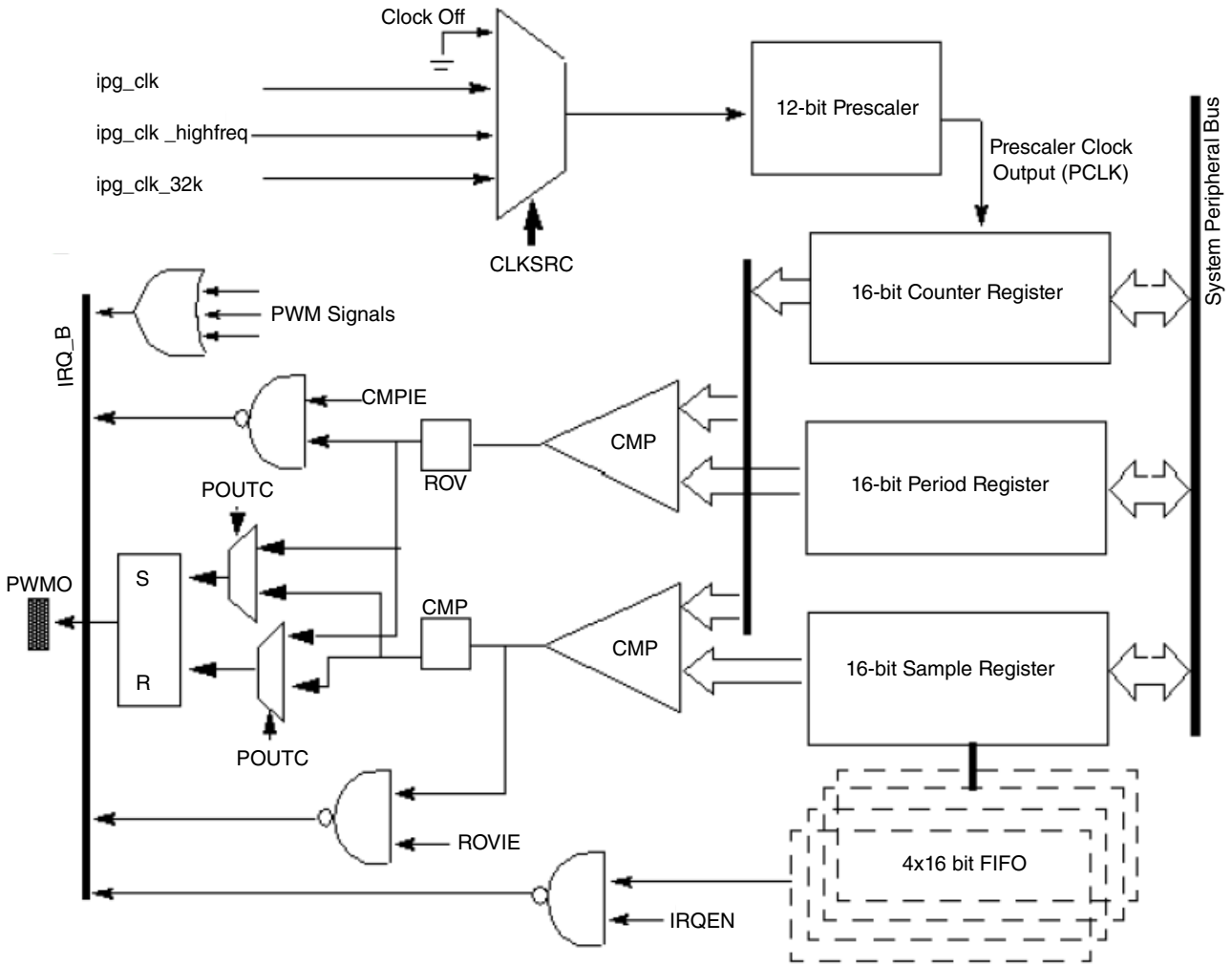


Figure 52-1. Pulse-Width Modulator Block Diagram

The following features characterize the PWM:

- 16-bit up-counter with clock source selection
- 4 x 16 FIFO to minimize interrupt overhead
- 12-bit prescaler for division of clock
- Sound and melody generation
- Active high or active low configured output
- Can be programmed to be active in low-power mode
- Can be programmed to be active in debug mode
- Interrupts at compare and rollover

52.2 External Signals

The PWM follows IP Bus protocol when interfacing with the processor core. PWM does not have any interface signals with any other block inside the chip except for clock and reset inputs from the Clock Control Module (CCM), System Reset Controller (SRC), and interrupt signals to the processor interrupt handler. There is a single output signal.

The following table outlines the external signals.

Table 52-1. PWM External Signals

Signal	Description	Pad	Mode	Direction
PWM1_OUT	This is the PWM1 functional output of the PWM. A modulated signal of the block is observed at this pin. It can be viewed as a clock signal whose period and duty cycle can be varied with different settings of the cycle of 50%.	DISP0_DAT8	ALT2	O
		GPIO_9	ALT4	
		SD1_DAT3	ALT3	
PWM2_OUT	This is the PWM2 functional output of the PWM. A modulated signal of the block is observed at this pin. It can be viewed as a clock signal whose period and duty cycle can be varied with different settings of the cycle of 50%.	DISP0_DAT9	ALT2	O
		GPIO_1	ALT4	
		SD1_DAT2	ALT3	
PWM3_OUT	This is the PWM3 functional output of the PWM. A modulated signal of the block is observed at this pin. It can be viewed as a clock signal whose period and duty cycle can be varied with different settings of the cycle of 50%.	SD1_DAT1	ALT2	O
		SD4_DAT1	ALT2	
PWM4_OUT	This is the PWM4 functional output of the PWM. A modulated signal of the block is observed at this pin. It can be viewed as a clock signal whose period and duty cycle can be varied with different settings of the cycle of 50%.	SD1_CMD	ALT2	O
		SD4_DAT2	ALT2	

52.3 Clocks

The table found here describes the clock sources for PWM. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 52-2. PWM Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock

Table continues on the next page...

Table 52-2. PWM Clocks (continued)

Clock name	Clock Root	Description
ipg_clk_32k	ckil_sync_clk_root	low-frequency reference clock (32 kHz)
ipg_clk_highfreq	perclk_clk_root	high-frequency reference clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

The clock that feeds the prescaler can be selected from:

- High-frequency reference clock (ipg_clk_highfreq) pat_ref or CKIH

  This is a high frequency clock, provided by the Clock Control Module (CCM). This clock should be on in the low power mode when the ipg_clk is turned off. Thus, the PWM can be run on this clock in the low power mode. The CCM is expected to provide this clock after synchronizing it to ahb_clk in the normal functional mode and then switch to the unsynchronized version in the low power mode.

- Low-frequency reference clock (ipg_clk_32k, CKIL)

  This is the 32 KHz low reference clock which is provided by the CCM. This clock should be on in the low power mode when ipg_clk is turned off. Thus, PWM can be run on this clock in the low power mode. The CCM is expected to provide this clock after synchronizing it to ahb_clk in the normal functional mode and then switch to the unsynchronized version in the low power mode.

- Peripheral clock (ipg_clk)

  This clock should be on in normal operations. In low power mode, it can be switched off.

- Peripheral access clock (ipg_clk_s)

  This clock is used for register read/write.

The clock input source is determined by the PWM control register field PWM_CR[CLKSRC]. The CLKSRC value should only be changed when the PWM is disabled.

A change in the value of the PRESCALER field of the control register is immediately reflected on its output clock frequency.

52.4 Functional Description

The following sections detail the PWM operation and function.

52.4.1 Operation

The output of the PWM is a toggling signal whose frequency and duty cycle can be modulated by programming the appropriate registers. It has a 16-bit up counter which counts from 0x0000 until the counter value equals the PWM_PR + 1. After this match occurs the counter is reset to 0x0000.

At the beginning of a count period cycle, the PWMO pin is set to one (default) and the counter begins counting up from 0x0000. The sample value in the sample FIFO is compared on each count of prescaler clock. When the sample and count values match, the PWMO signal is cleared to zero (default). The counter continues counting until the period match occurs and subsequently another period cycle begins.

When the PWM is enabled, the counter starts running and generates an output with the reset values in the period and sample registers. It is recommended that the programming of these registers be done before PWM is enabled.

A hardware reset results in all the PWM count and sample registers being cleared and the FIFO being flushed. The control register shows that FIFO is empty and it can be written into, and the PWM is disabled. A software reset has the same results, however the state of the DBGEN, STOPEN, DOZEN, and WAITEN bits in the control register are not affected. Software reset can be asserted even when the PWM is in disabled state.

52.4.1.1 FIFO

Digital sample values can be loaded into the pulse-width modulator as 16-bit words. The endianess can be changed using the BCTR and HCTR bits of the control register. A 4-word (16-bit) FIFO minimizes interrupt overhead. A maskable interrupt is generated when the number of data words fall below the water level set by the FWM field in the control register.

A write to the PWM_SAR sample register results in the value being stored into the FIFO if it is not full. A write when the FIFO is full sets FWE (FIFO write error) bit in the status register and the FIFO contents remain unchanged. The FIFO can be written at any time, but can be read only when the PWM is enabled. The PWM_SR[FIFOAV] field shows how many data words are currently contained in the FIFO and whether or not it can be written into.

A read on the sample register yields the current FIFO value that is being used, or will be used, by the PWM for generation on the output signal. Therefore, a write and a subsequent read on the sample register may result in different values being obtained.

52.4.1.2 Rollover and Compare Event

The counter is reset to 0x0000 after its value equals the PWM_PR[PERIOD] + 1 and resumes counting thereafter. This event is referred to as a rollover. For example, if PWM_PR[PERIOD] = 0x0000, the counter is reset when it equals 0x0001. When PWM_PR[PERIOD] = 0xFFFF or 0xFFFE, the counter is reset when it equals 0xFFFF. For more information, see the PWM Period Register (PWM_PR) description.

During a rollover event the output is either set (default), reset or has no effect according to the programming of the POUTC field in the control register. This event can also generate an interrupt if the respective interrupt enable bit is set in the control register.

When the counter value reaches the sample value, the output of the PWM is reset (default), set or has no effect according to the programming of the POUTC field of control register. This event is referred to as a compare event. This event can also generate an interrupt if the respective interrupt enable bit is set in the control register.

If the rollover event sets the PWM output signal, the compare event will reset it and vice versa for a particular programming configuration of POUTC field.

52.4.1.3 Low Power Mode Behavior

In low power mode, if the clock from the selected clock source is available, the PWM counter continues to run and an output is produced, depending on whether the control bit for that mode is set or not. In the absence of the clock itself, or if the corresponding low power bit in the control register is 0, the counter is reset and resumes counting when it exits the low power mode.

52.4.1.4 Debug Mode Behavior

In debug mode, PWM has the option of continuing to run or be halted. If the DBGEN bit is not set in the PWM_PWMCR, the PWM is halted. If the DBGEN bit is set, then the PWM will continue to run in the debug mode.

52.5 Enable Sequence for the PWM

The sequence found here should be used to enable the PWM.

1. Configure the desired settings for the PWM Control Register (PWMx_PWMCR) while keeping the PWM disabled (PWMx_PWMCR[0]=0).
2. Enable the desired interrupts in the PWM Interrupt Register (PWMx_PWMIR).
3. One to three initial samples may be written to the PWM Sample Register (PWMx_PWMSAR). The initial sample values will be loaded into the PWM FIFO even if the PWM is not yet enabled. Do not write a 4th sample because the FIFO will become full and trigger a FIFO Write Error (FWE). This error will prevent the PWM from starting once it is enabled.
4. Check the FIFO Write Error status bit (FWE), the Compare status bit (CMP) and the Roll-over status bit (ROV) in the PWM Status Register (PWMx_PWMSR) to make sure they are all zero. Any non-zero status bits should be cleared by writing a 1 to them.
5. Write the desired period to the PWM Period Register (PWMx_PWMPR).
6. Enable the PWM by writing a 1 to the PWM Enable bit, PWMx_PWMCR[0], while maintaining the other register bits in their previously configured state.

52.6 Disable Sequence for the PWM

The PWM can be disabled at any time by clearing the PWM enable bit, PWMx_PWMCR[0] to 0.

Any data remaining in the FIFO will not be produced at the PWM output after the PWM has been disabled and will remain in the FIFO until the PWM is enabled again. A software reset (setting PWMx_PWMCR[3] to 1) or a hardware reset will clear the FIFO and any remaining data will be lost.

52.7 PWM Memory Map/Register Definition

The PWM includes six user-accessible 32-bit registers.

PWM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
208_0000	PWM Control Register (PWM1_PWMCR)	32	R/W	0000_0000h	52.7.1/4551
208_0004	PWM Status Register (PWM1_PWMSR)	32	w1c	0000_0008h	52.7.2/4553
208_0008	PWM Interrupt Register (PWM1_PWMIR)	32	R/W	0000_0000h	52.7.3/4554

Table continues on the next page...

PWM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
208_000C	PWM Sample Register (PWM1_PWMSAR)	32	R/W	0000_0000h	52.7.4/4555
208_0010	PWM Period Register (PWM1_PWMPR)	32	R/W	0000_FFFEh	52.7.5/4556
208_0014	PWM Counter Register (PWM1_PWMCNR)	32	R	0000_0000h	52.7.6/4557
208_4000	PWM Control Register (PWM2_PWMCR)	32	R/W	0000_0000h	52.7.1/4551
208_4004	PWM Status Register (PWM2_PWMSR)	32	w1c	0000_0008h	52.7.2/4553
208_4008	PWM Interrupt Register (PWM2_PWMIR)	32	R/W	0000_0000h	52.7.3/4554
208_400C	PWM Sample Register (PWM2_PWMSAR)	32	R/W	0000_0000h	52.7.4/4555
208_4010	PWM Period Register (PWM2_PWMPR)	32	R/W	0000_FFFEh	52.7.5/4556
208_4014	PWM Counter Register (PWM2_PWMCNR)	32	R	0000_0000h	52.7.6/4557
208_8000	PWM Control Register (PWM3_PWMCR)	32	R/W	0000_0000h	52.7.1/4551
208_8004	PWM Status Register (PWM3_PWMSR)	32	w1c	0000_0008h	52.7.2/4553
208_8008	PWM Interrupt Register (PWM3_PWMIR)	32	R/W	0000_0000h	52.7.3/4554
208_800C	PWM Sample Register (PWM3_PWMSAR)	32	R/W	0000_0000h	52.7.4/4555
208_8010	PWM Period Register (PWM3_PWMPR)	32	R/W	0000_FFFEh	52.7.5/4556
208_8014	PWM Counter Register (PWM3_PWMCNR)	32	R	0000_0000h	52.7.6/4557
208_C000	PWM Control Register (PWM4_PWMCR)	32	R/W	0000_0000h	52.7.1/4551
208_C004	PWM Status Register (PWM4_PWMSR)	32	w1c	0000_0008h	52.7.2/4553
208_C008	PWM Interrupt Register (PWM4_PWMIR)	32	R/W	0000_0000h	52.7.3/4554
208_C00C	PWM Sample Register (PWM4_PWMSAR)	32	R/W	0000_0000h	52.7.4/4555
208_C010	PWM Period Register (PWM4_PWMPR)	32	R/W	0000_FFFEh	52.7.5/4556
208_C014	PWM Counter Register (PWM4_PWMCNR)	32	R	0000_0000h	52.7.6/4557

52.7.1 PWM Control Register (PWMx_PWMCR)

The PWM control register (PWM_PWMCR) is used to configure the operating settings of the PWM. It contains the prescaler for the clock division.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				FWM		STOPEN	DOZEN	WAITEN	DBGEN	BCTR	HCTR	POUTC		CLKSRC	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PRESCALER												SWR	REPEAT		EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWMx_PWMCR field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–26 FWM	FIFO Water Mark. These bits are used to set the data level at which the FIFO empty flag will be set and the corresponding interrupt generated 00 FIFO empty flag is set when there are more than or equal to 1 empty slots in FIFO 01 FIFO empty flag is set when there are more than or equal to 2 empty slots in FIFO 10 FIFO empty flag is set when there are more than or equal to 3 empty slots in FIFO 11 FIFO empty flag is set when there are more than or equal to 4 empty slots in FIFO
25 STOPEN	Stop Mode Enable. This bit keeps the PWM functional while in stop mode. When this bit is cleared, the input clock is gated off in stop mode. This bit is not affected by software reset. It is cleared by hardware reset. 0 Inactive in stop mode 1 Active in stop mode
24 DOZEN	Doze Mode Enable. This bit keeps the PWM functional in doze mode. When this bit is cleared, the input clock is gated off in doze mode. This bit is not affected by software reset. It is cleared by hardware reset. 0 Inactive in doze mode 1 Active in doze mode
23 WAITEN	Wait Mode Enable. This bit keeps the PWM functional in wait mode. When this bit is cleared, the input clock is gated off in wait mode. This bit is not affected by software reset. It is cleared by hardware reset.

Table continues on the next page...

PWMx_PWMCR field descriptions (continued)

Field	Description
	0 Inactive in wait mode 1 Active in wait mode
22 DBGEN	Debug Mode Enable. This bit keeps the PWM functional in debug mode. When this bit is cleared, the input clock is gated off in debug mode. This bit is not affected by software reset. It is cleared by hardware reset. 0 Inactive in debug mode 1 Active in debug mode
21 BCTR	Byte Data Swap Control. This bit determines the byte ordering of the 16-bit data when it goes into the FIFO from the sample register. 0 byte ordering remains the same 1 byte ordering is reversed
20 HCTR	Half-word Data Swap Control. This bit determines which half word data from the 32-bit IP Bus interface is written into the lower 16 bits of the sample register. 0 Half word swapping does not take place 1 Half words from write data bus are swapped
19–18 POUTC	PWM Output Configuration. This bit field determines the mode of PWM output on the output pin. 00 Output pin is set at rollover and cleared at comparison 01 Output pin is cleared at rollover and set at comparison 10 PWM output is disconnected 11 PWM output is disconnected
17–16 CLKSRC	Select Clock Source. These bits determine which clock input will be selected for running the counter. After reset the system functional clock is selected. The input clock can also be turned off if these bits are set to 00. This field value should only be changed when the PWM is disabled 00 Clock is off 01 ipg_clk 10 ipg_clk_highfreq 11 ipg_clk_32k
15–4 PRESCALER	Counter Clock Prescaler Value. This bit field determines the value by which the clock will be divided before it goes to the counter. 0x000 Divide by 1 0x001 Divide by 2 0xfff Divide by 4096
3 SWR	Software Reset. PWM is reset when this bit is set to 1. It is a self clearing bit. A write 1 to this bit is a single wait state write cycle. When the block is in reset state this bit is set and is cleared when the reset procedure is over. Setting this bit resets all the registers to their reset values except for the DBGEN, STOPEN, DOZEN, and WAITEN bits in this control register. 0 PWM is out of reset 1 PWM is undergoing reset
2–1 REPEAT	Sample Repeat. This bit field determines the number of times each sample from the FIFO is to be used. 00 Use each sample once 01 Use each sample twice 10 Use each sample four times 11 Use each sample eight times

Table continues on the next page...

PWMx_PWMCR field descriptions (continued)

Field	Description
0 EN	PWM Enable. This bit enables the PWM. If this bit is not enabled, the clock prescaler and the counter is reset. When the PWM is enabled, it begins a new period, the output pin is set to start a new period while the prescaler and counter are released and counting begins. To make the PWM work with softreset and disable/enable, users can do software reset by seting the SWR bit, wait software reset done, configure the registers, and then enable the PWM by setting this bit to "1" Users can also disable/enable the PWM if PWM would like to be stopped and resumed with same registers configurations . 0 PWM disabled 1 PWM enabled

52.7.2 PWM Status Register (PWMx_PWMSR)

The PWM status register (PWM_PWMSR) contains seven bits which display the state of the FIFO and the occurrence of rollover and compare events. The FIFOAV bit is read-only but the other four bits can be cleared by writing 1 to them. The FE, ROV, and CMP bits are associated with FIFO-Empty, Roll-over, and Compare interrupts, respectively.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									FWE	CMP	ROV	FE	FIFOAV		
W										w1c	w1c	w1c	w1c			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

PWMx_PWMSR field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 FWE	FIFO Write Error Status. This bit shows that an attempt has been made to write FIFO when it is full. 0 FIFO write error not occurred 1 FIFO write error occurred
5 CMP	Compare Status. This bit shows that a compare event has occurred. 0 Compare event not occurred 1 Compare event occurred

Table continues on the next page...

PWMx_PWMSR field descriptions (continued)

Field	Description
4 ROV	Roll-over Status. This bit shows that a roll-over event has occurred. 0 Roll-over event not occurred 1 Roll-over event occurred
3 FE	FIFO Empty Status Bit. This bit indicates the FIFO data level in comparison to the water level set by FWM field in the control register. 0 Data level is above water mark 1 When the data level falls below the mark set by FWM field
FIFOAV	FIFO Available. These read-only bits indicate the data level remaining in the FIFO. An attempted write to these bits will not affect their value and no transfer error is generated. 000 No data available 001 1 word of data in FIFO 010 2 words of data in FIFO 011 3 words of data in FIFO 100 4 words of data in FIFO 101 unused 110 unused 111 unused

52.7.3 PWM Interrupt Register (PWMx_PWMIR)

The PWM Interrupt register (PWM_PWMIR) contains three bits which control the generation of the compare, rollover and FIFO empty interrupts.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													CIE	RIE	FIE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWMx_PWMIR field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 CIE	Compare Interrupt Enable. This bit controls the generation of the Compare interrupt.

Table continues on the next page...

PWMx_PWMIR field descriptions (continued)

Field	Description
	0 Compare Interrupt not enabled 1 Compare Interrupt enabled
1 RIE	Roll-over Interrupt Enable. This bit controls the generation of the Rollover interrupt. 0 Roll-over interrupt not enabled 1 Roll-over Interrupt enabled
0 FIE	FIFO Empty Interrupt Enable. This bit controls the generation of the FIFO Empty interrupt. 0 FIFO Empty interrupt disabled 1 FIFO Empty interrupt enabled

52.7.4 PWM Sample Register (PWMx_PWMSAR)

The PWM sample register (PWM_PWMSAR) is the input to the FIFO. 16-bit words are loaded into the FIFO. The FIFO can be written at any time, but can be read only when the PWM is enabled. The PWM will run at the last set duty-cycle setting if all the values of the FIFO has been utilized, until the FIFO is reloaded or the PWM is disabled. When a new value is written, the duty cycle changes after the current period is over.

A value of zero in the sample register will result in the PWMO output signal always being low/high (POUTC = 00 it will be low and POUTC = 01 it will be high), and no output waveform will be produced. If the value in this register is higher than the PERIOD + 1, the output will never be set/reset depending on POUTC value.

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																SAMPLE															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWMx_PWMSAR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SAMPLE	Sample Value. This is the input to the 4x16 FIFO. The value in this register denotes the value of the sample being currently used.

52.7.5 PWM Period Register (PWMx_PWMPR)

The PWM period register (PWM_PWMPR) determines the period of the PWM output signal. After the counter value matches PERIOD + 1, the counter is reset to start another period.

PWMO (Hz) = PCLK(Hz) / (period +2)

A value of zero in the PWM_PWMPR will result in a period of two clock cycles for the output signal. Writing 0xFFFF to this register will achieve the same result as writing 0xFFFE.

A change in the period value due to a write in PWM_PWMPR results in the counter being reset to zero and the start of a new count period.

NOTE

Settings PWM_PWMPR to 0xFFFF when PWMx_PWMCR REPEAT bits are set to non-zero values is not allowed.

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																PERIOD															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

PWMx_PWMPR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PERIOD	Period Value. These bits determine the Period of the count cycle. The counter counts up to [Period Value] +1 and is then reset to 0x0000.

52.7.6 PWM Counter Register (PWMx_PWMCNR)

The read-only pulse-width modulator counter register (PWM_PWMCNR) contains the current count value and can be read at any time without disturbing the counter.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWMx_PWMCNR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Counter Value. These bits are the counter register value and denotes the current count state the counter register is in.

Chapter 53
Pixel Pipeline (PXP)

53.1 Overview

This document describes the micro-architecture for the Pixel Processing Pipeline used to process graphics buffers or composite video and graphics data before sending to an LCD display or TV encoder.

It is used to minimize the memory footprint required for the display pipeline and provide an area and performance optimized to both SDRAM-less and SRAM-based systems.

The PXP integrates of several independent processing stages into a cohesive strategy to create flexible pixel pipeline.

The PXP combines the following into a single processing engine:

- Scaling
- Color Space Conversion (CSC)
- Secondary Color Space Conversion (CSC2)
- Pixel Conversion Lookup Memory Table (LUT)
- Rotation

By integrating multiple blocks, intermediate buffer operations to external memory are removed, reducing external memory bandwidth, power, and software control complexity. The PXP block diagram is shown below.

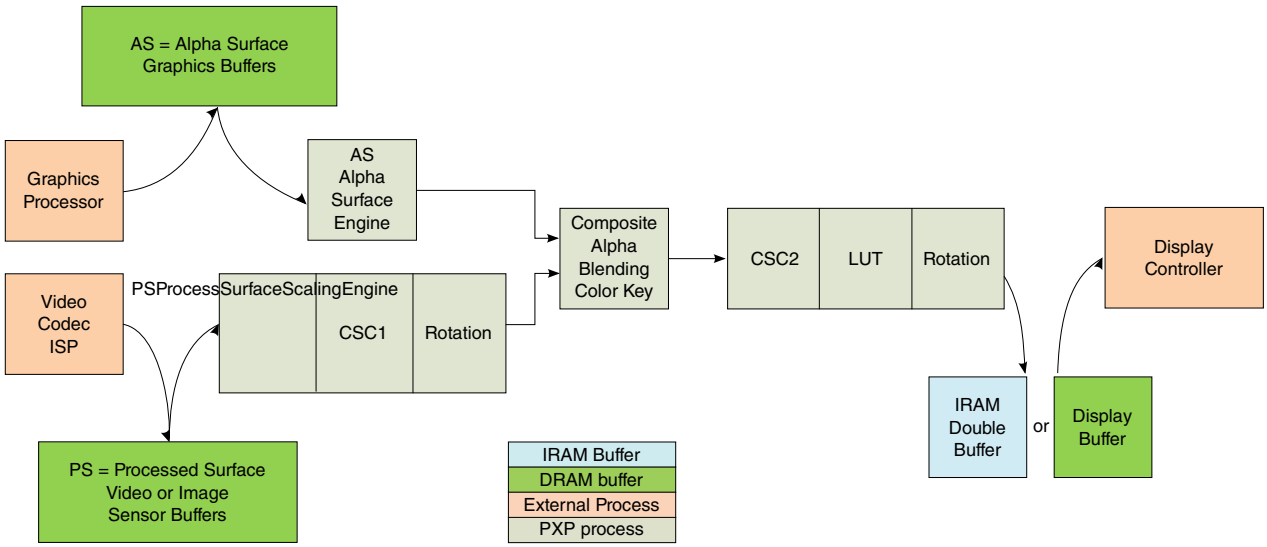


Figure 53-1. PXP Architecture

53.2 Clocks

The following table describes the clock sources for PXP. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 53-1. PXP Clocks

Clock name	Clock Root	Description
clk	epdc_axi_clk_root	PXP clock

53.3 Top-level architecture

The PXP consist of several pipelined blocks that perform the video source frame scaling, color space conversion, alpha-blending/color key algorithm, secondary CSC, pixel correction.

The entire pipeline operate within the requirements of the PXP architecture, and perform operations on either 8x8 or 16x16 pixel blocks in the representative source buffers. The entire pipeline operate within the context of two iteration counters that iterate through the appropriate grid of input blocks to produce the rotated output grid blocks in scan-line order.

Figure 53-1 shows the high-level architecture of the scaling, color space conversion, blending, pixel correction,rotation engines.The Alpha Surface Engine fetches one RGB graphics plane alpha surface (AS). The scaling engine fetches a single processed surface (PS), which can be blended with the AS surface.Although the PXP processes NxN pixel macro blocks, each of the AS or PS surfaces can have any pixel alignment within the output buffer. There are no restrictions and any pixel coordinates within the output buffer are valid. The upper left origin of the output buffer is defined as pixel 0,0. The upper left and lower right coordinates for each of the AS and PS are inclusive within the output buffer.

Figure 53-2 represents a sample output buffer configuration with both an AS and PS included. The alignment of each AS and PS within the output buffer can be at any arbitrary pixel locations. For example, the PS has an upper left coordinate (ULC) of 2,2 and a lower right coordinate (LRC) at pixel 13,13. The maximum value for the ULC and LRC for each of the AS and PS is bounded by the LRC of the output buffer, 15,15 for this example.

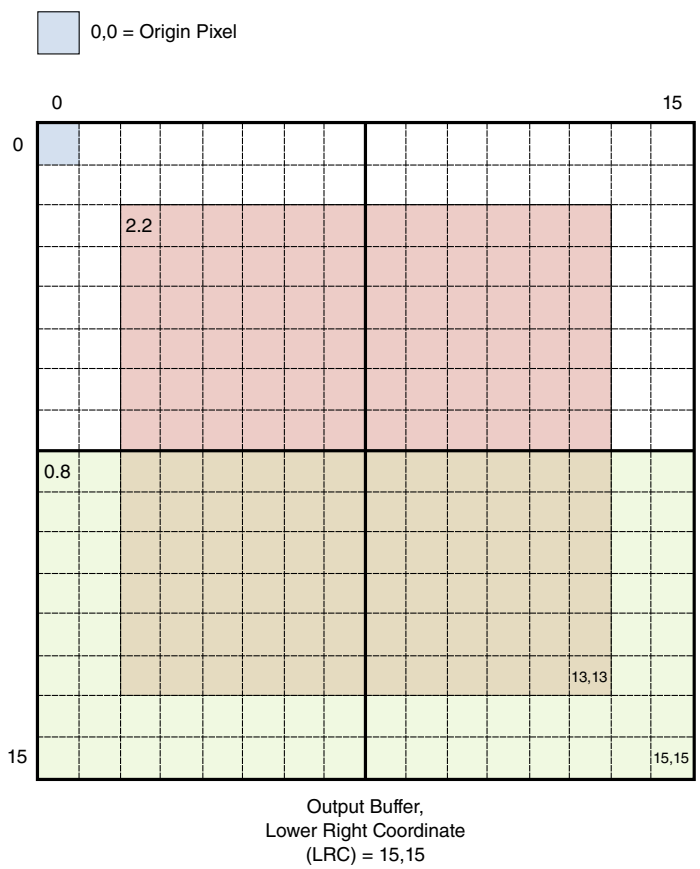


Figure 53-2. Sample output buffer configuration

The AS engine supports RGB pixel formats, and the PS engine supports RGB, YUV, and YCbCr pixel formats. The CSC1 can be used to convert to RGB pixel formats so that the PS surface can be blended with the AS surfaces in the compositing engine in the RGB color space. There is a single rotation engine in the PXP with a programmable location with two possible rotations, one after PS processing, other just before the transfer to display buffer. Rotation can occur after image composition or at the output of the PS engine. In the first scenario, all the data produced by the AS and PS engines is rotated. When the rotation module is programmed to rotate only PS images, the AS is not rotated, and AS pixels are combined with rotated PS surfaces. The CSC2 unit can convert to any RGB, YUV, or YCbCr color spaces for final output. Pixels can be corrected using a programmable LUT resource to achieve any desired pixels effects. For detailed information, please refer to Output Modes to RGBW4444CFA.

53.3.1 Processing Details

The PXP architecture has been driven primarily by the requirement that the output buffer must be processed and rotated without intermediate frame buffer stored in external memory.

This reduces the use of external memory bandwidth requirements thus reducing overall system power consumed.

Since the output of the rotation block must be NxN pixel blocks in scan order, the entire pipeline will operate on NxN pixel blocks. In essence, the pipeline will be able to operate on blocks in a random access fashion, but the entire pipeline will operate within the context of two iteration counters that will iterate through the horizontal and vertical input blocks to generate the required output block.

Processing Pipeline

The control block will coordinate the processing of the pixel blocks within the source and destination image buffers. It begins by issuing a command to each stage of the pipeline requesting that operations be done for the block at offset x, y. When the block accepts the command, it asserts its acknowledge signal for a single cycle to indicate the acceptance and allow the control unit to move to the next block.

When the PS and AS fetch engines have received a command, they will fetch the required data and place it into their fetch buffers. If compositing the RGB AS surface with the PS surface, then the output of the PS engine needs to be converted to the RGB color space using CSC1, since all compositing occurs in the RGB color space. For YUV output pixel formats, the CSC1 unit can be enabled to convert pixels into the RGB space for subsequent compositing with AS pixels. Then, the CSC2 module can convert the resulting pixels back into the YUV output color space. If the final output color space is

YUV and there is no compositing required (AS not present, for example), then both the CSC units can be bypassed and the pixel data path will pass the YUV pixels to the rotation engine. For YUV output formats, scaling operations, LUT, and rotation operations are still valid, but blending RGB AS surfaces with YUV PS surfaces is NOT supported. The two CSC units in the overall pixel data path must be used to achieve the desired source frame compositing and output pixel formatting.

The alpha blender/color key module will process a pixel any time that both inputs present valid data.

A handshake will be created between each stage and a pipeline controller to handle the advance of the pipeline and generation of the iteration counters. The pipeline controller will also maintain the interlocks with the LCD interface for the case where the LCD display and pixel processing pipeline use the SRAM to maintain the double buffer block intermediate buffer.

53.3.2 Scaling Operation

The scaling engine operates on YUV (or YCbCr) 422 or 420 and any RGB formatted pixels. Each color plane is sourced from color planes indicated by different base address registers.

The scaling source data can be stored as 3 individual planes for each Y, U, and V data, stored as two planes as a single Y and interleaved UV plane, or stored as a single plane with YUV/RGB interleaved on a per byte basis.

The scaled output image is presented to the CSC module as YUV444 or RGB888 pixels with a single byte for each color channel. The scaler can reduce an input image by a maximum factor of 16. In this case, the output image will be 1/16 the dimension of the input image in each of the X and Y axis. There are no limits, essentially, on increasing the source image size. The theoretical maximum increase is 4096 since a 12 bit fractional step function is used when scaling an input image. Scaling in either axis, X or Y is independent, so a source image can appear stretched in either direction.

All source images pass through the scale engine. The PXP alpha blend module and AS pixel streams are in the RGB888 format, so PS pixel buffers must be converted to the RGB888 format for alpha blending. The scaling engine works with the CSC1 module to translate YUV/YCbCr pixel formats to RGB888 for output frame buffer compositing using the alpha blender. The CSC2 module can by bypassed or enabled to convert pixels to any output color space. In the case of processing RGB pixels in the PS engine. The CSC1 unit can be bypassed so compositing can occur in the alpha engine.

The scaling operation is divided into two scaling steps. The first step is a decimation scaler, and the second step is a bilinear filter. The decimation filter provide a maximum down scaling factor of 8, and the subsequent bilinear filter provides a maximum scaling factor of 2. Combined, the maximum scaling factor can be up to 16. The decimation and bilinear scaling engines are independently programmable. There is also an initial offset that is programmable to allow more source data to be considered in the bilinear scaling engine.

53.3.3 Decimation Image Scaling

The first of two scaling engines is the decimation filter.

The intent of the decimation filter is to use as much source data as is possible to create the output image frame buffer. The decimation filter simply discards certain pixels from the source PS image depending on the reduction selected.

For RGB pixel formats, each color channel is treated equally since there is the same amount of pixel data within each color plane. For YUV422/420 formats, the chroma samples are already subsampled by 2. In these decimation scenarios, the chroma decimation factor is adjusted to account for the pre-decimation of the chroma samples. For example, since YUV422 is already sub-sampled by 2 horizontally, an X decimation factor of 2 does not apply to the YUV422 pixels in the X direction. All the chroma samples are passed on to the bilinear filter in this case. As another example, an X decimation factor of 4 will decimate the chroma samples by 2, since this factor combined with the pre-decimation factor of 2 in the pixel source buffers totals an overall decimation factor of 4.

The following example will show which pixels (in green) in a source RGB buffer that are passed to the bilinear filter for an X decimation factor of 2 and a Y decimation factor of 4. All pixels coincident with dashed lines are discarded.

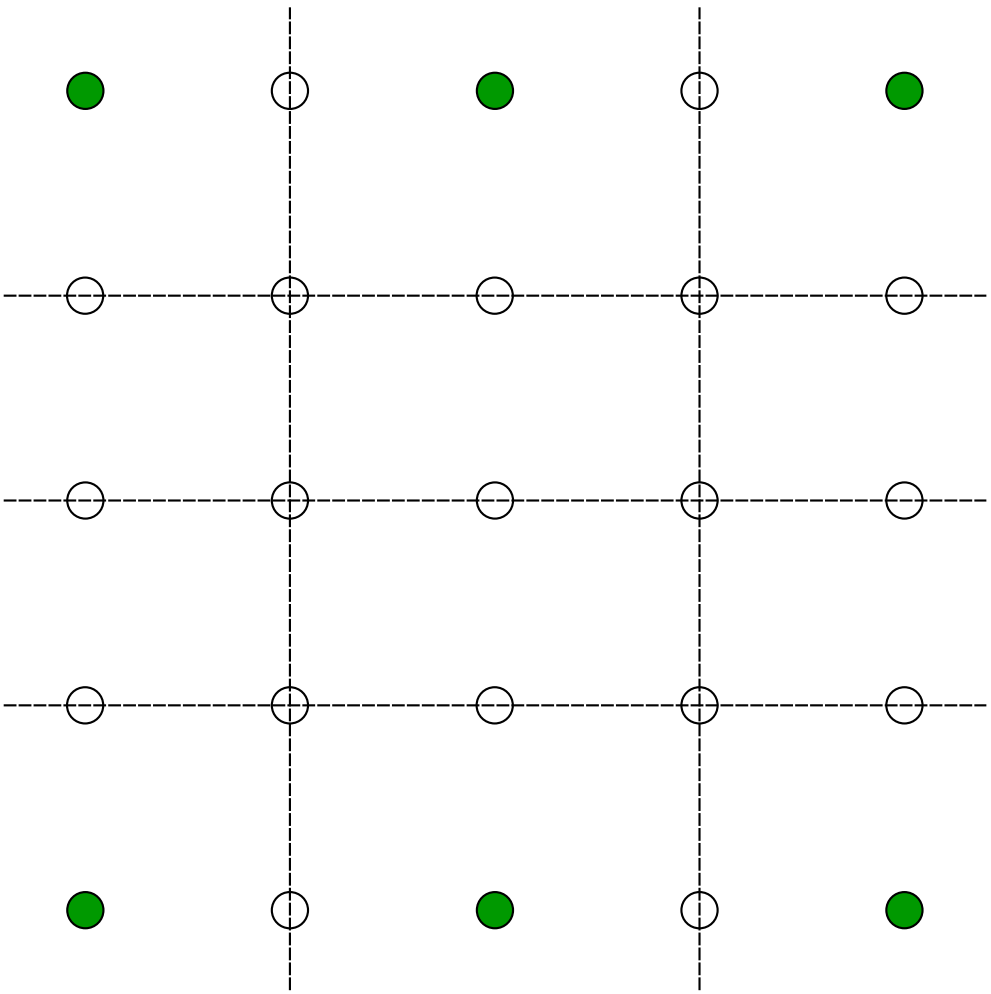


Figure 53-3. RGB decimation X /2, Y /4

Using the same decimation factor as the above scenario for RGB pixels, but using YUV420 source buffers, it can be shown that the decimation factor for the Y and UV components of data are decimated differently. This is due to the pre-decimation of the chroma samples in the source frame buffers. Figure 4: YUV420 decimation X /2, Y /4 indicates that the U/V samples in the X direction are not decimated, but the Y samples in the X direction are decimated by the factor of 2.

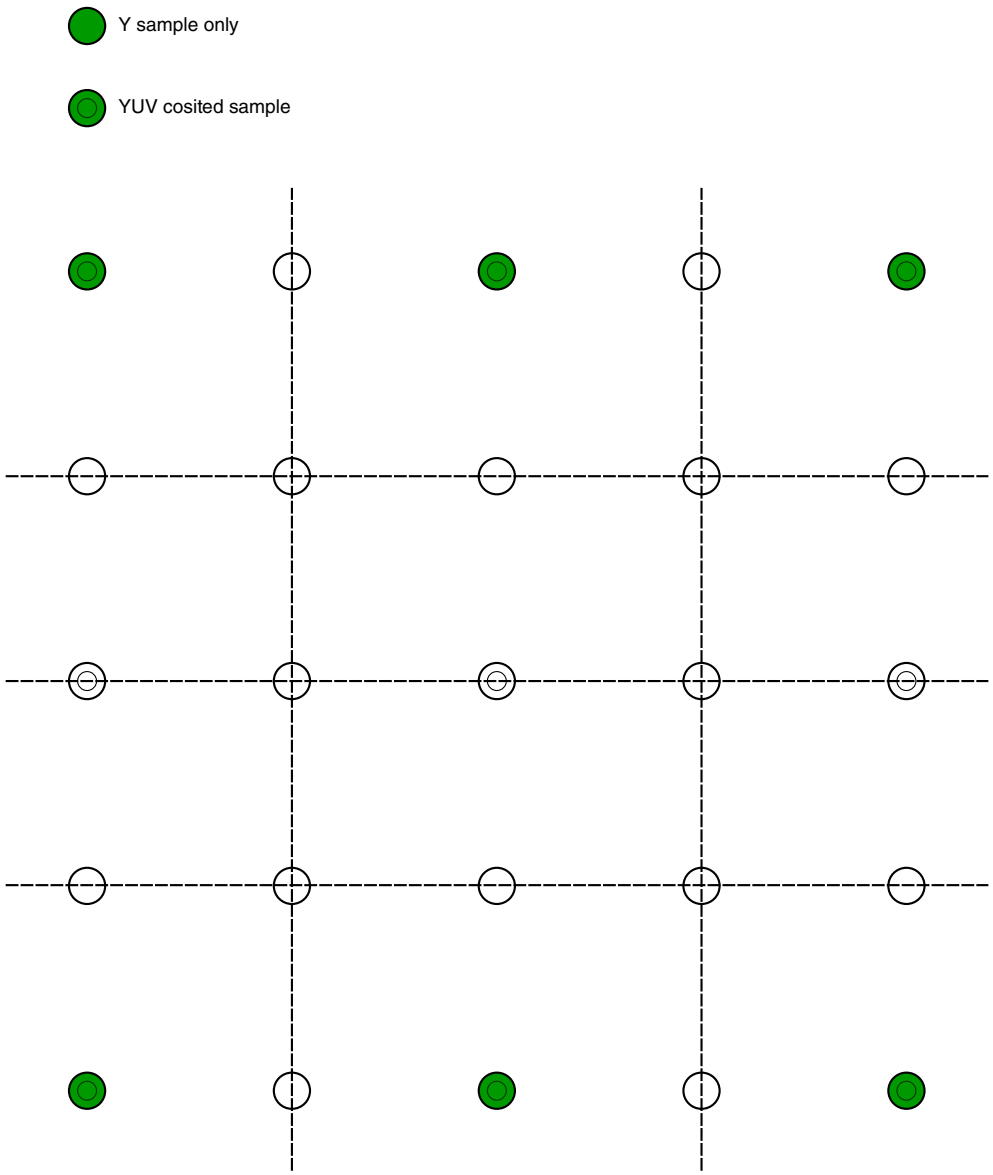


Figure 53-4. YUV420 decimation X /2, Y /4

53.3.4 Bilinear Image Scaling Filter

The PXP implements a bilinear scaling filter to resize an input image to a different resolution for display output.

The bilinear filter is a weighted average of the four nearest pixels that can be sourced to approximate the pixel in the output frame buffer.

When scaling YUV data, the UV values are offset by 0x80 (top bit inverted) to shift the signed UV bits into an unsigned equivalent with a range of 0 to 255. YCbCr data does not have to be shifted since it is defined as an unsigned byte. The REG_CSC1_COEF0[YCBCR_MODE] bit controls whether this operation is applied to the input UV bytes.

After scaling, the offset is removed so that the range for UV data is signed from -128 to 127.

The reason for this adjustment is based on the implementation of an unsigned scaling engine, and therefore, is to ensure that the scaled values are handled properly. Consider the following table:

Format	pixel0	pixel1	average	Result
decimal	-2	+2	0	Correct
CbCr	0x7E	0x82	0x80	Correct (0x80 is 0 in CbCr)
UV	0xFE	0x02	0x80	Incorrect (0x80 is -128 in UV)
decimal	-32	+16	-8	Correct
CbCr	0x60	0x90	0x78	Correct (0x78 is -8 in CbCr)
UV	0xE0	0x10	0x78	Incorrect (0x78 is +120 in UV)

To compute the output pixel value at position as indicated by P, consider the diagram below.

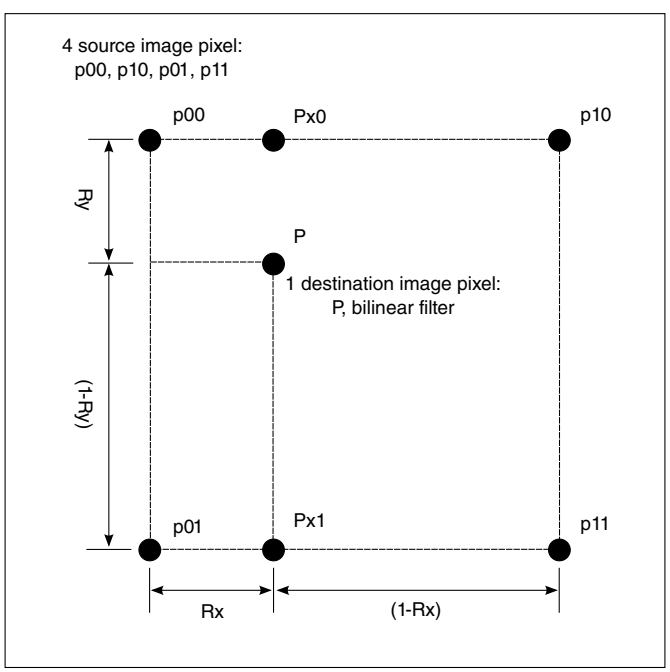


Figure 53-5. Output Pixel Value

A step function is used to indicate the position of the pixel "P" in the output frame. This position may not coincide with a single pixel position in the input frame buffer. In this case, the four closest pixels in the input frame are used to approximate the value of the pixel in the output frame.

The PXP scaler first computes a linear filter in the X axis to create the two intermediate pixel values Px0 and Px1. The step function's X fractional component is used to provide the weighting factor for blending p00 with p10 to provide Px0. Likewise, Px1 is also derived from a linear filter using p01 and p11.

The equations for Px0 and Px1 are as follows:

Px0 = p00*(1-Rx) + p10*Rx

Px1 = p01*(1-Rx) + p11*Rx

The PXP scaler uses the intermediate X pixels Px0 and Px1 and implements a bilinear filter on these two pixel values to produce the final pixel value at position P. The remainder of the step function for the Y axis is used to compute the weighted average pixel result. The equation for final filtered pixel is:

P = Px0*(1-Ry) + Px1*Ry

53.3.5 YUV 4:2:2 Image Scaling

The following figure illustrates the positioning of YUV samples for the 4:2:2 formats. There are twice as many Y luma samples then U and V chroma samples horizontally.

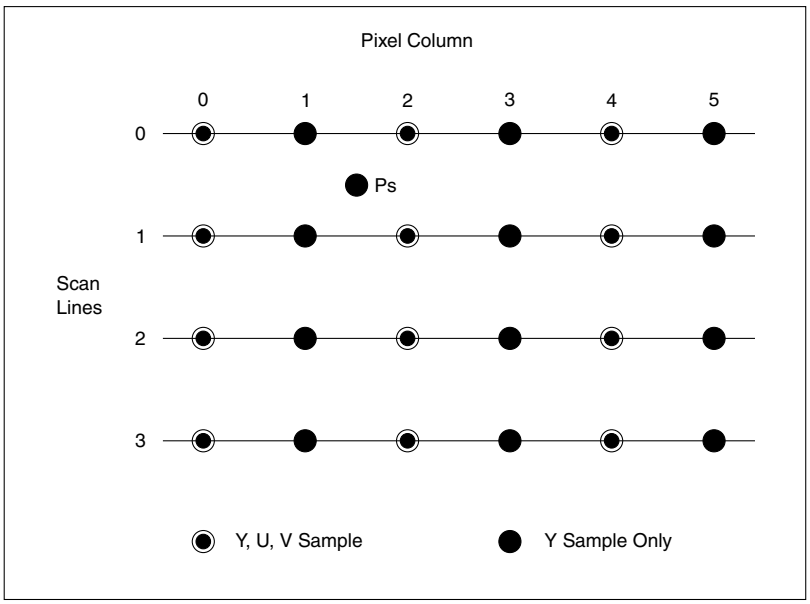


Figure 53-6. YUV Sample Positioning, 4:2:2

Consider the scaled output pixel Ps (pixel scaled) which has an accumulated step function of X=1.5 and Y=0.5. The remainder for the step function is Rx = 0.5 and Ry = 0.5. Or, the sub pixel position of output pixel Ps is half way between line 0 and 1 and half way between column 1 and 2.

The Y output component of Ps is simply the bilinear function of the four nearest Y samples from the input image. Specifically, the Y values at [1,0], [2,0], [1,1], and [2,1] are used to compute the Y for Ps.

For the U and V components of Ps, there are no samples present in the column position 1. The bilinear filter uses chroma components located at [0,0], [2,0], [0,1] and [2,1]. Since the chroma components are not sub sampled vertically, the remainder used to combine pixels vertically is Ry=0.5 (the same as for Y). However, horizontally, the scaling engine shifts the remainder by a factor of 2. So an X axis step function value of X=1.5 has a remainder Rx=0.75. Source chroma values are not replicated, they are completely interpolated using the four nearest chroma samples to approximate U and V at Ps.

53.3.6 YUV 4:2:0 Image Scaling

The following figure illustrates the positioning of YUV samples for the 4:2:0 formats. Chroma is sub sampled both horizontally and vertically. In this format, the chroma frame buffers contain ¼ the data that the luma frame buffers store.

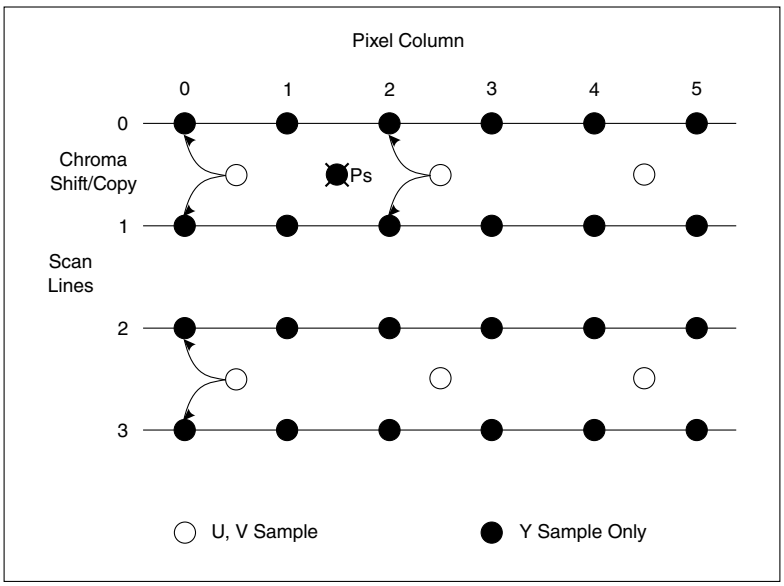


Figure 53-7. YUV Sample Positioning, 4:2:0

The Y output component for all scaled pixels in 4:2:0 formats are the same as for the 4:2:2 pixel formats.

The U and V output components have two considerations when computing the output pixel Ps.

1. All chroma samples from the input source image are shifted left and up by ½ a sample position of the input pixel matrix.
2. Odd scan lines are replicated using the previous even chroma scan line values. So, output image chroma values that map between even to odd scan lines are replicated in the vertical axis. In contrast, output image chroma values between odd to even scan lines are interpolated vertically.

The chroma values are interpolated horizontally as in the 4:2:2 pixel format.

As an example, consider the interpolated pixel Ps in the 4:2:0 diagram above. For the Y component, the interpolated output luma is a function of the Y values in the source frame buffer at position [1,0], [2,0], [1,1], [2,1].

For the U and V interpolated samples, the chroma values on scan line position 0.5 are shifted so that they coincide with the even luma sample points. They are also replicated so that a single chroma scan line is used twice. The chroma scan line at 0.5 is replicated to represent the 4:2:2 sample points for scan line 0 and 1. The chroma scan line at 2.5 is replicated to represent the 4:2:2 sample points for scan line 2 and 3. This pattern of chroma replication occurs for the entire source frame buffer during the scaling operation.

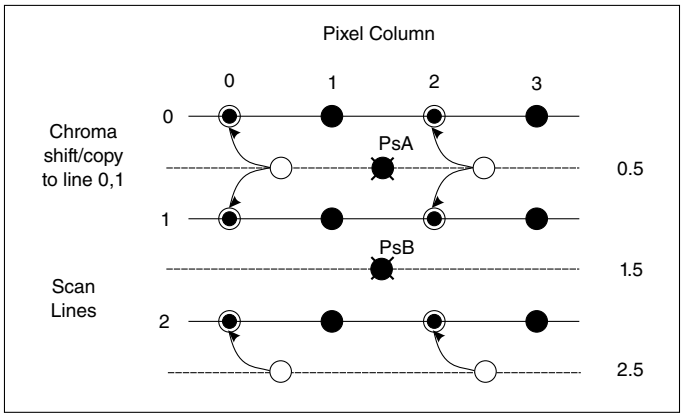


Figure 53-8. Scaled Chroma Computation Examples

The preceding diagram has two examples for the computation of the scaled chroma output pixel. For chroma at output position PsA (vertical position 0.5), interpolation occurs in the X axis using chroma values at column 0 and column 2. However, since line 0 and line 1 have equal chroma values due to chroma line replication, scaling in the Y axis results in replication of chroma values.

For chroma at output position PsB (vertical position 1.5), interpolation occurs in both the X and Y axis. The Y axis is an interpolation since the chroma values copied to scan line 1 and 2 and not the same.

In summary, any output image pixels that map to an odd scan line above and an even scan line below are interpolated vertically. Output image pixels that map to an even scan line above and an odd scan line below are replicated vertically.

53.3.7 RGB/YUV444 Image Scaling

For all RGB formats, the RGB pixels are converted up to RGB888 with 8 bits per each color component.

Then each color component is passed to the scaling engine and each component is treated in the same manor. The RGB scaling operation is the same as for the Y scaling operation described in the preceding sections. Also, YUV444 contains a byte for each color plane at each pixel location, so all three color components are scaled in the same manor.

53.3.8 Color Space Conversion (CSC)

The CSC unit converts pixels between color spaces. They are referred to as CSC1 and CSC2. The CSC1 unit is right after the scaling unit and is dedicated to converting from YUV to RGB. The CSC2 unit is a full-duplex color space converter that can convert into either RGB or YUV (or YCbCr) color spaces depending on the desired output pixel format. All coefficients are programmed as two's compliment numbers and CSC units can be bypassed if conversion is not desired at either position of these CSC units in the pixel data path.

53.3.9 CSC1 Operation

The CSC1 module receives scaled YUV/YCbCr444 pixels from the scale engine and converts the pixels to the RGB888 color space only if CSC1 is enabled.

The CSC1 module will convert only to the RGB color space and it can be bypassed to allow YUV pixels through the data path. These pixels are loaded into the pixel FIFO for processing by subsequent modules in the pixel data path.

The following equations are used to perform YUV/YCbCr -> RGB conversion. The constants will be stored in the PXP control registers as two's compliment values to allow flexibility in the implementation and to allow for differences in the video encode and decode operations. In addition, this provides a software mechanism to manipulate brightness or contrast.

R = C0(Y+Yoffset) + C1(V+UVoffset)

G = C0(Y+Yoffset) + C3(U+UVoffset) + C2(V+UVoffset)

B = C0(Y+Yoffset) + C4(U+UVoffset)

Note: In the equations above, U and V are synonymous with Cb and Cr in regards to the color space format of the source frame buffer.

Saturation of each color channel is checked and corrected for excursions outside the nominal YUV/YCbCr color spaces. Overflow for the three channels are saturated at 0x255 and underflow is saturated at 0x00.

The table below indicates the expected coefficients for YUV and YCbCr modes of operation:

Coefficient	YUV	YCbCr
Yoffset	0x000	0x1F0 (-16)
UVoffset	0x000	0x180 (-128)

Table continues on the next page...

C0	0x100 (1.00)	0x12A (1.164)
C1	0x123 (1.140)	0x198 (1.596)
C2	0x76B (-0.581)	0x730 (-0.813)
C3	0x79B (-0.394)	0x79C (-0.392)
C4	0x208 (2.032)	0x204 (2.017)

53.3.10 YUV versus YCbCr Support

By default, the PXP color space coefficients are set to support the conversion of YUV data to RGB data.

If YCbCr input is present, software must change the coefficient registers appropriately (see the register definitions for values). Software must also set the YCBCR_MODE bit in the COEFF0 register to ensure proper conversion of YUV versus YCBCR data.

53.3.11 CSC2 operation

The CSC2 module receives pixels in any color space and can convert the pixels into any of RGB, YUV, or YCbCr color spaces.

All coefficients are programmable and in the two's compliment notation. The output pixels are passed onto the LUT and rotation engine for further processing.

The following equations indicate the CSC2 modules ALU architecture.

Selecting RGB output in REG_CSC2_CTRL[CSC_MODE] configures the ALU in the following manor:

R = A1(Y-D1) + A2(U-D2) + A3(V-D3)

G = B1(Y-D1) + B2(U-D2) + B3(V-D3)

B = C1(Y-D1) + C2(U-D2) + C3(V-D3)

Selecting YUV output configures the ALU in the alternate manor:

Y = A1*R + A2*G + A3*B + D1

U = B1*R + B2*G + B3*B + D2

V = C1*R + C2*G + C3*B + D3

Saturation of each color channel is checked and corrected for excursions outside the nominal color space. Overflow for the three channels are saturated at 0x255 and underflow is saturated at 0x00.

53.3.12 Alpha Blending/Color Key

Regardless of pixel input format, the PS and AS pixels are normalized to 32-bits, organized as one alpha and three data bytes. Alpha blending occurs in the RGB space, if blending is required, PS pixels should be converted to RGB space. If no alpha blending is required, then YUV pixels can bypass the alpha blending ALU without color space conversion. All pixels are processed by the pixel ALU, but the ALU operations can be disabled to achieve pixel pass through for either PS or AS source pixels.

53.3.13 Alpha Blend

The alpha value for an individual pixel represents a mathematical weighting factor applied to the AS pixel. An alpha value of 0x00 corresponds to a transparent pixel and a value of 0xFF corresponds to an opaque pixel.

The effective alpha value for an AS pixel is determined by the AS_CTRL[ALPHA] and AS_CTRL[ALPHA_CTRL] register fields. If AS_CTRL[ALPHA_CTRL] = ALPHA_OVERRIDE , the alpha value for the pixel is taken from the AS_CTRL[ALPHA] . This can be useful for applying a constant alpha to an entire image or for image formats that don't include an alpha value. If AS_CTRL[ALPHA_CTRL] = ALPHA_MULTIPLY, the pixel's alpha value will be multiplied by the pixel's ALPHA value in order to allow scaling of the pixel's alpha or to provide better control for pixel formats such as RGB1555, which only contains a single bit of alpha.

For each color channel, the equation used to blend two source pixels is defined below:

Gá = PIO programmed global alpha (8-bit value).
Eá = Embedded alpha associated with AS pixel.
á = Gá * Eá + 0x80

The result for the red channel as an example:

R[7:0] = (á * PS.r) + ((1 - á) * AS.r)

When á is 0xff, the PS pixel will not be blended with the AS pixel, but PS will be passed as the output pixel and will not be blended with AS. In this case, AS will be discarded. Likewise, if á is 0x00 for a given pixel, PS will be loaded as the output pixel.

AS_CTRL[ALPHA_INVERT] provides the option to invert the final alpha value. This essentially inverts the effect the alpha value has on the AS and PS blending operation.

53.3.14 Color Key

The color key function is provided to create transparent effects on the output pixel.

Color keying is applied on the input pixels after they are converted to 8-bits for each red, green, and blue color channels (color keys are not applied directly to 16-bit pixel formats but to their corresponding 24-bit representation). A color key range is programmable for both PS and AS pixels. If the PS 24-bit pixel is within the PS color key range, then AS is passed through the pixel pipeline. In this case, alpha blending does NOT occur. Conversely, if PS is within the AS color key range, then PS is passed via the PXP data pipeline. If both PS and AS color key tests pass, then the back ground color register is passed onto following PXP processing components in the pipeline.

The condition for color keying to be satisfied is:

CK0.r.low <= PS.r <= CK0.r.high

CK0.g.low <= PS.g <= CK0.g.high

CK0.b.low <= PS.b <= CK0.b.high

For example, if the "red" 8-bit value for the PS pixel (or PS.r) is between the color key low and high values (CK0.r.l and CK0.r.h), the condition is true for the red color plane. When ALL three color planes meet this condition, then only the PS pixel is loaded into the output register.

To disable color keying, program the low color key register value to 0xff and the high value to 0x00. This will guarantee that the color key range test will never be true.

53.3.15 LUT

The lookup table (LUT) is used to modify pixels in a manner that is not linear and that cannot be achieved by the color space conversion modules.

Nonlinear response to the input pixels can be achieved based on how the lookup table is programmed.

Programming of the direct access LUT table can be facilitated by single PIO register writes or DMA access. For efficient loading of the LUT, DMA access should be used.

53.3.16 Lookup Modes

The LUT has four lookup modes. The lookup modes determine how the src_pixel is used to address the LUT memory.

The four lookup modes are:

1. DIRECT_Y8
2. DIRECT_RGB444
3. DIRECT_RGB454
4. CACHE_RGB565

The DIRECT modes access the LUT memory as a monolithic SRAM. The CACHE_RGB565 will access the memory as a 2-way set associative cache.

53.3.17 DIRECT_Y8

DIRECT_Y8 is used for a 256-byte lookup. In DIRECT_Y8, the most significant byte of the pixel is used to address the LUT entry.

This byte reflects the Y/R channel of the pixel data path. Luma, or monochrome, transformations are possible with this lookup mode. The address is generated as: src_pixel[23:16]. In DIRECT_Y8 operation, the memory is byte addressable.

53.3.18 DIRECT_RGB444

DIRECT_RGB444 is used for a 8KB (4K pixel) RGB444 to RGB565 lookup.

Pixel formats that are in the YUV color space at the position of the LUT in the PXP data path can also be converted. To take advantage of the full 16KBmemory, the REG_LUT_CTRL[SEL_8KB] bit can be used to select the upper or lower 8KB memory, thus facilitating the use of 2 separate 444 LUT tables. In DIRECT_RGB444, the src_pixel is RGB/YUV[23;0] data is used to generate the lookup address. The address is generated as: {R/Y[23:20],G/U[15:12],B/V[7:4]}. In DIRECT_RGB444, the memory is pixel (2 byte) addressable.

53.3.19 DIRECT_RGB454

DIRECT_RGB454 is used for a 16KB (8K pixel) RGB454 to RGB565 lookup.

Pixel formats that are in the YUV color space at the position of the LUT in the PXP data path can also be converted. In DIRECT_RGB454, the src_pixel is RGB/YUV[23;0] data is used to generate the lookup address. The address is generated as: {R/Y[23:20],G/U[15:11],B/V[7:4]}. In DIRECT_RGB454, the memory is pixel (2 byte) addressable.

53.3.20 CACHE_RGB565

The CACHE_RGB565 lookup is used for a 128KB (65K pixel) RGB/YUV565 to RGB/YUV565 lookup.

The 128KB memory requirement is too costly from an area perspective to implement a complete lookup table in the LUT's on chip memory. For this reason, a LRU (least recently used) 16KB 2-way set associative cache has been implemented to reference the full 128KB lookup table stored in external memory.

The 2-way set associative cache is organized in the following way:

- 16KB total data storage
- 512 entries split between 256 ways
- 6 pixels/entry cache line

Cache efficiency is very critical. For a cache miss, the PXP will be stalled until the cache line can be filled. For a 32 bit DDR memory interface, the latency can be calculated as follows:

cycles latency for read command through DDR controller + CAS Latency + # burst cycles for read data to be returned + # cycles latency for data through DDR controller to LUT + 1 cycle for cache access of new data

The src_pixel is RGB[23:0] data used to generate the lookup address. To improve the cache efficiency the RGB/YUV565 address and the lookup table must be formatted in the following way:

Address: A[15:0] = $R_7G_7G_6B_7$ $R_6G_5B_6$ $R_5G_4B_5$ $R_4G_3B_4$ $R_3G_2B_3$

The address is organized as follows:

- A[15:12] tag address, used to compare a hit between cache ways
- A[11:4] index address, used to select cache set (i.e. row of memory)
- A[3:0] block address, used to select Pixel in the cache line

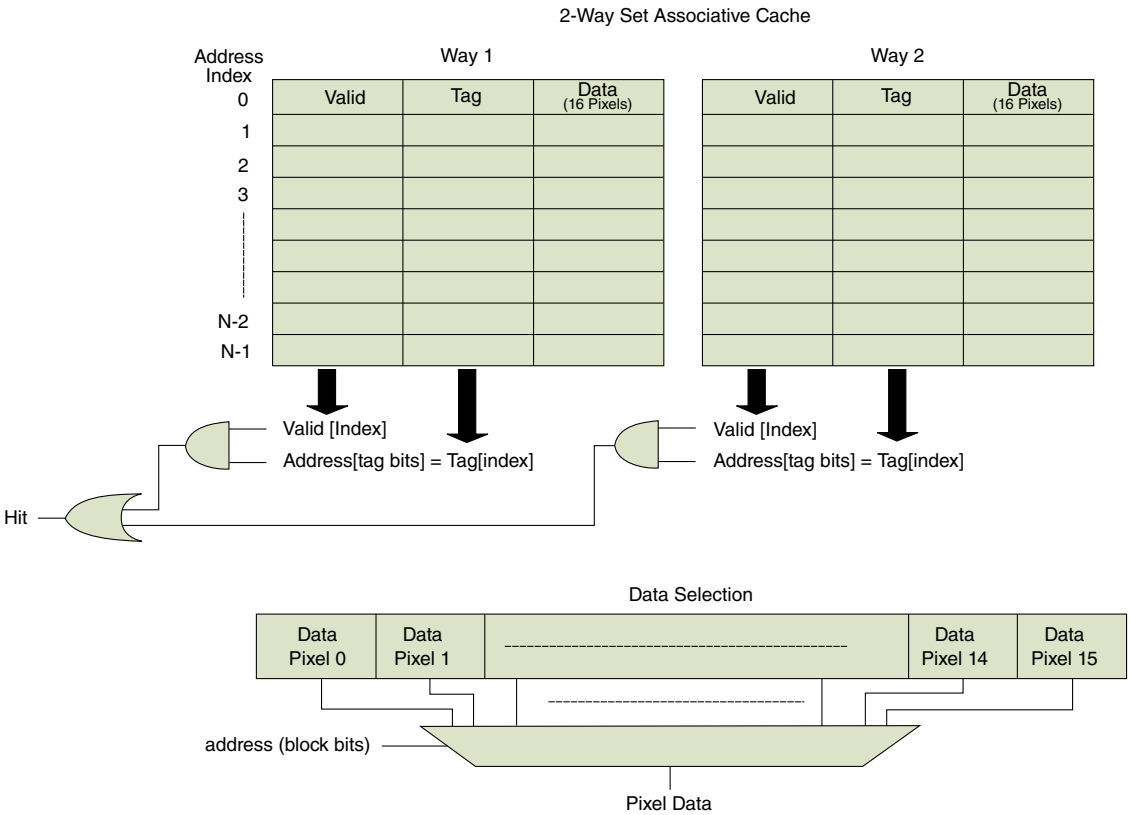


Figure 53-9. 2-Way Set Associative Cache and Data Selection

53.3.21 Output Modes

The LUT has three output modes for color space conversion.

1. Y8
2. RGBW4444CFA
3. RGB888

53.3.22 Y8

With out_mode set to Y8, in conjunction with DIRECT_Y8 lookup mode, the intended operation is Gama Correction.

Only the third byte is processed by the lookup table. The third byte is represented by the Y value or the R value in the data path since pixel data is either YUV[23:0] or RGB[23:0] where the Y or R byte encompass bits [23:16] respectfully. So, bits 15:0 are

always bypassed and left unchanged. Currently, the LUT is intended to process Y data when any YUV, Y8, or Y4 output pixel formats are selected. However, this resource can be enabled and used for any conceivable purpose.

Note: When the DIRECT_RGB444, DIRECT_RGB454 or CACHE_RGB565 lookup mode is selected in conjunction with the Y8 output mode the low order byte of the two bytes read from the LUT memory will be used as the Gama Correction value.

53.3.23 RGBW4444CFA

With REG_LUT_CTRL[OUT_MODE] set to RGBW4444CFA, the REG_CFA[DATA] is used to select one nibble from the LUT 16 bit output value.

The LUT memory lookup will contain a RGBW4444 value. The REG_CFA[DATA] will select the R,G,B or W nibble as the pixel value to present to the PXP data path based on the matrix defined by the REG_CFA[DATA] register. The 4 bit value is presented in the Y, or third byte lane, of the PXP data path. The final pixel transferred to the next PXP stage is {CFA[3:0],CFA[3:0],LUT[15:0]}

53.3.23.1 CFA Correction

The 32-bit REG__CFA[DATA] register is used to encode 16 CFA correction values.

The CFA correction values are encoded as follows:

- 00 selects R
- 01 selects G
- 10 selects B
- 11 selects W

The CFA correction uses 4x4 block processing. Figure 5; CFA mapping translation shows how CFA 4x4 blocks will iterate over the PXP's 8x8 or 16x16 pixel block being processed:

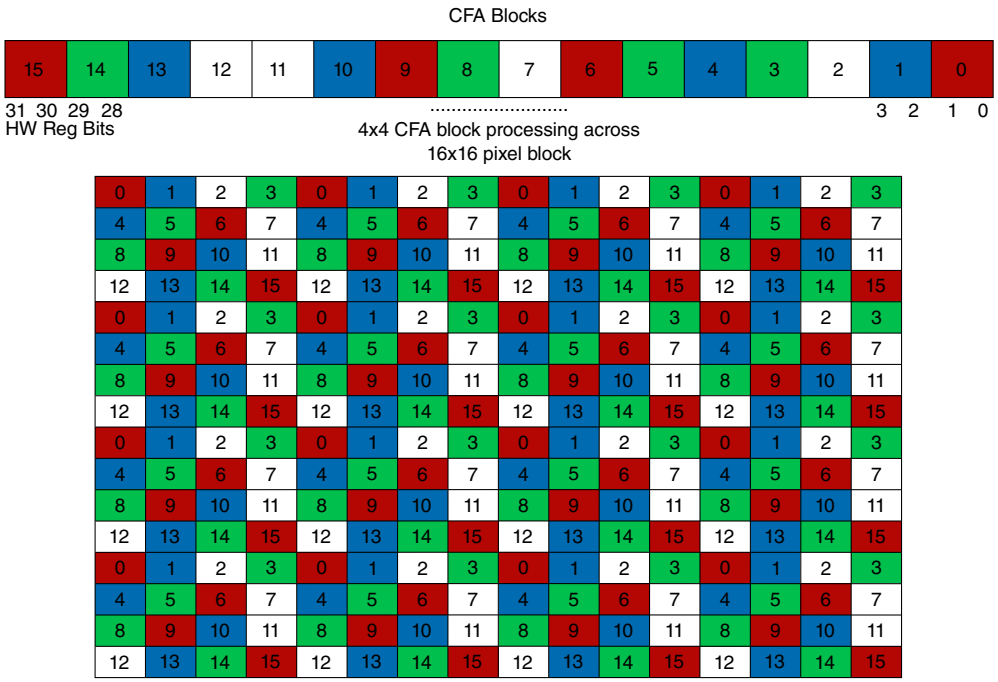


Figure 53-10. CFA mapping translation

53.3.24 RGB888

With out_mode set to RGB888, the memory output data is interpolated from RGB565 to RGB888.

The RGB[23:0] data is formatted as follows: R[7:3]R[7:5],G[7:2]G[7:6],B[7:3]B[7:5].

53.3.25 Rotation

There is a single rotation resource integrated into the PXP. The location of this resource within the PXP data path is programmable. Rotation can occur after compositing the AS and PS buffers in the output stage.

As an alternative configuration, the PS buffer can be rotated and later composited with the AS surface that is not rotated. There is a single configuration bit that provides the configuration of where rotation is implemented within the PXP.

To rotate graphics, the hardware must read pixels in one direction across a frame buffer and write them in a alternate orientation. For the 90 and 270 degree cases, this means that lines of pixels must either be read or written vertically in a frame buffer.

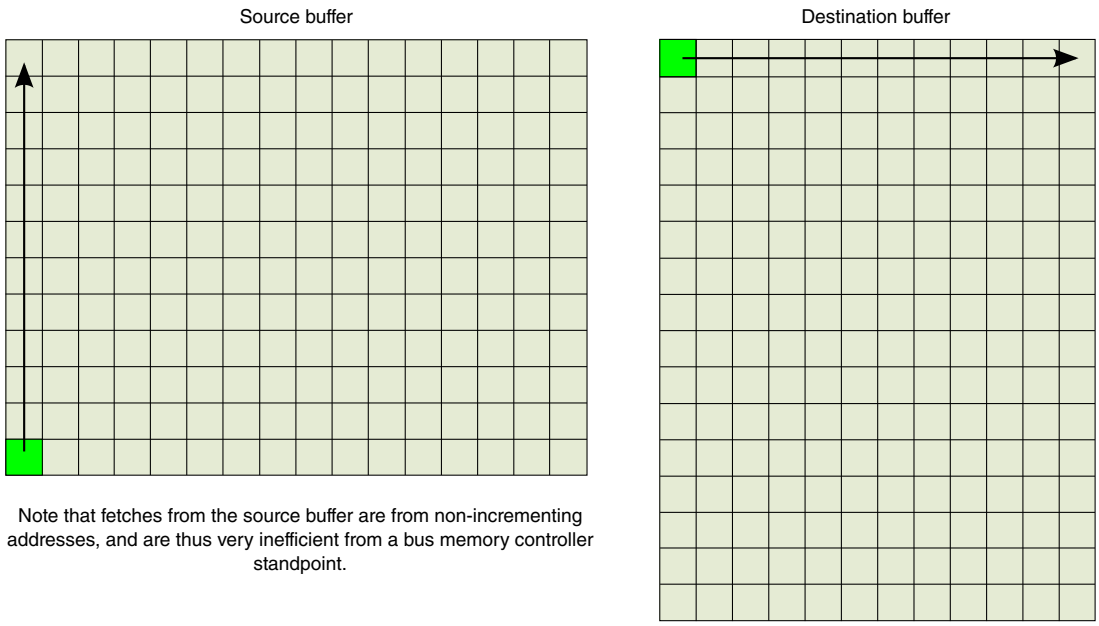


Figure 53-11. Rotation Read and Write

In order to rotate efficiently, multiple columns must be rotated to enable the engine to both fetch and store bursts of pixels, thus improving memory performance. The simplest method of doing this is to operate on square blocks of pixels. To rotate the image, each sub-block of pixels must be rotated by the required rotation angle.

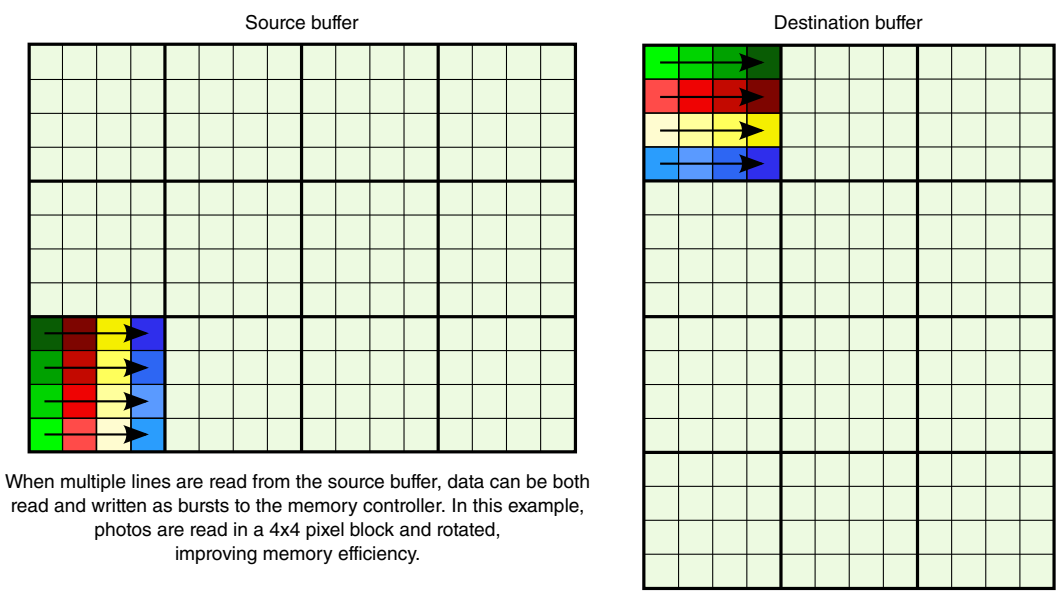


Figure 53-12. Rotated Sub-blocks

To manage the rotation process, the source image can be broken into a grid of sub-blocks that have coordinates as shown in the diagram below. In addition to rotating the sub-block, each block must be translated to a new coordinate location. For each of the rotation angles (0, 90, 180, 270), it is possible to define a simple algorithm for computing the new translated grid address. The hardware must then simply compute the memory address from the base grid address for both load and store operations.

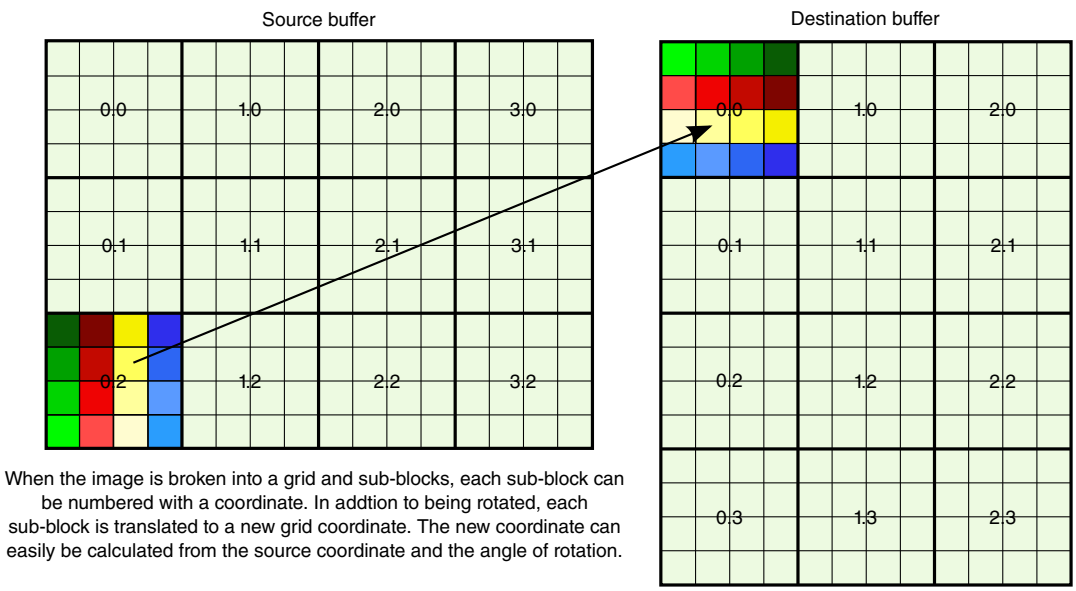


Figure 53-13. Grid of Sub-Blocks with Coordinates

In order to balance the requirements of reasonable burst sizes to the memory controller as well as keep the hardware storage requirements to a minimum, the blending/rotation engine will operate on either 8x8 or 16x16 pixel blocks. When using the Rotate engine with the input fetch engine, you need to program the input fetch engine to work in 8x8 block mode.

NOTE

An important artifact of the PXP is when rotating a source image and the output is NOT divisible by the block size selected. The output engine essentially truncates any output pixels after the desired number of pixels has been written. Since the output buffer is written as a horizontal row of blocks, the incorrect pixels could be truncated and the final output image can look shifted. In the case where the block size is programmed to 8x8, and the output size that is programmed is 12x12, then there is a remainder of 4 pixels that will be truncated in either the X and/or Y axis when the PXP operation

is complete. The output will be shifted by 4 pixels in this example. To compensate for this, the source base address needs to be adjusted so the correct pixels get truncated and the image does not look shifted. In this example, with 90 degrees of rotation, the PS base address should be adjusted by 4 times the actual PS base address -(4*pitch).

53.3.26 Output Buffer

The output buffer engine accepts data from the PXP pixel pipeline and issues requests to transfer the output pixels to external DRAM or the internal SRAM double buffer row of blocks.

53.3.27 Address calculator

Each of the blocks will manage its own fetch address using a common address calculator block that computes real addresses from a base address and relative block offset from the base.

Each block will then perform the multiple line fetches (or stores) required to perform the operation. This hides all the address buffer computations from the processing blocks and allows each block to simply track the coordinate of the block it is working on.

53.3.28 Block size selection

The PXP can be configured to process blocks that are either 8x8 pixels or 16x16 pixels with the REG_CTRL[BLOCK_SIZE] control bit.

When selecting a 16x16 pixel block size, the accesses to fetch AS and PS images and write the final frame buffer are more efficient since twice as much data is requested and processed per memory request.

When optimizing the system for memory bandwidth and image processing time, configure the PXP to process 16x16 pixel blocks.

53.3.29 Interlaced Video Support

The PXP has some minimal ability to generate interlaced video content from a progressive source. There two available options, based on the bandwidth requirements and how software is managing video frames.

The PXP can either interlace on the input side (by reading every other line of input data) or on the output side (by writing the individual lines of video into two separate fields). Generally, output interleaving should be used since it is the most flexible mode (it allows scaling and full overlay support) and it only requires a single pass of the PXP to generate two separate output fields.

Input interleaving can be beneficial in cases where the PXP is running at 60fps, since it requires fewer fetches to produce the output data. There is no direct hardware support for input interleaving, in that, there is no configuration bit that can be set to alter how the PXP processes a frame for input interleaving. Input interleaving is achieved by simply setting the source frame buffer pitch value to twice the value it would normally be set to for the equivalent progressive frame. The output parameters also need to be consistent with the desired processing effect. For example, the vertical resolution would be set to account for the reduced resolution to process the interlaced input buffers.

53.3.30 Theory of Operation

The PXP can be used to accelerate graphics operations by offloading graphics processing from the processor. The block can perform alpha blending and color key substitution on two RGB graphics buffers.

The PXP is organized as having a processed surface (PS) and an alpha surface (AS) that can be blended with the processed surface. There are no restrictions on the location of the AS or PS within the output surface (OS). As the PXP processes NxN blocks, operations are performed on a pixel by pixel basis. The AS and PS pixels are alpha blended, color keyed, process by CSC and LUT resources as individual pixel components. This allows efficient block processing with supporting arbitrary alignment for both the AS and PS surfaces. The resulting pixel block is then written to the corresponding block in the output buffer.

53.3.31 Pixel Handling

All pixels are internally represented as 32-bit values regardless of input or output pixel formats. The pixels get converted in the AS and PS buffer engines to 24-bit pixel values. There is also an 8-bit alpha value at stages up to the alpha blender within the PXP for

blending within the RGB color space. Compositing of AS and PS images can only occur in the RGB color space. If compositing is not required, then YUV pixels can be transferred and processed at all PXP pixel resource components.. The color orientation of pixels within the PXP can be controlled by the CSC1, CSC2 , and LUT resources.

For RGB, input pixels are converted into 32-bit pixel values using the following rules for both AS and PS:

1. 32-bit ARGB8888 pixels are read directly with no conversion.
2. 32-bit RGB888 pixels are assumed to have an alpha value of 0xFF (full opaque).
3. 6-bit RGB565 and RGB555 values are expanded into the corresponding 24-bit color space and assigned an alpha value of 0xFF (opaque). The expansion process replicates the upper pixel bits into the lower pixel bits (for instance a 16-bit RGB555 triplet of 0x1F/0x10/0x07 would be expanded to 0xFF/0x84/0x39).
4. 16-bit RGB1555 values are expanded into the corresponding 24-bit color space and assigned an alpha value of either 0x00 or 0xFF, based on the 1-bit alpha value in the pixel. The ALPHA_MULTIPLY function is useful in this scenario to allow scaling of the opaque pixels to a semi-transparent value.

Alpha values can be passed through the entire PXP data path and output in ARGB888 and ARGB555 pixel modes. Also, output pixels can be assigned an alpha value using the REG_OUT_CTRL[ALPHA] register. 16-bit pixels values are formed from the most significant bits of the 24-bit pixel values.

When YUV/YCbCr output formats are selected, all pixels are internally represented as either RGB or YUV pixels values. The CSC2 or LUT can convert internal RGB/YUV pixels into the correct output format. In this way, any PS color space can be blended with AS RGB pixels and output in any color space.

53.3.32 Output Buffer Composition

The output buffer will be rendered by composing each pixel block from the associated PS and AS buffers.

The AS pixel buffer can be blended or color-keyed with the associated data from the PS buffer (either the PS image pixels or REG_PS_BACKGROUND register based on PS programmed coordinates).

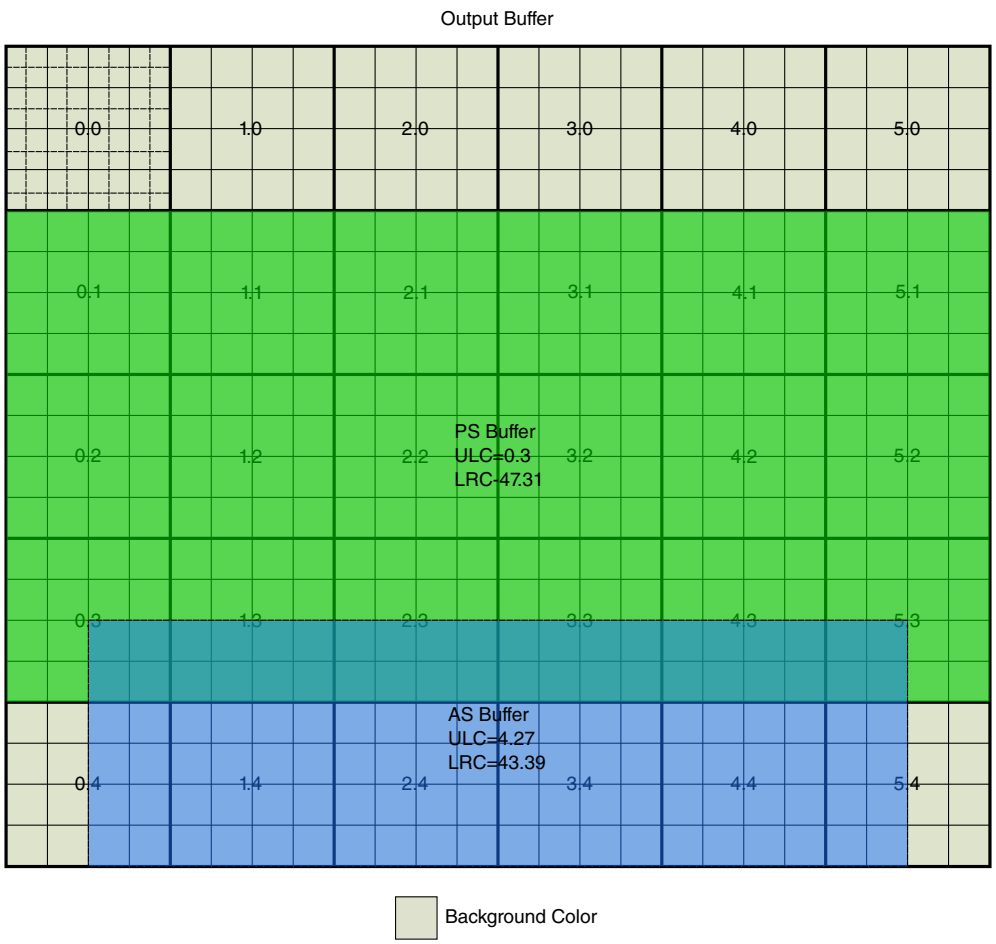


Figure 53-14. Output Buffer Composition

53.3.33 PS Image Processing

As the PXP processes image buffers, it iterates over the output buffer by fetching the corresponding input buffer blocks and processing the pixels embedded in these.

53.3.34 Letterboxing

At each pixel coordinate, the control logic determines if the PS pixel (argument also applies to AS pixels) will be used in rendering the output pixel.

This is determined by checking the output pixel's coordinates against the REG_OUT_PS_ULC and REG_OUT_PS_LRC (ULC and LRC in short) register contents. For pixels outside this region, the PS pixel will be loaded with the pixel value from REG_PS_BACKGROUND, which can be used to effectively control the letterboxing color. There are no block size or block boundary restrictions when setting the ULC or LRC for either the AS or PS. The only restriction is that the ULC and LRC are within the OUT LRC extents.

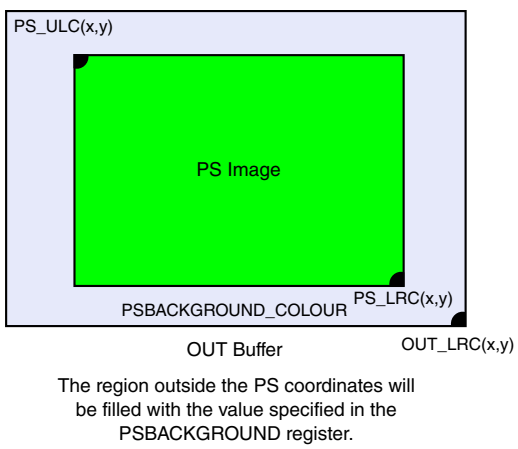


Figure 53-15. OUT Buffer

53.3.35 Clipping source images

A subset of the PS buffer can be used in rendering the output buffer. The PXP_PS_BUF register can indicate an offset into the PS buffer that will be used for display within the OUTPUT buffer.

The pixel at the address defined in the PXP_PS_BUF register will be the pixel that is displayed at the pixel coordinate indicated by PXP_OUT_PS_ULC within the output buffer. Essentially, the PXP_PS_BUF register can be used to establish an offset into the PS buffer thus clipping all PS buffer pixels that are at a lower address. The PXP_PS_PITCH will always indicate the number of bytes that are vertically adjacent in

the PS buffer. The settings in the PXP_PS_BUF, PXP_OUT_PS_ULC, and PXP_OUT_PS_LRC will determine the subset of the PS buffer, or clipped PS source buffer, that will be used in the output buffer.

It is important to note that when scaling the PS buffer, the coordinates of the PS buffer within the output buffer need to be consistent with the scaling factors and original PS buffer size.

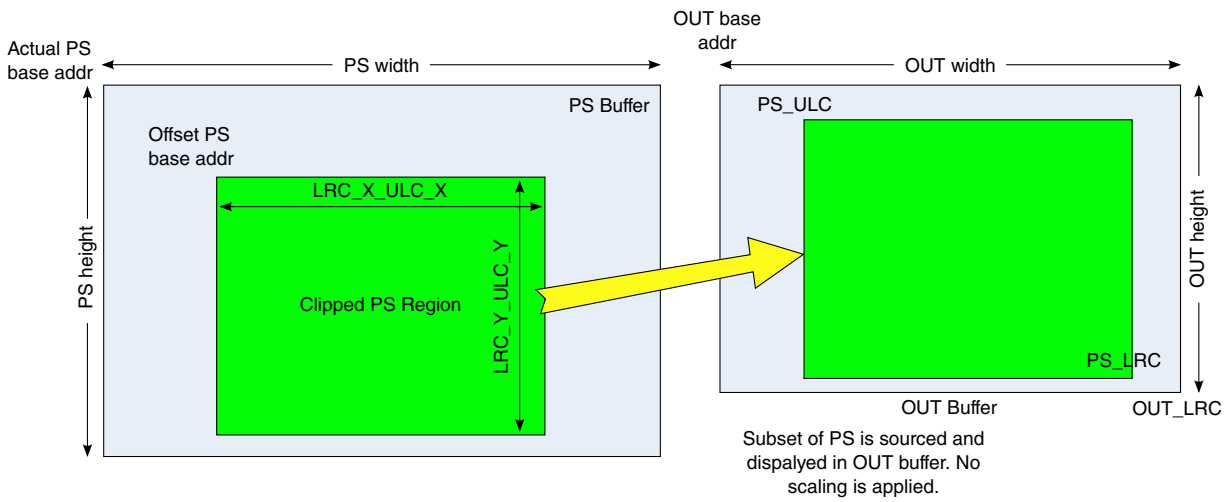


Figure 53-16. PS Buffer Scaling

When sourcing a subset of the PS image, it should fall completely within the PS buffer to avoid displaying incorrect data. The following conditions should be met:

x_base_addr_offset + x_scale * (LRC_X-ULC_X) <= PS_pitch

y_base_addr_offset + y_scale * (LRC_Y-ULC_Y) <= PS_size

The PXP hardware does not check for these conditions and will render the image as programmed. The following case could indicate invalid programming parameters for the PXP:

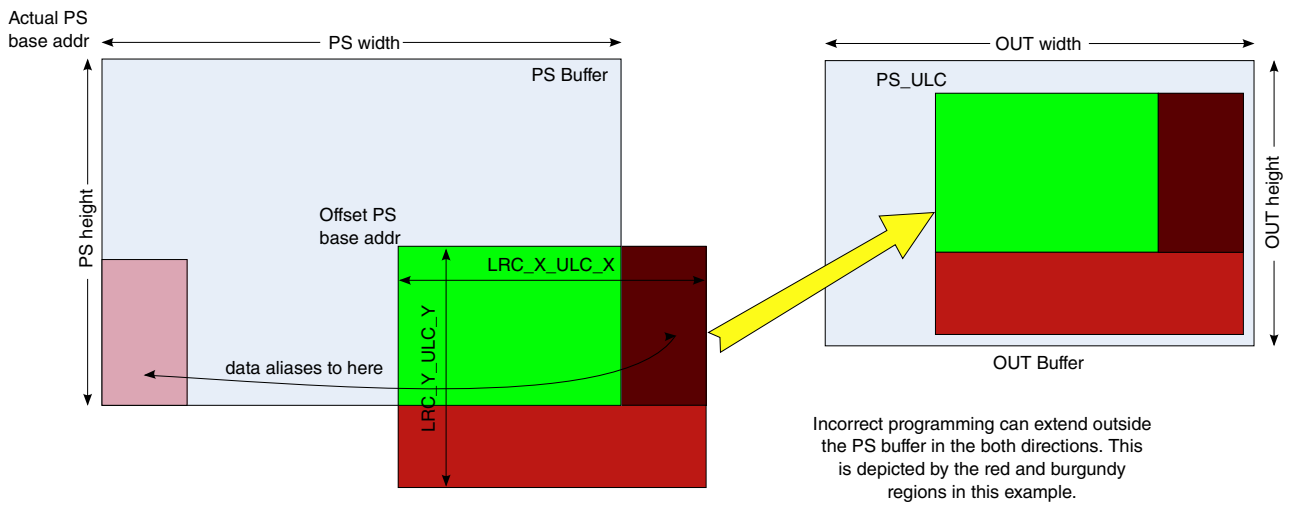


Figure 53-17. Example with Invalid Parameters

53.3.36 Color Key Processing

Pixels may be made transparent to the corresponding AS by using the PS color key registers.

If a PS pixel matches the range specified by the REG_PS_COLORKEYLOW and REG_PS_COLORKEYHIGH registers, the pixel from the associated AS will be displayed. If no AS is present for the pixel, a black pixel will be generated since the default AS pixel is 0x00000000 (transparent black pixel).

The most common use for this is when a bitmap does not support an alpha-field or for applications such as "green screen" where an image is substituted for a solid background color .



Figure 53-18. The PS image (player) and AS image (stadium)

The green portion of the background image can be color keyed to display the contents of the AS buffer for locations that match the color range. For this example, the color range is:

PS Colorkey: 00<R<80 70<G<ff 00<B<80

The resulting image becomes:



Figure 53-19. Resulting Image

53.3.37 In Place Processing (PS buffer is destination buffer)

The PXP also has the ability to process an image and write the resulting buffer back to the original PS buffer. This is referred to as "in place" rendering.

This could be useful for basic blit operations into the PS buffer. IN_PLACE operations are achieved by programming the OUT base address to the pixel location in the PS buffer that marks the upper left pixel of the update region. The actual region that is updated should be indicated by programming the ULC = (0,0) and the LRC = (X,Y). The region bounded by the coordinates will be updated, and the rest of the PS buffer will not be modified.

53.3.38 Alpha Surface (AS) Processing

The AS surface has a complete set of registers that determines how the AS effects the final OUT surface.

Most of the registers that exist for the PS surface also are defined for the AS surface where applicable. This is provided to replicate the SW interface for each PS and AS processes.

53.3.39 Alpha Handling

Alpha values in the AS are embedded in the source image pixels. For AS pixel formats that do not support an alpha value, the pixel is assigned an alpha value of 0xFF (opaque).

This can be modified by the AS control by setting either the ALPHA_MULTIPLY or ALPHA_OVERRIDE bit in the associated AS_CTRL register. If ALPHA_MULTIPLY is enabled, the 8-bit ALPHA value from the AS_CTRL register is multiplied by the source alpha before blending with the PS image. If the ALPHA_OVERRIDE bit is set, the 8-bit ALPHA value is simply substituted for the pixel.

53.3.40 Color Key Processing (AS_CTRL)

The AS_CTRL register also contains an ENABLE_COLORKEY bit that can be used to enable or disable color key substitution for the AS.

When enabled, the pixel values are compared to the ASCOLORKEYLOW and ASCOLORKEYHIGH registers to determine if a match has occurred. When an AS pixel matches the color key range, the pixel from the AS image is considered transparent and the corresponding PS pixel is rendered. If both the PS and AS pixels match their corresponding color key ranges, the AS pixel is displayed unmodified.

AS color keys are handled in a manner similar to PS color keys. The same images used in the PS color key example could be used with the images swapped. In this case, matches on the AS image to the ASCOLORKEY register would display the PS pixels.

53.4 Output Image Processing

Several PXP options affect the resulting output image.

53.4.1 Output Image Size

The PXP generates an output image in the resolution programmed by the REG_OUT_LRC. As the PXP processes pixels, it iterates over the NxN blocks (in output scan-block order) based on the final image resolution.

53.4.2 Output Format

The result of PXP operations are written to the buffer pointed to by the REG_OUT_BUF/ REG_OUT_BUF2 registers. The pixel format is controlled by the REG_OUT_CTRL[FORMAT] bit-field.

32-bit pixels are formed directly from the internal 24-bit representations and 16-bit pixel formats are generated by truncating the internal 24-bit values to the appropriate number of bits. For formats supporting an alpha value, the PXP assigns the alpha using the 8-bit value in the REG_OUT_CTRL[ALPHA] field. For ARGB1555, the most significant alpha bit is appended to the output pixel. Also, for ARGB4444, the most significant nibble is appended to the output pixel. Single and dual buffer YUV output formats are also available. Since each pixel in the data path is represented by a full YUV444 24bpp value, decimation reduces the output in cases of YUV422/420 output formats.

53.4.3 Rotation/Flip operations

The PXP supports four rotation angles in conjunction with vertical and horizontal flip options. The flip operations effectively take place before the rotation.

Rotations of 0, 90, 180, and 270 degrees are supported and any combination of rotation and flip are supported. There is no performance difference between any of these modes of operation.

53.5 Queuing PXP transactions

The PXP supports a primitive ability to queue up one operation while the current operation is running. This is enabled through the use of the REG_NEXT register.

When this register is written, it enables the PXP to reload its current register contents with the data found at the location pointed to by this address when it completes processing of the current frame. This feature may be useful in helping to reduce the interrupt latency in servicing the PXP, especially in cases where the PXPis using the on-chip SRAM buffer handshake (since the PXP must begin generating next frame data immediately).

If the PXP is idle when the REG_NEXT register is written, the PXP treats this as an indication that it should immediately load the values at the pointer and begin processing the frame. This ability should allow software to use the same routines when programming the PXP (so that the first frame doesn't differ from subsequent frames).

When loading values from the NEXT register, all registers in the PXP are reloaded. Some register loads have no effect

After writing the REG_NEXT register, the PXP will set the REG_NEXT[ENABLED] bit of the REG_NEXT register to indicate that the next command has been queued. Software should first check the status of this bit to ensure that a previous command has not been enabled. Likewise, after programming the first frame in a sequence of frames, software should poll this bit until it is sampled logic 1'b0 before queuing the next operation.

The PXP will issue interrupts from frames as they complete, regardless of whether they were started by writing the control registers directly or using the REG_NEXT register. When software receives an interrupt, it should check/clear the PXP's status register as normal, poll the REG_PXP[ENABLED] bit, and then issue the next operation. A queued operation may be cancelled by issuing a CLEAR operation to the REG_PXP[ENABLED] register bit. The SET and TOGGLE operations should never be used with this register.

53.6 Error Handling

The PXP does minimal checking on the control registers, so it is important that these are correctly specified. The PXP does monitor the bus transactions for errors and will report errors in the status register.

Upon receipt of a bus error, the PXP will set the ERROR interrupt and abort any further operations. Bus errors can be generated from any system access that results in an error response returned from the internal SIM Bus errors in the PXP are signaled as either a read or a write error, but do not indicate the failing address. Software may deduce the failing address from the current block status indicators.

53.6.1 Known PXP Limitations/Issues

The PXP has the following known limitations:

1. When using the NEXT register, the interrupt enable setting should remain the same for all frames. If not, the PXP will change the interrupt enable register value and possible cause the loss of an interrupt.
2. Rotations of 180/270 are not supported when performing LCD handshakes

53.7 PXP Memory Map/Register Definition

PXP Hardware Register Format Summary

PXP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20F_0000	Control Register 0 (PXP_CTRL)	32	R/W	C000_0000h	53.7.1/4597
20F_0004	Control Register 0 (PXP_CTRL_SET)	32	R/W	C000_0000h	53.7.1/4597
20F_0008	Control Register 0 (PXP_CTRL_CLR)	32	R/W	C000_0000h	53.7.1/4597
20F_000C	Control Register 0 (PXP_CTRL_TOG)	32	R/W	C000_0000h	53.7.1/4597
20F_0010	Status Register (PXP_STAT)	32	R/W	0000_0000h	53.7.2/4600
20F_0014	Status Register (PXP_STAT_SET)	32	R/W	0000_0000h	53.7.2/4600
20F_0018	Status Register (PXP_STAT_CLR)	32	R/W	0000_0000h	53.7.2/4600
20F_001C	Status Register (PXP_STAT_TOG)	32	R/W	0000_0000h	53.7.2/4600
20F_0020	Output Buffer Control Register (PXP_OUT_CTRL)	32	R/W	0000_0000h	53.7.3/4601
20F_0024	Output Buffer Control Register (PXP_OUT_CTRL_SET)	32	R/W	0000_0000h	53.7.3/4601

Table continues on the next page...

PXP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20F_0028	Output Buffer Control Register (PXP_OUT_CTRL_CLR)	32	R/W	0000_0000h	53.7.3/4601
20F_002C	Output Buffer Control Register (PXP_OUT_CTRL_TOG)	32	R/W	0000_0000h	53.7.3/4601
20F_0030	Output Frame Buffer Pointer (PXP_OUT_BUF)	32	R/W	0000_0000h	53.7.4/4603
20F_0040	Output Frame Buffer Pointer #2 (PXP_OUT_BUF2)	32	R/W	0000_0000h	53.7.5/4604
20F_0050	Output Buffer Pitch (PXP_OUT_PITCH)	32	R/W	0000_0000h	53.7.6/4605
20F_0060	Output Surface Lower Right Coordinate (PXP_OUT_LRC)	32	R/W	0000_0000h	53.7.7/4605
20F_0070	Processed Surface Upper Left Coordinate (PXP_OUT_PS_ULC)	32	R/W	0000_0000h	53.7.8/4606
20F_0080	Processed Surface Lower Right Coordinate (PXP_OUT_PS_LRC)	32	R/W	0000_0000h	53.7.9/4607
20F_0090	Alpha Surface Upper Left Coordinate (PXP_OUT_AS_ULC)	32	R/W	0000_0000h	53.7.10/ 4608
20F_00A0	Alpha Surface Lower Right Coordinate (PXP_OUT_AS_LRC)	32	R/W	0000_0000h	53.7.11/ 4609
20F_00B0	Processed Surface (PS) Control Register (PXP_PS_CTRL)	32	R/W	0000_0000h	53.7.12/ 4610
20F_00B4	Processed Surface (PS) Control Register (PXP_PS_CTRL_SET)	32	R/W	0000_0000h	53.7.12/ 4610
20F_00B8	Processed Surface (PS) Control Register (PXP_PS_CTRL_CLR)	32	R/W	0000_0000h	53.7.12/ 4610
20F_00BC	Processed Surface (PS) Control Register (PXP_PS_CTRL_TOG)	32	R/W	0000_0000h	53.7.12/ 4610
20F_00C0	PS Input Buffer Address (PXP_PS_BUF)	32	R/W	0000_0000h	53.7.13/ 4612
20F_00D0	PS U/Cb or 2 Plane UV Input Buffer Address (PXP_PS_UBUF)	32	R/W	0000_0000h	53.7.14/ 4612
20F_00E0	PS V/Cr Input Buffer Address (PXP_PS_VBUF)	32	R/W	0000_0000h	53.7.15/ 4613
20F_00F0	Processed Surface Pitch (PXP_PS_PITCH)	32	R/W	0000_0000h	53.7.16/ 4614
20F_0100	PS Background Color (PXP_PS_BACKGROUND)	32	R/W	0000_0000h	53.7.17/ 4614
20F_0110	PS Scale Factor Register (PXP_PS_SCALE)	32	R/W	1000_1000h	53.7.18/ 4615
20F_0120	PS Scale Offset Register (PXP_PS_OFFSET)	32	R/W	0000_0000h	53.7.19/ 4616
20F_0130	PS Color Key Low (PXP_PS_CLRKEYLOW)	32	R/W	00FF_FFFFh	53.7.20/ 4617
20F_0140	PS Color Key High (PXP_PS_CLRKEYHIGH)	32	R/W	0000_0000h	53.7.21/ 4618
20F_0150	Alpha Surface Control (PXP_AS_CTRL)	32	R/W	0000_0000h	53.7.22/ 4619

Table continues on the next page...

PXP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20F_0160	Alpha Surface Buffer Pointer (PXP_AS_BUF)	32	R/W	0000_0000h	53.7.23/ 4621
20F_0170	Alpha Surface Pitch (PXP_AS_PITCH)	32	R/W	0000_0000h	53.7.24/ 4622
20F_0180	Overlay Color Key Low (PXP_AS_CLRKEYLOW)	32	R/W	00FF_FFFFh	53.7.25/ 4622
20F_0190	Overlay Color Key High (PXP_AS_CLRKEYHIGH)	32	R/W	0000_0000h	53.7.26/ 4623
20F_01A0	Color Space Conversion Coefficient Register 0 (PXP_CSC1_COEF0)	32	R/W	0400_0000h	53.7.27/ 4624
20F_01B0	Color Space Conversion Coefficient Register 1 (PXP_CSC1_COEF1)	32	R/W	0123_0208h	53.7.28/ 4625
20F_01C0	Color Space Conversion Coefficient Register 2 (PXP_CSC1_COEF2)	32	R/W	079B_076Ch	53.7.29/ 4626
20F_01D0	Color Space Conversion Control Register (PXP_CSC2_CTRL)	32	R/W	0000_0001h	53.7.30/ 4627
20F_01E0	Color Space Conversion Coefficient Register 0 (PXP_CSC2_COEF0)	32	R/W	0000_0000h	53.7.31/ 4628
20F_01F0	Color Space Conversion Coefficient Register 1 (PXP_CSC2_COEF1)	32	R/W	0000_0000h	53.7.32/ 4629
20F_0200	Color Space Conversion Coefficient Register 2 (PXP_CSC2_COEF2)	32	R/W	0000_0000h	53.7.33/ 4629
20F_0210	Color Space Conversion Coefficient Register 3 (PXP_CSC2_COEF3)	32	R/W	0000_0000h	53.7.34/ 4630
20F_0220	Color Space Conversion Coefficient Register 4 (PXP_CSC2_COEF4)	32	R/W	0000_0000h	53.7.35/ 4631
20F_0230	Color Space Conversion Coefficient Register 5 (PXP_CSC2_COEF5)	32	R/W	0000_0000h	53.7.36/ 4631
20F_0240	Lookup Table Control Register (PXP_LUT_CTRL)	32	R/W	8001_0000h	53.7.37/ 4632
20F_0250	Lookup Table Control Register (PXP_LUT_ADDR)	32	R/W	0000_0000h	53.7.38/ 4634
20F_0260	Lookup Table Data Register (PXP_LUT_DATA)	32	R/W	0000_0000h	53.7.39/ 4635
20F_0270	Lookup Table External Memory Address Register (PXP_LUT_EXTMEM)	32	R/W	0000_0000h	53.7.40/ 4636
20F_0280	Color Filter Array Register (PXP_CFA)	32	R/W	0000_0000h	53.7.41/ 4637
20F_0290	Histogram Control Register (PXP_HIST_CTRL)	32	R/W	0000_0020h	53.7.42/ 4637
20F_02A0	2-level Histogram Parameter Register (PXP_HIST2_PARAM)	32	R/W	0000_0F00h	53.7.43/ 4638
20F_02B0	4-level Histogram Parameter Register (PXP_HIST4_PARAM)	32	R/W	0F0A_0500h	53.7.44/ 4639

Table continues on the next page...

PXP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20F_02C0	8-level Histogram Parameter 0 Register (PXP_HIST8_PARAM0)	32	R/W	0604_4000h	53.7.45/ 4640
20F_02D0	8-level Histogram Parameter 1 Register (PXP_HIST8_PARAM1)	32	R/W	0F0D_0B09h	53.7.46/ 4641
20F_02E0	16-level Histogram Parameter 0 Register (PXP_HIST16_PARAM0)	32	R/W	0302_0100h	53.7.47/ 4642
20F_02F0	16-level Histogram Parameter 1 Register (PXP_HIST16_PARAM1)	32	R/W	0706_0504h	53.7.48/ 4643
20F_0300	16-level Histogram Parameter 2 Register (PXP_HIST16_PARAM2)	32	R/W	0B0A_0908h	53.7.49/ 4644
20F_0310	16-level Histogram Parameter 3 Register (PXP_HIST16_PARAM3)	32	R/W	0F0E_0D0Ch	53.7.50/ 4645
20F_0320	PXP Power Control Register (PXP_POWER)	32	R/W	0000_0000h	53.7.51/ 4646
20F_0400	Next Frame Pointer (PXP_NEXT)	32	R/W	0000_0000h	53.7.52/ 4646

53.7.1 Control Register 0 (PXP_CTRL*n*)

The CTRL register contains controls for the PXP module.

PXP_CTRL: 0x000

PXP_CTRL_SET: 0x004

PXP_CTRL_CLR: 0x008

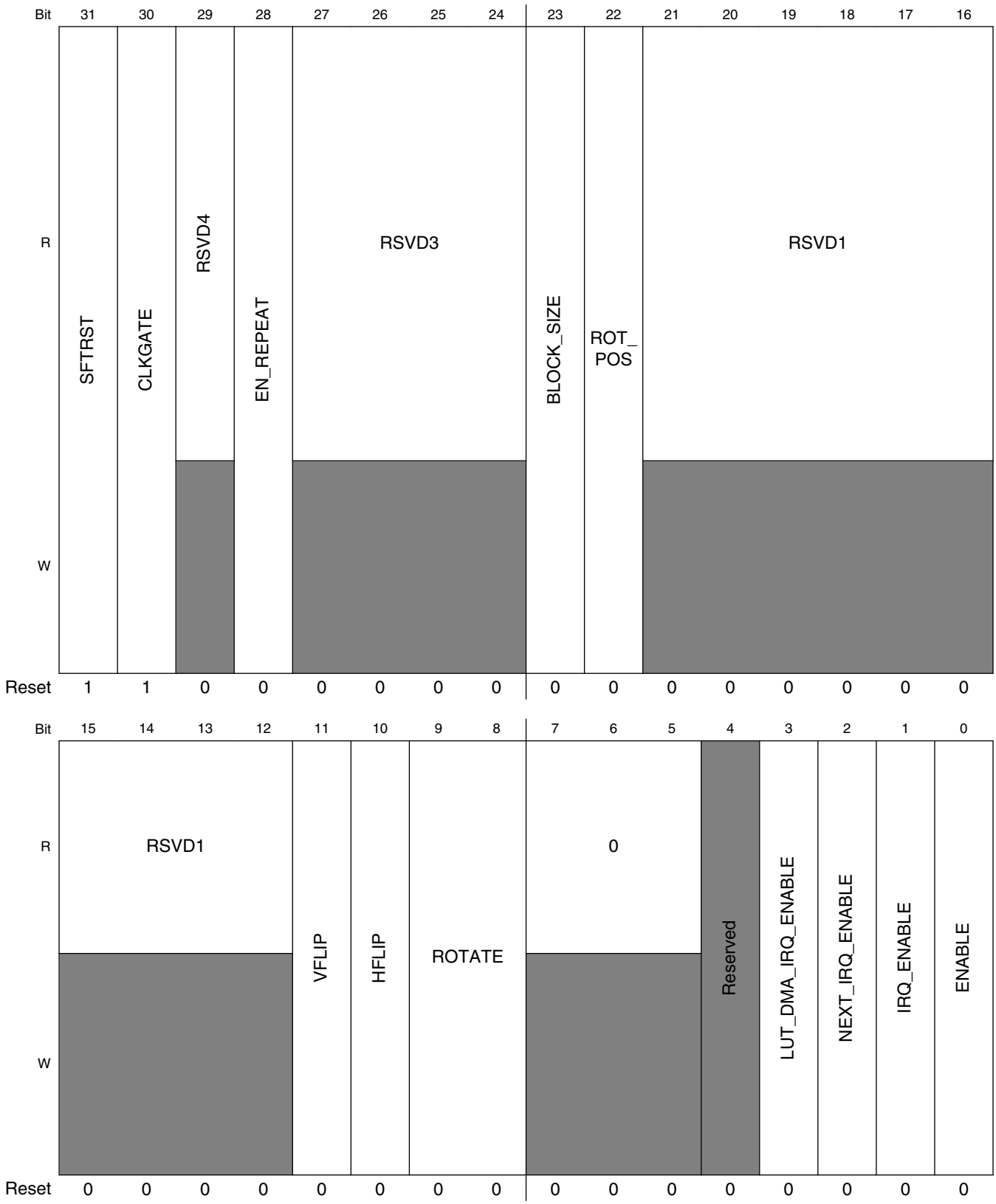
PXP_CTRL_TOG: 0x00C

The Control register contains the primary controls for the PXP block. The present bits indicate which of the sub-features of the block are present in the hardware.

EXAMPLE

```
PXP_CTRL_SET(BM_PXP_CTRL_SFTRST);
PXP_CTRL_CLR(BM_PXP_CTRL_SFTRST | BM_PXP_CTRL_CLKGATE);
```

Address: 20F_0000h base + 0h offset + (4d × i), where i=0d to 3d



PXP_CTRL*n* field descriptions

Field	Description
31 SFTRST	Set this bit to zero to enable normal PXP operation. Set this bit to one (default) to disable clocking with the PXP and hold it in its reset (lowest power) state. This bit can be turned on and then off to reset the PXP block to its default state.
30 CLKGATE	This bit must be set to zero for normal operation. When set to one it gates off the clocks to the block.
29 RSVD4	Reserved, always set to zero.
28 EN_REPEAT	Enable the PXP to run continuously. When this bit is set, the PXP will repeat based on the current configuration register settings. If this bit is not set, the PXP will complete the process and enter the idle state ready to accept the next frame to be processed. When the PXP NEXT feature is used, it has priority over the REPEAT mode, in that the new register settings are fetched first, and then the next PXP operation will continue.
27–24 RSVD3	Reserved, always set to zero.
23 BLOCK_SIZE	Select the block size to process. 0x0 **8X8** — Process 8x8 pixel blocks. 0x1 **16X16** — Process 16x16 pixel blocks.
22 ROT_POS	This bit controls where rotation will occur in the PXP datapath. Setting this bit to 1'b0 will place the rotation resources at the output stage of the PXP data path. Image compositing will occur before pixels are processed for rotation. Setting this bit to a 1'b1 will place the rotation resources before image composition. Only the PS can be rotated in this configuration and AS will not be rotated.
21–12 RSVD1	Reserved, always set to zero.
11 VFLIP	Indicates that the output buffer should be flipped vertically (effect applied before rotation).
10 HFLIP	Indicates that the output buffer should be flipped horizontally (effect applied before rotation).
9–8 ROTATE	Indicates the clockwise rotation to be applied at the output buffer. The rotation effect is defined as occurring after the FLIP_X and FLIP_Y permutation. 0x0 **ROT_0** — 0x1 **ROT_90** — 0x2 **ROT_180** — 0x3 **ROT_270** —
7–5 Reserved	This read-only field is reserved and always has the value 0.
4 ENABLE_LCD_ HANDSHAKE	This field is reserved. Reserved.
3 LUT_DMA_IRQ_ ENABLE	LUT DMA interrupt enable. When set, the PXP will issue an interrupt when the LUT DMA has finished transferring data.
2 NEXT_IRQ_ ENABLE	Next command interrupt enable. When set, the PXP will issue an interrupt when a queued command initiated by a write to the PXP_NEXT register has been loaded into the PXP's registers. This interrupt also indicates that a new command may now be queued.
1 IRQ_ENABLE	Interrupt enable. NOTE: When using the PXP_NEXT functionality to reprogram the PXP, the new value of this bit will be used and may therefore enable or disable an interrupt unintentionally.

Table continues on the next page...

PXP_CTRL*n* field descriptions (continued)

Field	Description
0 ENABLE	Enables PXP operation with specified parameters. The ENABLE bit will remain set while the PXP is active and will be cleared after the current operation completes. Software should use the IRQ bit in the PXP_STAT when polling for PXP completion.

53.7.2 Status Register (PXP_STAT*n*)

The PXP Interrupt Status register provides interrupt status information.

PXP_STAT: 0x010

PXP_STAT_SET: 0x014

PXP_STAT_CLR: 0x018

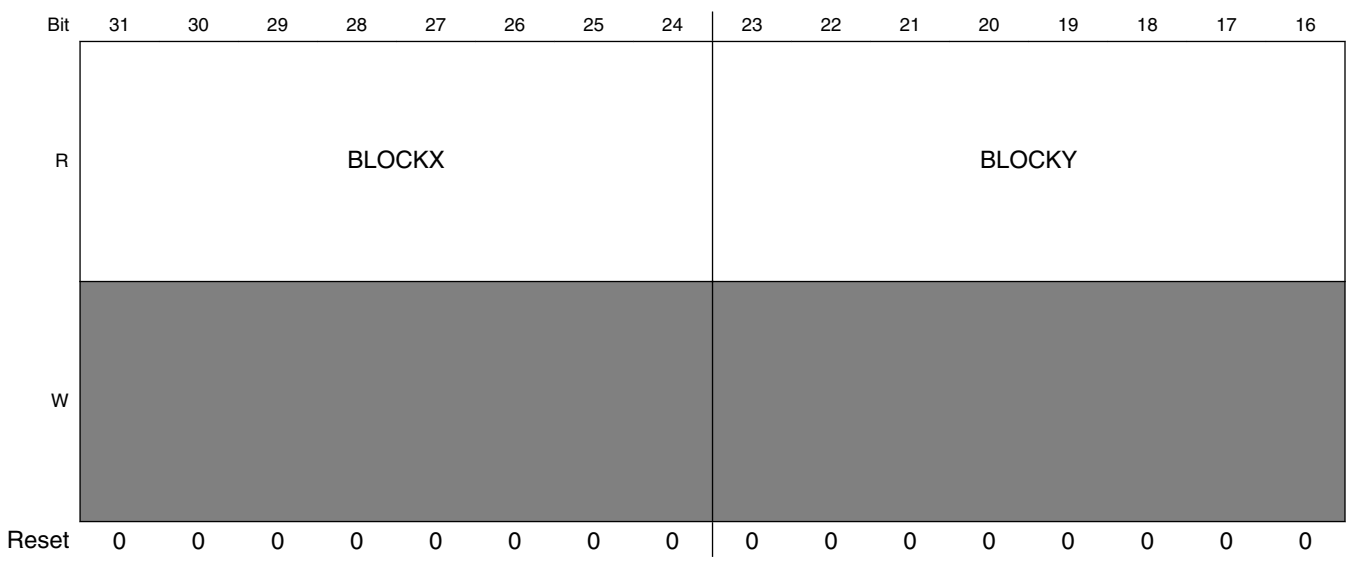
PXP_STAT_TOG: 0x01C

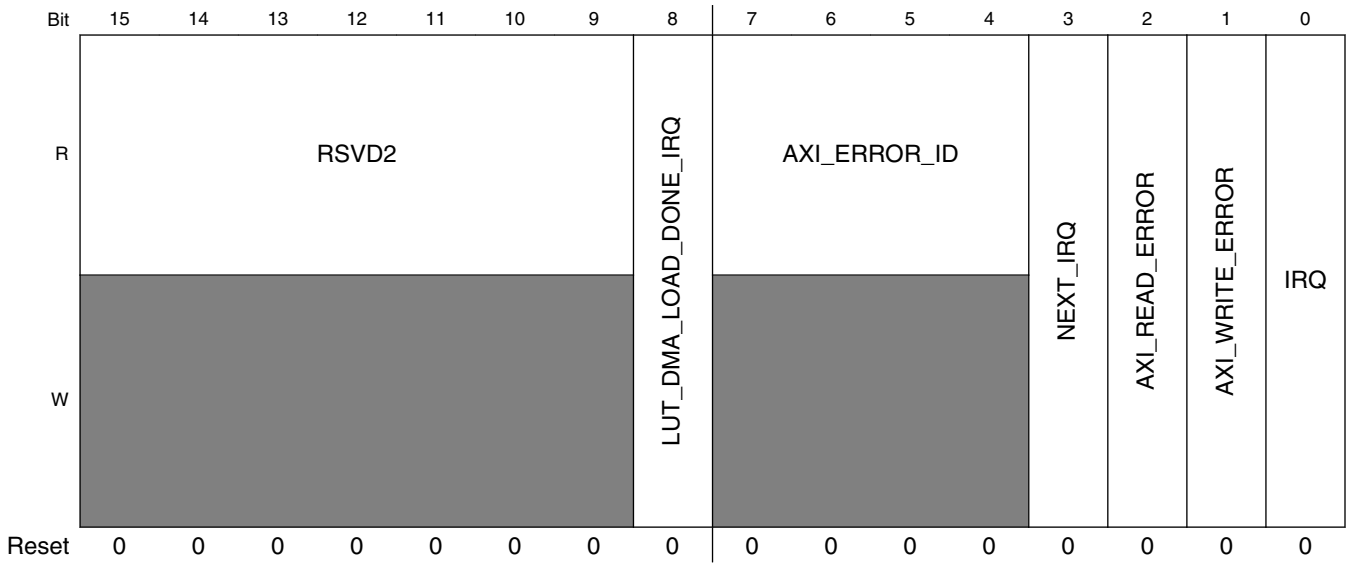
This register provides PXP interrupt status and the current X/Y block coordinate that is being processed.

EXAMPLE

```
PXP_STAT_CLR(BM_PXP_STAT_IRQ);  // clear CSC interrupt
```

Address: 20F_0000h base + 10h offset + (4d × i), where i=0d to 3d





PXP_STAT*n* field descriptions

Field	Description
31–24 BLOCKX	Indicates the X coordinate of the block currently being rendered.
23–16 BLOCKY	Indicates the X coordinate of the block currently being rendered.
15–9 RSVD2	Reserved, always set to zero.
8 LUT_DMA_ LOAD_DONE_ IRQ	Indicates that the LUT DMA transfer has completed.
7–4 AXI_ERROR_ID	Indicates the AXI ID of the failing bus operation.
3 NEXT_IRQ	Indicates that a command issued with the "Next Command" functionality has been issued and that a new command may be initiated with a write to the PXP_NEXT register.
2 AXI_READ_ ERROR	Indicates PXP encountered an AXI read error and processing has been terminated.
1 AXI_WRITE_ ERROR	Indicates PXP encountered an AXI write error and processing has been terminated.
0 IRQ	Indicates current PXP interrupt status. The IRQ is routed through the pxp_irq when the IRQ_ENABLE bit in the control register is set.

53.7.3 Output Buffer Control Register (PXP_OUT_CTRL*n*)

The OUT_CTRL register contains controls for the Output Buffer.

PXP_OUT_CTRL: 0x020

PXP_OUT_CTRL_SET: 0x024

PXP_OUT_CTRL_CLR: 0x028

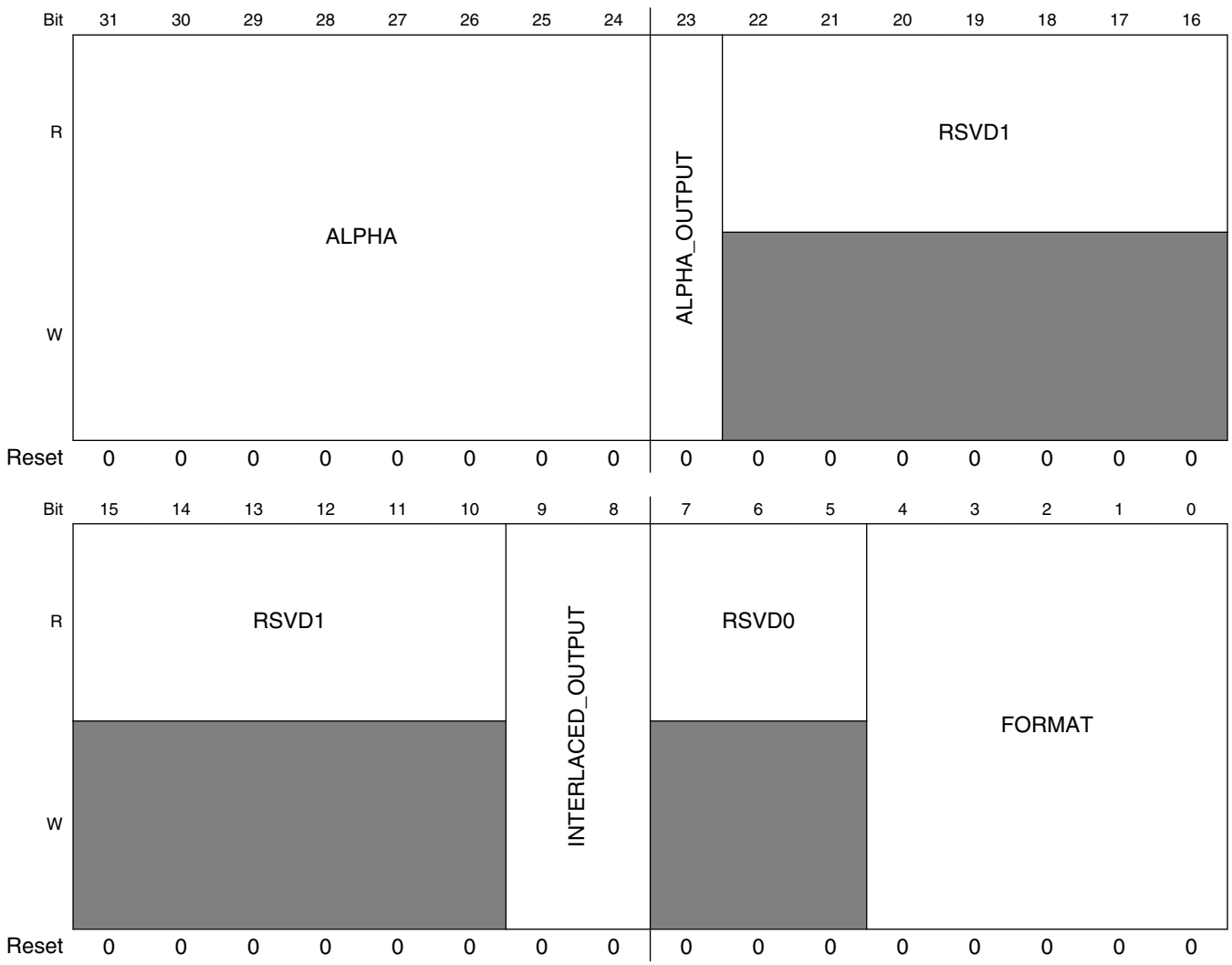
PXP_OUT_CTRL_TOG: 0x02C

The Control register contains the primary controls for the PXP block. The present bits indicate which of the sub-features of the block are present in the hardware.

EXAMPLE

```
PXP_CTRL_SET(BM_PXP_CTRL_SFTRST);
PXP_CTRL_CLR(BM_PXP_CTRL_SFTRST | BM_PXP_CTRL_CLKGATE);
```

Address: 20F_0000h base + 20h offset + (4d × i), where i=0d to 3d



PXP_OUT_CTRL*n* field descriptions

Field	Description
31–24 ALPHA	When generating an output buffer with an alpha component, the value in this field will be used when enabled to override the alpha passed through the pixel data pipeline.
23 ALPHA_ OUTPUT	Indicates that alpha component in output buffer pixels should be overwritten by PXP_OUT_CTRL[ALPHA]. If 0, retain their alpha value from the computed alpha for that pixel.
22–10 RSVD1	Reserved, always set to zero.
9–8 INTERLACED_ OUTPUT	Determines how the PXP writes it's output data. Output interlacing should not be used in conjunction with input interlacing. Splitting frames into fields is most efficient using output interlacing. 2-plane output formats AND interlaced output is NOT supported. 0x0 **PROGRESSIVE** — All data written in progressive format to the OUTBUF Pointer. 0x1 **FIELD0** — Interlaced output: only data for field 0 is written to the OUTBUF Pointer. 0x2 **FIELD1** — Interlaced output: only data for field 1 is written to the OUTBUF2 Pointer. 0x3 **INTERLACED** — Interlaced output: data for field 0 is written to OUTBUF and data for field 1 is written to OUTBUF2.
7–5 RSVD0	Reserved, always set to zero.
FORMAT	Output framebuffer format. The UV byte lanes are synonymous with CbCr byte lanes for YUV output pixel formats. For example, the YUV2P420 format should be selected when the output is YCbCr 2-plane 420 output format. 0x0 **ARGB8888** — 32-bit pixels 0x4 **RGB888** — 32-bit pixels (unpacked 24-bit pixel in 32 bit DWORD.) 0x5 **RGB888P** — 24-bit pixels (packed 24-bit format) 0x8 **ARGB1555** — 16-bit pixels 0x9 **ARGB4444** — 16-bit pixels 0xC **RGB555** — 16-bit pixels 0xD **RGB444** — 16-bit pixels 0xE **RGB565** — 16-bit pixels 0x10 **YUV1P444** — 32-bit pixels (1-plane XYUV unpacked) 0x12 **UYVY1P422** — 16-bit pixels (1-plane U0,Y0,V0,Y1 interleaved bytes) 0x13 **VYUY1P422** — 16-bit pixels (1-plane V0,Y0,U0,Y1 interleaved bytes) 0x14 **Y8** — 8-bit monochrome pixels (1-plane Y luma output) 0x15 **Y4** — 4-bit monochrome pixels (1-plane Y luma, 4 bit truncation) 0x18 **YUV2P422** — 16-bit pixels (2-plane UV interleaved bytes) 0x19 **YUV2P420** — 16-bit pixels (2-plane UV) 0x1A **YVU2P422** — 16-bit pixels (2-plane VU interleaved bytes) 0x1B **YVU2P420** — 16-bit pixels (2-plane VU)

53.7.4 Output Frame Buffer Pointer (PXP_OUT_BUF)

Output Framebuffer Pointer. This register points to the beginning of the output frame buffer. This pointer is used for progressive format and field 0 when generating interlaced output.

This register is used by the logic to point to the current output location for the output frame buffer.

EXAMPLE

```
PXP_OUT_BUF_WR( buffer );
```

Address: 20F_0000h base + 30h offset = 20F_0030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_OUT_BUF field descriptions

Field	Description
ADDR	Current address pointer for the output frame buffer. The address can have any byte alignment. 64B alignment is recommended for optimal performance.

53.7.5 Output Frame Buffer Pointer #2 (PXP_OUT_BUF2)

Output Framebuffer Pointer #2. This register points to the beginning of the output frame buffer for either field 1 when generating interlaced output or for the UV buffer when in YUV 2-plane output modes. Both interlaced output AND 2-plane output modes are not supported in a single PXP operation.

This register is used by the logic to point to the current output location for the field 1 or UV output frame buffer.

EXAMPLE

```
PXP_OUT_BUF_WR( field0 );  // buffer for interlaced field 0
PXP_OUT_BUF2_WR( field1 ); // buffer for interlaced field 1
```

Address: 20F_0000h base + 40h offset = 20F_0040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_OUT_BUF2 field descriptions

Field	Description
ADDR	Current address pointer for the output frame buffer. The address can have any byte alignment. 64B alignment is recommended for optimal performance.

53.7.6 Output Buffer Pitch (PXP_OUT_PITCH)

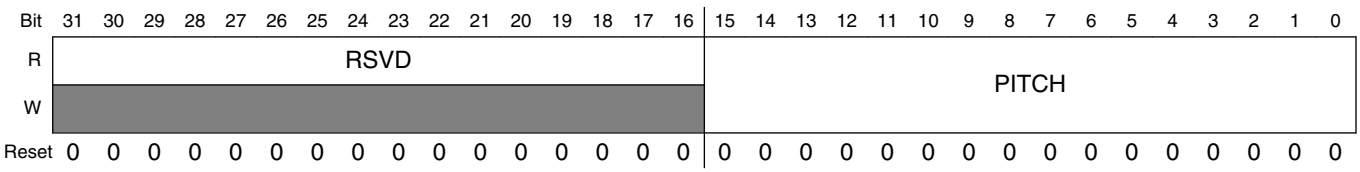
This register contains the output buffer pitch in bytes.

Any byte value will indicate the vertical pitch. This value will be used in output pixel address calculations.

EXAMPLE

```
PXP_OUT_PITCH_WR( 68 * 4 );  // The output buffer pitch is 68 pixels times 32 bits per pixel
```

Address: 20F_0000h base + 50h offset = 20F_0050h



PXP_OUT_PITCH field descriptions

Field	Description
31–16 RSVD	Reserved, always set to zero.
PITCH	Indicates the number of bytes in memory between two vertically adjacent pixels.

53.7.7 Output Surface Lower Right Coordinate (PXP_OUT_LRC)

This register contains the size, or lower right coordinate, of the output buffer NOT rotated. It is implied that the upper left coordinate of the output surface is always [0,0]. When rotating the framebuffer, the PXP will automatically swap the X/Y, or WIDTH/HEIGHT, to accomodate the rotated size.

This register sets the size of the output frame buffer in pixels, not blocks. The frame buffer need not be a multiple of NxN pixels. Partial blocks will be written for output frame buffer sizes that are not divisable by N pixels in either dimension.

EXAMPLE

```
PXP_OUT_LRC[X]=319;   // set width of output frame buffer to 320 pixels
PXP_OUT_LRC[Y]=243;  // set height of output frame buffer to 244 pixels which is not
divisible by block size N

PXP_OUT_LRC_WR( BF_PXP_OUT_LRC_X(319)  |  BF_PXP_OUT_LRC_Y(243) );
```

Address: 20F_0000h base + 60h offset = 20F_0060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVD1								X							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD0								Y							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_OUT_LRC field descriptions

Field	Description
31–30 RSVD1	Reserved, always set to zero.
29–16 X	Indicates number of horizontal PIXELS in the output surface (non-rotated). The output buffer pixel width minus 1 should be programmed. The image size is not required to be a multiple of 8 pixels. The PXP will clip the pixel output at this boundary.
15–14 RSVD0	Reserved, always set to zero.
Y	Indicates the number of vertical PIXELS in the output surface (non-rotated). The output buffer pixel height minus 1 should be programmed. The image size is not required to be a multiple of 8 pixels. The PXP will clip the pixel output at this boundary.

53.7.8 Processed Surface Upper Left Coordinate (PXP_OUT_PS_ULC)

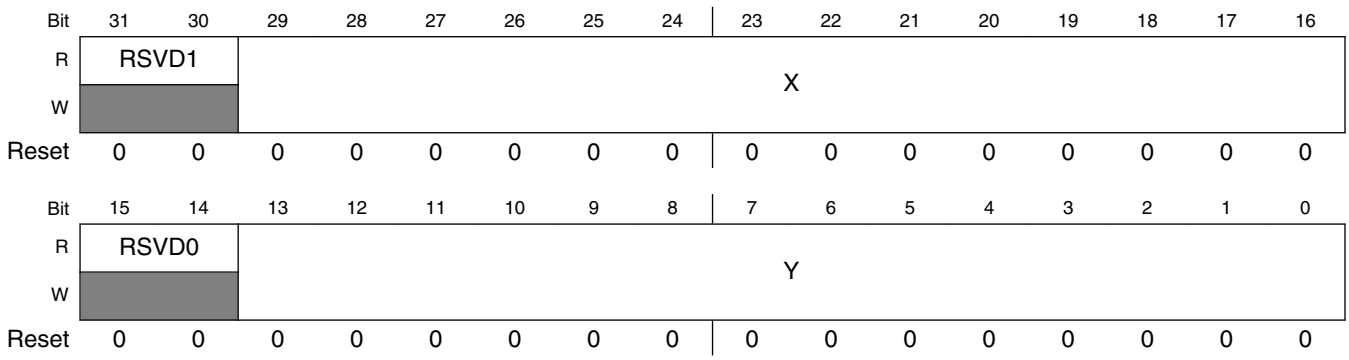
This register contains the upper left pixel coordinate for the Processed Surface in the OUTPUT buffer.

This register contains the upper left coordinate of the Processed Surface in the output frame buffer (in pixels). Values that are within the PXP_OUT_LRC X,Y extents are valid. The lowest valid value for these fields is 0,0. If the value of the PXP_OUT_PS_ULC is greater than the PXP_OUT_LRC, then no PS pixels will be fetched from memory, but only PXP_PS_BACKGROUND pixels will be processed by the PS engine. Pixel locations that are greater than or equal to the PS upper left coordinates, less than or equal to the PS lower right coordinates, and within the PXP_OUT_LRC extents will use the PS to render pixels into the output buffer.

EXAMPLE

```
PXP_OUT_PS_ULC_WR(0,0x0002_0002); // Processed Surface upper left coordinate at (X,Y) =
2,2.  The PS surface will not effect pixels in the first and second row and column of the
output buffer.
```

Address: 20F_0000h base + 70h offset = 20F_0070h



PXP_OUT_PS_ULC field descriptions

Field	Description
31–30 RSVD1	Reserved, always set to zero.
29–16 X	This field indicates the upper left X-coordinate (in pixels) of the processed surface (PS) in the output buffer.
15–14 RSVD0	Reserved, always set to zero.
Y	This field indicates the upper left Y-coordinate (in pixels) of the processed surface in the output buffer.

53.7.9 Processed Surface Lower Right Coordinate (PXP_OUT_PS_LRC)

This register contains the lower right extent for the Processed Surface in the OUTPUT buffer.

This register contains the lower right coordinate of the Processed Surface in the output frame buffer (in pixels). Values that are within the PXP_OUT_LRC X,Y extents are valid. The lowest valid value for these fields is 0,0. Pixel locations that are greater than or equal to the PS upper left coordinates, less than or equal to the PS lower right coordinates, and within the PXP_OUT_LRC extents will use the PS to render pixels into the output buffer.

EXAMPLE

```
PXP_OUT_PS_ULC_WR(0,0x03FF_03FF); // With this UL/LR pair of pixel coordinates, only one
pixel at OUT[X,Y]=1023,1023 will use the PS to contribute to its value.
PXP_OUT_PS_LRC_WR(0,0x03FF_03FF);
```

Address: 20F_0000h base + 80h offset = 20F_0080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVD1								X							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD0								Y							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_OUT_PS_LRC field descriptions

Field	Description
31–30 RSVD1	Reserved, always set to zero.
29–16 X	This field indicates the lower right X-coordinate (in pixels) of the processed surface (PS) in the output frame buffer.
15–14 RSVD0	Reserved, always set to zero.
Y	This field indicates the lower right Y-coordinate (in pixels) of the processed surface in the output frame buffer.

53.7.10 Alpha Surface Upper Left Coordinate (PXP_OUT_AS_ULC)

This register contains the upper left location for the Alpha Surface in the output buffer.

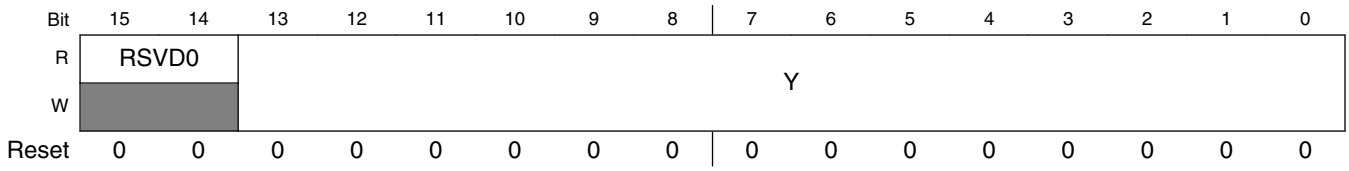
This register contains the upper left coordinate of AS in the output frame buffer (in pixels). Values that are within the PXP_OUT_LRC X,Y extents are valid. The lowest valid value for these fields is 0,0. Pixel locations that are greater than or equal to the upper left coordinates will use the AS to render pixels in the output buffer.

EXAMPLE

```
PXP_OUT_AS_ULC_WR(0,0x0001_0001); // Alpha Surface upper left coordinate at (X,Y) = 1,1.
The AS surface will not effect pixels in the first row or first column of the output buffer.
```

Address: 20F_0000h base + 90h offset = 20F_0090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVD1								X							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



PXP_OUT_AS_ULC field descriptions

Field	Description
31–30 RSVD1	Reserved, always set to zero.
29–16 X	This field indicates the upper left X-coordinate (in pixels) of the alpha surface (AS) in the output frame buffer.
15–14 RSVD0	Reserved, always set to zero.
Y	This field indicates the upper left Y-coordinate (in pixels) of the alpha surface in the output frame buffer.

53.7.11 Alpha Surface Lower Right Coordinate (PXP_OUT_AS_LRC)

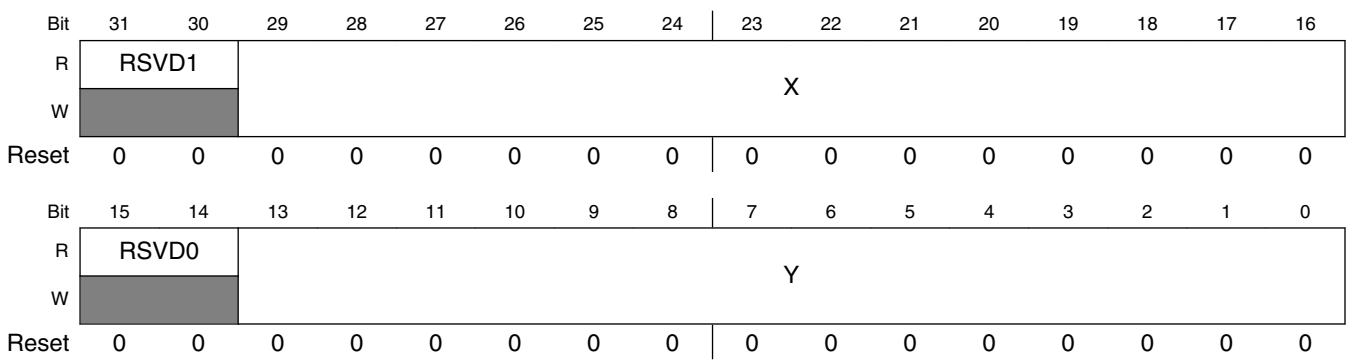
This register contains the lower right extent for Alpha Surface in the output buffer.

This register contains the lower right coordinate of AS in the output frame buffer (in pixels). Values that are within the PXP_OUT_LRC X,Y extents are valid. The lowest valid value for these fields is 0,0. Pixel locations that are less than or equal to the lower right coordinates will use the AS to render pixels in the output buffer.

EXAMPLE

```
PXP_AS_LRC_WR(0,0x03FF_03FF); // Alpha Surface lower right coordinate at (X,Y) = 1023,1023.
```

Address: 20F_0000h base + A0h offset = 20F_00A0h



PXP_OUT_AS_LRC field descriptions

Field	Description
31–30 RSVD1	Reserved, always set to zero.
29–16 X	This field indicates the lower right X-coordinate (in pixels) of the alpha surface (AS) in the output frame buffer.
15–14 RSVD0	Reserved, always set to zero.
Y	This field indicates the lower right Y-coordinate (in pixels) of the alpha surface in the output frame buffer.

53.7.12 Processed Surface (PS) Control Register (PXP_PS_CTRL*n*)

The PS_CTRL register contains controls for the Processed Surface Buffer.

PXP_PS_CTRL: 0x0B0

PXP_PS_CTRL_SET: 0x0b4

PXP_PS_CTRL_CLR: 0x0B8

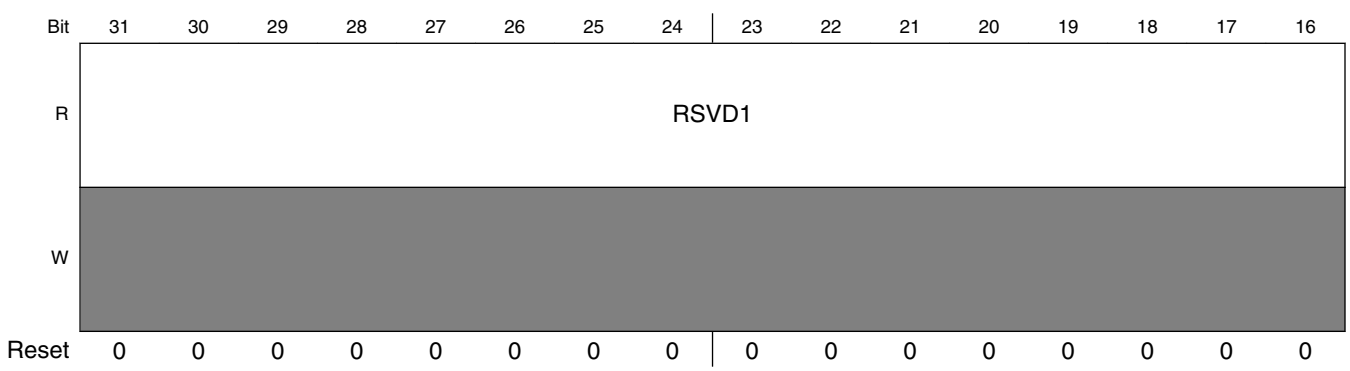
PXP_PS_CTRL_TOG: 0x0BC

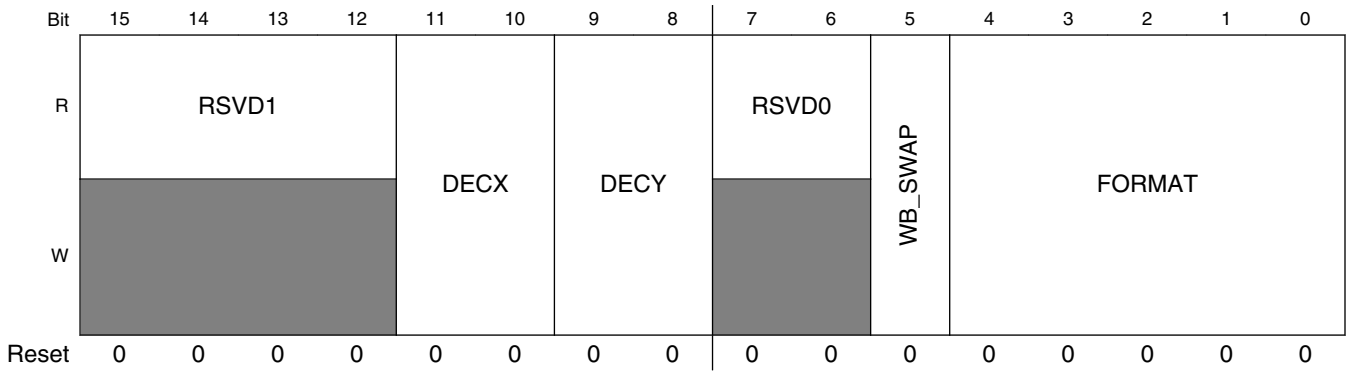
The Control register contains the primary controls for the PXP block. The present bits indicate which of the sub-features of the block are present in the hardware.

EXAMPLE

```
PXP_CTRL_SET(BM_PXP_CTRL_SFTRST);
PXP_CTRL_CLR(BM_PXP_CTRL_SFTRST | BM_PXP_CTRL_CLKGATE);
```

Address: 20F_0000h base + B0h offset + (4d × i), where i=0d to 3d





PXP_PS_CTRL*n* field descriptions

Field	Description
31–12 RSVD1	Reserved, always set to zero.
11–10 DECX	Horizontal pre decimation filter control. 0x0 **DISABLE** — Disable pre-decimation filter. 0x1 **DECX2** — Decimate PS by 2. 0x2 **DECX4** — Decimate PS by 4. 0x3 **DECX8** — Decimate PS by 8.
9–8 DECY	Verticle pre decimation filter control. 0x0 **DISABLE** — Disable pre-decimation filter. 0x1 **DECY2** — Decimate PS by 2. 0x2 **DECY4** — Decimate PS by 4. 0x3 **DECY8** — Decimate PS by 8.
7–6 RSVD0	Reserved, always set to zero.
5 WB_SWAP	Swap bytes in words. For each 16 bit word, the two bytes will be swapped.
FORMAT	PS buffer format. To select between YUV and YCbCr formats, see bit 31 of the CSC1_COEF0 register. 0x4 **RGB888** — 32-bit pixels (unpacked 24-bit format) 0xC **RGB555** — 16-bit pixels 0xD **RGB444** — 16-bit pixels 0xE **RGB565** — 16-bit pixels 0x10 **YUV1P444** — 32-bit pixels (1-plane XYUV unpacked) 0x12 **UYVY1P422** — 16-bit pixels (1-plane U0,Y0,V0,Y1 interleaved bytes) 0x13 **VYUY1P422** — 16-bit pixels (1-plane V0,Y0,U0,Y1 interleaved bytes) 0x14 **Y8** — 8-bit monochrome pixels (1-plane Y luma output) 0x15 **Y4** — 4-bit monochrome pixels (1-plane Y luma, 4 bit truncation) 0x18 **YUV2P422** — 16-bit pixels (2-plane UV interleaved bytes) 0x19 **YUV2P420** — 16-bit pixels (2-plane UV) 0x1A **YVU2P422** — 16-bit pixels (2-plane VU interleaved bytes) 0x1B **YVU2P420** — 16-bit pixels (2-plane VU) 0x1E **YUV422** — 16-bit pixels (3-plane format) 0x1F **YUV420** — 16-bit pixels (3-plane format)

53.7.13 PS Input Buffer Address (PXP_PS_BUF)

PS Input Buffer Address. This should be programmed to the starting address of the RGB data or Y (luma) data for the PS plane.

This register contains the pointer to the Luma/RGB buffer. If the application requires an offset into the PS buffer, then this address can be set so that the desired offset is achieved. Any byte address is valid. For best performance, 64B alignment is recommended.

EXAMPLE

```
PXP_PS_BUF_WR(image_rgb); // RGB image
PXP_PS_BUF_WR(image_y);   // Y (luma) image data
PXP_PS_UBUF_WR(image_u);  // U (Cb) image data
PXP_PS_VBUF_WR(image_v);  // V (Cr) image data
```

Address: 20F_0000h base + C0h offset = 20F_00C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_PS_BUF field descriptions

Field	Description
ADDR	Address pointer for the PS RGB or Y (luma) input buffer.

53.7.14 PS U/Cb or 2 Plane UV Input Buffer Address (PXP_PS_UBUF)

PS Chroma (U/Cb/UV) Input Buffer Address. This register points to the beginning of the PS U/Cb input buffer. In two plane operation, this register points to the beginning of the PS UV chroma input buffer.

This register contains the pointer to the Chroma U/Cb or 2 plane UV buffer. This register is unused when processing 1-plane buffer formats. If the application requires an offset into the PS buffer, then this address can be set so that the desired offset is achieved. Any byte address is valid. For best performance, 64B alignment is recommended.

EXAMPLE

```
PXP_PS_BUF_WR(image_y);   // Y (luma) image data
PXP_PS_UBUF_WR(image_u);  // U (Cb) image data
PXP_PS_VBUF_WR(image_v);  // V (Cr) image data
```

Address: 20F_0000h base + D0h offset = 20F_00D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_PS_UBUF field descriptions

Field	Description
ADDR	Address pointer for the PS U/Cb or 2 plane UV Chroma input buffer.

53.7.15 PS V/Cr Input Buffer Address (PXP_PS_VBUF)

PS Chroma (V/Cr) Input Buffer Address. This register points to the beginning of the PS V/Cr input buffer. In one or two plane operation, this register is not used. In monochrome modes Y8 and Y4, the low 16 bits are used as the U/V data in the datapath instead of sourcing U/V data from external buffers. In this case, it represents a fixed value for U/V data.

This register contains the pointer to the Chroma V/Cr buffer. For Y8/Y4 modes, the low 16 bits are used as the monochrome U and V values in the data path. Bits [15:8] represent the U data byte, and bits]7:0] represent the V data byte. Other than with Y8/Y4 input buffer formats, this register is unused when processing 1 or 2-plane buffer formats. If the application requires an offset into the PS buffer, then this address can be set so that the desired offset is achieved. Any byte address is valid. For best performance, 64B alignment is recommended.

EXAMPLE

```
PXP_PS_BUF_WR(image_y);   // Y (luma) image data
PXP_PS_UBUF_WR(image_u);  // U (Cb) image data
PXP_PS_VBUF_WR(image_v);  // V (Cr) image data
```

Address: 20F_0000h base + E0h offset = 20F_00E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_PS_VBUF field descriptions

Field	Description
ADDR	Address pointer for the PS V/Cr Chroma input buffer.

53.7.16 Processed Surface Pitch (PXP_PS_PITCH)

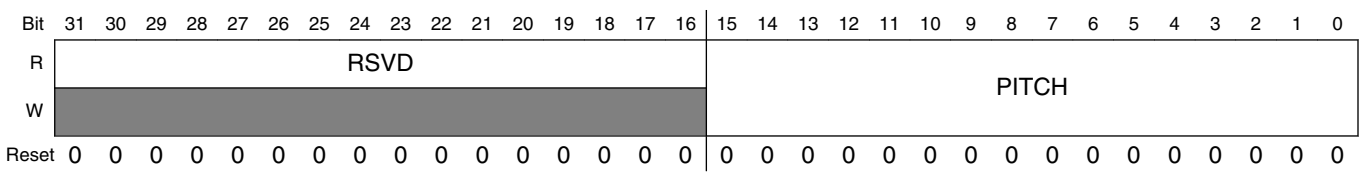
This register contains the processed surface pitch in bytes.

Any byte value will indicate the vertical pitch of the PS source frame buffer. This value will be used in PS pixel address calculations. This value has no relation to the UL and LR registers. It specifies how many bytes are between two vertically adjacent pixels in the input PS surface. For multi-plane formats, the Y buffer pitch should be programmed. For 2-plane YUV422, the UV pitch is the same as the Y pitch. For 3-plane YUV422, the U and V pitch is 1/2 the Y pitch. For 2-plane YUV420, the UV pitch is 1/2 the Y pitch. For 3-plane YUV420, the U and V pitch is 1/4 the Y pitch. All source buffers should comply with these U and V resolution reductions with respect to their Y source buffers.

EXAMPLE

```
PXP_PS_PITCH_WR( 64 * 4 );  // The output buffer pitch is 64 pixels times 32 bits per pixel
```

Address: 20F_0000h base + F0h offset = 20F_00F0h



PXP_PS_PITCH field descriptions

Field	Description
31–16 RSVD	Reserved, always set to zero.
PITCH	Indicates the number of bytes in memory between two vertically adjacent pixels.

53.7.17 PS Background Color (PXP_PS_BACKGROUND)

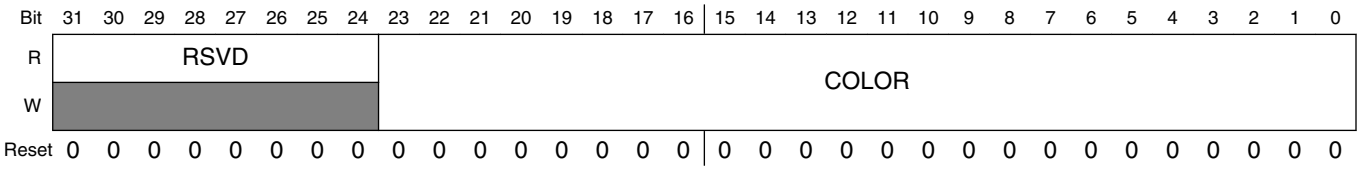
PS Background Pixel Color. This register provides a pixel value used when processing pixels outside of the region specified by the PS Coordinate registers. This value can effectively be used to set the color of the letterboxing region around the PS image.

This register contains a pixel value to be used for any PS pixels that fall outside the PS extents. This is effectively a background or letterbox color. The CSC1 control and datapath pixel format should be considered when selecting the background color.

EXAMPLE

```
PXP_PS_BACKGROUND_WR(0x00000000);  // letterbox is black
PXP_PS_BACKGROUND_WR(0x00800000);  // letterbox is dark red
PXP_PS_BACKGROUND_WR(0x00008000);  // letterbox is dark green
PXP_PS_BACKGROUND_WR(0x00000080);  // letterbox is dark blue
```

Address: 20F_0000h base + 100h offset = 20F_0100h



PXP_PS_BACKGROUND field descriptions

Field	Description
31–24 RSVD	Reserved, always set to zero.
COLOR	Background color (in 24bpp format) for any pixels not within the buffer range specified by the PS ULC/ LRC.

53.7.18 PS Scale Factor Register (PXP_PS_SCALE)

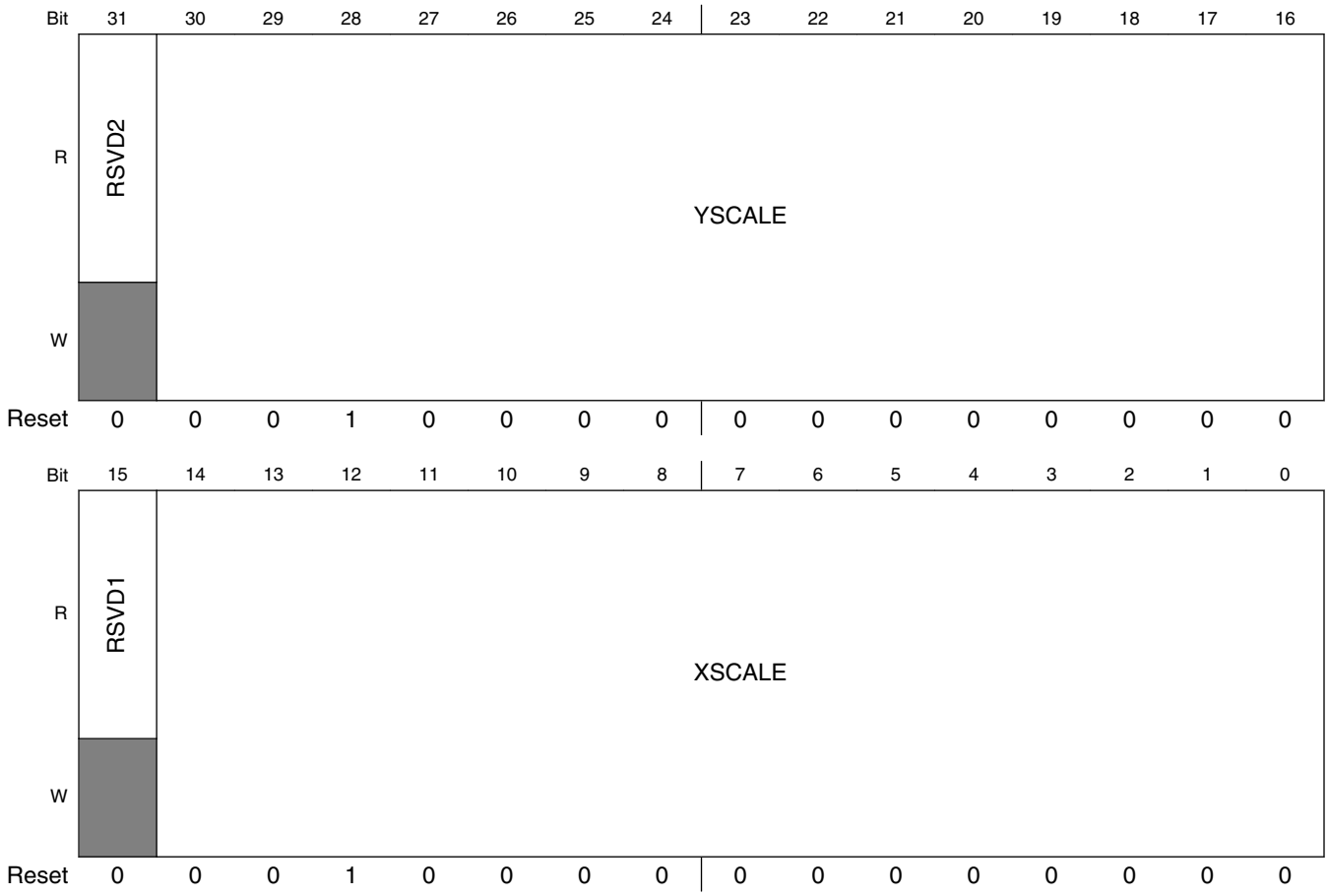
PS Scale Factor. This register provides the scale factor for the PS buffer.

The maximum down scaling factor is 1/2 such that the output image in either axis is 1/2 the size of the source. The maximum up scaling factor is 2^12 for either axis. The reciprocal of the scale factor should be loaded into this register. To reduce the PS buffer by a factor of two in the output frame buffer, a value of 10.0000_0000_0000 should be loaded into this register. To scale up by a factor of 4, the value of 1/4, or 00.0100_0000_0000, should be loaded into this register. To scale up by 8/5, the value of 00.1010_0000_0000 should be loaded.

EXAMPLE

```
PXP_PS_SCALE_WR(0x10001000);  // 1:1  scaling (0x1.000)
PXP_PS_SCALE_WR(0x08000800);  // 2x   scaling (0x0.800)
PXP_PS_SCALE_WR(0x20002000);  // 1/2x scaling (0x2.000)
```

Address: 20F_0000h base + 110h offset = 20F_0110h



PXP_PS_SCALE field descriptions

Field	Description
31 RSVD2	Reserved, always set to zero.
30–16 YSCALE	This is a two bit integer and 12 bit fractional representation (##.####_####_####) of the Y scaling factor for the PS source buffer. The maximum value programmed should be 2 since scaling down by a factor greater than 2 is not supported with the bilinear filter. Decimation and the bilinear filter should be used together to achieve scaling by more than a factor of 2.
15 RSVD1	Reserved, always set to zero.
XSCALE	This is a two bit integer and 12 bit fractional representation (##.####_####_####) of the X scaling factor for the PS source buffer. The maximum value programmed should be 2 since scaling down by a factor greater than 2 is not supported with the bilinear filter. Decimation and the bilinear filter should be used together to achieve scaling by more than a factor of 2.

53.7.19 PS Scale Offset Register (PXP_PS_OFFSET)

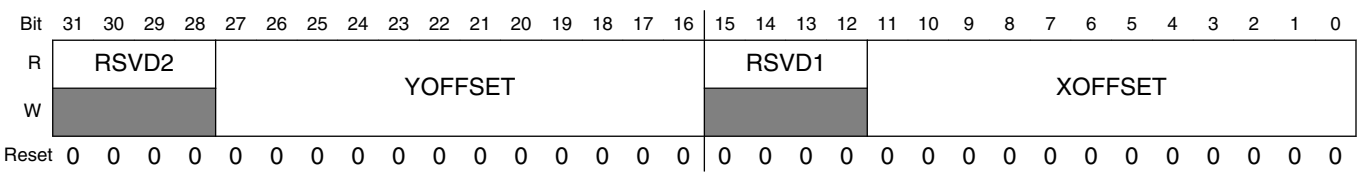
PS Scale Offset. This register provides the initial scale offset for the PS buffer.

The X and Y offset provides the ability to access the source image with a per sub-pixel granularity. This provides the capability to use all source pixels to effect the output PS image. The fixed offset values can be used for sub-pixel adjustments in the bilinear scaling filter. For example, when scaling an image down by a factor of 2, an initial offset of 0x0 would result in sub-sampling every other pixel. If a fixed offset of 0x800 (1/2), all pixels are used in scaling the final output pixel value. In this case, the first output pixel would be the sum of (1/2*P0) + (1/2*P1). This fixed offset is applied after the decimation filter stage, and before the bilinear filter stage.

EXAMPLE

```
PXP_PS_SCALE_WR(0x2000_2000);   // 1/2x scaling (0x2.000)
PXP_PS_OFFSET_WR(0x0800_0800);  // half-pixel offset in both X and Y to ensure averaging
versus pixel decimation
```

Address: 20F_0000h base + 120h offset = 20F_0120h



PXP_PS_OFFSET field descriptions

Field	Description
31–28 RSVD2	Reserved, always set to zero.
27–16 YOFFSET	This is a 12 bit fractional representation (0.####_####_####) of the Y scaling offset. This represents a fixed pixel offset which gets added to the scaled address to determine source data for the scaling engine.
15–12 RSVD1	Reserved, always set to zero.
XOFFSET	This is a 12 bit fractional representation (0.####_####_####) of the X scaling offset. This represents a fixed pixel offset which gets added to the scaled address to determine source data for the scaling engine.

53.7.20 PS Color Key Low (PXP_PS_CLRKEYLOW)

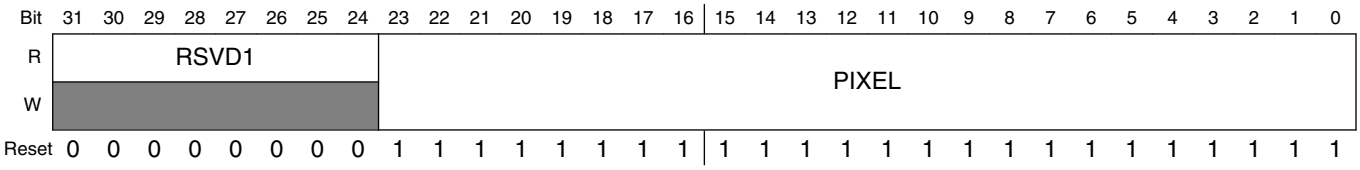
This register contains the color key low value for the PS buffer.

When processing an image, if the PXP finds a pixel in the PS buffer with a color that falls in the range between PXP_PS_CLRKEYLOW and PXP_PS_CLRKEYHIGH, it will insert the pixel from the AS channel. If the current AS pixel is letterboxed or if the AS also matches its colorkey range, the PXP_PS_BACKGROUND color is passed down the pixel pipeline.

EXAMPLE

```
                                        // colorkey values between
PXP_PS_CLRKEYLOW_WR (0x008000); // medium green and
PXP_PS_CLRKEYHIGH_WR(0x00FF00); // light green
```

Address: 20F_0000h base + 130h offset = 20F_0130h



PXP_PS_CLRKEYLOW field descriptions

Field	Description
31–24 RSVD1	Reserved, always set to zero.
PIXEL	Low range of color key applied to PS buffer. To disable PS colorkeying, set the low colorkey to 0xFFFFFF and the high colorkey to 0x000000.

53.7.21 PS Color Key High (PXP_PS_CLRKEYHIGH)

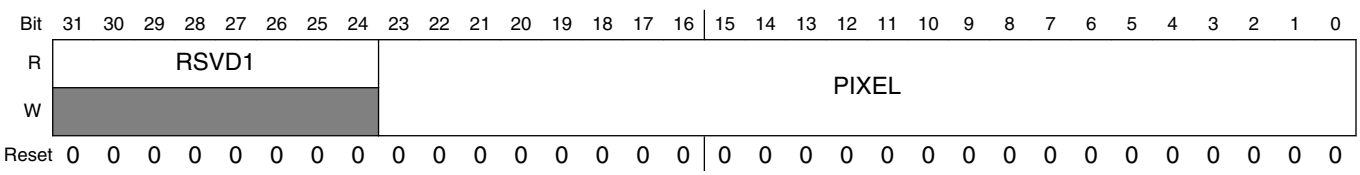
This register contains the color key high value for the PS buffer.

When processing an image, if the PXP finds a pixel in the PS buffer with a color that falls in the range between PXP_PS_CLRKEYLOW and PXP_PS_CLRKEYHIGH, it will insert the pixel from the AS channel. If the current AS pixel is letterboxed or if the AS also matches its colorkey range, the PXP_PS_BACKGROUND color is passed down the pixel pipeline.

EXAMPLE

```
                                        // colorkey values between
PXP_PS_CLRKEYLOW_WR (0x008000); // medium green and
PXP_PS_CLRKEYHIGH_WR(0x00FF00); // light green
```

Address: 20F_0000h base + 140h offset = 20F_0140h



PXP_PS_CLRKEYHIGH field descriptions

Field	Description
31–24 RSVD1	Reserved, always set to zero.
PIXEL	High range of color key applied to PS buffer. To disable PS colorkeying, set the low colorkey to 0xFFFFFF and the high colorkey to 0x000000.

53.7.22 Alpha Surface Control (PXP_AS_CTRL)

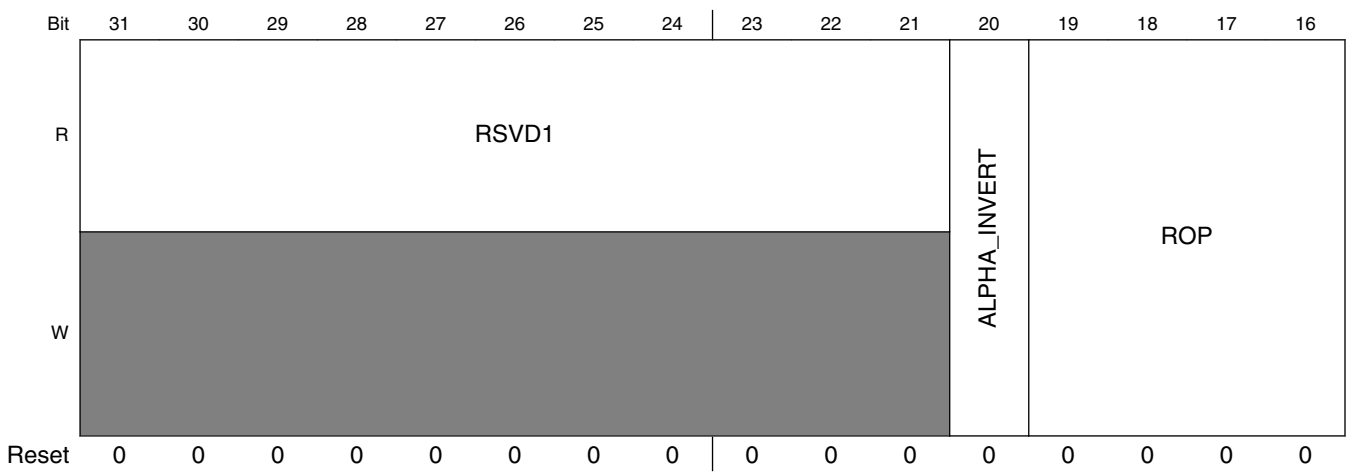
This register contains buffer control for the Alpha Surface 0 input buffer.

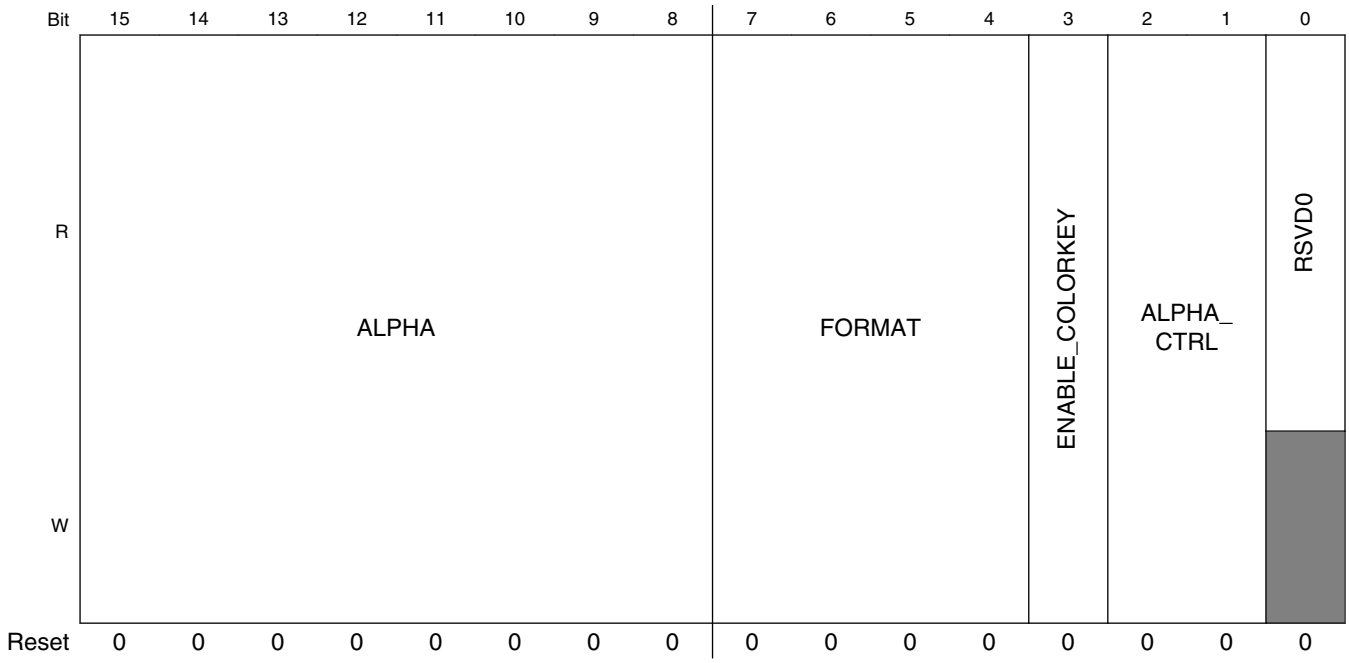
The Alpha Surface Parameter register provides additional controls for AS.

EXAMPLE

```
u32 asparam;
      asparam  = BF_PXP_ASPARAM_ENABLE    (1);
      asparam |= BF_PXP_ASPARAM_ALPHA_CTRL(BV_PXP_ASPARAM_ALPHA_CTRL__ROPs);
      asparam |= BF_PXP_ASPARAM_FORMAT    (BV_PXP_ASPARAM_FORMAT__ARGB8888);
      asparam |= BF_PXP_ASPARAM_ROP       (BV_PXP_ASPARAM_ROP__XORAS);
PXP_ASPARAM_WR(0,asparam);  // enable alpha surface to perform XOR ROP using RGB8888 AS
pixel format
```

Address: 20F_0000h base + 150h offset = 20F_0150h





PXP_AS_CTRL field descriptions

Field	Description
31–21 RSVD1	Reserved, always set to zero.
20 ALPHA_INVERT	Setting this bit to logic 0 will not alter the alpha value. A logic 1 will invert the alpha value and apply (1-alpha) for image composition.
19–16 ROP	Indicates a raster operation to perform when enabled. Raster operations are enabled through the ALPHA_CTRL field. 0x0 **MASKAS** — AS AND PS 0x1 **MASKNOTAS** — nAS AND PS 0x2 **MASKASNOT** — AS AND nPS 0x3 **MERGEAS** — AS OR PS 0x4 **MERGENOTAS** — nAS OR PS 0x5 **MERGEASNOT** — AS OR nPS 0x6 **NOTCOPYAS** — nAS 0x7 **NOT** — nPS 0x8 **NOTMASKAS** — AS NAND PS 0x9 **NOTMERGEAS** — AS NOR PS 0xA **XORAS** — AS XOR PS 0xB **NOTXORAS** — AS XNOR PS
15–8 ALPHA	Alpha modifier used when the ALPHA_MULTIPLY or ALPHA_OVERRIDE values are programmed in PXP_AS_CTRL[ALPHA_CTRL]. The output alpha value will either be replaced (ALPHA_OVERRIDE) or scaled (ALPHA_MULTIPLY) when selected.
7–4 FORMAT	Indicates the input buffer format for AS. 0x0 **ARGB8888** — 32-bit pixels with alpha 0x1 **RGBA888** — 2-bit pixel with alpha at low 8 bits 0x4 **RGB888** — 32-bit pixels without alpha (unpacked 24-bit format)

Table continues on the next page...

PXP_AS_CTRL field descriptions (continued)

Field	Description
	0x8 **ARGB1555** — 16-bit pixels with alpha 0x9 **ARGB4444** — 16-bit pixels with alpha 0xA **RGBA5551** — 16-bit pixel with alpha at low 1 bit 0xB **RGBA4444** — 16-bit pixel with alpha at low 4 bits 0xC **RGB555** — 16-bit pixels without alpha 0xD **RGB444** — 16-bit pixels without alpha 0xE **RGB565** — 16-bit pixels without alpha
3 ENABLE_ COLORKEY	Indicates that colorkey functionality is enabled for this alpha surface. Pixels found in the alpha surface colorkey range will be displayed as transparent (the PS pixel will be used).
2–1 ALPHA_CTRL	Determines how the alpha value is constructed for this alpha surface. Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. 0x0 **Embedded** — Indicates that the AS pixel alpha value will be used to blend the AS with PS. The ALPHA field is ignored. 0x1 **Override** — Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. 0x2 **Multiply** — Indicates that the value in the ALPHA field should be used to scale all pixel alpha values. Each pixel alpha is multiplied by the value in the ALPHA field. 0x3 **ROPs** — Enable ROPs. The ROP field indicates an operation to be performed on the alpha surface and PS pixels.
0 RSVD0	Reserved, always set to zero.

53.7.23 Alpha Surface Buffer Pointer (PXP_AS_BUF)

Alpha Surface 0 Buffer Address Pointer. This register points to the beginning of the Alpha Surface 0 input buffer.

This register is used to indicate the base address of the AS buffer.

EXAMPLE

```
u32* alpha_ptr;
PXP_ASn_WR(0,alpha_ptr);
```

Address: 20F_0000h base + 160h offset = 20F_0160h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_AS_BUF field descriptions

Field	Description
ADDR	Address pointer for the alpha surface 0 buffer.

53.7.24 Alpha Surface Pitch (PXP_AS_PITCH)

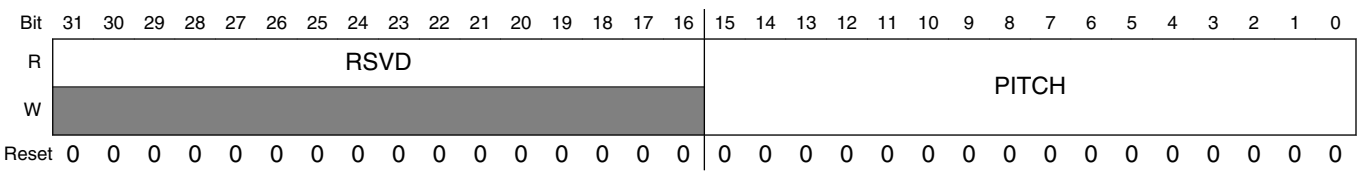
This register contains the alpha surface pitch in bytes.

Any byte value will indicate the vertical pitch. This value will be used in AS pixel address calculations. This value has no relation to the UL and LR registers. It specifies how many bytes are between two vertically adjacent pixels in the input AS surface.

EXAMPLE

```
PXP_AS_PITCH_WR( 1920 * 4 );  // The output buffer pitch is HD resolution at 32 bits per
pixel
```

Address: 20F_0000h base + 170h offset = 20F_0170h



PXP_AS_PITCH field descriptions

Field	Description
31–16 RSVD	Reserved, always set to zero.
PITCH	Indicates the number of bytes in memory between two vertically adjacent pixels.

53.7.25 Overlay Color Key Low (PXP_AS_CLRKEYLOW)

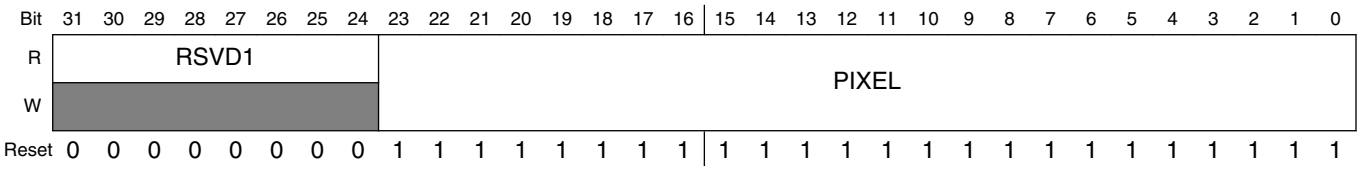
This register contains the color key low value for the AS buffer.

When processing an image, the if the PXP finds a pixel in the current overlay image with a color that falls in the range from the ASCOLORKEYLOW to ASCOLORKEYHIGH range, it will use the PS pixel value for that location. If no PS image is present or if the PS image also matches its colorkey range, the PS background color is used. Colorkey operations are higher priority than alpha or ROP operations.

EXAMPLE

```
                                  // colorkey values between
PXP_AS_CLRKEYLOW_WR (0x000000); // black and
PXP_AS_CLRKEYHIGH_WR(0x800000); // medium red
```

Address: 20F_0000h base + 180h offset = 20F_0180h



PXP_AS_CLRKEYLOW field descriptions

Field	Description
31–24 RSVD1	Reserved, always set to zero.
PIXEL	Low range of RGB color key applied to AS buffer. Each overlay has an independent colorkey enable.

53.7.26 Overlay Color Key High (PXP_AS_CLRKEYHIGH)

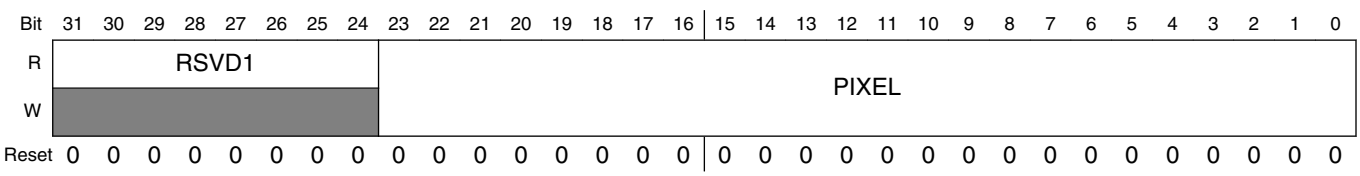
This register contains the color key high value for the AS buffer.

When processing an image, the if the PXP finds a pixel in the current overlay image with a color that falls in the range from the ASCOLORKEYLOW to ASCOLORKEYHIGH range, it will use the PS pixel value for that location. If no PS image is present or if the PS image also matches its colorkey range, the PS background color is used. Colorkey operations are higher priority than alpha or ROP operations.

EXAMPLE

```
                                    // colorkey values between
PXP_AS_CLRKEYLOW_WR (0x000000); // black and
PXP_AS_CLRKEYHIGH_WR(0x800000); // medium red
```

Address: 20F_0000h base + 190h offset = 20F_0190h



PXP_AS_CLRKEYHIGH field descriptions

Field	Description
31–24 RSVD1	Reserved, always set to zero.
PIXEL	High range of RGB color key applied to AS buffer. Each overlay has an independent colorkey enable.

53.7.27 Color Space Conversion Coefficient Register 0 (PXP_CSC1_COEF0)

This register contains color space conversion coefficients in two's compliment notation.

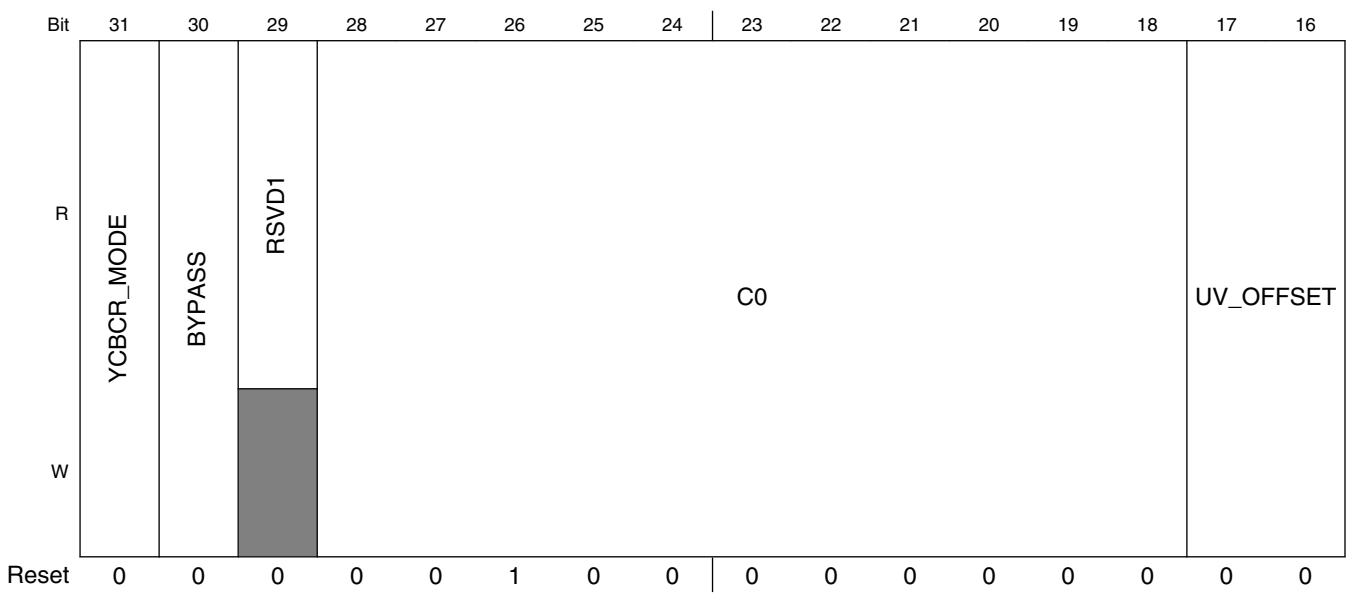
The coefficient 0 register contains coefficients used in the color space conversion algorithm. The Y and UV offsets are added to the source buffer to normalize them before the conversion. C0 is the coefficient that is used to multiply the luma component of the data for all three RGB components.

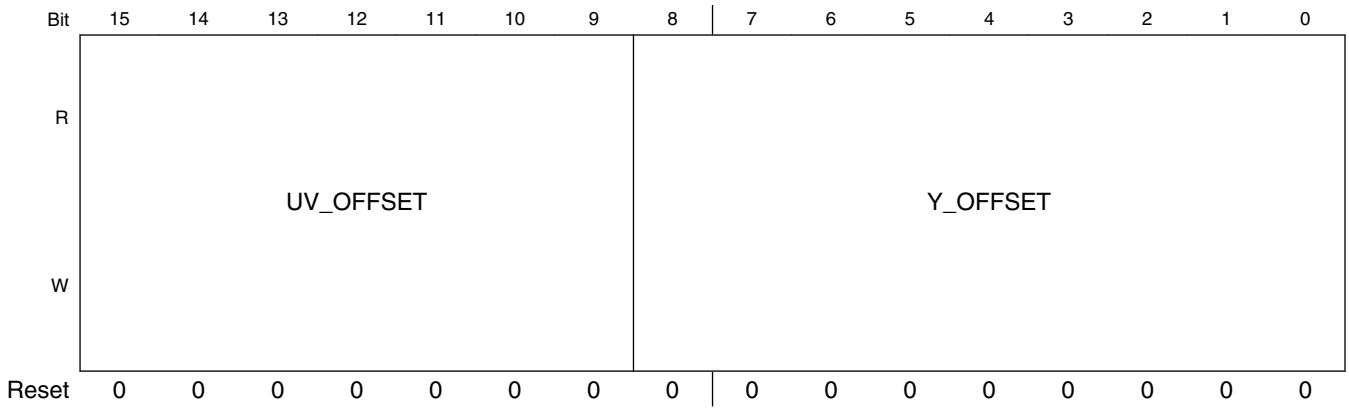
EXAMPLE

```
//  The equations used for Colorspace conversion are:
//     R = C0*(Y+YOFFSET)                    + C1(V+UV_OFFSET)
//     G = C0*(Y+YOFFSET) + C3(U+UV_OFFSET) + C2(V+UV_OFFSET)
//     R = C0*(Y+YOFFSET) + C4(U+UV_OFFSET)

   PXP_CSCCOEF0_WR(0x04030000); // YUV coefficients: C0, Yoffset, UVoffset
   PXP_CSCCOEF1_WR(0x01230208); // YUV coefficients: C1, C4
   PXP_CSCCOEF2_WR(0x076B079b); // YUV coefficients: C2, C3
```

Address: 20F_0000h base + 1A0h offset = 20F_01A0h





PXP_CSC1_COEF0 field descriptions

Field	Description
31 YCBCR_MODE	Set to 1 when performing YCbCr conversion to RGB. Set to 0 when converting RGB to YUV data. This bit changes the behavior of the scaler when performing U/V scaling.
30 BYPASS	Bypass the CSC unit in the scaling engine. When set to logic 1, bypass is enabled and the output pixels will be in the YUV/YCbCr color space. When set to logic 0, the CSC unit is enabled and the pixels will be converted based on the programmed coefficients.
29 RSVD1	Reserved, always set to zero.
28–18 C0	Two's compliment Y multiplier coefficient. YUV=0x100 (1.000) YCbCr=0x12A (1.164)
17–9 UV_OFFSET	Two's compliment phase offset implicit for CbCr data. Generally used for YCbCr to RGB conversion. YCbCr=0x180, YUV=0x000 (typically -128 or 0x180 to indicate normalized -0.5 to 0.5 range)
Y_OFFSET	Two's compliment amplitude offset implicit in the Y data. For YUV, this is typically 0 and for YCbCr, this is typically -16 (0x1F0)

53.7.28 Color Space Conversion Coefficient Register 1 (PXP_CSC1_COEF1)

This register contains color space conversion coefficients in two's compliment notation.

The Coeffient 1 register contains coefficients used in the color space conversion algorithm. C1 is the coefficient that is used to multiply the chroma (Cr/V) component of the data for the red component. C4 is the coefficient that is used to multiply the chroma (Cb/U) component of the data for the blue component. Both values should be coded as a two's compliment fixed point number with 8 bits right of the decimal.

EXAMPLE

```
PXP_CSCCOEF0_WR(0x04030000); // YUV coefficients: C0, Yoffset, UVoffset
PXP_CSCCOEF1_WR(0x01230208); // YUV coefficients: C1, C4
PXP_CSCCOEF2_WR(0x076B079b); // YUV coefficients: C2, C3
```

Address: 20F_0000h base + 1B0h offset = 20F_01B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD1					C1											RSVD0					C4										
W																																
Reset	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0

PXP_CSC1_COEF1 field descriptions

Field	Description
31–27 RSVD1	Reserved, always set to zero.
26–16 C1	Two's compliment Red V/Cr multiplier coefficient. YUV=0x123 (1.140) YCbCr=0x198 (1.596)
15–11 RSVD0	Reserved, always set to zero.
C4	Two's compliment Blue U/Cb multiplier coefficient. YUV=0x208 (2.032) YCbCr=0x204 (2.017)

53.7.29 Color Space Conversion Coefficient Register 2 (PXP_CSC1_COEF2)

This register contains color space conversion coefficients in two's compliment notation.

The Coefficient 2 register contains coefficients used in the color space conversion algorithm. C2 is the coefficient that is used to multiply the chroma (Cr/V) component of the data for the green component. C3 is the coefficient that is used to multiply the chroma (Cb/U) component of the data for the green component. Both values should be coded as a two's compliment fixed point number with 8 bits right of the decimal.

EXAMPLE

```
// NOTE: The default values for the CSCCOEF2 register are incorrect.  C2 should be 0x76B and
C3 should be 0x79C for proper operation.

   PXP_CSCCOEF0_WR(0x04030000); // YUV coefficients: C0, Yoffset, UVoffset
   PXP_CSCCOEF1_WR(0x01230208); // YUV coefficients: C1, C4
   PXP_CSCCOEF2_WR(0x076B079b); // YUV coefficients: C2, C3
```

Address: 20F_0000h base + 1C0h offset = 20F_01C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD1					C2											RSVD0					C3										
W																																
Reset	0	0	0	0	0	1	1	1	1	0	0	1	1	0	1	1	0	0	0	0	0	1	1	1	0	1	1	0	1	1	0	0

PXP_CSC1_COEF2 field descriptions

Field	Description
31–27 RSVD1	Reserved, always set to zero.
26–16 C2	Two's complement Green V/Cr multiplier coefficient. YUV=0x76B (-0.581) YCbCr=0x730 (-0.813)
15–11 RSVD0	Reserved, always set to zero.
C3	Two's complement Green U/Cb multiplier coefficient. YUV=0x79C (-0.394) YCbCr=0x79C (-0.392)

53.7.30 Color Space Conversion Control Register (PXP_CSC2_CTRL)

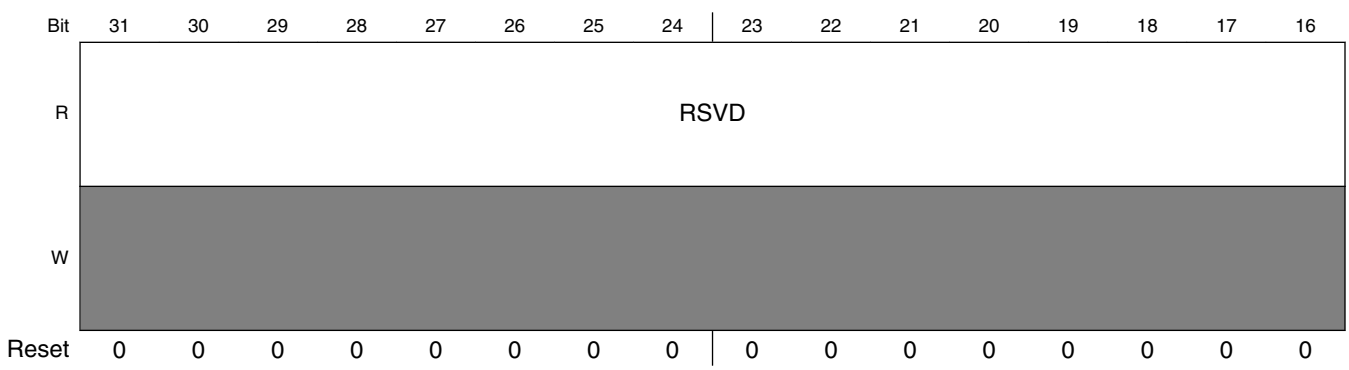
This register contains the control registers to configure the CSC module.

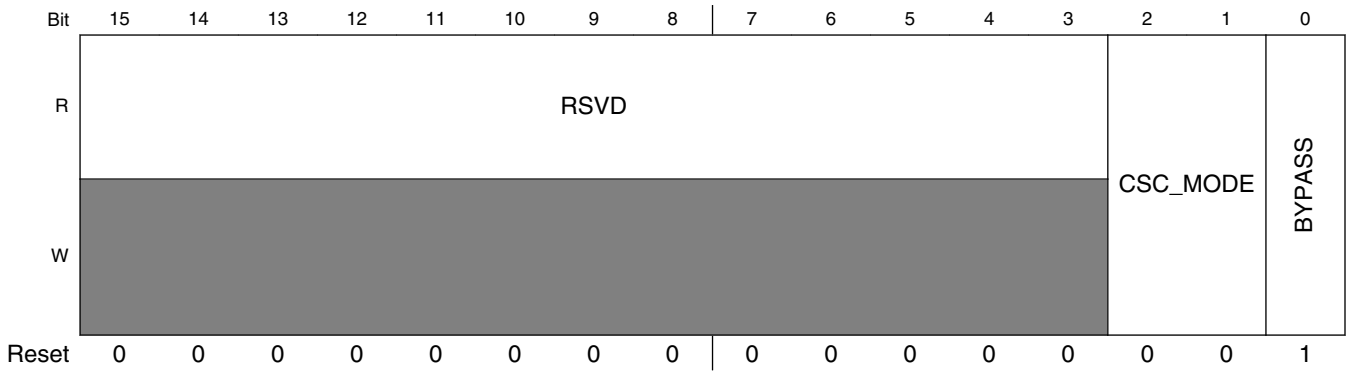
The CSC control register will configure the CSC module to perform color space conversion between the RGB/YUV/YCbCr color spaces.

EXAMPLE

```
//Converting from YUV/YCbCr color spaces to the RGB color space uses the
//following equation structure:
//
// R = A1(Y-D1) + A2(U-D2) + A3(V-D3)
// G = B1(Y-D1) + B2(U-D2) + B3(V-D3)
// B = C1(Y-D1) + C2(U-D2) + C3(V-D3)
//
//Converting from the RGB color space to YUV/YCbCr color spaces uses the
//following equation structure:
//
// Y = A1*R + A2*G + A3*B + D1
// U = B1*R + B2*G + B3*B + D2
// V = C1*R + C2*G + C3*B + D3
//
//All math is signed, so all coefficients come in as two's comp numbers
//
```

Address: 20F_0000h base + 1D0h offset = 20F_01D0h





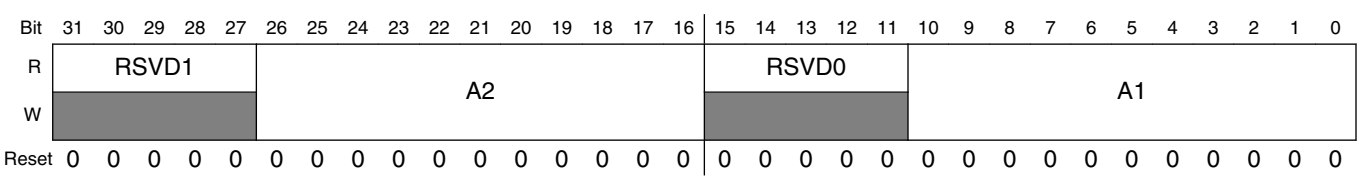
PXP_CSC2_CTRL field descriptions

Field	Description
31–3 RSVD	Reserved, always set to zero.
2–1 CSC_MODE	This field controls how the CSC unit operates on pixels when the CSC is not bypassed. 0x0 **YUV2RGB** — Convert from YUV to RGB. 0x1 **YCbCr2RGB** — Convert from YCbCr to RGB. 0x2 **RGB2YUV** — Convert from RGB to YUV. 0x3 **RGB2YCbCr** — Convert from RGB to YCbCr.
0 BYPASS	This bit controls whether the pixels entering the CSC2 unit get converted or not. When BYPASS is set, no operations occur on the pixels. When BYPASS is cleared, the selected CSC operation takes place.

53.7.31 Color Space Conversion Coefficient Register 0 (PXP_CSC2_COEF0)

This register contains color space conversion coefficients in two's complement notation.

Address: 20F_0000h base + 1E0h offset = 20F_01E0h



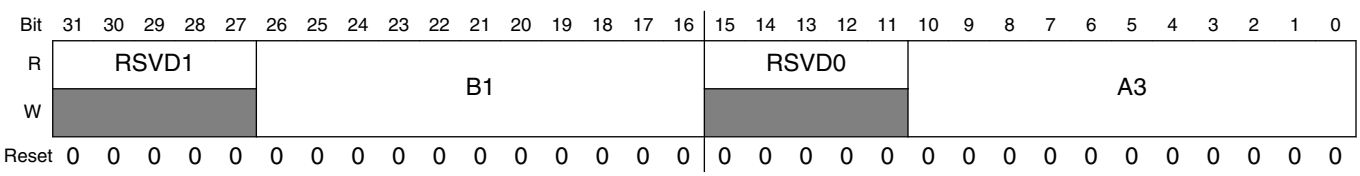
PXP_CSC2_COEF0 field descriptions

Field	Description
31–27 RSVD1	Reserved, always set to zero.
26–16 A2	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15–11 RSVD0	Reserved, always set to zero.
A1	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

53.7.32 Color Space Conversion Coefficient Register 1 (PXP_CSC2_COEF1)

This register contains color space conversion coefficients in two's complement notation.

Address: 20F_0000h base + 1F0h offset = 20F_01F0h



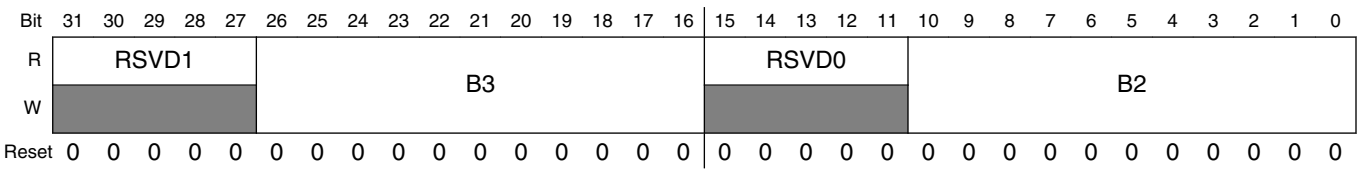
PXP_CSC2_COEF1 field descriptions

Field	Description
31–27 RSVD1	Reserved, always set to zero.
26–16 B1	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15–11 RSVD0	Reserved, always set to zero.
A3	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

53.7.33 Color Space Conversion Coefficient Register 2 (PXP_CSC2_COEF2)

This register contains color space conversion coefficients in two's complement notation.

Address: 20F_0000h base + 200h offset = 20F_0200h



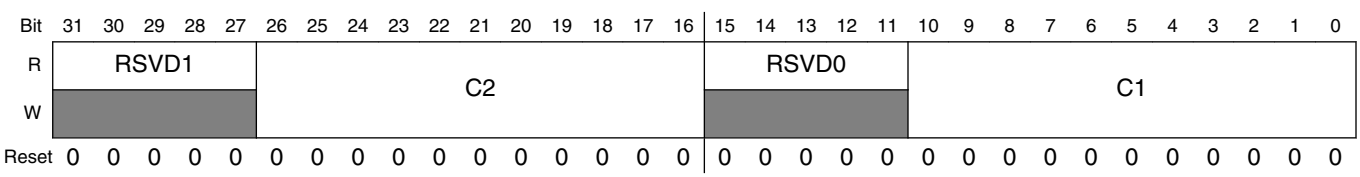
PXP_CSC2_COEF2 field descriptions

Field	Description
31–27 RSVD1	Reserved, always set to zero.
26–16 B3	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15–11 RSVD0	Reserved, always set to zero.
B2	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

53.7.34 Color Space Conversion Coefficient Register 3 (PXP_CSC2_COEF3)

This register contains color space conversion coefficients in two's complement notation.

Address: 20F_0000h base + 210h offset = 20F_0210h



PXP_CSC2_COEF3 field descriptions

Field	Description
31–27 RSVD1	Reserved, always set to zero.
26–16 C2	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

Table continues on the next page...

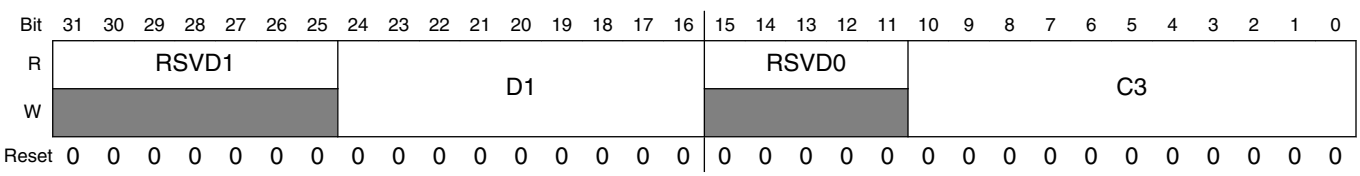
PXP_CSC2_COEF3 field descriptions (continued)

Field	Description
15–11 RSVD0	Reserved, always set to zero.
C1	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

53.7.35 Color Space Conversion Coefficient Register 4 (PXP_CSC2_COEF4)

This register contains color space conversion coefficients in two's complement notation.

Address: 20F_0000h base + 220h offset = 20F_0220h



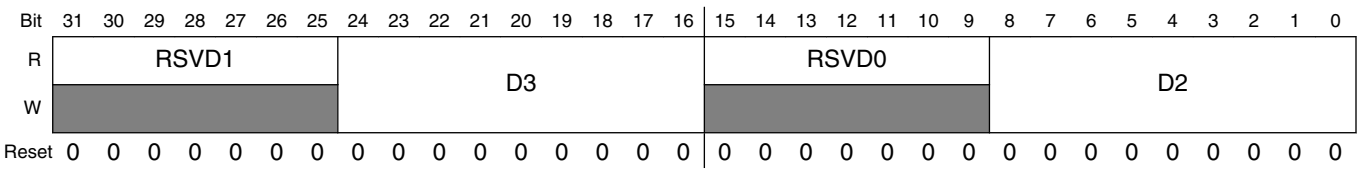
PXP_CSC2_COEF4 field descriptions

Field	Description
31–25 RSVD1	Reserved, always set to zero.
24–16 D1	Two's complement coefficient integer offset to be added.
15–11 RSVD0	Reserved, always set to zero.
C3	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

53.7.36 Color Space Conversion Coefficient Register 5 (PXP_CSC2_COEF5)

This register contains color space conversion coefficients in two's complement notation.

Address: 20F_0000h base + 230h offset = 20F_0230h



PXP_CSC2_COEF5 field descriptions

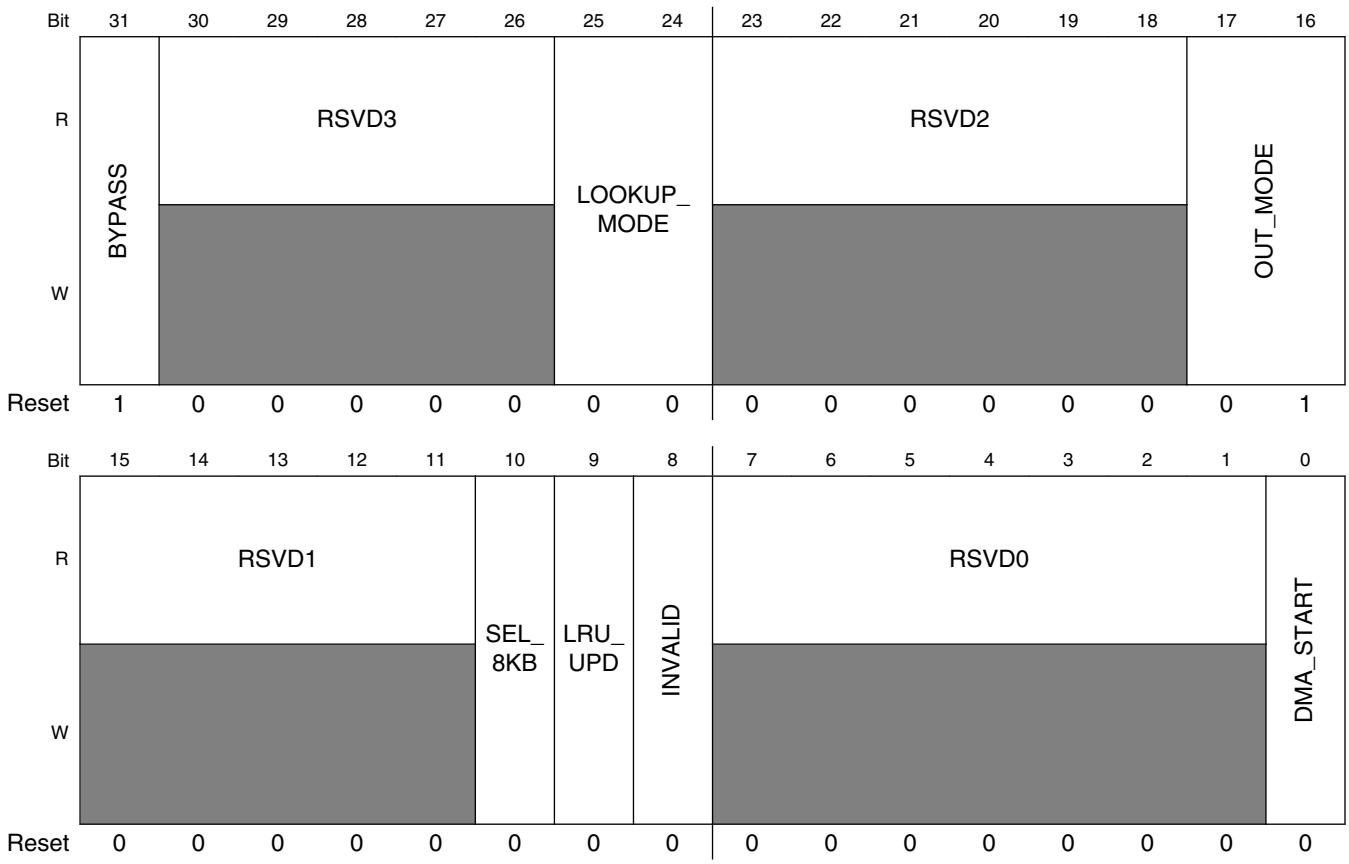
Field	Description
31–25 RSVD1	Reserved, always set to zero.
24–16 D3	Two's complement coefficient integer offset to be added.
15–9 RSVD0	Reserved, always set to zero.
D2	Two's complement D1 coefficient integer offset to be added.

53.7.37 Lookup Table Control Register (PXP_LUT_CTRL)

This register is used to access/control the Monochrome Lookup table.

The Y8 LUT input mode will take the high order data path byte and transform it using the LUT memory. This is an 8-bit to 8-bit transformation. The two low order bytes bypass the LUT and are not transformed, but bypassed without modification. This option can be used for monochrome gamma correction. The Direct Lookup mode will use the high nibble of each data byte and truncate the low nibble to generate the lookup address, i.e. R[7:0]G[7:0]B[7:0] -> R[7:4]G[7:4]B[7:4]. 4K pixels (12-bit address) with 2 bytes per pixel is supported in this mode. Cached Lookup mode will use the high order bits, R[7:3],G[7:2],B[7:3] or RGB565, to address the cached LUT memory. 64KB LUT tables, using 16KB of internal LUT memory, can be indirectly transformed to 16-bit output pixels (as in RGBW4444/RGB565). This is used for 16bpp gamma correction. Cache misses are internally managed by the PXP LUT Cache controller.

Address: 20F_0000h base + 240h offset = 20F_0240h



PXP_LUT_CTRL field descriptions

Field	Description
31 BYPASS	Setting this bit will bypass the LUT memory resource completely. No pixel transfermations will occur at this stage of the PXP pixel processing pipeline.
30–26 RSVD3	Reserved, always set to zero.
25–24 LOOKUP_MODE	Configure the input address for the 16KB LUT memory. The address into the LUT uses different parts of the pixel data path bytes. The data path is defined as three bytes, conceptually as RGB/YUV/YCbCr[23:0]. Also referred to as R/Y[7:0],G/U[7:0],B/V[7:0] 0x0 **CACHE_RGB565** — LUT ADDR = R[7:3],G[7:2],B[7:3]. Use all 16KB of LUT for indirect cached 128KB lookup. 0x1 **DIRECT_Y8** — LUT ADDR = 16'b0,Y[7:0]. Use only the first 256 bytes of LUT. Only the Y, or third data path byte, is tranformed. 0x2 **DIRECT_RGB444** — LUT ADDR = R[7:4],G[7:4],B[7:4]. Use one 8KB bank of LUT selected by SEL_8KB. 0x3 **DIRECT_RGB454** — LUT ADDR = R[7:4],G[7:3],B[7:4]. Use all 16KB of LUT.
23–18 RSVD2	Reserved, always set to zero.

Table continues on the next page...

PXP_LUT_CTRL field descriptions (continued)

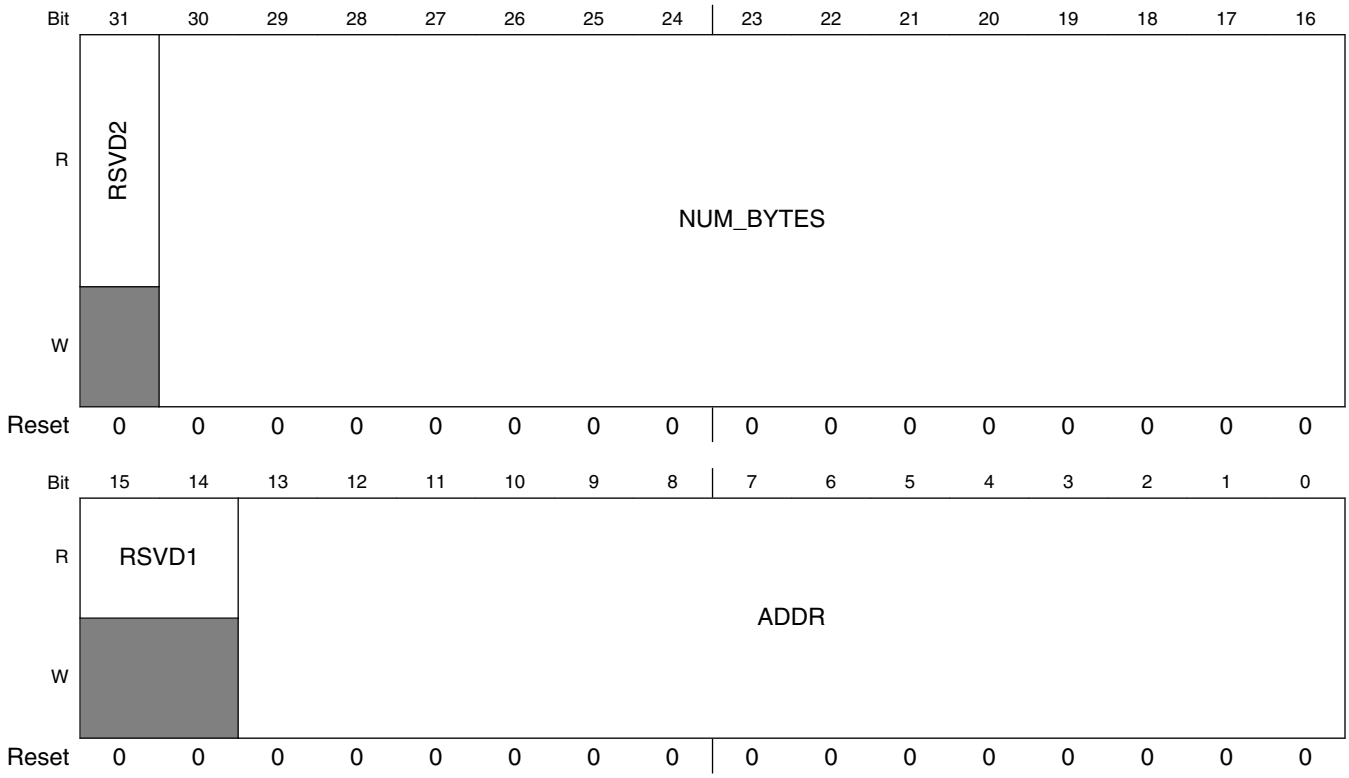
Field	Description
17–16 OUT_MODE	Select the output mode of operation for the LUT resource. There are four bytes [3-0] in the data path at the output of the LUT resource. Byte lane 3 is always bypassed and usually contains an alpha value. The LUT can be programmed to transform bytes 2,1,0 according to the options available in this field. 0x0 **RESERVED** — Reserved, not valid when using the LUT to transform pixels. 0x1 **Y8** — R/Y byte lane 2 lookup, bytes 1,0 bypassed. 0x2 **RGBW4444CFA** — Byte lane 2 = CFA_Y8, byte lane 1,0 = RGBW4444. 0x3 **RGB888** — RGB565->RGB888 conversion for Gamma correction.
15–11 RSVD1	Reserved, always set to zero.
10 SEL_8KB	Selects which 8KB bank of memory to use for direct 12bpp lookup modes. Logic 0 indicates first 8KB, logic 1 indicates second 8KB. Two direct LUT arrays can be stored and one can be selected for a given PXP operation.
9 LRU_UPD	Least Recently Used Policy Update Control: 1=> block LRU update for hit after miss. 0=> update LRU for all hits including hit after miss.
8 INVALID	Invalidate the cache LRU and valid bits. This bit will automatically reset when set to a logic 1.
7–1 RSVD0	Reserved, always set to zero.
0 DMA_START	Setting this bit will result in the DMA operation to load the PXP LUT memory based on PXP_LUT_ADDR_NUM_BYTES, PXP_LUT_ADDR_ADDR, and PXP_LUT_MEM_ADDR. This bit will automatically reset when set to a logic 1. Note: The LOOKUP_MODE must not be set to CACHE_RGB565 when starting and performing DMA transfers.

53.7.38 Lookup Table Control Register (PXP_LUT_ADDR)

This register is used to access/control the Monochrome Lookup table.

The Y8 LUT input mode will take the high order data path byte and transform it using the LUT memory. This is an 8-bit to 8-bit transformation. The two low order bytes bypass the LUT and are not transformed, but bypassed without modification. This option can be used for monochrome gamma correction. The Direct Lookup mode will use the high nibble of each data byte and truncate the low nibble to generate the lookup address, i.e. R[7:0]G[7:0]B[7:0] -> R[7:4]G[7:4]B[7:4]. 4K pixels (12-bit address) with 2 bytes per pixel is supported in this mode. Cached Lookup mode will use the high order bits, R[7:3],G[7:2],B[7:3] or RGB565, to address the cached LUT memory. 64KB LUT tables, using 16KB of internal LUT memory, can be indirectly transformed to 16-bit output pixels (as in RGBW4444/RGB565). This is used for 16bpp gamma correction. Cache misses are internally managed by the PXP LUT Cache controller.

Address: 20F_0000h base + 250h offset = 20F_0250h



PXP_LUT_ADDR field descriptions

Field	Description
31 RSVD2	Reserved, always set to zero.
30–16 NUM_BYTES	Indicates the number of bytes to load via a DMA operation. This field must be divisable by 8 and the least significant 3 bits must be 0. The value 8 indicates load 8 bytes from the external address indicated by PXP_LUT_MEM_ADDR to the LUT memory location indicated by PXP_LUT_CTRL_ADDR.
15–14 RSVD1	Reserved, always set to zero.
ADDR	LUT indexed address pointer. This address into the LUT memory is always four byte aligned for PIO access, and eight byte aligned for DMA access. The least two significant bits are not used to drive the LUT memory array. For PIO LUT access, when the LUT data register is written, the contents of the LUT at the address specified by this address field will be loaded with a 32-bit DWORD. This address pointer will be incremented after the LUT data is written. This will provide recursive writes to the LUT data register to initialize the entire LUT array with recursive writes to the LUT data register. For DMA access, this register indicates the LUT memory address of the 8 byte QWORD to be loaded. When using the NUM_BYTES field to load more than 8 bytes, the register should be programmed with the first LUT memory location to be filled and each load of the LUT memory will increment this address field until NUM_BYTES has been loaded.

53.7.39 Lookup Table Data Register (PXP_LUT_DATA)

This register is used to load data into the lookup table.

Address: 20F_0000h base + 260h offset = 20F_0260h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_LUT_DATA field descriptions

Field	Description
DATA	Writing this field will load 4 bytes, aligned to four byte boundaries, of data indexed by the ADDR field of the PXP_LUT_CTRL register.

53.7.40 Lookup Table External Memory Address Register (PXP_LUT_EXTMEM)

For DMA LUT memory loads, this is the base address from which data will be sourced to store into the LUT memory array. For Cached LUT memory pixel transformations, this register will store the base address of the full 64K pixel LUT translation table.

Address: 20F_0000h base + 270h offset = 20F_0270h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_LUT_EXTMEM field descriptions

Field	Description
ADDR	This register contains the external memory address used for LUT memory operation. For DMA LUT memory loads, this is the base address from which data will be sourced to store into the LUT memory array. For Cached LUT memory pixel transformations, this register will store the base address of the full 64K pixel LUT translation table.

53.7.41 Color Filter Array Register (PXP_CFA)

There are sixteen 2 bit values in this register each mapping a selected component to the output pixel. The two bit values are defined as 0=>R, 1=>G, 2=>B, and 3=>W. The first byte represents the repetitive pattern of RGBW pixels in the CFA for the first line segment of each processed PXP block. The second byte represents the pattern in the second line segment of the block, and so on. The first byte repeats two times for 8x8 macro block mode, and repeats four times for 16x16 block mode.

Address: 20F_0000h base + 280h offset = 20F_0280h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_CFA field descriptions

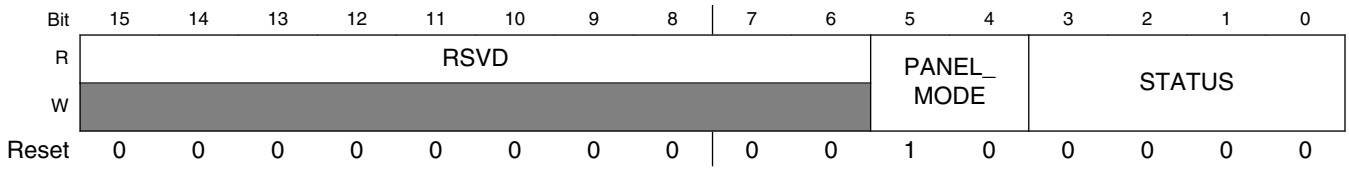
Field	Description
DATA	This register contains the Color Filter Array pattern for decimation of RGBW4444 16 bit pixels to individual R, G, B, W values. The pattern represents a replicated 4x4 color filter array for the entire output frame buffer.

53.7.42 Histogram Control Register (PXP_HIST_CTRL)

Provides control and status registers for the PXP's histogram classification algorithm.

Address: 20F_0000h base + 290h offset = 20F_0290h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVD															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



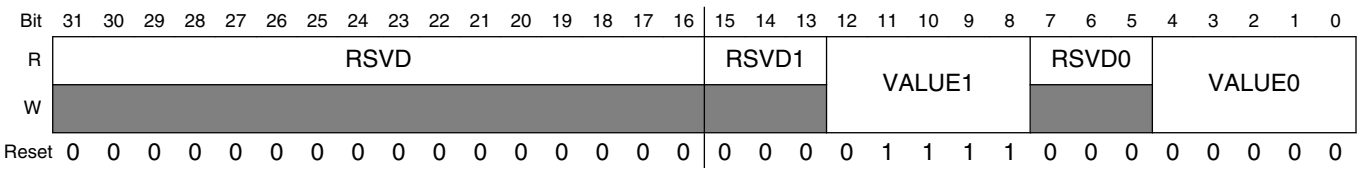
PXP_HIST_CTRL field descriptions

Field	Description
31–6 RSVD	Reserved, always set to zero.
5–4 PANEL_MODE	Specifies the EPDC panel grayscale depth. This value is used to specify the number of bits used in comparisons when matching pixels to histogram bins. All comparator values MUST be programmed such that their bit width is consistent with the value of this register field. For instance, if GRAY16 is selected, comparator values must be in the range of 0x0-0xF. 0x0 **GRAY4** — 4-bit grayscale 0x1 **GRAY8** — 8-bit grayscale 0x2 **GRAY16** — 16-bit grayscale 0x3 **GRAY32** — 32-bit grayscale
STATUS	Indicates which histogram matched the processed bitmap. Bit[0] indicates that the bitmap pixels were fully contained within the HIST2 (black / white) histogram. Bit[1] indicates that the bitmap pixels were fully contained within the HIST4 (2-bit grayscale) histogram. Bit[2] indicates that the bitmap pixels were fully contained within the HIST8 (3-bit grayscale) histogram. Bit[3] indicates that the bitmap pixels were fully contained within the HIST16 (4-bit grayscale) histogram.

53.7.43 2-level Histogram Parameter Register (PXP_HIST2_PARAM)

This register specifies the valid values for a 2-level histogram. If all pixels in a bitmap match the 2-level histogram values, STATUS[0] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the PANEL_MODE control field.

Address: 20F_0000h base + 2A0h offset = 20F_02A0h



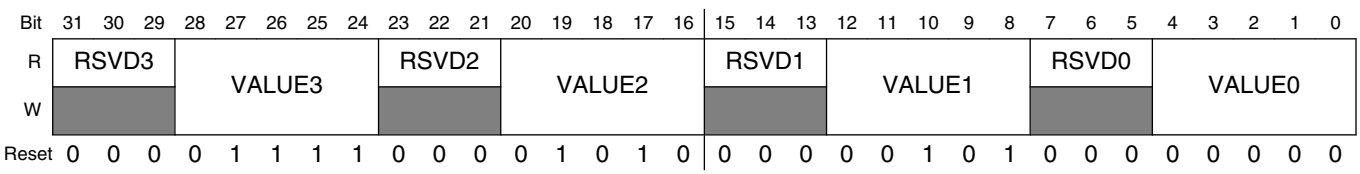
PXP_HIST2_PARAM field descriptions

Field	Description
31–16 RSVD	Reserved, always set to zero.
15–13 RSVD1	Reserved, always set to zero.
12–8 VALUE1	White value for 2-level histogram
7–5 RSVD0	Reserved, always set to zero.
VALUE0	Black value for 2-level histogram

53.7.44 4-level Histogram Parameter Register (PXP_HIST4_PARAM)

This register specifies the valid values for a 4-level histogram. If all pixels in a bitmap match the 4-level histogram values, STATUS[1] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the PANEL_MODE control field.

Address: 20F_0000h base + 2B0h offset = 20F_02B0h



PXP_HIST4_PARAM field descriptions

Field	Description
31–29 RSVD3	Reserved, always set to zero.

Table continues on the next page...

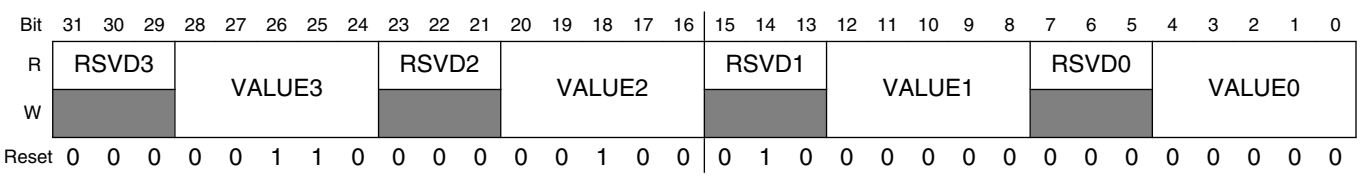
PXP_HIST4_PARAM field descriptions (continued)

Field	Description
28–24 VALUE3	GRAY3 (White) value for 4-level histogram
23–21 RSVD2	Reserved, always set to zero.
20–16 VALUE2	GRAY2 value for 4-level histogram
15–13 RSVD1	Reserved, always set to zero.
12–8 VALUE1	GRAY1 value for 4-level histogram
7–5 RSVD0	Reserved, always set to zero.
VALUE0	GRAY0 (Black) value for 4-level histogram

53.7.45 8-level Histogram Parameter 0 Register (PXP_HIST8_PARAM0)

This register specifies four of the valid values for an 8-level histogram. If all pixels in a bitmap match the 8-level histogram values, STATUS[2] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the PANEL_MODE control field.

Address: 20F_0000h base + 2C0h offset = 20F_02C0h



PXP_HIST8_PARAM0 field descriptions

Field	Description
31–29 RSVD3	Reserved, always set to zero.
28–24 VALUE3	GRAY3 value for 8-level histogram
23–21 RSVD2	Reserved, always set to zero.

Table continues on the next page...

PXP_HIST8_PARAM0 field descriptions (continued)

Field	Description
20–16 VALUE2	GRAY2 value for 8-level histogram
15–13 RSVD1	Reserved, always set to zero.
12–8 VALUE1	GRAY1 value for 8-level histogram
7–5 RSVD0	Reserved, always set to zero.
VALUE0	GRAY0 (Black) value for 8-level histogram

53.7.46 8-level Histogram Parameter 1 Register (PXP_HIST8_PARAM1)

This register specifies four of the valid values for an 8-level histogram. If all pixels in a bitmap match the 8-level histogram values, STATUS[2] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the PANEL_MODE control field.

Address: 20F_0000h base + 2D0h offset = 20F_02D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD7			VALUE7					RSVD6			VALUE6					RSVD5			VALUE5					RSVD4				VALUE4			
W																																
Reset	0	0	0	0	1	1	1	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0	1	1	0	0	0	0	1	0	0	1

PXP_HIST8_PARAM1 field descriptions

Field	Description
31–29 RSVD7	Reserved, always set to zero.
28–24 VALUE7	GRAY7 (White) value for 8-level histogram
23–21 RSVD6	Reserved, always set to zero.
20–16 VALUE6	GRAY6 value for 8-level histogram
15–13 RSVD5	Reserved, always set to zero.

Table continues on the next page...

PXP_HIST8_PARAM1 field descriptions (continued)

Field	Description
12–8 VALUE5	GRAY5 value for 8-level histogram
7–5 RSVD4	Reserved, always set to zero.
VALUE4	GRAY4 value for 8-level histogram

53.7.47 16-level Histogram Parameter 0 Register (PXP_HIST16_PARAM0)

This register specifies four of the valid values for a 16-level histogram. If all pixels in a bitmap match the 16-level histogram values, STATUS[3] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the PANEL_MODE control field.

Address: 20F_0000h base + 2E0h offset = 20F_02E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD3			VALUE3					RSVD2			VALUE2					RSVD1			VALUE1					RSVD0			VALUE0					
W																																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

PXP_HIST16_PARAM0 field descriptions

Field	Description
31–29 RSVD3	Reserved, always set to zero.
28–24 VALUE3	GRAY3 value for 16-level histogram
23–21 RSVD2	Reserved, always set to zero.
20–16 VALUE2	GRAY2 value for 16-level histogram
15–13 RSVD1	Reserved, always set to zero.
12–8 VALUE1	GRAY1 value for 16-level histogram
7–5 RSVD0	Reserved, always set to zero.
VALUE0	GRAY0 (Black) value for 16-level histogram

53.7.48 16-level Histogram Parameter 1 Register (PXP_HIST16_PARAM1)

This register specifies four of the valid values for a 16-level histogram. If all pixels in a bitmap match the 16-level histogram values, STATUS[3] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the PANEL_MODE control field.

Address: 20F_0000h base + 2F0h offset = 20F_02F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD7			VALUE7					RSVD6			VALUE6					RSVD5			VALUE5					RSVD4			VALUE4				
W																																
Reset	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0

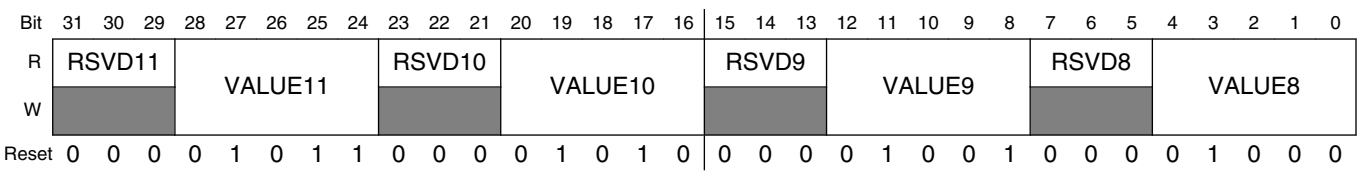
PXP_HIST16_PARAM1 field descriptions

Field	Description
31–29 RSVD7	Reserved, always set to zero.
28–24 VALUE7	GRAY7 value for 16-level histogram
23–21 RSVD6	Reserved, always set to zero.
20–16 VALUE6	GRAY6 value for 16-level histogram
15–13 RSVD5	Reserved, always set to zero.
12–8 VALUE5	GRAY5 value for 16-level histogram
7–5 RSVD4	Reserved, always set to zero.
VALUE4	GRAY4 value for 16-level histogram

53.7.49 16-level Histogram Parameter 2 Register (PXP_HIST16_PARAM2)

This register specifies four of the valid values for a 16-level histogram. If all pixels in a bitmap match the 16-level histogram values, STATUS[3] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the PANEL_MODE control field.

Address: 20F_0000h base + 300h offset = 20F_0300h



PXP_HIST16_PARAM2 field descriptions

Field	Description
31–29 RSVD11	Reserved, always set to zero.
28–24 VALUE11	GRAY11 value for 16-level histogram
23–21 RSVD10	Reserved, always set to zero.
20–16 VALUE10	GRAY10 value for 16-level histogram
15–13 RSVD9	Reserved, always set to zero.
12–8 VALUE9	GRAY9 value for 16-level histogram
7–5 RSVD8	Reserved, always set to zero.
VALUE8	GRAY8 value for 16-level histogram

53.7.50 16-level Histogram Parameter 3 Register (PXP_HIST16_PARAM3)

This register specifies four of the valid values for a 16-level histogram. If all pixels in a bitmap match the 16-level histogram values, STATUS[3] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the PANEL_MODE control field.

Address: 20F_0000h base + 310h offset = 20F_0310h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD15			VALUE15					RSVD14			VALUE14					RSVD13			VALUE13					RSVD12			VALUE12				
W																																
Reset	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0

PXP_HIST16_PARAM3 field descriptions

Field	Description
31–29 RSVD15	Reserved, always set to zero.
28–24 VALUE15	GRAY15 (White) value for 16-level histogram
23–21 RSVD14	Reserved, always set to zero.
20–16 VALUE14	GRAY14 value for 16-level histogram
15–13 RSVD13	Reserved, always set to zero.
12–8 VALUE13	GRAY13 value for 16-level histogram
7–5 RSVD12	Reserved, always set to zero.
VALUE12	GRAY12 value for 16-level histogram

53.7.51 PXP Power Control Register (PXP_POWER)

This register controls power states for PXP memories.

Address: 20F_0000h base + 320h offset = 20F_0320h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CTRL																				ROT_MEM_LP_STATE			LUT_LP_STATE_WAY1_BANKN			LUT_LP_STATE_WAY0_BANKN			LUT_LP_STATE_WAY0_BANK0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_POWER field descriptions

Field	Description
31–12 CTRL	Power control for the PXP.
11–9 ROT_MEM_LP_STATE	Select the low power state of the ROT memory. 0x0 **NONE** — Memory is not in low power state. 0x1 **LS** — Light Sleep Mode. Low leakage mode, maintain memory contents. 0x2 **DS** — Deep Sleep Mode. Low leakage mode, maintain memory contents. 0x4 **SD** — Shut Down Mode. Shut Down periphery and core, no memory retention.
8–6 LUT_LP_STATE_WAY1_BANKN	Select the low power state of the LUT's WAY0-BANK0,1,2,3 memory. 0x0 **NONE** — Memory is not in low power state. 0x1 **LS** — Light Sleep Mode. Low leakage mode, maintain memory contents. 0x2 **DS** — Deep Sleep Mode. Low leakage mode, maintain memory contents. 0x4 **SD** — Shut Down Mode. Shut Down periphery and core, no memory retention.
5–3 LUT_LP_STATE_WAY0_BANKN	Select the low power state of the LUT's WAY0-BANK1,2,3 memory. 0x0 **NONE** — Memory is not in low power state. 0x1 **LS** — Light Sleep Mode. Low leakage mode, maintain memory contents. 0x2 **DS** — Deep Sleep Mode. Low leakage mode, maintain memory contents. 0x4 **SD** — Shut Down Mode. Shut Down periphery and core, no memory retention.
LUT_LP_STATE_WAY0_BANK0	Select the low power state of the LUT's WAY0-BANK0 memory. 0x0 **NONE** — Memory is not in low power state. 0x1 **LS** — Light Sleep Mode. Low leakage mode, maintain memory contents. 0x2 **DS** — Deep Sleep Mode. Low leakage mode, maintain memory contents. 0x4 **SD** — Shut Down Mode. Shut Down periphery and core, no memory retention.

53.7.52 Next Frame Pointer (PXP_NEXT)

This register contains a pointer to a data structure used to reload the PXP registers at the end of the current frame.

To enable this functionality, software must write this register while the PXP is processing the current data frame (if the PXP is currently idle, this will also initiate an immediate load of registers from the pointer). The process of writing this register (WRITE operation) will set a semaphore in hardware to notify the control logic that a register reload operation must be performed when the current frame processing is complete. At the end of a frame, the PXP will fetch the register settings from this location, signal an interrupt to software, then proceed with rendering the next frame of data. Software may cancel the reload operation by issuing a CLEAR operation to this register. SET and TOGGLE operations should not be used when addressing this register. All registers will be reloaded with the exception of the following: STAT, CSCCOEFn, NEXT. All other registers will be loaded in the order they appear in the register map. After the pointer's contents have been loaded into the PXP's registers, the NEXT_IRQ interrupt will be issued (see the PXP_STATUS register).

EXAMPLE

```
// create register command structure in memory
  u32* pxp_commands0[48], pxp_commands1;
  u32  rc;

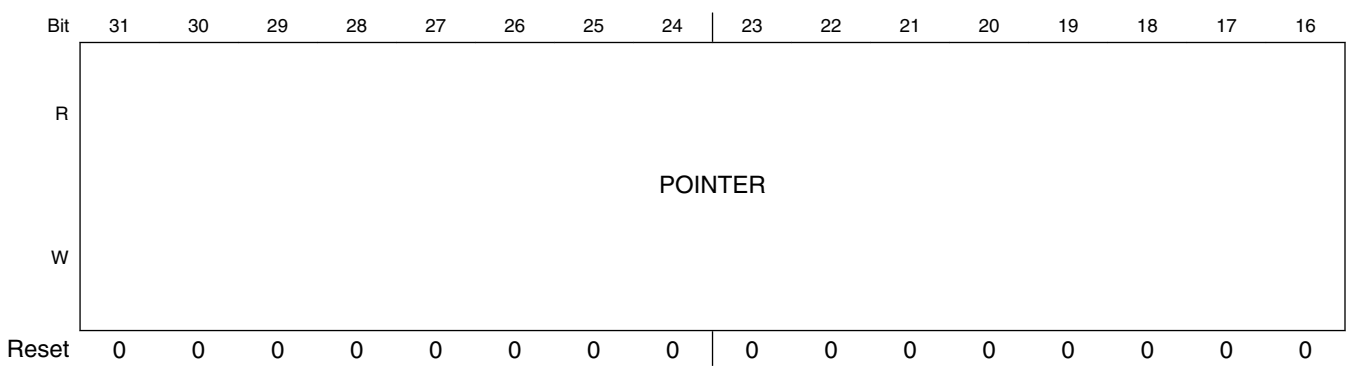
// initialize control structure for frame 0
  pxp_commands0[0] = ...; // CTRL
  pxp_commands0[1] = ...; // OUT Buffer
  ...
  pxp_commands0[47] = ..; // Overlay7 param2

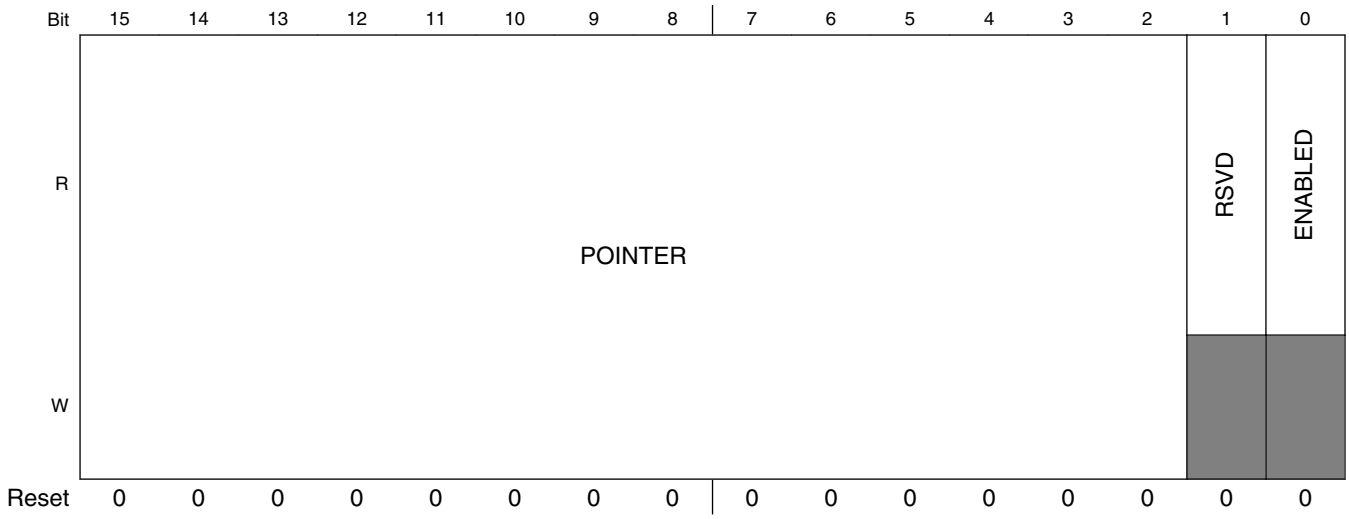
// initialize control structure for frame 1
  pxp_commands1[0] = ...; // CTRL
  pxp_commands1[1] = ...; // OUT Buffer
  ...
  pxp_commands1[47] = ..; // Overlay7 param2

// poll until a command isn't queued
  while (rc=PXP_NEXT_RD() & BM_PXP_NEXT_ENABLED );
  PXP_NEXT_WR(pxp_commands0);  // enable PXP operation 0 via command pointer

// poll until first command clears
  while (rc=PXP_NEXT_RD() & BM_PXP_NEXT_ENABLED );
  PXP_NEXT_WR(pxp_commands1);  // enable PXP operation 1 via command pointer
```

Address: 20F_0000h base + 400h offset = 20F_0400h





PXP_NEXT field descriptions

Field	Description
31–2 POINTER	A pointer to a data structure containing register values to be used when processing the next frame. The pointer must be 32-bit aligned and should reside in on-chip or off-chip memory.
1 RSVD	Reserved, always set to zero.
0 ENABLED	Indicates that the "next frame" functionality has been enabled. This bit reflects the status of the hardware semaphore indicating that a reload operation is pending at the end of the current frame.

Chapter 54
ROM Controller with Patch (ROMC)

54.1 Overview

The Read Only Memory Controller with ROM Patch (ROMC) acts as an interface between the Arm advanced high-performance bus (AHB - Lite) and the Read Only Memory. The ROMC consists of a ROM Controller and a ROM Patch. The ROM Patch is used to either patch code routines or fix data tables in the ROM area. There is an IP Bus interface to access the ROM Patch Registers. The following figure depicts the main functional sub-blocks of the ROMC.

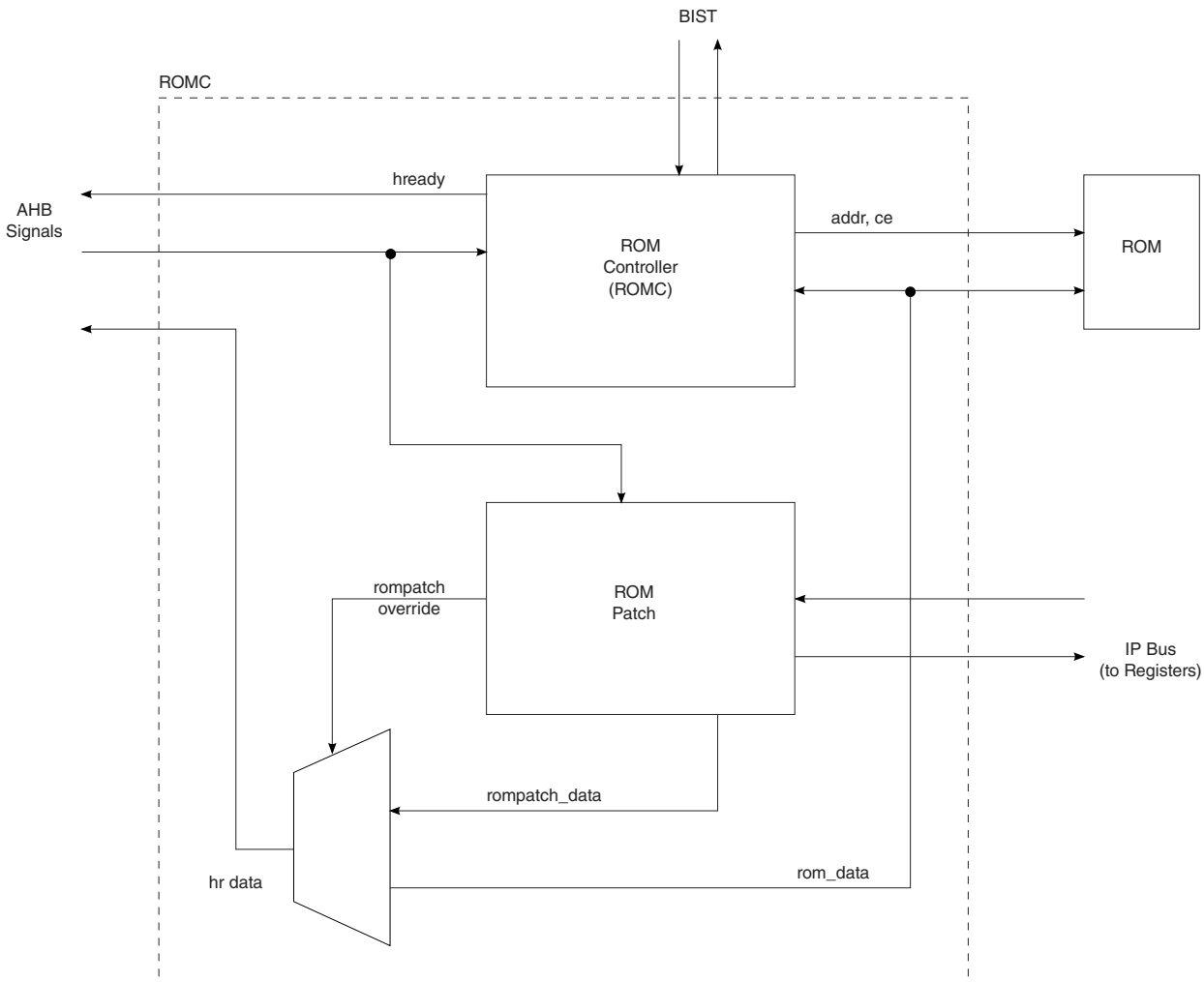


Figure 54-1. ROMC Block Diagram

54.1.1 Features

- Supports ROM size ranges from 16 Kbyte up to 4 Mbyte with increments of 1 Kbyte
- Supports opcode patching for a maximum of 16 different addresses in 4 Mbytes of ROM space
- Supports one-word data fixes for a max of 8 memory locations in 4 Mbytes of ROM space
- Supports patching of the Reset Vector (at 0x0000_0000) to allow external booting

54.1.2 Modes of Operation

There are two modes of operation: normal mode and BIST mode. In normal mode, the ROMC ensures correct reads from the ROM, assuming the memory complies with the characteristics and requirements for which the ROMC was designed.

54.1.2.1 Low Power Mode

There are two clock enables that are used to switch off parts of the ROMC logic when inactive. The first clock enable is used to disable the ROM Controller when the master connected to the AHB interface is not initiating a read to the ROM. The second clock enable is used to disable the registers used to program the ROM patch feature when the registers are not being accessed.

54.2 Clocks

The table found here describes the clock sources for ROMCP.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 54-1. ROMCP Clocks

Clock name	Clock Root	Description
hclk	ahb_clk_root	System / bus clock
hclk_reg	ipg_clk_root	System access clock
96krom_CLK	ahb_clk_root	ROM clock

54.3 Memory Map

54.3.1 ROM Memory Map in detail

The ROMC supports ROM sizes with a range of 16 Kbyte to 4 Mbyte with an increment of 1 Kbyte. The 16 Kbyte lower limit was chosen because the minimum size of security code on an Arm platform is approximately 16 Kbyte of code, which is only accessible in supervisor mode. Note that it is the MMU that controls whether any region of memory is secure.

The exception vectors must be secured as well, and must be put in the same area as the security code. Since they must reside at address 0x0000_0000, the entire 16 Kbyte of ROM which can only be accessible in supervisor mode is located at the very beginning of the platform memory map.

If the user chooses not to use the security code, a memory size smaller than 16 Kbyte can be connected to the platform (minimum of 1 Kbyte). The MMU can be programmed to allow any kind of access into this memory. However, if the ROM size is less than 16 Kbyte, memory aliasing will occur for all invalid addresses greater than the memory size but within the 16 Kbyte of space.

For ROM sizes bigger than 16 Kbyte, the rest of its physical size resides at the address starting at 0x0040_4000 (4 M+16 Kbyte) going up to [0x0040_4000 + (mem. size - 16Kbyte)].

54.4 Functional Description

This section is divided up into the ROM Controller Functional Description and the ROMC functional description.

54.4.1 ROM Controller (ROMC) Functional Description

54.4.1.1 Functionality overview

The ROMC serves two main functions. First, as an interface between the AHB-Lite bus on an Arm platform and the ROM. Second, it drives and receives several signals for the BIST engine. In normal mode of operation, the ROMC monitors the AHB-Lite for memory access requests and performs the memory operation to the ROM.

The ROMC includes the option to wait state all accesses from either the Arm or non-Arm masters to ROM in the event that timing requirements will not allow single HCLK clock cycle reads. If a wait state is required, the static inputs rom_wait_arm or rom_wait_alt_mstr can be set to 1 and accesses will take two HCLK clock cycles. If wait states are not required, rom_wait_arm or rom_wait_alt_mstr can be set to 0 and accesses will take one HCLK clock cycle to complete.

54.4.2 ROMC Functional Description

54.4.2.1 ROMC Disabling

All the bits in the ROMC_ROMPATCHENL register are cleared on Reset, disabling all the address comparators. Once the comparators have been enabled, the ROMC functions of data fixing and opcode patching can be quickly disabled by setting the DIS bit in the ROMC_ROMPATCHCNTL register. This bit is used to enable secure operations in which patching functions need to be disabled. This bit is cleared on Reset.

54.4.2.2 ROMC Event Priority

The ROMC has a total of 16 address comparators. The first 8 (0 through 7) comparators can be programmed for the data fixing function (through the 8 data fix enable bits in the ROMC_ROMPATCHCNTL register) while the rest are for opcode patching by default. This allows for potential multiple matching events involving both data fixing and opcode patch types. In these cases the ROMC assigns the highest priority to a data fixing event.

For example, if the ROMC is set up to data fix a certain address with comparator 4 and also opcode patch the same address with comparator 7, it will let comparator 4 have higher priority in indicating a match, and data from ROMC_ROMPATCHD4 will be put on the rompatch_romc_hrdata bus as the override value.

If multiple address matches of the same type level occur concurrently, then the ROMC will choose the source number based on the one with the highest source number. For example, the ROMC is setup to data fix the same location with address comparators 4 and 7, then address comparator 7 will have higher priority in indicating a match, and the value from ROMC_ROMPATCHD7 will be put on the rompatch_romc_hrdata bus as the orverride value. The same priority applies for an opcode patch event, except the override data is in the form of an SWI instruction with the comment field set to the source number with the highest priority.

54.4.2.3 Data Fixing

The data fixing feature allows ROM data to be updated by direct replacement when it is being read. This data usually originates from data tables, but can include Arm instructions. To enable data fixing on a certain address, this address value is written in to one of the first eight (0 through 7) of ROMC_ROMPATCHAxx registers and the same numbered bit set in the ROMC_ROMPATCHENL and ROMC_ROMPATCHCNTL registers. The data to be used for replacement is placed in the corresponding ROMC_ROMPATCHDxx.

The ROMC looks for a read access to ROM (either code fetch or data load) by snooping the AHB interface for read transactions. The address is compared with the values stored in the ROMC_ROMPATCHAxx[22:2] registers. If a match occurs from one of the comparators, the ROMC places the value in the corresponding ROMC_ROMPATCHDxx register on the read data bus by overriding the read data coming from the actual ROM (see the mux in Figure 54-1). The value on the read data bus is maintained until hready is asserted to terminate the access. In data fixing, the entire word is replaced so if a byte or half-word access occurs on a "data fix" location, the entire data word is replaced. The word being replaced is word aligned. (The two LSBs of the matching ROMC_ROMPATCHAxx are ignored in the data fix operation.)

54.4.2.4 Opcode Patching

The opcode patch feature provides the Arm core a mechanism to fetch updated versions of code routines that were originally programmed in ROM. This patching mechanism makes use of the SWI (software interrupt instruction) and a table of function pointers residing in writable memory. The opcode being patched is replaced with a SWI instruction by the ROMC. Subsequent processing of the SWI reads from a function pointers table to obtain the address of the replacement code. Execution resumes with this code patch.

To enable opcode patching of a certain address, this address value is written into one of the ROMPATCHAxx registers and the corresponding bit set in the ROMPATCHENL to enable the associated comparator. The register's LSB (ROMC_ROMPATCHxx[0]) should be set if THUMB mode patching is in effect for this address. The ROMC identifies a ROM read access by snooping the AHB interface. The address is compared with the values stored in the ROMC_ROMPATCHAxx[22:2] registers. If a match occurs from one of the comparators, the ROMC generates the opcode of a software interrupt (SWI) instruction with the comment field containing the number of the matching address comparator. This opcode and comment is placed on the read data bus until hready is asserted by the ROM controller to terminate the read access.

The type of SWI generated, (that is, either Arm or THUMB), is determined by the LSB of the ROMC_ROMPATCHAxx register associated with the opcode patch. This bit is cleared for Arm mode (32 bits). The ROMC generates a 32-bit SWI (opcode field is 0xEF, occupying bits [31:24] of the word), with the least significant 5 bits of the 24-bit comment field (bits [23:0]) containing the number of the matching address comparator. The rest of the comment field is filled with zeros. This means that the ROMC will use 16 of the 16777216 possible software interrupts. The ROMC overrides the read data from the ROM.

If the LSB of the matching ROMC_ROMPATCHAxx register is set, the opcode patch is in THUMB mode (16 bits or half word). The ROMC generates a 16-bit SWI instruction (opcode field is 0xDF, occupying bits [15:8] of the half word) with the least significant 5 bits of the 8-bit comment field containing with the source number of the address comparator. The rest of the comments field is filled with zeros. This means that the ROMC will use 16 of the 256 possible software interrupts. The ROMC puts this 16 bit SWI instruction value on the proper half of the rompatch_romc_hrdata bus. The other half is zeroed out. Which half of the bus contains the SWI opcode and comment depends on the mode (Big Endian or Little Endian) and the bit 1 of the matching ROMC_ROMPATCHAxx register. In Little Endian mode, the lower half is bits {15:0] and the upper half is bits {31:16]. The order is reversed in Big Endian mode.

In Little Endian mode (bigend signal negated), if bit 1 of the matching ROMC_ROMPATCHAxx is cleared (lower half word selected) then the SWI instruction is put on the lower 16 bits of the read data bus and the upper 16 bits are zeroed out. Only the lower 16 bits of the read data bus is overwritten by the ROMC data. If ROMC_ROMPATCHAxx[1] is set (upper half word selected), the SWI instruction is put on the upper 16 bits of the read data bus and the lower 16 bits are zeroed out. Only the upper 16 bits of the read data bus is overwritten.

In Big Endian mode (bigend asserted), if bit 1 of the matching ROMC_ROMPATCHAxx is cleared (lower half word selected) then the SWI instruction is put on the upper 16 bits of the read data bus while the lower 16 bits are zeroed out. Only the upper 16 bits of the read data bus is overwritten. If ROMC_ROMPATCHAxx[1] is set (upper word selected), the SWI instruction is put on the lower 16 bits and the upper 16 bits are zeroed out. Only the lower 16 bits of the read data bus is overwritten.

The eventual execution of the SWI causes the Arm to save the CPSR in SPSR_SVC, the address of the next instruction after the SWI in R14_SVC, enter Supervisor mode, and fetch the SWI vector at 0x8, which then takes it to a handler for further processing as described in the next section.

54.4.2.4.1 Typical Software Response to Opcode Patch

When the SWI handler executes it needs to determine whether the SWI was generated by the ROMC. This is done by loading the SWI instruction and extracting its comment field. The state of the Arm core (Arm or THUMB) when the SWI was executed dictates whether to load the instruction word (Arm) or half word (THUMB). This state information can be determined by testing the T bit (bit 5) of the SPSR. If it's set, the execution was in THUMB mode.

By convention, if the comment field of the SWI is greater than 16, the software interrupt was initiated by software (i.e. an operating system call), and a branch is taken to the appropriate handler routine for further processing. If the comment field is less than 16, the SWI was generated by the ROMC performing a code patch operation. In this case, the software then reads from a table of function pointers, using the value in the SWI comment field as the index into the table. The value that is read is the address of the code patch. This value is loaded into the PC to begin the execution of the code patch. The following code segment illustrates a typical handling of the SWI.

```
stmfd        sp!, {r0-r1,lr}       @ push register onto SWI stack
mrs         r0, spsr        @ get saved status register
tst          r0, #0x20     @ check if call was in THUMB mode
ldrneh      r0, [lr,#-2]       @ yes: load opcode half-word and
bicne       r0, r0, #0xff00   @ yes: extract THUMB comment
ldreq       r0, [lr,#-4]       @ no: load opcode word and
biceq       r0, r0, #0xff000000 @ no: extract ARM comment
                    @ now r0 has comment field
cmp         r0, #16      @ compare to 16 (maximum for ROMC)
ldrlt          lr, =rompatch_tbl_ptr @ < 16: get top of current ROMC
                    @ table; global variable which is
                    @ changeable per context
ldrlt         r1, [lr, r0, lsl #2]   @ < 16: read function pointer from
                    @ table assumed an array of pointers
                    @ patch functions
strlt          r1, [sp, #8]     @ < 16: store function pointer onto
                    @ stack in position of link register
ldmltfd       sp!, {r0-r1,pc}^      @ < 16: "fake" return from SWI, will
                    @ vector core to appropriate patch
                    @ function and set core back to previous
                    @ mode of operating
ldr          r1, =swi_hdler   @ >= 16: pointer to standard SWI
                    @ handler
mov         lr, pc         @ >= 16: set link register
bx          r1          @ >= 16: jump to standard SWI
                    @ handler
ldmfd        sp!, {r0-r1,pc}^      @ >= 16: pop registers from stack
```

54.4.2.5 External Boot Feature

Following a Reset event, the Arm issues an instruction fetch of the Reset Vector from address 0x0. This instruction, normally residing in ROM is usually a branch to a Reset handler or boot code which also normally resides in ROM. The ROMC external boot feature allows the bypassing of this code, using a different boot code residing perhaps in external memory.

This feature uses the data fix mechanism and works as follows: if the boot_int signal is negated when a Reset event occurred, the ROMC will perform a data fix of the Reset Vector at 0x0 with the following instruction (opcode 0xE59FF00C):

```
ldr          pc,[pc,#12]      @ read 0x0000_0014 for reset_vector
```

The value of PC when this instruction is executed is 8 so that a PC relative offset of 12 makes the source address 20 or 0x14. When this instruction executes, the Arm core reads from address 0x0000_0014, triggering a ROMC data fix operation which places the value

taken from the external boot address on the read data bus, with the two LSBs zeroed out. This value is returned to the Arm to be placed in the PC causing code fetch and execution to start from that address.

54.4.2.6 Alternate Masters and ROMC

The ROMC sits on the AHB bus of the internal ROM (ROMC). This means that the ROMC can modify values on the read data bus going to the master. Therefore, any master which reads an opcode patched or data patched location will read patched data.

54.5 ROMCP Memory Map/Register Definition

All registers are accessible through an IP Bus and can only be accessed in privileged mode. These registers can only be written with 32-bits stores and are clocked by hclk_reg.

ROMC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21A_C0D4	ROMC Data Registers (ROMC_ROMPATCH7D)	32	R/W	0000_0000h	54.5.1/4658
21A_C0D8	ROMC Data Registers (ROMC_ROMPATCH6D)	32	R/W	0000_0000h	54.5.1/4658
21A_C0DC	ROMC Data Registers (ROMC_ROMPATCH5D)	32	R/W	0000_0000h	54.5.1/4658
21A_C0E0	ROMC Data Registers (ROMC_ROMPATCH4D)	32	R/W	0000_0000h	54.5.1/4658
21A_C0E4	ROMC Data Registers (ROMC_ROMPATCH3D)	32	R/W	0000_0000h	54.5.1/4658
21A_C0E8	ROMC Data Registers (ROMC_ROMPATCH2D)	32	R/W	0000_0000h	54.5.1/4658
21A_C0EC	ROMC Data Registers (ROMC_ROMPATCH1D)	32	R/W	0000_0000h	54.5.1/4658
21A_C0F0	ROMC Data Registers (ROMC_ROMPATCH0D)	32	R/W	0000_0000h	54.5.1/4658
21A_C0F4	ROMC Control Register (ROMC_ROMPATCHCNTL)	32	R/W	0840_0000h	54.5.2/4659
21A_C0F8	ROMC Enable Register High (ROMC_ROMPATCHENH)	32	R	0000_0000h	54.5.3/4660
21A_C0FC	ROMC Enable Register Low (ROMC_ROMPATCHENL)	32	R/W	0000_0000h	54.5.4/4660
21A_C100	ROMC Address Registers (ROMC_ROMPATCH0A)	32	R/W	0000_0000h	54.5.5/4661
21A_C104	ROMC Address Registers (ROMC_ROMPATCH1A)	32	R/W	0000_0000h	54.5.5/4661
21A_C108	ROMC Address Registers (ROMC_ROMPATCH2A)	32	R/W	0000_0000h	54.5.5/4661
21A_C10C	ROMC Address Registers (ROMC_ROMPATCH3A)	32	R/W	0000_0000h	54.5.5/4661
21A_C110	ROMC Address Registers (ROMC_ROMPATCH4A)	32	R/W	0000_0000h	54.5.5/4661
21A_C114	ROMC Address Registers (ROMC_ROMPATCH5A)	32	R/W	0000_0000h	54.5.5/4661
21A_C118	ROMC Address Registers (ROMC_ROMPATCH6A)	32	R/W	0000_0000h	54.5.5/4661

Table continues on the next page...

ROMC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21A_C11C	ROMC Address Registers (ROMC_ROMPATCH7A)	32	R/W	0000_0000h	54.5.5/4661
21A_C120	ROMC Address Registers (ROMC_ROMPATCH8A)	32	R/W	0000_0000h	54.5.5/4661
21A_C124	ROMC Address Registers (ROMC_ROMPATCH9A)	32	R/W	0000_0000h	54.5.5/4661
21A_C128	ROMC Address Registers (ROMC_ROMPATCH10A)	32	R/W	0000_0000h	54.5.5/4661
21A_C12C	ROMC Address Registers (ROMC_ROMPATCH11A)	32	R/W	0000_0000h	54.5.5/4661
21A_C130	ROMC Address Registers (ROMC_ROMPATCH12A)	32	R/W	0000_0000h	54.5.5/4661
21A_C134	ROMC Address Registers (ROMC_ROMPATCH13A)	32	R/W	0000_0000h	54.5.5/4661
21A_C138	ROMC Address Registers (ROMC_ROMPATCH14A)	32	R/W	0000_0000h	54.5.5/4661
21A_C13C	ROMC Address Registers (ROMC_ROMPATCH15A)	32	R/W	0000_0000h	54.5.5/4661
21A_C208	ROMC Status Register (ROMC_ROMPATCHSR)	32	w1c	0000_0000h	54.5.6/4662

54.5.1 ROMC Data Registers (ROMC_ROMPATCH*n*D)

The ROMC data registers (ROMC_ROMPATCH 7D through ROMC_ROMPATCH 0D) store the data to use for the 8 1-word data fix events. Each register is associated with an address comparator (7 through 0). When a data fixing event occurs, the value in the data register corresponding to the comparator that has the address match is put on the romc_hrdata[31:0] bus until romc_hready is asserted by the ROM controller to terminate the access. A MUX external to the ROMC will select this data over that of romc_hrdata[31:0] in returning read data to the Arm core. The selection is done with the control bus rompatch_romc_hrdata_ovr[1:0] with both bits asserted by the ROMC.

If more than one address comparators match, the highest-numbered one takes precedence, and the value in corresponding data register is used for the patching event.

Address: 21A_C000h base + D4h offset + (4d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATAX																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ROMC_ROMPATCH*n*D field descriptions

Field	Description
DATAX	Data Fix Registers - Stores the data used for 1-word data fix operations. The values stored within these registers do not affect the writes to the memory system. They are selected over the read data from ROM when a data fix event occurs. If any part of the 1-word data fix is read, then the entire word is replaced. Therefore, a byte or half-word read will cause the ROMC to replace the entire word. The word is word address aligned.

54.5.2 ROMC Control Register (ROMC_ROMPATCHCNTL)

The ROMC control register (ROMC_ROMPATCHCNTL) contains the block disable bit and the data fix enable bits. The block disable bit provides a means to disable the ROMC data fix and opcode patching functions, even when the address comparators are enabled. The External Boot feature is not affected by this bit. The eight data fix enable bits (0 through 7), when set, assign the associated address comparators to data fix operations

NOTE

Bits 27 and 22 always read as 1s.

Address: 21A_C000h base + F4h offset = 21A_C0F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		DIS	Reserved												
Reset	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								DATAFIX							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ROMC_ROMPATCHCNTL field descriptions

Field	Description
31–30 -	This field is reserved. Reserved
29 DIS	ROMC Disable -- This bit, when set, disables all ROMC operations. This bit is used to enable secure operations. 0 Does not affect any ROMC functions (default) 1 Disable all ROMC functions: data fixing, and opcode patching
28–8 -	This field is reserved. Reserved
DATAFIX	**Data Fix Enable - Controls the use of the first 8 address comparators for 1-word data fix or for code patch routine.** 0 Address comparator triggers a opcode patch 1 Address comparator triggers a data fix

54.5.3 ROMC Enable Register High (ROMC_ROMPATCHENH)

The ROMC enable register high (ROMC_ROMPATCHENH) and ROMC enable register low (ROMC_ROMPATCHENL) control whether or not the associated address comparator can trigger a opcode patch or data fix event. This implementation of the ROMC only has 16 comparators, therefore ROMC_ROMPATCHENH and the upper half of ROMC_OMPATCHENL are read-only. ROMC_ROMPATCHENL[15:0] are associated with comparators 15 through 0. ROMC_ROMPATCHENLH[31:0] would have been associated with comparators 63 through 32.

Address: 21A_C000h base + F8h offset = 21A_C0F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ROMC_ROMPATCHENH field descriptions

Field	Description
-	This field is reserved. Reserved

54.5.4 ROMC Enable Register Low (ROMC_ROMPATCHENL)

The ROMC enable register high (ROMC_ROMPATCHENH) and ROMC enable register low (ROMC_ROMPATCHENL) control whether or not the associated address comparator can trigger a opcode patch or data fix event. This implementation of the ROMC only has 16 comparators, therefore ROMC_ROMPATCHENH and the upper half of ROMC_ROMPATCHENL are read-only. ROMC_ROMPATCHENL[15:0] are associated with comparators 15 through 0. ROMC_ROMPATCHENLH[31:0] would have been associated with comparators 63 through 32.

Address: 21A_C000h base + FCh offset = 21A_C0FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																ENABLE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

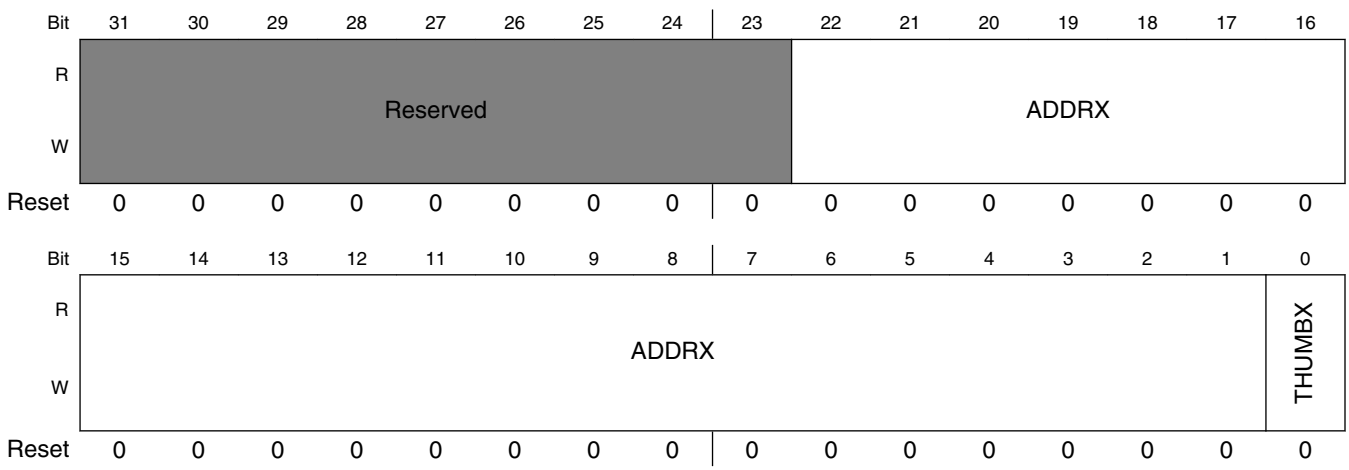
ROMC_ROMPATCHENL field descriptions

Field	Description
31–16 Reserved	This field is reserved.
ENABLE	**Enable** Address **Comparator -** This bit enables the corresponding address comparator to trigger an event. 0 Address comparator disabled 1 Address comparator enabled, ROMC will trigger a opcode patch or data fix event upon matching of the associated address

54.5.5 ROMC Address Registers (ROMC_ROMPATCH*n*A)

The ROMC address registers (ROMC_ROMPATCHA0 through ROMC_ROMPATCHA15) store the memory addresses where opcode patching begins and data fixing occurs. The address registers ROMC_ROMPATCHA0 through ROMC_ROMPATCHA15 are each 21 bits wide and dedicated to one 4 Mbyte memory space. Bits 21 through 2 are address bits, to be compared with romc_haddr[21:2] for a match; bit 1 is also an address bit used for half word selection. Bit 0 is the mode bit (set to 1 for THUMB mode). 1-word data fixing can only be used on the first 8 of the address comparators. ROMC_ROMPATCHA0 through ROMC_ROMPATCHA15 are associated each with address comparators 0 through 15.

Address: 21A_C000h base + 100h offset + (4d × i), where i=0d to 15d



ROMC_ROMPATCH*n*A field descriptions

Field	Description
31–23 -	This field is reserved. Reserved

Table continues on the next page...

ROMC_ROMPATCH*n*A field descriptions (continued)

Field	Description
22–1 ADDRX	Address Comparator Registers - Indicates the memory address to be watched. All 16 registers can be used for code patch address comparison. Only the first 8 registers can be used for a 1-word data fix address comparison. Bit 1 is ignored if data fix. Only used in code patch
0 THUMBX	THUMB Comparator Select - Indicates that this address will trigger a THUMB opcode patch or an Arm opcode patch. If this watchpoint is selected to be a data fix, then this bit is ignored as all data fixes are 1-word data fixes. 0 Arm patch 1 THUMB patch (ignore if data fix)

54.5.6 ROMC Status Register (ROMC_ROMPATCHSR)

The ROMC status register (ROMC_ROMPATCHSR) indicates the current state of the ROMC and the source number of the most recent address comparator event.

Address: 21A_C000h base + 208h offset = 21A_C208h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved														SW	Reserved
W															w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										SOURCE					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ROMC_ROMPATCHSR field descriptions

Field	Description
31–18 -	This field is reserved. Reserved
17 SW	ROMC AHB Multiple Address Comparator matches Indicator - Indicates that multiple address comparator matches occurred. Writing a 1 to this bit will clear this it. 0 no event or comparator collisions 1 a collision has occurred
16–6 -	This field is reserved. Reserved
SOURCE	ROMC Source Number - Binary encoding of the number of the address comparator which has an address match in the most recent patch event on ROMC AHB. If multiple matches occurred, the highest priority source number is used.

Table continues on the next page...

ROMC_ROMPATCHSR field descriptions (continued)

Field	Description
	0 Address Comparator 0 matched 1 Address Comparator 1 matched 15 Address Comparator 15 matched

Chapter 55
Smart Direct Memory Access Controller (SDMA)

55.1 Overview

The Smart Direct Memory Access (SDMA) controller offers highly-competitive DMA features combined with software-based virtual-DMA flexibility. It enables data transfers between peripheral I/O devices and internal/external memories.

The SDMA controller helps maximize system performance by off-loading the Arm core in dynamic data routing.

55.1.1 Block Diagram

The figure below shows a block diagram of the SDMA controller. It includes the custom RISC core along with its RAM, ROM, DMA units, and the scheduler.

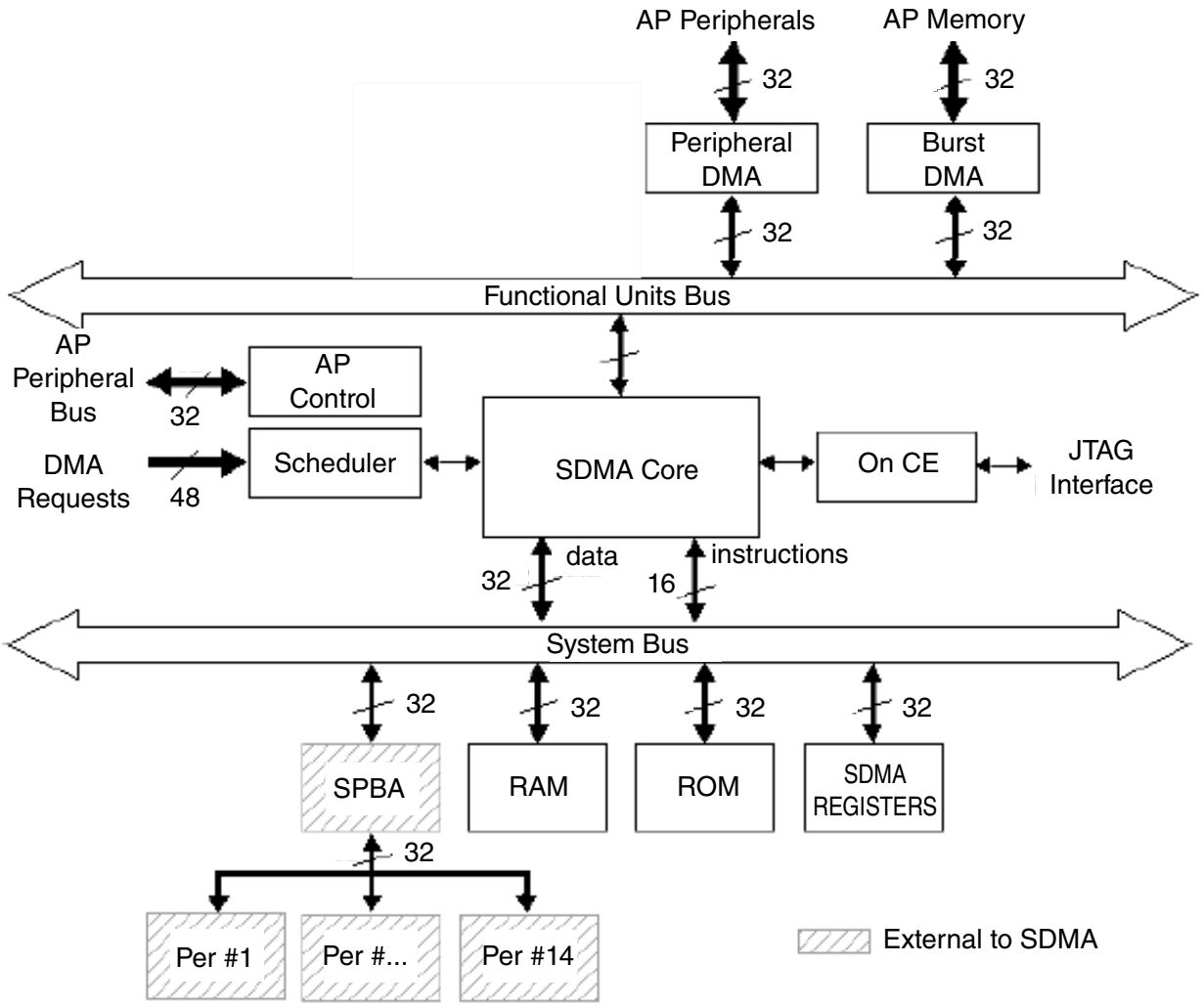


Figure 55-1. SDMA Block Diagram

The SDMA core executes short routines that perform DMA transfers; these routines are called *scripts*. The SDMA core interfaces to its own memory via the SDMA system bus. The SDMA system bus supports a 32-bit data path and a 16-bit address bus. The system bus datapath is used for both 16-bit instruction (program) memory access and 32-bit data access. DMA units interface to the core via the Functional Unit Bus and use dedicated registers to perform DMA transfers.

The SDMA memory contains a ROM and a RAM. The ROM contains startup scripts (for example, boot code) and other common utilities, which are referenced by the scripts that reside in the RAM. The internal RAM is divided into a context area and a script area (more details about this mapping are available in Instruction Memory Map and Data Memory Map).

Every transfer channel requires one context area to keep the contents of all the core and unit registers while inactive. Channel scripts are downloaded into the internal RAM by the SDMA using a dedicated channel that is started during the boot sequence. Downloads are invoked using commands and pointers provided by the Arm platform. Every channel contains a corresponding channel script located in RAM and/or ROM that can be reconfigured independently as-needed. Channel scripts can be stored in an external memory and downloaded when needed. The SDMA can be configured with any mixture of scripts to enable an endless combination of supported services.

The scheduler monitors and detects DMA requests, mapping them to channels, and mapping individual channels to a pre-configured priority. At any given point, the scheduler presents the highest priority channel that requires service to the SDMA core. A special SDMA core instruction is used to "conditionally yield" the current channel being executed to an eligible channel that requires service. If (and only if) there is an eligible channel pending, will the current channel execution be preempted.

There are two yield instructions that differently determine the eligible channels: In the first version, eligible channels are pending channels with a strictly higher priority than the current channel priority. In the second version (yieldge), eligible channels are pending channels with a priority that is greater or equal to the current channel priority. The scheduler detects devices that need service through its 48 DMA request inputs. After a request is detected, the scheduler determines the channel(s) that is (are) triggered by this request and marks it (them) as pending in the "Channel Pending (EP)" register. The priorities of all the pending channels are continuously evaluated in order to update the highest pending priority. The channel pending flag is cleared by the channel script when the transfer has completed.

The Arm platform control block contains the control registers used to configure the 32 individual channels. There are 48 Channel Enable registers, and every register maps one DMA request to any desired combination of channels. The 32 Priority registers are used to assign a programmable 1-of-7 level priority to every possible channel. This block also contains all other control registers that the Arm platform can access.

The 48 DMA requests that are connected to the scheduler come from a variety of sources. The "receive register full" and "transmit register empty" signals found in the UART and USB ports are typical examples of DMA requests that can be connected to the SDMA. These requests can be used to trigger a specific SDMA channel, or several channels.

There is an OnCE compatible debug port for product development. The OnCE includes support for setting breakpoints, single-step and trace, and register dump capability. In addition, all memory locations are accessible from the debug port.

55.1.2 Features

The following are the SDMA features:

- Multi-channel DMA supporting up to 32 time-division multiplexed DMA channels
- Hardware or software driven triggers for each channel
- 48 hardware driven triggers that can be mapped to any channel.
- Memory accesses including linear addressing, FIFO addressing and 2D addressing
- Fast context-switching with two-level, priority-based preemptive multi-tasking
- 16-bit instruction-set micro-RISC engine (the SDMA core)
- Two DMA units with some or all the following features:
  - Auto-flush and prefetch capability
  - Flexible address management (increment, decrement, and no address changes on source and destination address)
  - Misaligned data-transfer support
  - Uni-directional and bi-directional flows (copy mode)
  - Up to eight-word buffers for configurable burst transfers
- Support of byte-swapping
- An available API and library of scripts
- Little-Endian and Big-Endian modes
- Hardware handshakes for low-power entry sequence
- Security support to lock contents of the SDMA script RAM.
- 4-Kbyte ROM containing startup scripts (for example, boot code) and other common utilities that can be referenced by RAM-located scripts
- 8-Kbyte RAM area is divided into a processor context area and a code space area used to store channel scripts that are downloaded from the system memory
- Debug support, including a OnCE port, real-time monitors, and embedded cross-trigger events
- Supported clock frequencies in process:
  - Configurable clock options for the SDMA core and the Arm platform DMA units
    - 1:2 ratio with maximum of SDMA core running at Arm platform Peripheral Bus speed and DMA running at max DMA frequency.
    - 1:1 ratio when both SDMA core and Arm platform DMA clocks are set to the Arm platform Peripheral Bus speed.
- Peripheral bus interface for configuration register programming by the Arm platform
- The SDMA RISC engine (arithmetic and logic operations), which is referred to as the "SDMA core."
- An internal peripheral bus connected to the Shared Peripherals Bus Interface (SPBA) that enables access to up to 14 shared peripherals. SDMA supports 32-bit accesses to word peripherals and 16-bit accesses to half-word peripherals.

- The peripheral DMA unit that is hooked-up to the Arm platform Crossbar Switch to service Arm peripherals
- The burst DMA unit is able to perform burst accesses to the external memory
- All the DMA units are 32-bit AHB masters. They are connected to different buses, thus allowing concurrent accesses.

55.2 External Signals

The table found here describes the external signals of SDMA.

Table 55-1. SDMA External Signals

Signal	Description	Pad	Mode	Direction
SDMA_EXT_EVENT0	Event 0 signal	DISP0_DAT16	ALT4	I
		GPIO_17	ALT3	
SDMA_EXT_EVENT1	Event 1 signal	DISP0_DAT17	ALT4	I
		GPIO_18	ALT3	

55.3 Clocks

The following table describes the clock sources for SDMA. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information. For functional information regarding module clocks, see SDMA Clocks and Low Power Modes.

Table 55-2. SDMA Clocks

Clock name	Clock Root	Description
events_sync_clk (clk)	ahb_clk_root	Arm peripheral / events clock
ips_hostctrl_clk	ipg_clk_root	Host control clock
ap_ahb_clk	ahb_clk_root	Arm platform bus clock
core_clk	ipg_clk_root	Module / Core clock
tck	-	JTAG access clock

55.4 Functional Description

The figure below shows the SDMA topology, and is composed of the following components:

- SDMA Core (SDMA Core)
- SDMA Scheduler (Scheduler)
- Functional Units:
  - Burst DMA (Burst DMA Unit)
  - Peripheral DMA (Peripheral DMA Unit)
- Arm platform Control for Arm control register access.
- Internal RAM and ROM Memory (SDMA Programming Model)
- OnCE debug Port (The OnCE Controller)

The functional unit bus provides access by the SDMA core to the DMA units. The system bus provides access to SDMA internal memory and also supports up to 14 peripherals.

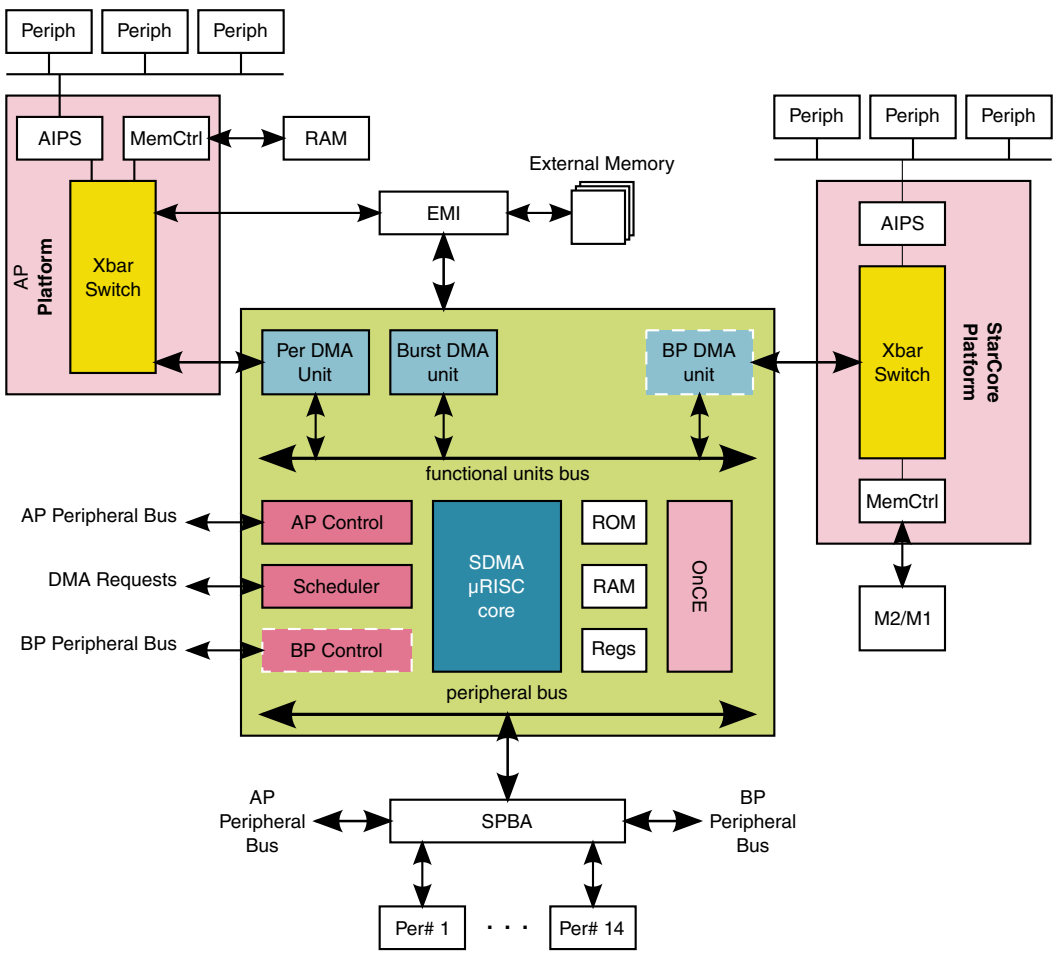


Figure 55-2. SDMA Connections

55.4.1 SDMA Core

The SDMA core is a customized RISC-like processor that is specifically developed to control DMA units and perform L1 tasks like byte-stuffing or framing.

The SDMA core incorporates on-chip debug capability using the OnCE.

The SDMA core is based on a 32-bit register architecture with 16-bit instructions. There are eight general purpose 32-bit registers, four flags (T, LM, SF, and DF), and four PCU registers (PC, RPC, SPC, and EPC) that can address 16,384 16-bit instructions.

55.4.1.1 SDMA Core Structure

The figure found here shows the structure of the SDMA core. It also shows the different registers, calculation resources, and possible data movements.

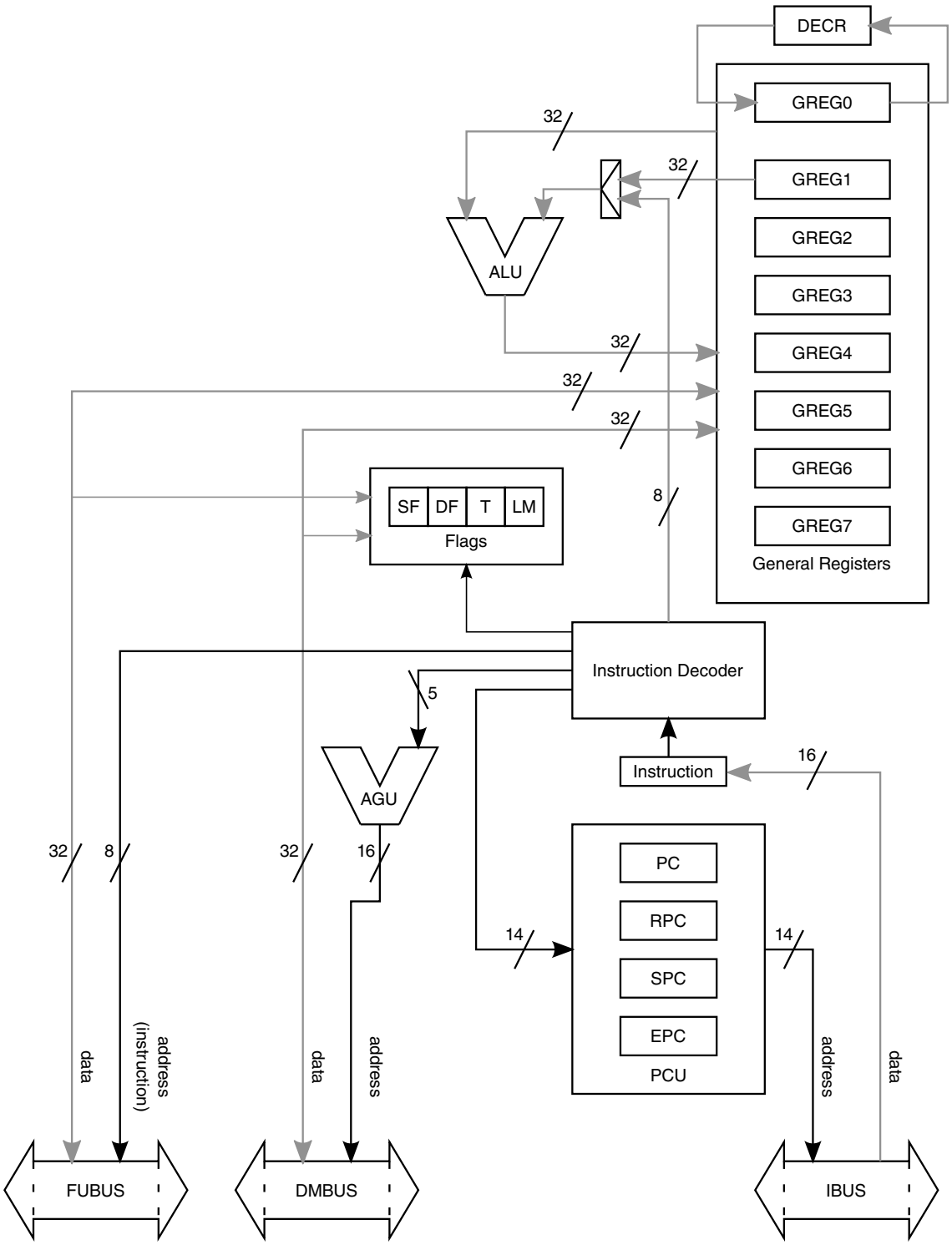


Figure 55-3. SDMA Core

- The Program Control Unit (PCU) is described in Program Control Unit (PCU). It handles the state of the core and generates the instruction fetch addresses. Instructions are retrieved from the Instruction Bus (IBUS) and stored in the SDMA

core instruction register prior to their decoding. The PCU contains the following registers:
    - The Program Counter (PC) contains the address of the current instruction.
    - The Return Program Counter (RPC) contains the address of the instruction that follows a jump to the subroutine.
    - The Start Program Counter (SPC) contains the address of the first instruction of the current hardware loop.
    - End Program Counter (EPC) contains the address of the last instruction of the current hardware loop.
- The other core registers are the general purpose registers (GREGn) and the flags.
    - The general purpose registers can be used to hold data and addresses. They can be loaded with immediate values (for example, 8-bit data that are encoded in the instruction), results of calculations that were performed with the ALU, 32-bit data that comes from the memory or peripherals via the Data Memory Bus (DMBUS), 32-bit data that comes from the DMAs via the Functional Units Bus (FUBUS) or another general purpose register. Their content can be the operands of the ALU, the data to send on either bus (DMBUS or FUBUS), or a pointer to memory (DMBUS address).
    - The general register 0 (GREG0) is also the hardware loop counter. In hardware loops, it cannot be used for any other purpose. This register uses a dedicated decrement unit (DECR) shown in Figure 55-3.
    - The flags reflect the status of operations:
        - SF and DF are set when the last load or store on either bus (FUBUS or DMBUS) received an error response.
        - LM is set when the core is executing instructions inside a hardware loop.
        - T is set when the ALU operation result was 0 or the loop counter reaches 0 (the latter is preponderant when an ALU operation is the last instruction of a hardware loop).
- The ALU has two operands: any general register and either a second general register or an immediate value. The result is always stored into the first general register. A NOP function can be utilized by moving a register's contents into itself (For example, the instruction: mov R0,R0).
- The 16-bit instructions are fetched via the instruction bus (IBUS) whose address is driven by the PC. The SDMA RAM and ROM are visible to the core as 16-bit devices through this interface.
- The memory (RAM and ROM), memory mapped registers, and external peripherals are accessed via the DMBUS. The address is always taken from a general register whose content is added to a 5-bit immediate value. This is the only available addressing mode. The DMBUS is a 32-bit data bus. Except for the peripherals that

are external to the SDMA, the address accuracy is the 32-bit word (for example, adding 1 to an address points to the next word, not the next byte).
- The functional units are accessed via the FUBUS connection. The data is exchanged with any general register, but the address (which in fact is the instruction and the selector of the functional unit) comes from an 8-bit field of the corresponding load or store.

55.4.1.2 Program Control Unit (PCU)

This part of the SDMA core is dedicated to the control of the RISC engine, as implied by the instructions that are executed. Its behavior is determined by the instruction type and the inputs of the SDMA.

It contains the PC, RPC, SPC, and EPC registers that are described in SDMA Core Structure.

55.4.1.2.1 Instruction Types

The state sequence and the delay of execution vary according to the type of the instruction. There are six possible categories of instructions, as follows:

1. Standard: Most of the instructions belong to this category, and always last 1 cycle.
2. ldf/stf: These are respectively the load and store instructions that access the functional units. They last 1+n cycles where n is the number of wait-states of the targeted functional unit.
3. ld/st: These are the load and store instructions that access the memory and peripherals. They last 1+n cycles where n is the number of wait-states of the targeted device (1 for the ROM, RAM, and memory mapped registers, 1 + the external peripheral wait-states). These instructions always last at least two cycles, but the core is able to handle them in one cycle. The first wait-state is inserted outside the core.
4. Branch: These are all the instructions that cause the Program Counter to point to another instruction other than the following one (for example, one that breaks the sequential flow). There are the absolute jumps, the conditional branches, the jump to the sub-routines, and the return from the sub-routine.
5. Loop,Modified Load or Store: The hardware loop instruction modifies the potential behavior of any load or store inside the loop (for example, when the LM flag is set). A jump may be implied after any such load or store if it received an error. The error causes an early exit of the loop, which means a jump to the instruction that follows the one that is pointed to by EPC. An additional cycle is required by the PCU to perform the jump (+1 to the ld/st/ldf/stf original execution delay). Although there is

usually an implicit jump after the last instruction of the loop when the PC goes back to SPC, this is performed at no cycle cost.

6. Done: The done, yield, or yieldge instructions are used to control channel switching. When no channel switching is performed, these instructions last a single cycle. When there is a change of channel or context switch, the delay is variable and depends on many factors (as detailed in Context Switching).

55.4.1.2.2 PCU States

The PCU state is visible through outputs of the SDMA (see Real-Time Debug Outputs) or the OnCE status register(see OnCE Status Register (OSTAT)).

The PCU state is a four-bit field that can take the values shown in the following table. Figure 55-4 shows the possible state transitions and the corresponding conditions.

Table 55-3. PCU States

Value	State	Description
0	Program	This is the usual instruction cycle.
1	Data	This state is inserted when there are wait-states during a load or a store on the data bus (ld/st type).
2	Change of Flow	This is the second cycle of any instruction that breaks the sequence of instructions (branch and done types). This state lasts only a single cycle; it is always followed by the Program state.
3	Error in Loop	This state is used when an error causes a hardware loop exit (loop-modified load or store type). This state only lasts a single cycle; it is always followed by the Program state.
4	Debug	The SDMA is stopped in debug mode.
5	Functional Unit	This state is inserted when there are wait-states during a load or a store on the functional units bus (ldf/stf type).
6	Sleep	No script is running: The core is idle after saving the last channel context.
7	Save	The context switch FSM is saving the current channel.
8	Program in Sleep	Same as Program except there is no associated channel, this state is used when instructions are executed after entering debug mode, whereas the core was in either Sleep mode.
9	Data in Sleep	This is the same as Data except there is no associated channel.
10	Change of Flow in Sleep	This is the same as Change of Flow except there is no associated channel. This state only lasts a single cycle, and is always followed by the Program in Sleep state.
11	Error in Loop in Sleep	This is the same as Error in Loop except there is no associated. channel. This state only lasts a single cycle, and is always followed by the Program in Sleep state.
12	Debug in Sleep	This is the same as Debug except the core was put in debug mode when no channel was active.
13	Functional Unit in Sleep	This is the same as Functional Unit except there is no associated channel.

Table continues on the next page...

Table 55-3. PCU States (continued)

Value	State	Description
14	Sleep after Reset	This shows that no script is running, and the core is idle after a reset. When a channel becomes active, no context is restored but the core starts its boot program located at address 0 (or the address available in register in Channel 0 Boot Address (SDMAARM_CHN0ADDR)).
15	Restore	The context switch FSM is restoring the next channel context.

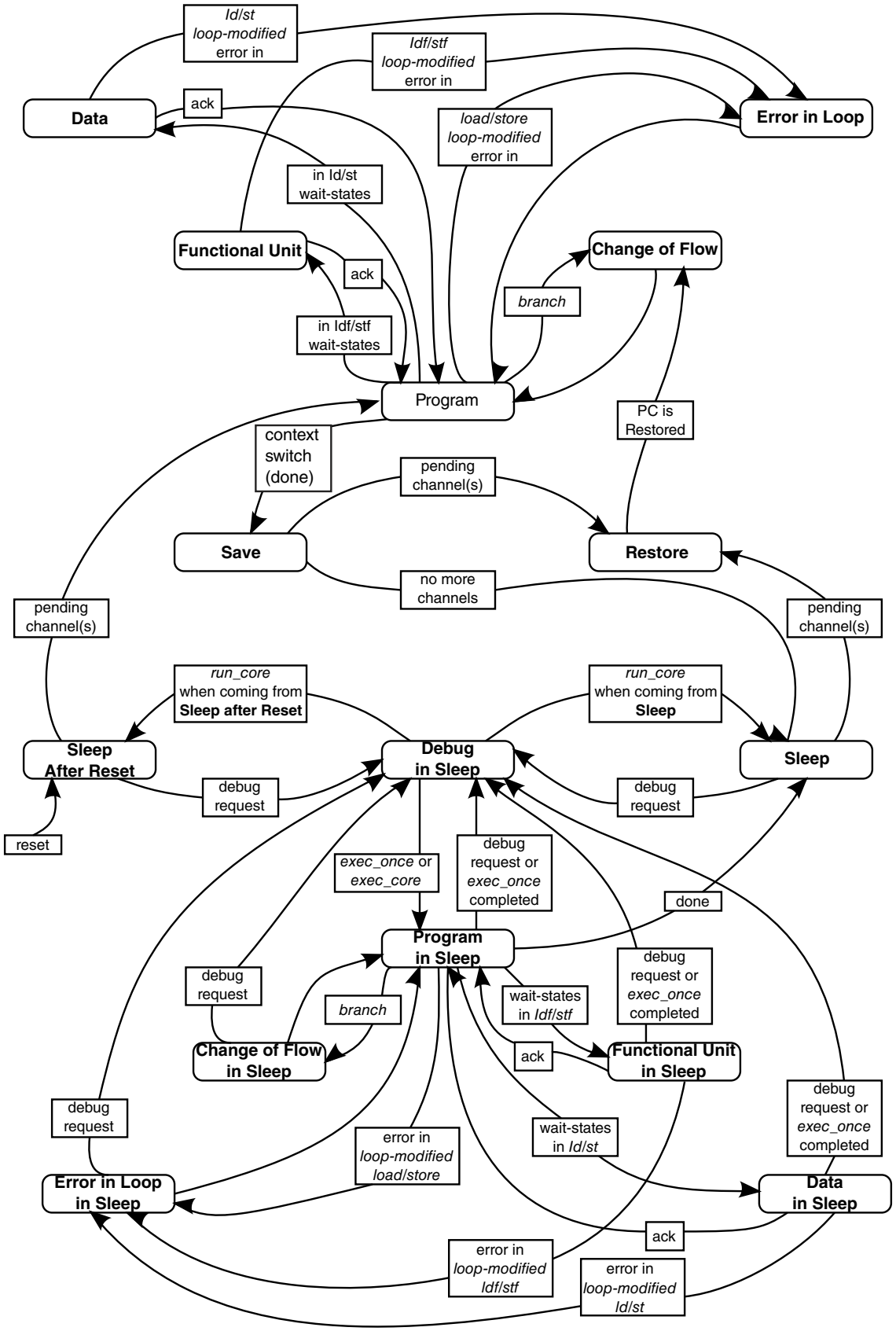


Figure 55-4. PCU State Diagram

55.4.1.3 SDMA Core Memory

The SDMA has two memory spaces: one for the instructions and one for the data. As both spaces share the same resources (ROM and RAM devices), the system bus manages possible conflicts when the core accesses the same resource for both an instruction read and a data read or write.

Program and data memory is further described in Address Space.

Instructions of 16-bit width are stored in 32-bit wide devices and can be accessed as data. The mapping is Big Endian: an even instruction address (terminated by 0) accesses the most significant part of the 32-bit data (bits [31:16]), and an odd instruction address (terminated by 1) accesses the least significant part of the 32-bit data (bits [15:0]). Instructions can be fetched out of internal ROM or RAM.

Data can be read from ROM, RAM, memory mapped registers, and external peripherals, and written to the same devices (except the ROM).

The ROM contains bootload scripts, channel scripts, and common subroutines which may be referenced by channel scripts elsewhere in the ROM or RAM.

The RAM is divided into a context area and a code space area which may be used to store channel scripts. The RAM contains undefined values after a hardware reset. Channel scripts and initial context values are downloaded into RAM using channel 0 which is reserved for bootload functions.

55.4.2 Scheduler

All channel scheduling hardware is included in the Scheduler.

55.4.2.1 Primary Functions

The scheduler is a hardware-based design used to coordinate the timely execution of 32 virtual DMA channels by the SDMA core on the basis of channel status and priority.

The scheduler performs the following functions:

- Monitors, detects, and registers the occurrence of any one of the 48 DMA requests
- Links a specific request to a channel or group of channels (channel mapping)
- Ignores requests that are not mapped to a previously configured channel
- Maintains a list of all the channels that are requesting service

- Assigns a pre-programmed priority level (1 of 7) to every channel requesting service
- Detects and flags overrun/underrun conditions

55.4.2.2 Channels and DMA Requests

55.4.2.2.1 Channels

A Virtual Channel (hereafter simply called a channel) manages a flow of data through the SDMA. Flows are typically unidirectional.

The SDMA can have up to 32 simultaneously operating channels, numbered from 0 to 31. Channel 0 is usually dedicated to control the SDMA script downloading. All the channels can be assigned by the Arm platform software.

55.4.2.2.2 DMA Requests

A DMA request is caused by externally (for example, external to the SDMA) controlled conditions (for example, UART receive FIFO reaches a threshold). The SDMA currently supports up to 48 DMA requests.

55.4.2.2.3 Mapping from DMA Requests to Channels and Priorities

A channel can stall waiting on a single DMA request. A single DMA request can awake more than one channel (in fact, any request can awake any combination of channels).

The mapping between DMA requests and channels is program-controlled. There is a storage element assigned for each of the 48 requests that contains a bitmap table of the channels that are awakened by the event.

Every channel also has a three-bit register that indicates its priority.

55.4.2.3 Scheduler Functional Description

Scheduler Overview describes the behavior of the SDMA scheduler-from the channel enabling conditions to the highest priority pending channel selection.

55.4.2.3.1 Scheduler Overview

The scheduler algorithm is built in hardware. It is provided with possibilities for the Arm platform to control its behavior.

The scheduler processes incoming DMA requests, maps detected requests to 0, one, or several channels, maintains a list of channels that are requesting service (pending channels), identifies the top priority and its associated channel, and selects the next active channel when the current channel yields.

The following figure shows a functional overview.

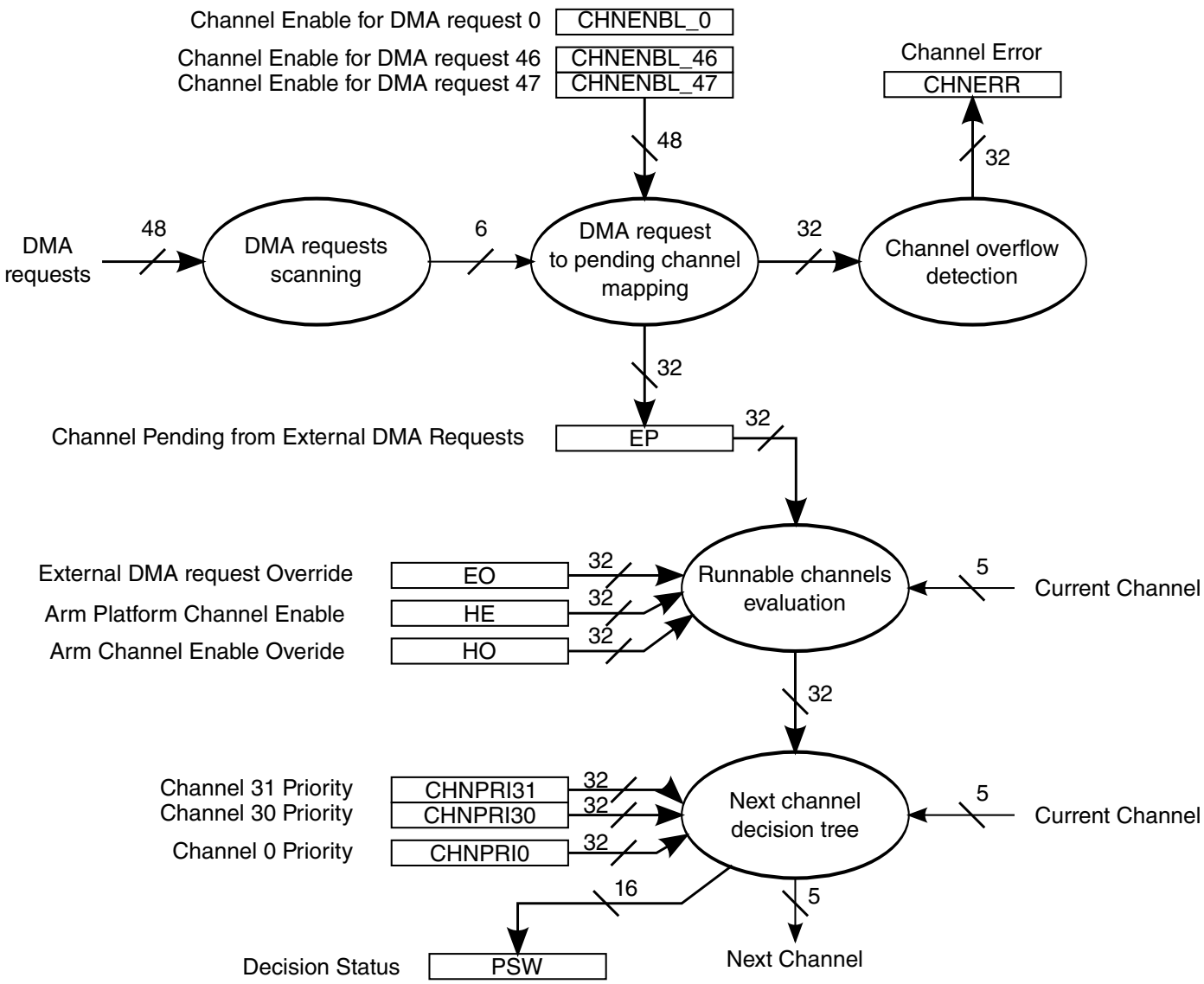


Figure 55-5. SDMA Hardware Scheduler

55.4.2.3.2 DMA Requests Scanning

The scheduler contains a 48-bit edge detection device that detects the rising edge of every DMA request and transmits the request number to the next stage.

The DMA requests are assumed to be generated on the same reference clock as the SDMA core clock; they are detected as soon as the signal goes from a 1-to-n-cycles low state to a 1-to-m-cycles high state.

This system is able to detect single-cycle pulses as well as level-based DMA requests such as a FIFO threshold crossing. In this case, the SDMA provides a memory mapped register that can be used by the channel script to monitor the DMA requests lines, and thus determines whether the data transfer is done or not done, and then continues with the transfer or closes the channel.

When several DMA requests are detected at the same time, they are forwarded to the next scheduler stage at the rate of one request per cycle. No request is lost.

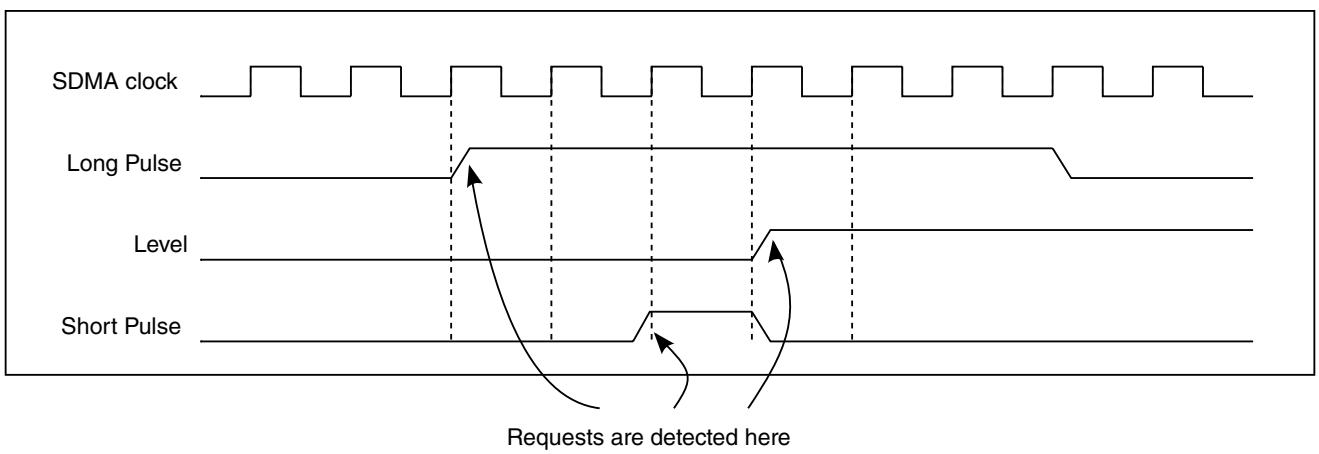


Figure 55-6. Examples of Valid DMA Requests

The DMA request inputs are connected to various sources that depend on the SoC. The exact list of DMA request inputs and their associated number is available in each respective project-specific chapter.

55.4.2.3.3 Mapping DMA Requests to Pending Channels

Whenever a DMA request is detected by the first stage, its number is used in the second stage to determine the channels that have to be activated.

This is performed with an array of 48 registers that are 32 bits wide: There are 48 Channel Enable Registers (CHNENBLn), one register per DMA request. The DMA request number selects the Channel Enable Registers, and every bit of this 32-bit register indicates that the corresponding channel must be activated when it is a 1.

This information is passed on the EP register. For every bit of the Channel Enable Register that is set, the corresponding bit of the EP register is also set, and the remaining bits of EP are left unchanged. The transformation of EP is summarized by the following equation:

EP = EP or CHNENBLn

The EP register is used to know which channels require service because they received a DMA request.

Typical contents of the CHNENBLn registers are all 0s, except for a single bit set. For example, a DMA request triggers one channel, but all 0s or several 1s are possible. One DMA request could activate several channels, and the channel execution sequence can be controlled by the channel priorities and numbers, as explained in the next sections. The following table illustrates an example configuration.

NOTE

From the table, the DMA request 0 is programmed to simultaneously trigger channels 0, 1, and 31. Also, DMA requests 30-47 are not used in this example. The remaining channels 2 to 30, are configured to be triggered by DMA requests 29 to 1, respectively.

Table 55-4. Channel Enable RAM Programming Example

DMA Request Number	Channel																																
	31																																0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table continues on the next page...

Table 55-4. Channel Enable RAM Programming Example (continued)

DMA Request Number	Channel																														
	31																														0
11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table continues on the next page...

Table 55-4. Channel Enable RAM Programming Example (continued)

DMA Request Number	Channel																														
	31																														0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

55.4.2.3.4 Channel Overflow

A channel overflow occurs when a DMA request requires service from channel *n* by setting bit *n* of the register EP, but this bit is already set, meaning channel *n* is already pending. This can come from an overrun/underrun condition.

This detection is possible only when the DMA requests are pulses, because a level-based DMA request stays high until it is serviced, even though an underrun or overrun condition occurs, thus preventing another edge detection of the DMA request.

The channel overflow information is saved in the 32-bit CHNERR register (1 bit per channel). You can configure the SDMA to trigger an interrupt to the Arm platform when there are 1s in CHNERR. Every bit of CHNERR is masked with the corresponding bit of INTRMASK and if it gives a 1, the corresponding bit of INTR is set, triggering the interrupt.

55.4.2.3.5 Runnable Channels Evaluation

The EP register is used in conjunction with several other 32-bit registers to determine the channels that are runnable.

Registers EO, DO, HO and HE, are controlled by the Arm platform. EP is controlled by the DMA requests and their mapping to channels.

Several channels may be runnable at any given time. The i^{th} channel is runnable if (and only if) the condition below is true:

(HE[i] or HO[i]) and (DO[i]) and (EP[i] or EO[i])

After reset, the HE[i], HO[i], EP[i], and EO[i] bits are all cleared whereas the DO[i] bits are all set. The functions associated with DO are not available for this device. When DO[i] is set, the scheduler condition becomes:

(HE[i] or HO[i]) and (EP[i] or EO[i])

The registers in these equations are controlled as follows:

- Arm platform (host) channel enable flag HE[i] may be set or cleared by the Arm platform with the HSTART and STOP_STAT registers. It can also be cleared by the i^{th} channel script.

  Typical usage is for the Arm platform to set this flag to activate the channel. The flag is cleared by the SDMA core when the transfer is done.

- Externally triggered channel pending flag EP[i] is set by the scheduler when the channel was activated by a DMA request. It can be cleared by the i^{th} channel script.
- The Arm platform channel override flag HO[i] may be set or cleared by the Arm platform. When set, it enables the i^{th} channel to run without the involvement of the Arm platform.

  Typical usage is for the Arm platform to set this flag for channels that do not need Arm platform supervision such as channels that are controlled by DMA request events (EP).

- DO should always be set to 1 so that the runnable channel evaluation considers only HO, HE, EP, and EO.
- Externally triggered channel override flag EO[i] may be set or cleared by the Arm platform. When set, it prevents the i^{th} channel from stopping and stalling on incoming peripheral DMA requests. This is the case when the channel is not handling data transfers with peripherals (for example, a memory to memory transfer).

The SDMA can clear the HE[i], and EP[i] bits by means of a done or notify instruction. The done instruction causes a reschedule; thus, enabling another channel to preempt the current one, while the notify instruction does not. The done and notify instructions can clear either HE[i] or EP[i] (never more than one at a time).

Table 55-5. Runnable Channel Selection Control

Register	Set by	Cleared By
HO	Write to HOSTOVR register	Write to HOSTOVR register
HE	Write to HSTART register	Write to STOP_STAT register or by the channel script with the done or notify instructions.
DO	Write to DSPOVR register	Write to DSPOVR register
EO	Write to EVTOVER register	Write to EVTOVER register
EP	Set by external DMA request event input.	By the channel script with the done or notify instructions

55.4.2.3.6 Next Channel Decision Tree

The next channel number is computed from the runnable channels list, the current channel number, and their respective priorities.

It is re-evaluated every cycle, but is only used when the current channel yields or terminates by executing a yield, yieldge, or done instruction.

The decision tree is based on the selection of the runnable channel that has the highest priority.

The highest priority channel is selected according to the following rules:

- Runnable channels are sorted by priority.
- If one of the channels with the highest priority had been preempted by a channel with a higher priority, but did not want to yield to a channel of the same priority (for example, it executed a yield, not a yieldge), it is elected as the next channel.
- The channels that belong to the highest priority group are sorted by their number and the channel that has the highest number in this group becomes the next channel. For example, if priorities are the same, channel 31 will be selected before channel 30.

When the current channel requires a reschedule with a yield(ge) or a done instruction, the context switch decision is based on the instruction parameter, the current channel number and priority, and the next channel number and priority. The possible cases are all listed in the following table. The grayed cells correspond to unusual cases that should not occur with a typical usage of the SDMA.

Table 55-6. Channel Switching Decision with a yield, yield(ge), or done

Instruction	Current Channel	Next Channel	Priorities Comparison	New Running Channel/Comments
yield (done 0)	Runnable	Not runnable	none	Current
	Runnable	Runnable	Current > Next	Current
			Current = Next	Current
			Current < Next	Next[1]
	Not runnable	Not runnable	none	none[2] (occurs when the channel was disabled by the Arm platform)
	Not runnable	Runnable	none	Next[1] (occurs when the channel was disabled by the Arm platform)
yieldge (done 1)	Runnable	Not runnable	none	Current
	Runnable	Runnable	Current > Next	Current
			Current = Next	Next[1]
			Current < Next	Next[1]

Table continues on the next page...

Table 55-6. Channel Switching Decision with a yield, yield(ge), or done (continued)

Instruction	Current Channel	Next Channel	Priorities Comparison	New Running Channel/Comments
	Not runnable	Not runnable	none	none[2] (occurs when the channel was disabled by the Arm platform)
	Not runnable	Runnable	none	Next[1] (occurs when the channel was disabled by the Arm platform)
done (done>1)	Not runnable	Not runnable	none	none[2]
	Runnable	Not runnable	none	Current[3] (occurs when the done instruction does not disable the channel runnable condition)
	Not runnable	Runnable	none	Next[1]
	Runnable	Runnable	none	Current[3] (occurs when the done instruction does not disable the channel runnable condition)

1. Current channel script execution is stopped, its context is saved; the next channel context is restored and its script execution resumes
2. Current channel context is saved and SDMA enters IDLE mode
3. Current channel context is saved, then restored, and the current channel script resumes execution

Finally, when the SDMA is in IDLE mode and a runnable channel is elected as the next channel, its context is immediately restored and the script execution resumes.

The *combinatorial-decision* tree supports dynamic modifications of the EP, EO, HE, HO, and DO flags as well as dynamic modifications of the channel priorities. The propagation times are detailed in Scheduler Pipeline Timing Diagram.

The decision tree status is available in the PSW register, which is continuously updated. It contains the next channel priority, the next channel number, the current channel priority, and the current channel number. When a priority is read as 0, it means the channel is not runnable.

A few examples of decisions are presented below:

- Channel 31 is running with priority 5, channels 13 and 24 are pending with the same priority 5; channel 24 is eligible as the next channel since 24 > 13.
- Channel 31 is running with priority 7, channels 13 and 24 are pending with priority 5; channel 31 is the next channel because its priority is greater than the other pending channels.
- Channels 7, 23, and 29 are pending with the same priority. Channel 7 is active and runs a yieldge; it is preempted by channel 29. After a period of time, channel 29 runs a yieldge, it is then preempted by channel 23 that is the selected channel since channel 29 is the current channel. Later, channel 23 runs a yieldge and is preempted

by channel 29. Channels 23 and 29 will go on switching after every yieldge until one of them terminates. It is only at that point that channel 7 becomes eligible again.

- Channel 11 is running with priority 3, and channel 15 is pending with priority 4. When the channel 31 script executes a yield instruction, it gets preempted by channel 15; then channels 6 and 18 with priority 3 become pending. Because channel 11 was preempted after executing a yield and there is no pending channel with a strictly greater priority, it is eligible as the next channel (although its number 11 < 18).

55.4.2.3.7 Scheduler State Diagram

The Figure 55-7 summarizes the behavior of the SDMA scheduler with details about the exact mechanism of the priority decision tree. It is important to understand the scheduler is a hardwired pipeline, which means all the stages are performed simultaneously every cycle, but a change on any given stage is reflected on the next stage after the delays presented in Scheduler Pipeline Timing Diagram.

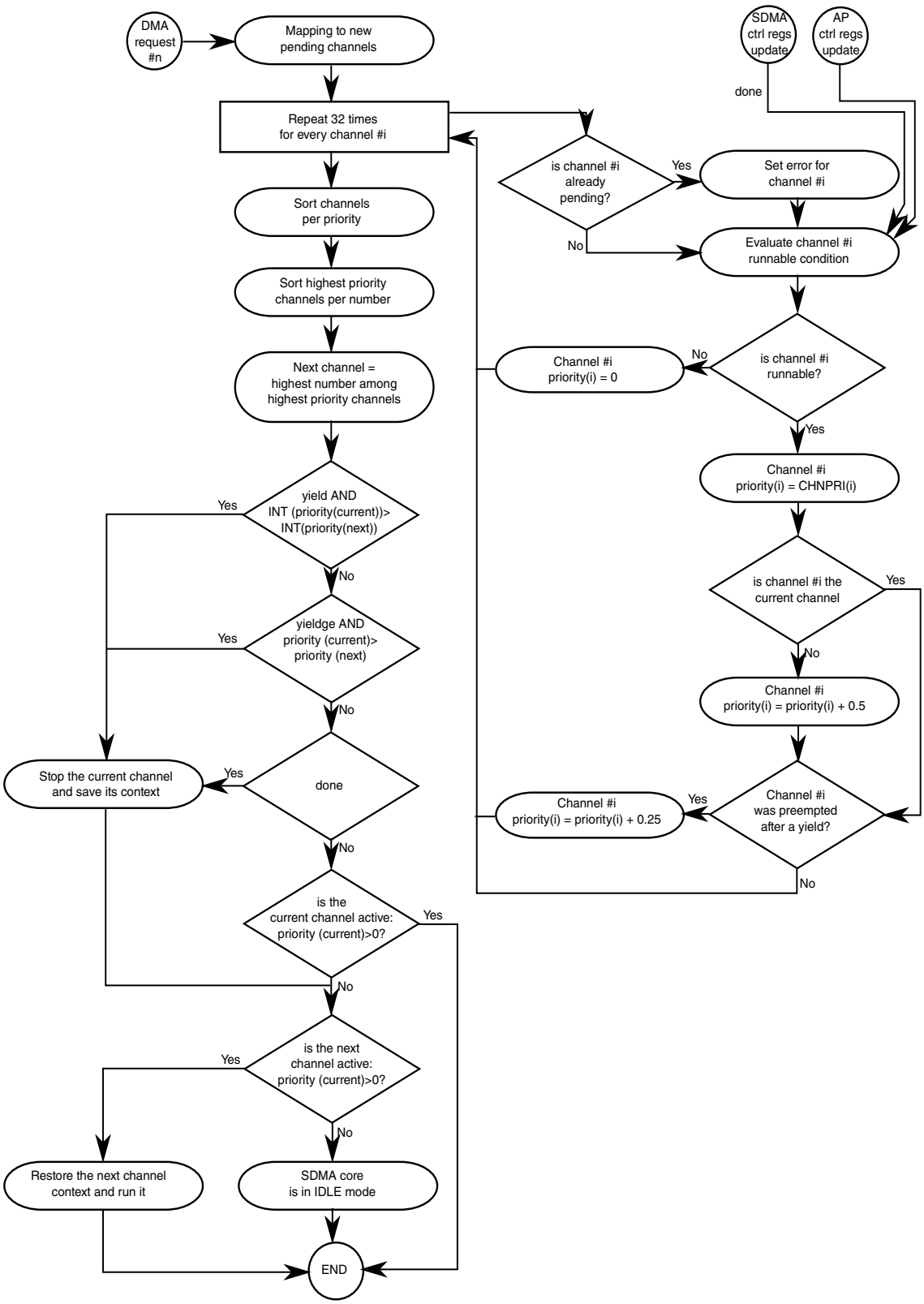


Figure 55-7. Scheduler State Diagram

55.4.2.3.8 Scheduler Pipeline Timing Diagram

The SDMA scheduler process of DMA-request and control-register modifications is not immediate.

The figure below shows the exact delays of all the tasks. The reference clock is the SDMA core clock.

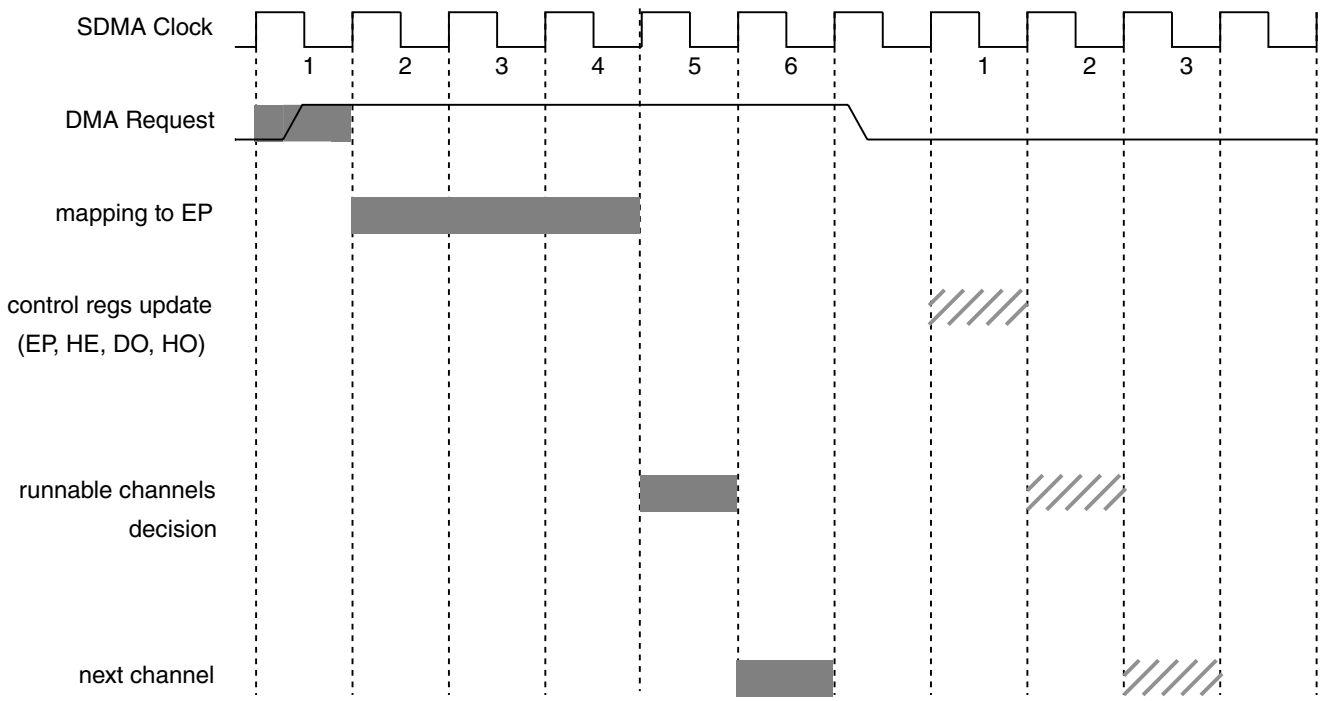


Figure 55-8. Scheduler Timing Diagram

Two numbers can be inferred from this timing diagram. First, it takes six SDMA core clock cycles to update the next channel from a DMA request. Second, it takes three SDMA core clock cycles to update the next channel from a direct modification of the condition registers (EP, DO, HE, or HO) by any processor. The processors that can modify these bits include SDMA with a done instruction or the Arm platform with a write access through the corresponding control port on their respective peripheral bus).

55.4.2.3.9 Channel-DMA Request Mapping

The 48 DMA request inputs to the SDMA scheduler are listed in project-specific chapters. Refer to the respective chapters for this information.

55.4.2.3.10 Examples: How to Start a Channel

A channel can be started when the following equation is true for channel *i*:

(HE[i] or HO[i]) and (DO[i]) and (EP[i] or EO[i])

Once this equation is true, the scheduler can start this channel according to the priority of all pending channels. Several examples of configuration are listed below:

1. To start a channel triggered by Arm platform software:
   - Initially, configure HO[i]=0, DO[i]=1, and EO[i]=1 using registers indicated in Table 55-5.
   - Arm platform software triggers the channel by writing to the HSTART register to set HE[i]=1, thereby setting the above equation true.
2. To start a channel triggered by DMA request event.
   - Initially, configure HO[i]=1, DO[i]=1, and EO[i]=0 using registers indicated in Table 55-5.
   - The DMA request is asserted to trigger the channel by setting EP[i]=1, which makes the above equation true.

55.4.2.4 Context Switching

On execution of a done or yield(ge) instruction, the current channel may be changed either because it has finished (which necessarily happens when the done instruction is executed), or it was preempted by a higher priority channel (which is possible but not systematic when the yield(ge) is executed).

Upon a channel change the SDMA goes through a context switch procedure.

When the current channel yields or ends, the context for that channel is saved into the context RAM locations for that channel. When the next channel starts running, its context is first restored from RAM.

Since context RAM is not yet initialized by reset, there will be no context restore at the beginning of the first channel (bootload channel) run after reset. It is expected that the bootload channel will be used to initialize the context for all other channels. When the bootload channel finishes running or yields, SDMA will enter its SAVE state and save that channel's context into RAM. Then, if the bootload channel is called again later, the context will be restored from RAM when the channel starts again.

The context structure for each channel is defined in Context Switching-Programming and Table 55-11. There will be one context area reserved for each channel. When a channel ends or yields, the SDMA core registers are automatically saved into the context RAM and later restored from the context RAM when the channel is next run. The total RAM space reserved for 32-channel contexts is either 3K or 4K depending on whether the SMSZ bit is set in the CHN0ADDR register, which enables an additional 8 words of scratch RAM for each context.

55.4.2.4.1 Context Switch Modes

The exact procedure to save the context of the old channel, and to restore the context of the new channel depends on the context switch mode selected by the Arm platform in the CONFIG control register.

The following are the context switch modes:

- By default, the "dynamic" context switch is set. This mode provides the most efficient context switch for an average of eight cycles to stop the current channel, save its context, restore the next channel context, and resume its execution. It consists of saving modified registers of the current channel in the background (for example, during the channel execution)-which leaves very few registers to save when the switch is decided-resuming execution of the next channel as soon as possible (for example, when the minimal set of registers is restored), and continuing the restore phase during this execution.
- In "dynamic with no loop" mode, the same principle is followed except the modified registers are only saved in the background when the loop flag is not set. This mode offers almost the same effectiveness as the previous one, but it prevents the system from accessing the RAM during loops to save power. This is the recommended mode for an efficient context-switch when the loop bodies are short.
- In "dynamic power" mode, no background saving is performed, which reduces power consumption to the minimum. The modified registers are only saved when the context switch starts. The restore phase is the same as before. This is the mode that achieves the optimal power consumption at the cost of a slower context-switch.
- In a "static" context switch, all the registers are saved when a context switch is decided, and all the registers are restored before starting the execution of the new channel. This mode enables a predictable behavior of the context switch since all the registers are restored prior to the channel start and all registers are saved after the channel termination.

NOTE

Static context mode should be used for the first channel called after reset to ensure that the all context RAM for that channel is initialized during the context SAVE phase when the channel is done or yields. Subsequent calls to the same channel or different channels may use any of the dynamic context modes. This will ensure that all context locations for the bootload channel are initialized, and prevent undefined values in context RAM from being loaded during the context restore if the channel is re-started later.

55.4.2.4.2 Context Switch Procedure

The Program Control Unit goes into the *save* state, the current context is spilled into memory, and the next channel context is restored according to the context-switch mode that was selected by the Arm platform.

The context switch procedure is as follows:

1. Load the current context's spill base address.
2. Spill the modified registers of the current channel to memory according to the selected context switch mode while the channel is running.

   On a done or yield(ge) that causes the channel preemption, the PCU goes into the *save* state. In *static* mode, all the registers are saved; whereas, in either *dynamic* mode, the registers that were modified but not yet saved are then saved, and the PCU registers and flags are finally saved.

3. Put the SDMA core into *sleep* and wait for new channels to be serviced. This step is skipped if there are pending channels when the current channel is saved.

   As soon as there is at least one pending channel, the PCU goes into its *restore* state to restore the context of the channel that was elected by the scheduler.

   Once a channel is elected, it remains the current channel until its script requests a rescheduling operation with a done or yield(ge) instruction. That means the current channel cannot be modified by the Arm platform, even if it is no more runnable or if its priority is modified.

   The Arm platform can however force a reschedule by writing the corresponding bit in the CONFIG register, which has the same effect as if the script had executed a done instruction. That feature should only be used to stop the SDMA in emergency cases.

4. Load the context base-address of the new channel.

   In "static" mode, all the registers are restored. In either "dynamic" modes, only the PCU registers are restored.

   The new channel is running. In "static" mode, no more activity regarding context restoring or saving is performed. In either "dynamic" modes, the registers are restored in the background every time an access to the context RAM is possible, and priority is given to restoring the registers that are required by the next instruction to be executed. When a register has not been restored and the next instruction needs it, this instruction gets stalled until the register was restored.

In "dynamic" and "dynamic with no loop" modes, background saving of dirty registers is performed every time an access to the context RAM is possible and allowed by the context switch mode.

NOTE

The contents of a channel context space in the context RAM depends on the selected context switch mode. In "dynamic" and "dynamic with no loop" modes, the contents of the context RAM tend to match the contents of the SDMA registers (except for the PCU registers and flags that are never saved in the background). In "dynamic power" and "static" modes, the contents of the context RAM remain unchanged until the channel terminates with a done or gets preempted.

55.4.2.4.3 Context Map in Memory

Refer to Context Switching-Programming.

55.4.3 Functional Units

The functional units are small systems that are used by the SDMA core to handle data transfers between the core and a bus domain external to the SDMA.

The SDMA core is able to control and exchange data with these systems by sending instructions and reading or writing data from/to the functional units' registers via the FUBUS. This is done with the ldf and stf instructions.

The following sections provide introductions to the available functional units. Functional Units Programming Model provides descriptions the functional units' behaviors.

55.4.3.1 Burst DMA Unit

The burst DMA unit enables the SDMA core to perform data transfers to and from the Arm platform memory.

It is optimized for accessing SDRAM-like devices. It does not provide control to assign a privilege level to the DMA access. The burst DMA unit provides the SDMA with means to do the following:

- Perform up to 8-beat read and write bursts to the Arm platform memory, which optimizes throughput when accessing SDRAM-type devices because of an internal, 36-byte FIFO
- Access the Arm platform memory at once or twice the SDMA core frequency
- Copy data from one Arm platform memory location to another Arm platform memory location at the Arm platform bus speed, which provides a very high throughput
- Control the method for addressing the Arm platform memory (automatic increment of addresses or frozen addresses-the former aimed at accessing RAM-like memory and the latter aimed at accessing single-address FIFOs)
- Enable or disable automatic prefetch when reading data from the Arm platform memory. When the prefetch mode is selected, the burst DMA automatically triggers external bursts to fill its FIFO without waiting for the SDMA core to request the corresponding data, greatly improving throughput.
- Rely on the DMA to automatically flush its FIFO content when there is enough data to generate an 8-beat burst to the Arm platform memory. Or, it forces a flush when a data transfer must terminate.
- In the former case, the SDMA core may only be stalled when it tries writing data and there is not enough room left in the FIFO. In the latter case, the core is stalled until the data is effectively written to the Arm platform memory.

  In automatic flush mode, the core receives an acknowledge that does not reflect the actual error status when the data is effectively written into the Arm platform memory. This error status is retrieved by a later access to the burst DMA. Terminating a write data transfer with a forced flush command guarantees that any bus error to the Arm platform memory is caught.

- Handle address alignment issues between the Arm platform memory map and the SDMA core data. This enables the core to read or write 32-bit data from the burst DMA, whereas the corresponding Arm platform address is not 32-bit aligned. This drastically improves the SDMA scripts' efficiency since the same loop that transfers 32 bits at a time can be used regardless of the start and end addresses in the Arm platform memory space.

This unit structure and registers are described in Burst DMA Structure and Burst DMA Registers.

55.4.3.1.1 Burst DMA Structure

The burst DMA is essentially made up of a 36-byte FIFO, address registers, and a controlling state-machine. The 36-byte FIFO enables eight-word buffering with address alignment, and the state-machine manages clock adaptation when required.

The burst DMA is depicted in the figure below.

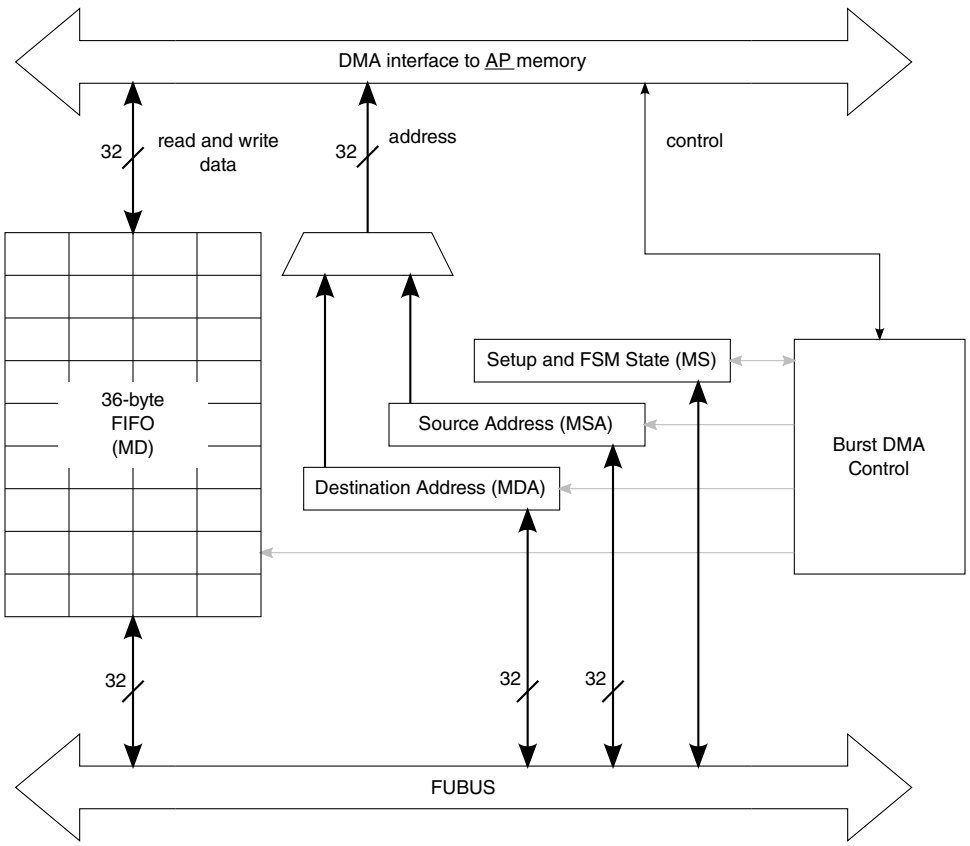


Figure 55-9. Burst DMA Structure

55.4.3.1.2 Burst DMA Registers

There are four registers, as follows, that may be accessed from the SDMA core:

- MSA (Memory Source Address) - Holds the source byte address in the Arm platform memory map for reading data from this location. This register is automatically modified every time the core reads new data from the FIFO.
- MDA (Memory Destination Address) - Holds the destination byte address in the Arm platform memory map for writing data to this location. This register is automatically modified every time the core writes new data into the FIFO.
- MD (Memory Data) - Labels the 36-byte FIFO access point: Reading a byte, halfword, or word from MD respectively retrieves the first 1, 2, or 4 bytes of the

FIFO (for example, the bytes that were stored first by the DMA state-machine when transferring data from the Arm platform memory).

- When the FIFO does not hold as many bytes as required by the SDMA core, the core is stalled until the missing bytes are read from the Arm platform memory. In the case of prefetch mode, the DMA controller decides when it should start a burst to Arm platform memory in order to reduce the risk to not have the required data for the future accesses of the core. When there is no prefetching, a burst is triggered when the required data is not available in the FIFO.

  Writing a byte, halfword, or word to MD stores 1, 2, or 4 bytes, respectively, at the end of the FIFO (for example, these bytes are transmitted to the Arm platform memory after all the other bytes that were previously stored in the FIFO). When the FIFO does not have enough room left to hold the written data, the SDMA core is stalled until a sufficient amount of FIFO contents are flushed out to the Arm platform memory. Flushing is decided by the DMA controller when there are enough bytes in the FIFO to perform the largest allowed burst to Arm platform memory (the exact size depends on the burst start address and the AHB 1 Kbyte boundary rule). However, the SDMA core has the ability to force the flushing operation at any time, for example, when at the end of the data transfer, prior to channel closure.

- MS (Memory Setup) - Contains the state of the burst DMA control, the two flags that define whether each address register is incremented after every access to the external memory, and another flag that is set when a bus error occurred.

55.4.3.1.3 Burst DMA Data Transfers

Three typical usages have been identified that involve the burst DMA: the data transfer startpoint, the endpoint, or both.

Every case requires a different procedure, as listed in the following sections:

55.4.3.1.3.1 Data Retrieval from the Arm platform Memory

The following steps retrieve data from Arm platform memory using the burst DMA unit:

- Set up the MS flags to reflect the mode for the source address (incremented or frozen according to the type of accessed device: memory or peripheral FIFO), then initialize the source address register itself (MSA).
- Read data from the FIFO using the *ldf MD* instruction as many times as needed. If an error occurred during the fetch from Arm platform memory, the DMA control tags the error status on the data and the SDMA core SF flag is set when reading this data from the FIFO.

55.4.3.1.3.2 Storing Data Into the Arm platform Memory

The following steps store data from Arm platform memory using the burst DMA unit:

- Set up the MS flags to reflect the mode for the destination address (incremented or frozen according to the type of accessed device: memory or peripheral FIFO), then initialize the destination address register itself (MDA).
- Store data into the FIFO using the *stf MD* instruction as many times as needed.
- When the transfer is finished and if the DMA worked in automatic flush mode, force the flush of the FIFO. This instruction is stalled until all the FIFO data is effectively sent to the Arm platform memory and the error status of the transfer is available in the DF flag.

55.4.3.1.3.3 Transferring Data Between Two Arm platform Memory Locations-Burst DMA Unit

The following steps copy data between two Arm platform memory locations using the burst DMA unit:

- Set up the MS flags to reflect the modes for the source and destination addresses (all the combinations are possible), then initialize the source address register (MSA) and the destination address register (MDA). Both addresses must be word-aligned.
- Use as many *stf MD* instructions with the *COPY* flag as needed. Every instruction triggers a burst read of a given number of words from the source address (this number is provided to the burst DMA via the SDMA core general purpose register, which is referenced in the *stf* instruction). Once all the data is loaded into the FIFO, the DMA empties it with a write burst of the same count to the destination address. The DMA acknowledges prior to instruction completion, which frees the SDMA core for other tasks at no delay cost.
- Once the transfer is done, there should be a final access to the burst DMA to check the error status.

55.4.3.2 Peripheral DMA Unit

The peripheral DMA unit is the second functional unit that connects the SDMA to the Arm platform memory.

Unlike the burst DMA, it does not support burst transfers and is optimized for accessing peripherals. It does not provide control to assign a privilege level to the DMA access. Its feature list comprises the following:

- Access to the Arm platform peripherals or memory at once or twice the SDMA core frequency

- Data copy from one Arm platform memory location to another Arm platform memory location at memory bus speed, improving throughput
- Control of the method for addressing the Arm platform memory (automatic increment or decrement of addresses or frozen addresses, the first ones aimed at accessing RAM-like memory and the last one aimed at accessing single-address FIFOs)
- Selectable automatic prefetch when reading data from the Arm platform memory. In prefetch mode, the peripheral DMA automatically fetches another data-without waiting for the SDMA core to request it-when its data register is empty, which improves the throughput
- Selectable automatic flush. In this mode, the SDMA core may only be stalled when it tries writing data and the previous write operation is not finished yet; whereas, in forced flush mode, the core is stalled until the data is effectively written to the Arm platform memory.
- In automatic flush mode, the core receives an acknowledge that does not reflect the actual error status when the data is effectively written into the Arm platform memory or the peripheral. This error status is retrieved by a later access to the peripheral DMA. Terminating a write data transfer with a forced flush command guarantees that any bus error to the Arm platform memory has been caught.

This unit structure and registers are described in Peripheral DMA Structure and Peripheral DMA Registers.

55.4.3.2.1 Peripheral DMA Structure

The peripheral DMA is made up of a 32-bit data register, two address registers, and a controlling state-machine. The state-machine manages clock adaptation, when required.

It is shown in the following figure.

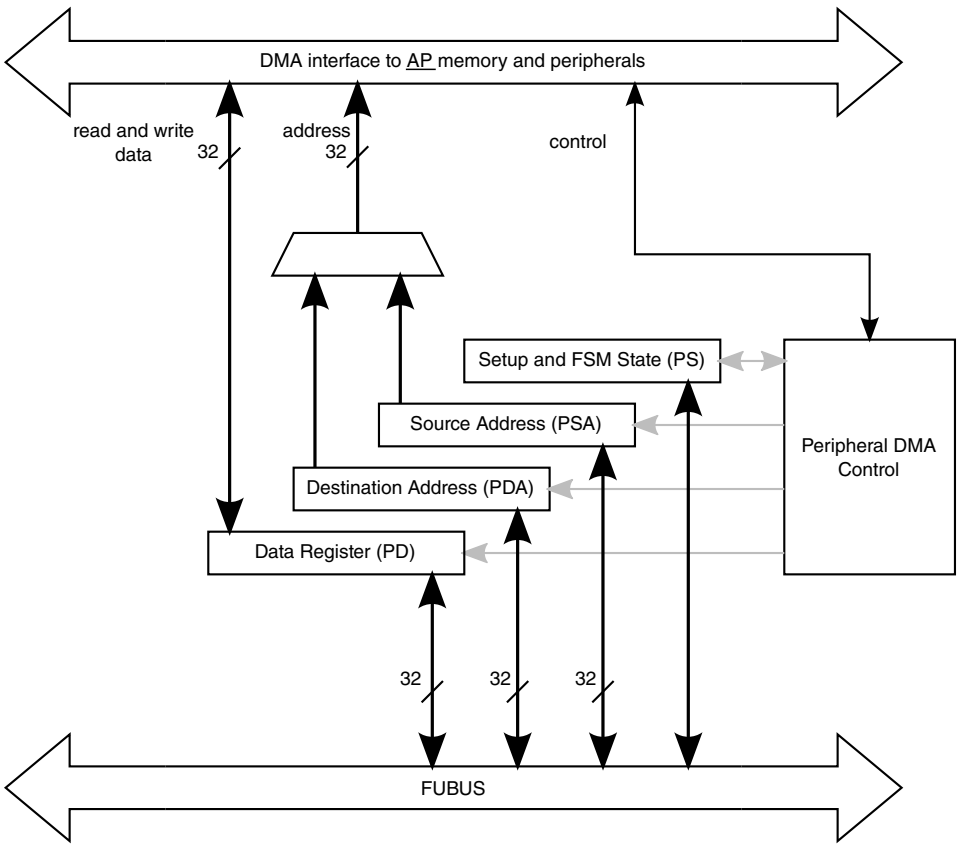


Figure 55-10. Peripheral DMA structure

55.4.3.2.2 Peripheral DMA Registers

According to Figure 55-10, the peripheral DMA has four registers that may be read or written by the SDMA core:

- *PD (Peripheral Data)* is the DMA 32-bit data register.
- *PSA (Peripheral Source Address)* holds the source byte address in the Arm platform memory map for reading data from this location. This register is automatically modified every time the core reads a new data from PD.

- *PDA (Peripheral Destination Address)* holds the destination byte address in the Arm platform memory map for writing data to this location. This register is automatically modified every time the core writes a new data into PD.
- *PS (Peripheral Setup)* contains the state of the peripheral DMA control, two configuration fields that define the way address registers are modified after every data access, two additional configuration fields that define the data size to access the source and destination devices, and another field that contains the latest transfer error status.

55.4.3.2.3 Peripheral DMA Data Transfers

There are three typical usages that involve the peripheral DMA, whether it is the data transfer start-point, endpoint, or both.

Every case requires a different procedure, as described in Data Retrieval from the Arm platform Memory or Peripheral, Storing Data into the Arm platform Memory or Peripheral, and Transferring Data Between Two Arm platform Memory Locations-Peripheral DMA Unit.

55.4.3.2.3.1 Data Retrieval from the Arm platform Memory or Peripheral

The following steps retrieve data from Arm platform memory using the peripheral DMA unit:

- Set up the PS fields to reflect the mode and data size for the source (incremented, decremented, or frozen address register; 8-bit, 16-bit, or 32-bit data transfers), then initialize the source address register itself (PSA) with an address that is aligned to the programmed data size.
- Read data from PD using the ldf PD instruction as many times as needed. If an error occurs during the fetch from the Arm platform memory or peripheral, the DMA control tags the error status on the data and the SDMA core SF flag is set when reading this data from PD.

55.4.3.2.3.2 Storing Data into the Arm platform Memory or Peripheral

The following steps store data to Arm platform memory using the peripheral DMA unit:

- Set up the PS fields to reflect the mode and data size for the destination (incremented, decremented, or frozen address register; 8-bit, 16-bit, or 32-bit data transfers), then initialize the destination address register itself (PDA) with an address that is aligned to the programmed data size.

- Store data into PD using the *stf PD* instruction as many times as needed.
- When the transfer is finished and if the peripheral DMA worked in automatic flush mode, force the flush of PD. This instruction is stalled until PD contents are effectively sent to the Arm platform memory or peripheral, and the error status of the transfer is available in the DF flag.

55.4.3.2.3.3 Transferring Data Between Two Arm platform Memory Locations-Peripheral DMA Unit

The following steps copy data between two Arm platform memory locations using the peripheral DMA unit:

- Set up the PS fields to reflect the modes and data size for the source and destination addresses (all the combinations of addressing modes are possible, but both data sizes must be identical), then initialize the source address register (PSA) and the destination address register (PDA). Both addresses must be aligned with the programmed data size.
- Use as many *stf PD* instructions with the *COPY* flag as needed. Every instruction triggers a single read from the source address; a single write of the received data immediately follows. The DMA acknowledges prior to instruction completion, which frees the SDMA core for other tasks at no delay cost.
- Once the transfer is done, there should be a final access to the peripheral DMA to check the error status.

55.4.4 SDMA Security Support

The SDMA provides support to SDMA software to block unauthorized updates to the scripts in RAM.

SDMA supports the following Security modes:

- Open Mode: has full control to load scripts and context into SDMA RAM. This is the default mode.
- Locked Mode: The Arm platform loads scripts and channel contexts at startup when it is still executing known safe software. When finished, it locks the SDMA to prevent further updates to RAM and selected registers. More details described in Locked Mode.

55.4.4.1 Locked Mode

The LOCK bit in the SDMA_LOCK register provides support for SDMA scripts to freeze RAM contents after the initial bootload routine to prevent future unauthorized updates to SDMA RAM.

After initial RAM contents are uploaded, Arm platform software can set the LOCK bit to secure the RAM contents to prevent future updates by an unauthorized. After the LOCK bit is written with a '1', the SDMA is "locked" until reset.

The LOCK bit can be read in the SDMA's internal memory map in the LOCK register (see Section SDMA LOCK (SDMAARM_SDMA_LOCK)). SDMA scripts which load information into RAM can check the value of the LOCK bit to determine if an upload to RAM is allowed. If not allowed, the script can refuse to allow the request to copy data into the RAM to continue. The exact use of the LOCK bit in SDMA scripts for security control will be described in SDMA software documentation (see SDMA Scripts).

While SDMA is locked, attempts to write to the SDMA_LOCK, CHN0ADR, ILLINSTADDR, and ONCE_ENB registers will be ignored. All registers remain readable. Writes to other registers are still allowed.

Once the SDMA is locked, the LOCK bit can only be cleared by a reset. A hardware reset will always clear the LOCK bit. A software reset initiated by writing to the RESET register will only clear the LOCK bit if the SRESET_LOCK_CLR bit in the SDMA_LOCK register is set. Since SDMA_LOCK register cannot be updated if SDMA is locked, the SRESET_LOCK_CLR bit must be configured before setting the LOCK bit. The SREST_LOCK_CLR bit will also be cleared by resets that clear the LOCK bit.

The SDMA RISC core uses the ILLINST and CHN0ADDR registers as pointers to determine where to jump to after an illegal instruction or upon boot after a reset. The LOCK bit prevents updates to these registers to protect against unauthorized changes to these pointers.

While SDMA is locked, the ONCE_ENB register cannot be written to prevent the OnCE under Arm platform control from being used to gain access to SDMA internal memory. If Arm platform control of the OnCE is enabled before setting the LOCK bit, the Arm platform can use the ONCE for debug purpose after LOCK is set.

55.4.5 OnCE and PCU Debug States

The SDMA has two different debug modes in which the OnCE performs debug instructions.

Refer to Figure 55-4 for an example of the PCU states in debug. The following are the two debug states:

- When a channel is running (that is, when CCR and CCPRI are different from 0, which can be read in the PSW register), SDMA can execute a SoftBkpt instruction from the channel script or receive a debug request. When either happens, the SDMA enters its "Classical" *Debug* state, which is described in OnCE and Real-Time Debug.
- When a channel is not running, the SDMA can be in *Sleep* state or in *Sleep after Reset* state. If a debug request is sent to the core, it enters its *Debug in Sleep* state. This debug mode works similarly to the "Classical" *Debug* state, except it returns to the original state (*Sleep* or *Sleep after Reset*) when the debug mode is left via the exec_core instruction of the OnCE. From this *Debug in Sleep* state, the SDMA can execute a program whereas no channel is running. If a new debug request is sent to the core or if a SoftBkpt is executed, it comes back to this *Debug in Sleep* state.

The OnCE is provided with several instructions that can be executed when the core is in either debug state. The following table summarizes the behavior of these OnCE debug instructions. There exists other secondary OnCE instructions that are described in OnCE and Real-Time Debug.

Table 55-7. SDMA in Debug Mode

Instruction	Debug	Debug in Sleep
exec_once	exec_once <instruction> SDMA executes the <instruction> and returns to the *Debug* state. The Program Counter (PC) is not incremented. This command must not be used with an instruction that modifies the PC value.	exec_once <instruction> SDMA executes the <instruction> and returns to the *Debug in Sleep* state. The Program Counter (PC) is not incremented. This command must not be used with an instruction that modifies the PC value.
run_core	run_core <instruction> SDMA executes the <instruction>, leaves the *Debug* state and continues executing the channel script from the position where it stopped. This command must not be used with an instruction that modifies the PC value.	run_core <instruction> SDMA executes the <instruction> and returns to its *Sleep* or *Sleep after Reset* initial state. This command must not be used with an instruction that modifies the PC value.
exec_core	exec_core <instruction> It is similar to run_core except it requires an instruction that changes the PC value (jump, branch...): the SDMA jumps to the new PC value, leaves the *Debug* state and starts executing instructions from this new PC value.	exec_core <instruction> If the previous state was *Sleep after Reset*, the SDMA returns to this state, and Chn0Addr value overrides the PC value. Otherwise, the SDMA jumps to the new PC value and starts executing instructions from this new PC.

NOTE

The feature exec_core in *Debug in Sleep* after *Sleep after Reset* was added for the Channel boot (channel 0) to allow the debugger to return to *Sleep after Reset* state with a new PC

value. The SDMA will be ready to boot at the Chn0Addr address.

55.4.6 SDMA Clocks and Low Power Modes

The SDMA receives several root clocks from the SoC clock controller block and performs adaptive clock gating to optimize its power consumption. From a user standpoint, clock gating and power mode selection are fully automatized inside the SDMA.

Root clock control is available from the SoC clock controller block.

There are numerous clock sources that are used in the SDMA. They belong to one of two possible clock domains listed in the following table, and have frequency constraints within each domain. Clocks are considered asynchronous between domains.

Within the Arm platform/SDMA clock domain, all clocks must come from the same DPLL. The Arm platform DMA interfaces (peripheral DMA and burst DMA) receive their clock from the Arm platform DMA clock source whose frequency can be once or twice the frequency of the SDMA core clock. The DMA interfaces are designed to work at the Arm platform DMA frequency, but the SDMA core is physically limited to a maximum 104 MHz frequency. Since this is lower than the maximum Arm platform DMA frequency, the SDMA core clock is tied to the Arm platform peripheral clock frequency.

The Arm platform Peripheral Bus Clock source must be an exact sub-frequency of the SDMA Core clock source (any integer value greater or equal to 1).

Table 55-8. Clocking Scheme

Clock Domain	Source Clock	Comments
Arm platform	SDMA core (SDMA main core)	Source clock for the core and all its operations; this clock is thus used by most of the SDMA sub-blocks.
	Arm platform DMA	DMA interface for the peripheral DMA and the burst DMA. It is balanced with the main clock source, and its frequency is either once or twice the main clock frequency.
	Arm platform peripheral	Connection to the Arm platform peripheral bus. It is a sub-frequency of the main clock frequency.
JTAG	TCK	Clock for JTAG access, limited to maximum of 1/8 of the SDMA core clock frequency.

The JTAG clock is sampled by the SDMA main clock to determine its rising edge. This simplifies design and clock management, but it also adds a ratio constraint between those two clocks. It is guaranteed the JTAG interface works properly when the frequency of TCK is lower than $1/8^{th}$ of the frequency of the SDMA main clock (which is about 8 MHz when the SDMA core clock frequency is 66 MHz).

55.4.6.1 Clock Gating and Low Power Modes

The SDMA automatically performs power saving without requiring user involvement. It implements two levels of automatic clock gating.

55.4.6.1.1 Coarse Clock Gating

Every sub-block clock comes from one of the five available sources, and is gated with the sub-block specific enabling condition.

The following table displays the sub-block clocks and their source. It also indicates the relationships that may exist between different sub-blocks clock enables.

Table 55-9. Sub-blocks Clocks

Sub-block	Source Clocks	Enabling Condition and Comments	Related Enabling Conditions
Core	SDMA Main Core	The core sub-block clock is running when the core is not in one of its sleep states (Sleep or Sleep after Reset) or there is a pending channel. Typically, the core sub-block clock is stopped once all the channels are processed and the core enters its sleep state. A new pending channel awakes the core sub-block clock.	None
Memories	SDMA Main Core	The clock activation only occurs during a core access.	Disabled when Core sub-block clock is disabled or no memory access in progress
Scheduler	SDMA Main Core	Its clock only runs when scheduling is needed: for example, when there are pending channels, upon reception of a DMA request, and anytime the Arm platform modifies the channel running conditions.	None
Arm platform Control	SDMA Main Core & Arm platform peripheral	The Arm platform peripheral clock is solely used to determine the frequency ratio with the SDMA main clock. The control registers' clock is based on *SDMA main clock*; it is active when the Arm platform or the SDMA modifies the contents of one of these registers.	None
Burst DMA	SDMA Main Core & Arm platform DMA	The burst DMA has two clocks: The first clock is derived from the SDMA main core clock and drives registers that are connected to the FUBUS. The second clock is derived from the Arm platform DMA clock and drives registers that are connected to the Arm platform DMA bus outside the SDMA. Both clocks are enabled	Disabled when Core sub-block clock is disabled

Table continues on the next page...

Table 55-9. Sub-blocks Clocks (continued)

Sub-block	Source Clocks	Enabling Condition and Comments	Related Enabling Conditions
		during active phases of data transfers (for example, these clocks are turned off when the burst DMA is not used by the running channel script).	
Peripheral DMA	SDMA Main Core & Arm platform DMA	The peripheral DMA has two clocks: The first clock is derived from SDMA main clock and drives registers that are connected to the FUBUS. The second clock is derived from the Arm platform DMA clock and drives registers that are connected to the Arm platform DMA bus outside the SDMA. Both clocks are enabled during active phases of data transfers (for example, these clocks are turned off when the peripheral DMA is not used by the running channel script).	Disabled when Core sub-block clock is disabled
OnCE	SDMA Main Core	The OnCE clock is derived from main source clock. It is disabled by default. In order to use the OnCE, its clock must be explicitly turned on, either by enabling the OnCE access from the Arm platform peripheral bus (register ONCE_ENB), or by driving the clk_gating_off input pin high. This is a SDMA input whose driver depends on the SoC implementation (typically a JTAG controller). The OnCE also receives the TCK input, which is the JTAG clock. It does not use it as a functional clock; the TCK input is sampled instead. Refer to Synchronization Implementation.	When enabled, all other clocks are systematically on (clock gating is off)

55.4.6.1.2 Refined Clock Gating

The SDMA implements a second level of clock gating on a register-per-register basis.

Unlike the first level that covers all the SDMA flip-flops, except the synchronizers (only five flip-flops are always running), the second level is only available for eligible registers, which amounts to about 90% of the SDMA flip-flops.

These gated registers are only clocked when the hardware logic detects a new data loading. This additional gating further reduces dynamic power consumption.

55.4.6.1.3 Low Power Modes and User Control

Power savings are automatically managed by the SDMA hardware without any user involvement; however, one can distinguish three different power modes: SLEEP, RUN, and DEBUG.

The following table describes these modes, and shows how to switch from one mode to another.

Table 55-10. Power Modes

Power Mode	Sub-blocks							Comments
	Core	Memories	Scheduler	Arm platform Control	Burst DMA	Peripheral DMA	OnCE	
SLEEP	off[1]	off	wait[2]	wait	off	off	off	Set when the PCU state is either *Sleep* or *Sleep after Reset* and the SDMA is not in DEBUG mode. This is the default mode after reset.
RUN	on[3]	wait	wait	wait	wait	wait	off	Set for the other PCU states that are reachable out of debug: *Program*, *Data*, *Change of Flow*, *Error in Loop*, *Debug*, *Functional Unit*, *Save,* or *Restore*.
DEBUG	on	on	on	on	on	on	on	Set regardless of the PCU state when clock gating is turned off to use the OnCE features (either *clk_gating_off* pin high or ONCE_ENB[0] set).

1. *off*: no clock
2. *wait*: only clocked when accessed or stimulated
3. *on*: clock is always running

It is possible to control the SDMA power mode. The procedures to force the SDMA into either mode are described in SLEEP Mode.

55.4.6.1.3.1 SLEEP Mode

This is the default mode after reset; therefore, resetting the SDMA forces this mode.

However, the common procedure is as follows:

- Ensure the *clk_gating_off* pin is low and ONCE_ENB[0] is cleared.
- Disable all channels (via the STOP_STAT control register, and the HO, DO, EO if necessary).
- Wait for the active channels to complete or force a reschedule via the reschedule bit in the RESET register.
- The SDMA is in SLEEP mode making it possible to completely shut off its clock from the chip level clock controller using the procedure described in Stop Mode Response.

55.4.6.1.3.2 RUN Mode

This is the default mode when a channel is running:

- Ensure the *clk_gating_off* pin is low and ONCE_ENB[0] is cleared.
- Activate at least one channel (via the HSTART control registers, a DMA request, and/or the HO, DO, EO register bits).

55.4.6.1.3.3 DEBUG Mode

The DEBUG mode must be set when one needs to use the debugging facilities of the SDMA.

- Ensure the SDMA clocks are running from the CCM.
- Set the *clk_gating_off* pin high or use the SDMA to set ONCE_ENB[0].

55.4.6.1.4 Stop Mode Response

The SDMA receives a stop request from the chip level clock controller. This request may be asserted when the chip enters the stop low power mode.

If the SDMA is running when the request is received, then the SDMA will complete all pending channels before returning to the SLEEP state. The SDMA sends an acknowledgement to the clock controller when the SLEEP state is entered indicating that the SDMAs clocks can be turned off.

55.4.6.2 Reset

After reset (either received from the reset block or a software reset required by the Arm platform), the SDMA is in IDLE mode. It will start its boot code located at address 0 once a channel is activated.

Activating a channel can be done by the Arm platform after programming a positive priority and setting the channel bit in the EVTPEND register.

There will not be a context RESTORE for the first channel (bootload channel) called after a reset because the context data in RAM has not been initialized. Static context mode should be used for the first channel called after reset to ensure that the all context RAM for that channel is initialized. Subsequent calls to the same channel or different channels may use any of the dynamic context modes

55.4.7 Software Interface

Appendix A fully describes the SDMA Application Programming Interface (API).

55.4.8 Initialization Information

This section discusses the following:

- Hardware Reset
- Channel Script Execution
- Initialization and Script Execution Setup Sequence

55.4.8.1 Hardware Reset

After reset, the program RAM, context RAM, data RAM, and RAM containing the channel enable registers (CHNENBLn) have unpredictable contents.

The active register set is assigned to channel 0 and the PC is initialized to all zeros. However, since the channel enable register is all zeros, there are no active channels and the SDMA is halted waiting for the boot channel to start.

The Arm platform will have to setup the SDMA in order to boot it. The CONFIG register must be initialized to determine the DMA/core clock ratio (1 or 2). Channel Enable Registers must also be initialized.

To start up the SDMA, the Arm platform first creates some channel control blocks (CCB) and buffer descriptors (BD) in Arm platform memory for the boot channel (channel 0) and then initializes the channel 0 pointer register (SDMA_MC0PTR) to the address of the first control block. Data Structures for Boot Code and Channel Scripts provides an overview of the data structure for the CCB and BD's. The SDMA_HSTART, SDMA_HOSTOVR and SDMA_EVTOVR registers are then configured according to Runnable Channels Evaluation to allow channel 0 to run.

Upon being enabled, the SDMA begins executing the script located at the address indicated by the Channel 0 Boot Address register (SDMA_CHN0ADDR) in the program memory. The reset value of SDMA_CHN0ADDR points to the default bootload script in ROM. This ROM script will read the channel 0 pointer register (SDMA_MC0PTR) to determine the location of the Channel Control Block (SDMA_CCB) in Arm platform memory. The script will then begin fetching by DMA the first channel control block which contains a pointer to the location channel 0 Buffer Descriptor chain which is also fetched via DMA. If the buffer descriptor contains a valid command, the script interprets the command in each buffer descriptor and proceeds to implement the command and move on to the next buffer descriptor control block. The buffer descriptor commands for

channel zero are typically set up to load SDMA's program RAM, Data RAM, and initial values for the channel contexts. Some channel scripts expect particular parameters to be passed

There are two ways to make the SDMA boot on a user-defined script. The OnCE (either via its JTAG interface or its Arm platform Control interface) can be used to download any code in the SDMA RAM and force the SDMA to boot on that code. Also, the SDMA_CHN0ADDR register in the Arm platform programming model can be modified to point to user code in RAM which would need to either have been loaded via the ONCE or default bootload routine (ex before a S/W reset).

55.4.8.2 Channel Script Execution

The execution of an SDMA script depends on both the instructions that make up the script, the data context upon which it operates, and commands or parameters allowed to the buffer. All these items must be initialized before the script is allowed to execute.

Each of the 32 channels has a separate context, but may share scripts and locations in data RAM.

Channels are initialized by the Arm platform by using channel 0 to download any required scripts and data values and the channels initial context. The context contains all the initial values of the SDMA core registers. This includes the Program Counter (PC) which is set to the start of the desired script in SDMA program memory.

The Arm platform selects which trigger conditions that must occur for the channel to start by configuring the SDMA_CHNENBL, SDMA_HOSTOVR and SDMA_EVTOVR registers. The trigger events include Arm platform setting HE (SDMA_HSTART) or a hardware DMA request asserts an event input to SDMA. The channel can become active according to its priority compared with other runnable channels when the selected trigger(s) cause the condition described in Runnable Channels Evaluation to evaluate as true.

The specific parameters to be passed to each script in the buffer descriptor or context are documented in the software documentation for each script. Please refer to SDMA Scripts for complete script documentation. Buffer Descriptor Format provides an overview of the buffer descriptor format.

55.4.8.3 Initialization and Script Execution Setup Sequence

To summarize, the following steps are minimally required to setup SDMA and run channel scripts.

- Perform Hardware Reset. The program RAM, context RAM, data RAM and SDMA_CHNENBLn registers have unpredictable contents after this reset.
- Initialize SDMA_CHNENBLn registers to map DMA request events to desired channels.
- Configure SDMA_CHNPRIn registers to select priority for runnable channels. A non-zero priority is required for the channel to run.
- Configure the SDMA_CONFIG register to select DMA to SDMA core clock ratio .
- Set up channel control blocks and buffer descriptors in Arm platform to specify the loading of SDMA program RAM and channel contexts for each SDMA channel to be used. Reference Data Structures for Boot Code and Channel Scripts.
- Configure SDMA_MC0PTR register with base address of Arm platform Channel Control Block base address.
- Initialize SDMA_CHNENBLn registers to map DMA request events to associated channel. Reference Mapping DMA Requests to Pending Channels.
- Configure SDMA_CHNPRIn registers to set priority for each channel to be run.
- For each channel to be run, configure SDMA_HOSTOVR (HO) and SDMA_EVTOVR (EO) registers to select which events (hardware and/or software trigger events) must occur for the channel to be runnable. Reference Runnable Channels Evaluation.
- Set bit 0 of the SDMA_HSTART register to set HE[0] and allow Channel 0 to run (assumes EO[0] andDO[0] were both set in previous step). This will cause SDMA to load the program RAM and channel contexts configured previously.
- Wait for Channel 0 to finish running. This is indicated by HI[0]=1 in the SDMA_SDMA_INTR register, or by optional interrupt to the Arm platform.
- Set the LOCK bit in the SDMA_SDMA_LOCK register to prevent un-authorized uploads of data to SDMA RAM.
- Additional channel scripts can now be run by enabling the selected software or hardware trigger event according to Runnable Channels Evaluation.

55.4.9 SDMA Programming Model

This section describes the programming model for the SDMA RISC engine, including its processor, memory, and internal control registers.

All addresses are related to the internal SDMA memory map, which is completely different from the Arm platform memory maps. The Arm platform processor has no access to any hardware resource described, except when those resources are described in Arm Platform Memory Map and Control Register Summary. .

55.4.9.1 State and Registers Per Channel

The SDMA can be seen as a set of 32 identical devices that are able to perform one data transfer channel each. Only one channel can work at a time, but every channel state is available at any time.

This chapter lists the components of every channel state.

55.4.9.2 General Purpose Registers

Each channel has eight general purpose registers of 32 bits for use by scripts. General register 0 has a dedicated function for the loop instruction, but otherwise can be used for any purpose.

55.4.9.3 Functional Unit State

Each channel context has some state that is part of the functional units.

The specific allocation of this state is part of the functional unit definition that is described in Burst DMA Unit Programming, Peripheral DMA Unit Programming .

This state must be saved/restored on context switches.

55.4.9.3.1 Program Counter Register (PC)

The PC is 14 bits. Since instructions are 16 bits in width and all memory in the SDMA is 32 bits in width, the low order bit of the PC selects which half of the 32-bit word contains the current instruction.

A low order bit of zero selects the most significant half of the word.[1]

55.4.9.3.2 Flags

Each channel has the following four flags:

- The T bit reflects the status of some arithmetic and test instructions. It is set when the result of an addition or a subtraction is zero and cleared otherwise. It is also the copy of the tested bits. Finally, it can also be set when the loop counter (GReg0) reaches zero. When the last instruction of the hardware loop is an operation that can modify the T flag, its effect on T is discarded and replaced by the GReg0 status.

1. For example, big-Endian.

- Two additional bits, SF and DF, are used to indicate error conditions resulting from loading data sources and storing to destinations, respectively. Access errors set these bits, and successful transactions clear them. They can also be cleared by specific instructions (CLRF and loop). The source fault (SF) is updated by the loads LD and LDF; the destination fault (DF) is updated by the stores ST and STF.
- Access errors are caused by several conditions including writing to the ROM, writing to a read-only memory mapped register, accessing an unmapped address, or any transfer error received by a peripheral when it is accessed.

  The SF and DF flags have a major impact on the behavior of the hardware loop: If SF or DF is set when starting a hardware loop and it is not masked by the loop instruction, the loop body will not be executed. Inside the loop body, if a load or store sets the corresponding SF or DF flag, the loop exits immediately. Testing the status of the T flag at the end of the loop (as well as testing both SF and DF) tells if the loop exited abnormally as any anticipated exit prevents GReg0 from reaching the zero value and thus setting the T flag. This is also valid if the fault occurs at the last instruction of the last loop.

- The last flag is the loop mode flag, LM, which is composed of two bits. The most significant bit indicates when the processor is currently operating in loop mode. It is set by the loop instruction and is cleared after execution of the last instruction of the last loop. The least significant bit is set when the program counter points to the last instruction of a loop on the last path. It is used for a channel that is restored with this configuration to know that the next program counter is EPC. As with the dynamic context switch Greg0, which indicates when the program must get out of the loop, it can be restored only on the last instruction of the loop. This, however, is too late to fetch the next instruction after the loop.

55.4.9.3.3 Return Program Counter (RPC)

The RPC is 14 bits. It is set by the jump to the subroutine instructions and used by the return from the subroutine instructions.

Instructions are available to transfer its contents to and from a general register.

55.4.9.3.4 Loop Mode Start Program Counter (SPC)

The SPC is 14 bits. It is set by the loop instruction to the location immediately following it.

55.4.9.3.5 Loop Mode End Program Counter (EPC)

The EPC is 14 bits. It is set by the loop instruction to the location of the next instruction after the loop.

55.4.9.4 Context Switching-Programming

Each channel has a separate context consisting of the eight general purpose registers and additional registers representing the state of the functional units.

The active registers and functional units contain the context of the active channel. The context of inactive channels is stored in SDMA RAM, which is part of the SDMA address space.

In a function of the selected context switching mode (Context Switching), modified registers by the program can be saved in the channel RAM space while the program is going on. In every cycle, a write access to the RAM is possible.

On a done or yield(ge) instruction, SDMA goes into "real" context switching. In one of the dynamic modes, modified registers not previously saved, as well as the PC-Loop registers, are stored into the context area of the channel that will be closed. The new PC-Loop registers are loaded from the context area of the new channel. All other registers are restored while the program is executed, giving priority to registers used by the decoded instruction. Therefore, in the best case, only the PC and Loop registers should be saved and restored during this context-switching phase, which only requires five SDMA cycles.

In static mode, the context switch stores all registers in the old channel RAM space, and restores all registers from the new channel RAM space. It requires 26 SDMA cycles.

The address of the context memory for channel *i* is *CONTEXT_BASE* + 24**i* or *CONTEXT_BASE* + *32*i w*here CONTEXT_BASE equals 0x0800. The table below presents the layout of a channel context in memory:

Table 55-11. Layout of a Channel Context in Memory for SDMA

OFFSET	31	30	29-16	15	14	13-0
0	SF	-	RPC	T	-	PC
1	LM		EPC	DF	-	SPC
2	GR0					
3	GR1					
4	GR2					
5	GR3					
6	GR4					
7	GR5					

Table continues on the next page...

Table 55-11. Layout of a Channel Context in Memory for SDMA (continued)

8	GR6
9	GR7
10	MDA (burst DMA)
11	MSA (burst DMA)
12	MS (burst DMA)
13	MD (burst DMA)
14	PDA (peripheral DMA)
15	PSA (peripheral DMA)
16	PS (peripheral DMA)
17	PD (peripheral DMA)
18	
19	
20	Reserved[1]
21	Reserved[1]
22	Reserved[1]
23	Reserved[1]
24	Scratch RAM (optional)
25	Scratch RAM (optional)
26	Scratch RAM (optional)
27	Scratch RAM (optional)
28	Scratch RAM (optional)
29	Scratch RAM (optional)
30	Scratch RAM (optional)
31	Scratch RAM (optional)

55.4.9.5 Address Space

The SDMA has four internal buses which are listed here.

- The Instruction bus reads instructions from the memory. Its address map is described in Instruction Memory Map.
- The Data bus (DMBUS) accesses the same memories as those visible on the Instruction bus, some memory-mapped registers (scheduler status and OnCE registers), and up to 14 peripherals. Its address map is described in Data Memory Map.

- The Functional Units bus (FUBUS) accesses the , Burst DMA, Peripheral DMA . The addressing mechanism is further detailed in Functional Units Programming Model.
- The Context Switch bus reads/writes registers into context-switch RAM space. It is a 64-bit bus dedicated for accessing this RAM space for updating the context of the running channel. While the program is going on, this bus has the lowest priority compared to the Instruction and Data buses, except for restoring a register needed for the decoded instruction to be executed. On the save part of a context switch (when the PCU is in its slave state), this is the only one used. On the restore part, the Instruction bus has the priority to read the next instruction at the restored PC and otherwise the Context Switch bus is used. It is not possible to control the actual data transfers that occur on this bus.

55.4.9.5.1 Instruction Memory Map

The instruction memory map is based on a 14-bit address bus and a 16-bit data (instruction) bus.

Instructions are fetched from either program ROM or program RAM. An SDMA script is able to change the contents of the program RAM, which is also visible from the data bus.

The first two instruction locations (at 0 and 1) are special. Location 0 is where the PC is set on reset. Location 1 is where the PC is set upon the execution of an illegal instruction. It is expected that both of these locations will contain a jmp to handle routines.

Table 55-12. SDMA Instruction Memory Space

Device	SDMA Address (Hex)	Base Address Label	Block Name	WS	Description
ROM	0x0000 ↓ 0x07FF	SDMA_IBUS_ROM_ADDR	-	0	4 Kbyte internal ROM with boot code and standard routines.
RAM	0x1000 ↓ 0x1FFF	SDMA_IBUS_RAM_ADDR	-	0	8 Kbyte internal RAM with channels context and user data/routines.

55.4.9.5.2 Data Memory Map

All of the data accessible to SDMA scripts make up the data memory space of the SDMA.

This address space has several components:

- ROM (also visible on the Instruction bus)
- RAM (also visible on the Instruction bus)

- Shared Peripherals Registers
- SDMA Internal Registers (scheduler, OnCE, and registers that are also accessible by the Arm platform)

SDMA scripts can read and write to the context RAM, data RAM, shared peripheral registers, and internal registers.

The address range is 16 bits and the data width is 32 bits. When accessing peripheral registers (USB and so on), the data width may be different. The exact address map for the peripherals depends on the project (as presented in each respective chapter).

Data access is performed with *ld* and *st* instructions that take the address from a general purpose register in the core (GReg*n*). The mapping between the general purpose register contents and the address bus is given in the following table:

Table 55-13. GReg*n* to DMBUS Address Mapping

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
sz	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
address															

Grayed bits are simply discarded but they must be cleared to ensure forward-script compatibility.

- sz (bit 31) indicates the peripheral data width: 0 is used for a 32-bit peripheral and 1 is used for a 16-bit peripheral.
- address (bits 15 down to 0) is the address of the accessed resource (internal memory, internal register, or shared peripheral).

Table 55-14. SDMA Data Memory Space

Device	SDMA Address (Hex)	Size	Description
ROM	0x0000 → 0x03FF	4 Kbyte	4 Kbyte internal ROM with boot code and standard routines
Reserved	0x0400 → 0x07FF	4 Kbyte	4 Kbyte Reserved
RAM	0x0800 → 0x0FFF	8 Kbyte	8 Kbyte internal RAM with channels contexts and user data/routines
per1	0x1000 → 0x1FFF	16 Kbyte	*peripheral 1* memory space (4 Kbyte peripheral's address space)
per2	0x2000 → 0x2FFF	16 Kbyte	*peripheral 2* memory space (4 Kbyte peripheral's address space)
per3	0x3000 → 0x3FFF	16 Kbyte	*peripheral 3* memory space (4 Kbyte peripheral's address space)
per4	0x4000 → 0x4FFF	16 Kbyte	*peripheral 4* memory space (4 Kbyte peripheral's address space)
per5	0x5000 → 0x5FFF	16 Kbyte	*peripheral 5* memory space (4 Kbyte peripheral's address space)
per6	0x6000 → 0x6FFF	16 Kbyte	*peripheral 6* memory space (4 Kbyte peripheral's address space)
Registers	0x7000 → 0x7FFF	16 Kbyte	Memory mapped registers
per7	0x8000 → 0x8FFF	16 Kbyte	*peripheral 7* memory space (4 Kbyte peripheral's address space)

Table continues on the next page...

Table 55-14. SDMA Data Memory Space (continued)

Device	SDMA Address (Hex)	Size	Description
per8	0x9000 → 0x9FFF	16 Kbyte	*peripheral 8* memory space (4 Kbyte peripheral's address space)
per9	0xA000 → 0xAFFF	16 Kbyte	*peripheral 9* memory space (4 Kbyte peripheral's address space)
per10	0xB000 → 0xBFFF	16 Kbyte	*peripheral 10* memory space (4 Kbyte peripheral's address space)
per11	0xC000 → 0xCFFF	16 Kbyte	*peripheral 11* memory space (4 Kbyte peripheral's address space)
per12	0xD000 → 0xDFFF	16 Kbyte	*peripheral 12* memory space (4 Kbyte peripheral's address space)
per13	0xE000 → 0xEFFF	16 Kbyte	*peripheral 13* memory space (4 Kbyte peripheral's address space)
per14	0xF000 → 0xFFFF	16 Kbyte	*peripheral 14* memory space (4 Kbyte peripheral's address space)

55.4.10 SDMA Initialization

Appendix A describes the setup of the SDMA . This section provides a quick description of several initialization procedures.

NOTE

There may be differences with the actual implementation in the API.

55.4.10.1 Hardware Reset-SDMA

After reset, the RAM that holds contexts, data, scripts, and the DMA request-channels matrix has unpredictable content.

The core registers are all reset to 0, including the PC; the PCU state is *Sleep after Reset*. No channel can be activated because all of the priorities are also reset to 0.

55.4.10.2 Standard Boot Sequence

The following is the standard boot sequence:

1. Initialize the CONFIG register-detailed in Configuration Register (SDMAARM_CONFIG)-to determine the Arm platform DMA/core clock ratio (1 or 2)
2. Initialize the DMA request-channels matrix (seeChannel Enable RAM (SDMAARM_CHNENBL*n*)).
3. Program the channel control registers-Channel Event Override (SDMAARM_EVTOVR), Channel BP Override (SDMAARM_DSPOVR), Channel

BP Override (SDMA_HOSTOVR), and Channel Event Pending (SDMAARM_EVTPEND)-according to the channel allocation.
4. Perform any necessary setup as required by the standard boot script in ROM (this is described in Appendix A).
5. Trigger channel 0 with the Channel Start (SDMAARM_HSTART) register, which starts the execution of the ROM script starting at address 0. This boot downloads channel scripts and contexts in RAM.

55.4.10.3 User-Defined Boot Sequence

The following is a user-defined boot sequence:

1. Initialize the Configuration Register (SDMAARM_CONFIG)Channel Enable RAM (SDMAARM_CHNENBL*n*), Channel Event Override (SDMAARM_EVTOVR), Channel BP Override (SDMAARM_DSPOVR), Channel Arm platform Override (SDMAARM_HOSTOVR), and Channel Event Pending (SDMAARM_EVTPEND).
2. Use the OnCE (either via its JTAG interface or its Arm platform control registers) to download any code in the SDMA RAM. Accessing the Memory describes how to write data to the RAM via the OnCE.
3. Use the OnCE instructions to make the PC default value point to the new boot script start address, or rely on the ROM startup script, which first jumps to the address in Channel 0 Boot Address (SDMAARM_CHN0ADDR). (This register default address points to the standard boot script.)

55.4.10.4 Script Loading and Context Initialization

The execution of an SDMA script depends on both the instructions that make up the script and the data context upon which it operates. Both must be initialized before the script is allowed to execute.

Each of the 32 channels has a separate data context, but may share scripts and locations in the data RAM.

The Arm platform manages the space in program RAM and data RAM. It also manages the assignment of SDMA channels to the device drivers that need them. Channels are initialized by the Arm platform via the channel 0 boot script. The boot channel downloads any required scripts with their data and the channels' initial contexts. Every context contains all the initial values of the registers, including the PC. Then the Arm platform can enable any channel that becomes active and begins fetching and executing instructions from its script.

55.4.11 Instruction Description

The following sections introduce the instruction of the SDMA.

Instruction set details are available in Instruction Set.

55.4.11.1 Scheduling Instructions

The following are scheduling instructions:

- done-The instruction causes certain scheduling or interrupt bits to be set or cleared, which may cause a change in the schedule-ability of the running channel. Then the instruction causes the SDMA to evaluate the current scheduling priorities and to choose the highest priority ready channel. If this channel is not the current channel, a context switch will take place. If there are no runnable channels, the SDMA will enter the stopped mode. The done 5 has a special usage reserved for debug, as explained in Debug Instructions.
- yield-These instructions are special cases of the done instruction. They do not modify the scheduling bits, but allow the highest pending channel (if it exists) to preempt the current channel if the pending channel priority is strictly greater than the current channel priority.
- yieldge-These instructions are special cases of the done instruction. They do not modify the scheduling bits, but allow the highest pending channel (if it exists) to preempt the current channel if the pending channel priority is strictly greater or equal to the current channel priority.
- notify-The notify instruction affects the scheduling bits, but does not cause rescheduling.

55.4.11.2 Conditional Branch Instructions

The conditional branch instructions of an 8-bit displacement, which is sign-extended and added to the current PC (which points to the next instruction) if the condition is satisfied.

Otherwise, control passes to the next sequential instruction.

- BF-Branch if False. The branch is taken if the T bit in the processor status is zero (false).
- BT-Branch if True. The branch is taken if the T bit in the processor status is one (true).

- BSF-Branch if Source Fault. The branch is taken if the SF bit in the processor status is one.
- BDF-Branch if Destination Fault. The branch is taken if the DF bit in the processor status is one.

55.4.11.3 Unconditional Jump Instructions

There are two varieties of unconditional control transfers: an absolute transfer and a through-register transfer.

Absolute transfers have a 14-bit address field that replaces the current PC.

- JMP-Jump. Causes the processor to jump to an absolute address encoded in the instruction itself.
- JSR-Jump to Subroutine. Causes the processor to jump to a subroutine, the address of which is encoded in the instruction itself.
- JMPR-Jump through Register. Causes the processor to jump to an absolute address contained in a General register. This instruction is meant to be used when more than one level of subroutines are required.
- JSRR-Jump to Subroutine through Register. Causes the processor to jump to a subroutine, the address of which is contained in a General register. This instruction is meant to be used when more than one level of subroutines are required.

55.4.11.4 Subroutine Return Instructions

The following are subroutine return instructions:

- RET-Return from Subroutine. The RET restores the contents of RPC to PC.
- LDRPC-Load from RPC to Register. THe LDRPC instruction is meant to be used when more than one level of subroutines are required. It stores the contents of RPC in any General register.

55.4.11.5 Loop Instruction

The following is a loop instruction:

LOOP-Enters Loop Mode. Before entering loop mode, the loop instruction can optionally clear the fault flags (SF and/or DF) based on a 2-bit field in the instruction. This feature is linked to the fact that setting SF or DF in loop mode will cause an immediate exit of the loop.

55.4.11.6 Miscellaneous Instructions

The following are miscellaneous instructions:

- CLRF-Clear Fault Flags. This instruction clears any combination of SF and DF.
- MOV r,s-This moves data from GReg[s] to GReg[r].
- LDI r,immediate-This loads GReg[r] with a zero-extended immediate value.

55.4.11.7 Logic Instructions

The following are logic instructions:

- XORr,s-This performs an exclusive or between GReg[r] and GReg[s], and stores the result in GReg[r].
- XORIr,immediate-This performs an exclusive or between GReg[r] and a zero-extended immediate value, and stores the result in GReg[r].
- ORr,s-This performs an or between GReg[r] and GReg[s], and stores the result in GReg[r].
- ORIr,immediate-This performs an or between GReg[r] and a zero-extended immediate value and, stores the result in GReg[r].
- ANDNr,s-This performs an and between GReg[r] and the negated GReg[s], and stores the result in GReg[r].
- ANDNIr,immediate-This performs an and between GReg[r] and the negated zero-extended immediate value, and stores the result in GReg[r].
- ANDr,s-This performs an and between GReg[r] and GReg[s], and stores the result in GReg[r].
- ANDIr,immediate-This performs an and between GReg[r] and a zero-extended immediate value, and stores the result in GReg[r].

55.4.11.8 Arithmetic Instructions

Arithmetic instructions modify the T bit in the processor status according to the result of the operation. The T bit is set if the result is zero, otherwise it is cleared.

- ADD r,s-This performs the addition of GReg[r] and GReg[s], and stores the result in GReg[r].
- ADDI r,immediate-This performs the addition of GReg[r] and a zero-extended immediate value, and stores the result in GReg[r].

- SUB r,s-This performs the subtraction of GReg[s] from GReg[r], and stores the result in GReg[r].
- SUBIr,immediate-This performs the subtraction of a zero-extended immediate value from GReg[r], and stores the result in GReg[r].

55.4.11.9 Compare Instructions

Compare instructions modify the T bit in the processor status according to the result of the operation. The T bit is set if the comparison is true, otherwise it is cleared.

NOTE

Only one version of the immediate form is implemented. Non-equality comparisons to immediate values will require two instructions.

- CMPEQ r,s-This sets T when registers GReg[r] and GReg[s] are equal.
- CMPEQIr,immediate-This sets T when register GReg[r] and the zero-extended immediate value are equal.
- CMPLTr,s-This sets T when register GReg[r] is less than and not equal to GReg[s]. The comparison is signed.
- CMPHS r,s-This sets T when register GReg[r] is greater than or equal to GReg[s]. The comparison is signed.

55.4.11.10 Test Instructions

Test instructions modify the T bit in the processor status according to the result of the operation. The T bit is set if any bit in the result is one, otherwise it is cleared.

- TSTr,s-This performs an and between GReg[r] and GReg[s], and sets T if the result is not zero.
- TSTIr,immediate-This performs an and between GReg[r] and a zero-extended immediate value, and sets T if the result is not zero.

55.4.11.11 Byte Permutation Instructions

These instructions shuffle the bytes in a register. For the purpose of describing these instructions, have the bytes in a register be numbered from the most significant as b_3, b_2, b_1, b_0.

- RORBr-The rotate right byte. The result is b_0, b_3, b_2, b_1.

- REVBr-The reverse bytes in word. The result is b_0, b_1, b_2, b_3.
- REVBLOr-The reverse, two low-order bytes. The result is b_3, b_2, b_0, b_1.

55.4.11.12 Bit Shift Instructions

The following are bit shift instructions:

- ROR1r-The rotate right 1 bit. This instruction does a circular right shift of 1 bit.
- LSR1r-The logical shift right 1 bit. This instruction shifts all bits to the right by 1. The high order bit is replaced by a 0.
- ASR1r-The arithmetic shift right 1 bit. This instruction shifts all bits to the right by 1. The high order bit is replaced by itself.
- LSL1r-The logical shift left 1 bit. This instruction shifts all bits to the left by 1. The low order bit is replaced by zero.

55.4.11.13 Bit Manipulation Instructions

- BCLRIr,n-The bit clear is immediate; clears bit number *i* in register *r*.
- BSETIr,n-The bit set is immediate; sets bit number *i* in register *r*.
- BTSTIr,n-The bit test is immediate; tests bit number *i* in register *r* (T becomes equal to the selected register bit).

55.4.11.14 SDMA Memory Access Instructions

All memory accesses are 32 bits.

Any memory location that is implemented with less than 32 bits (for example, peripheral registers) causes unimplemented bits to be read as 0s.

All memory accesses will cause either the SF or DF flags in the processor status to be set if they cause a fault.

What constitutes a fault, especially when accessing peripheral registers, is a property of the memory location.

- LDr,(b,d)-The load instruction creates an address by adding the displacement field (d) to the contents of the base register (b). The SDMA location at the resulting address is read and placed in the destination register (r).
- STr,(b,d)-The store instruction creates an address in the same manner as the load instruction. The register (r) is stored in the SDMA location at the resulting address.

55.4.11.15 Functional Unit Instructions

The functional unit instructions have an 8-bit field that is placed on the functional unit bus.

Some of these bits are used to select which functional unit should be involved in the transfer. The remaining bits are decoded by the selected functional unit so their specific use depends on the functional unit. See Functional Units Programming Model.

There are two functional unit instructions, as follows:

- LDFr,fub-The 8-bit field is placed on the functional unit bus and a read is issued to the selected functional unit. As a result of this instruction, the SF may be set in the processor status.
- STFr,fub-The 8-bit field is placed on the functional unit bus and a write is issued to the selected functional unit. As a result of this instruction, the DF may be set in the processor status.

55.4.11.16 Illegal Instructions

All instruction encodings that are illegal cause the following actions:

- The current PC (which points to one beyond the offending instruction) is put in the EPC register.
- The loop mode bit is cleared.
- The PC is set to the value stored in the Illegal Instruction Trap Address (SDMAARM_ILLINSTADDR) register (the default value is 0x0001).

ILLEGAL-Although any instruction other than those indicated in the SDMA specification will trigger the illegal instruction mechanism, the ILLEGAL instruction code is preferred as it will always be kept as *illegal* in the possible future versions of the SDMA core.

55.4.11.17 Debug Instructions

The following are debug instructions:

- SOFTBKPT-The software breakpoint instruction causes the core to stop and enter debug mode. The core can then be accessed and started by the OnCE debug block only.

- done 5-This instruction is used for debugging, as it copies the contents of the PCU registers and flags to the context memory. Information on this instruction is described in Saving the Context.
- CpShReg-This instruction copies the context memory into the PCU registers and flags. Modifying the corresponding memory location before executing this instruction enables you to have the channel continue from a new instruction address. This instruction is described in Restoring the Context.

55.4.12 Functional Units Programming Model

The functional unit instructions cause an 8-bit code, found in the low eight bits of the instruction, to be asserted on the functional unit control bus.

Some of these bits are used to select one of several functional units. Functional units which can be selected include SDMA registers such as MSA and MSD which are not mapped in the SDMA memory map, and are accessible only through the functional unit bus. These Functional Unit Registers are listed in the following table. In order to establish a programming convention, assume the selection bits are some number of the most significant bits of the 8-bit code. Furthermore, some number of the least significant bits is decoded by a given functional unit to establish the type of operation to perform.

Table 55-15. Functional Unit Registers

Functional Unit	Register	Register Name	Section/Page
Burst DMA Unit Programming	SDMSA	Memory Source Address Register	Memory Source Address Register (MSA)
	MDA	Memory Destination Address Register	Memory Destination Address Register (MDA)
	MD	Memory Data Buffer Register	Memory Data Buffer Register (MD) (Write) Burst DMA Write (stf) (Read) Burst DMA Read (ldf)
	MS	Memory State Register	State Register (MS)
Peripheral DMA Unit Programming	PSA	Peripheral Source Address Register	Peripheral Source Address Register (PSA)
	PDA	Peripheral Destination Address Register	Peripheral Destination Address Register (PDA)
	PD	Peripheral Data Buffer Register	Peripheral Data Register (PD) (Write) Peripheral DMA Write (stf)-Write Mode

Table continues on the next page...

Table 55-15. Functional Unit Registers (continued)

Functional Unit	Register	Register Name	Section/Page
			(Read) Peripheral DMA Read (ldf)-Read Mode
	PS	Peripheral State Register	Peripheral State Register (PS)

More information regarding the functional units can be found in Peripheral DMA Unit, and Burst DMA Unit.

55.4.12.1 Burst DMA Unit Programming

The DMA instructions control the DMA state machine and may cause a DMA cycle on the associated memory bus.

There are four registers associated with the burst DMA unit: a Memory Source Address register (MSA), a Memory Destination Address register (MDA), a Memory Data buffer (MD), and a state register (MS). The burst DMA has two different uses:

- A data transfer between External Memory Interface and SDMA general register
- A data transfer in copy mode where blocks of data are transferred from the source address to the destination address

55.4.12.1.1 Memory Source Address Register (MSA)

The source address register contains the pointer into EXTMC memory associated with the next read data transfer. It has byte granularity.

Reading the register with the ldf instruction has no side effects, and gives the address value in the EXTMC memory of the next data that is read by the SDMA during an ldf MD instruction.

Writing the source address register has two side effects: If the prefetch bit is set, a DMA read cycle (8-word read access) is issued with the new address. Any data still located in the buffer is lost. If there is valid write data in the buffer, it is necessary to force the DMA to completely flush it out before modifying MSA to guarantee all the data is effectively written to memory.

The MSA register has two modes of programming:

- Frozen-In frozen mode, the MSA register is not modified after DMA accesses.
- Incremented (default mode)-In incremental mode, MSA is incremented by the number of bytes transferred during read cycles.

55.4.12.1.2 Memory Destination Address Register (MDA)

The destination address register contains the pointer into EXTMC memory associated with the next write data transfer. It has byte granularity.

Reading the MDA register with the ldf instruction has no side effects. It gives the address value in the EXTMC memory where the next SDMA data (stf r,MD instruction) is stored when MD FIFO is flushed.

Writing the destination address register has one side effect. Any data still located in the buffer is lost. If there is valid write data in the buffer, it is necessary to force the DMA to completely flush it out before modifying MDA to guarantee all the data is effectively written to memory.

The MDA register has two modes of programming:

- Frozen-In frozen mode, the MDA register is not modified after DMA accesses.
- Incremented (default mode)-The MDA register is incremented by the number of bytes transferred during write cycles.

55.4.12.1.3 Memory Data Buffer Register (MD)

The data buffer register consists of a bank of 36 bytes that behave like FIFO.

This FIFO stores the eight words received when a read burst is triggered by the DMA (DMA is in read mode).

The MD register is in write mode after a writing in MDA or after an stf MD instruction.

In that case, a burst write access is automatically triggered when there are more than eight words in MD. For bandwidth optimization, any transfers between DMA and the EXTMC controller are based on burst accesses.

An ldf r,MD|SIZE instruction that reads the data buffer may cause a DMA cycle, as follows:

- If there are less bytes in the FIFO than the size parameter of the instruction. For instance, if only two bytes are available in MD and a 4-byte read is requested, a burst read access is executed to complete the two bytes.
- If the prefetch bit is set, and after reading there is enough space in the FIFO to store a full burst, a burst read access is triggered.

An stf r,MD|SIZE instruction that writes to the data buffer may cause a DMA cycle if the number of written bytes in MD is higher than 32 (eight words) or if the flush bit is set.

When DMA is used for data transfer between SDMA and EXTMC (reading or writing), no immediate error is possible because the block manages a data misalignment issue; therefore, it is allowed to read/write a word to/from a half-word address. However, the addresses (source or destination) must belong to the EXTMC memory mapping. The only potential error, in this mode, would be the error sent back by the EXTMC controller when an access to a super-user page is detected. The whole transfer on the DMA associated bus will be considered successful when there are no errors seen on the bus during the transfer. In copy mode, an immediate error could be returned to SDMA as described in Burst DMA Unit Error Management.

55.4.12.1.4 State Register (MS)

The state register contains the DMA state-machine value. It can be accessed in case of an error received during a transfer. MS is also accessed to set-up the conditional yielding feature.

The initialization value of this register is 0 and it consists of the following:

Table 55-16. SDMA_MS Structure

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	spriv	stype	0	0	dpriv	dtype
W																
R	0	0	0	0	y	d	e		0	0	n					
W																

Table 55-17. SDMA_MS Field Descriptions

Field	Description
31-22	Reserved
21 spriv	The spriv value is ignored for this device. 0 = valid value 1 = Reserved
20 stype	Source Mode. Indicates if MSA has to be incremented (or not) during accesses. 0 Frozen-MSA is not modified. 1 Incremented-MSA is incremented by the number of transferred bytes during read access.
19-18	Reserved
17 dpriv	The dpriv value is ignored for this device. 0 = valid value 1 = Reserved
16	Destination Mode. Indicates if MDA has to be incremented (or not) during accesses.

Table continues on the next page...

Table 55-17. SDMA_MS Field Descriptions (continued)

Field	Description
dtype	0 Frozen-MDA is not modified. 1 Incremented-MDA is incremented by the number of transferred bytes during write access.
15-12	Reserved
11 y	Conditional Yielding selector. When selected, theyield/yieldge instructions will not switch channels if the Burst DMA is in Write Mode, and it has less than four bytes in its FIFO. This is aimed at reducing the number of inefficient FIFO flushes due to context switches. 0 Always yields 1 Yields conditionally (when there are less than four bytes in the FIFO in write mode)
10 d	Access Direction or DMA Mode. DMA is in write mode when data was written into MD by stf MD instructions, or if a previous DMA cycle on the external bus was a write access. Writing MDA or MSA changes the DMA mode to the respective value. DMA is in read mode when a previous DMA cycle was a read access, and DMA stays in read mode when data is read by SDMA with an ldf MD instruction. Reading MDA or MSA does not change the DMA mode. 0 Read Mode 1 Write Mode
9-8 e	Error. Indicates if the previous access was acknowledged with a bus error. 00 No error was received. 01 *reserved* 10 Error mode 11 error read burst
7-6	Reserved
5-0 n	Number of bytes in the MD FIFO.

55.4.12.1.5 Burst DMA Write (stf)

When received from a stf instruction, the function code bits are interpreted as follows, depending on the addressed register:

Table 55-18. STF Code Bits

Register	7	6	5	4	3	2	1	0
MSA	s		p	freeze	r			spriv
MDA								dpriv
MD			f	cpy			sz	
MS								

Table 55-19. STF Code Bit Field Descriptions

Field	Description
7-6 s	Functional Unit selector 00 for Burst DMA
5 p (MSA)	Prefetch Flag 0 No prefetch 1 Prefetch required from new MSA
5 f (MD)	Forced Flush Flag 0 Automatic flush 1 FIFO contents are flushed (including the new written data).
4 freeze (MSA/MDA)	Address Freeze Mode 0 Address is normally incremented. 1 Address is frozen.
4 cpy (MD)	Copy Mode selection 0 Write Mode 1 Copy Mode
3-2 r	Register selection 00 MSA 01 MDA 10 MD 11 MS
1-0 sz (MD/MS)	Transfer Size 00 size 0 (no data stored in the FIFO) 01 byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)
0 spriv (MSA)	The spriv value is ignored for this device. 0 = valid value 1 = Reserved
0 dpriv (MDA)	The dpriv value is ignored for this device. 0 = valid value 1 = Reserved

The possible write instructions are listed in the table below (unused bits should always be cleared).

Table 55-20. Burst DMA STF Instruction List

Binary	Assembly	Comments
00_0_0_00_00	stf r,MSA	Writes content of the SDMA general register (r) to the source address register. MSA is in incremented mode.

Table continues on the next page...

Table 55-20. Burst DMA STF Instruction List (continued)

Binary	Assembly	Comments
00_0_1_00_00	stf r,MSA|FR	Writes content of the SDMA general register (r) to the source address register. MSA is in frozen mode.
00_1_0_00_00	stf r,MSA|PF	Writes content of the SDMA general register (r) to the source address register, and starts a read burst access. MSA is in incremented mode.
00_1_1_00_00	stf r,MSA|PF|FR	Writes content of the SDMA general register (r) to the source address register, and starts a read burst access.
00_0_0_01_00	stf r,MDA	Writes content of the SDMA general register (r) to the destination address register. MDA is in incremented mode.
00_0_1_01_00	stf r,MDA|FR	Writes content of the SDMA general register (r) to the destination address register. MDA is in frozen mode.
00_1_0_10_00	stf r,MD|SZ0|FL	No data transfers between the SDMA and MD, but all valid written data of the MD is flushed to the memory. An acknowledge or error is sent back to the SDMA core on transfer completion.
00_0_0_10_01	stf r,MD|SZ8	8-bit (byte) transfer to write buffer MD
00_1_0_10_01	stf r,MD|SZ8|FL	8-bit (byte) transfer to write buffer MD and flush after transfer. All valid written data of the MD is flushed to memory.
00_0_0_10_10	stf r,MD|SZ16	16-bit (half-word) transfer to write buffer MD
00_1_0_10_10	stf r,MD|SZ16|FL	16-bit (half-word) transfer to write buffer MD and flush after transfer. All valid written data of the MD is flushed to memory.
00_0_0_10_11	stf r,MD|SZ32	32-bit (word) transfer to write buffer MD
00_1_0_10_11	stf r,MD|SZ32|FL	32-bit (word) transfer to write buffer MD and flush after transfer. All valid written data of MD is flushed to memory.
00_0_1_10_00	stf r,MD|CPY	No data transfer between SDMA and MD but starts a copy transfer whose length is given by the 4 LSB of *r* register. (Maximum burst length is eight words.)
00_0_0_11_11	stf r,MS	32-bit (word) transfer to status register MS
00_0_0_11_00	stf r,MS|SZ0	Clears the error flag (if set). Other MS bits are unchanged; this instruction is also known as clref MS.

NOTE

When a flush bit is set, the SDMA flushes the FIFO including the newly written data. An acknowledge is sent to the core before the flush completes (except if size 0 is used). The goal of this flush bit is to force a flush, but it is recommended to use it only when needed (for example, when finishing a row of pixels during 2D data transfers). Indeed, if this bit is omitted and if there are more than 32 bytes in the FIFO, a burst write access is automatically triggered.

Since all the stf r,MD instructions (including the copy mode) acknowledge the SDMA core before the store is effective (except if size 0 is used), it is recommended to perform an ldf

from MS before terminating a channel in order to check the final error status. (The ldf from MS will stall the core until all the data was flushed out and the transfer status is known.)

After every stf MD instruction, the MDA is incremented by the number of bytes that are written in MD, except when it is programmed in frozen mode.

55.4.12.1.6 Burst DMA Read (ldf)

When received from an ldf instruction, the function code bits are interpreted as follows, depending on the addressed register:

Table 55-21. LDF Code Bits

Register	7	6	5	4	3	2	1	0
MSA	s				r			
MDA								
MD			p				sz	
MS								

Table 55-22. LDF Code Bit Field Descriptions

Field	Description
7-6 s	Functional Unit selector 00 for Burst DMA
5 p (MD)	Prefetch Flag 0 no prefetch 1 automatic prefetch
3-2 r	Register selection 00 MSA 01 MDA 10 MD 11 MS
1-0 sz (MD)	Transfer Size 00 reserved 01 byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)

The table below lists the possible write instructions (unused bits should always be cleared).

Table 55-23. Burst DMA LDF Instruction List

Binary	Assembly	Comments
00_0_0_00_00	ldf r,MSA	Copies the source address register value into an SDMA general register. It gives the memory address of the next data that will be read with an ldf MD instruction.
00_0_0_01_00	ldf r,MDA	Copies the destination address register value into an SDMA general register. It gives the memory address where the next incoming data will be flushed.
00_0_0_10_01	ldf r,MD|SZ8	8-bit (byte) read
00_1_0_10_01	ldf r,MD|SZ8|PF	8-bit (byte) read. If after this reading and the MD FIFO is empty, a burst read access at the MSA address is triggered.
00_0_0_10_10	ldf r,MD|SZ16	16-bit (half-word) read
00_1_0_10_10	ldf r,MD|SZ16|PF	16-bit (half-word) read. If after this reading, and the MD FIFO is empty, a burst read access at the MSA address is triggered.
00_0_0_10_11	ldf r,MD|SZ32	32-bit (word) read
00_1_0_10_11	ldf r,MD|SZ32|PF	32-bit (word) read. If after this reading and the MD FIFO is empty, a burst read access at the MSA address is triggered.
00_0_0_11_00	ldf r,MS	Copy the status register value into an SDMA general register.

NOTE

Read data is 0-extended before writing in the SDMA general registers. When reading the MD register, the DMA takes data from the FIFO if it is available. If part or whole data is not in the FIFO, an external burst read access is performed to provide the missing data. The SDMA is stalled as long as the required read data is not complete.

After every reading, MSA is incremented by the number of read bytes from MD FIFO, except when MSA is programmed in frozen mode.

55.4.12.1.7 Prefetch/Flush and Auto-Flush Management-Burst DMA Unit

The prefetch and auto-flush management enables the SDMA RISC machine to go on while a DMA access is performed.

When the RISC core requires a prefetch (p = 1) to the Burst DMA, it will receive an immediate transfer acknowledge before the DMA has finished the external access. This enables the RISC core to do other things like accessing another DMA machine.

The basic principle in prefetch mode is for the DMA to anticipate data reads from the SDMA RISC engine by fetching external bursts of data as soon as there is enough space in the DMA FIFO to store it. If ever the RISC engine required data that is not available in the FIFO, the read acknowledge is delayed until the data is available, but it does not have to wait until the burst completes.

The auto-flush basic principle is similar: An automatic flush is triggered every time there are eight words to be written in the FIFO. If the FIFO is full and the RISC engine requires another write, it is stalled until the burst has started and enough space was freed in the FIFO to store that new data. This means the SDMA RISC engine does not have to wait for the completion of a burst to receive its acknowledge and continue its processing.

In particular, an auto-flush is executed when DMA is in write mode and if the following is true:

- If the FIFO is empty and the first write is to a word-aligned address of any size (ex: the 2 LSB of MDA[1:0]= 0x0), the auto-flush is triggered immediately after the write of the 32'nd byte.
- If the FIFO is empty, and if MDA is an odd byte address (1, 3, 5, 7,...) and an stf MD|SZ8 is executed, the byte is flushed to memory. Once MDA increments to a word aligned address, the auto-flush will be triggered every 32 bytes.
- If the FIFO is empty, and if MDA is a half-word address (2, 6, 0xA,...) and an stf MD|SZ16 is executed, the two bytes of the incoming data are flushed to memory. Once MDA increments to a word aligned address, the auto-flush will be triggered every 32 bytes.
- If the FIFO is empty, and if MDA is not a word-aligned address (ex 1, 2, 3, 5, 6, 7, 9,...), and an stf MD|SZ32 is executed, the first 1 to 3 bytes will be flushed up to the next word aligned address. Afterwards, an auto-flush will be triggered each time the FIFO receives 32-bytes.
- Therefore, if an stf MD|SZ32 is executed with MDA equal to 0x1 and with an empty MD FIFO, the bytes located at addresses 1, 2, and 3 are flushed, and the byte located at address 4 remains in MD FIFO. This solves the misalignment issue. Additionally, the next write instructions (stf) complete the FIFO until it contains eight words; then a burst write is executed by the DMA to empty the FIFO. Protocol on the external bus does not support bursts of different data types (byte, half-word, or word).

  For example, consider the case where data is written using a byte access, stf MD|SZ8. The value of MDA during the very first byte write determines when the auto-flush will occur as follows:

- If MDA=0x0, the flush occurs following the write of byte 32
- If MDA=0x1, the flush occurs following the write of byte 1, byte 3 and byte 35.
- If MDA=0x2, the flush occurs following the write of byte 2 and byte 34.

- If MDA=0x3, the flush occurs following the write of byte 1 and byte 33.
- If MDA=0x4, the flush occurs following the write of byte 32

The flush command forces the DMA to flush all MD valid bytes to the EXTMC controller. An acknowledge is sent immediately to the SDMA, and any potential error is reported on a future access. It is thus essential to conclude a transfer with a last read from MS, which will stall the core until all data was flushed out and returned to the transfer status (acknowledge or error).

NOTE

During this kind of auto-flush (which occurs only at the beginning of a misaligned write transfer) no acknowledge is sent back to the SDMA, which is stalled until a flush is completed.

55.4.12.1.8 Data Alignment and Endianness-Burst DMA Unit

55.4.12.1.8.1 Burst DMA in Read Mode

For every read access to MD, the data returned to the SDMA core and the new FIFO state depends on the MSA status and the access size.

The FIFO is considered as a stack of 36 bytes: Data is fetched externally on a 32-bit bus, but the valid bytes only are stored in the FIFO and left-aligned (for a transfer of consecutive words, it is only the first word that may be truncated). The following table shows the FIFO byte alignment strategy and the corresponding MSA, the returned data, and the new FIFO state for any access size of an internal read from MD.

Table 55-24. FIFO Read Configuration

Before read		Internal read access size	Read data	After read	
MSA[1:0]	FIFO state			MSA[1:0]	FIFO state
00	x0 x1 x2 x3 y0 y1 y2 y3 z0 z1 z2 z3 and so on...	sz8	00 00 00 x0	01	x1 x2 x3 y0 y1 y2 y3 z0
		sz16	00 00 x0 x1	10	x2 x3 y0 y1 y2 y3 z0 z1
		sz32	x0 x1 x2 x3	00	y0 y1 y2 y3 z0 z1 z2 z3
01	x1 x2 x3 y0 y1 y2 y3 z0 z1 z2 z3 t0 and so on...	sz8	00 00 00 x1	10	x2 x3 y0 y1 y2 y3 z0 z1
		sz16	00 00 x1 x2	11	x3 y0 y1 y2 y3 z0 z1 z2
		sz32	x1 x2 x3 y0	01	y1 y2 y3 z0

Table continues on the next page...

Table 55-24. FIFO Read Configuration (continued)

Before read		Internal read access size	Read data	After read	
MSA[1:0]	FIFO state			MSA[1:0]	FIFO state
					z1 z2 z3 t0
10	x2 x3 y0 y1 y2 y3 z0 z1 z2 z3 t0 t1 and so on...	sz8	00 00 00 x2	11	x3 y0 y1 y2 y3 z0 z1 z2
		sz16	00 00 x2 x3	00	y0 y1 y2 y3 z0 z1 z2 z3
		sz32	x2 x3 y0 y1	10	y2 y3 z0 z1 z2 z3 t0 t1
11	x3 y0 y1 y2 y3 z0 z1 z2 z3 t0 t1 t2 and so on...	sz8	00 00 00 x3	00	y0 y1 y2 y3 z0 z1 z2 z3
		sz16	00 00 x3 y0	01	y1 y2 y3 z0 z1 z2 z3 t0
		sz32	x3 y0 y1 y2	11	y3 z0 z1 z2 z3 t0 t1 t2

55.4.12.1.8.2 Burst DMA in Write Mode

For every write access to the MD, the new FIFO state depends on the MDA status and the access size.

The FIFO is considered as a stack of 36 bytes: Data is stored in the FIFO according to the internal access size and the former MDA value. The following table shows the FIFO byte alignment strategy corresponding to MDA, as well as the new FIFO state for any access size of an internal write to MD.

Table 55-25. FIFO Write Configuration

Before write		Internal write access size	Written data	After write	
MDA[1:0]	FIFO state			MDA[1:0]	FIFO state
00	tt uu vv ww ?? ?? ?? ?? ?? ?? ?? ?? and so on...	sz8	?? ?? ?? x0	01	tt uu vv ww x0 ?? ?? ?? ?? ?? ?? ??
		sz16	?? ?? x0 x1	10	tt uu vv ww x0 x1 ?? ?? ?? ?? ?? ??
		sz32	x0 x1 x2 x3	00	tt uu vv ww x0 x1 x2 x3 ?? ?? ?? ??
01	tt uu vv ww	sz8	?? ?? ?? x0	10	tt uu vv ww

Table continues on the next page...

Table 55-25. FIFO Write Configuration (continued)

Before write		Internal write access size	Written data	After write	
MDA[1:0]	FIFO state			MDA[1:0]	FIFO state
	xx ?? ?? ?? ?? ?? ?? ?? and so on...				xx x0 ?? ?? ?? ?? ?? ??
		sz16	?? ?? x0 x1	11	tt uu vv ww xx x0 x1 ?? ?? ?? ?? ??
		sz32	x0 x1 x2 x3	01	tt uu vv ww xx x0 x1 x2 x3 ?? ?? ??
10	tt uu vv ww xx yy ?? ?? ?? ?? ?? ?? and so on...	sz8	?? ?? ?? x0	11	tt uu vv ww xx yy x0 ?? ?? ?? ?? ??
		sz16	?? ?? x0 x1	00	tt uu vv ww xx yy x0 x1 ?? ?? ?? ??
		sz32	x0 x1 x2 x3	10	tt uu vv ww xx yy x0 x1 x2 x3 ?? ??
11	tt uu vv ww xx yy zz ?? ?? ?? ?? ?? and so on...	sz8	?? ?? ?? x0	00	tt uu vv ww xx yy zz x0 ?? ?? ?? ??
		sz16	?? ?? x0 x1	01	tt uu vv ww xx yy zz x0 x1 ?? ?? ??
		sz32	x0 x1 x2 x3	11	tt uu vv ww xx yy zz x0 x1 x2 x3 ??

NOTE

If the FIFO mode changes from a write to a read mode, all remaining written bytes in MD are lost but no error is returned. Typically, this happens if an ldf MD is executed after stf MD instructions. Before a mode change, it is recommended to force the flush of a potential remaining byte by a stfMD|SZ0|FL instruction. In the same way, if a FIFO mode changes from a read to a write mode, all prefetched data present in the FIFO is lost and no error is returned.

55.4.12.1.8.3 Endianness-Burst DMA Unit

Big and Little Endian are supported by the Burst DMA, but data is always stored in MD in Big Endian.

Byte manipulation is performed when data is exchanged with an Burst controller (for example, during read or write burst accesses).

55.4.12.1.9 Burst DMA Unit Copy Mode

A mechanism is available to perform fast Arm-to-Arm transfers.

Data does not flow through the SDMA core: It is kept in the DMA FIFO. This mechanism is selected when writing MD with a special option in the instruction code (copy flag).

It is possible to transfer up to eight words in one SDMA instruction (this does not mean in one cycle). In this mode, every time an stf MD|CPY is executed, a read burst is executed and directly followed by a write burst transfer. Burst transfers are limited to eight words. The size of the transfer (in words)-given by the SDMA general register (4 LSB)-is also limited to eight. The following SDMA code shows how 100 bytes could be copied from the MSA address to the MDA address. This is sample code only.

Burst DMA copy mode example

```
          ldi r0,@src
          stf r0,MSA                          // Source address setup
          ldi r1,@dst
          stf r1,MSA                          // Destination address setup
          ldi r0,0x64                         // data transfer counter
          ldi r1,0x8
MAIN_XFER:
          cmphs r0,r1                         // Is r0 >= 0x8
          bf LAST_XFER                      // If not, jump to last transfer label
          stf r1,MD|CPY                        // Copy 8 words from MSA to MDA address.
          subi r0,0x8                         // Decrement counter
          jmp MAIN_XFER                     // return to main transfer loop
LAST_XFER:
          stf r0,MD|CPY
```

The main transfer loop is executed 12 times; then r0 equals 4 and the last transfer loop is run.

In this mode, an acknowledge is transmitted to the core as soon as the read burst can start; thus, a first copy instruction returns an immediate acknowledge and subsequent copy instructions will be acknowledged as soon as the previous copy has finished.

55.4.12.1.10 Burst DMA Unit Error Management

Another point to consider is the management of errors.

Because the DMA immediately sends an acknowledge to the RISC core (except for the stf MS|SZ0|FLS instruction), it assumes no error will occur. If an error occurs, it is flagged (transfer error acknowledge) for the following DMA access.

This should not be a problem if the DMA is used properly. The MD accesses are meant to stall the SDMA as little as possible to optimize throughput and hide calculation time. Therefore, final access to MS should be performed before closing a channel. This access waits until any pending operation is finished in the burst DMA and gather any remaining error.

In copy mode, an error could be immediately returned to the SDMA on execution of the ldf copy or stf copy instruction. It happens when MSA or MDA are not word addresses (for example, 0[4]). This is because copy mode must only be used for transferring a large packet of aligned data.

When an error is received during a *read* transfer to the external bus, which may occur during the burst accesses, the MD FIFO contains the valid beats of the burst, and the error flag of MS is set to 2'b11 (error read burst). It is possible to read MS ("n" field) to know how much valid data remains in MD and when MD is empty (after ldf instructions). The next read MD instruction sets the MS error flag to 2'b10 (error mode), and an error is sent back to the SDMA core. In error mode, it is possible to read MSA, which gives the address of the error data. Any attempt to read or write MD, or to modify MDA or MSA in error mode, gives rise to an error; therefore, an error flag must be reset by clearing MS at the end of the SDMA code section responsible for error management.

In "error read burst" mode, writing MDA, MSA, or MD, or starting a copy transfer by a stf MD|COPY instruction will cancel the error mode. The following table shows when an immediate error is sent back according to the executed instruction.

Table 55-26. Possibilities in ERROR READ BURST Mode

DMA Instruction	Immediate Error	Comments
stf rn, MD stf rn, MSA (|U |PF) stf rn, MDA stf rn,MD|COPY	NO	Error mode is reset. MSA, MDA, or MD are updated and a DMA cycle may start. For the stf MD|COPY, a copy loop is executed.
stf rn, MS	NO	MS is updated.

Table continues on the next page...

Table 55-26. Possibilities in ERROR READ BURST Mode (continued)

DMA Instruction	Immediate Error	Comments
ldf rn, MS ldf rn, MSA ldf rn, MDA	NO	MS, MSA, and MDA could be read in ERROR READ mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, MD	YES/NO	Immediate error if there is no more data available for read in the FIFO.

When an error is received during a *write* transfer, the error is reported to the next DMA access. In this case, an error is sent to the SDMA core and the DMA goes to its error mode. Reading MS gives the number of bytes that remain in MD; reading MDA gives the address of the error data. Any attempt to read or write MD, or to modify MDA or MSA in error mode, give rise to an error; therefore, an error flag must be reset by clearing MS at the end of the SDMA code section responsible for error management.

Table 55-27. Possibilities in ERROR Mode

DMA Instruction	Immediate Error	Comments
stf rn, MD stf rn, MSA stf rn, MDA	Yes	Any attempt to modify MD, MSA, MDA will raise an immediate error and burst DMA remains in error mode. When address registers are write-accessed, an error is returned.
stf rn, MS	No	This is the only way to exit error mode. MS[9:8] must be reset by an stf MS|SZ0 instruction.
ldf rn, MS ldf rn, MSA ldf rn, MDA	No	MS, MSA, and MDA could be read in error mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, MD	Yes	Whatever the DMA direction (read or write), an ldf rn triggers an immediate error.

55.4.12.1.11 Conditional Yielding-Burst DMA Unit

The standard SDMA transfer is based upon a hardware loop that has the following structure:

Hardware Loop

```
loop
load Rn,source            // can be ldf or ld
<computation>            // can be done through functional units
store Rn,dest              // can be st or stf
done 0               // yield
```

This structure needs to be kept independent of the functional units' particularities regarding the context switch. However, there can be variations in the context switch's efficiency, which can depend on the number of data received up to that point, and on the data itself.

The DMA, with its 8-word burst capability, has a preferable context switch period when its address register is 8-word aligned: It is the only moment that occurs once every eight loops when the succession of bursts is not broken by the context switch. When this is not the case, a context switch requires the storing (or loading) of less than eight words, which requires separate accesses and is far less efficient. The rest of the 8-word packet is stored (or loaded) after the context restore, and this is done as separate accesses.

The proposed solution is a conditional yielding, which occurs only when the DMA is in an optimum state. It does not require any modification to the scripts. The condition is decided at the DMA level.

The DMA can be programmed in two modes-conditional or always-true-for every channel, which provides complete flexibility. By default, the DMA is not in conditional mode.

The DMA condition is computed from the FIFO fill level and the various modes, as follows:

- When copy mode is selected, regardless of the transfer direction ('read' or 'write'), the condition is always true.
- In read mode, the condition is always true.
- In write mode, the condition is true when there are four bytes or less in the FIFO; it is false when there are more than four bytes. The 4-byte limit comes from the possibility of saving those bytes as MD with absolutely no impact on the bus accesses.

The aim at conditional yielding is to avoid splitting bus accesses (especially bursts).

55.4.12.2 Peripheral DMA Unit Programming

The peripheral DMA unit is connected to the Multi-Layer DMA Crossbar Switch of the Arm platform.

Its goal is to perform data transfers between any blocks connected to the DMA bus of this platform. These blocks are either peripherals or memories. The peripheral DMA could be seen as the Arm platform DMA controller.

The DMA performs data transfers in three modes:

- Read mode, where data is read from peripherals or from memory connected to the Arm platform and copied in a SDMA general register.
- Write mode, where data of a general register has to be written in a peripheral or a memory.
- Copy mode, where data is read from a peripheral (or memory) at a source address (PSA) and automatically written to a peripheral (or memory) at a destination address (PDA).

In copy mode, no SDMA general register is involved as transferred data only goes through the data register of the DMA.

The peripheral DMA has three addressing modes: frozen, incremented, and decremented, as follows:

- Frozen mode-When source or destination addresses are frozen, their value is not modified after a transfer. This mode is typically used for addressing peripheral FIFOs located at a fixed address.
- Incremented mode-When source or destination addresses are in incremented mode, after every transfer they are incremented by the number of bytes transferred.
- Decremented mode-In decremented mode, addresses are decremented by the number of bytes transferred.

The peripheral DMA registers are as follows:

- Two, 32-bit address registers (PSA and PDA) that respectively contain the source address for a read access and the destination address for a write access
- A 32-bit status register (PS) that contains information on the peripheral DMA configuration, such as the number of valid bytes in the data register, the error flag, the source and destination address mode, and so on.
- A 32-bit data register (PD) that stores data involved in a data transfer

55.4.12.2.1 Peripheral Source Address Register (PSA)

The source address register contains a pointer to a source peripheral or a memory associated with the next read data transfer. It has byte granularity.

It is based on the following:

- A 32-bit register (PSA) to store the address value
- A 2-bit register (stype) to store the source address mode (frozen, incremented, or decremented)
- A 2-bit register (ssize) to store the source target data path size (byte, half-word, or word)

Reading the register with the ldf instruction has no side effects and gives the address value of the next data that will be read by the SDMA during an ldf MD instruction. Writing the source address register may have side effects. If there is valid write data in the data register and the source address is changed, the write data is discarded. If the prefetch bit is set, a DMA read cycle is issued with the new address.

When PSA is to be written, you must specify the source target address mode, providing its size (byte, half-word, or word). This enables omission of the size field in all ldf MD instructions. When DMA performs a read cycle, its size is given by the value of the PSA source size register (ssize). If source is a memory in incremented mode, first programmed in word mode (stf PSA|SZ32|I), and if an SDMA script needs to read bytes from this memory, the size of the source target must be updated before executing new accesses. The source address mode and its size are given by labels added to the stf PSA instruction as described in the write section. The ssize and stype registers are part of the DMA status register (PS).

Writing to PSA may issue an immediate error if the source size is not compatible with the value to be written into the PSA register. For instance, writing a 2 in PSA and specifying that it is memory-accessed in word mode creates an immediate error.

55.4.12.2.2 Peripheral Destination Address Register (PDA)

The destination address register contains a pointer to a source peripheral or a memory associated with the next write data transfer. It has byte granularity.

It is based on the following:

- A 32-bit register (PDA) to store the address value
- A 2-bit register (dtype) to store the destination address mode (frozen, incremented, or decremented)
- A 2-bit register (dsize) to store the destination target data path size (byte, half-word, or word)

Reading the register with the ldf instruction has no side effects, and gives the address value of the next data that will be written by SDMA during an stfMD instruction. Writing the destination register has no side effect. Similar to the PSA register, the destination address mode and source are specified in the stf PDA instruction and may also generate an error in case of incorrect programming.

55.4.12.2.3 Peripheral Data Register (PD)

The data register of the peripheral DMA is a 32-bit register. When the destination address is correctly set up, any writing to PD will automatically flush the new input data.

The number of SDMA bytes that will be transferred is given by the PDA size register. Unlike other SDMA DMAs, PD is not a FIFO: It is not used to accumulate bytes that from the SDMA and must be packed before being sent to external memories. In read mode, and if the source address is correctly set up, an ldf instruction will empty PD. If a prefetch is required along with the instruction, the DMA will initiate a new read transfer.

Reading PD in prefetch mode only stalls the SDMA when the prefetched data is not yet available. Writing PD only stalls the SDMA if the previous write operation was not completed. As soon as the previous operation is over, the acknowledge is sent back to the SDMA RISC engine.

An error flag-part of PS-is set when an external access fails. The error is thus reported to the next SDMA instruction that involves the peripheral DMA.

55.4.12.2.4 Peripheral State Register (PS)

The state register contains the DMA state-machine value. It can be accessed in case of an error received during a transfer.

Although all PS fields can be written by an stf instruction, it is recommended to access only the error bit (to reset it). Modifying other PS fields will provide an un-guaranteed DMA behavior.

The initialization value of PS is 0, and it consists of the following structure:

Table 55-28. PS Structure

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	ssize		stype		dsize		dtype	
W																
R	0	0	0	0	0	d	e		0	0	0	0	0	n		
W																

Table 55-29. PS Field Descriptions

Field	Description
31-24	Reserved
23-22 ssize	Source Target Size. Determines the size of the read transfers on the external bus. It should match the accessed device characteristics. 00 *reserved* 01 Byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)

Table continues on the next page...

Table 55-29. PS Field Descriptions (continued)

Field	Description
21-20 stype	Source address Mode. Determines whether PSA is incremented, decremented, or kept unmodified after every read from the external bus. 00 Frozen Mode 01 Incremented Mode 10 Decremented Mode 11 *reserved*
19-18 dsize	Destination Target Size. Determines the size of the write transfers on the external bus. It should match the accessed device characteristics. 00 *reserved* 01 Byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)
17-16 dtype	Destination address Mode. Determines whether PDA is incremented, decremented, or kept unmodified after every write on the external bus. 00 Frozen Mode 01 Incremented Mode 10 Decremented Mode 11 *reserved*
15-11	Reserved
10 d	Direction Flag or DMA Mode. DMA is in write mode when data was written into PD by stf PD instructions, or if a previous DMA cycle on the external bus was a write access. Writing PDA or PSA does not change the DMA mode. DMA is in read mode when a previous DMA cycle was a read access, and DMA stays in read mode when data is read by the SDMA with an ldf PD instruction. Reading PDA or PSA does not change the DMA mode. 0 Read Mode 1 Write Mode
9-8 e	Error. Indicates if the previous access was acknowledged with a bus error. 00 No error was received. 01 *reserved* 10 Error mode 11 Error read
7-3	Reserved
2-0 n	number of bytes in PD

NOTE

dtype, dsize, stype, and ssize are updated when PSA and PDA are written.

55.4.12.2.5 Peripheral DMA Write (stf)-Write Mode

When written by an stf instruction, the function code bits are interpreted as follows:

Table 55-30. STF Code Bits

Register	7	6	5	4	3	2	1	0
PSA	s		p	ar	am		sz	
PDA								
PD			pdsel					
PS			pssel					

Table 55-31. STF Code Bits Field Descriptions

Field	Description
7-6 s	Functional Unit selector 11 for Peripheral DMA
5 p (PSA)	Prefetch Flag 0 no prefetch 1 automatic prefetch
4 ar (PSA/PDA)	Address Register Selector 0 PSA 1 PDA
3-2 am (PSA/PDA)	Address Mode. Determines how PSA or PDA is modified after every read or write access to the PD. 00 Frozen-Address registers are not modified after the transfer. 01 Incremented-Address registers are incremented by the number of transferred bytes. 10 Decremented-Address registers are decremented by the number of transferred bytes. 11 Updated-PSA and PDA are not modified. Either address mode is not modified, but the width of the data path is updated by the sz field.
1-0 sz	Transfer Size 00 *reserved* 01 byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)
5-0 pdsel	PD access selector 001000 is the only valid option
5-0 pssel	PS access selector 111111 writes to PS 001100 only clears the error flag in PS

Due to the large number of possible stf instructions, the following table provides only a short list of all the possible write instructions:

Table 55-32. Peripheral DMA STF Instruction List

Binary	Assembly	Comments
11_00_00_01 11_00_00_10 11_00_00_11	stf Rn, PSA|SZ8 |F stf Rn, PSA|SZ16|F stf Rn, PSA|SZ32|F	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is frozen.
11_10_00_01 11_10_00_10 11_10_00_11	stf Rn,PSA|SZ8 |F|PF stf Rn,PSA |SZ16|F|PF stf Rn,PSA |SZ32|F|PF	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • 1, 2, or 4 bytes are *fetched* from the peripheral source. • Source address is frozen.
11_00_01_01 11_00_01_10 11_00_01_11	stf Rn, PSA|SZ8 |I stf Rn, PSA|SZ16|I stf Rn, PSA|SZ32|I	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA + 1,2 or 4 after read PD.
11_10_01_01 11_10_01_10 11_10_01_11	stf Rn, PSA|SZ8 |I|PF stf Rn, PSA|SZ16|I|PF stf Rn, PSA|SZ32|I|PF	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA + 1, 2, or 4 after read PD. • 1, 2, or 4 bytes are *fetched* from the peripheral source.
11_00_10_01 11_00_10_10 11_00_10_11	stf Rn, PSA|SZ8 |D stf Rn, PSA|SZ16|D stf Rn, PSA|SZ32|D	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA-1,2, or 4 after read PD.
11_10_10_01 11_10_10_10 11_10_10_11	stf Rn, PSA|SZ8 |D|PF stf Rn, PSA|SZ16|D|PF stf Rn, PSA|SZ32|D|PF	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA-1,2, or 4 after read PD. • 1, 2, or 4 bytes are *fetched* from the peripheral source.
11_00_11_01 11_00_11_10 11_00_11_11	stf Rn, PSA|SZ8 |U stf Rn, PSA|SZ16 |U stf Rn, PSA|SZ32 |U	• *Update* source pointer to memory, which becomes a pointer to a memory accessed in byte, half-word, or word. • PSA value is not modified by Rn. • Bytes present in PD are lost.
11_10_11_01 11_10_11_10 11_10_11_11	stf Rn, PSA|SZ8 |PF|U stf Rn, PSA|SZ16 |PF|U stf Rn, PSA|SZ32 |PF|U	• *Update* source pointer, which becomes a pointer to a target accessed in byte, half-word, or word. • PSA value is not modified by Rn. • Bytes present in PD are lost. • 1, 2, or 4 bytes are *fetched* from the memory source.
11_01_00_01 11_01_00_10 11_01_00_11	stf Rn, PDA|SZ8 |F stf Rn, PDA|SZ16|F stf Rn, PDA|SZ32|F	• Destination is a byte, half-word, or word target at the Rn address, and any further PD write instructions will trigger byte, half-word, or word access to the destination. • Destination address is frozen.
11_01_01_01 11_01_01_10 11_01_01_11	stf Rn, PDA|SZ8 |I stf Rn, PDA|SZ16|I stf Rn, PDA|SZ32|I	• Destination is a byte, half-word, or word target at the Rn address, and any further PD write instructions will trigger byte, half-word, or word access to the destination. • Destination address is in incremented mode: PDA = PDA + 1, 2, or 4 after write PD.

Table continues on the next page...

Table 55-32. Peripheral DMA STF Instruction List (continued)

Binary	Assembly	Comments
11_01_10_01 11_01_10_10 11_01_10_11	stf Rn, PDA|SZ8 |D stf Rn, PDA|SZ16|D stf Rn, PDA|SZ32|D	• Destination is a byte, half-word, or word target at the Rn address, and any further PD write instructions will trigger byte, half-word, or word access to the destination. • Destination address is in incremented mode: PDA = PDA-1, 2, or 4 after write PD.
11_01_11_01 11_01_11_10 11_01_11_11	stf Rn, PDA|SZ8 |U stf Rn, PDA|SZ16 |U stf Rn, PDA| SZ32 |U	• *Update* destination pointer to memory, which becomes a pointer to a memory accessed in byte, half-word, or word. • PDA value is not modified by Rn • bytes present in PD are lost
11_00_10_00	stf Rn, PD	• Write "dsize" bytes of Rn in PD and automatically flush to destination target
11_11_11_11	stf Rn, PS	• Write status register
11_00_11_00	stf Rn,clrefPS	• Clear error flag if set

NOTE

When writing PD, size information is not important: It is embedded in the dsize field of PDA register. If dsize is 1, 2, or 4, then one, two, or four bytes from Rn is written to the PD register, and automatically flushed out to the destination target.

55.4.12.2.6 Peripheral DMA Read (ldf)-Read Mode

When received from an ldf instruction, the function code bits are interpreted as follows.

Table 55-33. LDF Code Bits

Register	7	6	5	4	3	2	1	0
PSA	s			ar	a			
PDA								
PD			p	cpy				
PS			pssel					

Table 55-34. LDF Code Bits Descriptions

Field	Description
7-6 s	Functional Unit selector 11 for Peripheral DMA
5 p (PD)	Prefetch Flag 0 no prefetch 1 automatic prefetch
4 ar (PSA/PDA)	Address Register Selector 0 PSA

Table continues on the next page...

Table 55-34. LDF Code Bits Descriptions (continued)

Field	Description
	1 PDA
4 cpy (PD)	Copy Mode 0 standard access 1 copy mode access
3 a	Register Set selection 0 PSA or PDA 1 PD or PS
5-0 pssel	PS access selector 111111 is the only valid option to read PS

Table 55-35. Peripheral DMA LDF Instruction List

Binary	Assembly	Comments
11_0_0_0_000	ldf Rn, PSA	Reads 32-bit of PSA value
11_0_1_0_000	ldf Rn, PDA	Reads 32-bit of PDA value
11_0_0_1_000	ldf Rn, PD	Reads programmed source size bytes of PD (0-extended)
11_1_0_1_000	ldf Rn, PD|PF	Reads programmed source size bytes of PD (0-extended), and starts a prefetch at PSA address.
11_0_1_1_000	ldf Rn, PD|COPY	Starts a copy transfer from the source target at the PSA address to the destination target at the PDA address. No data transmits through Rn, but Rn contents are lost (Rn is loaded with PD temporary contents that are *not* the copied data).
11_111111	ldf Rn, PS	Reads 32-bit of PS value

NOTE

When reading PD, size information is not important: It is embedded in the ssize field of the PSA register. If ssize is 1, 2, or 4, the one, two, or four bytes is transferred from PD to Rn. Read data is 0-extended.

55.4.12.2.7 Peripheral DMA Unit Copy Mode

Like burst DMA, the peripheral DMA unit has a copy mode that is used when data transfers do not involve SDMA general registers.

Data is read from the source target at a PSA address, stored in PD, and then automatically flushed to the destination target at the PDA address. Copy mode is only available for transfers that involve two targets of the same data path width.

Since copy mode is invoked with an ldf instruction, the *loaded* general purpose register loses its previous contents. (However, the new contents are unpredictable as they depend on temporary values that are seen on the external DMA bus.)

55.4.12.2.8 Error Management

Peripheral DMA generates two kinds of errors: the immediate error that sanctioned incorrect register programming; and the error triggered by the previous access and stored in the error flag of PS until a DMA instruction is executed.

55.4.12.2.8.1 Immediate Errors

The following table lists all incorrect DMA register setups.

Table 55-36. Immediate Errors with Peripheral DMA

Rn[1:0] values	DMA instruction	Comments
0x01 0x11	stf Rn, PSA|SZ16|F stf Rn, PSA|SZ16|I stf Rn, PDA|SZ16|F stf Rn, PDA|SZ16|I	If PSA points to a half-word peripheral or to a half-word address in memory, its value must be 0 modulo 2.
0x01 0x10 0x11	stf Rn, PSA|SZ32|F stf Rn, PSA|SZ32|I stf Rn, PDA|SZ32|F stf Rn, PDA|SZ32|I	If PSA points to a word peripheral or to a word address in memory, its value must be 0 modulo 4.
PSA[1:0]-PDA[1:0]	DMA instruction	Comments
0x01 0x10 0x11	stf Rn, PSA|SZ32|U stf Rn, PDA|SZ32|U	When PDA or PSA is updated and becomes a pointer to a word address in memory, its content must be 0 modulo 4.
0x01 0x11	stf Rn, PSA|SZ16|U stf Rn, PDA|SZ16|U	When PDA or PSA is updated and becomes a pointer to a half-word address in memory, its content must be 0 modulo 2.
Read/Write PD instruction	Comments	
stf Rn,PD ldf Rn,PD	If PDA size (dsize) has never been set up before an stf PD instruction (dsize=0) If PSA size (ssize) has never been set up before an ldf PD instruction (ssize=0)	
ldf Rn,PD|CPY	Copy mode is possible only between two targets whose data path width is identical. It is P8↔P8, P16↔P16, or P32↔P32 regardless of the way the address registers are incremented.	

55.4.12.2.8.2 Data Transfer Errors

When PSA and PDA are correctly set up, the only error that may arise for an ldf PD or stf PD instruction would be the error of the previous DMA cycle.

Error handling is driven by a single consideration: When an error occurred during a data read on the DMA interface, this error should appear as a transfer error to the core when the core attempts to retrieve the data that was not successfully read from the accessed device (memory or peripheral).

When an error occurred during a write access to the DMA interface, the data is still available in PD and should not be destroyed by subsequent core accesses: The core must be warned about the error issue.

There are three error handling mechanisms for each case: Read Error (First Phase), Write Error and Read Error (Second Phase), and Copy Mode Errors handling.

55.4.12.2.8.3 Read Error (First Phase)

If an error occurred during a prefetch command, the peripheral DMA enters its ERROR READ mode (PS[9:8]=11). In this mode, the error is reported on the next ldf PD instruction and writing PSA, PDA, or PD will cancel the error flag.

The block returns no error mode and instructions are normally executed (a DMA cycle may be triggered). Similarly, initiating a copy transfer will reset the error flag and start a copy transfer. The following table details which instructions can be executed in this mode.

Table 55-37. Possibilities in ERROR READ Mode

DMA Instruction	Immediate Error	Comments
stf rn, PD stf rn, PSA (IU IPF) stf rn, PDA ldf rn,PD|COPY	NO	Error mode is reset, PSA or PDA are updated, or a write cycle is started. For the ldf PD|COPY, a copy loop is executed.
stf rn, PS	NO	PS is updated.
ldf rn, PS ldf rn, PSA ldf rn, PDA	NO	PS, PSA, and PDA could be read in ERROR READ mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, PD	YES	Error of the previous read access is reported here and the peripheral DMA enters its ERROR mode.

55.4.12.2.8.4 Write Error and Read Error (Second Phase)

The peripheral DMA enters its ERROR mode (PS[9:8]=10) when the previous DMA write cycle failed, or, as explained in Read Error (First Phase), when an ldf PD is executed while the block is in ERROR READ mode. When a DMA cycle failed, address registers (PSA, PDA) are not modified and continue to point to the problematic address. In ERROR mode, stf instructions may raise an immediate error, and ldf instructions will not (as detailed in the table below).

Table 55-38. Possibilities in ERROR Mode

DMA Instruction	Immediate Error	Comments
stf rn, PD stf rn, PSA stf rn, PDA	YES	Any attempt to modify PD, PSA, or PDA will raise an immediate error, and the peripheral DMA stays in ERROR mode. When address registers are write accessed, an error is returned.
stf rn, PS	NO	This is the only way to exit the ERROR mode. PS[3] must be reset by an stf PS instruction.
ldf rn, PS ldf rn, PSA ldf rn, PDA	NO	PS, PSA, and PDA could be read in ERROR mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, PD	YES	Whatever the DMA direction (read or write), an ldf rn, PD instruction will show an immediate error.

55.4.12.2.8.5 Copy Mode Errors

Because copy mode is a write access that follows a read access, there are two possible cases of bus error.

When the read access incurs a bus error, the peripheral DMA behaves exactly as described in Read Error (First Phase) and Write Error and Read Error (Second Phase) : It enters its ERROR READ mode, and so on.

When the error occurred during the write access of the copy transfer, the DMA enables the core to retrieve the data that was read because it is assumed the read from the peripheral removed the data from its source device. Therefore, the data to be flushed is still in PD. Any subsequent access to PD triggers an error to the core, which should execute its error handling procedure.

Once the ERROR mode is left (after writing to PS), it is possible for the core to retrieve the data in PD with an ldf instruction or try to flush PD contents once again (for example, when the error was due to a full FIFO and the script waited for the FIFO to be emptied) with another ldf instruction in copy mode. This latter instruction detects that there is valid data in PD, tries to flush it, and thus skips the read phase of the copy instruction. This is a different behavior from the usual stf PD instruction that overwrites PD with the selected General Purpose register contents. The same mechanism can be used any time PD holds data that is not written because of a bus error on the DMA interface; when the data was written via a copy instruction, or via the usual stf PD instruction.

55.4.12.2.8.6 Error Check Example

The following code illustrates an example checking for both immediate and data transfer errors on a store to the PD register. The first bdf instruction checks for an immediate error, but if a data transfer error occurred it is reported until the next instruction to access the Peripheral DMA. A second check of the error flags is done after the ldf PS

instruction. The value of PS here can be ignored. The act of reading any register in Peripheral DMA while it is in an error mode that returns the error to the core to set either the SF or DF flag. Any error returned on an ldf command sets the SF flag and any error returned on an stf instruction sets the DF flag. This can create a situation as shown in the example where a bus error during a DMA write which would normally be considered as a destination fault is reported as a source fault because the error was reported to the SDMA core during an ldf instruction.

Table 55-39. Peripheral DMA Error Check

Function	Instruction	Comment
clrf	0	Clear SF and DF flags
stf	R4, PD	Write data to memory
bdf	error_routine	Check for immediate error from write to PD.
ldf	r3, PS	Read PS (PS value in R3 can be ignored)
bsf	error_routine	Check for bus error from "stf R4,PD". SF is set because it is a ldf instruction, even though the original error was a destination fault.

55.4.12.2.9 Peripheral DMA Unit Prefetch/Flush Management

There is no flush bit because every time data is stored in PD by a stf PD instruction-assuming PDA is correctly programmed-it is automatically flushed to the destination.

An acknowledge is returned in the cycle of the DMA instruction, and the SDMA is only stalled by an instruction that addresses the peripheral DMA when the previous DMA access is not over.

55.4.12.3 OnCE and Real-Time Debug

The On-Chip Emulation block (OnCE) is the debug interface to the SDMA.

It supports the access to all core internal devices (registers, memory, and so on), and provides a set of mechanisms that control the core. The OnCE is accessed by JTAG ports at the chip's board level, or by the host via its peripheral bus.

To reduce the size of the hardware material involved, all tasks supported by the OnCE are performed on the SDMA core. The architecture of the SDMA OnCE is relatively simple and very flexible.

The commands supported by the SDMA OnCE are listed in the following sections.

55.4.12.3.1 Memory and Register Access

A set of mechanisms is provided to access SDMA memory and register locations. Both reading and writing are allowed. The access is supported if the processor is in debug mode.

Those registers can also be accessed through the Arm platform Control interface when the OnCE is controlled by the Arm platform, as described in the "Using BP" section.

55.4.12.3.2 Hardware Breakpoints

An event detection unit is implemented to support memory breakpoints. The unit watches the data exchanged between the SDMA memory bus and the core.

A debug request is sent to the core when matching conditions occur. The unit supports mixed conditions based on address range, access type, and data value. Event detection unit configuration registers are memory mapped in the SDMA space (see Arm platform Channel 0 Pointer (SDMAARM_MC0PTR)): You can modify them through a regular memory access or the Arm platform control interface.

55.4.12.3.3 Watchpoints

One output pin is provided to monitor matching trigger conditions that are defined in the event detection unit.

55.4.12.3.4 Software Breakpoints

The SDMA instruction set contains a software breakpoint. Upon executing a software breakpoint instruction, the core suspends normal execution and enters debug mode.

No hardware step execution mode is implemented in the OnCE, but this feature may be implemented at the software level with this instruction.

55.4.12.3.5 Core Control

Commands are provided to monitor and control processor activity. You can halt the core, rerun the core from another address location, and get processor status.

Any hardware breakpoint on the instruction bus is not supported, but this feature may be implemented by inserting a software breakpoints program.

55.4.13 The OnCE Controller

The OnCE controller receives commands from the Arm platform or from the JTAG controller. Each command is interpreted before being sent to the core.

55.4.13.1 OnCE Commands

A small set of commands supports the communication between the OnCE and the external world.

This command set enables you to perform any of the following tasks: control processor activity, save core context, and execute an SDMA instruction from the OnCE. Combined together, these tasks perform more complex commands.

A full OnCE command contains a 4-bit instruction (the OnCE command opcode) and a variable length data field (the OnCE data). During command execution, the OnCE data is transferred in a OnCE internal register before being exchanged with the SDMA. Some data values are also exported. This mechanism creates a link between the processor and the external world. Nine commands are defined: The following table presents their formats.

Table 55-40. OnCE Command Opcode Values

Instruction Opcode	Name	Action	Register	Data Field Size	Mode
0000	rstatus	Reads the OnCE status register	STATUS	16-bit	normal/debug
0001	dmov	Updates general register GReg1	GREG1	32-bit	debug
0010	exec_once	Runs the instruction from the SDMA instruction register	INSTRUCTION	16-bit	debug
0011	run_core	Returns to normal execution	BYPASS	1-bit	debug
0100	exec_core	Returns to normal execution via a jump instruction that specifies the new address	INSTRUCTION	16-bit	debug
0101	debug_rqst	Stops the core after execution of current instruction	BYPASS	1-bit	normal
0110	rbuffer	Reads the real time buffer	RTB	32-bit	normal/debug
0111-1110	reserved	Reserved	BYPASS	1-bit	normal/debug
1111	bypass	Bypasses TARM platform controller	BYPASS	1-bit	normal/debug

Each instruction corresponds to a specific action performed on the OnCE. The nature of the associated data field is clearly identified. The dmov command is followed by a 32-bit data value (which is a data value for the SDMA); the exec_once and the exec_core commands are followed by a 16-bit data value (which is an instruction for the SDMA); the rstatus command is followed by a 16-bit control value (which is the content of the OnCE status register); the rbuffer command is followed by a 32-bit data value. The

debug_rqst and the run_core commands are followed by a single bit data field (this is a bypass value). Finally, the bypass instruction enables the SDMA JTAG TAP controller to be daisy-chained with another JTAG TAP controller. This is a JTAG-only feature. The set of commands is simple, but enables you to perform any possible task on the SDMA during a debug process.

55.4.13.2 Sending Commands to the OnCE Controller

The JTAG access is the standard access to the OnCE, but sometimes the JTAG is not available to fix some bugs (if the chip is in production for instance), an additional access is then required. Therefore, one Arm platform access to the OnCE is provided.

55.4.13.2.1 Using the JTAG Interface

A serial access is performed through the five JTAG pins TCK, TRST, TMS, TDI, and TDO. A Test Access Port controller is provided to decode the TMS control signal.

It produces shift-enable signals (shift_ir and shift_dr), and updates enable signals (update_ir and update_dr). It is fully compliant with the IEEE 1149.1 testability (JTAG) standard.

During the shift_ir state, the command opcode is shifted into the OnCE controller (for example, the signal from the TDI pin is shifted into the command register and the TDO pin receives the signal shifted out). After transferring the four bits of the command, an update_ir signal is asserted and the command is decoded. The target data register is now clearly identified and the corresponding control signal is produced, as follows: bypass enable signal (bp_en), instruction enable signal (inst_en), data enable (data_en), and status enable signal (stat_en).

During the shift_dr state, the TDI signal is shifted into one of the following target registers: bypass register (1 bit), SDMA instruction register (16 bits), SDMA data register (32 bits), or OnCE status register (16 bits). The TDO pin is connected to the output of the selected register to receive the signals shifted out.

The JTAG access is disabled when the Arm platform access is enabled.

55.4.13.2.2 Using the Arm platform

The Arm platform access to the OnCE is not the standard access, but it is required if the JTAG is not available.

For example, if the SDMA ROM is out of use on a chip in production, and the Arm platform needs to download new code and restart the SDMA, the OnCE can easily perform this operation. This type of debug operation justifies the use of an Arm platform access to the OnCE.

To drive the OnCE, the Arm platform uses some registers contained in the Arm platform Control block of the SDMA. These registers are accessed through the Arm platform peripheral bus. Most of these registers are connected to another register in the OnCE controller. Thus, accessing one of these registers is equivalent to accessing the associated register in the OnCE controller.

The set of registers in the Arm platform Control block is listed below:

- ONCE_ENB register (1 bit, read/write)-This 1-bit register enables the Arm platform access to the OnCE. When this bit is set, the signals from the JTAG are ignored. When it is cleared, all writing operations to the following registers through the Host Control interface are ignored. This register is reset on a JTAG reset.
- ONCE_CMD register (4 bits, read/write)-This 4-bit register receives the command opcode. It is connected to the command register in the controller. A write access to this register causes the associated command to be executed on the OnCE. For example, after writing "0001" in this register, a dmov command is executed.

NOTE

On the Arm platform side, the rstatus and bypass commands are not supported. This register is reset on a JTAG reset.

- ONCE_DATA register (32 bits, read/write)-This 32-bit register is connected to the SDMA data register. This register is used when executing a dmov or rbuffer command.

NOTE

Before requesting a dmov command, the 32-bit data to transfer must be written in the ONCE_DATA register. At the end of the execution, the register is updated with GReg1 former value. This register is reset on a JTAG reset.

- ONCE_INSTR register (16 bits, read/write)-This 16-bit register is connected to the SDMA instruction register. This register is used when executing an exec_core or an exec_once command.

NOTE

Before requesting an exec_core or an exec_once command, the appropriate instruction must be written in the ONCE_INSTR register. This register is reset on a JTAG reset.

- ONCE_STAT register (16 bits, read only)-A read access to the ONCE_STAT register returns the content of the OnCE status register (OSTAT). This register is read only.
- The bypass register is not useful when the Arm platform controls the OnCE, therefore no register is defined in the Arm platform Control block to access the bypass register.

55.4.13.2.3 Conflicts Between the JTAG and the Arm platform Accesses

When Arm platform access to the SDMA OnCE is enabled (that is, when the bit in the ONCE_ENB register is set), the JTAG access is disabled. This guarantees that the block is not accessed at the same time on both sides.

It is possible to check whether the JTAG access to the SDMA OnCE is enabled from the JTAG port. When the JTAG access is disabled, the SDMA TDO always returns 1. The check requires the following steps:

- Execute a dmov command from debug mode (with neither 0xffffffff nor 0x0 as dmov value: 0x5a5a5a5a is good).
- Execute another dmov command (the value here is not important).
- The returned value from the latter dmov command should be the original one if the JTAG access is enabled; if it is 0xffffffff instead of the original input value, this means the JTAG access is disabled.

55.4.13.3 Executing a Command from the OnCE

All the commands defined in OnCE Commands can be accessed through the JTAG. The Arm platform can access all these commands except the rstatus command.

On the Arm platform side, the OnCE status is directly accessed by reading the ONCE_STAT register.

55.4.13.3.1 Nature of the Commands

Two types of commands may be distinguished. First, there are two commands that do not interact with the core: rstatus and rbuffer. Those commands may be requested at any time: They do not depend on the core status.

NOTE

Each of these commands exports a data value or a status value from the SDMA.

There are also commands that interact with the core: dmov, run_core, exec_core, exec_once, and debug_rqst. These commands are core status dependent, as follows:

- During user mode only the debug_rqst is taken into account.
- During debug mode, all these commands are taken into account except the debug_rqst. For example, an exec_once command requested while not in debug mode has no effect.

55.4.13.3.2 Execution Request

The SDMA starts executing a task in debug mode when requested by the OnCE controller. The execution starting time depends on the type of access used to communicate with the OnCE.

If the JTAG is used, the request is send after decoding the update_dr state in the TAP controller. Therefore, always cross this state when sending a command through the JTAG. If the OnCE is driven from the Arm platform side, the request is sent after detecting a write access to the ONCE_CMD register. All the registers involved in this operation must be loaded first.

The following is an example of an exec_core command execution from the Arm platform side: After writing '010' in the ONCE_CMD register, the OnCE controller asks the SDMA to execute the instruction contained in the ONCE_INSTR register. The instruction involved should be available in the ONCE_INSTR register before the beginning of the execution.

55.4.13.3.3 Command Execution

The following list shows the commands and details how each command is executed:

- rstatus command execution-The rstatus command exports the content of the OnCE status register (OSR). If the JTAG is used, the status information is captured in the OnCE status register during the capture_dr state, and shifted out after 16 TCK clock cycles in the shift_dr state. The rstatus command is not supported on the Arm platform side, but a status register is provided instead. The rstatus may be performed in both debug and user modes.
- dmov command execution-The dmov command accesses SDMA internal registers. Executing a dmov instruction exchanges the 32-bit data values between the SDMA data register and the general register GReg[1].
- If the JTAG is used, the content of GReg1 is captured in the SDMA data register during the capture_dr state, then it is shifted out after 32 TCK clock cycles in the shift_dr state. During the update_dr state, GReg1 is updated with the new, shifted-in 32-bit data value. If the OnCE is driven from the Arm platform side, the data values

contained in GReg1 and the SDMA data register are exchanged after detecting a write access to the ONCE_CMD register. The ONCE_DATA register must therefore be loaded first.

- exec_once command execution-The exec_once command executes the instruction loaded in the SDMA instruction register. The command may only be requested from debug mode. The SDMA returns to debug mode at the end of the execution.
- Change of flow instructions as well as instructions that may cause a context switch are not supported: The comprehensive list comprises done/yield/yiedge (except done 5), BF, BT, BSF, BDF, JMP, JSR, JMPR, JSRR, RET, and LOOP, as well as all the illegal instructions.

  No other command should be requested before the SDMA returns to debug mode. The SDMA status (for example, whether it is in debug mode or not) can be detected by polling with the rstatus OnCE command, monitoring the debug_mode pin, or checking the OnCE Status Register (SDMAARM_ONCE_STAT) register via the Arm platform control interface.

NOTE

Most of the instructions are single-cycle, which omits the step of polling the status. Loads and stores to DMA units are typical instructions that might require this polling.

  If the JTAG is used, the 16-bit instruction is shifted in the SDMA instruction register after 16 TCK clock cycles in the shift_dr state. A request is sent to the core when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the Arm platform side, the request is sent to the SDMA when detecting a write access to the ONCE_CMD register. The ONCE_INSTR register must be therefore be loaded first.

- run_core command execution-The run_core command leaves debug mode and resume normal program execution. The next instruction executed is the last instruction decoded before entering debug mode. Be sure to restore core context before re-running the core. This procedure is detailed in Restoring the Context.
- If the JTAG is used, a 1-bit bypass value is shifted in the bypass register in the shift_dr state. The SDMA is rerun when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the Arm platform side, the core is rerun when detecting a write access to the ONCE_CMD register.

- exec_core command execution-The exec_core command resumes program execution from any address. The 16-bit instruction provided with the exec_core overwrites the last instruction decoded before entering debug mode. This command is designed to support change of flow instructions, so that a program execution can be restarted from any address. After executing an exec_core command, the SDMA leaves debug mode. The exec_core command is usually used with a jmp instruction.

- If the JTAG is used, the 16-bit branch instruction is shifted in the SDMA instruction register after 16 TCK clock cycles in the shift_dr state. The SDMA is rerun when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the Arm platform side, the SDMA reruns when detecting a write access to the ONCE_CMD register. The ONCE_INSTR register must therefore be loaded first. For example, to restart the SDMA from the program address 0x100, the instruction loaded should be a jump to address 0x100 instruction.
- debug_rqst command execution-The debug_rqst command puts the SDMA in debug mode. If the JTAG is used, a 1-bit bypass value is shifted in the bypass register during the shift_dr state. A debug request is sent to the SDMA when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the Arm platform side, the debug request is sent when detecting a write access to the ONCE_CMD register. When the SDMA is already in debug mode, this command is simply ignored.
- rbuffer command execution-The rbuffer command exports the content of the real time buffer (RTB). If the JTAG is used, the content of the real time buffer (RTB) is captured in the SDMA data register during the capture_dr state. The register is completely shifted out after maintaining the shift_dr state during 32 TCK clock cycles. If the OnCE is driven from the Arm platform side, the content of the RTB is captured in the ONCE_DATA register after detecting a write access to the ONCE_CMD register.
- bypass command execution-This command is only available from the JTAG interface. It enables daisy-chaining of the SDMA JTAG TAP controller with other JTAG TAP controllers. This command does not change the SDMA state and can be executed in any mode (run, debug, or sleep). It selects the bypass register of the TAP controller.

55.4.13.4 Registers Descriptions

See SDMACORE, and SDMAARM, for detailed information on each register.

55.4.13.4.1 Event Cell Counter Register (ECOUNT)

The event cell counter register is a 16-bit register that contains the number of times minus one that an event detection occurs before generating a debug request.

This register should be written before attempting to use the event detection counter during an event detection process. The event cell counter register is cleared on a JTAG reset.

55.4.13.4.2 Event Cell Address Registers (EAA or EAB)

The event cell contains two address registers-the event cell address register (a), called EAA, and the event cell address register (b), called EAB. Every address register is a 16-bit register that stores a user-defined address value. This value computes one of the following address conditions: addra_cond or addrb_cond. Every address register is cleared on a JTAG reset.

55.4.13.4.3 Event Cell Address Mask Register (EAM)

The event cell address mask register is a 16-bit register that contains a user-defined address mask value. This mask is applied to the address value latched from the memory address bus before comparing addresses.

NOTE

There is a common address mask value for the two address comparators. If bit *i* of this register is set, then bit *i* of the address value latched from the memory bus does not influence the result of the address comparison. The event cell address mask register is cleared on a JTAG reset.

55.4.13.4.4 Event Cell Data Register (ED)

The event cell data register is a 32-bit register that contains a user-defined data value. This data value is an input for the data comparator, which generates the data_cond condition.

The event cell data register is cleared on a JTAG reset.

55.4.13.4.5 Event Cell Data Mask Register (EDM)

The event cell data mask register is a 32-bit register that contains a user-defined data mask value. This mask is applied to the data value latched from the memory bus before comparing data.

Setting bit *i* of the event cell data mask register means that bit *i* of the data value latched from the address bus does not influence the result of the data comparison. The event cell data mask register is cleared on a JTAG reset.

55.4.13.4.6 Real Time Buffer Register (RTB)

The real Time Buffer register is a 32-bit register that stores and retrieves run-time information without putting the SDMA in debug mode.

Refer to Real Time Buffer for more details.

55.4.13.4.7 Event Control Register (ECTL)

The event cell control register is a 16-bit register that defines cell event occurrence conditions.

The event cell control register is cleared on a JTAG reset. See also OnCE Event Detection Unit for more details.

55.4.13.4.8 Trace Buffer (TB)

The Trace Buffer register retrieves the information in the Trace Buffer.

See Trace Buffer for more details.

55.4.13.4.9 OnCE Status Register (OSTAT)

The OnCE status register is a 16-bit register that contains processor and event detection unit status. The OSTAT is a read-only register.

Refer to OnCE Status Register (SDMAARM_ONCE_STAT) for detailed description of the individual fields in the OSTAT register.

The following figure shows the OSTAT structure.

Table 55-41. OnCE Status Register (OnCE)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PST[3:0]				RCV	EDR	ODR	SWB	MST					ECDR[2:0]		

Where PST[3:0] is the SDMA core state, RCV is set when the real-time buffer (RTB) is modified. EDR, ODR, and SWB are set, respectively, when the SDMA has entered debug mode because of an external debug request, a OnCE debug_rqst command, or a software breakpoint. MST is set when the OnCE is controlled from the Arm platform control interface, and when ECDR is a three-flag set that shows the event cell condition(s) that put the core in debug mode. The OSTAT never provides more than one reason for entering debug mode.

There are two ways of accessing OSTAT content, as follows:

1. Send an rstatus command to the OnCE controller through the JTAG, or read the ONCE_STAT register through the Arm platform access. Executing the rstatus command through the JTAG can be performed in both user and debug modes.

2. Perform an SDMA read access to the location in the SDMA core memory map (OSTAT register) debug mode using the exec_once command. With this method of access, the SDMA state reflected by the PST (processor status bit) is always DATA.

The register may also be accessed by a running application.

55.4.13.5 JTAG Interface Requirements

Because the signals received from the JTAG (running on TCK) are transferred to the OnCE controller (running on the SDMA clock), a synchronization mechanism is required.

55.4.13.5.1 TCK Speed Limitation

In the JTAG top-level layer, the TDO signal is always captured on a TCK falling edge. To guarantee a stable TDO signal from the SDMA during this operation, a falling edge detection is performed on TCK.

Before being latched in the *I* flip-flop (see Figure 55-11) on TCK falling edge, the TDO signal must be stable at the input of the flip-flop. This condition is verified if the TCK period is superior to the following delay:

worst-case edge detection delay + negative-edge signal propagation delay + JTAG top-level logic propagation delay

The frequency relationship, TCK < CLK/8, limitation guarantees that all operations are performed as expected.

55.4.13.5.2 Synchronization Implementation

The figure found here shows the synchronization mechanism.

Flip-flops tck0, tck1, and tck2 perform falling- and rising-edge detections on TCK. They generate the posedge_detected and negedge_detected nets that are used to sample the TDI and TMS inputs into the respective tdi and tms flip-flops, and update the tdo flip-flop to yield the TDO output. In the design, the only signal that might go metastable is the output of the tck0 flip-flop. This signal is captured in the tck1 flip-flop and no logical operation is performed on it to minimize a metastability propagation risk.

The TDI and TMS flip-flops also cannot go metastable: The propagation time of the rising-edge detection signal through tck0, tck1, and tck2 guarantees that the TDI and TMS inputs are stable when captured in the TDI and TMS flip-flops.

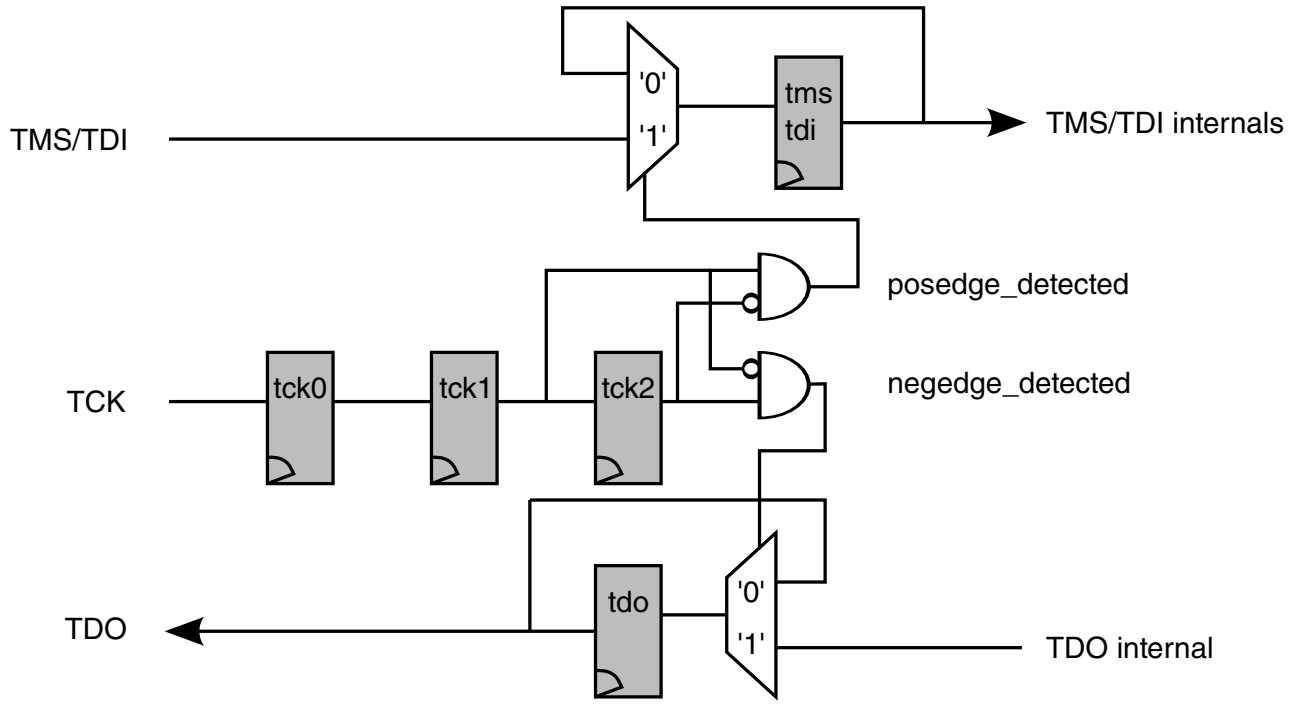


Figure 55-11. OnCE Synchronization Layer

The following figure shows synchronization timings. It takes three CLK clock cycles to synchronize TDI on the SDMA clock.

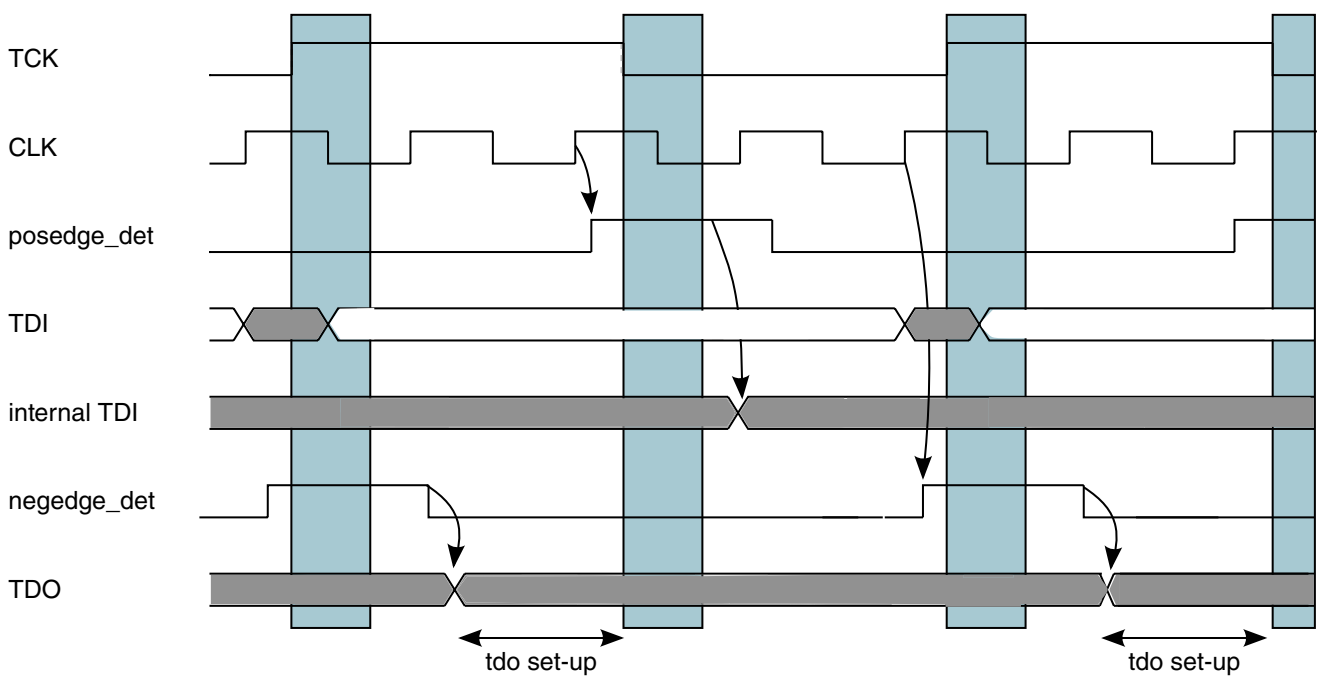


Figure 55-12. Synchronization Timings

55.4.13.5.3 JTAG Controller Start-Up Recommended Procedure

To ensure correct TAP controller initialization, it is recommended to use the following procedure:

1. Assert JTAG reset TRSTB (for example, set low).
2. Set TMS low.
3. Wait for 1 TCK clock.
4. Release JTAG reset TRSTB (for example, set high).
5. Wait for a minimum of five TCK cycles.

55.4.14 Using the OnCE

This section provides the elements necessary to run the OnCE during a debug process.

In addition to the basic set of commandsdescribed in OnCE Commands, more complex commands can be built to meet users' requirements.

55.4.14.1 Activating Clocks in Debug Mode

For power consumption issues, some clocks in the SDMA are disabled when not needed.

This is the case for instances when the SDMA is in sleep mode. Clock gating management depends on the interface used to control the OnCE.

- For the JTAG access, the SDMA clock gating must be turned off via the clk_gating_off input.
- For the Arm platform access, the SDMA clock gating is automatically turned off when the Arm platform access is enabled (see OnCE Enable (SDMAARM_ONCE_ENB)).

55.4.14.2 Getting the Current Status

Most of the commands the OnCE supports have an impact on the status of the SDMA.

It is not permissible to request the execution of an instruction on the SDMA from the OnCE while the SDMA is not in debug mode. Such a violation may cause unpredictable behavior, and it might be necessary to reset the SDMA.

Therefore, the value of the PST bits provided in the OnCE status register should always be checked before sending any request to the SDMA.

55.4.14.3 Methods of Entering Debug Mode

A debug request may be asserted at any time, but it is not always taken into account immediately. Debug mode cannot be entered in the middle of an instruction, or during the save or restore states of a context switch.

The request is ignored when the core is already in debug mode. Refer to Figure 55-4, which shows all possible transitions to the debug state, as there are several ways to enter debug mode.

55.4.14.3.1 External Debug Request During Reset

To enter debug mode after exiting reset, the external debug line has to be maintained high. This line is handled by the JTAG top-level block.

NOTE

The SDMA detects the debug requests only if the SDMA clock is running (see Activating Clocks in Debug Mode). The debug request line should be not be maintained high when the SDMA is in debug mode.

NOTE

The debug_rqst command (from the OnCE command set) is not supported during system reset.

55.4.14.3.2 Debug Request During Normal Activity

During normal activity, the SDMA enters debug mode when the following is true:

1. If the debug request line from the JTAG top-level is asserted, or
2. If the OnCE controller receives a debug_rqst command.

   The debug_rqst command can be sent by the JTAG access or by an access on the Arm platform side (if the Arm platform access is enabled).

55.4.14.3.3 Software Breakpoint Instruction

The SDMA enters debug mode at the end of the execution of a software breakpoint instruction. This instruction must be inserted in program flow executed by the core.

55.4.14.3.4 Event Detection Unit Matching Condition

If the event detection is enabled, a debug request is sent to the core after detecting a matching condition on the SDMA memory bus.

See OnCE Event Detection Unit for more details.

55.4.14.4 Executing Instructions in Debug Mode

The OnCE supports a mechanism to execute instructions in debug mode. If the SDMA is in debug mode, then the exec_once command can be used to execute an SDMA instruction from the OnCE controller. The SDMA returns to debug mode at the end of each execution.

Some instructions are not supported by the exec_once command: done/yield/yiedge (except done 5), BF, BT, BSF, BDF, JMP, JSR, JMPR, JSRR, RET, and LOOP, as well as all the illegal instructions are not supported.

NOTE

While instructions are executed in debug mode from the OnCE, the program counter of the SDMA is not incremented.

55.4.14.5 Command Sequences Examples

This section provides examples of command sequences that run the SDMA in debug mode. These sequences are available for both the Arm platform and JTAG accesses.

The following presents the syntax used in this section. The data field provided with each command is put in parenthesis with the command name. A '-' is used if the data field provided is a *don't care* value.

```
my_command(data_field);                 // executing my_command with a data field
my_command(-);                        // executing my_command with a don't care data field
```

The value returned by the command (if there is one) is referred by an assignment. In case the value returned by the command is not used, the assignment is omitted. For an Arm platform access, the value returned (it is always a data value) is obtained by reading back into the SDMA data register.

```
data_out = my_command(data_in); // returning a data value
```

To clarify the syntax, the instructions' opcodes are referred to by their names. In practice, use the corresponding 16-bit encoding.

55.4.14.5.1 Getting the SDMA Status

NOTE

Before executing any command that affects the SDMA (like dmov or exec_once), check that the SDMA is in debug mode.

Use the following snippet:

```
rstatus();        // read SDMA status until the SDMA is in debug mode
...
rstatus();
```

If the SDMA is not in debug mode, then a debug request must be generated. In this case, the SDMA enters debug mode at the end of the execution of the current instruction. Use this snippet:

```
debug_rqst(-);   // debug request
```

In the following sections, it is assumed that the SDMA was successfully put into debug mode.

55.4.14.5.2 Saving the Context

The first debug task is to save the SDMA context, which is the content of the eight general-purpose registers, the loop and PC-related registers, and the flags.

Use the general register GReg[1] as an intermediate register to export the entire context of the SDMA.

The following example shows how to save GReg[0], GReg[1], GReg[2] and GReg[3]. The sequence of commands used to export additional general registers is very similar to this.

Save GReg[0], GReg[1], GReg[2], and GReg[3]

```
GReg1_data = dmov(-);                          // the value exported is the content of
GReg[1]
exec_once("mov GReg1,GReg0");                      // puts the content of GReg[0] into
GReg[1]
GReg0_data = dmov(-);                          // the value exported is the content of
GReg[0]
exec_once("mov GReg1, GReg2");                     // puts the content of GReg[2] into
GReg[1]
GReg2_data = dmov(-);                          // the value exported is the content of
GReg[2]
exec_once("mov GReg1, GReg3");                     // puts the content of GReg[3] into
GReg[1]
GReg3_data = dmov(-);                          // the value exported is the content of
GReg[3]
```

Get the value of the internal flags (SF, DF, T, and LM), of the loop related registers (EPC and SPC), and of the PC-related registers (PC and RPC). Use a done 5, which is the formatting instruction dedicated to the debug. This instruction formats the flags and the values contained in the registers. It also writes the resulting values into the channel

context memory. It should not be used when entering debug from the IDLE state (for example, with no active channel script running on the SDMA), because it will update a channel context that may belong to any channel.

```
exec_once("done 5");                    // formatting the value of flags and registers
```

At this point, the channel context should be up-to-date in memory, and debug operations should now be possible. However, the context can be exported with the following instructions:

Exporting the Context

```
dmov(ctx_base_addr);                              // loading GReg[1] with the channel
context base address
exec_once("ld GReg0,(GReg1,0)");                      // get RPC-PC into GReg0
exec_once("ld GReg1, (GReg1,1)");                     // get SPC-EPC into GReg1
Loop_data = dmov(-);                           // read back the value of Loop registers
exec_once("mov GReg1, GReg0");                    // puts the PC info into GReg1
PC_data = dmov(-);                             // reads back the content of the PC registers
```

After this sequence of operations, the entire SDMA context is exported via the OnCE.

55.4.14.5.3 Restoring the Context

At this point in the operation, restore the context of the SDMA. It can be different from the original context located in memory, and the content previously saved into the debugging application via the OnCE.

The example found hereshows how it is possible to modify the current channel context.

Modifying the Current Channel Context

```
dmov(Loop_data);                              // put Loop former value into GReg[1]
exec_once("mov GReg0, GReg1");                    // copy to GReg[0]
dmov(PC_data);                                // put PC former value into GReg[1]
exec_once("mov GReg2, GReg1");                    // copy to GReg[2]
dmov(ctx_base_addr);                              // put channel context base address into
GReg[1]
exec_once("st GReg0, (GReg1,1)");                     // restore Loop context
exec_once("st GReg2, (GReg1,0)");                     // restore PC context
```

Once the context in memory is the desired context (with or without applying the previous instruction sequence), it can be restored to the *real* PC and loop registers in the SDMA core:

```
exec_once("cpShReg");                    // restore flags and PC & loop related registers
```

After this command, the SDMA core PC, RPC, SPC, EPC registers, as well as the flags contain the same data as what is stored in the context RAM for the current channel.

The following example shows how to restore the context of general registers GReg[0], GReg[1], GReg[2] and GReg[3].

Restoring the General Register Context

```
dmov(GReg3_data);                    // put GReg[3] restore value in GReg[1]
exec_once("mov GReg3, GReg1"); // restore GReg[3]
dmov(GReg2_data);                    // put GReg[2] restore value in GReg[1]
exec_once("mov GReg2, GReg1"); // restore GReg[2]
dmov(GReg0_data);                    // put GReg[0] restore value in GReg[1]
exec_once("mov GReg0, GReg1"); // restore GReg[0]
dmov(GReg1_data);                    // restore GReg[1]
```

At this point, it is possible to restart the normal program execution.

NOTE

Every SDMA core general register value can be modified by a mov instruction, which makes modification of these registers easy during debug. Unfortunately, there is no such instruction as a mov to directly modify the contents of either PCU register or flag (PC, RPC, SPC, EPC, T, LM, SF, or DF). The cpShReg instruction is meant to provide a means for changing these register contents via the context memory.

55.4.14.5.4 Accessing the Memory

In the example shown here, it is assumed that the SDMA context is entirely saved. If true, it is permissible to modify the general purpose registers during debugging activity.

To perform a memory read access, the target address is stored via the OnCE in GReg[1], then the load instruction is executed on the SDMA (the data loaded from the memory overwrites the address contained in GReg[1]), and then the result value is read back via the OnCE.

```
macro READ:            dmov(target_addr);                                  // put the target
address in GReg[1]
                   exec_once("ld GReg1,(GReg1,0)");                          // execute the
load instruction
                   res_data = dmov(-);                                 // exports the result
data value
```

For a memory write access, the target address is written in GReg[0], and the value to store is written in GReg[1]. Then the store instruction is executed on the SDMA.

```
macro WRITE:            dmov(target_addr);                                   // puts the
target address in GReg[1]
                   exec_once("mov GReg0,GReg1");                          // puts the target
address in GReg[0]
                   dmov(target_data);                                  // puts the target
data in GReg[1]
                   exec_once("st GReg1,(GReg0,0)");                          // performs the
store operation
```

This sequence is shown as an example; however, many other sequences are possible.

NOTE

This sequence of commands can also be applied to memory-mapped registers.

55.4.14.5.5 Resuming Program Execution

Before resuming program execution, it is assumed that the SDMA context is properly restored. There are two ways to restart the SDMA.

Start by executing the last instruction fetched before entering debug mode, as follows.

```
run_core(-);                          // resume execution from where we stopped before
```

If necessary, restart the execution from a different address. In this case, use the exec_core command. The data field provided with this command must be the encoding of a jump instruction.

```
exec_core("jmp start_addr"); // rerun the SDMA from another address
```

In these two examples, the SDMA exits debug mode and keeps executing the code fetched from the memory.

55.4.14.5.6 Single Stepping in RAM

To execute a program step-by-step from the RAM, insert software breakpoints in the program flow at appropriate places so that the SDMA only executes one instruction before returning to debug mode.

First, read the next instruction to execute in the RAM. Then, depending on the value of this instruction, compute the address where a software breakpoint instruction should be inserted. The instruction at the corresponding address must be saved, and, the software breakpoint instruction is inserted. After restarting the SDMA, there is only one instruction executed before meeting the software breakpoint.

The following example shows the macro functions READ and WRITE, which correspond to the sequence of commands (described above) used to access the memory.

NOTE

The data read from the memory are 32-bit values, while the instructions are 16-bit values only. This is why it is best to only use addresses divided by two when accessing the memory.

READ and WRITE Macro Functions

```
next_instr = READ(run_addr/2);                        // read the next instruction to execute
// the tool now has to compute the address where the breakpoint
// instruction should be inserted, this address is the "bkpt_addr"
instr_save = READ(bkpt_addr/2);                        // save the instruction before
overwriting
STORE("bkpt instruction",bkpt_addr/2);                       // store the bkpt instruction
in memory
exec_core("jmp run_addr");                            // rerun the SDMA
rstatus(-);                                         // wait for the SDMA to enter debug mode
...
rstatus(-);
```

```
STORE(instr_save,bpkt_addr/2);                    // restore the instruction
overwritten
```

In case of branched conditional instructions, a breakpoint instruction should be written at the two possible target addresses.

55.4.14.5.7 Single Stepping in ROM

No single-step mechanism is supported in ROM. The program code can be loaded in the RAM, where the single-step mechanism can be executed.

55.4.14.6 OnCE Event Detection Unit

The event detection unit watches signals from the data memory bus (DMBUS), which the SDMA core uses to access its RAM, ROM, and memory mapped registers.

A debug request is sent to the OnCE controller when user-defined conditions on address and/or data values are true.

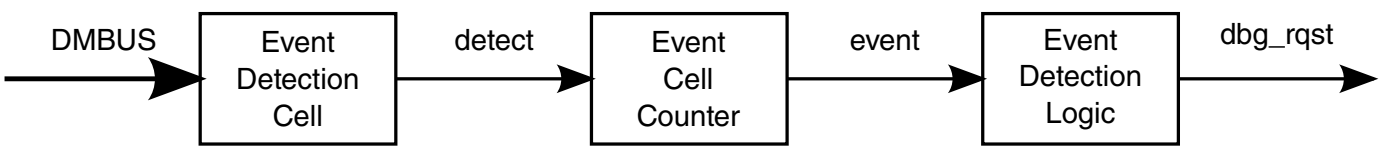


Figure 55-13. Event Detection Unit

A counter, provided with the detection cell, is decreased after an event detection. A debug request is sent to the core only when the counter reaches the value of 0. It is possible to disable the use of the counter if a debug request has to be generated after each event detection.

The event cell is the basic block that supports hardware breakpoints on an address value and/or data values coming from the SDMA memory bus. The trigger condition that generates the debug request is a mixed condition based on those values.

The following figure shows the event cell architecture. The event cell contains the address (stored in the memory address register) and the data (stored in the memory data register) used during the last memory access. There are some user-defined reference values located in memory mapped registers-the event cell addresses, the event cell address mask, the event cell data, and the event cell data mask. These registers are accessed by standard load/store instructions just like regular memory locations.

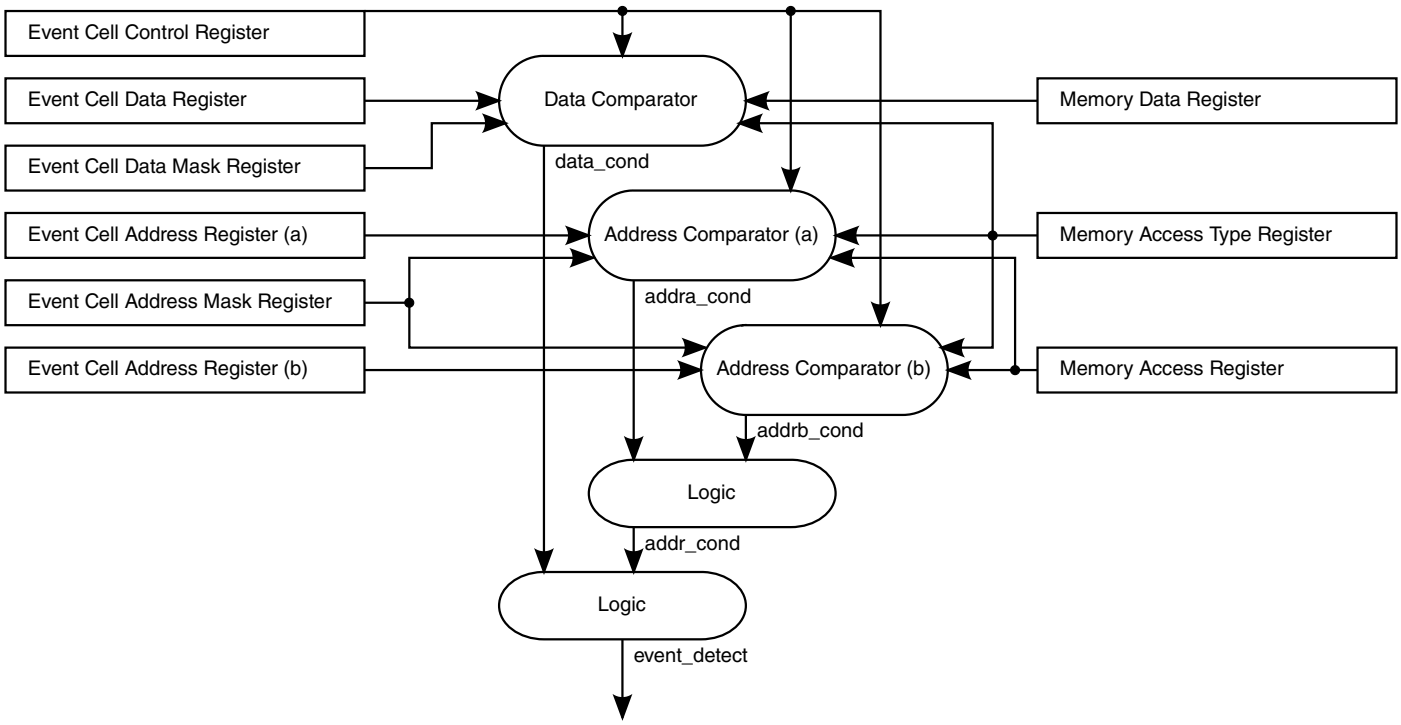


Figure 55-14. Event Cell Architecture

To define a memory breakpoint, three conditions are taken into account: The first two conditions are comparisons of the current memory address with user-defined reference addresses (these conditions are called addressA and addressB). The third condition consists of a comparison between the data received on the DMBUS and a user-defined reference data (this condition is called data). An intermediate address condition is set to express a dependency between addressA and addressB conditions.

55.4.14.7 Clock Gating and Reset

This section details how to use the clocks and handle the reset signals.

55.4.14.7.1 Clocks

Because the SDMA uses clock gating to save power, it is necessary to disable the clock gating and force the clocks to be enabled when using the OnCE.

When the OnCE is accessed through its JTAG interface, clock gating must be disabled outside the SDMA via a dedicated SDMA input port clk_gating_off. The reason why detection is not performed automatically by the SDMA internal hardware is that it would cost power to monitor activity on the JTAG interface.

When the OnCE is accessed through the Arm platform Control interface, clock gating is automatically turned off. This is done when bit 0 of the ONCE_ENB register (see OnCE Enable (SDMAARM_ONCE_ENB)) is set. A write access to this register is possible even when the OnCE clock is not running. If the Arm platform access is used, the bit in the ONCE_ENB register must be set before any attempt to access any other OnCE register.

55.4.14.7.2 Resets

The OnCE reset is different from the SDMA main reset.

Normally, activating the SDMA reset while keeping the OnCE reset inactive (when possible) enables you to reset the core without having to reprogram the OnCE.

55.4.14.8 Real Time Features

To rebuild the skeleton of a program execution, it is necessary to store the addresses of the program instructions where jumps are taken: A trace buffer is therefore provided. A real time buffer has also been added to receive data values written during a program execution.

The content of this register may be exported through JTAG ports without stopping the core.

55.4.14.8.1 Trace Buffer

The Trace Buffer is a 32-stage buffer that contains appropriate information to identify the 32 last changes of flow detected during a program execution.

The following figure shows an overview of the Trace Buffer.

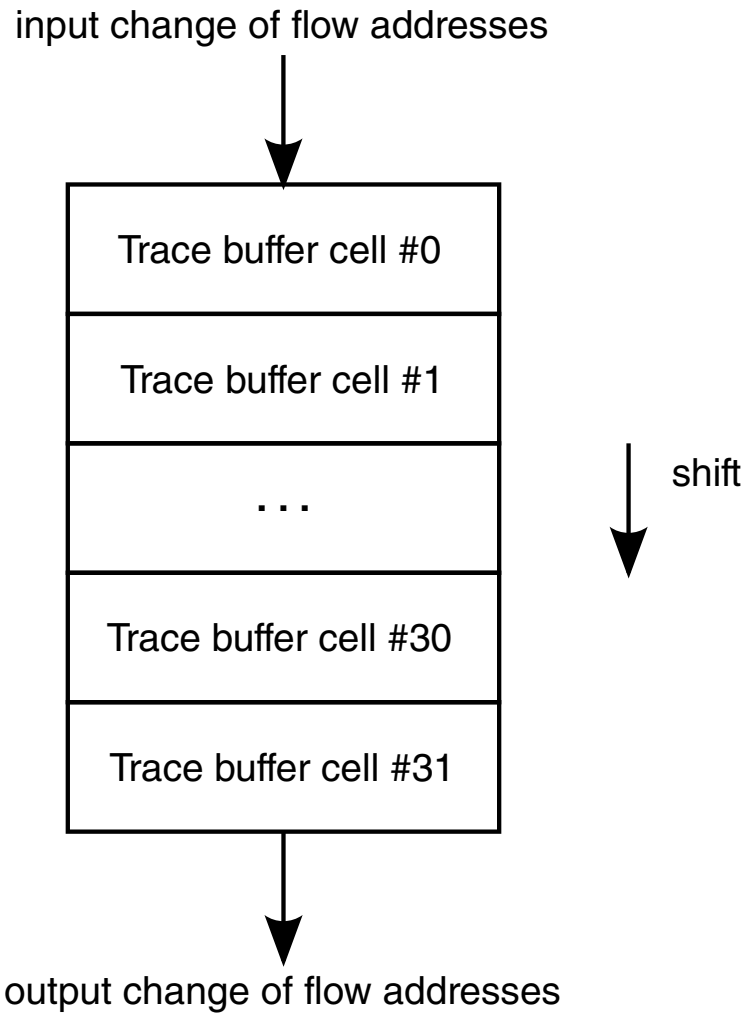


Figure 55-15. Trace Buffer

Each cell of the trace buffer contains two reference addresses and a flag. The flag is set when the addresses stored in the cell correspond to a valid change of flow; otherwise, the flag is cleared. The three most significant bits are unused.

After every change of flow detection, the address of current instruction and the address of the target instruction are stored at the top of the Trace Buffer (cell #0). The flag in the cell is set to indicate that a valid change of flow was detected. Former cell values are shifted one level down. The Trace Buffer contains the 32 last changes of flow. All the flags are reset on a software or a hardware reset, and after each transition from debug mode to user mode.

A memory mapped register of SDMA core, the Trace Buffer register (TB), is provided to read the content of the Trace Buffer. This operation should be done in debug mode. Performing a read access to the Trace Buffer register returns the content of the bottom of the Trace Buffer (cell #31). After every read access, the trace buffer is shifted one level down, and the flag at the top of the trace buffer is cleared.

A typical OnCE command sequence that retrieves the oldest change-of-flow information is a follows:

```
exec_once("mov r1, TB");                              // stores the oldest change-of-flow in
GReg1
dmov(-);                                          // retrieves GReg1 contents
```

This sequence requires the SDMA to be put in debug mode.

55.4.14.8.2 Real Time Buffer

The Real Time Buffer register (RTB) is a memory mapped register that can be accessed as a regular memory location by the SDMA core during program execution. This register is located in the OnCE.

Executing ar rbuffer command (see The OnCE Controller for further details) exports the content of this register through JTAG ports.

When a write access is performed at the memory location corresponding to the RTB, the receive flag (for example, the RCV bit) is set in the OnCE Status Register (OSR). This flag is cleared at the end of the execution of a rbuffer command.

NOTE

Every write access to the RTB memory location updates the RTB register even if the RCV flag is set. The RTB is cleared on a JTAG reset.

55.4.14.8.3 Emulation Pin

The debug_matched_event emulation pin reflects the matching condition status detected by the Event Detection Unit.

Since it can be necessary to detect conditions without triggering debug requests, it is possible to disable the generation of debug requests by the Event Detection Unit and still have the matching condition available on the emulation pin. This can be done by clearing the EN flag in the ECTL register.

55.4.14.8.4 Real-Time Debug Outputs

The table found here shows the debug signals that are available at the SDMA boundaries. Their availability at chip boundaries depends on the project.

Table 55-42. Real-Time Debug Output Pins

Pin	Description
debug_core_state[3:0]	The core_state bits reflect the state of the SDMA core. • The "Program" state is the usual instruction execution cycle. • The "Data" state is inserted when there are wait-states during a load or a store on the data bus (ld or st). • The "Change of Flow" state is the second cycle of any instruction that breaks the sequence of instructions (jumps and channel switching instructions). • The "Change of Flow in Loop" state is used when an error causes a hardware loop exit. • The "Debug" state means the SDMA is in debug mode. • The "Functional Unit" state is inserted when there are wait-states during a load or a store on the functional units bus (ldf or stf). • In "Sleep" modes, no script is running (this is the core idle state); the "after Reset" is slightly different because no context restoring phase will happen when a channel is triggered: The script located at address 0 is executed (boot operation). • The "in Sleep" states are the same as above except they do not have any corresponding channel: they are used when entering debug mode after reset; the reason is that it is necessary to return to the "Sleep after Reset" state when leaving debug mode. 0 Program 1 Data 2 Change of Flow 3 Change of Flow in Loop 4 Debug 5 Functional Unit 6 Sleep 7 Context Switch Saving Channel 8 Program in Sleep 9 Data in Sleep 10 Change of Flow in Sleep 11 Change of Flow in Loop in Sleep 12 Debug in Sleep 13 Functional Unit in Sleep 14 Sleep after Reset 15 Context Switch Restoring Channel
debug_yield	Pulse that is active when a yield (done 0) or a yieldge (done 1) instruction is executed. 0 - 1 yield/yieldge executed
debug_core_run	Active when the SDMA core is executing instructions. 0 Debug or sleep mode

Table continues on the next page...

Table 55-42. Real-Time Debug Output Pins (continued)

Pin	Description
	1 Run mode
debug_event_channel_sel	Indicates if debug_event_channel displays current channel or last received event 0- debug_event_channel[5:0] gives the number of the current channel 1- debug_event_channel[5:0] gives the number of the last received event
debug_event_channel[5:0]	Gives the number of any DMA request as soon as it is received or the number of the current channel. The value of debug_event_channel_sel indicates if debug_event_channel displays the current channel or last received event. The signal debug_event_channel_sel must be observed to determine what information is provided on debug_event_chanel at any given time.
debug_pc[13:0]	Program Counter value; it has a meaning when the core is in run mode.
debug_mode	Set when the core is in debug. 0 - 1 Core is in debug
debug_bus_error	Set when an error was received during a load or a store (ld, st, ldf, or stf instruction) and registered in SF or DF flag. 0 No error during last load/store 1 Error during last load/store
debug_bus_device[4:0]	Indicates the device or functional unit that is accessed by the current instruction. The debug_bus_device output is always valid when in sleep mode, debug mode, or executing any instruction that does not access the functional units or the memory mapped devices, "no access" is output. 0 No access 1 MSA 2 MDA 3 MD 4 MS 5 PSA 6 PDA 7 PD 8 PS 9 RESERVED 10 RESERVED 11 RESERVED 12 RESERVED 13 CA 14 CS 15 Reserved 16 Memory (RAM or ROM) 17 Memory mapped register

Table continues on the next page...

Table 55-42. Real-Time Debug Output Pins (continued)

Pin	Description
	18 Peripheral #1 19 Peripheral #2 20 Peripheral #3 21 Peripheral #4 22 Peripheral #5 23 Peripheral #6 24 Peripheral #7 25 Peripheral #8 26 Peripheral #9 27 Peripheral #10 28 Peripheral #11 29 Peripheral #12 30 Peripheral #13 31 Peripheral #14
debug_bus_rwb	Indicates the direction of the access given by debug_bus_device 0 Write access (st or stf) 1 Read access (ld or ldf)
debug_matched_dmbus	Pulse indicating the OnCE event detection unit has detected a match on the data bus during an access to memory (RAM or ROM), a memory mapped register or a peripheral that is hooked to the SDMA. 0 - 1 data bus match detected
debug_rtbuffer_write	Pulse indicating when the real-time buffer is written by the core. 0 - 1 RTB was modified
debug_evt_chn_lines[7:0]	Eight lines that generate short pulses when DMA requests are received or channels are (re)started. Every line is controlled through two parameters defined in registers Cross-Trigger Events Configuration Register 1 (SDMAARM_XTRIG_CONF1) (as described in SDMAARM). The following two parameters are available for every line: • CNF-Indicates what is monitored on the line: 0 for a channel start, 1 for a DMA request reception • NUM[5:0]-Gives the number of the DMA request or channel to monitor

The matched_event emulation pin reflects the matching condition status detected by the Event Detection Unit. Because it can be necessary to detect conditions without triggering debug requests, it is possible to disable the generation of debug requests by the Event Detection Unit and still have the matching condition available on the emulation pin. This can be done by clearing the EN flag in the ECTL register.

All real-time debug outputs are disabled by default (for example, they are stuck to 0) to avoid power consumption when they are not used. They are enabled when bit 11 (RTDOBS) of the Configuration Register (SDMAARM_CONFIG) is set. Signals provided to the system JTAG controller for SDMA debug mode status will also be enabled when the *clk_gating_off* input is asserted.

55.5 Instruction Set

55.5.1 Instruction Encoding

This section presents a short summary of the instruction codes. All context switch instructions are listed for information only; they cannot function properly out of the context switch routine.

```
x...x - don't care

rrr - destination/source general register

sss - additional source general register

bbb - general register used as address base register

ddddd - address displacement

nnnnn - bit number
uuuuuuuu - function unit command bits

pppppppp - branch displacement (signed)

iiiiiiii - 8-bit immediate

jjj - control bit to clear

ff - flag to clear
00000jjj00000000        - done (done,yield,wait)
00000jjj00000001        - notify
00000xxx00000010        - reserved
00000xxx00000011        - reserved
00000xxx00000100        - reserved
0000000000000101     - softBkpt
0000000100000101     - reserved
0000001000000101     - reserved
0000001100000101     - reserved
0000010000000101     - reserved
0000010100000101     - reserved
0000011000000101     - reserved
0000011100000101     - reserved
0000000000000110     - ret
0000000100000110     - reserved
0000001000000110     - reserved
0000001100000110     - reserved
0000010000000110     - reserved
0000010100000110     - reserved
0000011000000110     - reserved
```

```
0000011100000110     - reserved
000000ff00000111       - clrf ff
0000010000000111     - reserved
0000010100000111     - reserved
0000011000000111     - reserved
0000011100000111     - illegal
00000rrr00001000        - jmpr r
00000rrr00001001        - jsrr
00000rrr00001010        - ldrpc r
00000rrr00001011        - reserved
00000rrr000011xx          - reserved
00000rrr00010000        - revb
00000rrr00010001        - revblo
00000rrr00010010        - rorb
00000rrr00010011        - reserved
00000rrr00010100        - ror1
00000rrr00010101        - lsr1
00000rrr00010110        - asr1
00000rrr00010111        - lsl1
00000rrr001nnnnn        - bclri r,n
00000rrr010nnnnn        - bseti r,n
00000rrr011nnnnn        - btsti r,n
00000xxx10000xxx           - reserved
00000rrr10001sss           - mov
00000rrr10010sss           - xor
00000rrr10011sss           - add
00000rrr10100sss           - sub
00000rrr10101sss           - or
00000rrr10110sss           - andn
00000rrr10111sss           - and
00000rrr11000sss           - tst
00000rrr11001sss           - cmpeq
00000rrr11010sss           - cmplt
00000rrr11011sss           - cmphs
0000011011100000     - reserved
0000011011100001     - reserved
0000011011100010     - cpShReg
0000011011100011     - reserved
0000011011100100     - reserved
0000011011100101     - reserved
0000011011100110     - reserved
0000011011100111     - reserved
00000xxx11101xxx           - reserved
00000xxx11110xxx           - reserved
00000xxx11111xxx           - reserved
00001rrriiiiiiii               - ldi r,i
00010rrriiiiiiii               - xori r,i
00011rrriiiiiiii               - addi r,i
00100rrriiiiiiii               - subi r,i
00101rrriiiiiiii               - ori r,i
00110rrriiiiiiii               - andni r,i
00111rrriiiiiiii               - andi r,i
01000rrriiiiiiii               - tsti r,i
01001rrriiiiiiii               - cmpeqi r,i
01010rrrdddddbbb         - ld r,(d,b)
01011rrrdddddbbb         - st r,u
01100rrruuuuuuuu         - ldf r,u
01101rrruuuuuuuu         - stf r,u
011100xxxxxxxxxx               - reserved
011101xxxxxxxxxx               - reserved
011110ffnnnnnnnn       - Loop ff flags are reset
01111100pppppppp     - bf pc=pc+signed(pppppppp)+1
01111101pppppppp     - bt pc=pc+signed(pppppppp)+1
01111110pppppppp     - bsf pc=pc+signed(pppppppp)+1
01111111pppppppp     - bdf pc=pc+signed(pppppppp)+1
10aaaaaaaaaaaaaa     - jmp absolute
11aaaaaaaaaaaaaa     - jsr absolute
```

55.5.2 SDMA Instruction Set

This section describes all the useful instructions from the SDMA set.

Table 55-43. SDMA Instruction List

Instruction	Description	Page
ADD	Addition	ADD (Addition)
ADDI	Add with Immediate Value	ADDI (Add with Immediate Value)
AND	Logical AND	AND (Logical AND)
ANDI	Logical AND with Immediate Value	ANDI (Logical AND with Immediate Value)
ANDN	Logical AND NOT	ANDN (Logical AND NOT)
ANDNI	Logical AND with Negated Immediate Value	ANDNI (Logical AND with Negated Immediate Value)
ASR1	Arithmetic Shift Right by 1 Bit	ASR1 (Arithmetic Shift Right by 1 Bit)
BCLRI	Bit Clear Immediate	BCLRI1 (Bit Clear Immediate)
BDF	Conditional Branch if Destination Fault	BDF (Conditional Branch if Destination Fault)
BF	Conditional Branch if False	Functional Units Programming Model
BSETI	Bit Set Immediate	BSETI (Bit Set Immediate)
BSF	Conditional Branch if Source Fault	BSF (Conditional Branch if Source Fault)
BT	Conditional Branch if True	BT (Conditional Branch if True)
BTSTI	Bit Test immediate	BTSTI (Bit Test immediate)
CLRF	Clear Arm platform flags	CLRF (Clear Arm platform flags)
CMPEQ	Compare for Equal	CMPEQ (Compare for Equal)
CMPEQI	Compare with Immediate for Equal	CMPEQI (Compare with Immediate for Equal)
CMPHS	Compare for Higher or Same	CMPHS (Compare for Higher or Same)
CMPLT	Compare for Less Than	CMPLT (Compare for Less Than)
cpShReg	Update Context of PCU Registers and Flags	cpShReg (Update Context of PCU Registers and Flag)
DONE	DONE, Yield	DONE (DONE, Yield)
ILLEGAL	ILLEGAL Instruction	ILLEGAL (ILLEGAL Instruction)
JMP	Unconditional Jump Immediate	JMP (Unconditional Jump Immediate)
JMPR	Unconditional Jump	JMPR (Unconditional Jump)
JSR	Unconditional Jump to Subroutine Immediate	JSR (Unconditional Jump to Subroutine Immediate)
JSRR	Unconditional Jump to Subroutine	JSRR (Unconditional Jump to Subroutine)
LD	Load Register	LD (Load Register)
LDF	Load Register from Functional Unit	LDF (Load Register from Functional Unit)

Table continues on the next page...

Table 55-43. SDMA Instruction List (continued)

Instruction	Description	Page
LDI	Load Register with Immediate Value	LDI (Load Register with Immediate Value)
LDRPC	Load from RPC to Register	LDRPC (Load from RPC to Register)
LOOP	Hardware Loop	LOOP (Hardware Loop)
LSL1	Logical Shift Left by 1 Bit	LSL1 (Logical Shift Left by 1 Bit)
LSR1	Logical Shift Right by 1 Bit	LSR1 (Logical Shift Right by 1 Bit)
MOV	Logical Move	MOV (Logical Move)
NOTIFY	Notify to Arm platform	NOTIFY (Notify to Arm platform)
OR	Logical OR	OR (Logical OR)
ORI	Logical OR with Immediate Value	ORI (Logical OR with Immediate Value)
RET	Return from Subroutine	RET (Return from Subroutine)
REVB	Reverse Byte Order	REVB (Reverse Byte Order)
REVBLO	Reverse Low Order Bytes	Reverse Low Order Bytes(REVBLO)
ROR1	Rotate Right by 1 Bit	ROR1 (Rotate Right by 1 Bit)
RORB	Rotate Right by 1 Byte	RORB (Rotate Right by 1 Byte)
SOFTBKPT	Software Breakpoint	SOFTBKPT (Software Breakpoint)
ST	Store Register	ST (Store Register)
STF	Store Register in Functional Unit	STF (Store Register in Functional Unit)
SUB	Subtract	SUB (Subtract)
SUBI	Subtract with Immediate	SUBI (Subtract with Immediate)
TST	Test with Zero	TST (Test with Zero)
TSTI	Test Immediate	TSTI (Test Immediate)
XOR	Logical Exclusive OR	XOR (Logical Exclusive OR)
XORI	Exclusive OR with Immediate	XORI (Exclusive OR with Immediate)

55.5.2.1 ADD (Addition)

Operation:

```
GReg[r] ← GReg[s] + GReg[r]

T ← (GReg[r] == 0)
```

Assembler:

```
Syntax: add r,s

Example: add 0,3
```

ADD GReg[3] and GReg[0] and store the result in GReg[0]

CPU Flags: T

Cycles: 1

Description: Performs the ADDition of the source general register *s* and the destination general register *r*, and stores the result in the destination general register *r*. The T flag is set if the result of the operation is 0. It is cleared if the result is not 0.

Instruction Format:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	0	1	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.2 ADDI (Add with Immediate Value)

Operation:

```
GReg[r] ← GReg[r] + immediate

T ← (GReg[r] == 0)
```

Assembler:

```
Syntax: addi r,immediate

Example: add 6,112
```

ADD GReg[6] and decimal value 112 and store the result in GReg[6]

CPU Flags: T

Cycles: 1

Description: Adds a 0-extended immediate value to a general register; stores the result in the general register. The flag T is set when the result of the operation is 0; otherwise, it is cleared. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

 ...

11111110 - 254

11111111 - 255
```

55.5.2.3 AND (Logical AND)

Operation:

```
GReg[r] ← GReg[s] & GReg[r]
```

Assembler:

```
Syntax: and r,s

Example: and 1,2
```

AND GReg[1] and GReg[2] and store the result in GReg[1]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the AND of the source general register *s* and the destination general register *r*, and stores the result in the destination general register *r*.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	1	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.4 ANDI (Logical AND with Immediate Value)

Operation:

```
GReg[r] ← GReg[r] & immediate
```

Assembler:

```
Syntax: andi r,immediate

Example: andi 7,45
```

AND GReg[7] and decimal value 45 and store the result in GReg[7]

CPU Flags: unaffected

Cycles: 1

Description: Performs an AND between a 0-extended immediate value and a general register; stores the result in the general register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

55.5.2.5 ANDN (Logical AND NOT)

Operation:

```
GReg[r] ← ~GReg[s] & GReg[r]
```

Assembler:

```
Syntax:andn r,s

Example: andn 3,4
```

AND GReg[3] and NOT GReg[4] (bit inverted) and store the result in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the AND of the negation of the source general register *s* and the destination general register *r*, and stores the result in the destination general register *r*.

Instruction Format:

Table 55-44. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	1	0	s	s	s

Instruction Fields:

rrr /sss - destination register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.6 ANDNI (Logical AND with Negated Immediate Value)

Operation:

```
GReg[r] ← GReg[r] & ~immediate
```

Assembler:

```
Syntax: andni r,immediate

Example: andni 0,2
```

AND GReg[0] and decimal value -3 (inverted 32-bit value 2) and store the result in GReg[0]

CPU Flags: unaffected

Cycles: 1

Description: Performs an AND between the negation of a 0-extended 8-bit immediate value and a general register; stores the result in the general register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

55.5.2.7 ASR1 (Arithmetic Shift Right by 1 Bit)

Operation:

```
GReg[r]:{b31,b30,...,b1,b0} ← GReg[r]:{b31,b31,b30,...,b1}
```

Assembler:

```
Syntax: asr1 r

Example: asr1 3
```

divide by 2 the signed value of GReg[3] and store the result in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Shift the bits of any general register to the right and keep the same sign: The left bit (bit 31) is kept untouched.

Instruction Format:

Table 55-45. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	1	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.8 BCLRI1 (Bit Clear Immediate)

Operation:

```
GReg[r]:{b31,...,b(i+1),0,b(i-1),...,b0} ← GReg[r]:{b31,...,b(i+1),b(i),b(i-1),...,b0}
```

Assembler:

```
Syntax: bclri r,i

Example: bclri 1,12
```

clear bit 12 in GReg[1]

CPU Flags: Unaffected

Cycles: 1

Description: Clear the bit of register r specified by the 5-bit immediate field

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table continues on the next page...

0	0	0	0	0	r	r	r	0	0	1	i	i	i	i	i

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiii - immediate value:

```
00000 - 0

00001 - 1

...

11110 - 30

11111 - 31
```

55.5.2.9 BDF (Conditional Branch if Destination Fault)

Operation:

```
if (DF == 1) PC ← PC + 1 + displacement else PC ← PC + 1
```

Assembler:

```
Syntax:bdf label

Example: bdf LLL
```

Jump to LLL if DF is set, or go to the next instruction if DF is cleared; the displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: If flag DF is set, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag DF is cleared, no jump is performed: The next instruction is located at the next PC address.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	p	p	p	p	p	p	p	p

Instruction Fields:

ppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110 - 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 - (-2)

11111111 - (-1)
```

55.5.2.10 BF (Conditional Branch if False)

Operation:

```
if (T == 0)

PC ← PC + 1 + displacement

else

PC ← PC + 1
```

Assembler:

```
Syntax: bf label

Example: bf LLL
```

Jump to LLL if T is cleared, or go to the next instruction if T is set. The displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: Conditional branch: If flag T is cleared, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag T is set, no jump is performed: The next instruction is located at the next PC address.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	0	0	p	p	p	p	p	p	p	p

Instruction Fields:

pppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110 - 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 - (-2)

11111111 - (-1)
```

55.5.2.11 BSETI (Bit Set Immediate)

Operation:

```
GReg[r]:{b31,...,b(i+1),1,b(i-1),...,b0} ← GReg[r]:{b31,...,b(i+1),b(i),b(i-1),...,b0}
```

Assembler:

```
Syntax: bseti r,i

Example: bseti 6,5
```

Set bit 5 in GReg[6]

CPU Flags: Unaffected

Cycles: 1

Description: Sets bit number i in the selected General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	1	0	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiii - bit number field:

```
00000 - 0

00001 - 1

...

11110 - 30

11111 - 31
```

55.5.2.12 BSF (Conditional Branch if Source Fault)

Operation:

```
if (SF == 1) PC ← PC + 1 + displacement else PC ← PC + 1
```

Assembler:

```
Syntax: bsf label

Example: bsf LLL
```

Jump to LLL if SF is set, or go to the next instruction if SF is cleared. The displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: Conditional branch: If flag SF is set, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag SF is cleared, no jump is performed: The next instruction is located at the next PC address.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	0	p	p	p	p	p	p	p	p

Instruction Fields:

pppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110 - 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 - (-2)

11111111 - (-1)
```

55.5.2.13 BT (Conditional Branch if True)

Operation

```
if (T == 1)

PC ← PC + 1 + displacement

else

PC ← PC + 1
```

Assembler

```
Syntax: bt label

bt LLL
```

Jump to LLL if T is set, or go to the next instruction if T is cleared. The displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: Conditional branch: If flag T is set, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag T is cleared, no jump is performed: The next instruction is located at the next PC address.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	0	1	p	p	p	p	p	p	p	p

pppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110- 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 -  (-2)

11111111 - (-1)
```

55.5.2.14 BTSTI (Bit Test immediate)

Operation:

```
T ← GReg[r]:b(i)
```

Assembler:

```
Syntax: btsti r,i

Example: btsti 2,29
```

Test bit 29 in GReg[2] and copy its value in flag T

CPU flags: T

Cycles: 1

Description: T is loaded with the value of bit number i from the selected general register.

Instruction Format:

Table 55-46. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	1	1	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiii - bit number field:

```
0000 - 0

0001 - 1

...

11110 - 30

11111 - 31
```

55.5.2.15 CLRF (Clear Arm platform flags)

Operation:

```
 if (ff%2 == 0)

SF ← 0

if (ff/2 == 0)

DF ← 0
```

Assembler:

```
Syntax: clrf ff

Example: clrf 2
```

Clear flag SF and keep flag DF unchanged

CPU Flags: SF, DF

Cycles: 1

Description: Clears a selection of the Arm platform fault flags: SF, DF, both SF and DF or none can be cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	f	f	0	0	0	0	0	1	1	1

Instruction Fields:

ff - flags field:

```
00 - clear SF and clear DF

01 - clear DF

10 - clear

SF 11 - no clear
```

55.5.2.16 CMPEQ (Compare for Equal)

Operation:

```
T ← (GReg[s] == GReg[r])
```

Assembler:

```
Syntax: cmpeq r,s

Example: cmpeq 7,5
```

Compare GReg[7] and GReg[5] and set flag T if they are equal

CPU flags: T

Cycles: 1

Description: Subtracts the destination general register *r* from the source general register *s*, and sets T if the result is 0, clears T if the result is not 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	0	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.17 CMPEQI (Compare with Immediate for Equal)

Operation:

```
T ← (GReg[r] == immediate)
```

Assembler:

```
Syntax: cmpeqi r,immediate

Example: cmpeqi 2,13
```

Compare GReg[2] and decimal value 13 and set flag T if they are equal

CPU Flags: T

Cycles: 1

Description: Subtracts the 0-extended 8-bit immediate value from the general register, and sets T if the result is 0, clears T if the result is not 0. The immediate value is the low-order byte of the instruction.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - destination register field:

```
000 - GReg[0]
```

```
001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

55.5.2.18 CMPHS (Compare for Higher or Same)

Operation:

```
T ← (GReg[r] ≥ GReg[s])
```

Assembler:

```
Syntax: cmphs r,s

Example: cmphs 0,1
```

Compare GReg[0] and GReg[1] and set flag T if GReg[0] is higher than or equal to GReg[1]

CPU Flags: T

Cycles: 1

Description: Compares the destination general register *r* and the source general register *s*, and sets T if the destination general register *r* is higher than or equal to the source general register *s*, clears T otherwise. The comparison is unsigned.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	1	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.19 CMPLT (Compare for Less Than)

Operation:

```
T ← (GReg[r] < GReg[s])
```

Assembler:

```
Syntax: cmplt r,s

Example: cmplt 7,4
```

Compare GReg[7] and GReg[4] and set flag T if GReg[7] is lower than GReg[4]

CPU Flags: T

Cycles: 1

Description: Compares the destination general register *r* and the source general register *s*, and sets T if the destination general register *r* is lower than the source general register *s*, clears T otherwise. The comparison is signed.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	1	0	s	s	s

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]
```

```
011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.20 cpShReg (Update Context of PCU Registers and Flag)

Assembler:

```
Syntax: cpShReg
```

CPU Flags: none

Cycles: 1

Description: SF, RPC, T, PC,LM, EPC, DF, and SPC registers are updated according to the value of their corresponding bits in the context memory. This instruction must only be used in debug mode via the OnCE. It reverses the done *5* operation.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	0	1	1	1	0	0	0	1	0

55.5.2.21 DONE (DONE, Yield)

Operation:

```
if (jjj&6 == 2) HE[CCR] ← 0

if (jjj == 3) HI[CCR] ← 1

if (jjj == 4) EP[CCR] ← 0


if ((jjj == 0) && (NCP > CCP)) CCR ← NCR

else if ((jjj == 1) && (NCP >= CCP))

CCR ← NCR

else

CCR ← NCR
```

(CCR stands for Current Channel Register; NCR stands for Next Channel Register)

Assembler:

```
Syntax: done jjj

Example: done 3
```

Clear HE bit for the current channel, send an interrupt to the Arm platform for the current channel and reschedule.

CPU Flags: Unaffected

Cycles: Variable if a context switch is done, 1 otherwise

Description: Clears one of the channel enabling bits (HE or EP for the corresponding channel number) if required. Sends an interrupt to the corresponding Arm platform by setting the appropriate flag, if required (HI for the corresponding channel number). Reschedules according to the mode and the NCP (Next Channel Priority) and CCP (Current Channel Priority) values. According to the scheduling decision, the NCR (Next Channel Register) is copied to the CCR (Current Channel Register) and channel contexts are switched. If several channels with the same highest priority are pending, they are ordered by their number from 31 down to 0. The higher number is selected (for example, channel 26 is selected if channels 3, 12, 14, and 26 with the same highest priority are pending). If no flag is modified, the reschedule can allow the replacement of the current channel by another channel with a priority strictly greater than the current channel priority (yield). Or, it can allow the replacement of the current channel by another channel with a priority greater than or equal to the current channel priority (yieldge). In the latter case, the selected channel will always be the first one with the same priority, starting from channel number 31 down to channel 0 (the current channel does not belong to the set of selectable channels).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	j	j	j	0	0	0	0	0	0	0	0

jjj - Channel Flags field:

000 - No channel flags affected: Reschedule only if the next channel priority is greater than current channel priority (yield)

001 - No channel flags affected: Reschedule only if the next channel priority is greater than or equal to the current channel priority (yieldge)

010 - Clear HE for the current channel and reschedule 011 - Clear HE, set HI for the current channel and reschedule 100 - Clear EP for the current channel and reschedule

101 - Reserved for debug to copy relevant registers into context memory

110 - RESERVED

111 - RESERVED

For the scheduling rules, refer to Scheduler Functional Description. Every possible done instruction is further described as follows:

- done 0/yield is executed by a channel script when it accepts preemption by a higher priority channel;
- done 1/yieldge is executed by a channel script when it accepts preemption by a higher priority channel and it also accepts a roll-up with other channels that have the same priority;
- done 2 is executed by a channel script that was triggered by a Arm platform start via the Channel Start (SDMAARM_HSTART) register, when its task is completed and it requires termination;
- done 3 is executed by a channel script that was triggered by a Arm platform start via the Channel Start (SDMAARM_HSTART) register, when its task is completed, it requires termination and it needs to trigger an interrupt to the Arm platform upon closure;
- done 4 is executed by a channel script that was triggered by a DMA request, when its task is completed and it requires termination;
- done 5 is used in debug mode only; it copies the PCU registers and flags to the context memory of the current channel;

55.5.2.22 ILLEGAL (ILLEGAL Instruction)

Operation:

```
PC ← 0001
```

Assembler:

```
Syntax: illegal
```

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the Illegal instruction routine located at address 0001. All unauthorized instructions result in an Illegal instruction behavior; however, the ILLEGAL instruction must be used to guarantee software compatibility with future versions of the SDMA.

Instruction Format

Table 55-47. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

55.5.2.23 JMP (Unconditional Jump Immediate)

Operation:

```
PC ← absolute_address
```

Assembler:

```
Syntax: jmp label

Example: jmp LLL
```

The assembler translates the label to the exact address

CPU Flags:Unaffected

Cycles: 2

Description: Jumps to the absolute address contained the lower 14 bits of the instruction (the PC is a 14-bit register).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	a	a	a	a	a	a	a	a	a	a	a	a	a	a

aaaaaaaaaaaaaa - address field:

```
00000000000000 - 0

00000000000001 - 1

...

11111111111110 - 16382

11111111111111 - 16383
```

55.5.2.24 JMPR (Unconditional Jump)

Operation:

```
PC ← GReg[r]
```

Assembler:

```
Syntax: jmpr r

Example: jmpr 0
```

Jump to address stored in GReg[0]

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the absolute address contained in a General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	0	1	0	0	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.25 JSR (Unconditional Jump to Subroutine Immediate)

Operation:

```
RPC ← PC + 1

PC ← absolute_address
```

Assembler:

```
Syntax: jsr r

Example:jsr LLL
```

Jumps to subroutine starting at LLL; the assembler translates the label to exact address

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the subroutine located at the absolute address contained the lower 14 bits of the instruction (the PC is a 14-bit register).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	a	a	a	a	a	a	a	a	a	a	a	a	a	a

aaaaaaaaaaaaaa - address field:

```
00000000000000 - 0

00000000000001 - 1

...

11111111111110 - 16382

11111111111111 - 16383
```

55.5.2.26 JSRR (Unconditional Jump to Subroutine)

Operation:

```
RPC ← PC + 1

PC ← GReg[r]
```

Assembler:

```
Syntax: jsrr r

Example:jsrr 5
```

Jumps to subroutine located at address stored in GReg[5]

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the subroutine at address contained in a General Register

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	0	1	0	0	1

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.27 LD (Load Register)

Operation:

```
GReg[r] ← [GReg[b] + displacement]

if (transfer_error)

SF ← 1

else

SF ← 0
```

Assembler:

```
Syntax: ld r,(b,displacement)

Example: ld 1,(2,23)
```

Loads data into GReg[1]; the data is located at address obtained by adding decimal value 23 to GReg[2]

CPU Flags: SF

Cycles: 2+n where n is 0 for ROM, RAM or memory mapped registers, and n is the number of wait-states of the peripheral for a peripheral access

Description: Adds a 5-bit 0-extended displacement to a base address in General Register b; the result is the address of the data to fetch on the DM bus. The data received from the bus is stored in the destination General Register r. If an error occurs during the transfer, the flag SF is set, else it is cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	0	r	r	r	d	d	d	d	d	b	b	b

rrr / bbb - register field:

```
000 - GReg[0]

001 - GReg[1]

...

111 - GReg[7]
```

ddddd - displacement value:

```
00000 - 0

00001 - 1

...

11111 - 31
```

55.5.2.28 LDF (Load Register from Functional Unit)

Operation:

```
GReg[r] ← [fu_address]

if (transfer_error)

SF ← 1

else

SF ←0
```

fu_address is an 8-bit field and depends on addressed functional unit

Assembler:

```
Syntax: ldf r,fu_address

Example: ldf 0,13
```

Loads data coming from the Burst DMA register MD into GReg[0]; it is a 32-bit access with no prefetch

CPU Flags: SF

Cycles: 1+n where n is the number of wait-states that may be inserted by the functional unit

Description: Sends an 8-bit address on the Functional Unit Bus (FU bus) and stores the data received from the bus in the destination General Register r. If an error occurs during the transfer, the flag SF is set, else it is cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	r	r	r	f	f	f	f	f	f	f	f

See the following sections for more details of the LDF instruction usage with each functional unit:

- Burst DMA Read (ldf) for Burst DMA
- Peripheral DMA Read (ldf)-Read Mode for Peripheral DMA

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

ffffffff - functional unit source register and action (unspecified values are reserved):

```
00000000 - MSA

00000100 - MDA

00001001 - MD byte

00001010 - MD halfword

00001011 - MD word

00001100 - MS

00101001 - MD byte - prefetch

00101010 - MD halfword - prefetch

00101011 - MD word - prefetch

01000000 - DSA
```

11000000 - PSA

11001000 - PD

11010000 - PDA

11011000 - PD in copy mode (rrr contents are lost)

11101000 - PD - prefetch next data

11111111 - PS

55.5.2.29 LDI (Load Register with Immediate Value)

Operation:

GReg[r] ← immediate

Assembler:

Syntax: ldi r,immediate

Example: ldi 6,1

loads decimal value 1 into GReg[6]

CPU Flags: Unaffected

Cycles: 1

Description: Stores a 0-extended immediate value in a General Register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]

iiiiiiii - immediate value:

00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255

55.5.2.30 LDRPC (Load from RPC to Register)

Operation:

GReg[r] ← RPC

Assembler:

Syntax: ldrpc r

Example: ldrpc 3

copies RPC to GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Stores the contents of the RPC in a General Register. That instruction may be used to have more than one level of subroutines.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	0	1	0	1	0

Instruction Fields:

rrr - register field:

000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

```
101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.31 LOOP (Hardware Loop)

Operation:

```
if (ff%2 == 0)
   SF ← 0
if (ff/2 == 0)
  DF ← 0
if ((GReg[0] == 0) || (SF == 1) || (DF == 1))
   PC ← PC + loop_size + 1
else
   {
      SPC ← PC + 1
      EPC ← PC + loop_size + 1
      LM ← 1
      PC ← PC + 1
   }
```

during every instruction execution in the loop:

```
if ((SF == 1) || (DF == 1))
 {
    LM ← 0
    PC ← EPC
  }
else if ((PC + 1) == EPC)
 {
    GReg[0] ← GReg[0] - 1
    if (GReg[0] == 0)
      {
         LM ← 0
         PC ← EPC
```

```
      }
    else
        PC ← SPC
  }
else
  PC ← nextPC(instruction)
```

after the execution of the last instruction of the loop body:

```
if (GReg[0] == 0)
  T ← 1
else
  T ← 0
```

Assembler:

```
Syntax: loop n{,ff}
Example: loop 3,1
```

Executes GReg[0] times the instructions comprised between PC+1 and PC+3 (included); ff=1 clears the DF flag before starting the loop. When omitted, the ff field is set to 0 (clearing both SF and DF).

CPU Flags: LM[1:0], T

Cycles: 2 when the loop count (GReg[0]) is 0 or SF or DF is set at loop start, 1+1 when the loop starts but exits abnormally (SF or DF set inside the loop which adds 1 cycle to the offending load or store to jump to EPC), 1 when the loop is executed normally

Description: The loop instruction executes a sequence of instructions several times. The number of times is given by the contents of GReg[0], the loop counter. SDMA will jump to the first instruction after the end of the loop if the value in GReg[0] is 0. Otherwise the SDMA enters loop mode. It sets the most significant bit of the LM flag that will only be reset once the last instruction of the last loop is executed. The instructions in the loop are executed GReg[0] times. The management of fault flags (SF and DF) is as follows. When entering the hardware loop, SF and DF can be cleared according to the ff field of the instruction. After that operation, if any flag is still set the loop will not be executed. The SDMA will jump to the first instruction after the end of the loop without entering loop mode. During the execution of the loop, if any fault flag is set by a LD, LDF, ST, or STF instruction, the SDMA will immediately exit loop mode and jump to the first instruction after the end of the loop. In that case, GReg0 is not decremented for that last piece of the loop body execution (even if the SF or DF flag is set at the last instruction of the loop body). The T flag reflects the state of GReg[0] after the end of the loop, which is an indicator of the complete execution of the loop. If the loop exited because of an error (SF

or DF set), GReg[0] will not be 0 at the end of the loop, hence T will be cleared. If the loop executes without fault, GReg[0] will be 0 at the end of the loop, hence T will be set. The boundary case when a source or destination fault occurs at the last instruction of the last loop is considered as an anticipated exit of the loop, which causes the T flag to be cleared. If the last instruction executed before leaving the hardware loop also tries to modify the T flag, the flag is updated according to the value of GReg[0], NOT according to the result of the last executed instruction.

Limitations:

1. 1. Jump instructions (JMP, JMPR, JSR, JSRR, BF, BT, BSF, BDF) are not allowed inside the hardware loop.
2. 2. GReg[0] cannot be written to inside the hardware loop (it can be read).
3. 3. The empty loop (0 instruction in the body) is forbidden.
4. 4. If GReg[0] == 0 at the start of the loop, which causes a jump to EPC, the T flag is not updated.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	0	f	f	n	n	n	n	n	n	n	n

Instruction Fields:

```
ff - flags field:

00 - clear SF and clear DF

01 - clear DF

10 - clear SF

11 - no clear
```

nnnnnnnn - loop size

```
00000000 - empty loop: forbidden value

00000001 - 1 instruction in the loop

00000010 - 2 instructions in the loop

...

11111111 - 255 instructions in the loop
```

55.5.2.32 LSL1 (Logical Shift Left by 1 Bit)

Operation:

```
GReg[r]:{b30,...,b1,b0,0} ← GReg[r]:{b31,b30,...,b1,b0}
```

Assembler:

```
Syntax: lsl1 r

Example: lsl1 2
```

multiplies by 2 the value in GReg[2]

CPU Flags: Unaffected

Cycles: 1

Description: Shift the bits of any General Register to the left. The right bit (bit 0) is set to 0. No overflow is detected by the hardware.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	1	1

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.33 LSR1 (Logical Shift Right by 1 Bit)

Operation:

```
GReg[r]:{0,b31,b30,...,b1} ← GReg[r]:{b31,b30,...,b1,b0}
```

Assembler:

```
Syntax: lsr1 r

Example: lsr1 4
```

divides by 2 the unsigned value contained in GReg[4]

CPU Flags: Unaffected

Cycles: 1

Description: Shift the bits of any General Register to the right. The left bit (bit 31) is set to 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	0	1

Instruction Fields:

rrr - destination register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.34 MOV (Logical Move)

Operation:

```
GReg[r] ← GReg[s]
```

Assembler:

```
Syntax: mov r,s

Example: mov 4,0
```

copies GReg[0] to GReg[4]

CPU Flags: Unaffected

Cycles: 1

Description: Move the contents of the source General Register s to the destination General Register r.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	0	0	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.35 NOTIFY (Notify to Arm platform)

Operation:

```
 if (jjj & 4 == 0)
 {
   if (jjj&2 == 2)
      HE[CCR] ← 0
   if (jjj&1== 1)
       HI[CCR] ← 1
  }
else if (jjj == 4)
   EP[CCR] ← 0
else
```

(CCR stands for Current Channel Register)

Assembler:

```
Syntax: notify jjj

Example: notify 3
```

clears the HE bit for the current channel and sends an interrupt to the Host for the current channel

CPU Flags: Unaffected

Cycles: 1

Description: Clears one of the channel enabling bits (HE or EP for the corresponding channel number) if required, sends an interrupt to the corresponding Arm platform by setting the appropriate flag if required (HI for the corresponding channel number).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	j	j	j	0	0	0	0	0	0	0	1

jjj - Channel Flags field:

```
000 - unused

001 - set HI for the current channel

010 - clear HE for the current channel

011 - clear HE, set HI for the current channel

100 - clear EP for the current channel

101 - RESERVED

110 - RESERVED

111 - RESERVED
```

55.5.2.36 OR (Logical OR)

Operation:

```
GReg[r] ← GReg[s] | GReg[r]
```

Assembler:

```
Syntax: or r,s

Example: or 3,6
```

ORs GReg[3] and GReg[6] and stores the result in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the OR of the source General Register s and the destination General Register r, and stores the result in the destination General Register r.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	0	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.37 ORI (Logical OR with Immediate Value)

Operation:

```
GReg[r] ← GReg[r] | immediate
```

Assembler:

```
Syntax: ori r,immediate

Example: ori 1,56
```

ORs GReg[1] and the decimal value 56 and stores the result in GReg[1]

CPU Flags: unaffected

Cycles: 1

Description: Performs an OR between a 0-extended 8-bit immediate value and a General Register; stores the result in the General Register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

55.5.2.38 RET (Return from Subroutine)

Operation:

```
PC ← RPC
```

Assembler:

```
Syntax: ret
```

CPU Flags: Unaffected

Cycles: 2

Description: Return from subroutine.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

55.5.2.39 REVB (Reverse Byte Order)

Operation:

```
GReg[r]:{B3,B2,B1,B0} ← GReg[r]:{B0,B1,B2,B3}
```

Assembler:

```
Syntax: revb r

Example:  revb 5
```

reverses bytes order in GReg[5]

CPU Flags: Unaffected

Cycles: 1

Description: Reverse the byte order of any General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	0	0	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.40 Reverse Low Order Bytes(REVBLO)

Operation:

```
GReg[r]:{B3,B2,B0,B1} ← GReg[r]:{B3,B2,B1,B0}
```

Assembler:

```
Syntax: revblo r

Example: revblo 0
```

reverses low order bytes in GReg[0]

CPU Flags: Unaffected

Cycles: 1

Description: Reverse both low order bytes of any General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	0	0	1

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.41 ROR1 (Rotate Right by 1 Bit)

Operation:

```
GReg[r]:{b0,b31,b30,...,b1} ← GReg[r]:{b31,b30,...,b1,b0}
```

Assembler:

```
Syntax: ror1 r

Example: ror1 3
```

rotates bits to the right in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Rotate the bits of any General Register to the right.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	0	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.42 RORB (Rotate Right by 1 Byte)

Operation:

```
GReg[r]:{B0,B3,B2,B1} ← GReg[r]:{B3,B2,B1,B0}
```

Assembler:

```
Syntax: rorb r

Example: rorb 2
```

rotates bytes to the right in GReg[2]

CPU Flags: Unaffected

Cycles: 1

Description: Rotate the bytes of any General Register to the right.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	0	1	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.43 SOFTBKPT (Software Breakpoint)

Operation:

Stops the current script and enters debug mode

Assembler:

```
softbkpt
```

CPU Flags: Unaffected

Description: When the core executes this instruction, it has the same effect as receiving a debug request from the OnCE or via the external debug request input: the script execution halts, the PCU enters its debug state and waits for the OnCE commands that are described in OnCE and Real-Time Debug.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

55.5.2.44 ST (Store Register)

Operation:

```
[GReg[b] + displacement] ← GReg[r]

if (transfer_error)
```

```
 DF ← 1

 else

DF ← 0
```

Assembler:

```
Syntax: st r,(b,displacement)

Example: st 7,(0,9)
```

stores the value from GReg[7] into memory at address obtained by adding decimal value 9 to GReg[0]

CPU Flags: DF

Cycles: 2+n where n is 0 for ROM, RAM or memory mapped registers, and n is the number of wait-states of the peripheral for a peripheral access

Description: Adds a 5-bit 0-extended displacement to a base address in General Register b; the result is the address of the data to store on the DM bus. The data sent on the bus comes from the source General Register r. If an error occurs during the transfer, the flag DF is set, else it is cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	1	r	r	r	d	d	d	d	d	b	b	b

Instruction Fields:

rrr / bbb - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

ddddd - displacement value:

```
00000 - 0

00001 - 1
```

```
...

11111 - 31
```

55.5.2.45 STF (Store Register in Functional Unit)

Operation:

```
[fu_address] ← GReg[r] 0

if (transfer_error) 0

DF ← 1 0

else 0

DF ← 0
```

fu_address is an 8-bit field

Assembler:

```
Syntax: stf r,fu_address

Example: stf 3,0x2B
```

stores the 32-bit contents of GReg[3] to the Burst DMA register MD; waits until the flush to external memory is completed

CPU Flags: DF

Cycles: 1+n where n is the number of wait-states that may be inserted by the functional unit

Description: Sends an 8-bit address on the Functional Unit Bus (FU bus) and sends the contents of the source General Register r on the bus. If an error occurs during the transfer, the flag DF is set, else it is cleared.

Table 55-48. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	1	r	r	r	f	f	f	f	f	f	f	f

See the following sections for more details of the STF instruction usage with each functional unit:

- Burst DMA Write (stf) for Burst DMA
- Peripheral DMA Write (stf)-Write Mode for Peripheral DMA

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

ffffffff - functional unit destination register and action (unspecified values are reserved):

00000000 - MSA in incremented mode

00000100 - MDA in incremented mode

00001001 - MD byte

00001010 - MD halfword

00001011 - MD word

00001100 - clear MS error flag

00001111 - MS

00010000 - MSA in frozen mode

00010100 - MDA in frozen mode

00011000 - MD in copy mode - number of words in rrr

00100000 - MSA in incremented mode - start prefetch

00101000 - MD no data - flush

00101001 - MD byte - flush

00101010 - MD halfword - flush

00101011 - MD word - flush

00110000 - MSA in frozen mode - start prefetch

11000001 - PSA in frozen mode - 8-bit data width

11000010 - PSA in frozen mode - 16-bit data width

11000011 - PSA in frozen mode - 32-bit data width

11000101 - PSA in incremented mode - 8-bit data width

11000110 - PSA in incremented mode - 16-bit data width

11000111 - PSA in incremented mode - 32-bit data width

11001000 - PD

11001001 - PSA in decremented mode - 8-bit data width

11001010 - PSA in decremented mode - 16-bit data width

11001011 - PSA in decremented mode - 32-bit data width

11001100 - clear PS error flag

11001101 - PSA data width becomes 8-bit

11001110 - PSA data width becomes 16-bit

11001111 - PSA data width becomes 32-bit

11010001 - PDA in frozen mode - 8-bit data width

11010010 - PDA in frozen mode - 16-bit data width

11010011 - PDA in frozen mode - 32-bit data width

11010101 - PDA in incremented mode - 8-bit data width

11010110 - PDA in incremented mode - 16-bit data width

11010111 - PDA in incremented mode - 32-bit data width

11011001 - PDA in decremented mode - 8-bit data width

11011010 - PDA in decremented mode - 16-bit data width

11011011 - PDA in decremented mode - 32-bit data width

11011101 - PDA data width becomes 8-bit

11011110 - PDA data width becomes 16-bit

11011111 - PDA data width becomes 32-bit

11100001 - PSA in frozen mode - 8-bit data width - prefetch data

11100010 - PSA in frozen mode - 16-bit data width - prefetch data

11100011 - PSA in frozen mode - 32-bit data width - prefetch data

11100101 - PSA in incremented mode - 8-bit data width - prefetch data

11100110 - PSA in incremented mode - 16-bit data width - prefetch data

11100111 - PSA in incremented mode - 32-bit data width - prefetch data

11101001 - PSA in decremented mode - 8-bit data width - prefetch data

11101010 - PSA in decremented mode - 16-bit data width - prefetch data

11101011 - PSA in decremented mode - 32-bit data width - prefetch data

11101101 - PSA data width becomes 8-bit - prefetch data

11101110 - PSA data width becomes 16-bit - prefetch data

11101111 - PSA data width becomes 32-bit - prefetch data

11111111- PS

55.5.2.46 SUB (Subtract)

Operation:

```
GReg[r] ← GReg[r] - GReg[s]

T ← (GReg[r] == 0)
```

Assembler:

```
Syntax: sub r,s

Example: sub 4,7
```

SUBtracts GReg[7] from GReg[4] and stores the result in GReg[4]

CPU Flags: T

Cycles: 1

Description: Subtracts the source General Register s from the destination General Register r, and stores the result in the destination General Register r. The T flag is set if the result of the operation is 0; it is cleared if the result is not 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	0	0	s	s	s

Instruction Fields:

rrr / sss - register fields:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.47 SUBI (Subtract with Immediate)

Operation:

```
GReg[r] ← GReg[r] - immediate

T ← (GReg[r] == 0)
```

Assembler:

```
Syntax: sub r,immediate

Example: sub 1,255
```

SUBtracts decimal value 255 from GReg[1] and stores the result in GReg[1]

CPU Flags: T

Cycles: 1

Description: Subtracts a 0-extended 8-bit immediate value from a General Register; stores the result in the General Register. The flag T is set when the result of the operation is 0; otherwise, it is cleared. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

55.5.2.48 TST (Test with Zero)

Operation:

```
T ← ((GReg[s] & GReg[r]) != 0)
```

Assembler:

```
Syntax: tst r,s

Example: tst 2,3
```

ANDs GReg[2] and GReg[3] and sets T if the result is non-null

CPU Flags: T

Cycles: 1

Description: Performs the AND of the source General Register s and the destination General Register r, and sets T if the result is not 0, clears T if the result is 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	0	0	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

55.5.2.49 TSTI (Test Immediate)

Operation:

```
T ← ((GReg[r] & immediate) != 0)
```

Assembler:

```
Syntax: tsti r,immediate

Example: tsti 5,13
```

ANDs GReg[5] and decimal value 13 and sets T if the result is non-null

CPU Flags: T

Cycles: 1

Description: Performs the AND of a 0-extended 8-bit immediate value and the destination General Register r, and sets T if the result is not 0, clears T if the result is 0. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - destination register field:

```
000 - GReg[0]
```

```
001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

55.5.2.50 XOR (Logical Exclusive OR)

Operation:

```
GReg[r] ← GReg[s] ^ GReg[r]
```

Assembler:

```
Syntax: xor r,s

Example: xor 0,3
```

XORs GReg[0] and GReg[3] and stores the result in GReg[0]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the eXclusive OR of the source General Register s and the destination General Register r, and stores the result in the destination General Register r.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	0	1	0	s	s	s

Instruction Fields:

rrr / sss - register field:

000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]

55.5.2.51 XORI (Exclusive OR with Immediate)

Operation:

GReg[r] ← GReg[r] ^ immediate

Assembler:

Syntax: xori r,immediate

Example: xor 7,5

XORs GReg[5] and decimal value 5 and stores the result in GReg[7]

CPU Flags: Unaffected

Cycles: 1

Description: Performs an eXclusive OR between a 0-extended 8-bit immediate value and a General Register; stores the result in the General Register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

```
100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

55.5.2.52 YIELD, YIELDGE (DONE, Yield)

By default, unsupported assembler syntax. Can be aliased to the corresponding done instructions (yield = done 0; yieldge = done 1). Refer to the done instruction description DONE (DONE, Yield) .

55.6 Software Restrictions

55.6.1 Unsupported Burst DMA Access Sequence

The SDMA does not support triggering a pre-fetch followed by a flush of the Burst DMA without reading or writing any data. If the flush occurs while the background pre-fetch DMA operation is still in progress, it could result in un-defined behavior.

An example of the sequence which could result in undefined results is shown in the following example:

Instruction sequence not supported

```
stf r1, MSA|PF          ; Update source address, triggers data pre-fetch in the
                        ; background
mov R0,R0               ; Execute multiple assembly instructions, none of which
                        ; read
mov R0,R0               ; or write data to/from MD
stf MD|SZ0|FL           ; Flush FIFO without writing data. If the pre-fetch is still
                        ; in progress when this instruction is executed, there
                        ; could be undefined operation
```

A work-around to avoid any undesirable results is to first read MD to ensure the pre-fetch is complete before the flush is attempted.

Work-Around to previous example

```
stf r1, MSA|PF          ; Update source address, triggers data pre-fetch.
mov R0,R0               ; Execute multiple assembly instructions, none of which
                        ; read
mov R0,R0               ; or write data to/from MD
ldf r2, MD              ; dummy read of MD to ensure pre-fetch is complete
                        ; before the next instruction
stf MD|SZ0|FL           ; Flush FIFO without writing data
```

55.7 Application Notes

55.7.1 Data Structures for Boot Code and Channel Scripts

SDMA boot code downloads the different channel contexts and the scripts that will be executed on SDMA channels during the application.

The boot code is run after reset when channel 0 is started by the Arm platform. The boot code is also known as channel 0 script.

The boot code is based on the Channel Control Block (CCB) and Buffer Descriptor (BD) mechanisms that are data structures located into the Arm platform memory space. With these data structures, it is possible to instruct SDMA to download scripts and contexts but also to dump a context or a script to a destination data buffer. Channel scripts also use the CCB and BD data structures to pass instructions and/or pointers to data to be copied.

The format, processing, and field definition of the CCB and BD are defined and performed entirely by the software script rather than the SDMA hardware. An overview of the format and structure is provided here, but for complete details refer to the SDMA software documentation (see SDMA Scripts).

The CCB and BD data structures are accessed by SDMA using DMA and processed by the SDMA scripts. The ROM contains common sub-routines for processing these data structures which may be called by the bootload and channel scripts.

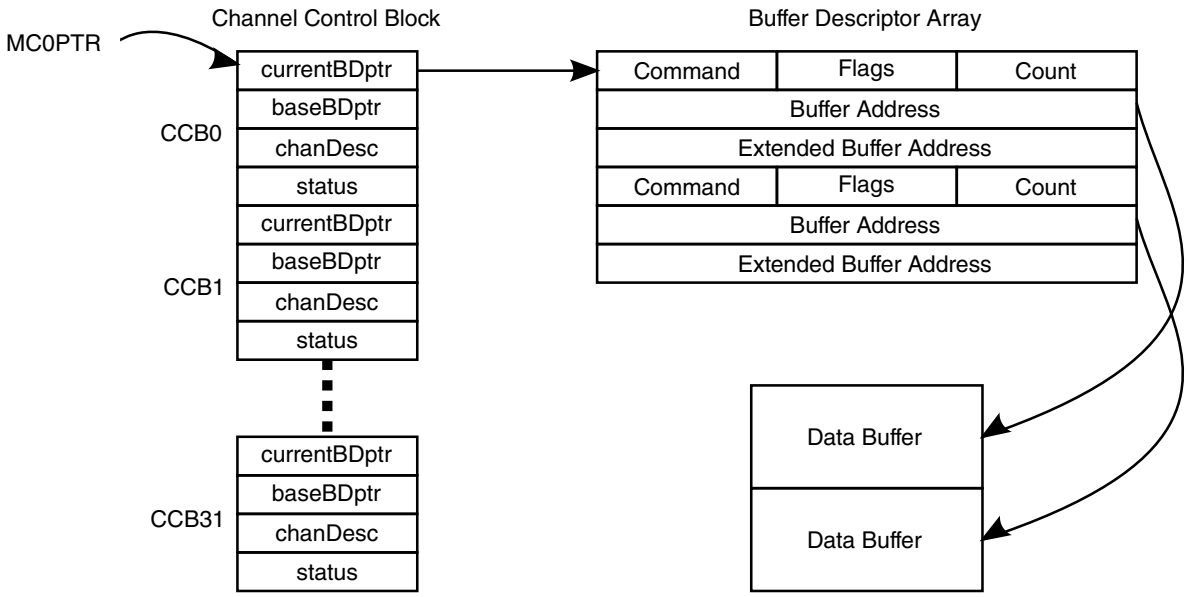


Figure 55-16. Data Structures Layout

The previous figure shows an example how these data structures are linked to pass command and pointers to data buffers. The SDMA's MC0PTR register holds the base address of the Channel 0 Control Block (CCB0). The Channel 0 control block holds a pointer to the array of buffer descriptors. The buffer descriptors are used to tell the channel 0 (boot channel) what to do as described Buffer Descriptor Format.

55.7.1.1 Buffer Descriptor Format

Buffer descriptors are three longs (32-bit words) in size as, shown in the figure found here.

A buffer descriptor describes the properties of the data buffer it points to. The buffer descriptors can be used for linear or circular data buffers in the Arm platform processor memory. The CCB contains a pointer to the base BD as well as the current BD.

Table 55-49. Buffer Descriptor

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Command								-	-	L	R	I	C	W	D	Count															
Buffer Address																															
Extended Buffer Address																															

Table 55-50. Buffer Descriptor Field Descriptions

Field	Description
31-24 Command	Command. The command field is used to differentiate operations performed within a script when the script accesses this particular buffer descriptor. The use of this field can be defined by the script. The command values defined for the bootload script are defined in Buffer Descriptor Commands for Bootload scripts. Refer to the individual script definition in script library documents in SDMA Scripts for command field definitions for other scripts.
23	Reserved
22	Reserved
21 L	Last Buffer Descriptor: This bit is set in SDMA IPC scripts to indicate to the receiving Core that the transfer has ended. Whenever the source finishes transferring the count it wanted to transfer, it sets LAST_BIT in the destination BD, to let the destination know that transfer is over.This bit also tells the destination software that when it processes the destination BDs, they need not process any BD after the BD with the LAST_BIT set.For example, when the DSP prepares a single buffer descriptor with count equals to 25 and Arm platform prepares a single buffer descriptor with count equals 100. When 25 bytes have been transferred from DSP to Arm platform, the DSP buffer descriptor is normally closed while the Arm platform buffer descriptor will have the L bit set and the byte count updated to 25.
20 R	erroR. Indicates an error occurred on the channel's buffer descriptor requested command. Some scripts may overwrite the command field with an error code indicating the source of the error. 0 No Error 1 Error
19 I	Interrupt. When SDMA has finished to process data transfer attached to this buffer descriptor, send an interrupt to the Arm platform. 0 No Interrupt 1 Interrupt the processor when BD is complete
18 C	Continuous. This buffer is allowed to receive multiple transmit buffers or is allowed to transmit to multiple receive buffers.The Continuous bit is decoded at the end of the processing of a BD to determine if the SDMA script must open a new BD to potentially continue the data transfer. 0 No further buffer descriptors 1 SDMA should move to the next Buffer descriptor after this one
17 W	Wrap. Indicates if this buffer descriptor is the last one for the channel control block. When encountering this bit set, the SDMA scripts updates the CurrentBD pointer to point to the first Buffer Descriptor of the array. This bit is set if the Arm platform wants to organize the array of BD in a circular way (like a ring). When all BD have been processed and if Wrap bit and CONtinuous bit are set in the last BD, the SDMA script will wrap around and it will try to re-open the first BD. 0 No Error 1 Wrap to first buffer descriptor after this one is processed.
16 D	D - "Done": bit 16: indicates the "ownership" of the buffer descriptor. When D=0 the host owns the buffer descriptor; when D=1 SDMA owns the buffer descriptor. In the case of the channel 0, D=1 indicates the SDMA has not yet processed this buffer, D=0 indicates the SDMA has processed this buffer. 0 Arm platform owns the buffer. 1 SDMA owns the buffer
15-0 Count	Count. the count field (bit 15-0) indicates the size of the data to be transmitted, the size of the data buffer pointed to by the buffer descriptor. The SDMA memory structure is different for program memory (16-bits shorts/half-words) and data memory (32-bits long). For channel 0 buffer descriptors, Count is expressed in 16-bit half-words when PM is addressed and in 32-bit words when DM is addressed. Count is typically expressed in bytes for other channel scripts, but the unit is dependant on the script.
31-0	Buffer address. Address pointer to the data buffer.
31-0	Extended buffer address. Additional pointer or other information required by some scripts.

The buffer descriptors form an array of programmable size. If the last buffer descriptor is marked by the Wrap flag-bit W=1, the array of buffer descriptor is treated as a ring with some logically continuous portion owned by the Arm platform with D=0, and the remainder owned by the SDMA with D=1. The count field of the buffer descriptor indicates how much data has been transmitted.

If Arm platform has prepared 3 buffers to be filled by the SDMA script, it has also prepared 3 BD, one for each buffer. The *Cont* and *Wrap* bits are used to organize the buffers in a circular way. For example, *CONT*inous bit is set to 1 in the 2 first BDs and *Wrap* is set in the 3rd BD. The SDMA script opens and processes BD#1. Since *CONT*inous bit is set for this BD, the SDMA will open the second BD and it will process it. Each time a BD is processed, its *Done* bit is reset by the SDMA. After the 3rd BD, if *CONT*inous is not set but if *Wrap* is set, the SDMA script stops here and the next time the channel will be triggered, the script will open the BD pointed by the currentBDptr pointer of the CCB and it will correspond to the first buffer descriptor.

If the *CONT*inous bit and *Wrap* bits are both set in the 3rd BD, the script will close it and it will try to open the first BD. An error may occur at this point if the BD#1 has already been processed and its *Done* bit is 0. The SDMA script cannot process a BD with a *Done* bit to 0. It means the BD is not ready to be processed. To avoid this situation, the *CONT*inous bit should not be set for the last BD if *Wrap* is set, and the Interrupt flag must set for the last BD. It will warn the owner of the BD that all the BDs have been processed and it has to re-set to 1 the *Done* bit of all the BD's if it desires the SDMA to fill them again. Basically, if the Arm platform expects the SDMA to fill up the buffers in a circular fashion, then it's the responsibility of the Arm platform to set the *Done* bit of a buffer descriptor at an appropriate time.

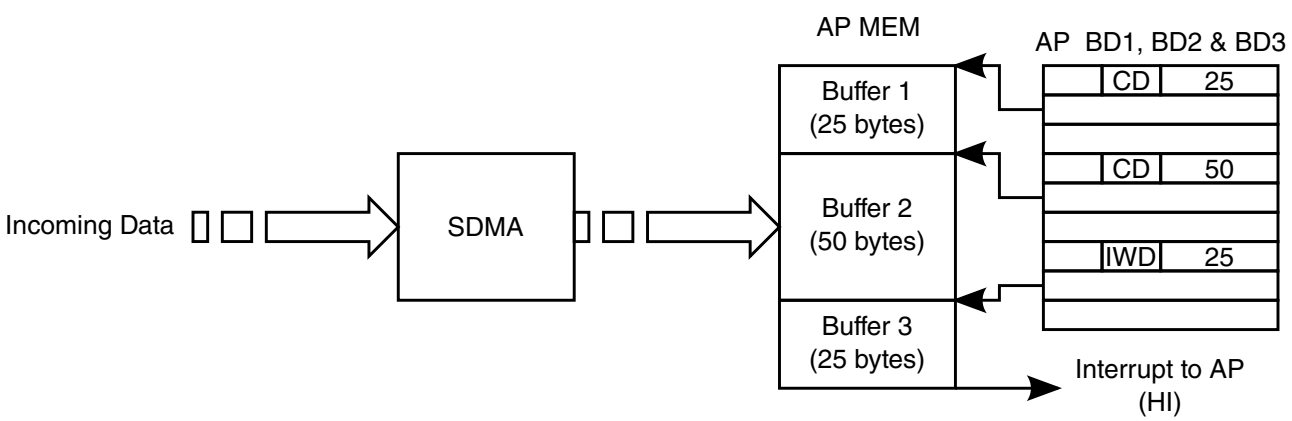


Figure 55-17. Buffer Descriptor Flow

The previous figure shows an example buffer descriptor flow. When the incoming data is stored and fills the first buffer of 25 bytes, the SDMA script opens the second BD because the CONTinuous bit was set. Then next incoming data is put in the second buffer. After receiving 50 bytes, the second buffer descriptor is also closed. The Done bit is reset and the third BD is opened. After receiving another 25 bytes, the third buffer is full and an interrupt is sent to the Arm platform because the Interrupt flag is set in the 3rd BD. The CONTinuous flag is not present the transfer is over. The next time the script will be triggered, the BD to be opened will be the first buffer descriptor since the Wrap flag was set in the 3rd BD. It is the Arm platform responsibility to set the Done bit of all the BD if it wants to use the same buffers.

55.7.1.2 Buffer Descriptor Commands for Bootload scripts

The command field of the buffer descriptor is defined separately for each script.

The following table lists the buffer descriptor commands defined for the channel 0 bootload script.

Table 55-51. Channel Zero Buffer Descriptor Commands

Command Field (binary)	Command	Description	Buffer Address	Extended Buffer Address
0000_0001 (0x01)	C0_SET_DM	Load SDMA data memory (RAM) from Arm platform memory buffer	Arm platform memory source address	SDMA memory destination address
0000_0010 (0x02)	C0_GET_DM	Copy SDMA data memory (RAM) to Arm platform memory buffer	Arm platform memory destination address	SDMA memory source address
0000_0100 (0x04)	C0_SET_PM	Load SDMA program memory (RAM) from Arm platform memory buffer	Arm platform memory source address	SDMA memory destination address
0000_0110 (0x06)	C0_GET_PM	Copy SDMA program memory (RAM) to Arm platform memory buffer	Arm platform memory destination address	SDMA memory source address
cccc_c111 (0x07 | CHN)	C0_SETCTX	Load Context for channel cccc into SDMA RAM from Arm platform memory buffer	Arm Platform memory source address	-
cccc_c011 (0x03 | CHN)	C0_GETCTXT	Copy Context for channel ccccc from SDMA RAM to Arm platform memory buffer	Arm platform memory destination address	-

The Channel 0 bootload commands are summarized as follows:

- C0_SET_[PM-DM]: load the buffer descriptor data in the SDMA local memory at the address pointed to by the "extended buffer address" field. The SDMA RAM can be seen as a Program Memory (PM, 16-bit address) or Data Memory (DM 32-bit address). When C0_SET_PM is used, the count field is expressed in "shorts" (16-bit

half words), this command can be used to download scripts. When C0_SET_DM is used, the count field is expressed in "long" (32-bit words), this command can be used to download channel contexts to the context channel area in RAM.

- C0_GET_[PM-DM]: write to the buffer descriptor's data buffer the content of the SDMA local memory from the address pointed to by the "extended buffer address" field for the length defined by the count in the buffer descriptor. C0_GET_PM is used to dump some part of the Program Memory (may be used to dump context of a channel), therefore count is expressed in "shorts"; while C0_GET_DM is used to dump to the buffer descriptor's data buffer, so the count field is in "longs."
- C0_SETCTX: load a context into the SDMA context page area. The handling script decodes the channel number from the 5 MSB of the command field of the buffer descriptor. Using the channel number the script computes the offset of the context data pointer for the channel relative to the context page base to use as the destination address in SDMA memory. Then the C0_SET_DM command explained above is invoked to load SDMA RAM from memory. The counter indicates the size in words of the context structure.
- Command value: (in binary) cccc c111, where ccccc is the channel number (5 bits). For instance, 0x0F means set context for channel 1, 0xFF means set context for channel 31.
- C0_GETCTX: write to the buffer descriptor's data buffer the content of the SDMA context page area. The handling script decodes the channel number from the 5 MSB of the command field of the buffer descriptor. Using this channel number, the script computes the offset of the context data pointer for the channel relative to the context page base to use as the source address for the copy. Then the C0_GET_DM command explained above is invoked to copy the context to memory. The counter indicates the size in words of the context structure.
- Command value: (in binary): cccc c011, where ccccc is the channel number (5 bits). For instance, 0x03 means get context of channel 1, 0xFB means get context of channel 31.

NOTE

To download channel context, C0_SETDM and C0_SETCTXT command can be used but the second one is easier because the channel number is embedded into the command field, whereas with the C0_SETDM, the pointer to the channel context area must be written into the extended buffer address field of the buffer descriptor.

55.7.1.3 Example of Buffer Descriptors for Channel 0.

Figure 55-19 illustrates the buffer descriptors that must be set in Arm platform memory space, before execution of boot code, to download contexts and scripts of channels 1, 4, and 10. After boot code execution, SDMA memory will be populated with the different contexts and scripts as presented in the following figure.

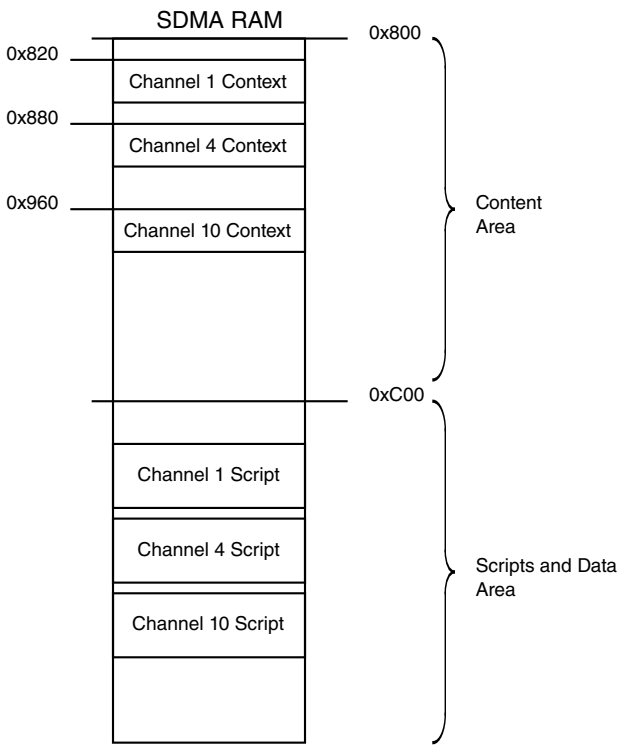
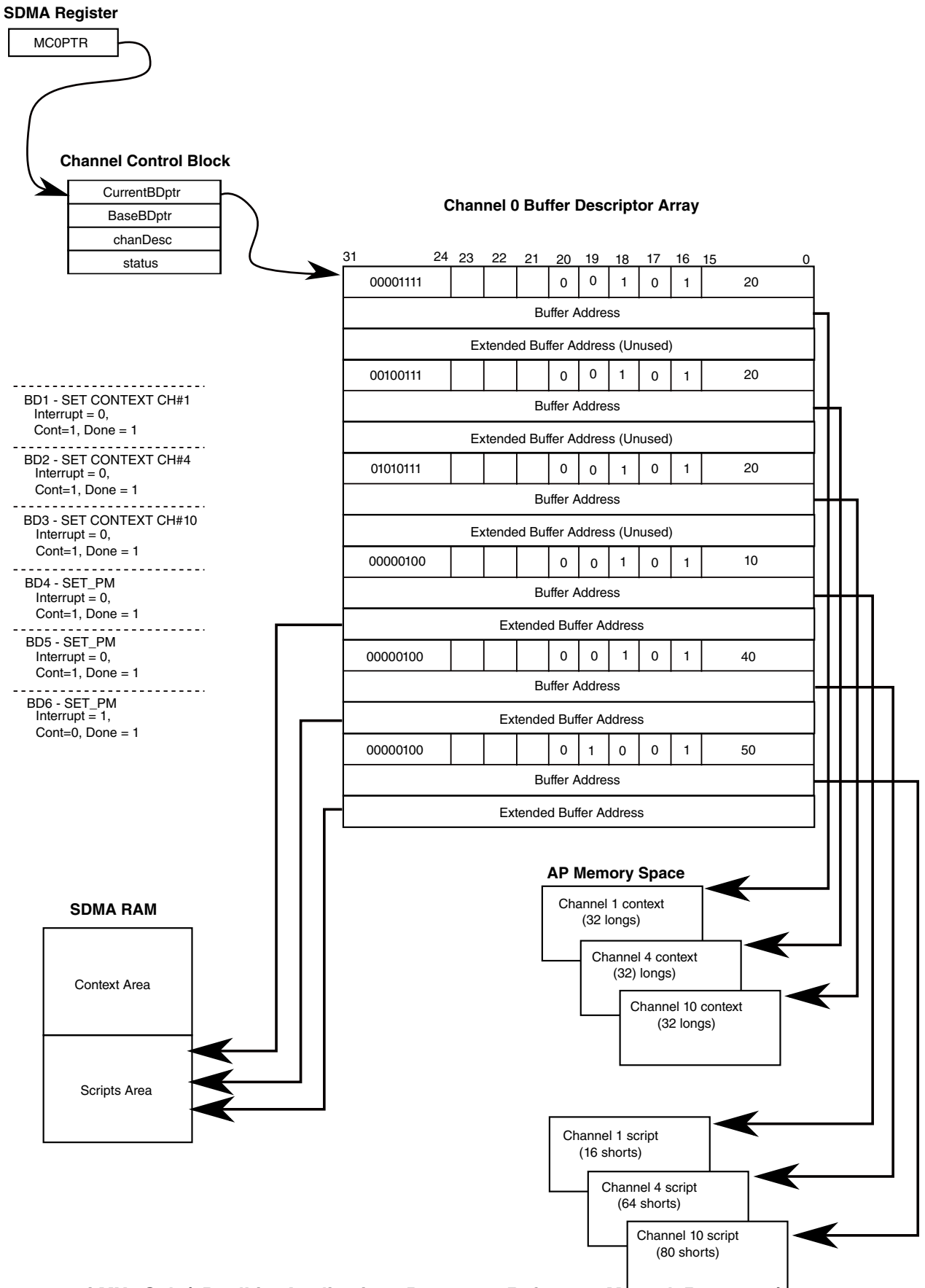


Figure 55-18. Example of SDMA RAM After Boot Session


SDMA Register
MC0PTR
Channel Control Block
CurrentBDptr
BaseBDptr
chanDesc
status
Channel 0 Buffer Descriptor Array
31 24 23 22 21 20 19 18 17 16 15 0
00001111 0 0 1 0 1 20
Buffer Address
Extended Buffer Address (Unused)
00100111 0 0 1 0 1 20
Buffer Address
Extended Buffer Address (Unused)
01010111 0 0 1 0 1 20
Buffer Address
Extended Buffer Address (Unused)
00000100 0 0 1 0 1 10
Buffer Address
Extended Buffer Address
00000100 0 0 1 0 1 40
Buffer Address
Extended Buffer Address
00000100 0 1 0 0 1 50
Buffer Address
Extended Buffer Address
BD1 - SET CONTEXT CH#1
Interrupt = 0,
Cont=1, Done = 1
BD2 - SET CONTEXT CH#4
Interrupt = 0,
Cont=1, Done = 1
BD3 - SET CONTEXT CH#10
Interrupt = 0,
Cont=1, Done = 1
BD4 - SET_PM
Interrupt = 0,
Cont=1, Done = 1
BD5 - SET_PM
Interrupt = 0,
Cont=1, Done = 1
BD6 - SET_PM
Interrupt = 1,
Cont=0, Done = 1
AP Memory Space
Channel 1 context
(32 longs)
Channel 4 context
(32) longs)
Channel 10 context
(32 longs)
SDMA RAM
Context Area
Scripts Area
Channel 1 script
(16 shorts)
Channel 4 script
(64 shorts)
Channel 10 script
(80 shorts)

55.7.1.4 Channel Context

There are 32 channel context memory structures pointed to by the local save area pointer. These channel context memory structures are fixed.

The script in the SDMA computes the memory offset for a given channel based on the structure length and channel number. Figure below shows the structure of the channel context as it is saved in the SDMA local memory (RAM).

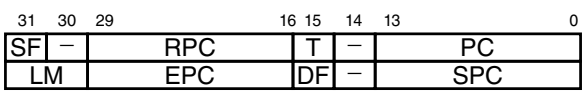
A channel context consists in 24 words, one per register. A total of 32 words are reserved for every channel. The additional 8 words are called scratch ram and they are dedicated to each channel. This memory area is commonly used for stack management.

The structure is divided in 4 areas:

- Channel status registers
- General purpose registers
- Functional units state registers reflecting the state of the Arm platform DMAs (Burst and Peripheral DMA).
- Scratch RAM

The details of the channel context status registers are described in the following figure.

The PC field of the first long register must point to the SDMA RAM address where the script that will be executed on the channel is located and this value equals the one stored in the extended buffer address of the buffer descriptor with C0_SETPM command.



SF: Source fault while loading data
RPC: Return program counter
T: Test bit: status of arithmetic and test instructions
PC: Program counter
LM: Loop mode
EPC: Loop end program counter
DF: Destination fault while storing data
SPC: Loop Start program counter

Figure 55-20. SDMA State Registers (ShPC, ShLoop)

55.7.2 Typical Data Transfer Supported by SDMA DMA Units

This section presents a library of SDMA scripts that perform data transfers through the peripheral DMA and the burst DMA units.

The Arm platform memory and peripherals are devices that either the peripheral DMA or the burst DMA can access. The scripts are given for a peripheral DMA whose address registers are programmed in incremented mode when internal memory is involved. See the following table for the summary.

Table 55-52. Typical Data Transfers Summary

Data Transfer	Peripheral DMA	Burst DMA	Comments
Arm platform External Memory ↔ Arm platform External Memory		3	Copy mode Script example, see Burst DMA Unit Copy Mode and External Memory to External Memory.
Arm platform Peripheral ↔ Arm platform Peripheral	3		Copy mode if same data path width Script example, see Peripheral to Peripheral Transfer.
Arm platform External Memory ↔ Arm platform Peripheral	3	3	Data transit through SDMA Script example, see Transfer Between Peripheral and External Memory.
Arm platform External Memory ↔ Arm platform Internal Memory		3	Copy mode Script example, see Transfer Between External Memory and Internal Memory.
Arm platform Internal Memory ↔ Arm platform Internal Memory		3	Copy mode Script example, see Internal Memory to Internal Memory.
Arm platform Internal memory ↔ Arm platform Peripheral	3		Data transit through SDMA Script example, see Transfer Between Peripheral and Internal Memory.

NOTE

These scripts are provided as examples of how to use DMA blocks to perform required data transfers: They are not "official" programs.

55.7.2.1 External Memory to External Memory

This section describes the SDMA script that performs data moves in external memory.

For this particular data transfer, only the burst DMA is used. It is programmed in copy mode, so no data transmits through an SDMA general register.

The SDMA core only monitors data transfer status. It is assumed source and destination address values are already present in two SDMA general registers (r1 and r2). For this example, it is also assumed that a 32-bit word-to-move for source-to-destination address is present in r0 and equals 64.

Data Moves in External Memory

```
1         stf r1,MSA                          // Source address setup

2         stf r2,MDA                          // Destination address setup

3         ldi r0,0x64                         // 64 words must be transferred from MSA to
MDA

4         ldi r1,0x8

MAIN_XFER:

5         cmphs r0,r1                         // Is r0 >= 0x8

6         bf LAST_XFER                      // If not, jump to last transfer label

7         stf r1,MD|CPY                        // Copy 8 words from MSA to MDA address.

8         subi r0,0x8                         // Decrement counter

9         jmp MAIN_XFER                     // return to main transfer loop

LAST_XFER:

10        stf r0,MD|CPY                        // perform last transfer
```

All instructions are performed in one cycle (jumps excepted). Instruction 7 triggers a copy transfer: A read burst access of 8-word starts, data is staged in MD and then a write burst of 8 words is executed. Instruction 8, 9, 5, and 6 are executed while the burst access is in progress. If this access is not complete when instruction 7 is executed a second time, SDMA stalls on this instruction as long as the previous copy transfer is not over. In this case, the instruction is no longer a one-cycle instruction.

During the main loop (MAIN_XFER), r1 always equals 8, so burst lengths are 8 words. On the last ldf |CPY instruction (10), r1 equals the reminder of r0 divided by 8; therefore, the length of bursts triggered in copy mode equal r1 value, which is between 1 and 7.

55.7.2.2 Peripheral to Peripheral Transfer

For this data transfer, only the peripheral DMA is used.

It is programmed in copy mode, so no data will transmit through the SDMA general register used in the ldf instruction, but the contents of the general register are lost. The SDMA core only monitors the transfer.

55.7.2.2.1 Source and Destination Target Have the Same Data Path Width

When the source and destination target have the same data path width, the following is true:

- Source target is a *half-word* (16-bit) peripheral located at address 0x1002.
- Destination is a *half-word* (16-bit) peripheral located at address 0x2006.

It is assumed the address values are already present in two SDMA general registers (r1, r2). The script for a transfer of 10 half-word is as follows:

Same Data Path Width for Source and Destination

```
//SETUP SECTION
1         stf r1, PSA|SZ16|F                    //r1=0x1002 Source address register setup
2         stf r2, PDA|SZ16|F                    //r2=0x2006 Destination address register
setup
3         bdf ERROR_ADDR_SETUP
4         ldi r0,0xa                           //loop counter is 10
//MAIN LOOP TRANFER
copy_loop:

5         loop 2,0
6           ldf r7,PD|CPY                      //Reads 1 half-word from src and writes to
dest.
7           yield
8         bdf ERROR_DURING_XFER
ERROR_ADDR_SETUP:
                    //correction of PSA/PDA setup and jumps to main loop transfer
ERROR_DURING_XFER:
//flag error is set,
//PS can be read to know if error occurs during read or write access.
```

If a data transfer must occur between two word peripherals, only the setup section should be updated. The transfer itself is always performed by the hardware loop instruction.

All instructions are executed in one cycle (change of flow excepted). On instruction 6, a single read access is triggered, read data is staged in PD, and a write-to-destination is executed. When the transfers are in progress, the SDMA can execute he next instructions in parallel. If instruction 6, which performs the copy transfer, is executed while the previous access is not over, SDMA is stalled and instruction ldf is a multi-cycle instruction.

55.7.2.2.2 Source and Destination Target Have a Different Data Path Width

When the source and destination target have a different data path width, copy mode cannot be used, and any attempt to initiate a copy transfer immediately raises an error, which is stored in the SF flag.

The following example shows the SDMA code that could transfer 10 words from a *word* (32-bit) peripheral to a *half-word* peripheral whose addresses are preliminary and stored in r1 and r2.

Different Data Path Width for Source and Destination

```
//SETUP SECTION
1         stf r1, PSA|SZ32|F|PF                    //r1=0x1000 and prefetch data
2         stf r2, PDA|SZ16|F                      //r2=0x2006
3         bdf ERROR_ADDR_SETUP
4         ldi r0,0xa                             //loop counter is 10
//MAIN LOOP TRANFER
main_loop_xfer_16_16:
5         loop 6,0
6           ldf r7,PD                           //copy 32-bit of PD in r7
7           stf r7,PD                            //store 16 LSB of r7 in PD and a flush is
executed
8           rorb r7
9           rorb r7                            //16 MSB --> 16 LSB
10          stf r7,PD                            //store 16 LSB of r6 in PD and a flush.
11          yield
```

On instruction 1, when the source address register is programmed and a data prefetch is required, a read access is executed. In parallel, the SDMA executes instructions 2 to 5. On instruction 6, the SDMA tries to read data that was fetched by instruction 1. If data is ready, the ldf will be a one cycle instruction; otherwise, the SDMA is stalled as long as the read access is not finished. Then, the 16 LSB of the read data is stored in PD and automatically flushed to the destination peripheral. In parallel, the SDMA executes the rotation instructions (8, 9), and stores the 16 MSB of the read data into PD. If a previous write access is finished, instruction 10 will be a one-cycle instruction.

The main loop transfer may appear inefficient, but due to wait states imposed to the peripheral DMA each time an external access is performed, a software pipeline is in place. During the time needed to flush PD, the SDMA executes the move and rotation operations. SDMA executes instructions in parallel with DMA accesses.

55.7.2.3 Transfer Between Peripheral and External Memory

55.7.2.3.1 Peripheral to External Memory Transfer

A transfer from a peripheral to the external memory controller involves the peripheral DMA and the burst DMA.

The code for transferring 100 word from word peripheral to the external memory would be as follows:

Peripheral to External Memory Transfer

```
//SETUP SECTION source and destination addresses are already in r1 and r2
1         stf r1, PSA|SZ16|F|PF                     //r1=0x1000 and prefetch 32-bit data
2         stf r2, MDA                             //r2=0x2000, setup burst DMA destination
address
3         bdf ERROR_ADDR_SETUP
4         ldi r0,0x64                             //loop counter is 100
5
          //MAIN LOOP TRANFER
6         loop 3,0
```

```
7           ldf r1,PD|PF                    // read 32 bits of PD and initiate a new read
access.
8           stf r1,MD|32                     // store 32 bits of r1 in the MD fifo.
9           yield
10        ldf r1,PD                       // last word data is read
11        stf r1,MD|32|FL                    // to flush all remaining bytes of MD
```

On instruction 1, the source address register of the peripheral DMA is programmed and data is fetched. This data is stored in PD and the SDMA reads PD during instruction 7, which is a one-cycle instruction that is read-access finished. On the same instruction (7), a data prefetch is required and a read access to the source peripheral is executed. In parallel, the SDMA stored the previous read data into the data register of MD. When MD (which is an eight-word FIFO) is full, a burst write access is executed to empty the FIFO. As long as the next SDMA instructions do not access the burst DMA, they will be one-cycle instructions. The following figures show how the peripheral DMA and burst DMA work in parallel.

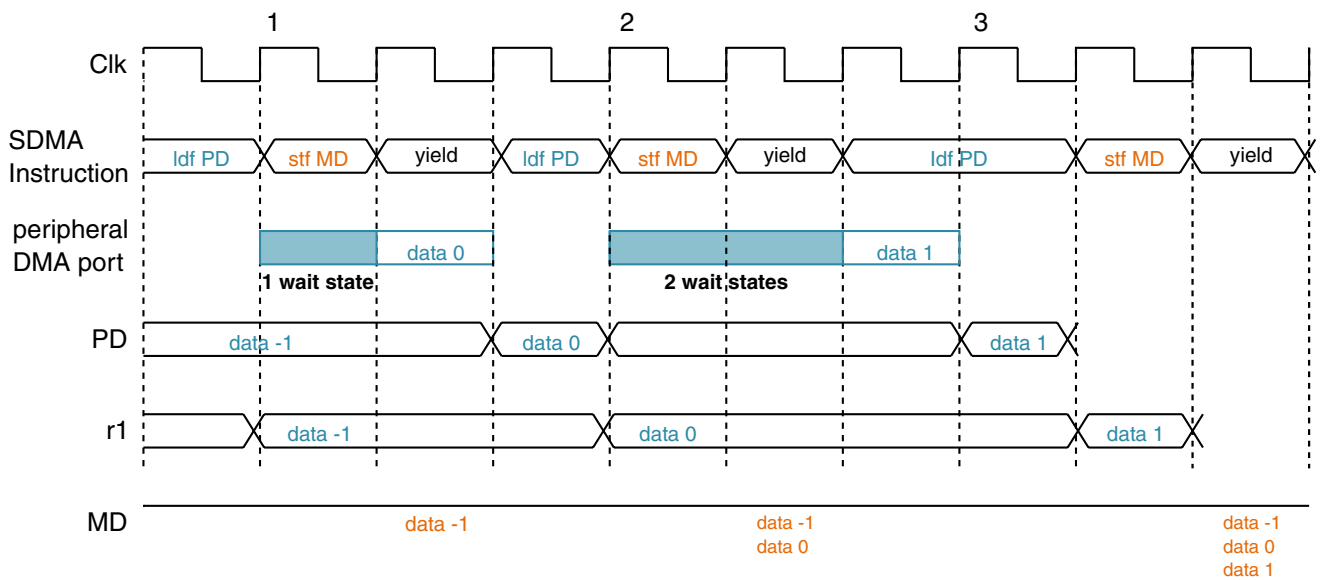


Figure 55-21. Peripheral to External Memory Example (1)

As seen in the figure above, the read access triggered by the ldf PD instruction is symbolized by the blue bar when in progress. After wait states, the read data (data 0, data 1) is stored in PD on the clk rising edge. On edge 2, data 0 is available in PD so it can be transferred to the SDMA general register r1, and then stored in MD FIFO. On edge 3, data 1 is not in PD; therefore, SDMA is stalled on the ldf instruction, which lasts two cycles. The figure below shows an example of when MD FIFO is full with data.

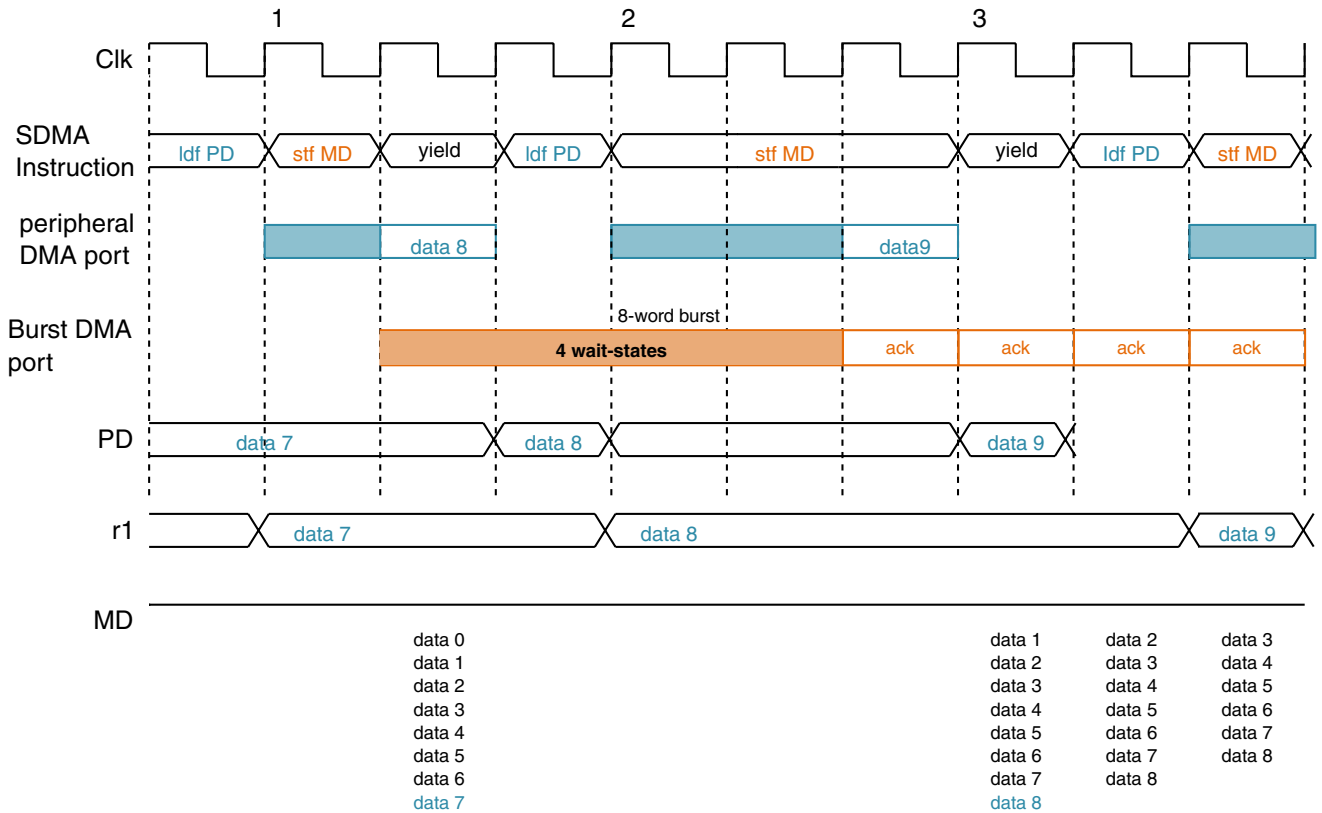


Figure 55-22. Peripheral to External Memory Example (2)

In the previous figure, the write bar means the burst DMA is performing a write burst access. The latency to have the first write acknowledge is four cycles. SDMA is stalled on instruction stf because no acknowledge was received, MD FIFO is full, and there is no empty slot to store data 9. When an acknowledge is sampled by the burst DMA, FIFO is shifted and data 8 is written. As long as there is at least one empty slot in MD FIFO, the stf MD instruction lasts one cycle.

55.7.2.3.2 External Memory to Peripheral Transfer

A transfer from the external memory to a peripheral involves the peripheral DMA and the burst DMA.

The code for transferring 100 word from external memory to a word peripheral would be as follows:

External Memory to Peripheral Transfer

```
//SETUP SECTION source and destination addresses are already in r1 and r2
1         stf r1, MSA|PF              //r1=0x1000 and starts a 8-word read burst
2         stf r2, PDA|SZ32|P           //r2=0x2010, setup peripheral DMA destination address
3         bdf ERROR_ADDR_SETUP
4         ldi r0,0x64                //loop counter is 100
//MAIN LOOP TRANFER
6         loop 3,0
```

```
7           ldf r1,MD|32|PF          // read 32 bits of MD and initiate a new read access
                                  // if MD is empty after this reading.
8           stf r1,PD               // store 32 bits of r1 in the PD.
9           yield
          10 ldf r1,MD|32            // last word data is read
11        stf r1,PD                 // last write access
```

On instruction 1, a read burst of 8 words begins. Read data is staged into MD. On instruction 7 (and if data is available in MD), 32 bits are copied into r1. Then instruction 8 writes them into PD and an automatic flush is executed. The SDMA core, peripheral DMA, and burst DMA can work in parallel as long as no SDMA instruction tries to start a new write access on the peripheral DMA while the previous access is still in progress, or as long as there is data in MD when the SDMA tries to read it.

55.7.2.4 Transfer Between External Memory and Internal Memory

Since the internal memory (Arm platform RAM) is accessed via the peripheral DMA and the external memory is accessed via the burst DMA, the SDMA scripts that are described in Transfer Between Peripheral and External Memory can be reused. The exception is that the peripheral DMA address registers (PSA or PDA, depending on the script) should be programmed in incremented mode rather than frozen mode.

55.7.2.4.1 Internal Memory to Internal Memory

The internal memory can only be accessed via the peripheral DMA, so the script described in Peripheral to Peripheral Transfer can be reused with a different programming of the peripheral DMA address registers.

55.7.2.4.2 Transfer Between Peripheral and Internal Memory

For this transfer, the peripheral DMA is also used in copy mode.

The SDMA script is very similar to the one described in Peripheral to Peripheral Transfer, except for the peripheral DMA address registers programming.

55.8 Arm Platform Memory Map and Control Register Definitions

The Arm platform controls the SDMA by means of several interface registers. Those registers are described in the current section.

All registers are clocked with the SDMA clock (which means the Arm platform must ensure that the SDMA clock is running when it wants to access any register).

SDMAARM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_C000	Arm platform Channel 0 Pointer (SDMAARM_MC0PTR)	32	R/W	0000_0000h	55.8.1/4861
20E_C004	Channel Interrupts (SDMAARM_INTR)	32	w1c	0000_0000h	55.8.2/4861
20E_C008	Channel Stop/Channel Status (SDMAARM_STOP_STAT)	32	w1c	0000_0000h	55.8.3/4861
20E_C00C	Channel Start (SDMAARM_HSTART)	32	R/W	0000_0000h	55.8.4/4862
20E_C010	Channel Event Override (SDMAARM_EVTOVR)	32	R/W	0000_0000h	55.8.5/4862
20E_C014	Channel BP Override (SDMAARM_DSPOVR)	32	R/W	FFFF_FFFFh	55.8.6/4863
20E_C018	Channel Arm platform Override (SDMAARM_HOSTOVR)	32	R/W	0000_0000h	55.8.7/4863
20E_C01C	Channel Event Pending (SDMAARM_EVTPEND)	32	w1c	0000_0000h	55.8.8/4863
20E_C024	Reset Register (SDMAARM_RESET)	32	R	0000_0000h	55.8.9/4864
20E_C028	DMA Request Error Register (SDMAARM_EVTERR)	32	R	0000_0000h	55.8.10/ 4865
20E_C02C	Channel Arm platform Interrupt Mask (SDMAARM_INTRMASK)	32	R/W	0000_0000h	55.8.11/ 4865
20E_C030	Schedule Status (SDMAARM_PSW)	32	R	0000_0000h	55.8.12/ 4866
20E_C034	DMA Request Error Register (SDMAARM_EVTERRDBG)	32	R	0000_0000h	55.8.13/ 4866
20E_C038	Configuration Register (SDMAARM_CONFIG)	32	R/W	0000_0003h	55.8.14/ 4867
20E_C03C	SDMA LOCK (SDMAARM_SDMA_LOCK)	32	R/W	0000_0000h	55.8.15/ 4868
20E_C040	OnCE Enable (SDMAARM_ONCE_ENB)	32	R/W	0000_0000h	55.8.16/ 4869
20E_C044	OnCE Data Register (SDMAARM_ONCE_DATA)	32	R/W	0000_0000h	55.8.17/ 4870
20E_C048	OnCE Instruction Register (SDMAARM_ONCE_INSTR)	32	R/W	0000_0000h	55.8.18/ 4870
20E_C04C	OnCE Status Register (SDMAARM_ONCE_STAT)	32	R	0000_E000h	55.8.19/ 4870
20E_C050	OnCE Command Register (SDMAARM_ONCE_CMD)	32	R/W	0000_0000h	55.8.20/ 4872

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_C058	Illegal Instruction Trap Address (SDMAARM_ILLINSTADDR)	32	R/W	0000_0001h	55.8.21/ 4873
20E_C05C	Channel 0 Boot Address (SDMAARM_CHN0ADDR)	32	R/W	0000_0050h	55.8.22/ 4873
20E_C060	DMA Requests (SDMAARM_EVT_MIRROR)	32	R	0000_0000h	55.8.23/ 4874
20E_C064	DMA Requests 2 (SDMAARM_EVT_MIRROR2)	32	R	0000_0000h	55.8.24/ 4874
20E_C070	Cross-Trigger Events Configuration Register 1 (SDMAARM_XTRIG_CONF1)	32	R/W	0000_0000h	55.8.25/ 4875
20E_C074	Cross-Trigger Events Configuration Register 2 (SDMAARM_XTRIG_CONF2)	32	R/W	0000_0000h	55.8.26/ 4877
20E_C100	Channel Priority Registers (SDMAARM_SDMA_CHNPRI0)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C104	Channel Priority Registers (SDMAARM_SDMA_CHNPRI1)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C108	Channel Priority Registers (SDMAARM_SDMA_CHNPRI2)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C10C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI3)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C110	Channel Priority Registers (SDMAARM_SDMA_CHNPRI4)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C114	Channel Priority Registers (SDMAARM_SDMA_CHNPRI5)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C118	Channel Priority Registers (SDMAARM_SDMA_CHNPRI6)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C11C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI7)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C120	Channel Priority Registers (SDMAARM_SDMA_CHNPRI8)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C124	Channel Priority Registers (SDMAARM_SDMA_CHNPRI9)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C128	Channel Priority Registers (SDMAARM_SDMA_CHNPRI10)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C12C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI11)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C130	Channel Priority Registers (SDMAARM_SDMA_CHNPRI12)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C134	Channel Priority Registers (SDMAARM_SDMA_CHNPRI13)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C138	Channel Priority Registers (SDMAARM_SDMA_CHNPRI14)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C13C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI15)	32	R/W	0000_0000h	55.8.27/ 4878

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_C140	Channel Priority Registers (SDMAARM_SDMA_CHNPRI16)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C144	Channel Priority Registers (SDMAARM_SDMA_CHNPRI17)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C148	Channel Priority Registers (SDMAARM_SDMA_CHNPRI18)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C14C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI19)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C150	Channel Priority Registers (SDMAARM_SDMA_CHNPRI20)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C154	Channel Priority Registers (SDMAARM_SDMA_CHNPRI21)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C158	Channel Priority Registers (SDMAARM_SDMA_CHNPRI22)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C15C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI23)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C160	Channel Priority Registers (SDMAARM_SDMA_CHNPRI24)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C164	Channel Priority Registers (SDMAARM_SDMA_CHNPRI25)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C168	Channel Priority Registers (SDMAARM_SDMA_CHNPRI26)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C16C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI27)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C170	Channel Priority Registers (SDMAARM_SDMA_CHNPRI28)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C174	Channel Priority Registers (SDMAARM_SDMA_CHNPRI29)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C178	Channel Priority Registers (SDMAARM_SDMA_CHNPRI30)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C17C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI31)	32	R/W	0000_0000h	55.8.27/ 4878
20E_C200	Channel Enable RAM (SDMAARM_CHNENBL0)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C204	Channel Enable RAM (SDMAARM_CHNENBL1)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C208	Channel Enable RAM (SDMAARM_CHNENBL2)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C20C	Channel Enable RAM (SDMAARM_CHNENBL3)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C210	Channel Enable RAM (SDMAARM_CHNENBL4)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C214	Channel Enable RAM (SDMAARM_CHNENBL5)	32	R/W	0000_0000h	55.8.28/ 4878

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_C218	Channel Enable RAM (SDMAARM_CHNENBL6)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C21C	Channel Enable RAM (SDMAARM_CHNENBL7)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C220	Channel Enable RAM (SDMAARM_CHNENBL8)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C224	Channel Enable RAM (SDMAARM_CHNENBL9)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C228	Channel Enable RAM (SDMAARM_CHNENBL10)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C22C	Channel Enable RAM (SDMAARM_CHNENBL11)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C230	Channel Enable RAM (SDMAARM_CHNENBL12)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C234	Channel Enable RAM (SDMAARM_CHNENBL13)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C238	Channel Enable RAM (SDMAARM_CHNENBL14)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C23C	Channel Enable RAM (SDMAARM_CHNENBL15)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C240	Channel Enable RAM (SDMAARM_CHNENBL16)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C244	Channel Enable RAM (SDMAARM_CHNENBL17)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C248	Channel Enable RAM (SDMAARM_CHNENBL18)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C24C	Channel Enable RAM (SDMAARM_CHNENBL19)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C250	Channel Enable RAM (SDMAARM_CHNENBL20)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C254	Channel Enable RAM (SDMAARM_CHNENBL21)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C258	Channel Enable RAM (SDMAARM_CHNENBL22)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C25C	Channel Enable RAM (SDMAARM_CHNENBL23)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C260	Channel Enable RAM (SDMAARM_CHNENBL24)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C264	Channel Enable RAM (SDMAARM_CHNENBL25)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C268	Channel Enable RAM (SDMAARM_CHNENBL26)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C26C	Channel Enable RAM (SDMAARM_CHNENBL27)	32	R/W	0000_0000h	55.8.28/ 4878

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_C270	Channel Enable RAM (SDMAARM_CHNENBL28)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C274	Channel Enable RAM (SDMAARM_CHNENBL29)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C278	Channel Enable RAM (SDMAARM_CHNENBL30)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C27C	Channel Enable RAM (SDMAARM_CHNENBL31)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C280	Channel Enable RAM (SDMAARM_CHNENBL32)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C284	Channel Enable RAM (SDMAARM_CHNENBL33)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C288	Channel Enable RAM (SDMAARM_CHNENBL34)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C28C	Channel Enable RAM (SDMAARM_CHNENBL35)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C290	Channel Enable RAM (SDMAARM_CHNENBL36)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C294	Channel Enable RAM (SDMAARM_CHNENBL37)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C298	Channel Enable RAM (SDMAARM_CHNENBL38)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C29C	Channel Enable RAM (SDMAARM_CHNENBL39)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C2A0	Channel Enable RAM (SDMAARM_CHNENBL40)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C2A4	Channel Enable RAM (SDMAARM_CHNENBL41)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C2A8	Channel Enable RAM (SDMAARM_CHNENBL42)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C2AC	Channel Enable RAM (SDMAARM_CHNENBL43)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C2B0	Channel Enable RAM (SDMAARM_CHNENBL44)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C2B4	Channel Enable RAM (SDMAARM_CHNENBL45)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C2B8	Channel Enable RAM (SDMAARM_CHNENBL46)	32	R/W	0000_0000h	55.8.28/ 4878
20E_C2BC	Channel Enable RAM (SDMAARM_CHNENBL47)	32	R/W	0000_0000h	55.8.28/ 4878

55.8.1 Arm platform Channel 0 Pointer (SDMAARM_MC0PTR)

Address: 20E_C000h base + 0h offset = 20E_C000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MC0PTR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_MC0PTR field descriptions

Field	Description
MC0PTR	**Channel 0 Pointer** contains the 32-bit address, in Arm platform memory, of channel 0 control block (the boot channel). Appendix A fully describes the SDMA Application Programming Interface (API). The Arm platform has a read/write access and the SDMA has a read-only access.

55.8.2 Channel Interrupts (SDMAARM_INTR)

Address: 20E_C000h base + 4h offset = 20E_C004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HI[31:0]																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_INTR field descriptions

Field	Description
HI[31:0]	The Arm platform Interrupts register contains the 32 HI[i] bits. If any bit is set, it will cause an interrupt to the Arm platform. This register is a "write-ones" register to the Arm platform. When the Arm platform sets a bit in this register the corresponding HI[i] bit is cleared. The interrupt service routine should clear individual channel bits when their interrupts are serviced, failure to do so will cause continuous interrupts. The SDMA is responsible for setting the HI[i] bit corresponding to the current channel when the corresponding `done` instruction is executed.

55.8.3 Channel Stop/Channel Status (SDMAARM_STOP_STAT)

Address: 20E_C000h base + 8h offset = 20E_C008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HE																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_STOP_STAT field descriptions

Field	Description
HE	This 32-bit register gives access to the Arm platform Enable bits. There is one bit for every channel. This register is a "write-ones" register to the Arm platform. When the Arm platform writes 1 in bit i of this register, it clears the HE[i] and HSTART[i] bits. Reading this register yields the current state of the HE[i] bits.

55.8.4 Channel Start (SDMAARM_HSTART)

Address: 20E_C000h base + Ch offset = 20E_C00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HSTART_HE																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_HSTART field descriptions

Field	Description
HSTART_HE	The HSTART_HE registers are 32 bits wide with one bit for every channel. When a bit is written to 1, it enables the corresponding channel. Two physical registers are accessed with that address (HSTART and HE), which enables the Arm platform to trigger a channel a second time before the first trigger is processed. • This register is a "write-ones" register to the Arm platform. Neither HSTART[i] bit can be set while the corresponding HE[i] bit is cleared. • When the Arm platform tries to set the HSTART[i] bit by writing a one (if the corresponding HE[i] bit is clear), the bit in the HSTART[i] register will remain cleared and the HE[i] bit will be set. • If the corresponding HE[i] bit was already set, the HSTART[i] bit will be set. The next time the SDMA channel *i* attempts to clear the HE[i] bit by means of a `done` instruction, the bit in the HSTART[i] register will be cleared and the HE[i] bit will take the old value of the HSTART[i] bit. • Reading this register yields the current state of the HSTART[i] bits. This mechanism enables the Arm platform to pipeline two HSTART commands per channel.

55.8.5 Channel Event Override (SDMAARM_EVTOVR)

Address: 20E_C000h base + 10h offset = 20E_C010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_EVTOVR field descriptions

Field	Description
EO	The Channel Event Override register contains the 32 EO[i] bits. A bit set in this register causes the SDMA to ignore DMA requests when scheduling the corresponding channel.

55.8.6 Channel BP Override (SDMAARM_DSPOVR)

Address: 20E_C000h base + 14h offset = 20E_C014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DO																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

SDMAARM_DSPOVR field descriptions

Field	Description
DO	This register is reserved. All DO bits should be set to the reset value of 1. A setting of 0 will prevent SDMA channels from starting according to the condition described in Runnable Channels Evaluation. 0 - Reserved 1 - Reset value.

55.8.7 Channel Arm platform Override (SDMAARM_HOSTOVR)

Address: 20E_C000h base + 18h offset = 20E_C018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_HOSTOVR field descriptions

Field	Description
HO	The Channel Arm platform Override register contains the 32 HO[i] bits. A bit set in this register causes the SDMA to ignore the Arm platform enable bit (HE) when scheduling the corresponding channel.

55.8.8 Channel Event Pending (SDMAARM_EVTPEND)

Address: 20E_C000h base + 1Ch offset = 20E_C01Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EP																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_EVTPEND field descriptions

Field	Description
EP	The Channel Event Pending register contains the 32 EP[i] bits. Reading this register enables the Arm platform to determine what channels are pending after the reception of a DMA request. • Setting a bit in this register causes the SDMA to reevaluate scheduling as if a DMA request mapped on this channel had occurred. This is useful for starting up channels, so that initialization is done before awaiting the first request. The scheduler can also set bits in the EVTPEND register according to the received DMA requests. • The EP[i] bit may be cleared by the `done` instruction when running the channel *i* script. This a "write-ones" mechanism: Writing a '0' does not clear the corresponding bit.

55.8.9 Reset Register (SDMAARM_RESET)

Address: 20E_C000h base + 24h offset = 20E_C024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														RESCHED	RESET
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_RESET field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

SDMAARM_RESET field descriptions (continued)

Field	Description
1 RESCHED	When set, this bit forces the SDMA to reschedule as if a script had executed a `done` instruction. This enables the Arm platform to recover from a runaway script on a channel by clearing its HE[i] bit via the STOP register, and then forcing a reschedule via the RESCHED bit. The RESCHED bit is cleared when the context switch starts.
0 RESET	When set, this bit causes the SDMA to be held in a software reset. The internal reset signal is held low 16 cycles; the RESET bit is automatically cleared when the internal reset signal rises.

55.8.10 DMA Request Error Register (SDMAARM_EVTERR)

Address: 20E_C000h base + 28h offset = 20E_C028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHNERR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_EVTERR field descriptions

Field	Description
CHNERR	This register is used by the SDMA to warn the Arm platform when an incoming DMA request was detected and it triggers a channel that is already pending or being serviced. This probably means there is an overflow of data for that channel. • An interrupt is sent to the Arm platform if the corresponding channel bit is set in the INTRMASK register. • This is a "write-ones" register for the scheduler. It is only able to set the flags. The flags are cleared when the register is read by the Arm platform or during SDMA reset. • The CHNERR[i] bit is set when a DMA request that triggers channel *i* is received through the corresponding input pins and the EP[i] bit is already set; the EVTERR[i] bit is unaffected if the Arm platform tries to set the EP[i] bit, whereas, that EP[i] bit is already set.

55.8.11 Channel Arm platform Interrupt Mask (SDMAARM_INTRMASK)

Address: 20E_C000h base + 2Ch offset = 20E_C02Ch

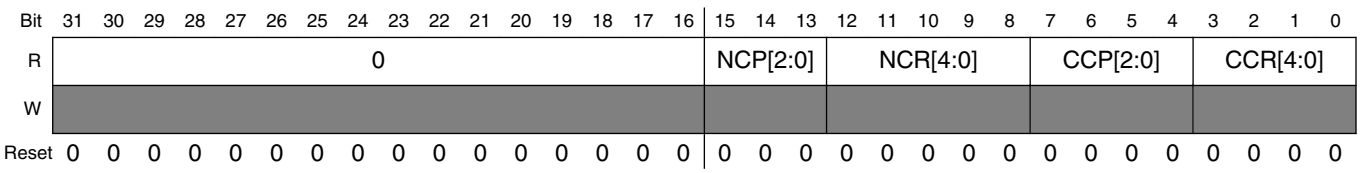
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HIMASK																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_INTRMASK field descriptions

Field	Description
HIMASK	The Interrupt Mask Register contains 32 interrupt generation mask bits. If bit HIMASK[i] is set, the HI[i] bit is set and an interrupt is sent to the Arm platform when a DMA request error is detected on channel *i* (for example, EVTERR[i] is set).

55.8.12 Schedule Status (SDMAARM_PSW)

Address: 20E_C000h base + 30h offset = 20E_C030h

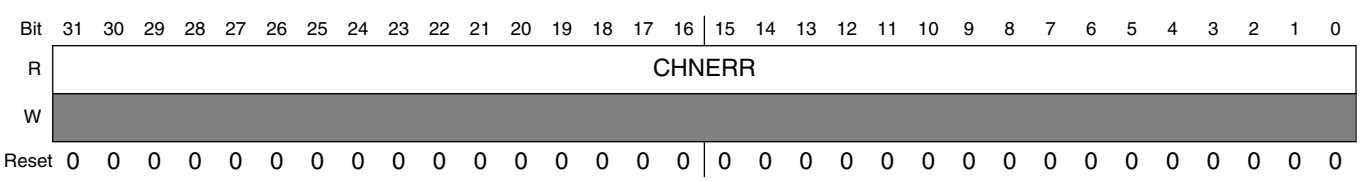


SDMAARM_PSW field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–13 NCP[2:0]	The Next Channel Priority gives the next pending channel priority. When the priority is 0, it means there is no pending channel and the NCR value has no meaning. 0 No running channel 1 Active channel priority
12–8 NCR[4:0]	The Next Channel Register indicates the number of the next scheduled pending channel with the highest priority.
7–4 CCP[2:0]	The Current Channel Priority indicates the priority of the current active channel. When the priority is 0, no channel is running: The SDMA is idle and the CCR value has no meaning. In the case that the SDMA has finished running the channel and has entered sleep state, CCP will indicate the priority of previous running channel. 0 No running channel 1 Active channel priority
CCR[4:0]	The Current Channel Register indicates the number of the channel that is being executed by the SDMA. SDMA. In the case that the SDMA has finished running the channel and has entered sleep state, CCR will indicate the previous running channel.

55.8.13 DMA Request Error Register (SDMAARM_EVTERRDBG)

Address: 20E_C000h base + 34h offset = 20E_C034h

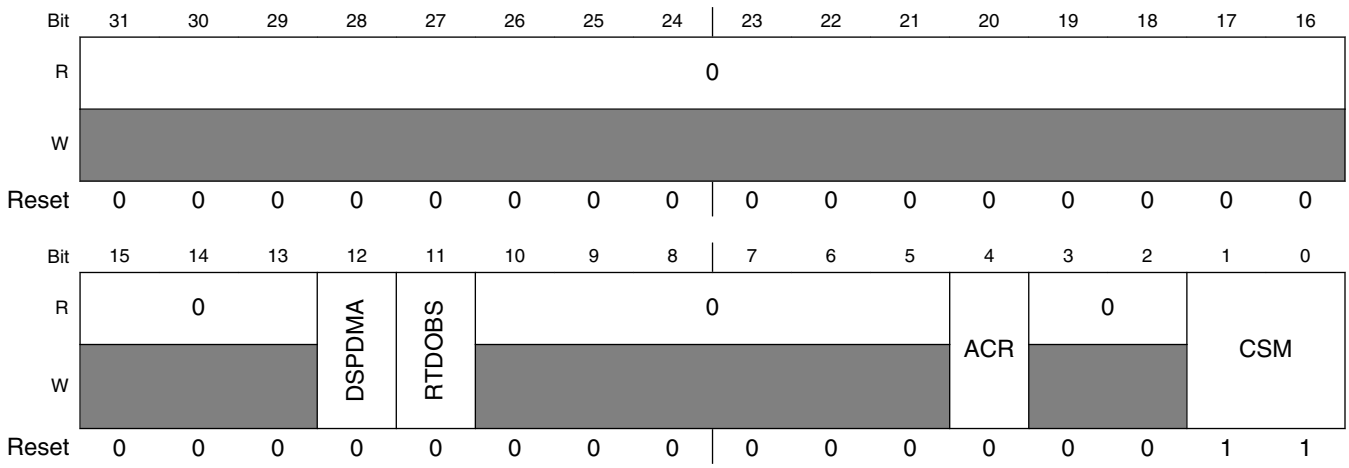


SDMAARM_EVTERRDBG field descriptions

Field	Description
CHNERR	This register is the same as EVTERR, except reading it does not clear its contents. This address is meant to be used in debug mode. The Arm platform OnCE may check this register value without modifying it.

55.8.14 Configuration Register (SDMAARM_CONFIG)

Address: 20E_C000h base + 38h offset = 20E_C038h



SDMAARM_CONFIG field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value 0.
12 DSPDMA	This bit's function is reserved and should be configured as zero. 0 - Reset Value 1 - Reserved
11 RTDOBS	Indicates if Real-Time Debug pins are used: They do not toggle by default in order to reduce power consumption. 0 RTD pins disabled 1 RTD pins enabled
10–5 Reserved	This read-only field is reserved and always has the value 0.
4 ACR	Arm platform DMA / SDMA Core Clock Ratio. Selects the clock ratio between Arm platform DMA interfaces (burst DMA and peripheral DMA) and the internal SDMA core clock. The frequency selection is determined separately by the chip clock controller. This bit has to match the configuration of the chip clock controller that generates the clocks used in the SDMA. 0 Arm platform DMA interface frequency equals twice core frequency 1 Arm platform DMA interface frequency equals core frequency
3–2 Reserved	This read-only field is reserved and always has the value 0.

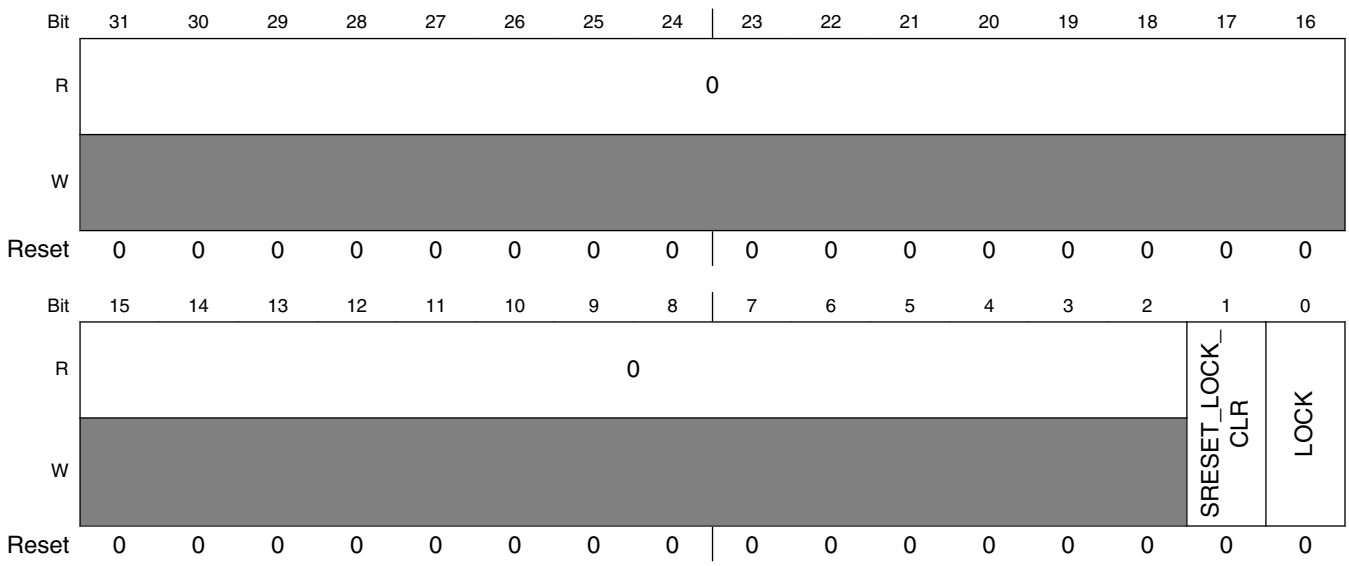
Table continues on the next page...

SDMAARM_CONFIG field descriptions (continued)

Field	Description
CSM	Selects the Context Switch Mode. The Arm platform has a read/write access. The SDMA cannot modify that register. The value at reset is 3, which selects the dynamic context switch by default. That register can be modified at anytime but the new context switch configuration will only be taken into account at the start of the next restore phase. NOTE: The first call to SDMA's channel 0 Bootload script after reset should use static context switch mode to ensure the context RAM for channel 0 is initialized in the channel SAVE Phase. After Channel 0 is run once, then any of the dynamic context modes can be used. 0 static 1 dynamic low power 2 dynamic with no loop 3 dynamic

55.8.15 SDMA LOCK (SDMAARM_SDMA_LOCK)

Address: 20E_C000h base + 3Ch offset = 20E_C03Ch



SDMAARM_SDMA_LOCK field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
1 SRESET_LOCK_CLR	The SRESET_LOCK_CLR bit determine if the LOCK bit is cleared on a software reset triggered by writing to the RESET register. This bit cannot be changed if LOCK=1. SREST_LOCK_CLR is cleared by conditions that clear the LOCK bit. 0 Software Reset does not clear the LOCK bit. 1 Software Reset clears the LOCK bit.

Table continues on the next page...

SDMAARM_SDMA_LOCK field descriptions (continued)

Field	Description
0 LOCK	The LOCK bit is used to restrict access to update SDMA script memory through ROM channel zero scripts and through the OnCE interface under Arm platform control. The LOCK bit is set: • The SDMA_LOCK, ONCE_ENB,CH0ADDR, and ILLINSTADDR registers cannot be written. These registers can be read, but writes are ignored. • SDMA software executing out of ROM or RAM may check the LOCK bit in the LOCK register Lock Status Register (SDMACORE_SDMA_LOCK) to determine if certain operations are allowed, such as up-loading new scripts. Once the LOCK bit is set to 1, only a reset can clear it. The LOCK bit is cleared by a hardware reset. LOCK is cleared by a software reset only if SRESET_LOCK_CLR is set. 0 LOCK disengaged. 1 LOCK enabled.

55.8.16 OnCE Enable (SDMAARM_ONCE_ENB)

Address: 20E_C000h base + 40h offset = 20E_C040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															ENB
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_ONCE_ENB field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 ENB	The OnCE Enable register selects the OnCE control source: When cleared (0), the OnCE registers are accessed through the JTAG interface; when set (1), the OnCE registers may be accessed by the Arm platform through the addresses described, as follows. • After reset, the OnCE registers are accessed through the JTAG interface. • Writing a 1 to ENB enables the Arm platform to access the ONCE_* as any other SDMA control register. • When cleared (0), all the ONCE_xxx registers cannot be written. The value of ENB cannot be changed if the LOCK bit in the SDMA_LOCK register is set.

55.8.17 OnCE Data Register (SDMAARM_ONCE_DATA)

Address: 20E_C000h base + 44h offset = 20E_C044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_ONCE_DATA field descriptions

Field	Description
DATA	Data register of the OnCE JTAG controller. Refer to OnCE and Real-Time Debug for information on this register.

55.8.18 OnCE Instruction Register (SDMAARM_ONCE_INSTR)

Address: 20E_C000h base + 48h offset = 20E_C048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																INSTR															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_ONCE_INSTR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
INSTR	Instruction register of the OnCE JTAG controller. Refer to OnCE and Real-Time Debug for information on this register.

55.8.19 OnCE Status Register (SDMAARM_ONCE_STAT)

Address: 20E_C000h base + 4Ch offset = 20E_C04Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PST[3:0]				RCV	EDR	ODR	SWB	MST	0				ECDR		
W																
Reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_ONCE_STAT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–12 PST[3:0]	The Processor Status bits reflect the state of the SDMA RISC engine. Its states are as follows: • The "Program" state is the usual instruction execution cycle. • The "Data" state is inserted when there are wait-states during a load or a store on the data bus (ld or st). • The "Change of Flow" state is the second cycle of any instruction that breaks the sequence of instructions (jumps and channel switching instructions). • The "Change of Flow in Loop" state is used when an error causes a hardware loop exit. • The "Debug" state means the SDMA is in debug mode. • The "Functional Unit" state is inserted when there are wait-states during a load or a store on the functional units bus (ldf or stf). • In "Sleep" modes, no script is running (this is the RISC engine idle state). The "after Reset" is slightly different because no context restoring phase will happen when a channel is triggered: The script located at address 0 will be executed (boot operation). • The "in Sleep" states are the same as above except they do not have any corresponding channel: They are used when entering debug mode after reset. The reason is that it is necessary to return to the "Sleep after Reset" state when leaving debug mode. 0 Program 1 Data 2 Change of Flow 3 Change of Flow in Loop 4 Debug 5 Functional Unit 6 Sleep 7 Save 8 Program in Sleep 9 Data in Sleep 10 Change of Flow in Sleep 11 Change Flow in Loop in Sleep 12 Debug in Sleep 13 Functional Unit in Sleep 14 Sleep after Reset 15 Restore
11 RCV	After each write access to the real time buffer (RTB), the RCV bit is set. This bit is cleared after execution of an `rbuffer` command and on a JTAG reset.
10 EDR	This flag is raised when the SDMA has entered debug mode after an external debug request.
9 ODR	This flag is raised when the SDMA has entered debug mode after a OnCE debug request.
8 SWB	This flag is raised when the SDMA has entered debug mode after a software breakpoint.
7 MST	This flag is raised when the OnCE is controlled from the Arm platform peripheral interface. 0 The JTAG interface controls the OnCE. 1 The Arm platform peripheral interface controls the OnCE.
6–3 Reserved	This read-only field is reserved and always has the value 0.

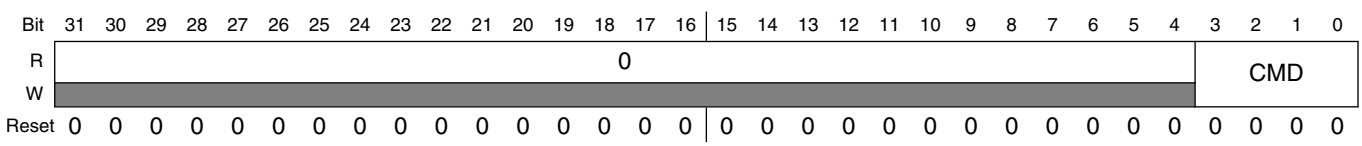
Table continues on the next page...

SDMAARM_ONCE_STAT field descriptions (continued)

Field	Description
ECDR	Event Cell Debug Request. If the debug request comes from the event cell, the reason for entering debug mode is given by the EDR bits. If all three bits of the EDR are reset, then it did not generate any debug request. If the cell did generate a debug request, then at least one of the EDR bits is set (the meaning of the encoding is given below). The encoding of the EDR bits is useful to find out more precisely why the debug request was generated. A debug request from an event cell is generated for a specific combination of the addra_cond, addrb_cond, and data_cond conditions. The value of those fields is given by the EDR bits. 0 1 matched addra_cond 1 1 matched addrb_cond 2 1 matched data_cond

55.8.20 OnCE Command Register (SDMAARM_ONCE_CMD)

Address: 20E_C000h base + 50h offset = 20E_C050h

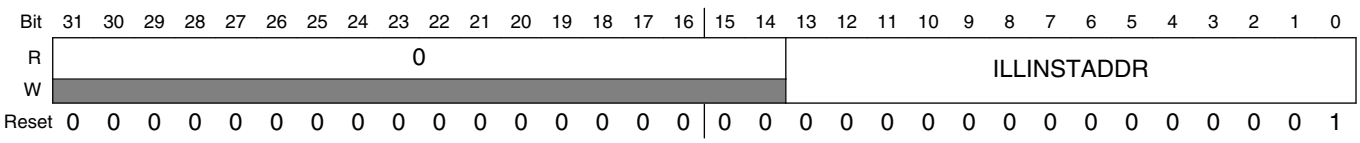


SDMAARM_ONCE_CMD field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value 0.
CMD	Writing to this register will cause the OnCE to execute the command that is written. When needed, the ONCE_DATA and ONCE_INSTR registers should be loaded with the correct value before writing the command to that register. For a list of the OnCE commands and their usage, see OnCE and Real-Time Debug. **NOTE:** 7-15 *reserved* 0 rstatus 1 dmov 2 exec_once 3 run_core 4 exec_core 5 debug_rqst 6 rbuffer

55.8.21 Illegal Instruction Trap Address (SDMAARM_ILLINSTADDR)

Address: 20E_C000h base + 58h offset = 20E_C058h

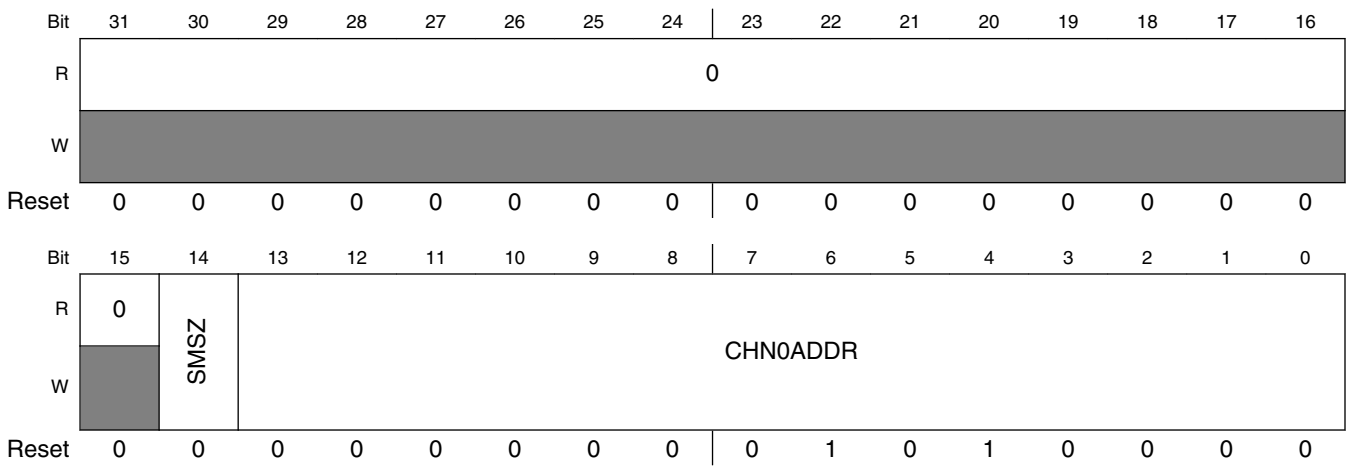


SDMAARM_ILLINSTADDR field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
ILLINSTADDR	The Illegal Instruction Trap Address is the address where the SDMA jumps when an illegal instruction is executed. It is 0x0001 after reset. The value of ILLINSTADDR cannot be changed if the LOCK bit in the SDMA_LOCK register is set.

55.8.22 Channel 0 Boot Address (SDMAARM_CHN0ADDR)

Address: 20E_C000h base + 5Ch offset = 20E_C05Ch



SDMAARM_CHN0ADDR field descriptions

Field	Description
31–15 Reserved	This read-only field is reserved and always has the value 0.
14 SMSZ	The bit 14 (Scratch Memory Size) determines if scratch memory must be available after every channel context. After reset, it is equal to 0, which defines a RAM space of 24 words for each channel. All of this area stores the channel context. By setting this bit, 32 words are reserved for every channel context,

Table continues on the next page...

SDMAARM_CHN0ADDR field descriptions (continued)

Field	Description
	which gives eight additional words that can be used by the channel script to store any type of data. Those words are never erased by the context switching mechanism. The value of SMSZ cannot be changed if the LOCK bit in the SDMA_LOCK register is set. 0 24 words per context 1 32 words per context
CHN0ADDR	This 14-bit register is used by the boot code of the SDMA. After reset, it points to the standard boot routine in ROM (channel 0 routine). By changing this address, you can perform a boot sequence with your own routine. The very first instructions of the boot code fetch the contents of this register (it is also mapped in the SDMA memory space) and jump to the given address. The reset value is 0x0050 (decimal 80). The value of CHN0ADDR cannot be changed if the LOCK bit in the SDMA_LOCK register is set.

55.8.23 DMA Requests (SDMAARM_EVT_MIRROR)

Address: 20E_C000h base + 60h offset = 20E_C060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EVENTS																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_EVT_MIRROR field descriptions

Field	Description
EVENTS	This register reflects the DMA requests received by the SDMA for events 31-0. The Arm platform and the SDMA have a read-only access. There is one bit associated with each of 32 DMA request events. This information may be useful during debug of the blocks that generate the DMA requests. The EVT_MIRROR register is cleared following read access. 0 DMA request event not pending 1 DMA request event pending

55.8.24 DMA Requests 2 (SDMAARM_EVT_MIRROR2)

Address: 20E_C000h base + 64h offset = 20E_C064h

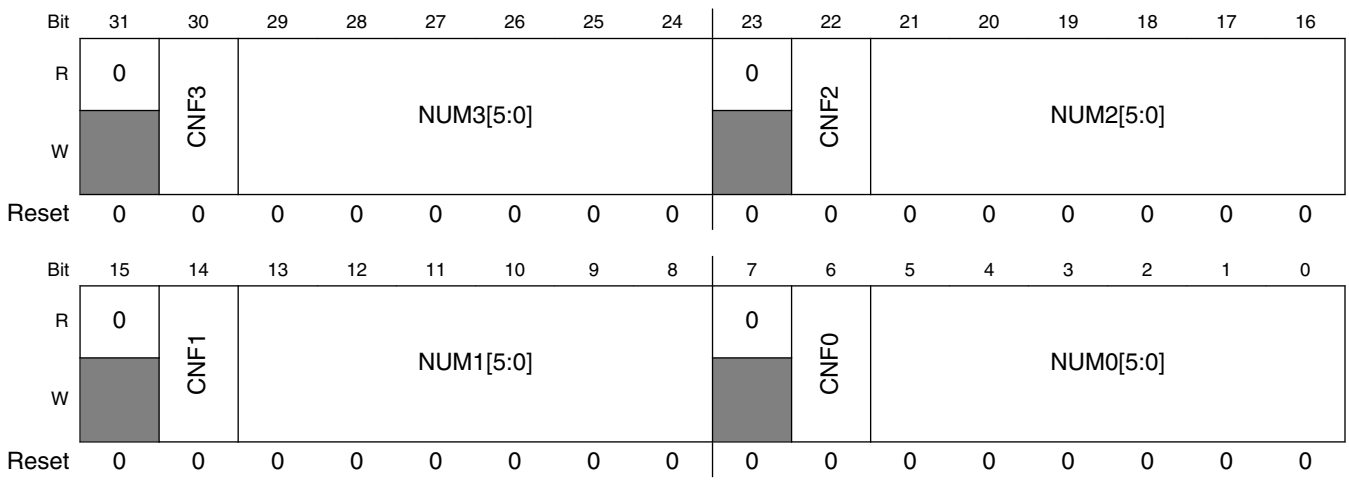
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																EVENTS[47:32]															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_EVT_MIRROR2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
EVENTS[47:32]	This register reflects the DMA requests received by the SDMA for events 47-32. The Arm platform and the SDMA have a read-only access. There is one bit associated with each of DMA request events. This information may be useful during debug of the blocks that generate the DMA requests. The EVT_MIRROR2 register is cleared following read access. 0 - DMA request event not pending 1- DMA request event pending

55.8.25 Cross-Trigger Events Configuration Register 1 (SDMAARM_XTRIG_CONF1)

Address: 20E_C000h base + 70h offset = 20E_C070h



SDMAARM_XTRIG_CONF1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30 CNF3	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by the reception of a DMA request or by the starting of a channel script execution. 0 channel 1 DMA request
29–24 NUM3[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
23 Reserved	This read-only field is reserved and always has the value 0.

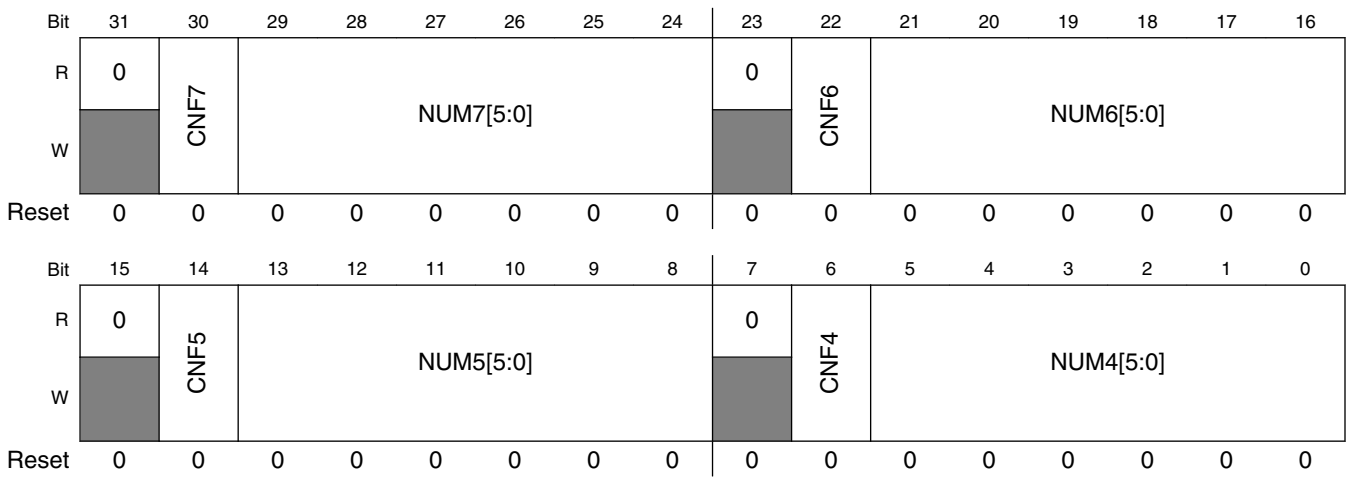
Table continues on the next page...

SDMAARM_XTRIG_CONF1 field descriptions (continued)

Field	Description
22 CNF2	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
21–16 NUM2[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
15 Reserved	This read-only field is reserved and always has the value 0.
14 CNF1	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
13–8 NUM1[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
7 Reserved	This read-only field is reserved and always has the value 0.
6 CNF0	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
NUM0[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.

55.8.26 Cross-Trigger Events Configuration Register 2 (SDMAARM_XTRIG_CONF2)

Address: 20E_C000h base + 74h offset = 20E_C074h



SDMAARM_XTRIG_CONF2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30 CNF7	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
29–24 NUM7[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
23 Reserved	This read-only field is reserved and always has the value 0.
22 CNF6	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
21–16 NUM6[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
15 Reserved	This read-only field is reserved and always has the value 0.
14 CNF5	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution

Table continues on the next page...

SDMAARM_XTRIG_CONF2 field descriptions (continued)

Field	Description
	0 channel 1 DMA request
13–8 NUM5[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
7 Reserved	This read-only field is reserved and always has the value 0.
6 CNF4	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
NUM4[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.

55.8.27 Channel Priority Registers (SDMAARM_SDMA_CHNPRI*n*)

Address: 20E_C000h base + 100h offset + (4d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																													CHNPRIn		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_SDMA_CHNPRI*n* field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
CHNPRIn	This contains the priority of channel number *n*. Useful values are between 1 and 7; 0 is reserved by the SDMA hardware to determine when there is no pending channel. Reset value is 0, which prevents the channels from starting.

55.8.28 Channel Enable RAM (SDMAARM_CHNENBL*n*)

Address: 20E_C000h base + 200h offset + (4d × i), where i=0d to 47d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ENBLn																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_CHNENBL*n* field descriptions

Field	Description
ENBLn	This 32-bit value selects the channels that are triggered by the DMA request number *n*. If ENBLn[i] is set to 1, bit EP[i] will be set when the DMA request *n* is received. These 48 32-bit registers are physically located in a RAM, with no known reset value. It is thus essential for the Arm platform to program them before any DMA request is triggered to the SDMA, otherwise an unpredictable combination of channels may be started.

55.9 BP Memory Map and Control Register Definitions

The following section describes SDMA control registers available to the BP.

NOTE

These registers are physically implemented in all platforms, but are not accessible when the SDMA BP control port is not connected. Reset values are calculated to allow the system to work when those registers cannot be accessed.

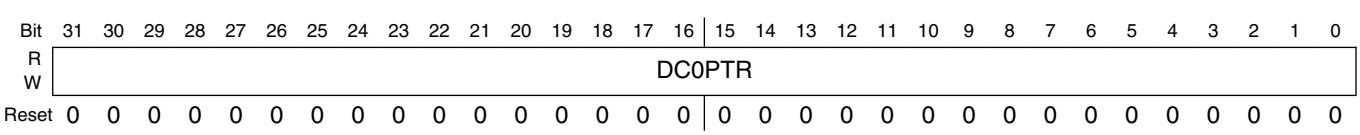
All registers are clocked with the SDMA clock (which means the SDMA clock must be running when the BP wants to access any register).

SDMABP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_C000	Channel 0 Pointer (SDMABP_DC0PTR)	32	R/W	0000_0000h	55.9.1/4879
20E_C004	Channel Interrupts (SDMABP_INTR)	32	w1c	0000_0000h	55.9.2/4880
20E_C008	Channel Stop/Channel Status (SDMABP_STOP_STAT)	32	R/W	0000_0000h	55.9.3/4880
20E_C00C	Channel Start (SDMABP_DSTART)	32	R	0000_0000h	55.9.4/4881
20E_C028	DMA Request Error Register (SDMABP_EVTERR)	32	R	0000_0000h	55.9.5/4881
20E_C02C	Channel DSP Interrupt Mask (SDMABP_INTRMASK)	32	R/W	0000_0000h	55.9.6/4882
20E_C034	DMA Request Error Register (SDMABP_EVTERRDBG)	32	R	0000_0000h	55.9.7/4882

55.9.1 Channel 0 Pointer (SDMABP_DC0PTR)

Address: 20E_C000h base + 0h offset = 20E_C000h



SDMABP_DC0PTR field descriptions

Field	Description
DC0PTR	**Channel 0 Pointer** contains the 32-bit address, in BP memory, of the array of channel control blocks starting with the one for channel 0 (the control channel). This register should be initialized by the BP before it enables a channel (for example, channel 0). See the API document SDMA Scripts User Manual for the use of this register. The BP has a read/write access and the SDMA has a read-only access.

55.9.2 Channel Interrupts (SDMABP_INTR)

Address: 20E_C000h base + 4h offset = 20E_C004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DI																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABP_INTR field descriptions

Field	Description
DI	The BP Interrupts register contains the 32 DI[i] bits. If any bit is set, it will cause an interrupt to the BP. • This register is a "write-ones" register to the BP. When the BP sets a bit in this register, the corresponding DI[i] bit is cleared. • The interrupt service routine should clear individual channel bits when their interrupts are serviced; failure to do so will cause continuous interrupts. • The SDMA is responsible for setting the DI[i] bit corresponding to the current channel when the corresponding `done` instruction is executed.

55.9.3 Channel Stop/Channel Status (SDMABP_STOP_STAT)

Address: 20E_C000h base + 8h offset = 20E_C008h

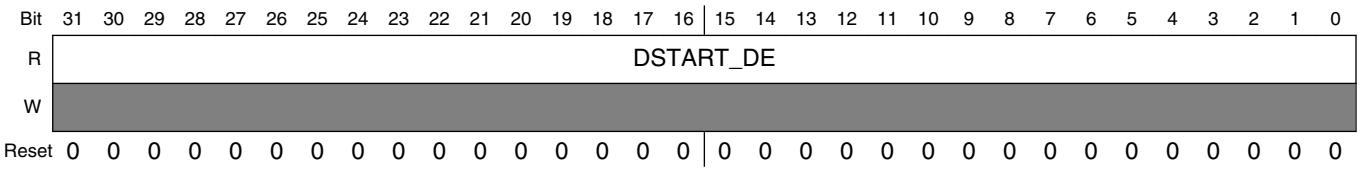
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DE																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABP_STOP_STAT field descriptions

Field	Description
DE	This 32-bit register gives access to the BP (DSP) Enable bits, DE. There is one bit for every channel. • This register is a "write-ones" register to the BP. • When the BP writes 1 in bit *i* of this register, it clears the DE[i] and DSTART[i] bits. • Reading this register yields the current state of the DE[i] bits.

55.9.4 Channel Start (SDMABP_DSTART)

Address: 20E_C000h base + Ch offset = 20E_C00Ch

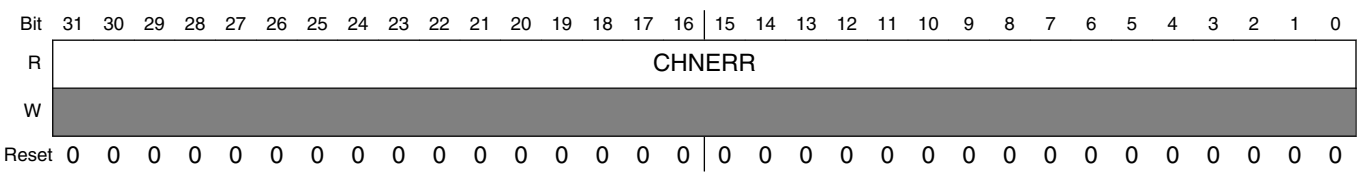


SDMABP_DSTART field descriptions

Field	Description
DSTART_DE	The DSTART_DE registers are 32 bits wide with one bit for every channel. • When a bit is written to 1, it enables the corresponding channel. • Two physical registers are accessed with that address (DSTART and DE), which enables the BP to trigger a channel a second time before the first trigger was processed. • This register is a "write-ones" register to the BP. Neither DSTART[i] bit can be set while the corresponding DE[i] bit is cleared. • When the BP tries to set the DSTART[i] bit by writing a one (if the corresponding DE[i] bit is clear), the bit in the DSTART[i] register will remain cleared and the DE[i] bit will be set. If the corresponding DE[i] bit was already set, the DSTART[i] bit will be set. • The next time the SDMA channel *i* attempts to clear the DE[i] bit by means of a `done` instruction, the bit in the DSTART[i] register will be cleared and the DE[i] bit will take the old value of the DSTART[i] bit. • Reading this register yields the current state of the DSTART[i] bits. This mechanism enables the BP to pipeline two DSTART commands per channel.

55.9.5 DMA Request Error Register (SDMABP_EVTERR)

Address: 20E_C000h base + 28h offset = 20E_C028h



SDMABP_EVTERR field descriptions

Field	Description
CHNERR	This register is used by the SDMA to warn the BP when an incoming DMA request was detected; it then triggers a channel that is already pending or being serviced, which may mean there is an overflow of data for that channel. An interrupt is sent to the BP if the corresponding channel bit is set in the INTRMASK register. • This is a "write-ones" register for the scheduler. It is only able to set the flags. The flags are cleared when the register is read by the BP or during an SDMA reset. • The CHNERR[i] bit is set when a DMA request that triggers channel *i* is received through the corresponding input pins and the EP[i] bit is already set. The EVTERR[i] bit is unaffected if the BP tries to set the EP[i] bit when that EP[i] bit is already set.

55.9.6 Channel DSP Interrupt Mask (SDMABP_INTRMASK)

Address: 20E_C000h base + 2Ch offset = 20E_C02Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DIMASK																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABP_INTRMASK field descriptions

Field	Description
DIMASK	The Interrupt Mask Register contains 32 interrupt generation mask bits. If bit DIMASK[i] is set, the DI[i] bit is set and an interrupt is sent to the BP when a DMA request error is detected on channel *i* (for example, EVTERR[i] is set).

55.9.7 DMA Request Error Register (SDMABP_EVTERRDBG)

Address: 20E_C000h base + 34h offset = 20E_C034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHNERR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABP_EVTERRDBG field descriptions

Field	Description
CHNERR	This register is the same as EVTERR except reading it does not clear its contents. This address is meant to be used in debug mode. The BP OnCE may check this register value without modifying it.

55.10 SDMA Internal (Core) Memory Map and Internal Register Definitions

The actual SDMA memory mapped registers are summarized in the following sections; for peripherals' memory maps, refer to the respective chapters.

The following definitions serve as a key for the SDMA internal register summary.

SDMACORE memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20E_C000	Arm platform Channel 0 Pointer (SDMACORE_MC0PTR)	32	R	0000_0000h	55.10.1/ 4884
20E_C008	Current Channel Pointer (SDMACORE_CCPTR)	32	R	0000_0000h	55.10.2/ 4884
20E_C00C	Current Channel Register (SDMACORE_CCR)	32	R	0000_0000h	55.10.3/ 4884
20E_C010	Highest Pending Channel Register (SDMACORE_NCR)	32	R	0000_0000h	55.10.4/ 4885
20E_C014	External DMA Requests Mirror (SDMACORE_EVENTS)	32	R	0000_0000h	55.10.5/ 4886
20E_C018	Current Channel Priority (SDMACORE_CCPRI)	32	R	0000_0000h	55.10.6/ 4887
20E_C01C	Next Channel Priority (SDMACORE_NCPRI)	32	R	0000_0000h	55.10.7/ 4887
20E_C020	OnCE Event Cell Counter (SDMACORE_ECOUNT)	32	R/W	0000_0000h	55.10.8/ 4888
20E_C024	OnCE Event Cell Control Register (SDMACORE_ECTL)	32	R/W	0000_0000h	55.10.9/ 4888
20E_C028	OnCE Event Address Register A (SDMACORE_EAA)	32	R/W	0000_0000h	55.10.10/ 4890
20E_C02C	OnCE Event Cell Address Register B (SDMACORE_EAB)	32	R/W	0000_0000h	55.10.11/ 4890
20E_C030	OnCE Event Cell Address Mask (SDMACORE_EAM)	32	R/W	0000_0000h	55.10.12/ 4890
20E_C034	OnCE Event Cell Data Register (SDMACORE_ED)	32	R/W	0000_0000h	55.10.13/ 4891
20E_C038	OnCE Event Cell Data Mask (SDMACORE_EDM)	32	R/W	0000_0000h	55.10.14/ 4891
20E_C03C	OnCE Real-Time Buffer (SDMACORE_RTB)	32	R/W	0000_0000h	55.10.15/ 4892
20E_C040	OnCE Trace Buffer (SDMACORE_TB)	32	R	0000_0000h	55.10.16/ 4892
20E_C044	OnCE Status (SDMACORE_OSTAT)	32	R	0000_0000h	55.10.17/ 4893
20E_C048	Channel 0 Boot Address (SDMACORE_MCHN0ADDR)	32	R	0000_0000h	55.10.18/ 4895
20E_C04C	ENDIAN Status Register (SDMACORE_ENDIANNESS)	32	R	0000_0001h	55.10.19/ 4896
20E_C054	Lock Status Register (SDMACORE_SDMA_LOCK)	32	R	0000_0000h	55.10.20/ 4897
20E_C058	External DMA Requests Mirror #2 (SDMACORE_EVENTS2)	32	R	0000_0000h	55.10.21/ 4898

55.10.1 Arm platform Channel 0 Pointer (SDMACORE_MC0PTR)

Address: 20E_C000h base + 0h offset = 20E_C000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MC0PTR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_MC0PTR field descriptions

Field	Description
MC0PTR	Contains the address-in the Arm platform memory space-of the initial SDMA context and scripts that are loaded by the SDMA boot script running on channel 0.

55.10.2 Current Channel Pointer (SDMACORE_CCPTR)

Address: 20E_C000h base + 8h offset = 20E_C008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																CCPTR															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_CCPTR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
CCPTR	Contains the start address of the context data for the current channel: Its value is *CONTEXT_BASE* + 24* *CCR* or *CONTEXT_BASE* + 32* *CCR* where CONTEXT_BASE = 0x0800. The value 24 or 32 is selected according to the programmed channel scratch RAM size in the register shown in Channel 0 Boot Address (SDMAARM_CHN0ADDR) .

55.10.3 Current Channel Register (SDMACORE_CCR)

Address: 20E_C000h base + Ch offset = 20E_C00Ch

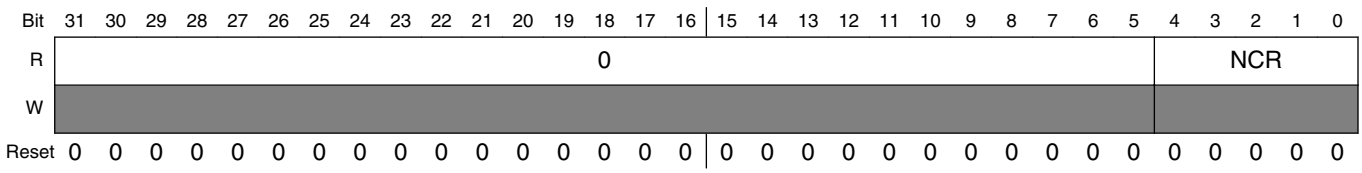
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											CCR				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_CCR field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
CCR	Contains the number of the current running channel whose context is installed. In the case that the SDMA has finished running the channel and has entered sleep state, CCR will indicate the previous running channel. The PST bits in the OSTAT register indicate when the SDMA is in sleep state.

55.10.4 Highest Pending Channel Register (SDMACORE_NCR)

Address: 20E_C000h base + 10h offset = 20E_C010h



SDMACORE_NCR field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
NCR	Contains the number of the pending channel that the scheduler has selected to run next.

55.10.5 External DMA Requests Mirror (SDMACORE_EVENTS)

NOTE

This register is very useful in the case of DMA requests that are active when a peripheral FIFO level is above the programmed watermark. The activation of the DMA request (rising edge) is detected by the SDMA logic and it can enable one or several channels. One of the channels accesses the peripheral and reads or writes a number of data that matches the watermark level (for example, if the watermark is four words, the channel reads or writes four words).

If the channel is effectively executed long after the DMA request was received, reading or writing the watermark number of data may not be sufficient to reset the DMA request (for example, if the FIFO watermark is four and at the channel execution it already contains nine pieces of data). This means no new rising edge may be detected by the SDMA, although there still remains transfers to perform. Therefore, if the channel were terminated at that time, it would not be restarted, causing potential overrun or underrun of the peripheral.

The proposed mechanism is for the channel to check this register after it has performed the "watermark" number of accesses to the peripheral. If the bit for the DMA request that triggers this channel is set, it means there is still another watermark number of data to transfer. This goes on until the bit is cleared. The same script can be used for multiple channels that require this behavior. The script can determine its channel number from the CCR register and infer the corresponding DMA request bit to check. It needs a reference table that is coherent with the request-channel matrix that the Arm platform programmed.

Address: 20E_C000h base + 14h offset = 20E_C014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EVENTS																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_EVENTS field descriptions

Field	Description
EVENTS	Reflects the status of the SDMA's external DMA requests. It is meant to allow any channel to monitor the states of these SDMA inputs. This register displays EVENTS 0-31. The EVENTS2 register displays events 32-47.

55.10.6 Current Channel Priority (SDMACORE_CCPRI)

Address: 20E_C000h base + 18h offset = 20E_C018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																													CCPRI		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_CCPRI field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
CCPRI	Contains the 3-bit priority of the channel whose context is installed. It is 0 when no channel is running. **NOTE:** 1-7 current channel priority 0 no running channel

55.10.7 Next Channel Priority (SDMACORE_NCPRI)

Address: 20E_C000h base + 1Ch offset = 20E_C01Ch

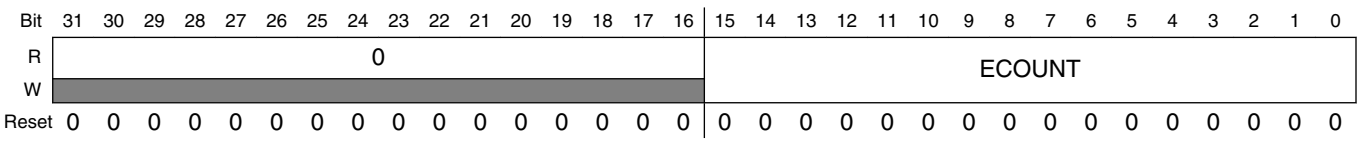
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																													NCPRI		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_NCPRI field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
NCPRI	Contains the 3-bit priority of the channel the scheduler has selected to run next. It is 0 when no other channel is pending.

55.10.8 OnCE Event Cell Counter (SDMACORE_ECOUNT)

Address: 20E_C000h base + 20h offset = 20E_C020h

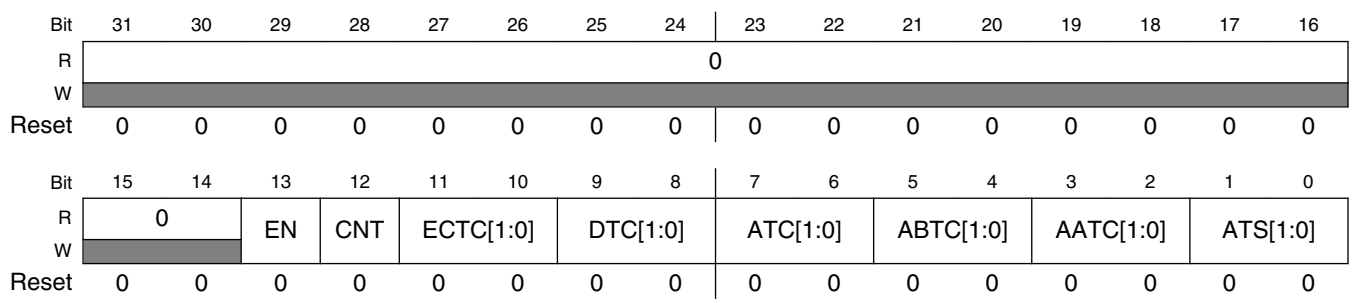


SDMACORE_ECOUNT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
ECOUNT	The event cell counter contains the number of times minus one that an event detection must occur before generating a debug request. • This register should be written before any attempt to use the event detection counter during an event detection process. • The counter is cleared on a JTAG reset.

55.10.9 OnCE Event Cell Control Register (SDMACORE_ECTL)

Address: 20E_C000h base + 24h offset = 20E_C024h



SDMACORE_ECTL field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
13 EN	Event Cell Enable. If the EN bit is set, the event cell is allowed to generate debug requests (the cell is awakened). If it is cleared, the event detection unit is disabled and no hardware breakpoint is generated, but matching conditions are still reflected on the emulation pin. 0 Cell is disabled. 1 Cell is enabled.
12 CNT	Event Counter Enable. The event counter enable bit determines if the cell counter is used during the event detection. In order to use the event counter during an event detection process, the event cell counter register should be loaded with a value equal to the number of times minus one that an event occurs before

Table continues on the next page...

SDMACORE_ECTL field descriptions (continued)

Field	Description
	a debug request is sent. After every event detection, the counter is decreased. When the counter reaches the value 0, the event detection cell sends a debug request to the core. The event counter register should be written and the EN bit should be set before each new event detection process uses the event counter. 0 Counter is disabled. 1 Counter is enabled.
11–10 ECTC[1:0]	The event cell trigger condition bits select the combination of address and data matching conditions that generate the final address/data condition. During program execution, if this event cell trigger condition goes to 1, a debug request is sent to the SDMA. The EN bit must be set to enable the debug request generation. 00 address ONLY 01 data ONLY 10 address AND data 11 address OR data
9–8 DTC[1:0]	The data trigger condition bits define when data is considered matching after comparison with the data register of the event detection unit. The operations are performed on unsigned values. 00 equal 01 not equal 10 greater than 11 less than
7–6 ATC[1:0]	The address trigger condition bits select how the two address conditions (addressA and addressB) are combined to define the global address matching condition. The supported combinations are described, as follows. 00 addressA ONLY 01 addrA AND addrB 10 addrA OR addrB 11 reserved
5–4 ABTC[1:0]	The Address B Trigger Condition (ABTC) controls the operations performed by address comparator B. All operations are performed on unsigned values. This comparator B outputs the addressB condition. 00 equal 01 not equal 10 greater than 11 less than
3–2 AATC[1:0]	The Address A Trigger Condition (AATC) controls the operations performed by address comparator A. All operations are performed on unsigned values. This comparator A outputs the addressA condition. 00 equal 01 not equal 10 greater than 11 less than
ATS[1:0]	The access type select bits define the memory access type required on the SDMA memory bus. 00 read ONLY 01 write ONLY 10 read or write 11 -

55.10.10 OnCE Event Address Register A (SDMACORE_EAA)

Address: 20E_C000h base + 28h offset = 20E_C028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																EAA															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_EAA field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
EAA	Event Cell Address Register A computes an address A condition. It is cleared on a JTAG reset.

55.10.11 OnCE Event Cell Address Register B (SDMACORE_EAB)

Address: 20E_C000h base + 2Ch offset = 20E_C02Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																EAB															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_EAB field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
EAB	Event Cell Address Register B computes an address B condition. It is cleared on a JTAG reset.

55.10.12 OnCE Event Cell Address Mask (SDMACORE_EAM)

Address: 20E_C000h base + 30h offset = 20E_C030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																EAM															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_EAM field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
EAM	The Event Cell Address Mask contains a user-defined address mask value. This mask is applied to the address value latched from the memory address bus before performing the address comparison. **NOTE:** There is a common address mask value for both address comparators. If bit *i* of this register is set, then bit *i* of the address value latched from the memory bus does not influence the result of the address comparison. The register is cleared on a JTAG reset.

55.10.13 OnCE Event Cell Data Register (SDMACORE_ED)

Address: 20E_C000h base + 34h offset = 20E_C034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ED																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_ED field descriptions

Field	Description
ED	The event cell data register contains a user defined data value. This data value is an input for the data comparator which generates the data condition. It is cleared on a JTAG reset.

55.10.14 OnCE Event Cell Data Mask (SDMACORE_EDM)

Address: 20E_C000h base + 38h offset = 20E_C038h

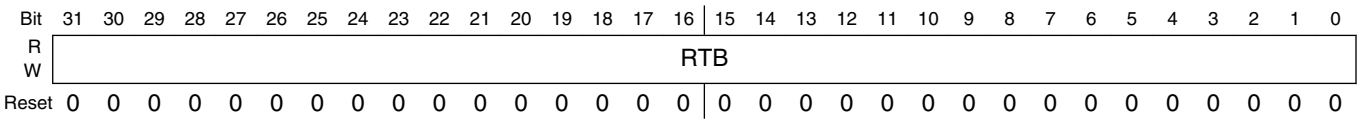
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EDM																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_EDM field descriptions

Field	Description
EDM	The event cell data mask register contains the user-defined data mask value. • This mask is applied to the data value latched from the memory bus before performing the data comparison. • Setting bit *i* of the event cell data mask register means that bit *i* of the data value latched from the address bus does not influence the result of the data comparison. • The data mask is cleared on a JTAG reset.

55.10.15 OnCE Real-Time Buffer (SDMACORE_RTB)

Address: 20E_C000h base + 3Ch offset = 20E_C03Ch

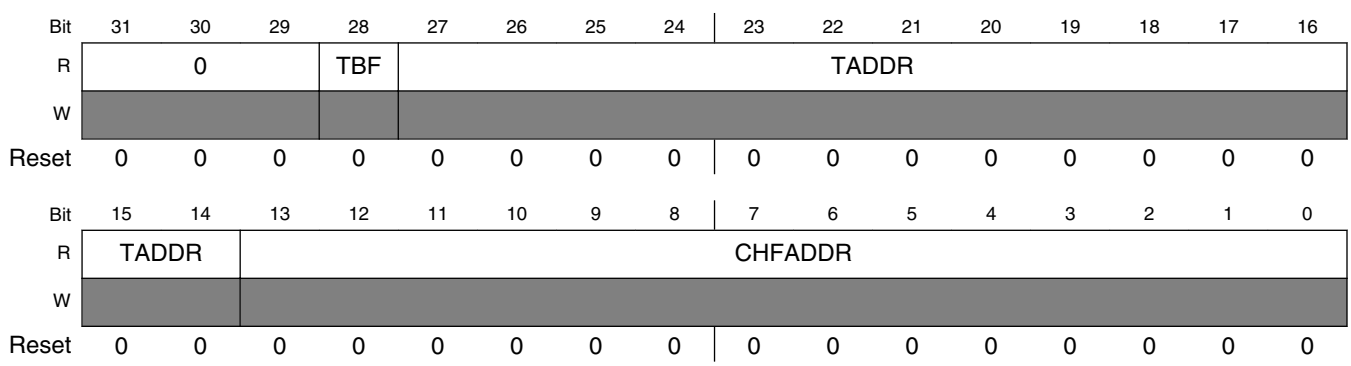


SDMACORE_RTB field descriptions

Field	Description
RTB	The Real Time Buffer register stores and retrieves run time information without putting the SDMA in debug mode. Writing to that register triggers a pulse on a specific real-time debug pin whose connection depends on the chip implementation. The RTB value can be accessed by the OnCE under Arm platform or JTAG control using the rbuffer command.

55.10.16 OnCE Trace Buffer (SDMACORE_TB)

Address: 20E_C000h base + 40h offset = 20E_C040h

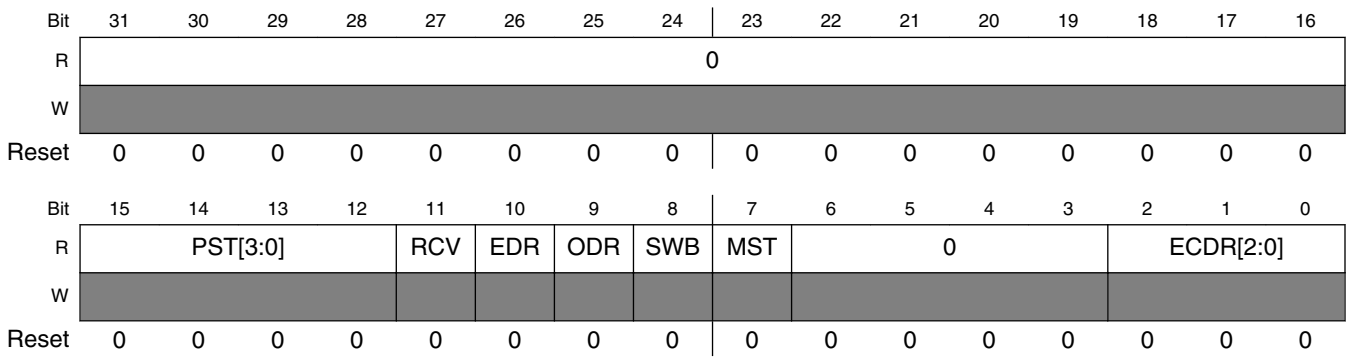


SDMACORE_TB field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28 TBF	The Trace Buffer Flag is set when the buffer contains the addresses of a valid change of flow. The contents of the buffer should be ignored otherwise. 0 Invalid information 1 Valid information
27–14 TADDR	The target address is the address taken after the execution of the change of flow instruction.
CHFADDR	The change of flow address is the address where the change of flow is taken when executing a change of flow instruction.

55.10.17 OnCE Status (SDMACORE_OSTAT)

Address: 20E_C000h base + 44h offset = 20E_C044h



SDMACORE_OSTAT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–12 PST[3:0]	The Processor Status bits reflect the state of the SDMA RISC engine. • The "Program" state is the usual instruction execution cycle. • The "Data" state is inserted when there are wait-states during a load or a store on the data bus (ld or st). • The "Change of Flow" state is the second cycle of any instruction that breaks the sequence of instructions (jumps and channel-switching instructions). • The "Change of Flow in Loop" state is used when an error causes a hardware loop exit. • The "Debug" state means the SDMA is in debug mode. • The "Functional Unit" state is inserted when there are wait-states during a load or a store on the functional units bus (ldf or stf). • In "Sleep" modes, no script is running (this is the RISC engine idle state). The "after Reset" is slightly different because no context restoring phase will happen when a channel is triggered: The script located at address 0 will be executed (boot operation). • The "in Sleep" states are the same as above except they do not have any corresponding channel. They are used when entering debug mode after reset; the reason is that it is necessary to return to the "Sleep after Reset" state when leaving debug mode. 0 Program 1 Data 2 Change of Flow 3 Change of Flow in Loop 4 Debug 5 Functional Unit 6 Sleep 7 Save 8 Program in Sleep 9 Data in Sleep 10 Change of Flow in Sleep 11 Change Flow Loop Sleep 12 Debug in Sleep 13 Functional Unit in Sleep

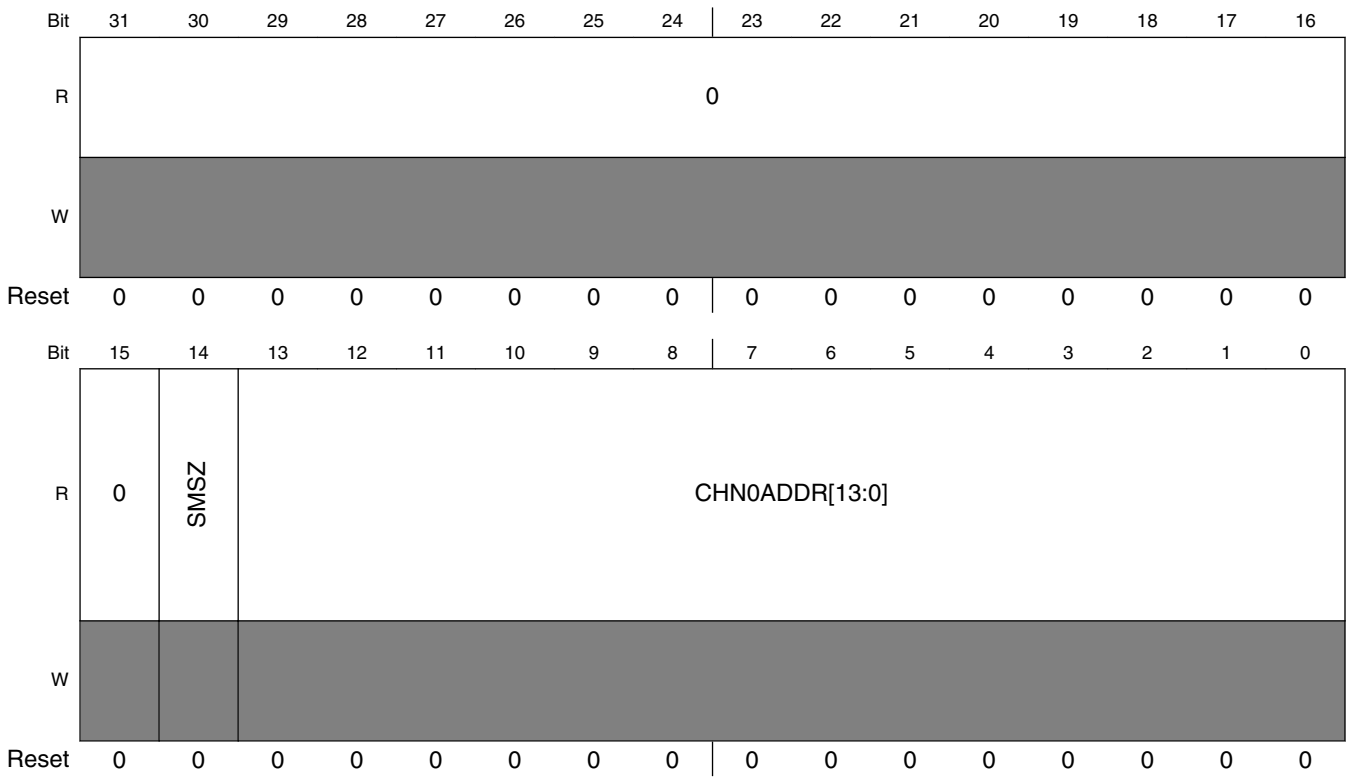
Table continues on the next page...

SDMACORE_OSTAT field descriptions (continued)

Field	Description
	14 Sleep after Reset 15 Restore
11 RCV	After each write access to the real time buffer (RTB), the RCV bit is set. This bit is cleared after execution of an `rbuffer` command and on a JTAG reset.
10 EDR	This flag is raised when the SDMA has entered debug mode after an external debug request.
9 ODR	This flag is raised when the SDMA has entered debug mode after a OnCE debug request.
8 SWB	This flag is raised when the SDMA has entered debug mode after a software breakpoint.
7 MST	This flag is raised when the OnCE is controlled from the Arm platform peripheral interface. 0 JTAG interface controls the OnCE. 1 Arm platform peripheral interface controls the OnCE.
6–3 Reserved	This read-only field is reserved and always has the value 0.
ECDR[2:0]	Event Cell Debug Request. If the debug request comes from the event cell, the reason for entering debug mode is given by the EDR bits. The encoding of the EDR bits is useful to find out more precisely why the debug request was generated. A debug request from an event cell is generated for a specific combination of the addressA, addressB, and data conditions; the value of those fields is given by the EDR bits. If all three bits of the EDR are reset, then it did not generate any debug request. If the cell did generate a debug request, then at least one EDR bit is set; the meaning of the encoding is as follows: 0 1 matched addressA condition 1 1 matched addressB condition 2 1 matched data condition

55.10.18 Channel 0 Boot Address (SDMACORE_MCHN0ADDR)

Address: 20E_C000h base + 48h offset = 20E_C048h

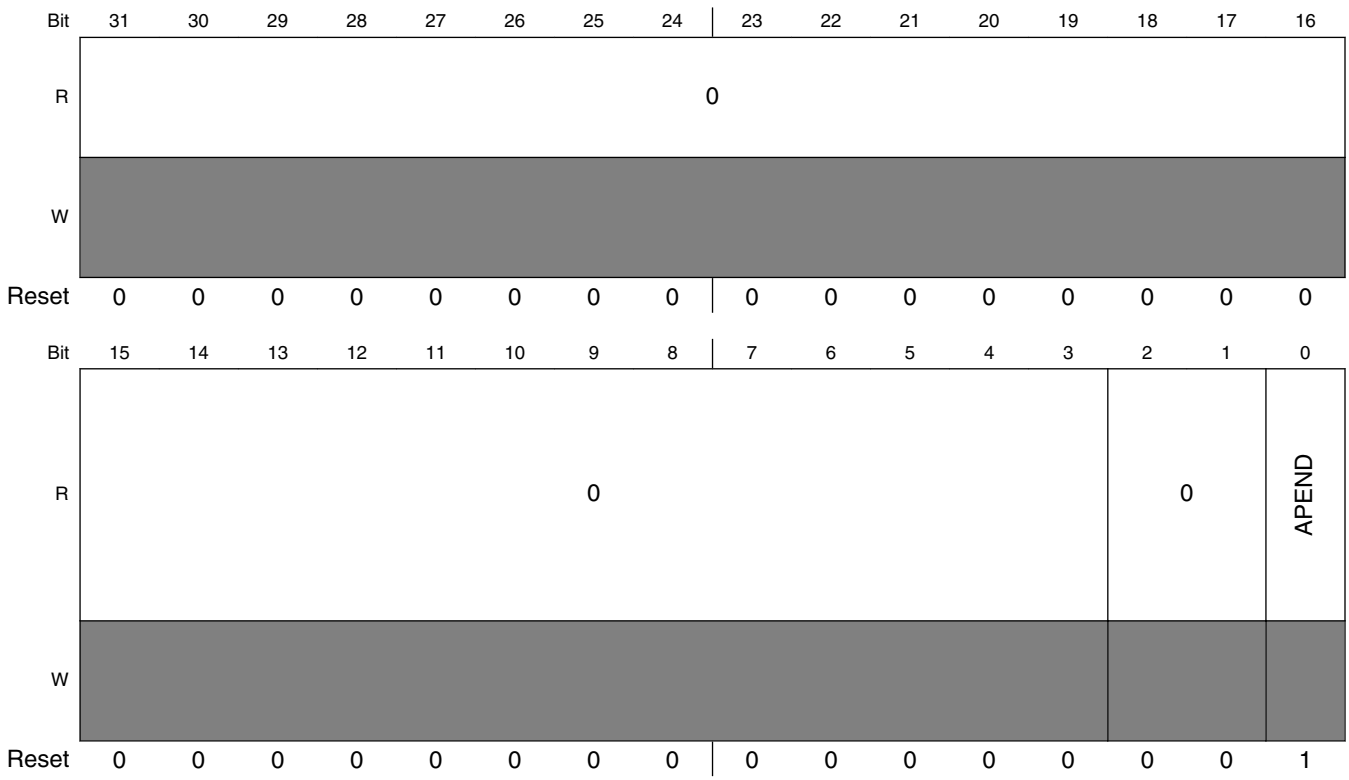


SDMACORE_MCHN0ADDR field descriptions

Field	Description
31–15 Reserved	This read-only field is reserved and always has the value 0.
14 SMSZ	The bit 14 (Scratch Memory Size) determines if scratch memory must be available after every channel context. After reset, it is equal to 0, which defines a RAM space of 24 words for each channel. All of this area stores the channel context. By setting this bit, 32 words are reserved for every channel context, which gives eight additional words that can be used by the channel script to store any type of data. Those words are never erased by the context switching mechanism. 0 24 words per context 1 32 words per context
CHN0ADDR[13:0]	Contains the address of the channel 0 routine programmed by the Arm platform; it is loaded into a general register at the very start of the boot and the SDMA jumps to the address it contains. By default, it points to the standard boot routine in ROM.

55.10.19 ENDIAN Status Register (SDMACORE_ENDIANNESS)

Address: 20E_C000h base + 4Ch offset = 20E_C04Ch

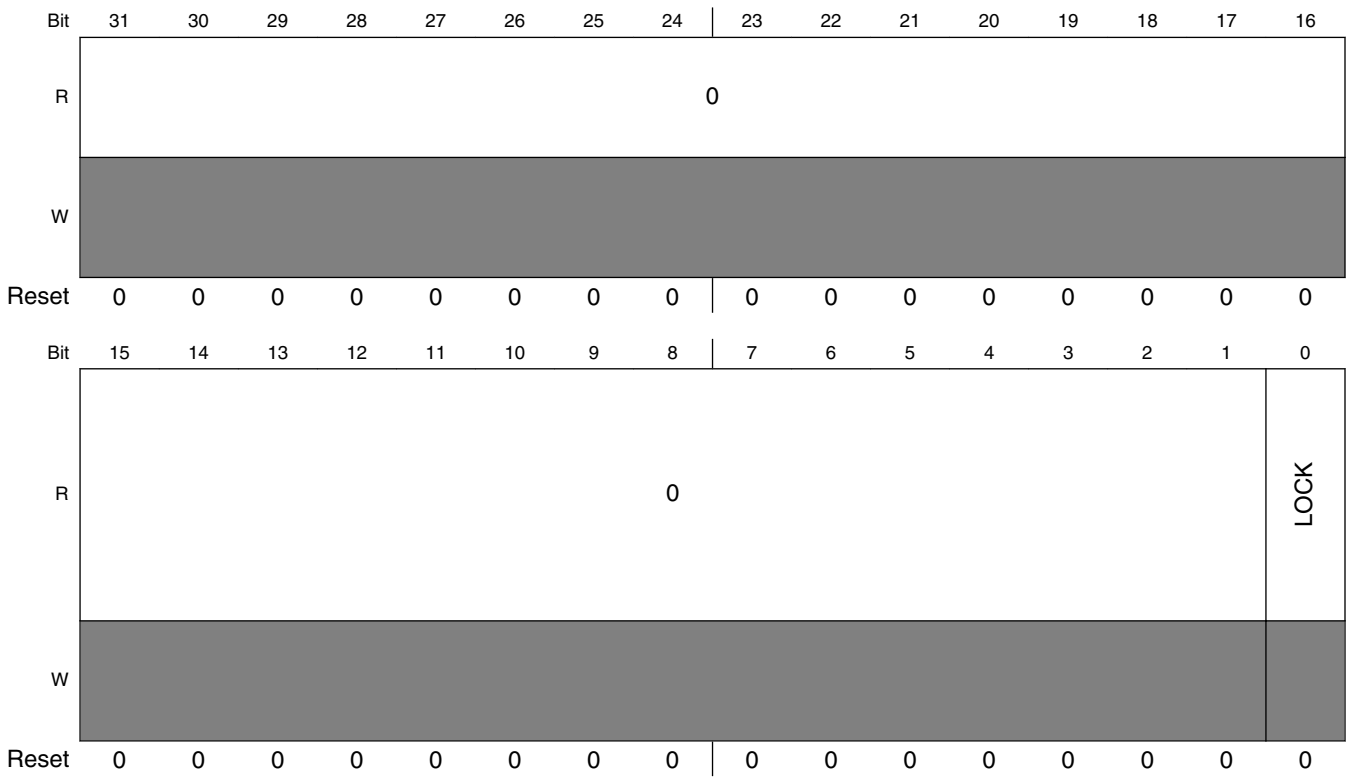


SDMACORE_ENDIANNESS field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 APEND	APEND indicates the endian mode of the Peripheral and Burst DMA interfaces. This bit is tied to logic '1' indicating little-endian mode. 0 - Arm platform is in big-endian mode 1 - Arm platform is in little-endian mode

55.10.20 Lock Status Register (SDMACORE_SDMA_LOCK)

Address: 20E_C000h base + 54h offset = 20E_C054h



SDMACORE_SDMA_LOCK field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 LOCK	The LOCK bit reports the value of the LOCK bit in the SDMA_LOCK status register. SDMA software may use this value to determine if certain operations such as loading of new scripts is allowed. 0 - LOCK bit clear 1 - LOCK bit set

55.10.21 External DMA Requests Mirror #2 (SDMACORE_EVENTS2)

Address: 20E_C000h base + 58h offset = 20E_C058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																EVENTS																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_EVENTS2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
EVENTS	Reflects the status of the SDMA's external DMA requests. It is meant to allow any channel to monitor the states of these SDMA inputs. This register displays EVENTS 32-47. The separate EVENTS register displays events 0-31.

55.11 SDMA Peripheral Registers

Refer to the respective peripherals' chapters for more information.

Chapter 56
System JTAG Controller (SJC)

56.1 Overview

The System JTAG Controller (SJC) provides debug and test control with the maximum security.

The test access port (TAP) is designed to support features compatible with the IEEE Standard 1149.1 v2001 (JTAG).

The figure below shows an overview of the JTAG architecture.

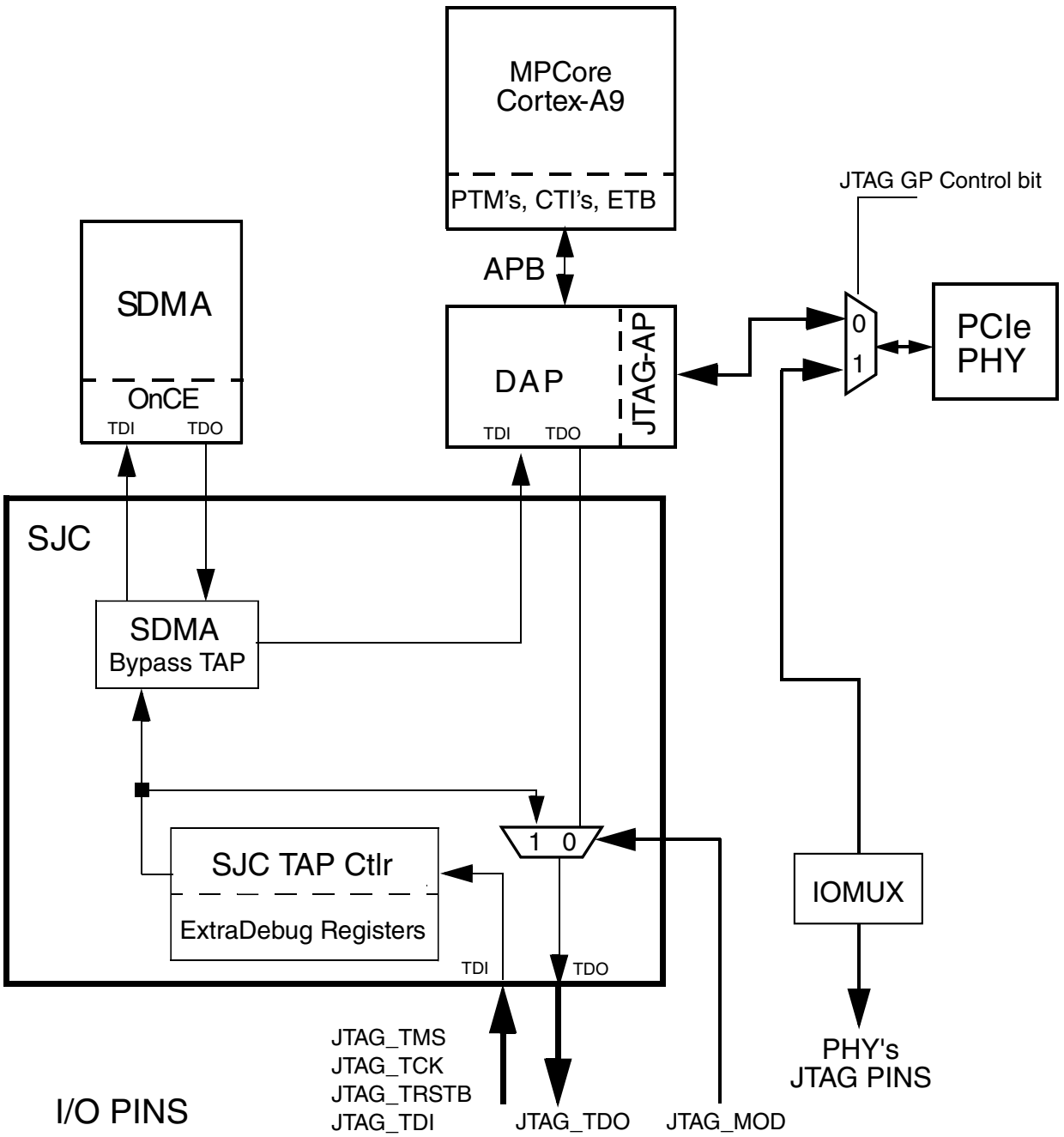


Figure 56-1. System JTAG Controller (SJC) Block Diagram

56.1.1 Features

The System JTAG Controller (SJC) provides the following capabilities:

- JTAG IEEE1149.1 mandatory instructions, see EXTEST Instruction, SAMPLE/PRELOAD Instruction , and BYPASS Instruction .

- JTAG IEEE1149.1 optional instructions, see SJC ID_CODE Instruction (SJC IDCODE), and HIGHZ Instruction.
- JTAG IEEE P1149.1 (standard JTAG) interface to off-chip test and development equipment including an SJC-only mode for true IEEE 1149.1 compliance, used primarily for board-level implementation of boundary scan.
- IEEE P1149.6 (JTAG) mandatory instructions, see EXTEST_PULSE instruction and EXTEST_TRAIN instruction. These two instructions enable edge-detecting behavior on the signal path containing AC pins.
- Debug-related control and status, such as putting selected cores into reset and/or debug mode and the ability to monitor individual core status signals via JTAG.
- Provides means for accessing each OnCE/ICE TAP controller independently to control a target system (see Modes of Operation).
- ExtraDebug logic (see ENABLE_ExtraDebug Instruction).
- The maximum clock speed of the SJC is one-eight of the lowest frequency of the accessed OnCE/ICE. For example in normal operation (no core in low-power mode), this frequency is one-eight of the SDMA frequency if this core is present in the TDI-TDO chain (serially connected with other cores or standalone). The user must also consider the 25 MHz frequency limitation on the CE bus.
- Core compliant modes to support standalone core debuggers (see Modes of Operation).
- Multi-cores daisy chained mode (default one) to support multi-core debuggers (see Modes of Operation).

Detailed information about the SJC is provided in the Security Reference Manual. Contact your NXP representative for information about obtaining this document.

56.1.2 Modes of Operation

The SJC modes are controlled through both the TAP select register (SJC_TSR) and the MOD input port.

The MOD port (typically connected to pad of the same name) selects between two possible topologies of TAP connections while using SJC functions, as seen at SoC level:

- Negating it (this should be the default state) selects all the TAPs (SJC, SDMA, DAP) to be connected in the TDI-TDO chain, which is referred to as "daisy chain" mode, throughout this chapter.
- Asserting it only selects the SJC TAP to be connected in the TDI-TDO chain.

IEEE1149.1 standard features are enabled by configuring the SJC input pin: MOD. Refer to the following table for MOD settings details:

Table 56-1. SJC Modes

MOD	Name	Description
0	Daisy chain ALL	For common SW debug (High speed and production)
1	SJC only	IEEE1149.1 JTAG compliant mode

The following figure shows the SJC mode selection flow. The numbers shown in parenthesis below each block name indicates the TAP's IR length.

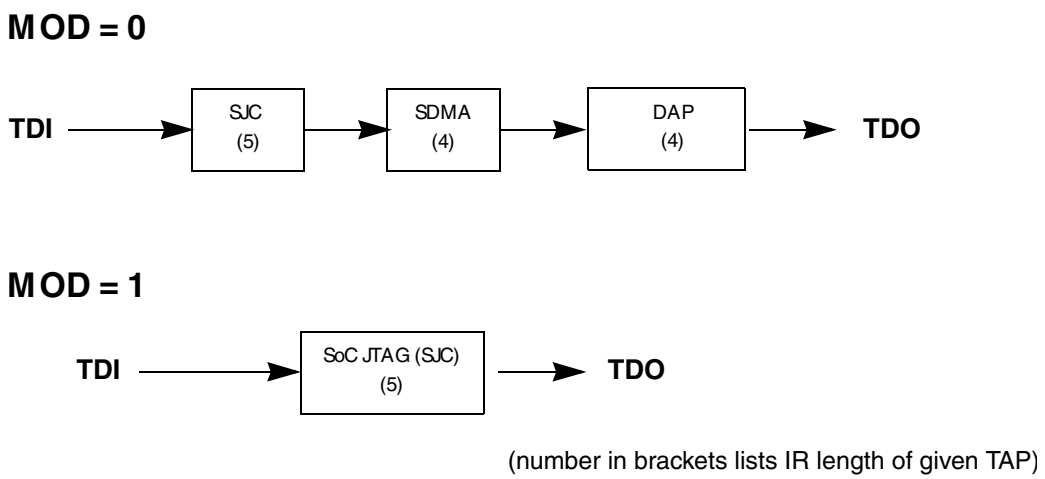


Figure 56-2. SJC Mode Selection Using MOD Pin Sampling

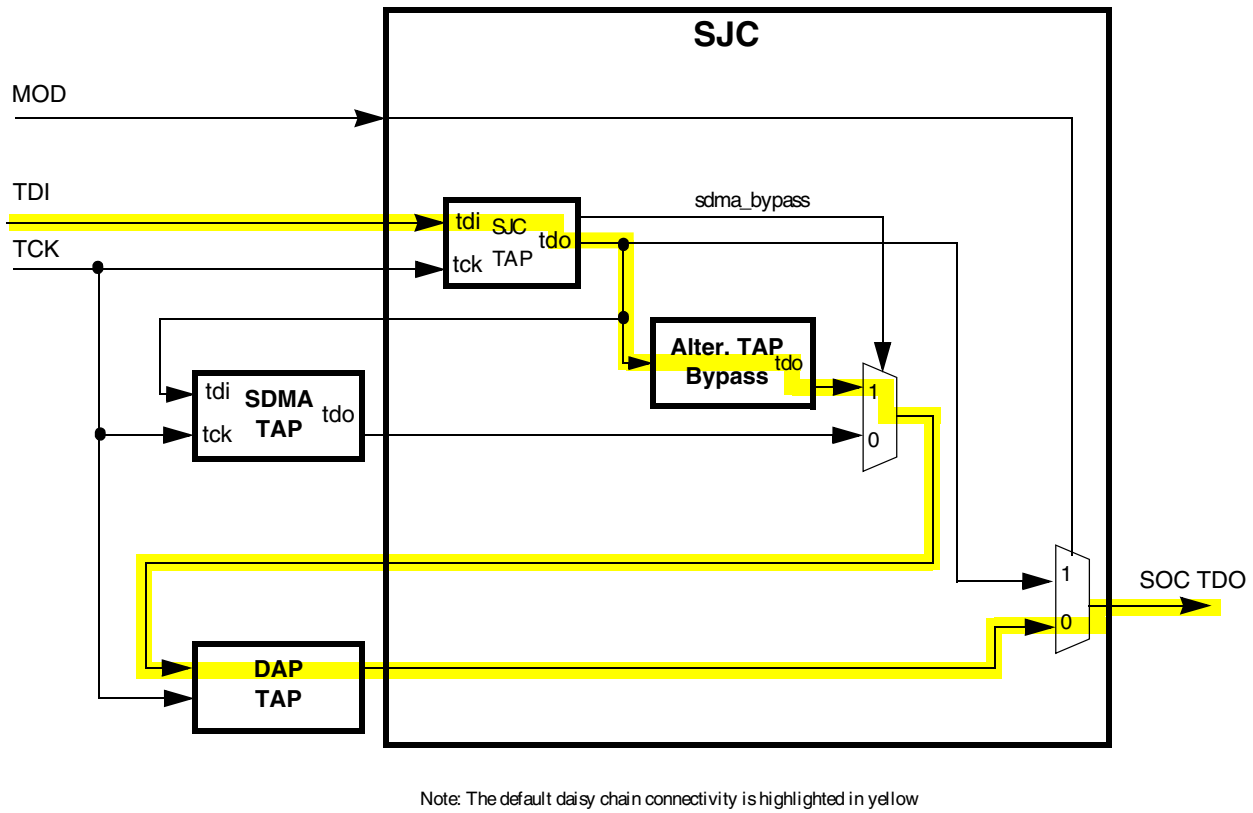
The Connect SDMA bit inside TAP select register controls the SDMA TAP bypass.

- When negated (should be the default state), the SDMA TAP is bypassed with a single D-FF (Flip-flop) during Shift-Dr path
- When asserted SDMA TAP is connected inside the chain
- When taking the SDMA into bypass or out of bypass (by writing to tapsel reg), additional cycle with TMS '0' should be given

The TAP selection block (TSB) provides a simple method of integrating various pieces of IP that have embedded TAPs.

- Provides a way to connect up multiple TAPs within a single SoC
- Identify the SJC TAP as the master TAP which controls the boundary chain (for IEEE 1149.1 standard compliance)
- Follow the state of SJC TAP, and when the Test-Logic-Reset (TLR) state is reached, reset all TAPs

The figure below shows the TAP Selection Block and SOC TAP Chain Scheme.



Note: The default daisy chain connectivity is highlighted in yellow

Figure 56-3. TAP Selection Block and SoC TAP Chain Scheme

NOTE

It is the responsibility of the user to ensure that in any configuration of the TAP controllers chosen, all of the TAPs in the chain comply with the demands of TCK clock frequency as well as the required ratio between TCK clock frequency and that of the core's to which the TAP refers.

56.2 External Signals

The table found here describes the external signals of SJC.

Table 56-2. SJC External Signals

Signal	Description	Pad	Mode	Direction
JTAG_DE_B (DE_B)	SoC debug request/acknowledge pin. The DE_IN_B pin is used to propagate an external debug request event to the core(s). This functionality must be enabled first, by set of DE_to_ARM /	GPIO_16	ALT7	I/O

Table continues on the next page...

Table 56-2. SJC External Signals (continued)

Signal	Description	Pad	Mode	Direction
	DE_to_SDMA bits in SJC's DCR register. It is SoC implementation dependent, whether this pin can also be used to reflect the debug acknowledge event back from the cores (in the case where an Open-Drain scheme is used externally).			
JTAG_MOD (MOD)	SJC mode selection. This pin is sampled at TRST reset to determine two possible modes for the TAP connection configuration.	JTAG_MOD	No Muxing	I
JTAG_TCK (TCK)	Test Clock (TCK). This is used to synchronize the test logic and includes an internal pull-up resistor	JTAG_TCK	No Muxing	I
JTAG_TDI (TDI)	Test Data Input (TDI). Serial test instruction and data are received through the test data input (TDI) pin. TDI is sampled on the rising edge of TCK and includes an internal pullup resistor	JTAG_TDI	No Muxing	I
JTAG_TDO (TDO)	Test Data Output (TDO). The serial output for test instructions and data. TDO is tri-statable and is actively driven in the shift-IR and shift-DR controller states. TDO changes on the falling edge of TCK	JTAG_TDO	No Muxing	O
JTAG_TMS (TMS)	Test Mode Select (TMS). This is used to sequence the test controller's state machine. TMS is sampled on the rising edge of TCK and includes an internal pullup resistor	JTAG_TMS	No Muxing	I
JTAG_TRSTB (TRSTB)	Test Reset (TRST). This is used to asynchronously initialize the test controller. The TRST pin has an internal pullup resistor	JTAG_TRSTB	No Muxing	I

56.2.1 External Signal Overview

The SJC provides test and debug control with a minimum number of contacts.

The figure below shows SJC connections to external contacts and other chip blocks.

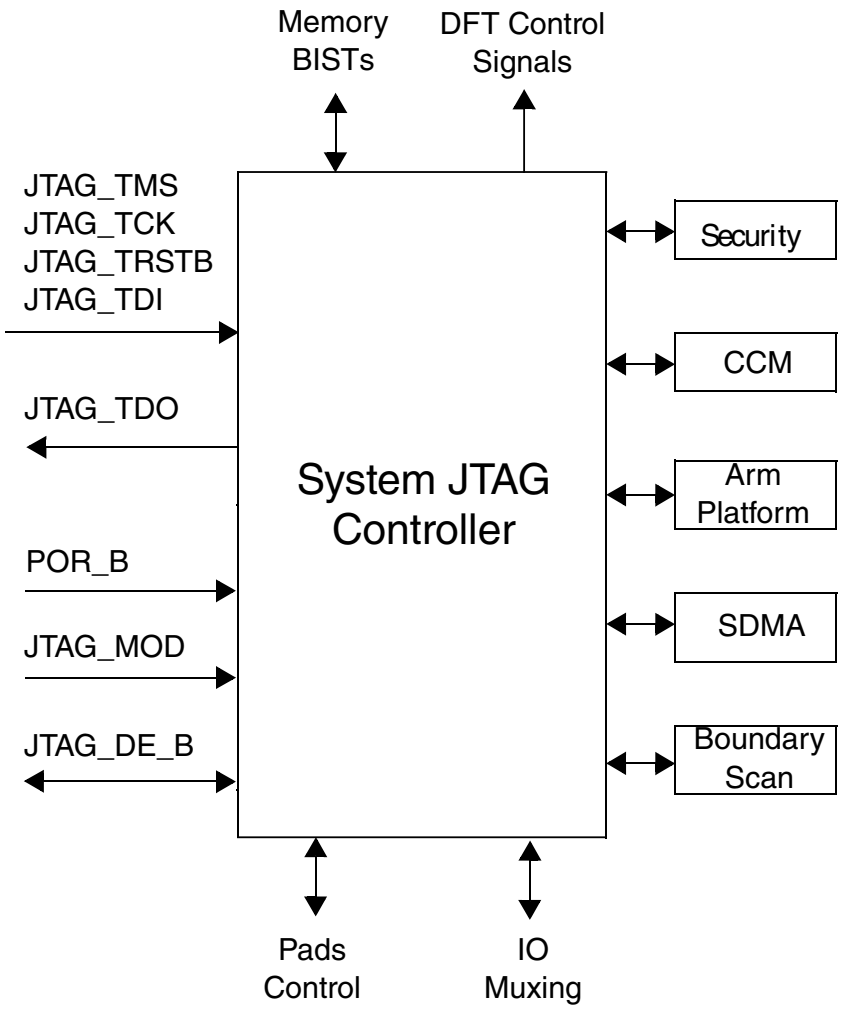


Figure 56-4. SJC Connections

56.2.2 TAP Controller

The TAP controller is responsible for interpreting the sequence of logical values on the TMS signal. It is a synchronous state machine that controls the operation of the JTAG logic. The value shown adjacent to each arc represents the value of the TMS signal sampled on the rising edge of TCK signal. For a description of the TAP controller states, refer to the appropriate IEEE 1149.1 document.

The state machine is shown in the following figure.

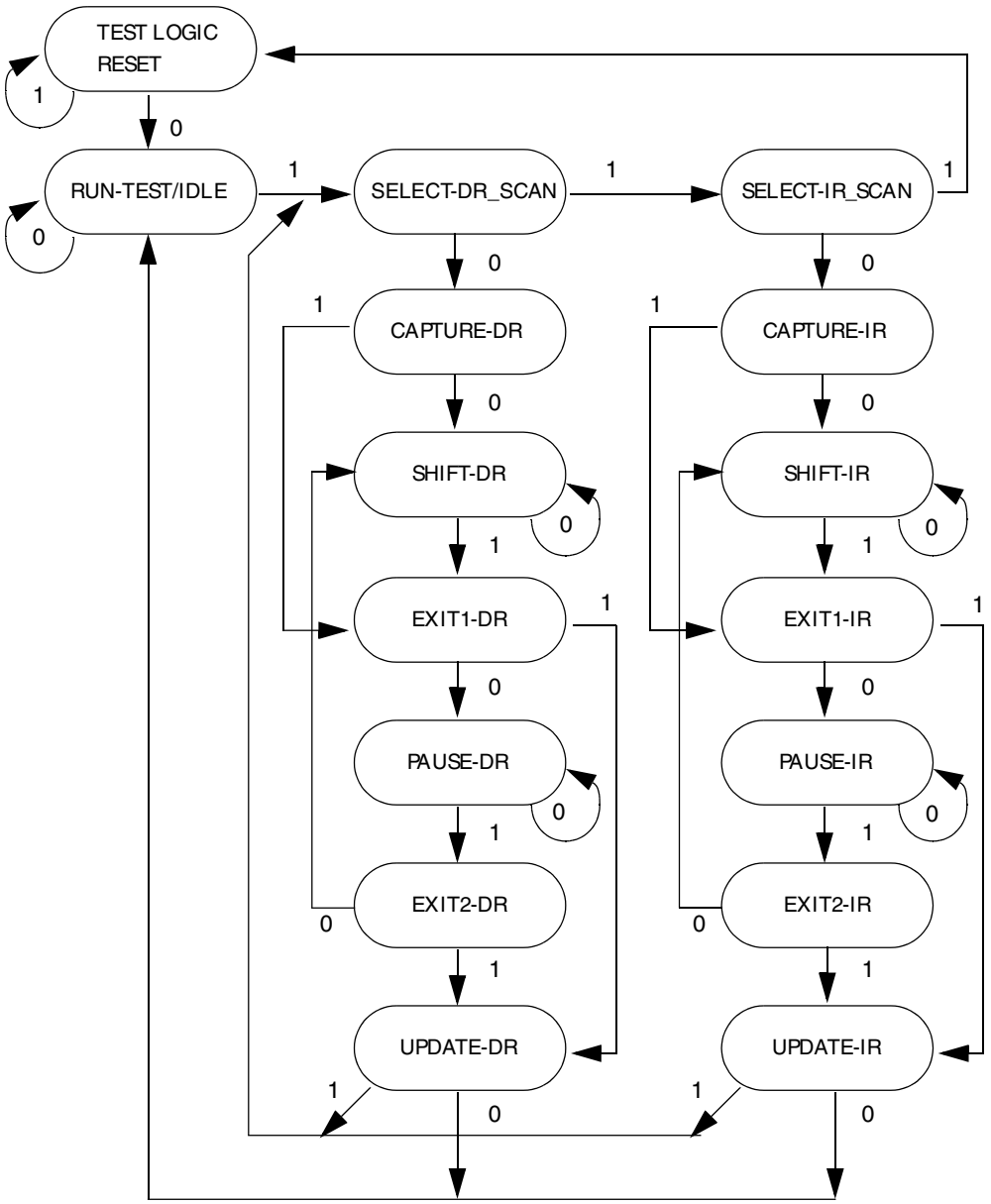


Figure 56-5. TAP Controller State Machine

The change of the JTAG state machine occurs on the rising edge of TCK. TMS and TDI change on the falling edge of TCK. TDO also changes on the falling edge of TCK following entry into the Shift_DR or Shift_IR states (TDO_EN is the enable of the tristate buffer driving the TDO output).

The figure below shows the timings of the SJC signals.

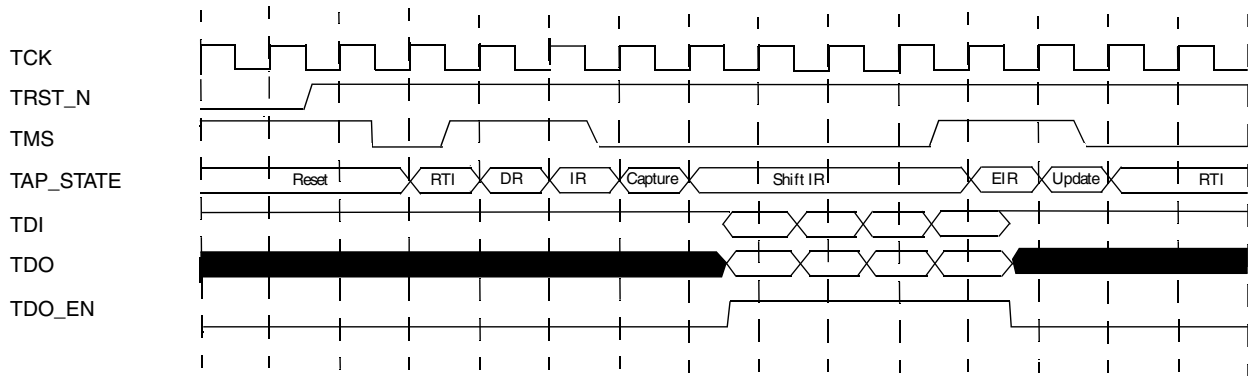


Figure 56-6. SJC Signals Timing Diagram

56.2.3 Accessing ExtraDebug Registers

Accessed through the Select-DR-Scan path, the ExtraDebug shift register consists of 38 bits (maximum) comprising a 32-bit data field (max length, see extradebug register description), a 5 bit address field and read/write bit.

The write actually takes place when the JTAG TAP controller enters the Update-DR state. On a read, the data field is ignored (the user should shift only 5 times to enter Read=1 and the address), the read takes place on the next path through DR at the Capture-DR state, the data is shifted-out during the Shift-DR state.

On the second path for a read access, simultaneous write access is not supported: command converter software shifts in zeros so the TAP decodes a write to the CSR (read-only register) which does not have any effect on the circuit.

The number of shift depends on the width of the accessed register as explained in the following diagrams.

First a write access (one path through Select-DR-Scan):

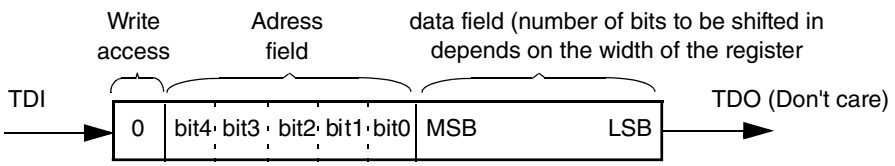


Figure 56-7. TDI/TDO on write access

Then a read access (requires two paths through Jtag DR Scan path):

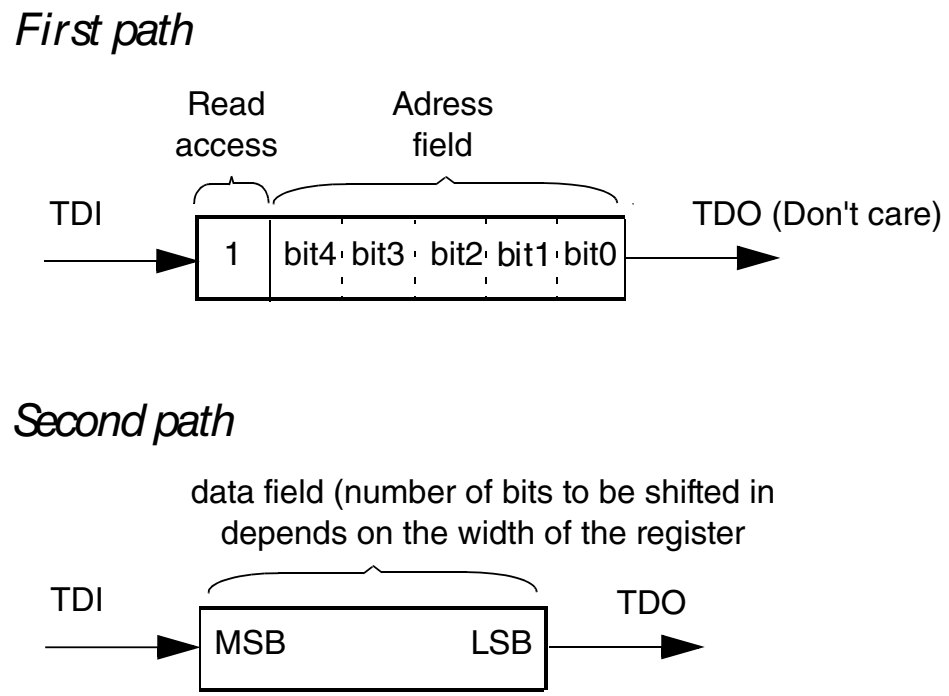


Figure 56-8. TDI/TDO on Read Access

For example, write value 0b1010_1100 to Debug Control Register (address = 0b00110).

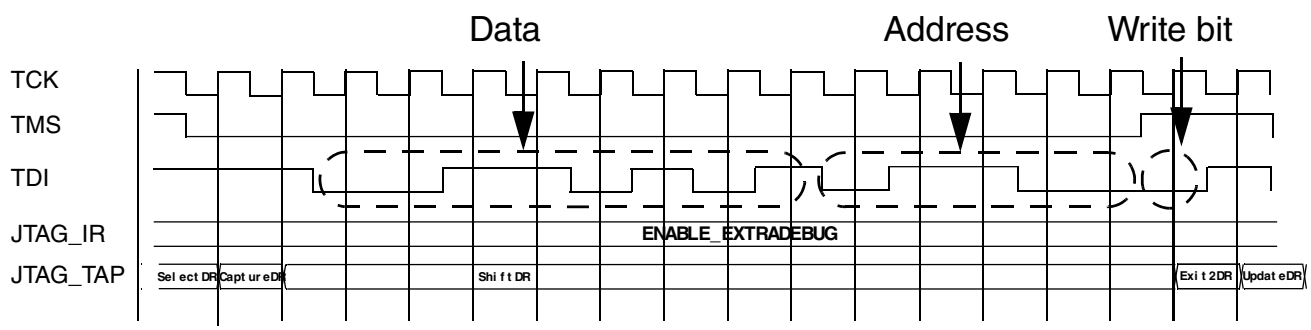


Figure 56-9. Example: Write Access to DCR

The SJC registers have different levels of security (Refer to JTAG Security Modes):

- Secured- accessible only in mode 2 (supposed correct response entered), mode 3 and mode 4.
- Unsecured- accessible in all modes

The level of security of each register is indicated in its name or description, in "Programmable Registers" section.

A single DE_B pin is dedicated for debug request input/output in bidirectional open drain functionality (including an internal pull-up device).

Bits 6:5 in DCR register serve as mask bits, controlling the propagation of external debug request to each recipients (Arm Platform, SDMA).

The bits 1:0 define the propagation enable of IR debug request to recipient cores.

The following figure shows the DE_B Pin Select Logic.

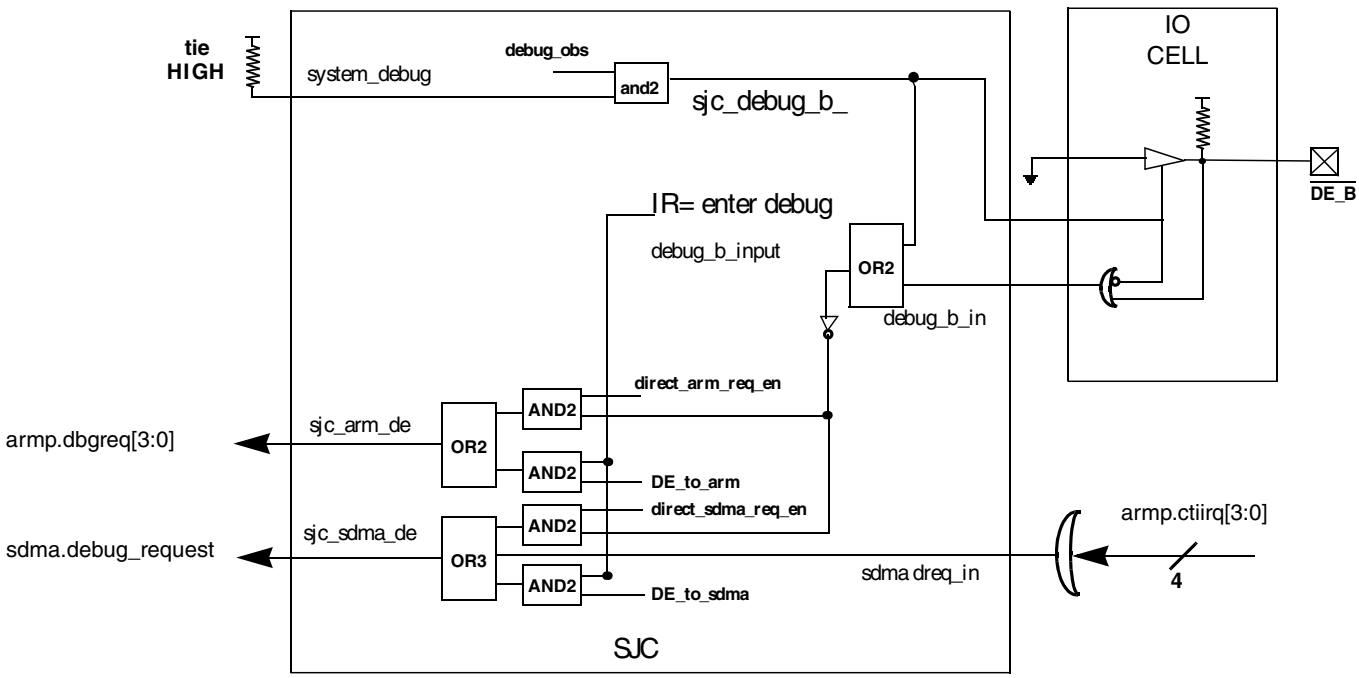


Figure 56-10. DE_B Pin Select Logic

For security reasons, bits for output and input propagation control are at their negated values after reset. A user cannot put the cores in debug mode through DE_B without any Jtag access.

The configuration after reset prevents propagation of debug requests / acknowledges to or from the cores.

56.3 TAP Selection Block (TSB)

As described in Modes of Operation, the SJC can access cores in different modes selected through a TSB.

56.3.1 Select Mode Using Software

Conceptually, the SJC_TSR is a data register which is accessed through Access TSR IR instruction of SJC TAP.

The following figure shows the process of using reserved IR to access the SJC_TSR.

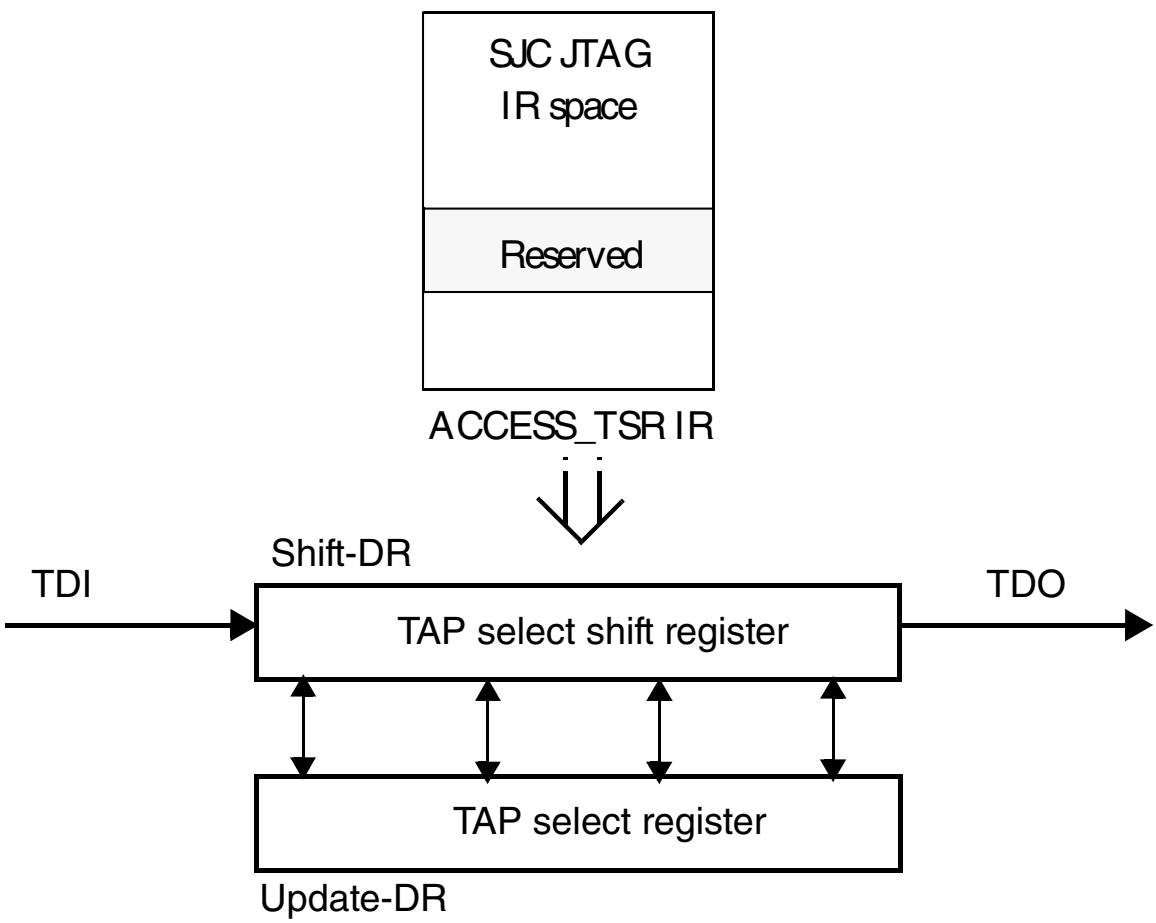


Figure 56-11. Using Reserved IR to Access the TAP Select Register (SJC_TSR)

The SJC_TSR can only be changed during the update-DR state of the TSB JTAG state machine. This is necessary to prevent a TAP that is being selected from losing synchronization with the TSB state machine when the TSB state machine returns to run-test-idle. Therefore, an associated shift register for the SJC_TSR is loaded into the

SJC_TSR during the update-DR state (see the figure above). The shift register must also capture the state of the SJC_TSR when in the Capture-DR state for visibility of the contents of the SJC_TSR. See TAP Select Instruction , for more information.

56.4 Boundary Scan Register (BSR)

The Boundary Scan Register (BSR) in the JTAG implementation contains bits for all device signal and clock pins and associated control signals.

All SoC bidirectional pins have a single register bit in the boundary scan register for pin data, and are controlled by an associated control bit in the boundary scan register.

56.5 SJC Instruction Register (SJIR)

The SJC Instruction register is provided in the following table. The SJC Instruction register is 5 bits wide.

Table 56-3. SJC Instruction Register (SJIR)

Code					SJC IR
B4	B3	B2	B1	B0	
0	0	0	0	0	SJC IDCODE
0	0	0	0	1	SAMPLE/PRELOAD
0	0	0	1	0	EXTEST
0	0	0	1	1	HI-Z
0	0	1	0	0	ENABLE_ExtraDebug
0	0	1	0	1	ENTER_DEBUG (secured)
0	0	1	1	0	Reserved
0	0	1	1	1	TAP select
0	1	0	0	0	EXTEST_PULSE
0	1	0	0	1	EXTEST_TRAIN
0	1	0	1	0	Reserved
0	1	0	1	1	Reserved
0	1	1	0	0	Security Output challenge
0	1	1	0	1	Security Enter response
-					Reserved
1	1	1	1	1	BYPASS

The instruction register is reset to 0b00000 in the test-logic-reset controller state which is equivalent to the SJC IDCODE instruction.

During the capture-IR controller state, the parallel inputs to the instruction register are loaded with the code 01 in the least significant bits as required by the standard; the most significant bits are loaded with the values 00, leading to a capture value of 0b00001.

56.5.1 SJC ID_CODE Instruction (SJC IDCODE)

Selects the ID register, and the system logic controls the I/O pins. This instruction is provided as a public instruction to allow the manufacturer, part number and version of a component to be determined through the TAP.

The table below shows the ID register configuration.

Table 56-4. ID Configuration Register (IDCODE)

IDCODE				ID Configuration Register												
	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	Version Information[3:0]				Part Number (Bits 27-16)											
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x
Note:																
	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	Part Number (Bits 15-12)				Manufacturer Identity											1
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	x	x	x	x	0	0	0	0	0	0	0	1	1	1	0	1
Note:																

Table 56-5. SJC ID Configuration Register Description (SJC IDCODE)

Field	Description
31-28 Version Information	IC/SoC Version information number. Initial value: '0000' This number is subject to changes, for new IC/SoC (System On A Chip) revision releases.
27-12 Part Number	Customer Part Number The 16-bit Part Number value is unique for every NXP SoC / IC. See System Debug chapter for exact register value for a specific SoC.
11-1 Manufacturer Identity	Manufacturer Identity NXP Manufacturer Identity code.

Table continues on the next page...

Table 56-5. SJC ID Configuration Register Description (SJC IDCODE) (continued)

	Bits [11:1] - 00000001110
0	Tied to logic 1.

One application of the ID register is to distinguish the manufacturer(s) of components on a board when multiple sourcing is used. As more components emerge which conform to the IEEE 1149.1 standard, it is desirable to allow for a system diagnostic controller unit to blindly interrogate a board design to determine the type of each component in each location. This information is also available for factory process monitoring and for failure mode analysis of assembled boards.

Once the IDCODE instruction is decoded, it selects the ID register which is a 32 Bit data register. Because the bypass register loads a logic 0 at the start of a scan cycle, whereas the ID register loads a logic 1 into its least significant bit, examination of the first bit of data shifted out of a component during a test data scan sequence immediate following exit from Test-Logic-Reset controller state shows whether such a register is included in the design. When the IDCODE instruction is selected, the operation of the test logic has no effect on the operation of the on-chip system logic as required by the IEEE 1149.1 standard.

56.5.2 SAMPLE/PRELOAD Instruction

Selects the boundary scan register and the system logic controls the I/O pins.

The SAMPLE/PRELOAD instruction provides two separate functions:

- First, it provides a means to obtain a snapshot of system data and control signals. The snapshot occurs on the rising edge of TCK in the capture-DR controller state. The data can be observed by shifting it transparently through the boundary scan register.
- The second function of SAMPLE/PRELOAD is to initialize the boundary scan register output cells prior to selection of EXTEST. This initialization ensures that known data appears on the outputs when entering the EXTEST instruction.

NOTE

Because there is no internal synchronization between the JTAG clock (TCK) and the system clock (CLK), the user must provide some form of external synchronization to achieve meaningful results.

For more details on the function and use of SAMPLE/PRELOAD, refer to the appropriate IEEE 1149.1 document.

56.5.3 EXTEST Instruction

Selects the boundary scan register, and the 1149.1 test logic has control of the I/O pins.

By using the TAP controller, the register is capable of:

- Scanning user-defined values into the output buffers,
- Capturing values presented to input pins
- Controlling the direction of bidirectional pins,
- Controlling the output drive of tri-statable output pins.

For more details on the function and use of EXTEST, refer to the appropriate IEEE 1149.1 document.

The EXTEST instruction also asserts internal reset for the cores (through CCM, refer to Figure 56-14) to force a predictable internal state while performing external boundary scan operations.

56.5.4 HIGHZ Instruction

All output drivers, including the two-state drivers, are turned off (that is, high impedance). The instruction selects the bypass register.

In this mode, all internal pullup resistors on all the pins (except for the TMS, TDI, TCK, TRSTB pins) are disabled. This disabling functionality is not built into SJC, but should be implemented by some logic in the SOC/IO Pads.

For more details on the function and use of HIGHZ, refer to the IEEE 1149.1 document.

The HIGHZ instruction also asserts internal reset for the cores (through CCM, refer to Figure 56-14) to force a predictable internal state while performing external boundary scan operations.

56.5.5 BYPASS Instruction

Selects the single Bit bypass register and the system logic controls the I/O pins.

This creates a shift-register path from TDI to the bypass register and, finally, to TDO, circumventing the boundary scan register. This instruction is used to enhance test efficiency when a component other than the SoC Core based device becomes the device under test.

When the bypass register is selected by the current instruction, the shift-register stage is set to a logic zero on the rising edge of TCK in the capture-DR controller state. Therefore, the first bit to be shifted out after selecting the bypass register is always a logic zero.

For more details on the function and use of BYPASS, refer to the appropriate IEEE 1149.1 document.

56.5.6 ENABLE_ExtraDebug Instruction

The TDI and TDO pins are connected directly to the ExtraDebug registers, the SJC TAP controller remaining connected to TDI and TMS.

The ExtraDebug shift register consists of 38 bits (maximum) comprising a 32-bits data field (maximum length, see Accessing ExtraDebug Registers,), a 5 bits address field and read/$\overline{\text{write}}$ bit. On a register read, the data field does not need to be filled in. The particular ExtraDebug register connected between TDI and TDO at a given time is selected by the ExtraDebug controller depending on the ExtraDebug Address being currently decoded. All communication with the ExtraDebug controller is done through the Select-DR-Scan path of the JTAG TAP Controller.

56.5.7 ENTER_DEBUG instruction

The ENTER_DEBUG instruction is used to generate a debug request event to SDMA and the Arm Core Platform simultaneously (practically, inherited minimal skew is expected, due to difference in event signal propagation in the different modules).

The TDI and TDO are connected to the Instruction Register (IR). After the acknowledgment of the Debug Mode is received (can be checked by reading the Core Status Register part of the ExtraDebug logic), the user can perform system debug functions on the cores.

NOTE

The ENTER_DEBUG event issue to the cores, can be masked, by bits in DCR register.

NOTE

It is user's responsibility to shift-in another IR value (like IDCODE) before trying to bring the cores out of debug mode, as the debug request signals to the cores remains asserted as long as ENTER_DEBUG IR is in place.

NOTE

The user need to check that cores are in debug mode (watching debug acknowledge signal) before leaving ENTER_DEBUG instruction, otherwise debug request might not take affect.

56.5.8 TAP Select Instruction

By means of TAP select instruction a user can access TAP select register and by controlling its only bit SDMA Bypass, control whether SDMA TAP is bypassed or not.

Table 56-6. TAP Select Register (TSR)

	TAP Select Register
	BIT 0
	Connect SDMA
TYPE	rw
RESET	0
Note:	

Table 56-7. TAP Select Register Description

Field	Description
0 SDMA Bypass	Connect SDMA Control whether SDMA TAP is bypassed or not: • 0 - SDMA TAP is bypassed by the alternate TAP inside SJC (emulating 4-bit IR and 1-bit bypass path). • 1 - SDMA TAP is connected to the TDI-TDO chain. **NOTE:** Additional cycle with TMS '0' should be inserted, after writing to this register, to allow the SDMA tap be sync before SDMA get into / out of bypass.

56.5.9 EXTEST_PULSE instruction

The EXTEST_PULSE instruction implements test behaviors for AC pins and simultaneously behaves identically to IEEE Std 1149.1 EXTEST for DC pins.

56.5.10 EXTEST_TRAIN instruction

The EXTEST_TRAIN instruction implements test behaviors for AC pins and simultaneously behaves identically to IEEE Std 1149.1 EXTEST for DC pins.

56.6 Security

JTAG manipulation is one of the known hackers' ways of executing unauthorized program code, getting control over the OS and run code in privileged modes.

The SJC provides a debug access to several H/W blocks including the Arm processor and the system bus. This allows for program control and manipulation as well as visibility into system peripherals and memory. The PTM interface allow bus transactions to be traced. Together these tools provide the hacker all the access needed to completely comprise the system. Means must be provided to block any malicious JTAG access.

The SJC provides a way of regulating the JTAG access.

The following are the different JTAG security modes:

- **Mode #1: No Debug**-Maximum Security. All security sensitive JTAG features are permanently blocked.
- **Mode #2: Secure JTAG**-High security. JTAG use is regulated by secret key based authentication mechanism.
- **Mode #3: JTAG Enabled**-Low security. JTAG always enabled.

The JTAG security modes are configured using eFUSEs which can burned after packaging by applying electrical signals. The fuse burning is irreversible process, once a fuse is burned (e-fuse or laser fuse) it is impossible to change the fuse back to the un-burned state.

56.6.1 JTAG Security Modes

JTAG can be in one of JTAG security modes which is selected by setting the SJC eFuse configuration. The physical location of the fuses is not in the SJC.

56.6.1.1 Mode 1: No Debug - Maximum Security

No Debug JTAG security mode provides the highest security level.

In this mode, all JTAG features are disabled except for:

- ScanBoundary Scan
- MBIST, all modes except for debug modes which enable controlled memory contents output
- PLL BIST

- BIST monitor mode, allowing routing to external pins BIST pass/fail/invoke information
- PLL bypass- Bypass Arm or/and USB PLL.
- Visibility of the following status bits: power mode - normal, standby, stop, shutdown, and so on

These features do not reduce the security level of the product, and they allows to perform important tests and board connectivity checks.

56.6.1.2 Mode 2: Secure JTAG - High Security

The Secure JTAG mode limits the JTAG access by using challenge/response based authentication mechanism. Any access to JTAG port is being checked. Only authorized debug devices (that is, devices having the right response) can access the JTAG, unauthorized JTAG access attempts are denied.

The intent of this mode is to allow return field testing. When a secured JTAG device is being returned for debugging, this mode allows authorized re-activation of the JTAG.

56.6.1.2.1 Challenge/Response Mechanism in System JTAG Mode

When SJC is in Sysytem JTAG mode the authentication process is as follows:

1. Shift Output Challenge instruction to IR.
2. Passing through Capture-DR state of the SJC and by performing Shift-DR operations Challenge code can be accessed from TDO.
3. Shift Enter Response instruction to IR. By performing Shift-DR, operations enter Response code value through TDI. As Update-DR state is entered, Response code is compared with the correct one.

In Fixed challenge-response pair mode, each part has its individual challenge - response pair which is determined at manufacturing time, and does not change later on. The SJC compares the user's response to the expected response.

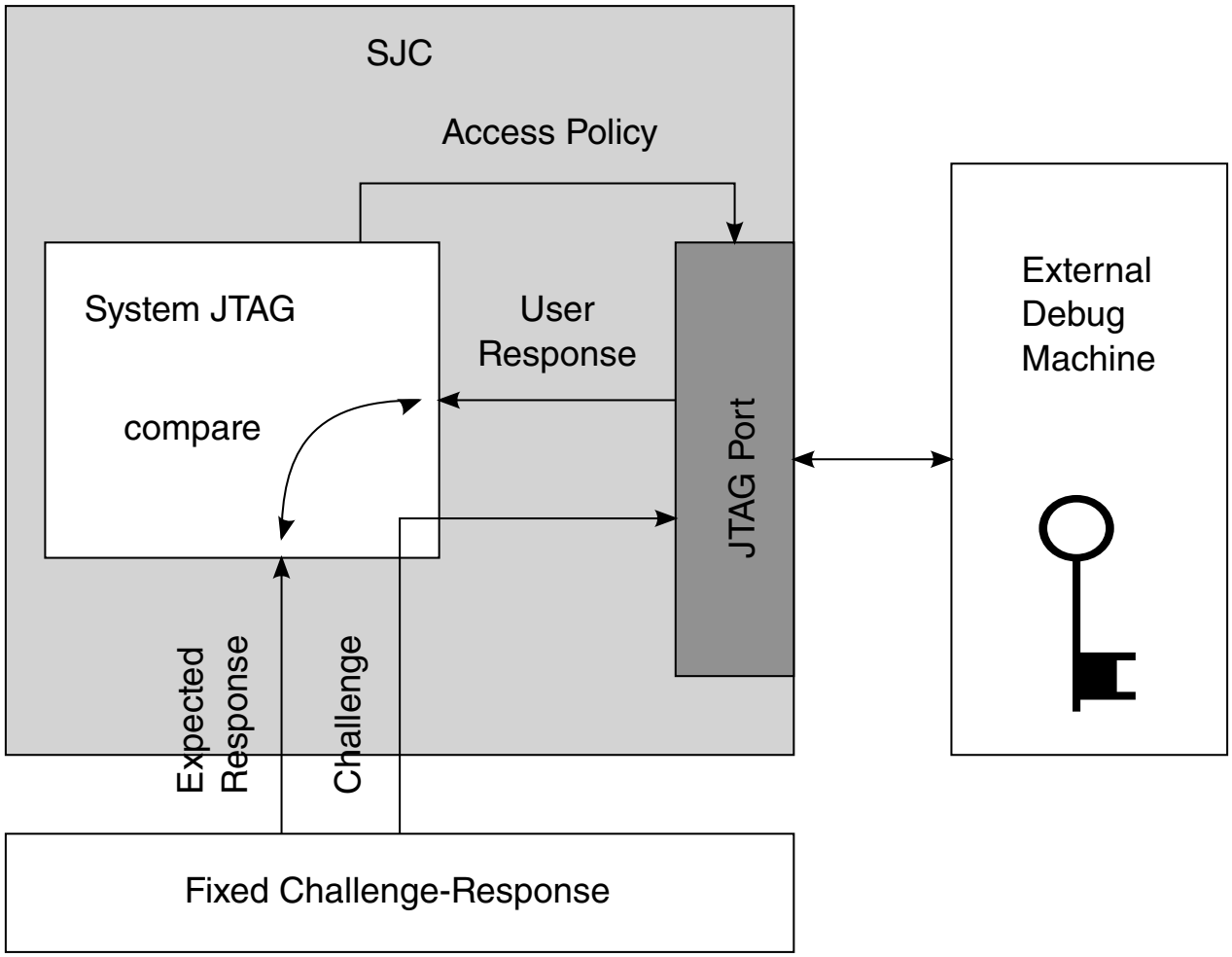


Figure 56-12. Mode #2 - Secure JTAG with Fixed Challenge-response Pair

56.6.1.3 Mode 3: JTAG Enabled - Low Security

In the JTAG Enabled JTAG security mode, all JTAG features are enabled.

56.6.2 Software Enabled JTAG

To increase the flexibility of the SJC, an option to enable the JTAG via software is added and is available only in Secure JTAG mode. By writing '1' to HAB_JDE (HAB JTAG DEBUG ENABLE) bit in the eFuse controller module, the JTAG is opened, regardless of its security mode. It is the responsibility of software to assert or negate this bit.

Additionally, a corresponding lock bit is available (in the eFuse control module) to ensure that only trusted software is able to set the JDE bit. When the LOCK bit is set, no future change of JDE is possible, until the next POR (power-on-reset) cycle.

The platform initialization software should set the LOCK bit for JDE bit before transferring control to the application code.

The S/W JTAG enable allows JTAG enabling without activating the challenge-Response mechanism (which requires JTAG access tool enhancement or special H/W). The JTAG S/W enable does not allow debug in case of boot or memory fault as it requires reset before entering debug.

This feature can be permanently blocked by burning the dedicated eFuse.

NOTE

The S/W enabled JTAG feature reduces the overall security level of the system as it relies on S/W protections. If this feature is not required, it is strongly recommended to burn the JTAG_HEO eFuse which disables this feature.

56.6.3 Kill Trace

The kill trace signal disables any output of the PTM block. The PTM can be accessed either via JTAG port and/or by direct software code. Blocking the JTAG port also yields assertion of the kill trace signal. This resulted in blocking of trace port. The intention of this action is to block any attempt to break into the system via software manipulation of the debug modules. The kill trace, when active, prevents trace output even in case where it can be activated via chip pin.

The kill trace feature needs to be activated by burning a dedicated eFuse. If the fuse is left intact, kill trace is never activated as seen in Figure 56-13.

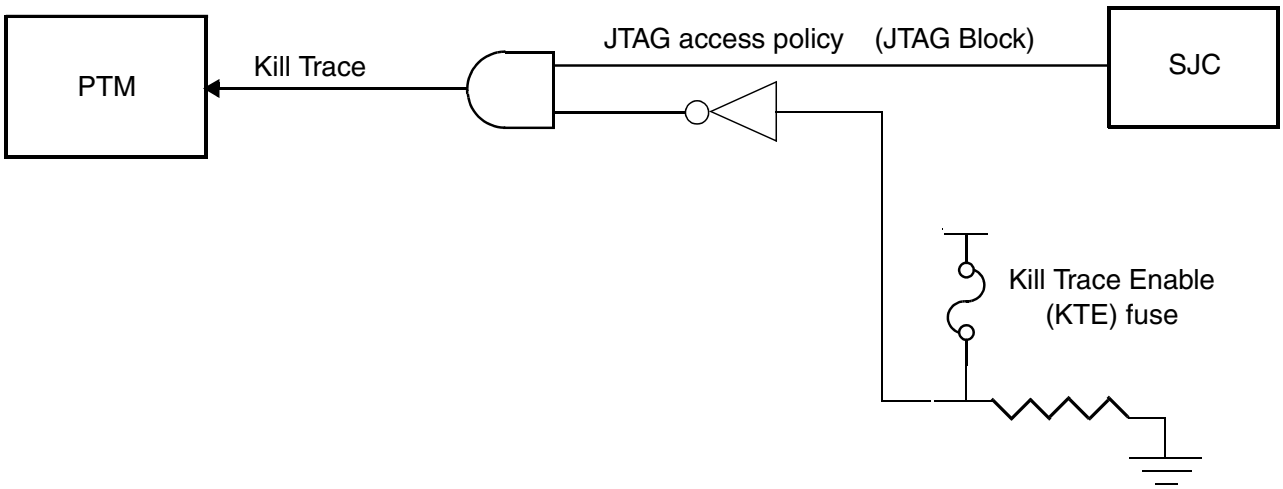


Figure 56-13. Kill Trace eFuse

The kill trace is asserted when "kill trace enable" fuse is burned and "ipt_secur_block" signal in SJC is asserted, which happens when at least one of the following is true:

- Mode #2 (Secure JTAG) and no code has been entered
- Mode #2 (Secure JTAG) with incorrect response entered
- Mode #1 (No debug)
- TRST_B signal is active
- POR has not ever been asserted

56.6.4 SJC Disable Fuse

In addition to the different JTAG security modes that are implemented internally in the System JTAG Controller (SJC), there is an option to disable the SJC functionality by eFuse configuration. This creates additional JTAG mode that is, JTAG Disabled with highest level of JTAG protection. In this mode all JTAG features are disabled.

Specifically, the following debug features are disabled in addition to the features that were already disabled in No Debug JTAG mode:

- Memory BIST
- Boundary scan register (SJC_BSR)
- Non-Secure JTAG control registers (PLL configuration, Deterministic Reset, PLL bypass)
- Non-Secure JTAG status registers (Core status)
- Chip Identification Code (IDCODE)

56.7 Functional Description

This section provides a complete functional description of the block.

56.7.1 Static Core Debug

The SJC JTAG TAP controller is fully compatible with the IEEE 1149.1a-2001 Standard Test Access Port and Boundary Scan Architecture specifications.

The Arm MPCore platform debug system (named CoreSight) including the real-time Program Trace Macrocell (PTM), are controlled via the Debug Access Port (DAP) module. Refer to Arm MPCore and PTM Technical reference manuals for more details.

The SDMA has a TAP controller to manage its own OnCE, see SDMA OnCE specifications for more details.

The OnCE and ICE provide a mean of interacting with the cores and their peripherals non-intrusively so that a user may examine registers, memories to facilitate hardware and software development. Refer to TAP Selection Block (TSB), for more information.

56.7.2 Reset Mechanism

The following figure shows the SJC reset logic

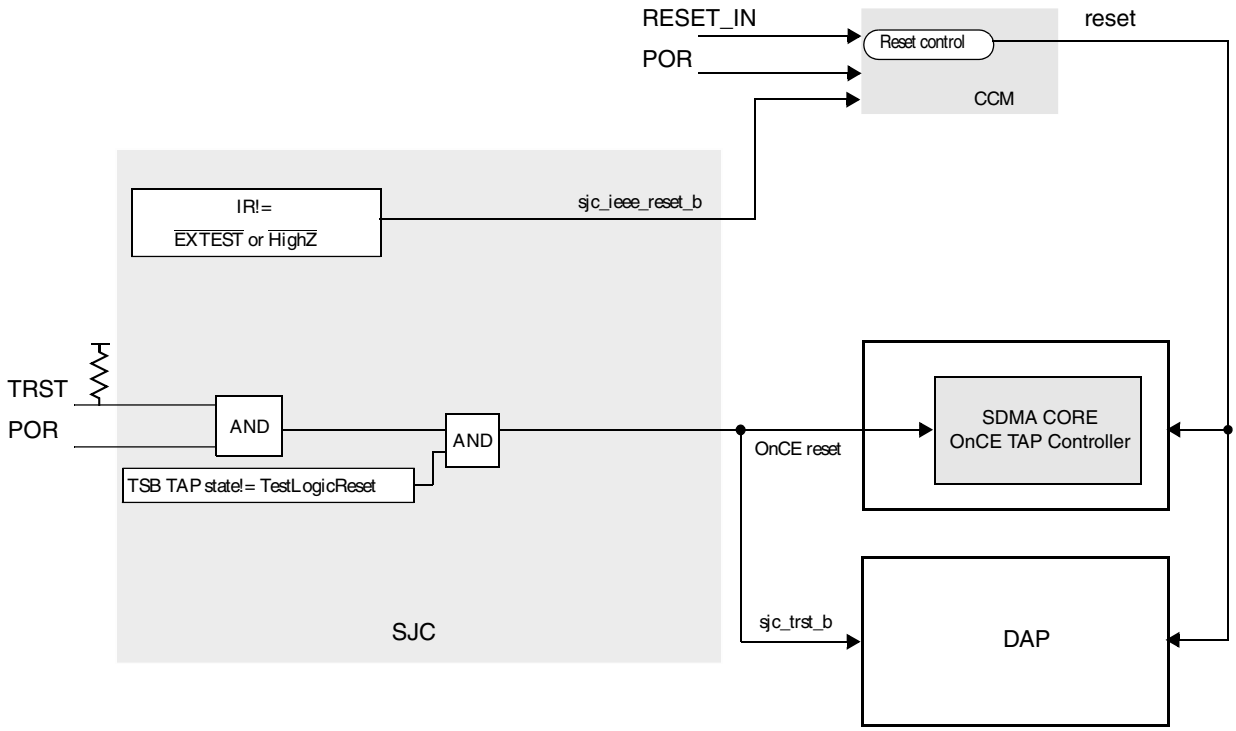


Figure 56-14. SJC Reset Logic

NOTE

- Asserting TRSTB in any scan mode resets the TCR loosing the testmode configuration and selects default TAP.
- SJC generates an IEEE reset signal to the CCM when in one of the IEEE modes HIGHZ or EXTEST. This signal generates a system reset to the cores until exit from one of these modes.
- The TSB generates Once/ICE reset (either TRSTB if implemented or other) when its TAP state reaches Test-Logic-Reset (meaning that TAP accessed is also reaching Test-Logic-Reset).

56.8 Initialization/Application Information

The control afforded by the output enable signals using the boundary scan register and the EXTEST instruction requires a compatible circuit-board test environment to avoid device-destructive configurations. The user must avoid situations in which the SJC output drivers are enabled into actively driven networks.

There are two constraints related to the JTAG interface:

- Ensure that the JTAG test logic is kept transparent to the system logic by forcing TAP into the Test-Logic-Reset controller state. During power-up, SJC's internal TRSTB is asserted as IC's POR_B is asserted which forces the TAP controller into this state. After that, if TMS either remains unconnected or is connected to VCC, then the TAP controller cannot leave the Test-Logic-Reset state, regardless of the state of TCK.
- DE_B is an IO pin with pullup and care must be taken of the direction when driving this signal.

56.9 SJC Memory Map/Register Definition

In addition to the standard accessible JTAG registers (per IEEE1149.1 standard) listed in SJC Instruction Register (SJIR) , the chip contains the following registers accessed using the ExtraDebug mechanism, controlled via "ENABLE_ExtraDebug" IR instruction.

NOTE

SJC registers are only accessible by JTAG interface. They are not memory mapped to processor address space, so the absolute addresses provided by default in the SJC memory map are not valid.

This section assumes the JTAG controller is accessed in standalone mode or daisy chained (defined by TAP Selection Block) using the appropriate TSB configuration.

See "System Debug" chapter for more details about the general purpose register descriptions that are unique to this chip.

SJC memory map

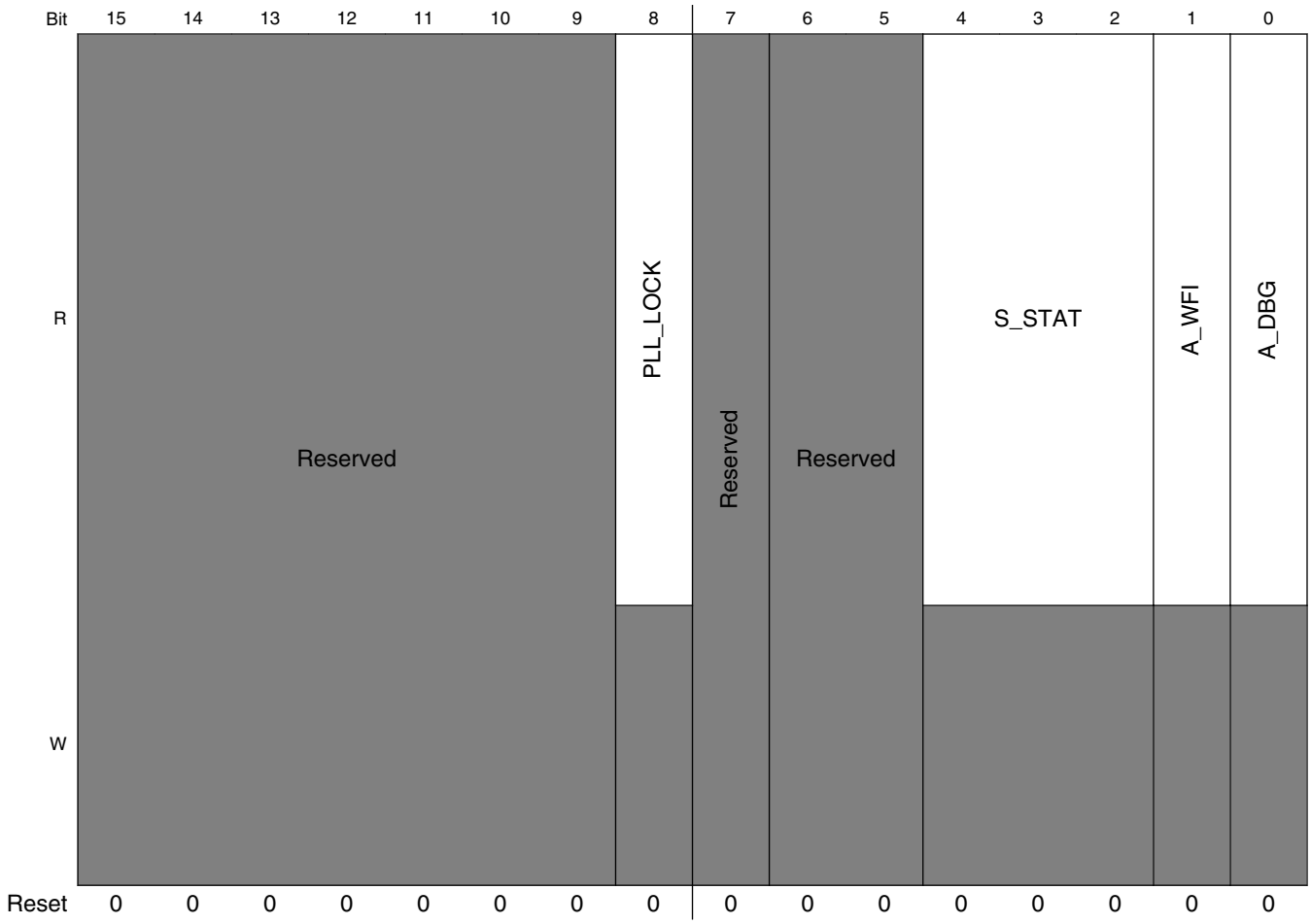
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	General Purpose Unsecured Status Register 1 (SJC_GPUSR1)	32	R	0000_0000h	56.9.1/4925
1	General Purpose Unsecured Status Register 2 (SJC_GPUSR2)	32	R	0000_0000h	56.9.2/4927
2	General Purpose Unsecured Status Register 3 (SJC_GPUSR3)	32	R	0000_0000h	56.9.3/4927
3	General Purpose Secured Status Register (SJC_GPSSR)	32	R	0000_0000h	56.9.4/4928
4	Debug Control Register (SJC_DCR)	32	R/W	0000_0000h	56.9.5/4929
5	Security Status Register (SJC_SSR)	32	R	See section	56.9.6/4931
7	General Purpose Clocks Control Register (SJC_GPCCR)	32	R/W	0000_0000h	56.9.7/4934

56.9.1 General Purpose Unsecured Status Register 1 (SJC_GPUSR1)

The General Purpose Unsecured Status Register 1 is a read only register used to check the status of the different Cores and of the PLL. The rest of its bits are for general purpose use.

Address: 0h base + 0h offset = 0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

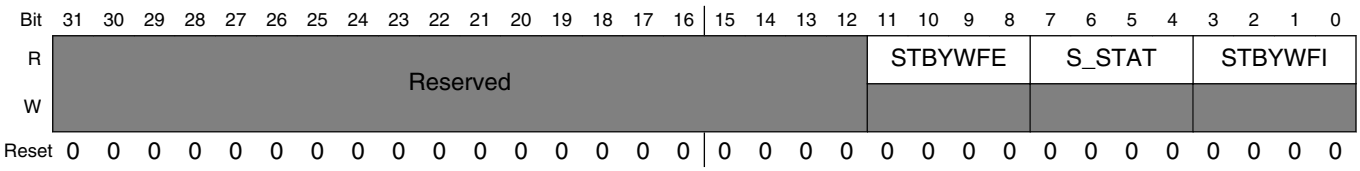


SJC_GPUSR1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8 PLL_LOCK	PLL_LOCK A Combined PLL-Lock flag indicator, for all the PLL's.
7 -	This field is reserved. Reserved
6–5 -	This field is reserved. Reserved.
4–2 S_STAT	3 LSBits of SDMA core statusH.
1 A_WFI	Arm core wait-for interrupt bit Bit 1 is the Arm core standbywfi (stand by wait-for interrupt). When this bit is HIGH, Arm core is in wait for interrupt mode.
0 A_DBG	Arm core debug status bit Bit 0 is the Arm core DBGACK (debug acknowledge) DBGACK can be overwritten in the Arm core DCR to force a particular DBGACK value. Consequently interpretation of the DBGACK value is highly dependent on the debug sequence. When this bit is HIGH, Arm core is in debug.

56.9.2 General Purpose Unsecured Status Register 2 (SJC_GPUSR2)

Address: 0h base + 1h offset = 1h

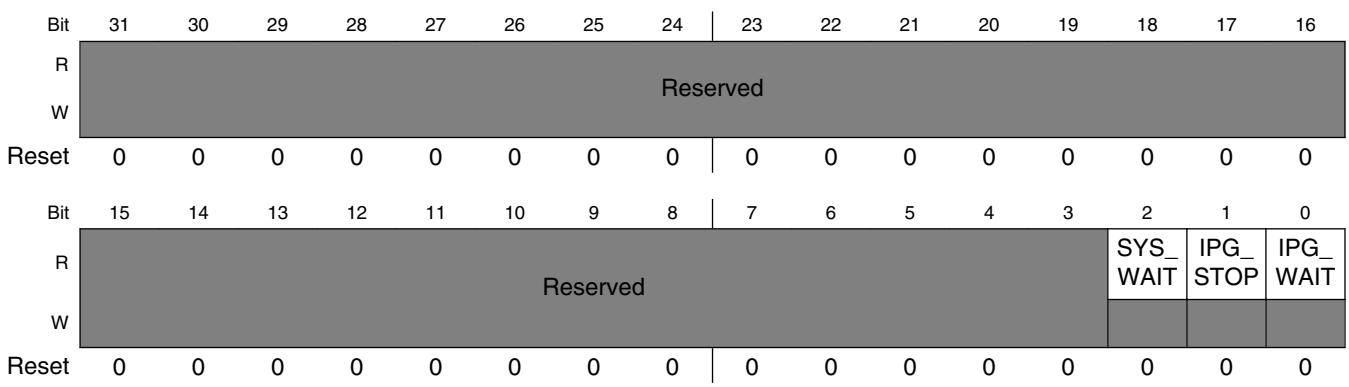


SJC_GPUSR2 field descriptions

Field	Description
31–12 -	This field is reserved. Reserved
11–8 STBYWFE	STBYWFE[3:0] Reflecting the "Standby Wait For Event" signals of all cores.
7–4 S_STAT	S_STAT[3:0] SDMA debug status bits: debug_core_state[3:0]
STBYWFI	STBYWFI[3:0] These bits provide status of "Standby Wait-For-Interrupt" state of all Arm cores.

56.9.3 General Purpose Unsecured Status Register 3 (SJC_GPUSR3)

Address: 0h base + 2h offset = 2h



SJC_GPUSR3 field descriptions

Field	Description
31–3 -	This field is reserved. Reserved

Table continues on the next page...

SJC_GPUSR3 field descriptions (continued)

Field	Description
2 SYS_WAIT	System In wait Indication on System in wait mode (from CCM).
1 IPG_STOP	IPG_STOP CCM's "ipg_stop" signal indication
0 IPG_WAIT	IPG_WAIT CCM's "ipg_wait" signal indication

56.9.4 General Purpose Secured Status Register (SJC_GPSSR)

The General Purpose Secured Status Register is a read-only register used to check the status of the different critical information in the SoC. This register cannot be accessed in secure modes.

Address: 0h base + 3h offset = 3h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPSSR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

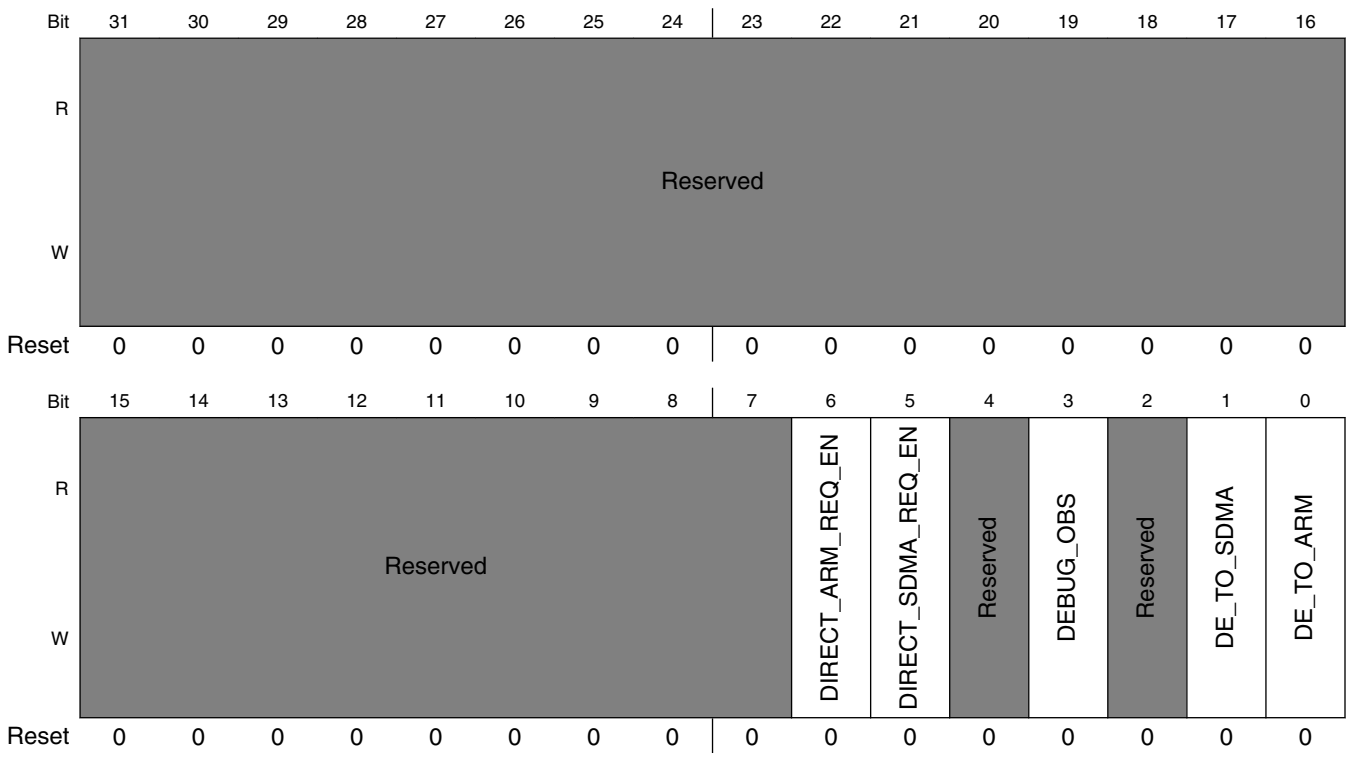
SJC_GPSSR field descriptions

Field	Description
GPSSR	General Purpose Secured Status Register Register is used for testing and debug.

56.9.5 Debug Control Register (SJC_DCR)

This register is used to control propagation of debug request from DE_B pad to the cores and debug signals from internal logic to the DE_B pad.

Address: 0h base + 4h offset = 4h



SJC_DCR field descriptions

Field	Description
31–7 -	This field is reserved. Reserved
6 DIRECT_ARM_REQ_EN	Pass Debug Enable event from DE_B pin to Arm platform debug request signal(s). This bit controls the propagation of debug request DE_B to the Arm platform. 0 Disable propagation of system debug to (DE_B pin) to Arm platform. 1 Enable propagation of system debug to (DE_B pin) to Arm platform.
5 DIRECT_SDMA_REQ_EN	Debug enable of the sdma debug request This bit controls the propagation of debug request DE_B to the sdma. 0 Disable propagation of system debug to (DE_B pin) to sdma. 1 Enable propagation of system debug to (DE_B pin) to sdma.
4 -	This field is reserved. Reserved

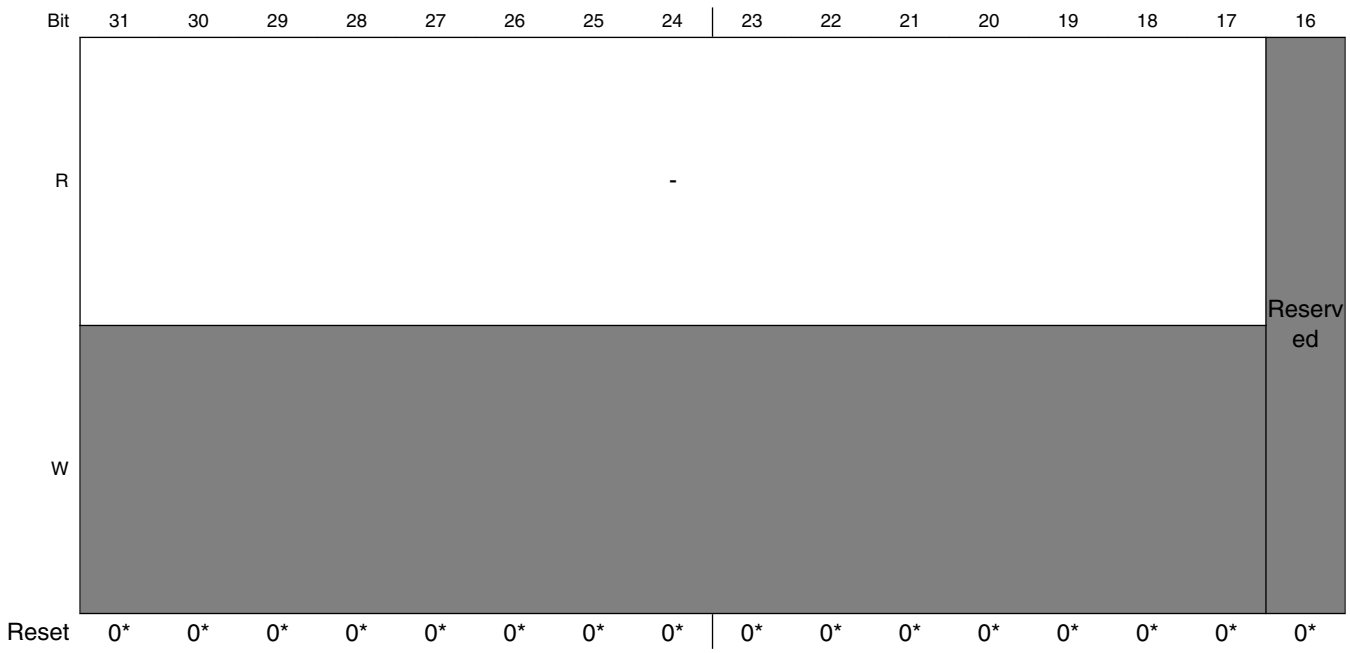
Table continues on the next page...

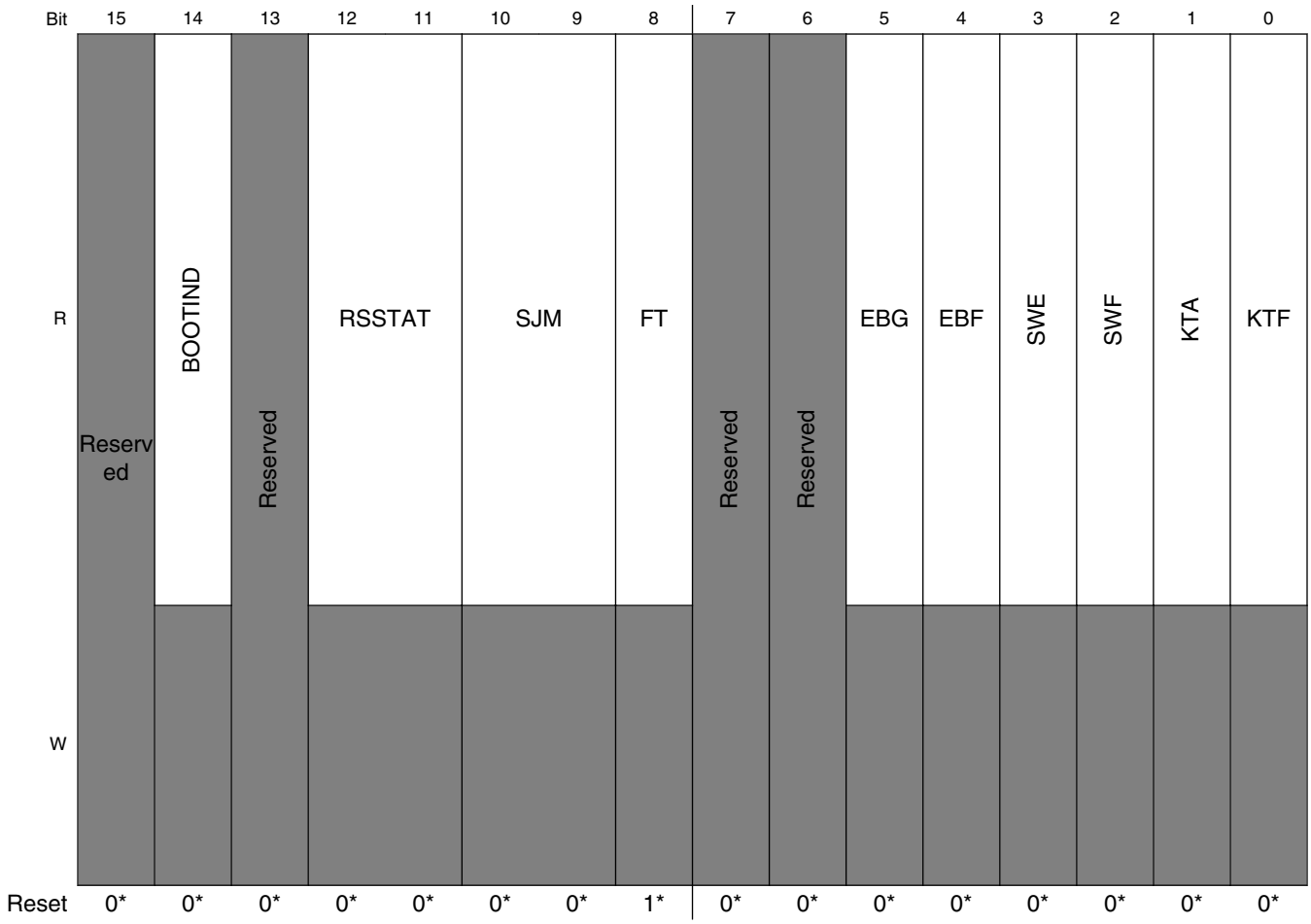
SJC_DCR field descriptions (continued)

Field	Description
3 DEBUG_OBS	Debug observability This bit controls the propagation of the "system debug" input to SJC (driven by the ECT logic), to the DE_B pad. (This logic can be used to pass debug acknowledge event from ECT out to the PAD, for example). The SJC's "system_debug" input is tied to logic HIGH value, therefore, set of "debug_obs" bit, will result in unconditional assertion of DE_B pad. 0 Disable propagation of system debug to DE_B pin 1 Unconditional assertion of pad DE_B
2 -	This field is reserved. Reserved
1 DE_TO_SDMA	SDMA debug request input propagation This bit controls the propagation of debug request to SDMA, when the JTAG state machine is put in "ENTER_DEBUG" IR instruction.. 0 Disable propagation of debug request to SDMA 1 Enable propagation of debug request to SDMA
0 DE_TO_ARM	Arm platform debug request input propagation This bit controls the propagation of debug request to Arm platform ("dbgreq"), when the JTAG state machine is put in "ENTER_DEBUG" IR instruction. 0 Disable propagation of debug request to Arm platform 1 Enable propagation of debug request to Arm platform

56.9.6 Security Status Register (SJC_SSR)

Address: 0h base + 5h offset = 5h





* Notes:
- The SJM reset value, reflects the JTAG security state, as defined by status of JTAG_SMODE[1:0] fuses. See the SJM bitfield description for details on valid values.

SJC_SSR field descriptions

Field	Description
31–17 -	Reserved.
16–15 -	This field is reserved. Reserved
14 BOOTIND	Boot Indication Inverted Internal Boot indication, i.e inverse of SRC: "src_int_boot" signal
13 -	This field is reserved. Reserved
12–11 RSSTAT	Response status Response status bits 00 Response wasn't entered 01 Response was entered but not verified

Table continues on the next page...

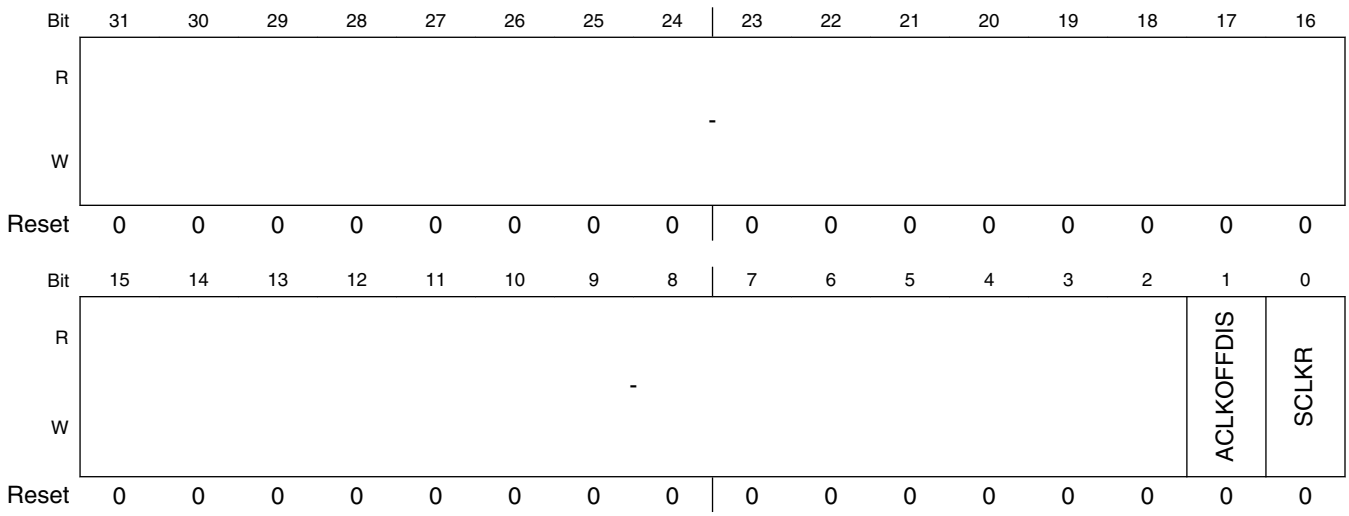
SJC_SSR field descriptions (continued)

Field	Description
	10 Response was entered and is incorrect 11 Response is correct
10–9 SJM	SJC Secure mode Secure JTAG mode, as set by external fuses. 00 No debug (#1) 01 Secure JTAG (#2) 10 Reserved 11 JTAG enabled (#3)
8 FT	Fuse type Fuse type bit - e-fuse or laser fuse 0 E-fuse technology 1 Laser fuse technology
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5 EBG	External boot granted External boot enabled, requested and granted 1 granted 0 not granted
4 EBF	External Boot fuse Status of the external boot disable fuse 0 (intact) - external boot is allowed 1 (burned) - external boot is disabled
3 SWE	SW enable SW JTAG enable status 1 enabled 0 disabled
2 SWF	Software JTAG enable fuse Status of the no SW disable JTAG fuse 0 (intact) - SW enable possible 1 (intact) - no SW enable possible
1 KTA	Kill Trace is active 1 active 0 not active
0 KTF	Kill Trace Enable fuse value 0 (intact) - kill trace is never active 1 (burned) - kill trace functionality enabled

56.9.7 General Purpose Clocks Control Register (SJC_GPCCR)

This register is used to configure clock related modes in SOC, see System Configuration chapter for more information. Those bits are directly connected to JTAG outputs. Bit 0 of GPCCR controls SDMA clocks invocation. When out of reset, the SDMA is in sleep mode with no SDMA clock running. Unlike events, debug requests does not wake SDMA if it is in sleep mode. The debug request is recognized by the SDMA only when it exits sleep mode upon reception of an event. To be able to enter debug mode even if no event is triggered, the SDMA clock on bit needs to be set prior to sending the debug request (clear at reset).

Address: 0h base + 7h offset = 7h



SJC_GPCCR field descriptions

Field	Description
31–2 -	Reserved
1 ACLKOFFDIS	Disable/prevent Arm platform clock/power shutdown
0 SCLKR	SDMA Clock ON Register - This bit forces the clock on of the SDMA

Chapter 57
Secure Non-Volatile Storage (SNVS)

57.1 SNVS overview

The low-power (battery-backed) section incorporates a secure real time counter, a monotonic counter, and a general-purpose register. The LP portion of the SNVS is powered by a battery that maintains the state of the SNVS_LP registers when the chip is powered off.

NOTE

For the security features of the SNVS, see the *Security Reference Manual for i.MX 6Dual, 6Quad, 6Solo, and 6DualLite Families of Application Processors (IMX6DQ6SDLSRM).*

57.1.1 SNVS features

The following table summarizes the features:

Table 57-1. SNVS feature summary

Feature	What it does
(nonsecure) Real Time Counter (HP-RTC)	• The counter is driven by a dedicated clock, which is off when the system power is down • Programmable time alarm interrupt • Periodic interrupt can be generated with different frequencies
Monotonic Counter (LP-MC)	• The Monotonic Counter state is nonvolatile. • The counter can only increment. • The counter is a non-rollover counter • The counter value is invalidated in case of security violation.
General-Purpose Register	• The general-purpose register state is nonvolatile.
Register access protection	• Privileged software access policy • Registers can be programmed only when the system security monitor is in functional state. • Some registers/values can only be programmable once per boot cycle.

57.1.2 Modes of operation

The SNVS operates in either the system power-down or the system power-up mode of operation.

During system power-down, the SNVS_HP is powered down. The SNVS_LP is powered from the backup power supply and is electrically isolated from the rest of the chip. In this mode, the SNVS_LP retains the state of its registers.

During the system power-up, the SNVS_HP and SNVS_LP are both powered up and all SNVS functions are operational.

57.2 External signals

The table in this section describes the external signals of the SNVS.

Table 57-2. SNVS external signals

Signal	Description	Pad	Mode	Direction
SNVS_PMIC_ON_REQ	Wake-up signal	PMIC_ON_REQ	Not multiplexed	O
SNVS_TAMPER	Tamper signal	TAMPER	Not multiplexed	I

57.3 Clocks

The table in this section describes the clock sources for the SNVS.

See Clock Controller Module (CCM) for the clock setting, configuration, and gating information.

Table 57-3. SNVS clocks

Clock name	Clock root	Description
hp_ipg_clk	ipg_clk_root	HP peripheral clock
hp_ipg_clk_s	ipg_clk_root	HP peripheral access clock used for clocking the registers on the bus R/W accesses
ipg_hp_rtc_clk	ckil_sync_clk_root	The HP RTC clock advances the RTC, doesn't have to be synchronous with any module clock.
lp_ipg_clk	ipg_clk_root	LP peripheral clock
lp_ipg_clk_s	ipg_clk_root	LP peripheral access clock used for the register R/W accesses

57.4 SNVS structure

The SNVS block is divided into two major submodules based on power supply: the high power domain (SNVS_HP) and the low power domain (SNVS_LP). They are powered as follows:

- SNVS_LP - dedicated always-powered-on domain
- SNVS_HP - system (chip) power domain

The following figure illustrates the low power and chip power domains of SNVS.

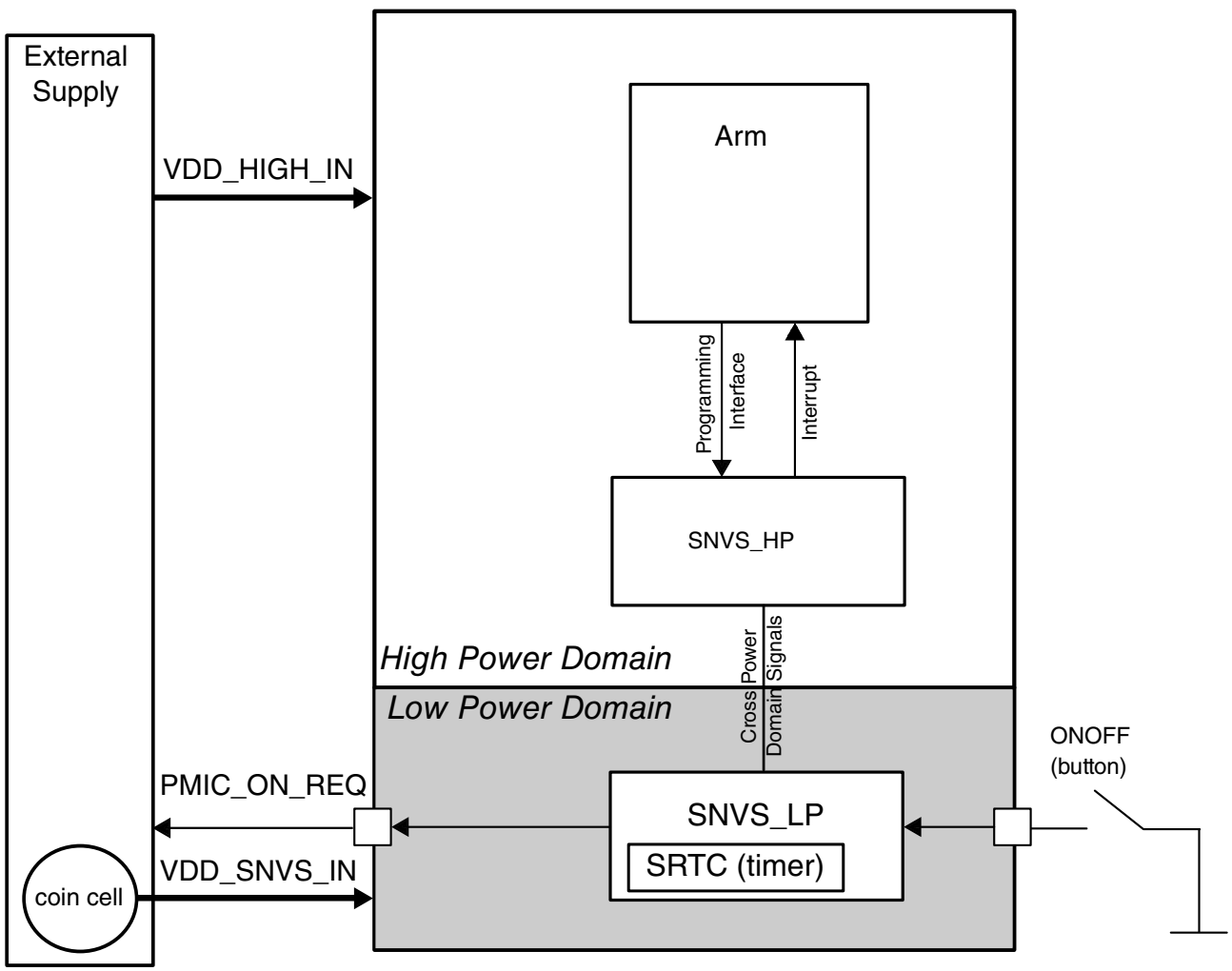


Figure 57-1. SNVS Power Domains

The SNVS_HP section implements all features that enable system communication and provisioning of the SNVS_LP section.

The SNVS_LP section provides hardware that enables secure storage and protection of sensitive data.

57.4.1 SNVS_HP (high-power domain)

The SNVS_HP is partitioned into these functional units:

- IP bus interface
- SNVS_LP interface
- Real-time counter with alarm
- Control and status registers

The SNVS_HP is in the chip's power-supply domain and thus receives the power along with the rest of the chip. The SNVS_HP provides an interface between the SNVS_LP and the rest of the system; there is no way to access the SNVS_LP registers except through the SNVS_HP. For access to the SNVS_LP registers, the SNVS_HP must be powered up. It uses a register access permission policy to determine whether the access to the particular registers is permitted.

57.4.2 Non-secure real-time counter

The SNVS _HP has an autonomous non-secure real-time counter. The counter is not active and is reset when the system is powered down. The HP RTC can be used by any application; it has no privileged software access restrictions.The counter can be synchronized with the SNVS_LP SRTC by writing to a specific bit in the SNVS_HP control register.

57.4.2.1 Calibrating the time counter

The RTC accuracy may suffer from a drift in the clock, which is used to increment the RTC register. To compensate for this drift, a clock calibration mechanism can adjust the RTC value. It is up to the system processor to decide whether the calibration is required or not. If the RTC correction is required, enable the mechanism and set the calibration value in the control register. The calibration value is a 5-bit value including the sign bit, which is implemented in the 2's complement.

If the calibration mechanism is enabled, the calibration value is added or subtracted from the RTC on a periodic basis, once per 32768 cycles of the RTC clock.

This table shows the available correction range:

Table 57-4. Time counter calibration settings

Calibration value setting	Correction in counts per 32768 cycles of the counter clock
01111	+15
:	:
00010	+2
00001	+1
00000	0
11111	-1
11110	-2
:	:
10001	-15
10000	-16

57.4.2.2 Time counter alarm

The SNVS_HP non-secure RTC has its own time-alarm register. Any application can update this register. The SNVS_HP time alarm can generate interrupts to alert the host processor and can wake up the host processor from one of its low-power modes. Note that this alarm can't wake up the entire system if it is powered off because this alarm would also be powered off.

57.4.2.3 Periodic interrupt

The SNVS_HP non-secure RTC incorporates a periodic interrupt. The periodic interrupt is generated when a zero-to-one or one-to-zero transition occurs on the selected bit of the RTC. The periodic interrupt source is chosen from the 16 bits of the HP RTC according to the PI_FREQ field setting in the HP control register. This bit selection also defines the frequency of the periodic interrupt.

This figure shows the SNVS_HP RTC and its interrupts:

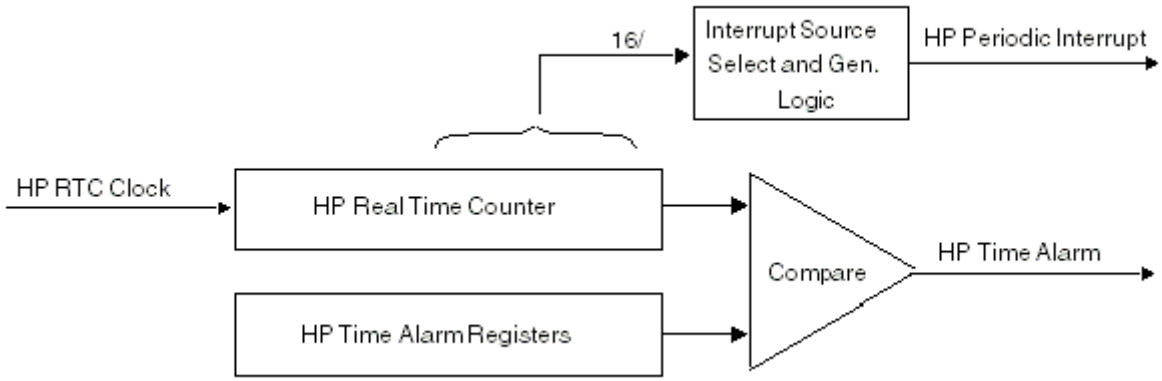


Figure 57-2. SNVS_HP RTC, alarm, and interrupts

57.5 SNVS_LP (low-power domain)

The SNVS_LP has these functional units:

- Non-rollover monotonic counter
- General-purpose register
- Control and status registers

The SNVS_LP is a data storage subsystem. Its purpose is to store and protect system data, regardless of the main system power state.

The SNVS_LP is in the always-powered-up domain, which is a separate power domain with its own power supply.

57.5.1 Behavior during system power down

When the chip power-supply domain loses power, the The SNVS_LP continues to operate normally and ignores all inputs from the SNVS_HP.

57.5.2 Monotonic Counter (MC)

This figure shows the MC and its rollover security violation:

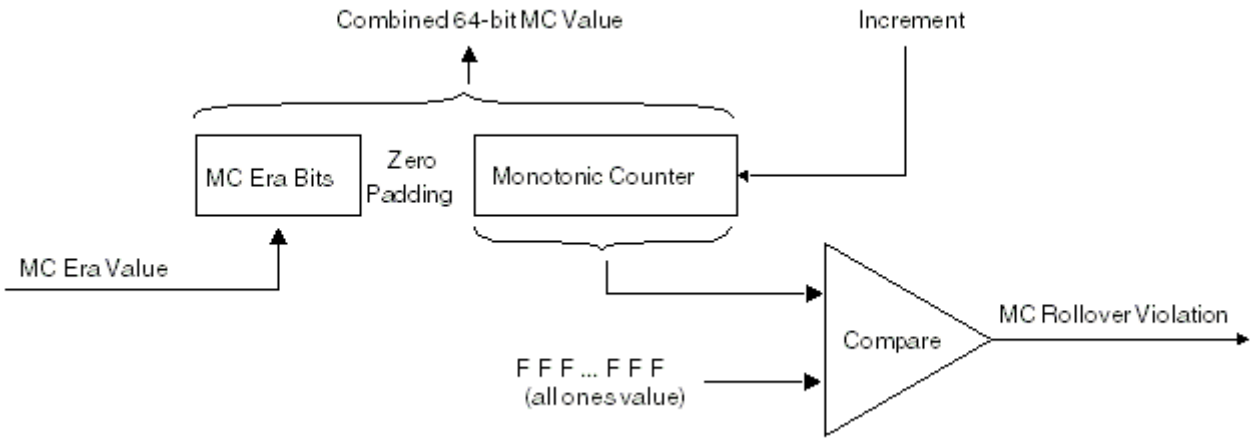


Figure 57-3. SNVS_LP monotonic counter

Some security applications require a Monotonic Counter (MC) that can't be exhausted or returned to any previous value during the product's lifetime. Because the MC should never repeat a number, it can't be reset or cycled back to its starting count. If it reaches its maximum value it does not rollover. Instead, a monotonic counter rollover indication is generated to the SNVS_LP tamper monitor. This generates an interrupt to the host processor.

The MON_COUNTER fields of the MC register are implemented in flip flops within the LP section. If LP power is lost, the MON_COUNTER value will be lost. To preserve monotonicity in this event, the most significant bits of MC (the MC_ERA_BITS field) are derived from fuses. This means that the MC_ERA_BITS value is preserved across LP section power failures. The next time that the chip powers up following an LP section power failure, the chip's boot firmware will note that the monotonic counter value is invalid and will blow another of the fuses that drive the MC_ERA_bits field. This will result in a larger value in the MC_ERA MC field and since this field forms the most-significant bits of MC, this guarantees that the new value of MC is greater than any of its past values.

57.6 SNVS reset and system powerup

This table describes the reset actions for the SNVS.

Table 57-5. Reset summary

Reset	Source	Characteristics	Internally resets
HP hard	ipg_hard_async_reset_b	active-low, asynchronous	All SNVS_HPSNVS_LP registers and flops.

Table continues on the next page...

Table 57-5. Reset summary (continued)

Reset	Source	Characteristics	Internally resets
LP Power On Reset (POR)	lp_por_b	active-low, asynchronous	All SNVS_LP registers and flops.
LP software reset	software	active-high, synchronous, one cycle	All SNVS_LP registers and flops. The LP software reset can be asserted if not disabled.

57.6.1 PMIC interface

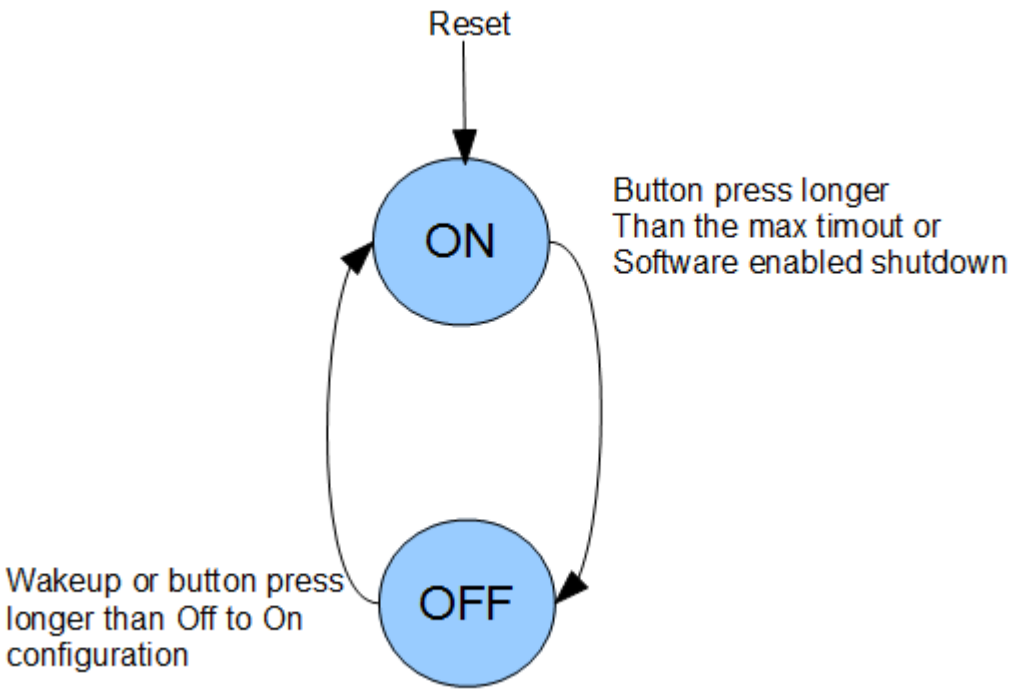
The on/off logic inside the SNVS_LP allows for connecting directly to a PMIC or other voltage-regulator device. The logic takes the button input signal and then outputs the PMIC "ON" request and the "Power Off" interrupt. The PMIC logic also supports the SNVS_LP tamper logic which allows to wake the system up when a tamper event has happened while in the OFF state. The logic has two different modes of operation (the dumb and smart modes).

Dumb PMIC mode:

The dumb PMIC mode uses the PMIC "ON" request to issue a level signal for the ON and OFF states. The dumb PMIC mode has many different configuration options which include debounce, off-to-on time, and maximum timeout.

- Debounce—the debounce is set to 0 ms. The debounce is used to generate the set_pwr_off_irq interrupt. While it is in the ON state and the button is pressed longer than the debounce time, the set_pwr_off_irq is generated.
- Off-to-on time—the off-to-on is set to 0 ms. This configuration supports the time it takes to request the power-on after the configured button press time is reached. When the button is pressed longer than the configuration time, the state machine transitions from the OFF to the ON state.
- Max timeout—the maximum timeout configuration is set to 5 s. This configuration supports the time it takes to request the power down after the button is pressed for the defined time.

The dumb PMIC mode uses a 2-state state machine, as shown in the following figure. The output of the pmic_en_b is generated by the state of the state machine.



Smart PMIC mode:

The smart PMIC mode is not supported on this chip.

57.7 SNVS interrupts and alarms

SNVS provides these interrupt and alarm lines:

- Functional interrupt (active-low)
- Real-time clock period interrupt
- Power off (button) interrupt

This table summarizes all SNVS interrupts and alarm sources.

Table 57-6. Interrupts and alarms summary

Interrupt	Source	Default configuration[1]	Configuration options
SNVS functional interrupt	RTC time alarm	Disable	Enable/Disable
	RTC periodic interrupt	Disable	Enable/Disable
SNVS power off (button) interrupt	BTN input signal	50 ms debounce	Debounce time

1. Default behavior refers to the setting after the LP/HP reset.

57.8 Programming guidelines

This section provides the initialization and application information for the SNVS module.

57.8.1 RTC control bits setting

All SNVS registers are programmed from the register bus. Therefore, any changes are synchronized with the IP clock. Several registers can also change synchronously with the RTC clock after they are programmed. To avoid IP clock and RTC clock synchronization issues, these values can only be programmed when the corresponding function is disabled. This table presents the list of these values with the control bit setting required for programming:

Table 57-7. RTC synchronized values list

Function	Value/register	Control bit setting
HP section		
HP real-time counter	HPRTCMR and HPRTCLR registers	RTC_EN = 0—HPRTCMR/HPRTCLR can be programmed RTC_EN = 1—HPRTCMR/HPRTCLR can't be programmed
HP time alarm	HPTAMR and HPTALR registers	HPTA_EN = 0—HPTAMR/HPTALR can be programmed HPTA_EN = 1—HPTAMR/HPTALR can't be programmed
HP time calibration value	HPCALB_VAL value	HPCALB_EN = 0—HPCALB_VAL can be programmed HPCALB_EN = 1—HPCALB_VAL can't be programmed

Perform these steps to program the synchronized values:

1. Check the enable bit value. If it is set, clear it.
2. Verify that the enable bit is cleared.

   There are two reasons to verify the enable bit's setting:

   - The enable bit clearing does not happen immediately. It takes three IPclock cycles and two RTC clock cycles to change the enable bit's value.
   - If the enable bit is locked for programming, it can't be cleared.
3. Program the desired value.
4. Set the enable bit. It takes three IP clock cycles and two RTC clock cycles for the bit to set.

NOTE

Incrementing the value programmed into the RTC registers by two compensates for the two RTC clock cycle delays that are required to enable the counter.

57.8.2 RTC value read

There are two scenarios when the software can read the corrupted values from the RTC (HPRTCMR and HPRTCLR) registers:

- The RTC counters are incremented by the slow 32-kHz clock, which is asynchronous to the system clock. The counter value is synchronized to the system clock before the software reads that. The synchronization register can capture the counter value in the middle of the counter update. In this case, it is not guaranteed that all bits are properly sampled by the synchronization register; the value read by the software can be wrong.
- The RTC value is longer than the single bus read transaction of 32 bits. Therefore, the software reads two registers with each holding a portion of the counter value. After reading one of these registers but before reading the second register, both registers can update their values. In this case, the value combined by the software is incorrect.

To avoid these issues, it is strongly recommended that the software performs two consecutive reads of the RTC value:

- If the two consecutive reads are similar, the value is correct.
- If the two consecutive reads are different, perform two more reads.

The worst-case scenario can require three sessions of two consecutive reads.

57.8.3 General initialization guidelines

Perform these steps to properly initialize the module:

1. Enable the interrupts in the SNVS control and configuration registers.
2. Program the SNVS general functions/configurations.
3. User-specific: Set the lock bits.

57.9 SNVS Memory Map/Register Definition

This section contains detailed register descriptions for the SNVS registers. Each description includes a standard register diagram and a register table. The register table provides detailed descriptions of the register bit and field functions in the bit order.

The SNVS registers consist of two types:

- Privileged read/write accessible
- Non-privileged read/write accessible

The privileged read/write accessible registers can only be accessed for read/write by the privileged software. Unauthorized write accesses are ignored, and unauthorized read accesses return zero. The non-privileged software can access privileged access registers when the non-privileged software access enable bit is set in the SNVS_HP command register.

- Non-secure
- Trusted
- Secure

The non-privileged read/write accessible registers are read/write accessible by any software.

The following table shows the SNVS memory map. The LP register values are set only on the LP POR and are unaffected by the system (HP) POR. The HP registers are set only on the system POR and are unaffected by the LP POR.

NOTE

For more information on the security-related bitfields, see the *Security Reference Manual for i.MX 6Dual, 6Quad, 6Solo, and 6DualLite Families of Application Processors* (IMX6DQ6SDLSRM).

SNVS memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_C000	SNVS_HP Lock register (SNVS_HPLR)	32	R/W	0000_0000h	57.9.1/4948
20C_C004	SNVS_HP Command register (SNVS_HPCOMR)	32	R/W	0000_0000h	57.9.2/4950
20C_C008	SNVS_HP Control register (SNVS_HPCR)	32	R/W	0000_0000h	57.9.3/4952

Table continues on the next page...

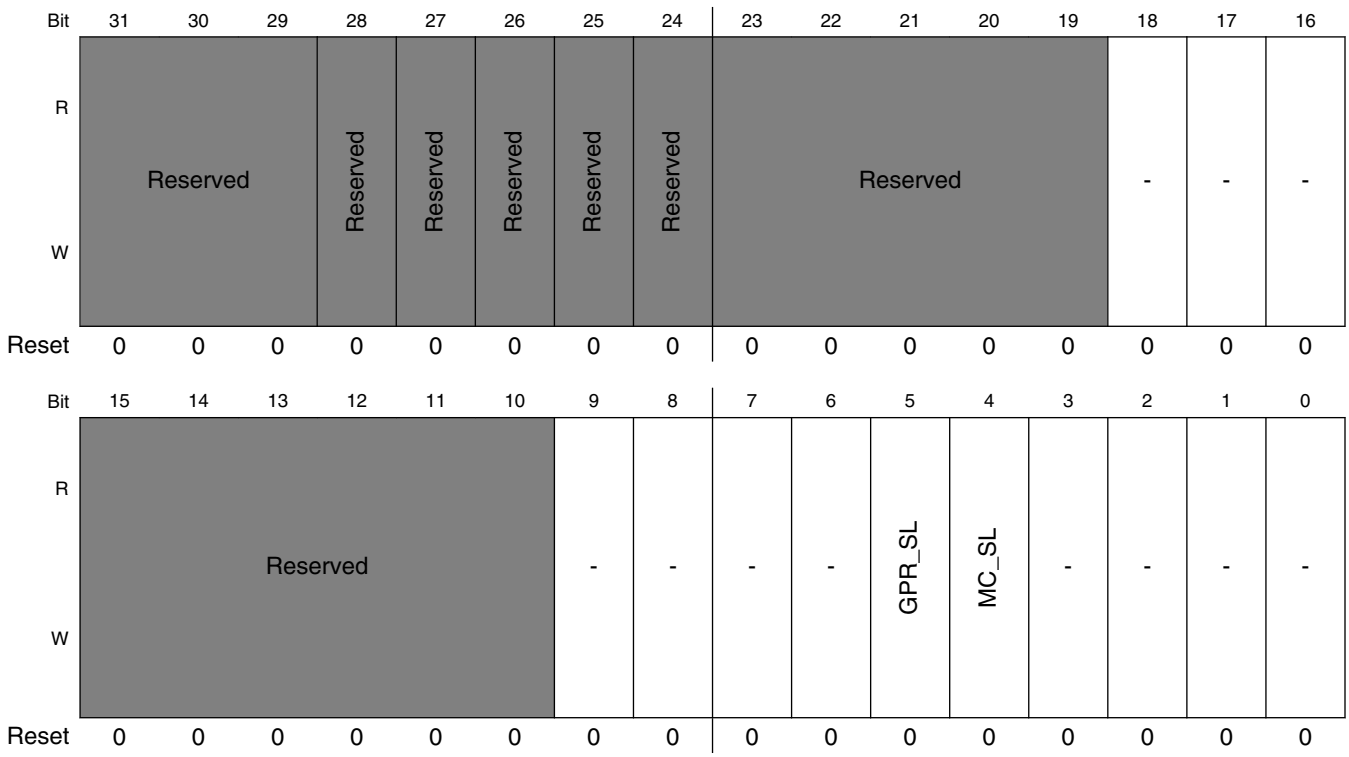
SNVS memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_C014	SNVS_HP Status register (SNVS_HPSR)	32	R/W	8000_0000h	57.9.4/4955
20C_C024	SNVS_HP Real-Time Counter MSB Register (SNVS_HPRTCMR)	32	R/W	0000_0000h	57.9.5/4957
20C_C028	SNVS_HP Real-Time Counter LSB Register (SNVS_HPRTCLR)	32	R/W	0000_0000h	57.9.6/4958
20C_C02C	SNVS_HP Time Alarm MSB Register (SNVS_HPTAMR)	32	R/W	0000_0000h	57.9.7/4958
20C_C030	SNVS_HP Time Alarm LSB Register (SNVS_HPTALR)	32	R/W	0000_0000h	57.9.8/4959
20C_C034	SNVS_LP Lock Register (SNVS_LPLR)	32	R/W	0000_0000h	57.9.9/4960
20C_C038	SNVS_LP Control Register (SNVS_LPCR)	32	R/W	0000_0020h	57.9.10/ 4962
20C_C04C	SNVS_LP Status Register (SNVS_LPSR)	32	R/W	0000_0008h	57.9.11/ 4964
20C_C05C	SNVS_LP Secure Monotonic Counter MSB Register (SNVS_LPSMCMR)	32	R/W	0000_0000h	57.9.12/ 4966
20C_C060	SNVS_LP Secure Monotonic Counter LSB Register (SNVS_LPSMCLR)	32	R/W	0000_0000h	57.9.13/ 4967
20C_C068	SNVS_LP General-Purpose Register (SNVS_LPGPR)	32	R/W	0000_0000h	57.9.14/ 4967
20C_CBF8	SNVS_HP Version ID Register 1 (SNVS_HPVIDR1)	32	R	003E_0100h	57.9.15/ 4968
20C_CBFC	SNVS_HP Version ID Register 2 (SNVS_HPVIDR2)	32	R	0000_0000h	57.9.16/ 4968

57.9.1 SNVS_HP Lock register (SNVS_HPLR)

The SNVS_HP lock register contains the lock bits for the SNVS registers. This is a privileged write register.

Address: 20C_C000h base + 0h offset = 20C_C000h



SNVS_HPLR field descriptions

Field	Description
31–29 -	This field is reserved.
28 -	This field is reserved.
27 -	This field is reserved.
26 -	This field is reserved.
25 -	This field is reserved.
24 -	This field is reserved.
23–19 -	This field is reserved.

Table continues on the next page...

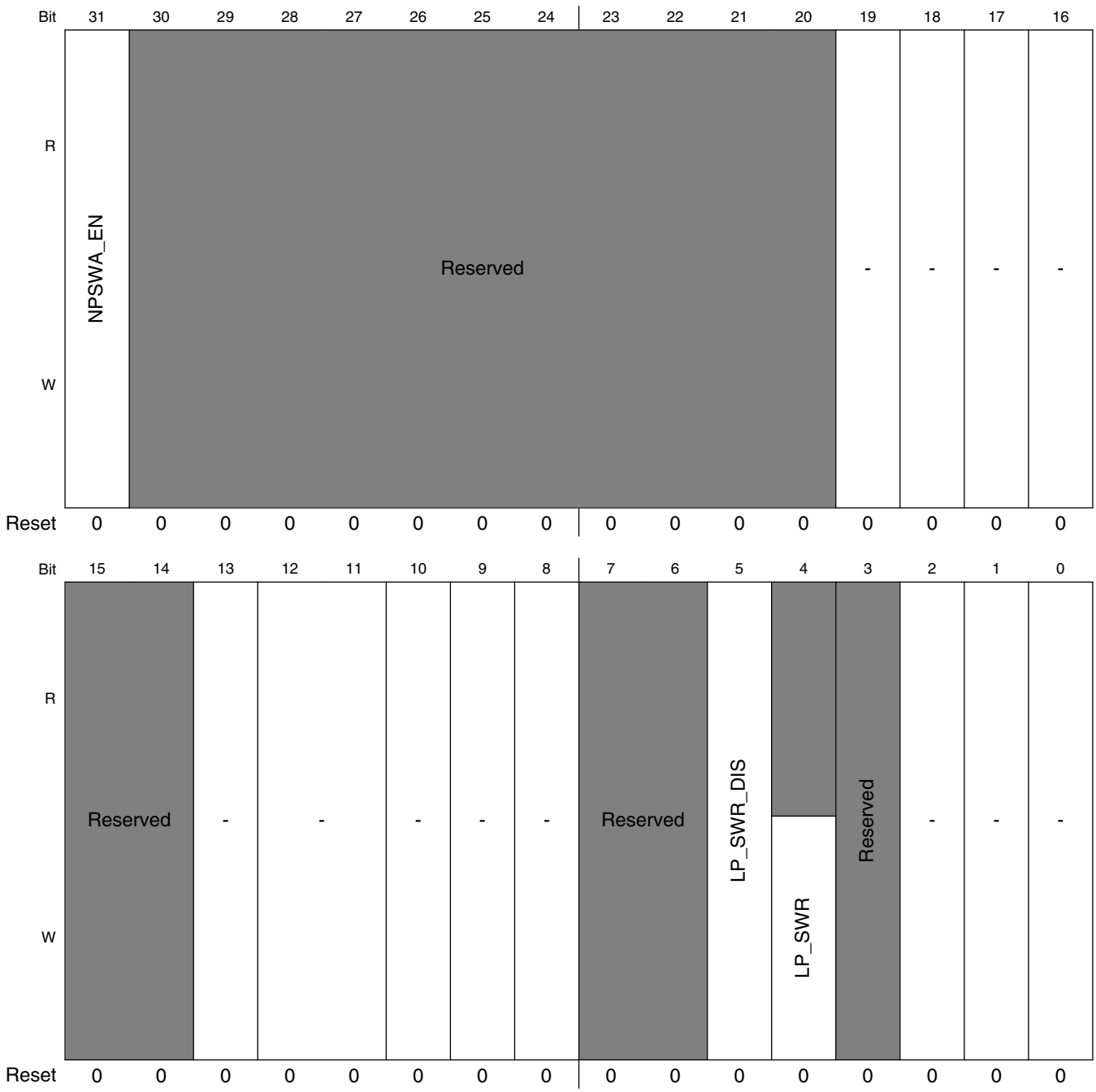
SNVS_HPLR field descriptions (continued)

Field	Description
18 -	Security-related field
17 -	Security-related field
16 -	Security-related field
15–10 -	This field is reserved.
9 -	Security-related field
8 -	Security-related field
7 -	Security-related field
6 -	Security-related field
5 GPR_SL	General-Purpose Register Soft Lock When set, it prevents any writes to the GPR. Once set, this bit can only be reset by the system reset. 0 Write access is allowed. 1 Write access is not allowed.
4 MC_SL	Monotonic Counter Soft Lock When set, it prevents any writes (increments) to the MC registers and the MC_ENV bit. Once set, this bit can only be reset by the system reset. 0 Write access (increment) is allowed. 1 Write access (increment) is not allowed.
3 -	Security-related field
2 -	Security-related field
1 -	Security-related field
0 -	Security-related field.

57.9.2 SNVS_HP Command register (SNVS_HPCOMR)

The SNVS_HP command register contains the command, configuration, and control bits for the SNVS block. This is a privileged write register.

Address: 20C_C000h base + 4h offset = 20C_C004h



SNVS_HPCOMR field descriptions

Field	Description
31 NPSWA_EN	Non-Privileged Software Access Enable When set, it allows non-privileged software to access all SNVS registers, including those that are privileged-software read/write access only. 0—only the privileged software can access the privileged registers. 1—any software can access the privileged registers.
30–20 -	This field is reserved.
19 -	Security-related field
18 -	Security-related field
17 -	Security-related field
16 -	Security-related field
15–14 -	This field is reserved.
13 -	Security-related field
12–11 -	Security-related field
10 -	Security-related field
9 -	Security-related field
8 -	Security-related field
7–6 -	This field is reserved.
5 LP_SWR_DIS	LP Software Reset Disable When set, it disables the LP software reset. Once set, this bit can only be reset by the system reset. 0 LP software reset is enabled. 1 LP software reset is disabled.
4 LP_SWR	LP Software Reset When set, it resets the SNVS_LP section. This bit can't be set when the LP_SWR_DIS bit is set. This self-clearing bit is always read as zero. 0 No action 1 Reset LP section
3 -	This field is reserved.
2 -	Security-related field
1 -	Security-related field

Table continues on the next page...

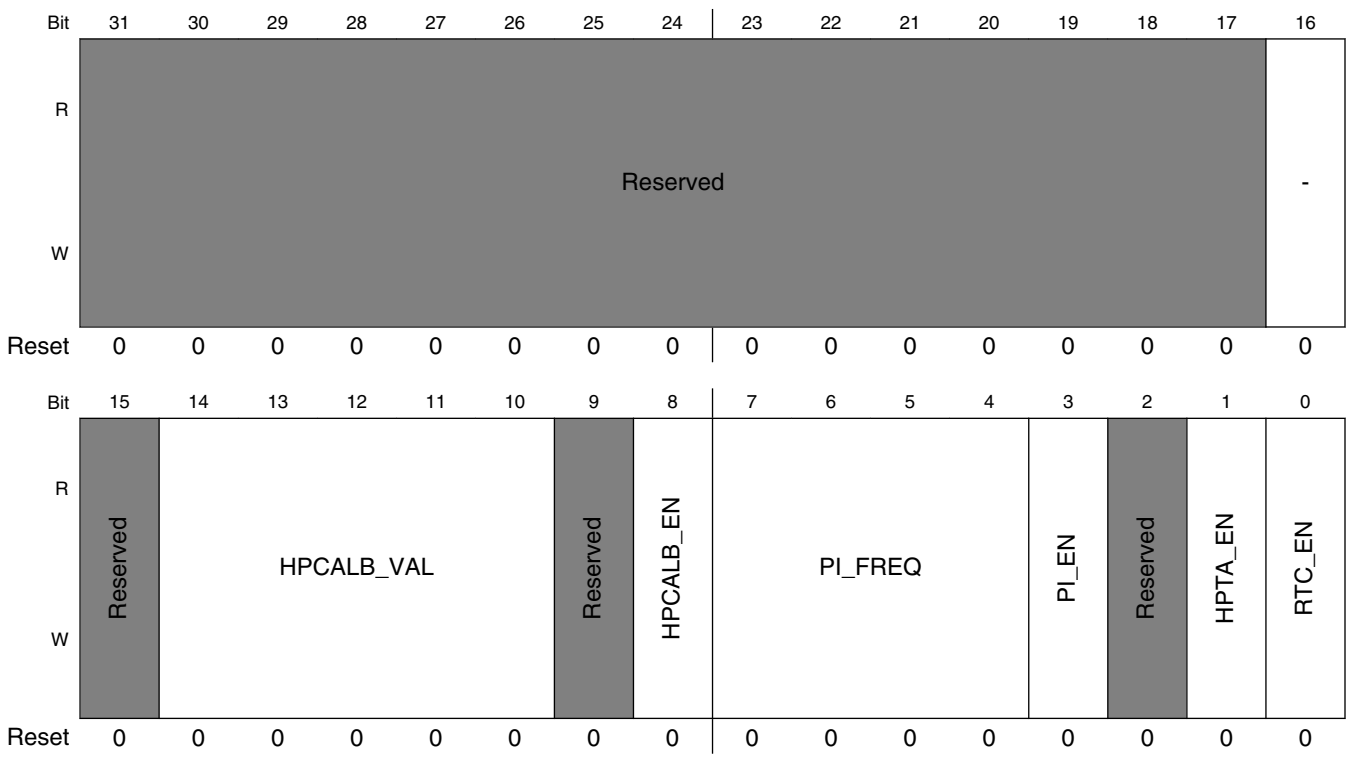
SNVS_HPCOMR field descriptions (continued)

Field	Description
0 -	Security-related field

57.9.3 SNVS_HP Control register (SNVS_HPCR)

The SNVS_HP control register contains various control bits of the HP section of the SNVS.

Address: 20C_C000h base + 8h offset = 20C_C008h



SNVS_HPCR field descriptions

Field	Description
31–17 -	This field is reserved.
16 -	Security-related field
15 -	This field is reserved.
14–10 HPCALB_VAL	HP Calibration Value

Table continues on the next page...

SNVS_HPCR field descriptions (continued)

Field	Description
	Defines the signed calibration value for the HP real-time counter. This field can be programmed only when the RTC calibration is disabled (HPCALB_EN is not set). This is a 5-bit 2's complement value, hence the allowable calibration values are in the range from -16 to +15 counts per 32768 ticks of the counter. 00000 +0 counts per each 32768 ticks of the counter 00001 +1 counts per each 32768 ticks of the counter 00010 +2 counts per each 32768 ticks of the counter 01111 +15 counts per each 32768 ticks of the counter 10000 -16 counts per each 32768 ticks of the counter 10001 -15 counts per each 32768 ticks of the counter 11110 -2 counts per each 32768 ticks of the counter 11111 -1 counts per each 32768 ticks of the counter
9 -	This field is reserved.
8 HPCALB_EN	HP Real-Time Counter Calibration Enabled Indicates that the time-calibration mechanism is enabled. 0 HP timer calibration is disabled. 1 HP timer calibration is enabled.
7–4 PI_FREQ	Periodic Interrupt Frequency Defines the frequency of the periodic interrupt. The interrupt is generated when a zero-to-one or one-to-zero transition occurs on the selected bit of the HP real-time counter, and the real-time counter and the periodic interrupt are both enabled (RTC_EN and PI_EN are set). It is recommended to program this field when the periodic interrupt is disabled (PI_EN is not set). The possible frequencies are: 0000 - Bit 0 of the RTC is selected as the source of the periodic interrupt. 0001 - Bit 1 of the RTC is selected as the source of the periodic interrupt. 0010 - Bit 2 of the RTC is selected as the source of the periodic interrupt. 0011 - Bit 3 of the RTC is selected as the source of the periodic interrupt. 0100 - Bit 4 of the RTC is selected as the source of the periodic interrupt. 0101 - Bit 5 of the RTC is selected as the source of the periodic interrupt. 0110 - Bit 6 of the RTC is selected as the source of the periodic interrupt. 0111 - Bit 7 of the RTC is selected as the source of the periodic interrupt. 1000 - Bit 8 of the RTC is selected as the source of the periodic interrupt. 1001 - Bit 9 of the RTC is selected as the source of the periodic interrupt. 1010 - Bit 10 of the RTC is selected as the source of the periodic interrupt. 1011 - Bit 11 of the RTC is selected as the source of the periodic interrupt. 1100 - Bit 12 of the RTC is selected as the source of the periodic interrupt. 1101 - Bit 13 of the RTC is selected as the source of the periodic interrupt. 1110 - Bit 14 of the RTC is selected as the source of the periodic interrupt. 1111 - Bit 15 of the RTC is selected as the source of the periodic interrupt.
3 PI_EN	HP Periodic Interrupt Enable The periodic interrupt can be generated only if the HP real-time counter is enabled. 0 HP periodic interrupt is disabled. 1 HP periodic interrupt is enabled.
2 -	This field is reserved.

Table continues on the next page...

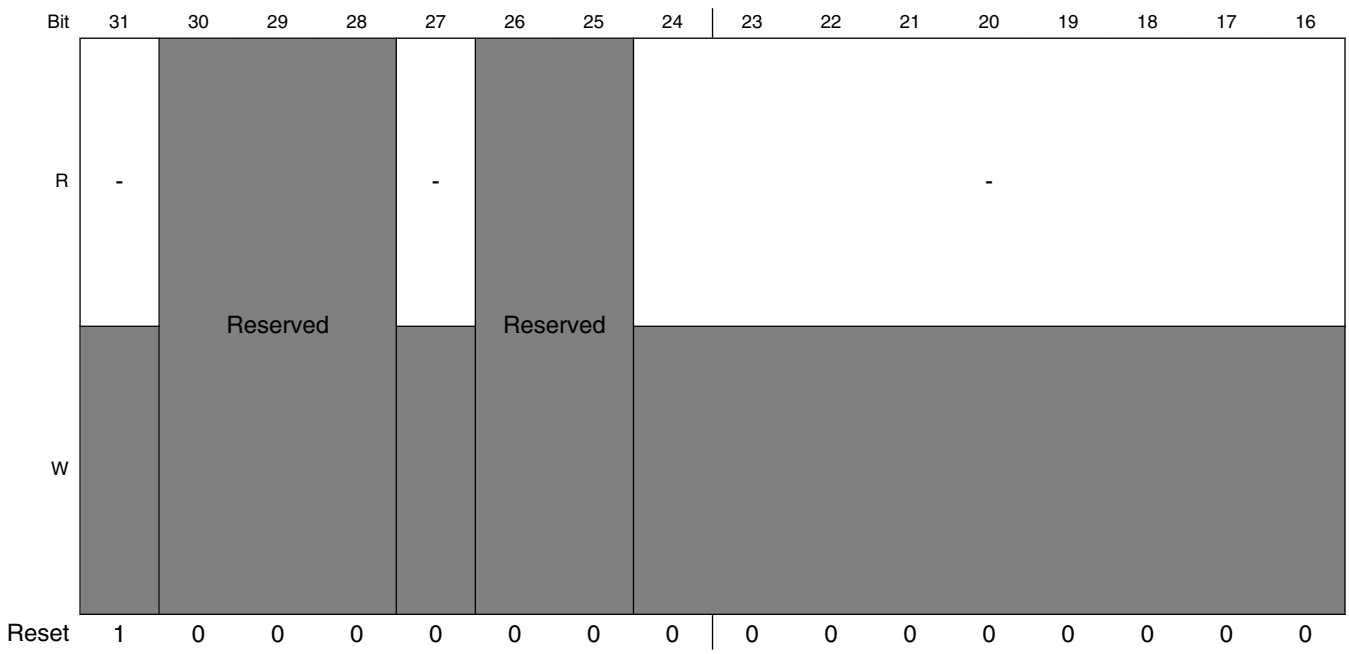
SNVS_HPCR field descriptions (continued)

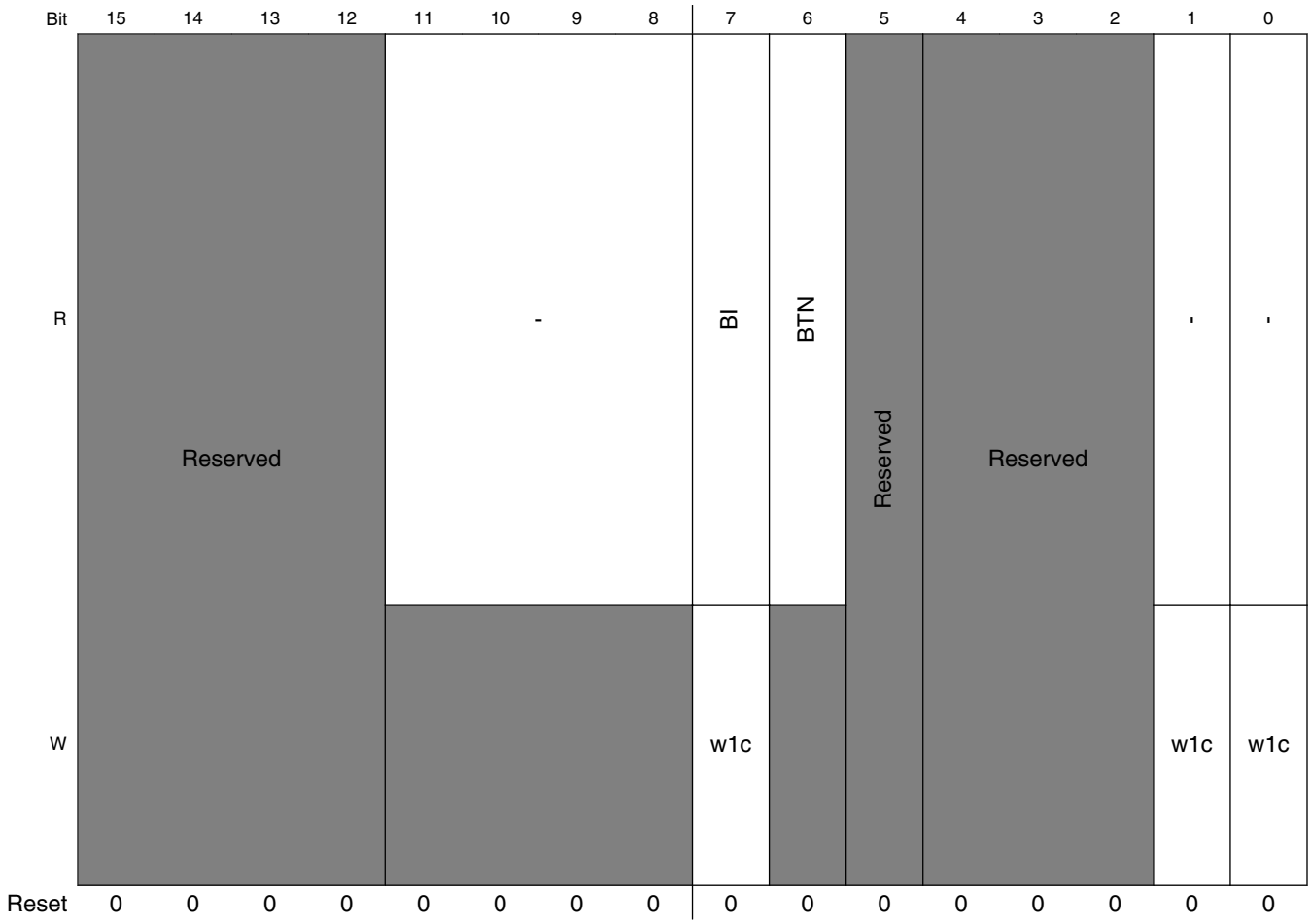
Field	Description
1 HPTA_EN	HP Time Alarm Enable When set, the time alarm interrupt is generated if the value in the HP time alarm registers is equal to the value of the HP real-time counter. 0 HP time alarm interrupt is disabled. 1 HP time alarm interrupt is enabled.
0 RTC_EN	HP Real-Time Counter Enable 0 RTC is disabled. 1 RTC is enabled.

57.9.4 SNVS_HP Status register (SNVS_HPSR)

The SNVS_HP status register reflects the internal state of the SNVS.

Address: 20C_C000h base + 14h offset = 20C_C014h





SNVS_HPSR field descriptions

Field	Description
31 -	Security-related field
30–28 -	This field is reserved.
27 -	Security-related field
26–25 -	This field is reserved.
24–16 -	Security-related field
15–12 -	This field is reserved.
11–8 -	Security-related field
7 BI	Button interrupt. The ipi_snvs_btn_int_b signal was asserted.
6 BTN	Value of the BTN input. This is the external button used for the PMIC control.

Table continues on the next page...

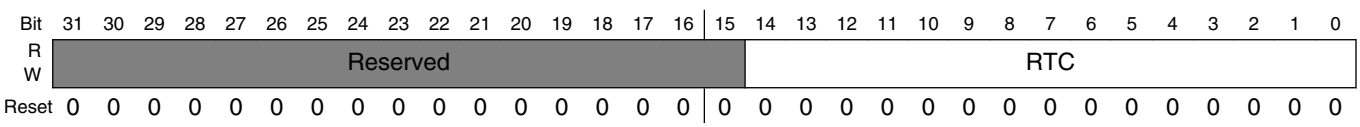
SNVS_HPSR field descriptions (continued)

Field	Description
	0—BTN is not pressed. 1—BTN is pressed.
5 -	This field is reserved.
4–2 -	This field is reserved.
1 -	Security-related field
0 -	Security-related field

57.9.5 SNVS_HP Real-Time Counter MSB Register (SNVS_HPRTCMR)

The SNVS_HP real-time counter MSB register contains the most significant bits of the HP real-time counter.

Address: 20C_C000h base + 24h offset = 20C_C024h



SNVS_HPRTCMR field descriptions

Field	Description
31–15 -	This field is reserved.
RTC	HP Real-Time Counter Most significant 15 bits. This register can be programmed only when the RTC is not active (RTC_EN bit is not set).

57.9.6 SNVS_HP Real-Time Counter LSB Register (SNVS_HPRTCLR)

The SNVS_HP real-time counter LSB register contains the 32 least significant bits of the HP real-time counter.

Address: 20C_C000h base + 28h offset = 20C_C028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RTC																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SNVS_HPRTCLR field descriptions

Field	Description
RTC	HP Real-Time Counter Least significant 32 bits. This register can be programmed only when the RTC is not active (the RTC_EN bit is not set).

57.9.7 SNVS_HP Time Alarm MSB Register (SNVS_HPTAMR)

The SNVS_HP time alarm MSB register contains the most significant bits of the SNVS_HP time alarm value.

Address: 20C_C000h base + 2Ch offset = 20C_C02Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																HPTA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SNVS_HPTAMR field descriptions

Field	Description
31–15 -	This field is reserved.
HPTA	HP Time Alarm Most significant 15 bits. This register can be programmed only when the HP time alarm is disabled (the HPTA_EN bit is not set).

57.9.8 SNVS_HP Time Alarm LSB Register (SNVS_HPTALR)

The SNVS_HP time alarm LSB register contains the 32 least significant bits of the SNVS_HP time alarm value.

Address: 20C_C000h base + 30h offset = 20C_C030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HPTA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

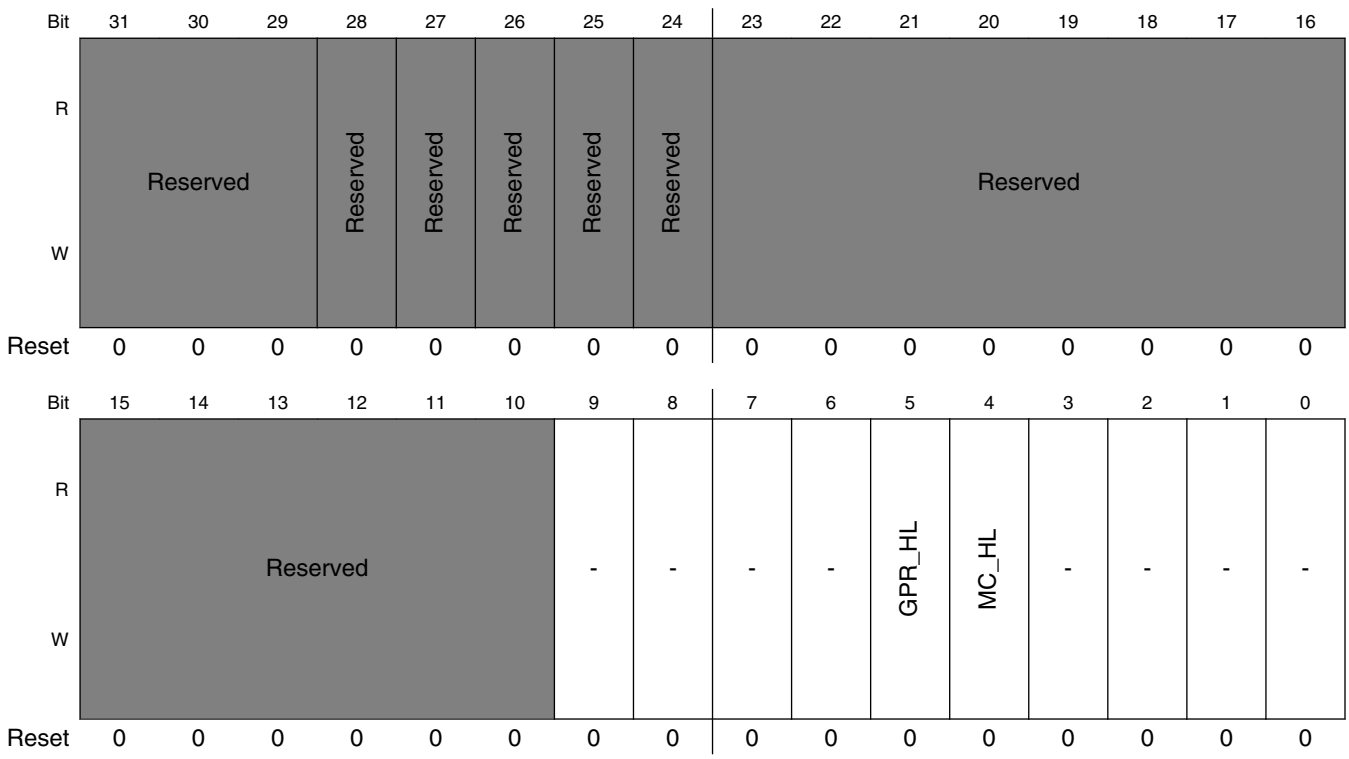
SNVS_HPTALR field descriptions

Field	Description
HPTA	HP Time Alarm The least significant bits. This register can be programmed only when the HP time alarm is disabled (the HPTA_EN bit is not set).

57.9.9 SNVS_LP Lock Register (SNVS_LPLR)

The SNVS_LP lock register contains the lock bits for the SNVS_LP registers.

Address: 20C_C000h base + 34h offset = 20C_C034h



SNVS_LPLR field descriptions

Field	Description
31–29 -	This field is reserved.
28 -	This field is reserved.
27 -	This field is reserved.
26 -	This field is reserved.
25 -	This field is reserved.
24 -	This field is reserved.
23–10 -	This field is reserved.

Table continues on the next page...

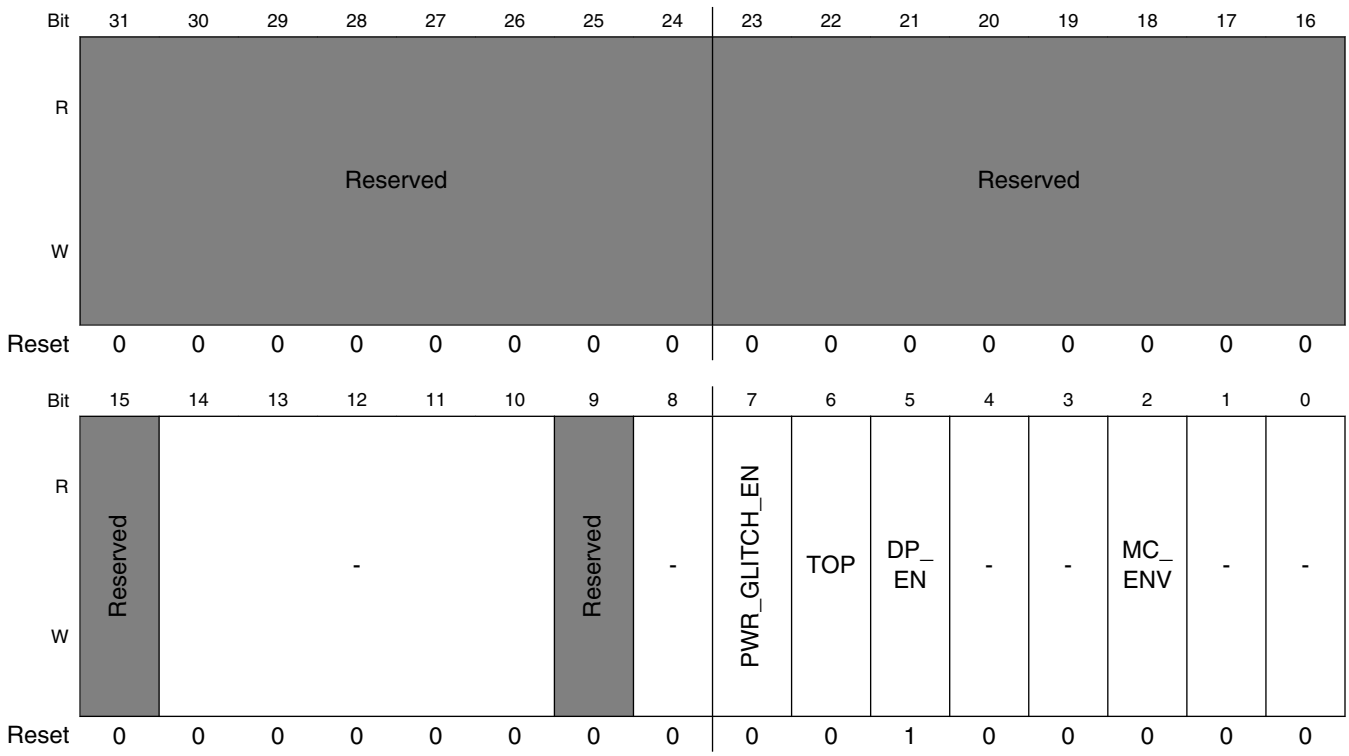
SNVS_LPLR field descriptions (continued)

Field	Description
9 -	Security-related field
8 -	Security-related field
7 -	Security-related field
6 -	Security-related field
5 GPR_HL	General-Purpose Register Hard Lock When set, it blocks any writes to the GPR. Once set, this bit can only be reset by the LP POR. 0 Write access is allowed. 1 Write access is not allowed.
4 MC_HL	Monotonic Counter Hard Lock When set, it blocks any writes (increments) to the MC registers and the MC_ENV bit. Once set, this bit can only be reset by the LP POR. 0 Write access (increment) is allowed. 1 Write access (increment) is not allowed.
3 -	Security-related field
2 -	Security-related field
1 -	Security-related field
0 -	Security-related field

57.9.10 SNVS_LP Control Register (SNVS_LPCR)

The SNVS_LP control register contains various control bits of the LP section of the SNVS.

Address: 20C_C000h base + 38h offset = 20C_C038h



SNVS_LPCR field descriptions

Field	Description
31–24 -	This field is reserved.
23–16 -	This field is reserved.
15 -	This field is reserved.
14–10 -	Security-related field
9 -	This field is reserved.
8 -	Security-related field

Table continues on the next page...

SNVS_LPCR field descriptions (continued)

Field	Description
7 PWR_GLITCH_EN	By default, the detection of a power glitch does not cause the pmic_en_b signal to be asserted. Setting the power glitch enable bit to 1 enables the power-glitch event for the PMIC. 0—disabled 1—enabled
6 TOP	Turn off System Power Asserting this bit causes a signal to be sent to the power management IC to turn the system power off. This bit clears once the power is off. This bit is only valid when the dumb PMIC is enabled. 0 Leave the system power on. 1 Turn the system power off.
5 DP_EN	Dumb PMIC Enabled When set, the software can control the system power. When cleared, the system requires the smart PMIC to automatically turn the power off. **NOTE:** This SoC is internally hard-wired to the dumb PMIC mode. Changing the state of this field only affects the behavior of the SNVS[TOP] field. 0 Smart PMIC is enabled. (not supported) 1 Dumb PMIC is enabled.
4 -	Security-related field
3 -	Security-related field
2 MC_ENV	Monotonic Counter Enable and Valid When set, the MC can be incremented (by a write transaction to the LPSMCMR or LPSMCLR). This bit can't be changed once the MC_SL or MC_HL bits are set. 0 MC is disabled or invalid. 1 MC is enabled and valid.
1 -	Security-related field
0 -	Security-related field

57.9.11 SNVS_LP Status Register (SNVS_LPSR)

The SNVS_LP status register reflects the internal state and behavior of the SNVS_LP.

Address: 20C_C000h base + 4Ch offset = 20C_C04Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-	-	Reserved									-	Reserved	SPO	EO	-
W														w1c	w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					-	-	-	-	-	-	-	-	MCR	-	-
W														w1c		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

SNVS_LPSR field descriptions

Field	Description
31 -	Security-related field
30 -	Security-related field
29–21 -	This field is reserved.
20 -	Security-related field.
19 -	This field is reserved.
18 SPO	Set Power Off The SPO bit is set when the set_pwr_off_irq interrupt is triggered, which happens when the software writes a 1 to the TOP bit in the LPCR or when the power button is pressed longer than the configured debounce time. Writing to the SPO bit clears the set_pwr_off_irq interrupt. 0 Emergency off is not detected. 1 Emergency off is detected.
17 EO	Emergency Off This bit is set when a power off is requested. 0 Emergency off is not detected. 1 Emergency off is detected.
16 -	Security-related field
15–11 -	This field is reserved.
10 -	Security-related field
9 -	Security-related field
8 -	Security-related field
7 -	Security-related field
6 -	Security-related field
5 -	Security-related field
4 -	Security-related field
3 -	Security-related field
2 MCR	Monotonic Counter Rollover 0 MC did not reach its maximum value. 1 MC reached its maximum value.

Table continues on the next page...

SNVS_LPSR field descriptions (continued)

Field	Description
1 -	Security-related field
0 -	Security-related field

57.9.12 SNVS_LP Secure Monotonic Counter MSB Register (SNVS_LPSMCMR)

The SNVS_LP secure monotonic counter MSB register contains the monotonic counter era bits and the most-significant 16 bits of the monotonic counter. The monotonic counter is incremented by one if there is a write command to the LPSMCMR register or the LPSMCLR register.

Address: 20C_C000h base + 5Ch offset = 20C_C05Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MC_ERA_BITS																MON_COUNTER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SNVS_LPSMCMR field descriptions

Field	Description
31–16 MC_ERA_BITS	Monotonic Counter Era Bits These bits are the inputs to the module and are typically connected to the fuses.
MON_COUNTER	Monotonic Counter Most-Significant 16 Bits The MC is incremented by one when: • A write transaction to the LPSMCMR register or the LPSMCLR register is detected. • The MC_ENV bit is set. • The MC_SL and MC_HL bits are not set.

57.9.13 SNVS_LP Secure Monotonic Counter LSB Register (SNVS_LPSMCLR)

The SNVS_LP secure monotonic counter LSB register contains the 32 least-significant bits of the monotonic counter. The MC is incremented by one if there is a write command to the LPSMCMR register or the LPSMCLR register.

Address: 20C_C000h base + 60h offset = 20C_C060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MON_COUNTER																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SNVS_LPSMCLR field descriptions

Field	Description
MON_COUNTER	Monotonic Counter bits The MC is incremented by one when: • A write transaction to the LPSMCMR register or the LPSMCLR register is detected. • The MC_ENV bit is set. • The MC_SL and MC_HL bits are not set.

57.9.14 SNVS_LP General-Purpose Register (SNVS_LPGPR)

The SNVS_LP general-purpose register is a 32-bit read/write register located in the low-power domain. Because the LPGPR is located in the battery-backed power domain, the LPGPR can be used by any application for retaining data during the SoC power-down mode.

Address: 20C_C000h base + 68h offset = 20C_C068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SNVS_LPGPR field descriptions

Field	Description
GPR	General-Purpose Register When the GPR_SL or GPR_HL bit is set, the register can't be programmed.

57.9.15 SNVS_HP Version ID Register 1 (SNVS_HPVIDR1)

The SNVS_HP Version ID Register 1 is a read-only register that contains the current version of the SNVS. The version consists of a module ID, a major version number, and a minor version number.

Address: 20C_C000h base + BF8h offset = 20C_CBF8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IP_ID																MAJOR_REV								MINOR_REV							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

SNVS_HPVIDR1 field descriptions

Field	Description
31–16 IP_ID	SNVS block ID
15–8 MAJOR_REV	SNVS block major version number
MINOR_REV	SNVS block minor version number

57.9.16 SNVS_HP Version ID Register 2 (SNVS_HPVIDR2)

The SNVS_HP version ID register 2 is a read-only register that indicates the current version of the SNVS. The version ID register 2 consists of these fields: integration options, ECO revision, and configuration options.

Address: 20C_C000h base + BFCh offset = 20C_CBFCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IP_ERA								INTG_OPT								ECO_REV								CONFIG_OPT							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SNVS_HPVIDR2 field descriptions

Field	Description
31–24 IP_ERA	Era of the IP design 00h - Era 1 or 2 03h - Era 3

Table continues on the next page...

SNVS_HPVIDR2 field descriptions (continued)

Field	Description
	04h - Era 4 05h - Era 5
23–16 INTG_OPT	SNVS Integration Option
15–8 ECO_REV	SNVS ECO Revision
CONFIG_OPT	SNVS Configuration Option

Chapter 58
Shared Peripheral Bus Arbiter (SPBA)

58.1 Overview

The Shared Peripheral Bus Arbiter (SPBA) is a three-to-one IP Bus interface arbiter. Three masters arbitrate for shared peripheral access through the SPBA.

The SPBA has three primary functions:

- The IP Bus Line switches a master to one peripheral
- The Masters arbiter arbitrates between the three masters to solve concurrent access or restricted access to peripherals
- The Control Registers and Ownership Control includes a set of registers which are reachable through software and permit the access scheme to be defined for each peripheral (Resource Ownership and Access Control). It generates signals for the external steering logic of interrupts and DMA signals.

The figure below shows the SPBA block diagram

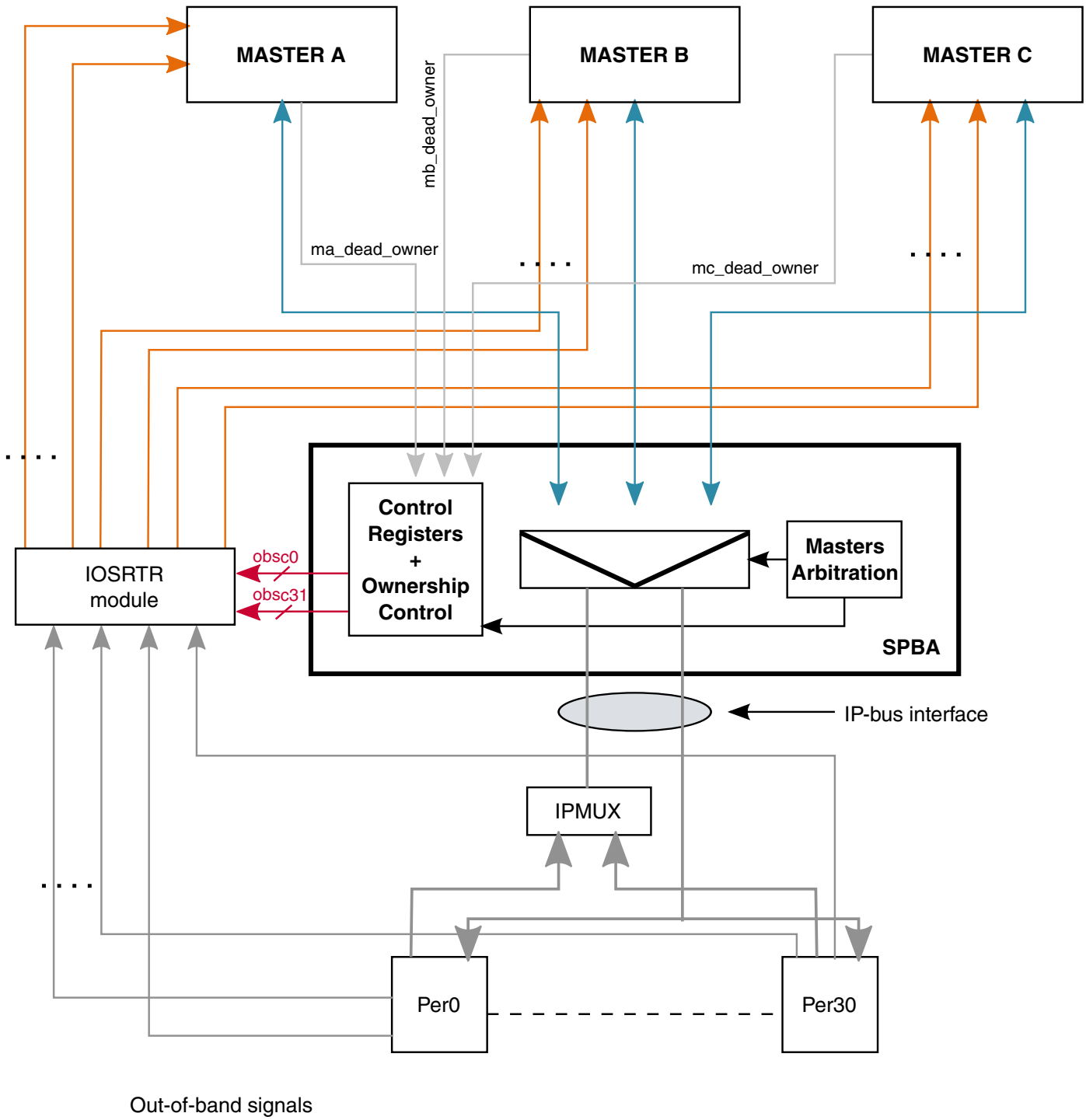


Figure 58-1. SPBA Block Diagram

58.1.1 Features

The SPBA includes the following features:

- Three IP Bus masters arbitration: Master A, B and C
- Support for DMA masters
- 32-bit data
- Supports up to 31 shared peripherals, each consuming 16 kilobytes of address space
- SPBA can be considered the 32nd peripheral, used for resource ownership and access control of the 31 peripherals
- Provides 31 sets of out of band steering control (OBSC) signals to the off-block steering logic
- Operating frequency up to 67 MHz
- Clocks: ipg_clk, ipg_clk_s

58.1.2 Modes of operation

SPBA behavior is transparent when accessing a peripheral, though it has these distinct modes of operation.

Reset/Abort

The SPBA has a hardware reset which initializes all registers, arbitration and peripherals rights registers (PRRs).

An abort signal input is provided allowing each master to abort its current access and release ownership (in case of master reset sequence).

Functional

Once a master request is granted, its IP Bus signals are steered to the requested peripheral.

Standby

No clock needed. The SPBA needs clocks only during access to the PRRs, arbitration, and abort phases. It generates two clock enable signals indicating when the clocks must be provided.

Configuration

During this phase, a master accesses the SPBA PRRs. The SPBA memory-mapped registers are seen as a shared peripheral.

58.2 Clocks

The table found here describes the clock sources for SPBA.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 58-1. SPBA Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

58.3 Functional description

58.3.1 Masters arbitration

The arbitration mechanism determines which port will control the master port, based on a simple round-robin arbitration scheme.

There are several use cases to consider.

- Only one master request per access. The master is switched to the shared peripheral bus, without arbitration. Figure 58-2 shows the MB request on the global module enable signal, served without wait state.
- If two masters simultaneously access SPBA, the last granted master is held off using the <master>_ips_xfr_wait output signal (default value is high). When the master is granted sips_xfr_wait, shared IP Bus peripheral is connected to <master>_ips_xfr_wait outputs.
- If three masters simultaneously access SPBA, then the last two granted masters are held off using <master>_ips_xfr_wait. Figure 58-3 shows a case in which the last two accesses granted are MA and MB. The requests are used even if they are in the same cycle.
- If after reset, at the first multiple access, no master has been granted, the priority is static: Master A (MA), Master B (MB) and last Master C (MC) port.
- No master request. No master switch to shared peripherals.

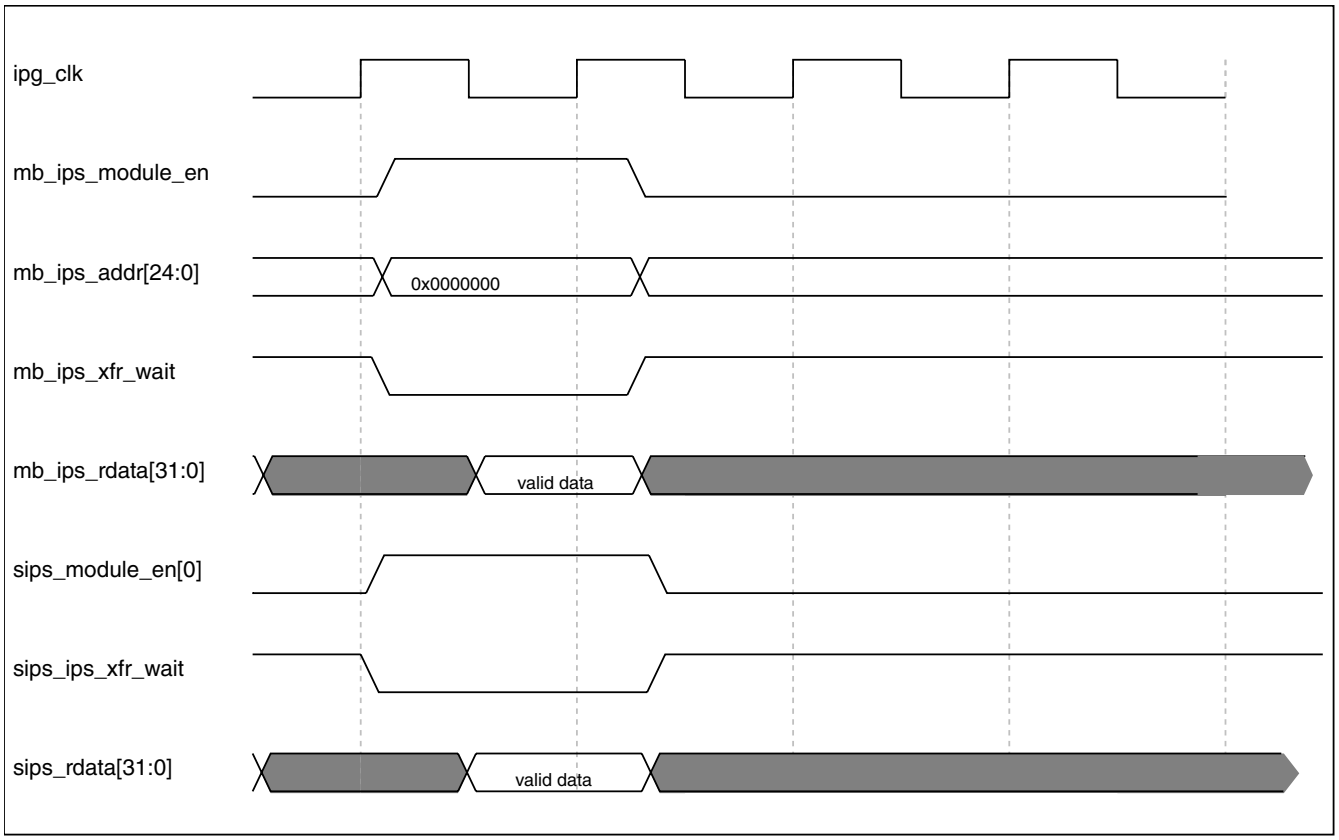


Figure 58-2. Example of one master request, no SPBA arbitration

The following figure assumes MA and MB have been the last two masters granted in the previous transfers (MA then MB).

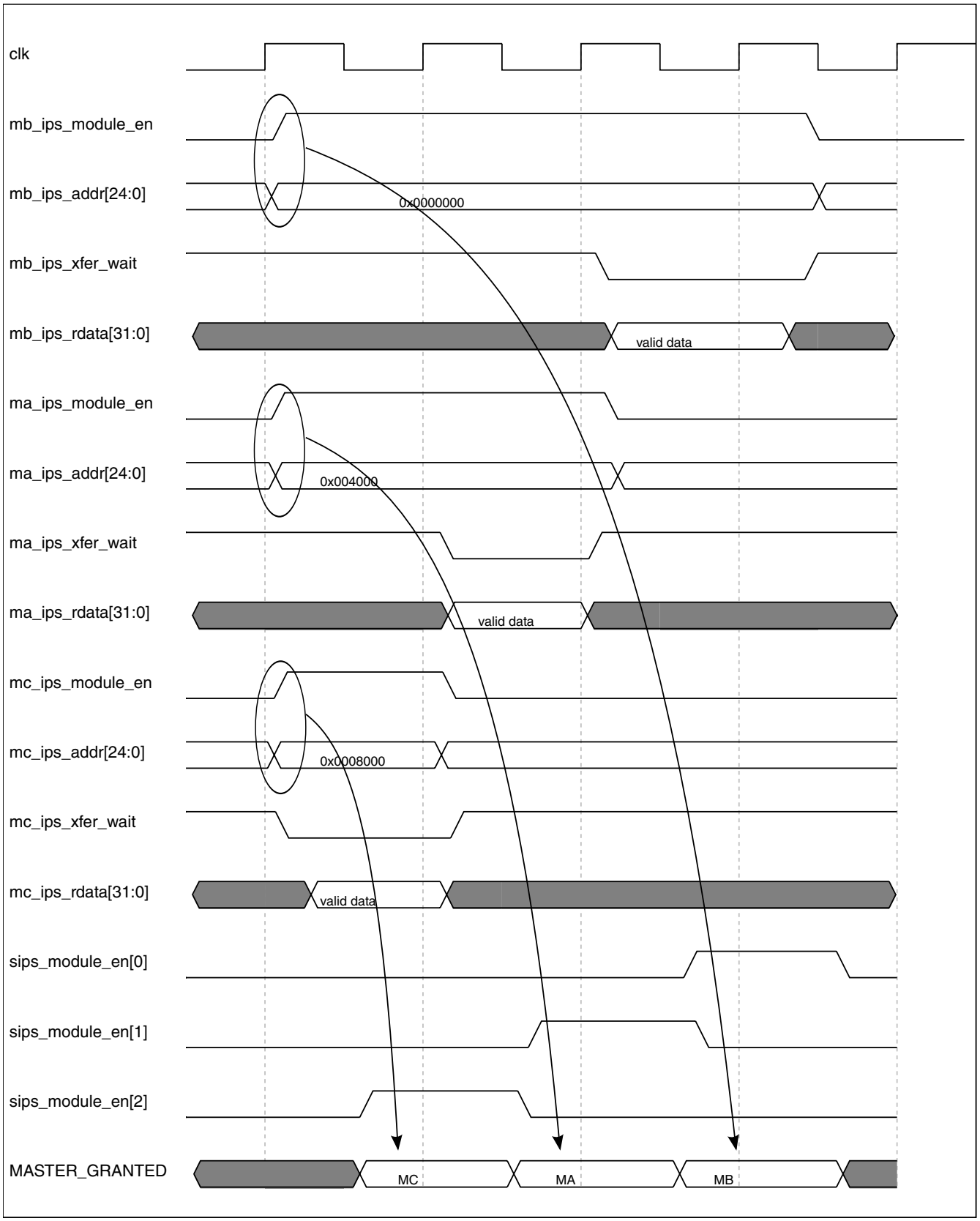


Figure 58-3. Example of three master requests: Masters already granted are "waited";

58.4 Resource ownership control

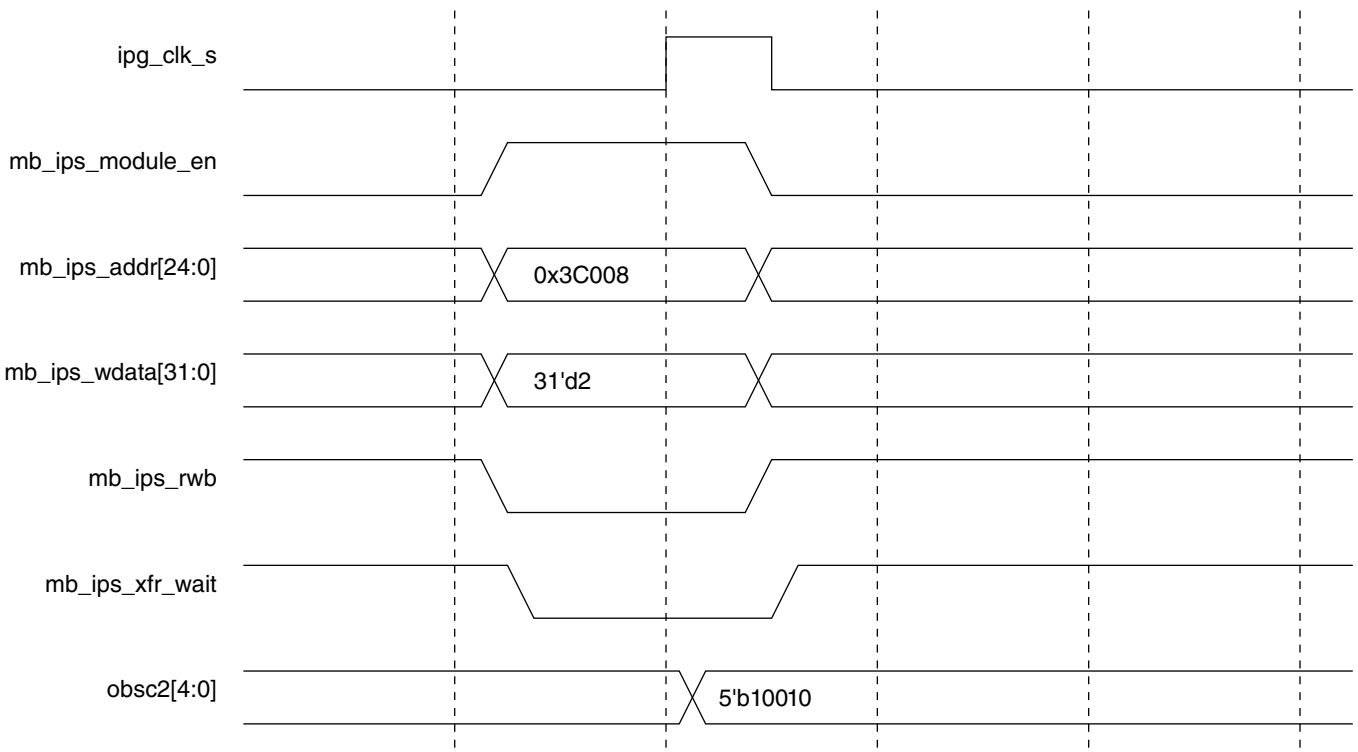
The resource ownership control regulates access to the shared peripherals and determines the steering of out-of-band signals.

58.4.1 Access control

58.4.1.1 Peripheral access

The peripheral access (resource access) of the requesting master is given by the corresponding RAR bit of the Peripheral Right Register. It determines if the master has access privilege to the resource.

Any attempt at access made by a requesting master whose access privilege bit is not set (in the PRR) is terminated with a bus error (**<master>_ips_xfr_err** is asserted by SPBA logic). The master that owns the resource can lock the peripheral for itself and/or grant other masters access to the peripheral by setting the appropriate bit(s) in the RAR field.



Master B is taking ownership of peripheral 2 by writing 3'b010 in the SPBA peripheral 2 right register (rarfield)
This ownership can be checked on obsc2 output as roi2[1:0] = 2'b10 and rar2[2:0] = 3'b010
(obsc[4:0] = {roi2[1], roi2[0], rar2[2], rar2[1], rar2[0]})

Figure 58-4. Example of one master B gaining ownership of peripheral 2

58.4.1.2 Peripheral Right Register access

The ROI bits of the Peripheral Right Register (PRR) determine which master is allowed to make write access to PRR. The identification of the requesting master is compared to the ROI bits of the PRR to determine if the master has ownership of the corresponding register.

Any attempted write access to a PRR already owned by another master will be ignored.

58.4.2 Owner election

When the peripheral is not owned by any master (ROI="00", after coming out of reset for instance), the first master to perform successfully a write to the RAR bits of the PRR is granted ownership of the peripheral and its associated PRR.

After writing to the PRR (RAR bit(s)), the master must read it back to make sure that it was granted ownership. If the RMO field is 2'b11, then the ownership claim is successful. If RMO is 2'b10, another master claimed ownership before this master was able to complete its write. This resolves the case in which two or more masters attempt to write the PRR at the same time; only the first master will be granted ownership. However all masters must read the PRR to determine if this case occurred, and if so, whether they were the first master which was granted ownership.

NOTE

A master that has been granted ownership of the PRR does not automatically have the right access to the peripheral; it must still set its own RAR bits in the PRR to access the peripheral.

58.4.3 Ending ownership

Ownership may be voluntarily ended by the owning master, or automatically upon assertion of a master-specific dead_owner signal.

The former is appropriate for software-controlled yielding of ownership. The latter is appropriate for automatic yielding of ownership when the owner has gone into reset.

When a master is reset, it clears the ROI bits of the PRRs owned by the corresponding master. When the owner is dead (in reset), all peripherals previously owned by that master must be changed to the un-owned state.

NOTE

It is the programmer's responsibility to make sure the peripherals are placed in an appropriate state before ending ownership.

58.4.3.1 Software Controlled Ownership Ending

The ROI bits will be automatically cleared when the master that owns the PRR access right clears (write) the RAR bits (Table 2).

It will then end the ownership of the PRR.

58.4.4 The Un-owned State

During the time when the peripheral is un-owned (i.e the ROI field contains all 0's), all masters have full access to it (RAR bits can then be modified by a master if ROI[1:0] = 2'b0).

In such cases it is necessary for software to ensure any necessary coherency in the resource, there is no hardware protection.

58.5 SPBA Memory Map/Register Definition

The SPBA control registers (Peripheral Right Registers) are mapped as a virtual shared peripheral.

SPBA can support up to 31 shared peripherals. Each of them has its own Peripheral Right Register (PRR) accessible within the SPBA memory-mapped registers, and consists of the Requesting Master Owner, the Resource Owner ID and the Resource Access Right fields.

SPBA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
203_C000	Peripheral Rights Register (SPBA_PRR0)	32	R/W	0000_0007h	58.5.1/4982
203_C004	Peripheral Rights Register (SPBA_PRR1)	32	R/W	0000_0007h	58.5.1/4982
203_C008	Peripheral Rights Register (SPBA_PRR2)	32	R/W	0000_0007h	58.5.1/4982
203_C00C	Peripheral Rights Register (SPBA_PRR3)	32	R/W	0000_0007h	58.5.1/4982
203_C010	Peripheral Rights Register (SPBA_PRR4)	32	R/W	0000_0007h	58.5.1/4982
203_C014	Peripheral Rights Register (SPBA_PRR5)	32	R/W	0000_0007h	58.5.1/4982
203_C018	Peripheral Rights Register (SPBA_PRR6)	32	R/W	0000_0007h	58.5.1/4982
203_C01C	Peripheral Rights Register (SPBA_PRR7)	32	R/W	0000_0007h	58.5.1/4982
203_C020	Peripheral Rights Register (SPBA_PRR8)	32	R/W	0000_0007h	58.5.1/4982

Table continues on the next page...

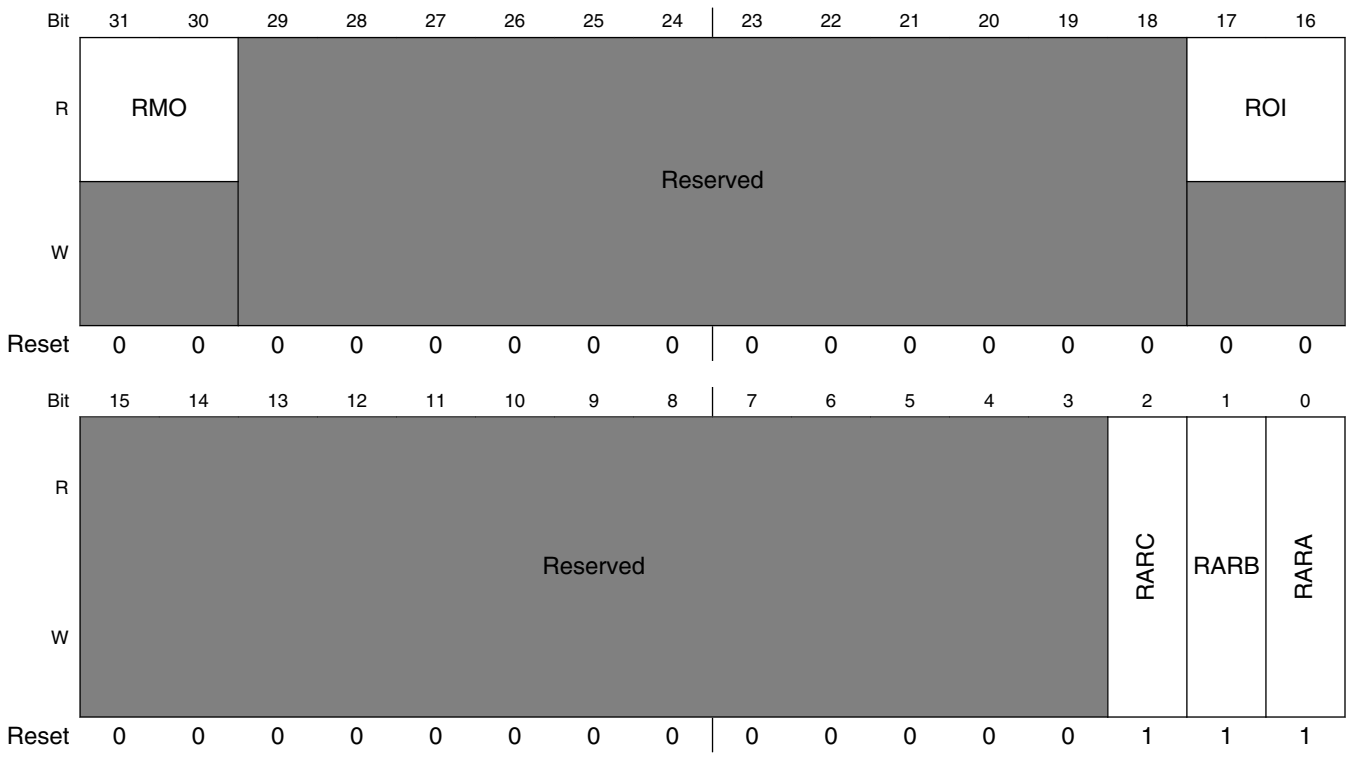
SPBA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
203_C024	Peripheral Rights Register (SPBA_PRR9)	32	R/W	0000_0007h	58.5.1/4982
203_C028	Peripheral Rights Register (SPBA_PRR10)	32	R/W	0000_0007h	58.5.1/4982
203_C02C	Peripheral Rights Register (SPBA_PRR11)	32	R/W	0000_0007h	58.5.1/4982
203_C030	Peripheral Rights Register (SPBA_PRR12)	32	R/W	0000_0007h	58.5.1/4982
203_C034	Peripheral Rights Register (SPBA_PRR13)	32	R/W	0000_0007h	58.5.1/4982
203_C038	Peripheral Rights Register (SPBA_PRR14)	32	R/W	0000_0007h	58.5.1/4982
203_C03C	Peripheral Rights Register (SPBA_PRR15)	32	R/W	0000_0007h	58.5.1/4982
203_C040	Peripheral Rights Register (SPBA_PRR16)	32	R/W	0000_0007h	58.5.1/4982
203_C044	Peripheral Rights Register (SPBA_PRR17)	32	R/W	0000_0007h	58.5.1/4982
203_C048	Peripheral Rights Register (SPBA_PRR18)	32	R/W	0000_0007h	58.5.1/4982
203_C04C	Peripheral Rights Register (SPBA_PRR19)	32	R/W	0000_0007h	58.5.1/4982
203_C050	Peripheral Rights Register (SPBA_PRR20)	32	R/W	0000_0007h	58.5.1/4982
203_C054	Peripheral Rights Register (SPBA_PRR21)	32	R/W	0000_0007h	58.5.1/4982
203_C058	Peripheral Rights Register (SPBA_PRR22)	32	R/W	0000_0007h	58.5.1/4982
203_C05C	Peripheral Rights Register (SPBA_PRR23)	32	R/W	0000_0007h	58.5.1/4982
203_C060	Peripheral Rights Register (SPBA_PRR24)	32	R/W	0000_0007h	58.5.1/4982
203_C064	Peripheral Rights Register (SPBA_PRR25)	32	R/W	0000_0007h	58.5.1/4982
203_C068	Peripheral Rights Register (SPBA_PRR26)	32	R/W	0000_0007h	58.5.1/4982
203_C06C	Peripheral Rights Register (SPBA_PRR27)	32	R/W	0000_0007h	58.5.1/4982
203_C070	Peripheral Rights Register (SPBA_PRR28)	32	R/W	0000_0007h	58.5.1/4982
203_C074	Peripheral Rights Register (SPBA_PRR29)	32	R/W	0000_0007h	58.5.1/4982
203_C078	Peripheral Rights Register (SPBA_PRR30)	32	R/W	0000_0007h	58.5.1/4982
203_C07C	Peripheral Rights Register (SPBA_PRR31)	32	R/W	0000_0007h	58.5.1/4982

58.5.1 Peripheral Rights Register (SPBA_PRR*n*)

This register controls master ownership and access for a peripheral.

Address: 203_C000h base + 0h offset + (4d × i), where i=0d to 31d



SPBA_PRR*n* field descriptions

Field	Description
31–30 RMO	Requesting Master Owner. This 2-bit register field indicates if the corresponding resource is owned by the requesting master or not. This register is reset to 2'b0 if ROI = 2'b0. 00 **UNOWNED** — The resource is unowned. 01 Reserved. 10 **ANOTHER_MASTER** — The resource is owned by another master. 11 **REQUESTING_MASTER** — The resource is owned by the requesting master.
29–18 -	This field is reserved. Reserved
17–16 ROI	Resource Owner ID. This field indicates which master (one at a time) can access to the PRR for rights modification. This is a read-only register. After reset, ROI bits are cleared ("00" -> un-owned resource).

Table continues on the next page...

SPBA_PRR*n* field descriptions (continued)

Field	Description
	A master performing a write access to the an un-owned PRR will get its ID automatically written into ROI, while modifying RARx bits. It can then read back the RMO, RAR, ROI bits to make sure RMO returns the right value, ROI bits contain its ID and RARx bits are correctly asserted. Then no other master (whom ID is different from the one stored in ROI) will be able to modify RAR fields. Owner master of a peripheral can assert its dead_owner signal, or write 1'b0 in the RARx to release the ownership (ROI[1:0] reset to 2'b0). 00 **UNOWNED** — Unowned resource. 01 **MASTER_A** — The resource is owned by master A port. 10 **MASTER_B** — The resource is owned by master B port. 11 **MASTER_C** — The resource is owned by master C port.
15–3 -	This field is reserved. Reserved
2 RARC	Resource Access Right. Control and Status bit for master C. This field indicates whether master C can access the peripheral. From 0 up to 3 masters can have permission to access a resource (all the master can be granted on a peripheral, but only one access at a time will be granted by SPBA). 0 **PROHIBITED** — Access to peripheral is not allowed. 1 **ALLOWED** — Access to peripheral is granted.
1 RARB	Resource Access Right. Control and Status bit for master B. This field indicates whether master B can access the peripheral. From 0 up to 3 masters can have permission to access a resource (all the master can be granted on a peripheral, but only one access at a time will be granted by SPBA). 0 **PROHIBITED** — Access to peripheral is not allowed. 1 **ALLOWED** — Access to peripheral is granted.
0 RARA	Resource Access Right. Control and Status bit for master A. This field indicates whether master A can access the peripheral. From 0 up to 3 masters can have permission to access a resource (all the master can be granted on a peripheral, but only one access at a time will be granted by SPBA). 0 **PROHIBITED** — Access to peripheral is not allowed. 1 **ALLOWED** — Access to peripheral is granted.

Chapter 59
Sony/Philips Digital Interface (SPDIF)

59.1 Overview

The Sony/Philips Digital Interface (SPDIF) audio block is a stereo transceiver that allows the processor to receive and transmit digital audio.

The SPDIF transceiver allows the handling of both SPDIF channel status (CS) and User (U) data and includes a frequency measurement block that allows the precise measurement of an incoming sampling frequency.

A recovered clock is provided to drive both internal components in the system, such as ESAI ports, and external components, such as A/Ds or D/As, with clocking control provided via related registers.

As the SPDIF internal data width is 24-bit, the eight most-significant bits of all registers return zeros.

The figure below shows a block diagram of the SPDIF transceiver data paths (receiver and transmitter) and its interface.

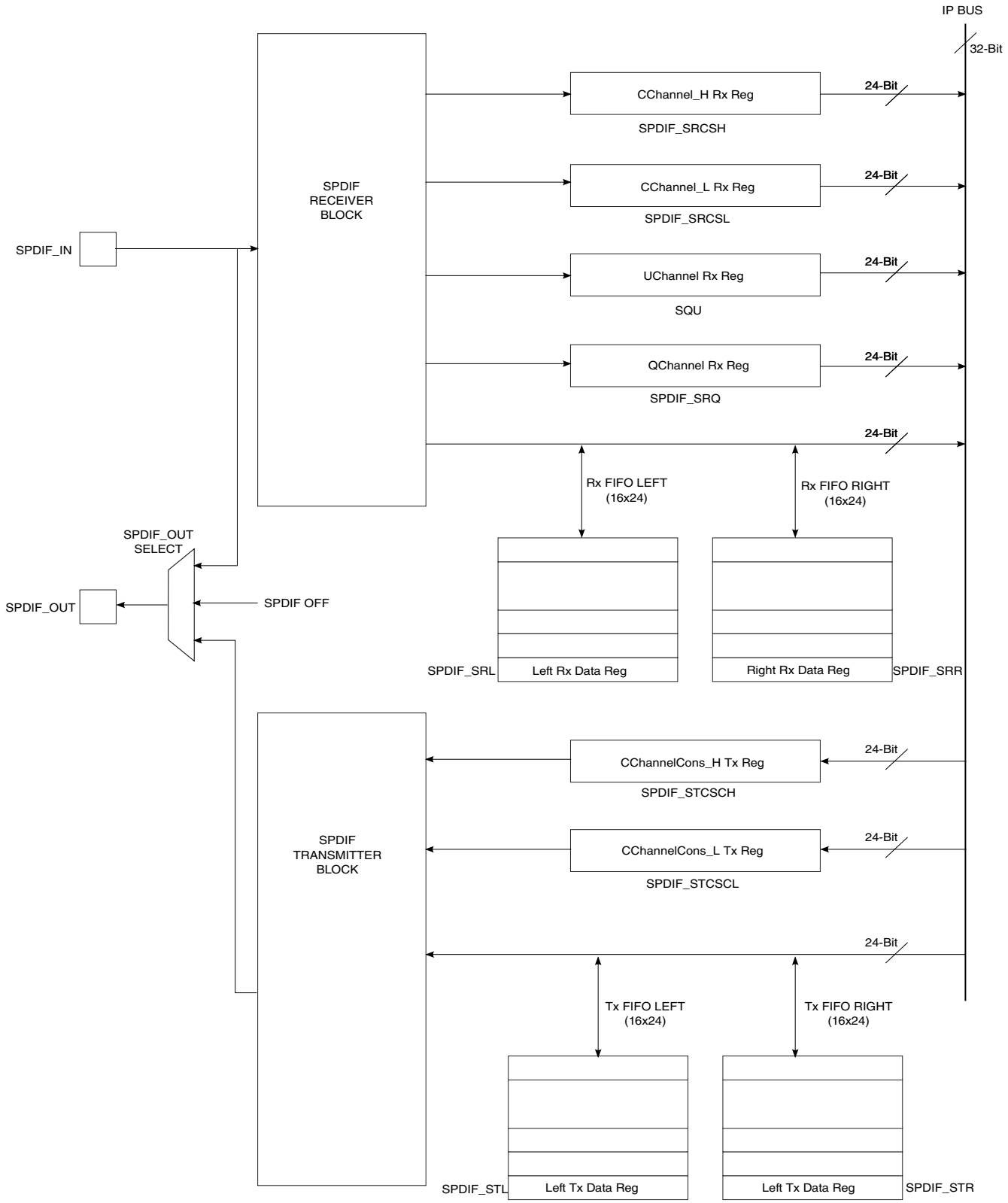


Figure 59-1. SPDIF Transceiver Data Interface Block Diagram

59.2 External Signals

The following table describes the external signals of SPDIF:

Table 59-1. SPDIF External Signals

Signal	Description	Pad	Mode	Direction
SPDIF_EXT_CLK	External clock signal	ENET_CRS_DV	ALT3	I
		RGMII_TXC	ALT2	
SPDIF_IN	Input line	EIM_D21	ALT7	I
		ENET_RX_ER	ALT3	
		GPIO_16	ALT4	
		KEY_COL3	ALT6	
SPDIF_LOCK	Lock signal	ENET_MDIO	ALT6	O
		GPIO_7	ALT6	
SPDIF_OUT	Output line signal	EIM_D22	ALT6	O
		ENET_RXD0	ALT3	
		GPIO_17	ALT4	
		GPIO_19	ALT2	
SPDIF_SR_CLK	SR Lock signal	ENET_REF_CLK	ALT6	O
		GPIO_8	ALT6	

59.3 Clocks

The table found here describes the clock sources for SPDIF.

Please see clock control block for clock setting, configuration and gating information.

Table 59-2. SPDIF Clocks

Clock name	Clock Root	Description
gclkw_t0	ipg_clk_root	Global clock
ipg_clk_s	ipg_clk_root	Peripheral access clock
tx_clk	spdif0_clk_root	Module Tx clock

59.4 Functional Description

The SPDIF is composed of two parts: SPDIF Receiver and SPDIF Transmitter.

The SPDIF receiver extracts the audio data from each SPDIF frame and places the data in the SPDIF Rx left and right FIFOs. The Channel Status and User Bits are also extracted from each frame and placed in the corresponding registers. The SPDIF receiver also provides a bypass option for direct transfer of the SPDIF input signal to the SPDIF transmitter.

For the SPDIF transmitter, the audio data is provided by the processor via the SPDIFTxLeft and SPDIFTxRight registers. The Channel Status bits are also provided via the corresponding registers. The SPDIF transmitter generates a SPDIF output bitstream in the biphase mark format (IEC60958), which consists of audio data, channel status and user bits.

In the SPDIF transmitter, the IEC60958 biphase bit stream is generated on both edges of the SPDIF Transmit clock. The SPDIF Transmit clock is generated by the SPDIF internal clock generate block and the sources are from outside of the SPDIF block. For the SPDIF receiver, it can recover the SPDIF Rx clock. Both the Rx clock and Tx clock are sent to the ASRC.Figure 1 shows the clock structure of the SPDIF transceiver.

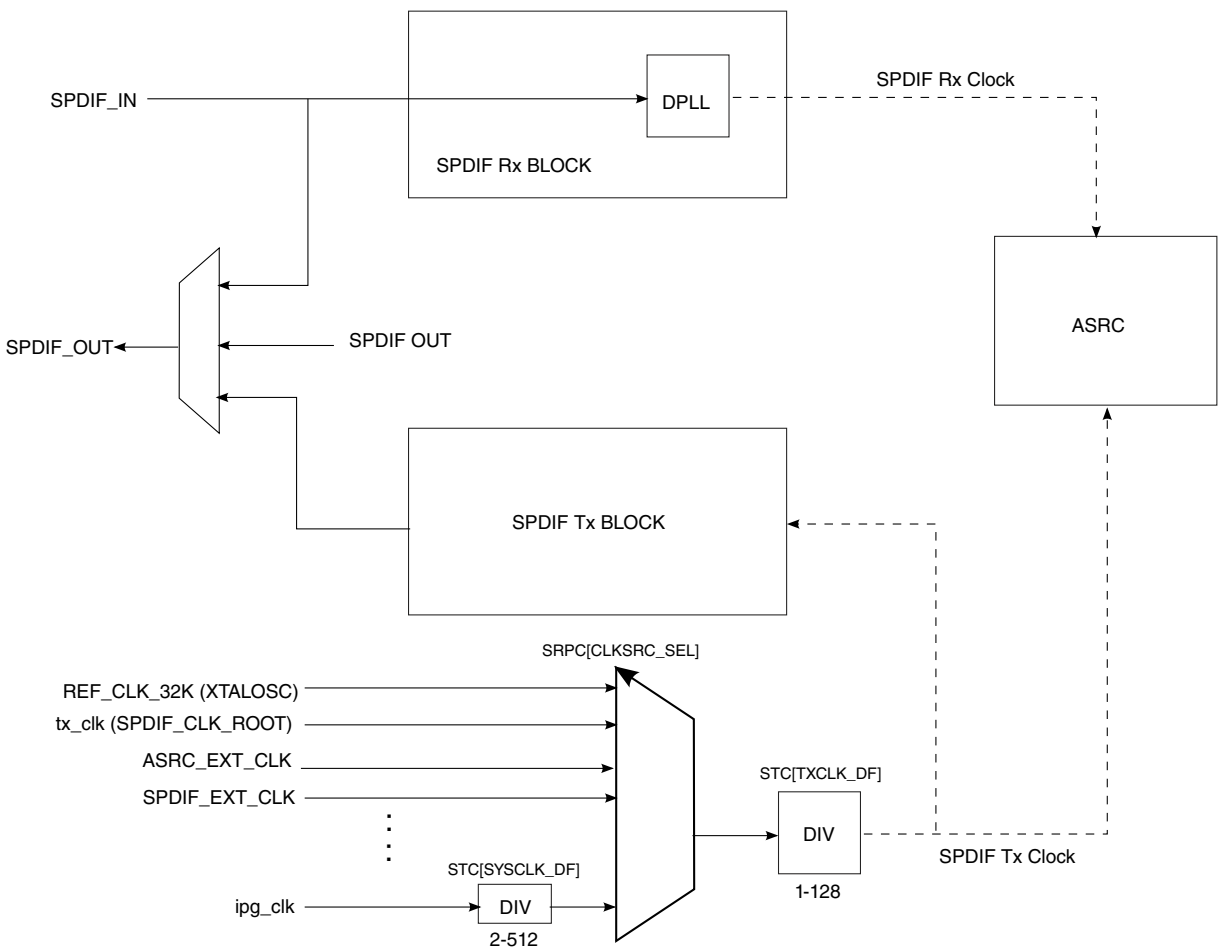


Figure 59-2. SPDIF Transceiver Clock Diagram

59.4.1 SPDIF Receiver

The SPDIF receiver extracts the audio data from each SPDIF frame and places the data in Rx left and right FIFOs.

The Tx left and right FIFOs are 16-deep and 24-bit-wide (equal to the audio data width). The Channel Status and User Bits are also extracted from each frame and placed in corresponding registers. The SPDIF receiver also provides a bypass option for direct transfer of the SPDIF input signal to the SPDIF transmitter.

The SPDIF receiver handles the main data audio stream and recovers the bit clock from the SPDIF input signal. The sample rate can be determined from the frequency measuring block. Additionally, the receiver supports the SPDIF C and U channels. The SPDIF C and U channel data is interfaced directly to memory-mapped registers.

All the data registers are controlled by the Interrupt Control Block and transferred to the memory-mapped IP bus.

The following functions are performed by the SPDIF receiver:

- Audio Data Reception see Audio Data Reception
- Channel Status bits Reception see Channel Status Reception
- U Channel bits Reception see User Bit Reception
- Validity Flag Reception see Validity Flag Reception
- SPDIF Receiver Exception support see SPDIF Receiver
- SPDIF Lock Detection

59.4.1.1 Audio Data Reception

The SPDIF Receiver block extracts the audio data from the IEC60958 stream, and outputs this via Rx left and right FIFOs to the memory-mapped registers SPDIFRxLeft and SPDIFRxRight.

Data from the SPDIF receiver is buffered in receive FIFO, and can be read by the processor from the memory-mapped registers.

- **SPDIF receiver data registers - Behavior on overrun, underrun**

The SPDIF Data Receive registers (SPDIFRxLeft and SPDIFRxRight) have individual FIFOs for left and right channel. As a result, there is always the possibility that left and right FIFOs may go out of sync due to FIFO underruns and FIFO overruns that affect

only one part (left or right) of any FIFOs. To prevent this from happening, hardware has been added to the device. Two mechanisms to prevent mismatch between the FIFOs are available.

If a SPDIF Data Rx FIFO overrun occurs on e.g. the right half of the FIFO, the sample that caused the overrun is not written to the right half (due to overrun). Special hardware will make sure the next sample is not written to the left half of the FIFO. If the overrun occurs on the left half of the FIFO, the next sample is not written to the right half of the FIFO.

- **SPDIF receiver data registers - Automatic resynchronization of FIFOs**

An automatic FIFO resynchronization feature is available. It can be enabled and disabled separately for every FIFO. If it is enabled, the hardware will check to see if the left and right FIFOs are in sync. If that is not the case, it will set the filling pointer of the right FIFO to be equal to the filling pointer of the left FIFO.

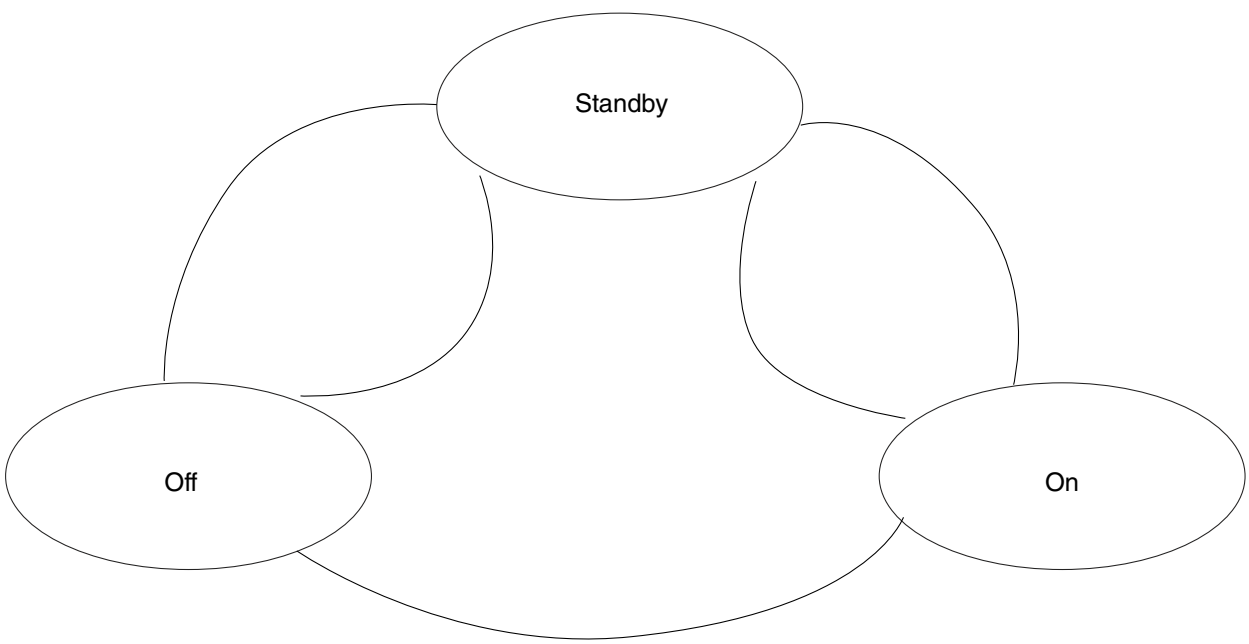


Figure 59-3. FIFO Auto-resync Controller State Machine

The operation is explained from the state diagram shown above. Every FIFO auto-resync controller has a state machine with 3 states: Off, StandBy and On. In the On state, the filling of the left FIFO is compared with the filling of right, and if they are not equal, right is made equal to left, and an interrupt is generated.

The controller will stay in Off state when the feature is disabled. When not disabled, the state machine will go to Off state on any processor read or write to the FIFO. It will go from On or Off to Standby on any left sample read from SPDIF Tx FIFOs, or on any left sample write to SPDIF Rx FIFOs. The controller will go from Standby to On on any right sample read from SPDIF Tx FIFO, or on any right sample write to SPDIF Rx FIFO. There is a control bit in the SPDIFConfig register to enable/disable the feature for the SPDIF Rx FIFO and SPDIF Tx FIFO.

59.4.1.1.1 Application Note

The automatic FIFO resynchronization can be switched on, and will avoid all mismatches between left and right FIFOs, if the software obeys the following rules: 1. When the left data is read or written to the left FIFO, in the same place of the program, data must be read or written to the right FIFO. Maximum time difference between left and right is 1/2 sample clock. (E.g. if sample frequency is 44 Khz, approximately 10 micro-seconds. For 88 Khz, approximately 5 micro-seconds.) 2. Write/read data to FIFO s at least 2 samples at the time. If there is a mismatch Left-Right, the resync logic may go on only 1 sample clock after last data is read/written to the FIFO. Also acceptable is polling the FIFO, if at least part of the time 2 samples will be read/written to it.

- **SPDIF receiver - Additional features**

There are three exceptions associated with the SPDIF Receivers FIFOs

- full
- under/overrun
- resync

When the "full" condition is set for processor data input registers, the processor should read data from the FIFO, before overrun occurs. When "full" is set, and the FIFO contains e.g. 6 samples, it is acceptable for the software to read first 6 samples from the LEFT address, followed by 6 samples from the RIGHT address, or 6 samples from the RIGHT address, followed by 6 samples from the LEFT address, or 1 sample LEFT, followed by 1 sample RIGHT repeated 6 times. There is no order specified.

The implementation for SPDIF Rx is a double FIFO, one for left and one for right. "full" is set when both FIFOs are full. "underrun, overrun" are set when one of the FIFOs do underrun or do overrun. The resync interrupt means hardware took special action to resynchronize left and right FIFOs.

The FIFO level at which the "full" interrupt is generated, is programmable via the Full Select field in the SPDIFConfigReg register.

Rx FIFO on and Rx FIFO reset.

Two additional control fields of the SPDIF Rx FIFO are the on/off select and FIFO reset fields.

If on/off select is set to off, all-zero will be read from the FIFO, irrespective of the data received over the SPDIF interface.

If FIFO reset is set, the FIFO is blocked at "1 sample in FIFO". In this, the full interrupt will be on if FullSelect is set to "00". If FullSelect is set to any other value, interrupt will be off. The other interrupts are always off.

59.4.1.2 Channel Status Reception

A total of 48 channel status bits are received in two registers. No interpretation is performed by the SPDIF receiver block.

Channel Status Bits are ordered first bit left. CS-channel MSB bit "0" is located in bit position 23 in the memory-mapped register SPDIFRxCChannel_h. CS-channel bit "23" is considered the LSB bit 0 in the register. C-channel bit 24 to 47 is seen as [23:0] bits of register SPDIFRxCChannel_l.

59.4.1.2.1 Channel Status Interrupt

When the value of a new SPDIF "CS" channel status frame is loaded in the register, an interrupt is generated. The interrupt is cleared when the processor writes the corresponding bit in the InterruptStat register.

59.4.1.3 User Bit Reception

There are two modes for U Channel reception, CD and non-CD. As is decided by USyncMode (bit 1 of CDText_Control register).

- **Behavior of U Channel receive interface on incoming CD U Channel Sub-code in SPDIF receiver**.

This mode is selected if UsyncMode, bit 1 in register CD Text control is set "1".

The CD sub-code stream embedded into the SPDIF U channel consists of a sequence of packets. Every packet is made up 98 "symbols". The first two symbols of every packet are "sync symbols", the other 96 symbols are "data symbols".

Any sequence found in the SPDIF U channel stream starting with a leading one, followed by 7 information bits, is recognized as a "data symbol". Subsequent data symbols are separated by "pauses". During the "pause", "zero bits" are seen on the SPDIF U channel.

Data symbols are coming in MSB first. The MSB is the leading one.

When a "long pause" is seen between 2 subsequent "data symbols", the SPDIF receiver will assume the reception of one or more "sync symbols". Table below gives details.

Table 59-3. Sync Control Bits

Number of U Channel zero bits	Corresponding number of sync symbols
0-1	Unpredictable, not allowed
2-10	0
11-22	1
23-34	2
35-46	3
>45	Unpredictable, not allowed

The recognition of the number of sync symbols derives from the fact that the U channel transmitter in the CD channel decoder will transmit one symbol on average every 12 SPDIF channel bits. On this average rate, there is a maximum tolerance of 5%.

The SPDIF receiver is tolerant of symbol errors. Due to the physical nature of the transmission of the data over the CD disc, not more than 1 out of any 5 consecutive user channel symbols may be in error. The error may cause a change in data value, which is not detected by this interface, or it may cause a data symbol to be seen as a sync symbol, or a sync symbol to be seen as a data symbol. However, not more than 1 out of any 5 consecutive user channel symbols should be affected in this way.

The SPDIF U channel circuitry recognizes the 98-symbol packet structure, and sends the 96 symbol payload to the processor application. The 96 symbol payload is transmitted to the processor via 2 registers:

- The SPDIFRxUChannel register. In this register, data is presented 3 symbols at the time to the processor. Every time 3 new valid symbols, received on the SPDIF U Channel are present, the UChannelRxFull interrupt is asserted. For one 98-symbol packet, 96 symbols are carried across SPDIFRxUChannel. To transfer all this data, 32 UChannelRxFull interrupts are generated.
- The QChannelReceive register. In this register, only the Q bit of the packet is accumulated.Operation is similar to UChannelReceive. Because only Q-bit is transferred, only 96 Q-bits are transferred for any 98-symbol packet. To transfer this data, 4 QChannelRxFull interrupts are generated. When QChannelRxFull occurs, it is coincident with UChannelRxFull. There is only one QChannelRxFull for every 8 UChannelRxFull. The convention is that most significant data is transmitted first, and is left-aligned in the registers.
- Timing regarding packet boundary is extracted by hardware. The last UChannelRxFull corresponding to a given packet should be coincident with the last

QChannelRxFull. In this last U, Q channel interrupt, symbols 95-98 are received, Q channel bits 67-98. The interrupts are coincident with UQSyncFound, flagging last symbols of the current frame.

- When the start of the new packet is found before the current packet is complete (less than 98 symbols in the packet), the UQFrameError interrupt is set. The application software should read out UChannelReceive and QchannelReceive registers, discard the value, and assume the start of a new packet.
- As already said, packet sync extraction is tolerant for single-symbol errors. Packet sync detection is based on the recognition of the sequence data-sync-sync-data in the symbol stream, because this is the only syncing sequence that is not affected by single errors. If the sync symbols are not found 98 symbols after the previous occurrence, it is assumed to be destroyed by channel error, and a new sync symbols is interpolated.
- Normally, only data bytes are passed to the application software. Every databyte will have its most significant bit set. If sync symbols are passed to the application software, they are seen as all-zero symbols. Sync symbols can only end up in the data stream due to channel error.

- **Behavior of U Channel receive interface on incoming non-CD data**.

This mode is selected if UsyncMode, bit 1 in register CD Text control is set'0'.

In non-CD mode, the SPDIF U channel stream is recognized as a sequence of "data symbols". No packet recognition is done.

Any sequence found in the SPDIF U channel stream starting with a leading one, followed by 7 information bits, is recognized as a "data symbol". Subsequent data symbols are separated by "pauses". During the "pause", "zero bits" are seen on the SPDIF U channel.

3 consecutive data symbols seen in the SPDIF U Channel stream are grouped together into the SPDIFRxUChannel register. First symbol is left, last symbol is right aligned. When SPDIFRxUChannel contains 3 new data symbols, UChannelRxFull is asserted.

In this mode, the operation of QchannelRx and associated interrupt QchannelRxFull is reserved, undefined. And the operation of UQFrameError and UQSyncFound is also reserved, undefined.

The U channel is extracted, and output by the SPDIF Rx on SPDIFRxUChannel-Stream.

When incoming SPDIF data parity error or bit error is detected, and if the next SPDIF word for that channel is error-free, the SPDIF word in error is replaced with the average of the previous word and next word. When incoming SPDIF data parity error or bit error is detected, and the next SPDIF word is in error, the previous SPDIF word is repeated.

59.4.1.4 Validity Flag Reception

An interrupt is associated with the Validity flag. (interrupt 16 - SPDIFValNoGood). This interrupt is set every time a frame is seen on the SPDIF interface with the validity bit set to "invalid".

59.4.1.5 SPDIF Receiver Interrupt Exception Definition

Several SPDIF exceptions can trigger an interrupt.

They are:

- Control Status channel change. Set when SPDIFRxCChannel_l register is updated. The register is updated for every new C-Channel received. The exception is reset on write to InterruptClear register.
- SPDIF Illegal Symbol. Set on reception of illegal symbol during SPDIF receive. Reset by writing register InterruptClear.[1]
- SPDIF bit error. Set on reception of bit error. (Parity bit does not match). Reset on write to InterruptClear register.
- Receive data FIFO full. Set when SPDIF receive data FIFO is full.
- Receive data FIFO underrun/overrun. Set when there is a underrun/overrun on the SPDIF receive data FIFO.
- Receive data FIFO resynchronization. Set when a resynchronization event occurs on the SPDIF receive data FIFO.
- Receive U Channel buffer full. Set when next 24 bits of U channel code are available.
- Receive Q Channel buffer overrun. Set when Q channel buffer overrun.
- Receive U Channel buffer overrun. Set on U channel buffer overrun.
- Receive Q Channel buffer full. Set when next 24 bits of Q channel code are available.
- Receive UQ sync found. Set when UQ channel sync found.
- Receive UQ frame error. Set when UQ frame error found.

1. The SPDIF input is a biphase/mark modulated signal. The time between any two successive transitions of the SPDIF signal is always 1, 2 or 3 SPDIF symbol periods long. The SPDIF receiver will parse the stream, and split it in so-called symbols. It recognizes s1, s2 and s3 symbols, depending on the length of the symbols. Not all sequences of these symbols are allowed. To give an example, a sequence s2-s1-s1-s1-s2 cannot occur in a no-error SPDIF signal. If the receiver finds such an illegal sequence, the illegal symbol interrupt is set. No corrective action is undertaken. When the interrupt occurs, this means that(a) The SPDIF signal is destroyed by noise (b) The SPDIF frequency changed.

59.4.1.6 Standards Compliance

The SPDIF interface is compatible with the Tech 3250-E standard of the European Broadcasting Union, except clause 6.3.3 and the IEC60958-3 Ed2 for relevant topics.

Supported input frequency range is 12 Khz up to 96 Khz. (fully compliant) and 96 Khz up to 176 Khz (Can interface with compliant SPDIF transmitter within same cabinet, making reasonable assumptions on jitter added due to interconnecting wire.)

Tolerated jitter on SPDIF input signals are 0.25 bit peak-peak for high frequencies. There is no jitter limit for low frequencies. The user channel extraction in CD mode is capable of coping with single-symbol errors, and still retrieve U channel frames on correct boundaries. This capability is required for reliable reception of CD-Text from some Philips CD channel decoders. This capability was deemed more important than compliance with the IEC60958 annex A.3 standard, and for this reason user channel reception is not compliant with IEC60958 annex A.3. However, the interface is capable to receive U channel inserted by a typical CD channel decoder. Also, in this case, it is more robust and tolerant for channel error than what is required by IEC60958 annex A.3.

59.4.1.7 SPDIF PLOCK Detection and Rxclk Output

Using the high speed system clock, the internal DPLL can extract the bit clock (advanced pulse) from the input bitstream. When this internal DPLL is locked, the LOCK bit of PhaseConfig Register will be set, and the SPDIF Lock output pin SPDIF_LOCK will be asserted.

After DPLL has locked, the pulses are generated, and the average pulse rate is 128 x the sampling frequency.(For a 44.1 Khz input sampling frequency, the average pulse rate = 128 x 44.1 Khz.) The pulse signal is used in the FreqMeas circuit to generate the frequency measurement result.

59.4.1.8 Measuring Frequency of SPDIF_RxClk

The internal DPLL can extract the bit clock (advanced plus) from the input bitstream. To do that, it is necessary to measure the frequency of the incoming signal in relationship with the system clock (BUS_CLK).

Associated with it are two registers, PhaseConfig and FreqMeas. The circuit will measure the frequency of the incoming clock as a function of the BUS_CLK. The circuit is a second-order filter. The output is a value represented by an unsigned number stored in the 24-bit FreqMeas register, giving the frequency of the source as a function of the BUS_CLK.

FreqMeas[23:0] = FreqMeas_CLK / BUS_CLK * 2^{10} * GAIN.

For example, if the GAIN is selected as 8*(2**10) (PhaseConfig[5:3] = 3'b011), the actual result

FreqMeas_CLK / BUS_CLK is equal to FreqMeas[23:0] / 2^{23}.

59.4.2 SPDIF Transmitter

Audio data for the SPDIF transmitter is provided by processor via the SPDIFTxLeft and SPDIFTxRight registers.

Clocking for SPDIF transmitter is selected through a multiplexer from several clock sources (see TxClk_Source for clock source inputs). The SPDIF transmitter clock source can be divided down as needed using Txclk_DF. The SPDIF transmitter output can be chosen from either the SPDIF transmitter block, directly from the SPDIF receiver (via the output multiplexer), or disabled.

The SPDIF transmitter generates a SPDIF output bitstream in IEC60958 biphase mark format, consisting of audio data, channel status.

59.4.2.1 Audio Data Transmission

Audio data for the SPDIF transmitter is provided by the processor via SPDIFTxLeft and SPDIFTxRight registers. They send audio data to Tx left and right FIFOs. The Tx left and right FIFOs are also 16-deep and 24-width (equal to the audio data width).

- **SPDIF transmitter data registers - Behavior on overrun, underrun**

The SPDIF Data Transmit registers (SPDIFTxLeft and SPDIFTxRight) have individual FIFOs for left and right channel. As a result, there is always the possibility that left and right FIFOs may go out of sync due to FIFO underruns and FIFO overruns that affect only one part (left or right) of any FIFO. To prevent this from happening, hardware has been added on the device. Two mechanisms to prevent mismatch between the FIFOs are available.

If SPDIF Tx FIFO underruns on the right half of the FIFO, no sample leaves that FIFO (because it was already empty). Special hardware will make sure that the next sample read from the left FIFO will not leave the FIFO (no read strobe is generated). If the underrun occurs on the left half of the FIFO, next read strobe to the right FIFO is blocked.

- **SPDIF transmitter data registers - Automatic resynchronization of FIFOs**

See Audio Data Reception.

- **SPDIFTxLeft, SPDIFTxRight details**

With SPDIF Tx FIFOs three exceptions are associated.

- empty
- under/overrun
- resync

When the empty condition is set for processor data output registers, the processor should write data to the FIFO, before underrun occurs. When empty is set and, for instance, 6 samples need to be written, it is acceptable for the software to write first 6 samples from the LEFT address, followed by 6 samples from the RIGHT address, or 1 sample LEFT, followed by 1 sample RIGHT repeated 6 times. Left should be written before right. The implementation of all data out FIFOs is a double FIFO, one for left and one for right. Empty is set when both FIFOs are empty. Underrun, overrun are set when one of the FIFOs do underrun or do overrun. Resync is set when the hardware resynchronizes left and right FIFOs.

On receiving underrun, overrun interrupt, synchronization between Left and Right words in the FIFOs may be lost. Synchronization will not be lost when the underrun or overrun comes from the IEC60958 side of the FIFO. If the processor reads or writes more data from, for example, left than from right, synchronization will be lost. If automatic resynchronization is enabled, and if the software obeys the rules to let this work, resynchronization will be automatic.

59.4.2.2 Channel Status Transmission

A total of 48 Consumer channel status bits are transmitted from two registers. Channel Status Bits are ordered first bit left.

CS-channel MSB bit "0" is located in bit position 23 in the memory-mapped register SPDIFTxCChannelCons_h. CS-channel bit "23" is considered bit 0 in the register. C-channel bits 24-47 are seen as MSB-LSB bits of register SPDIFTxCChannelCons_l.

59.4.2.3 Validity Flag Transmission

The validity bit setting is performed via bit 5 of the SPDIF_SCR register.

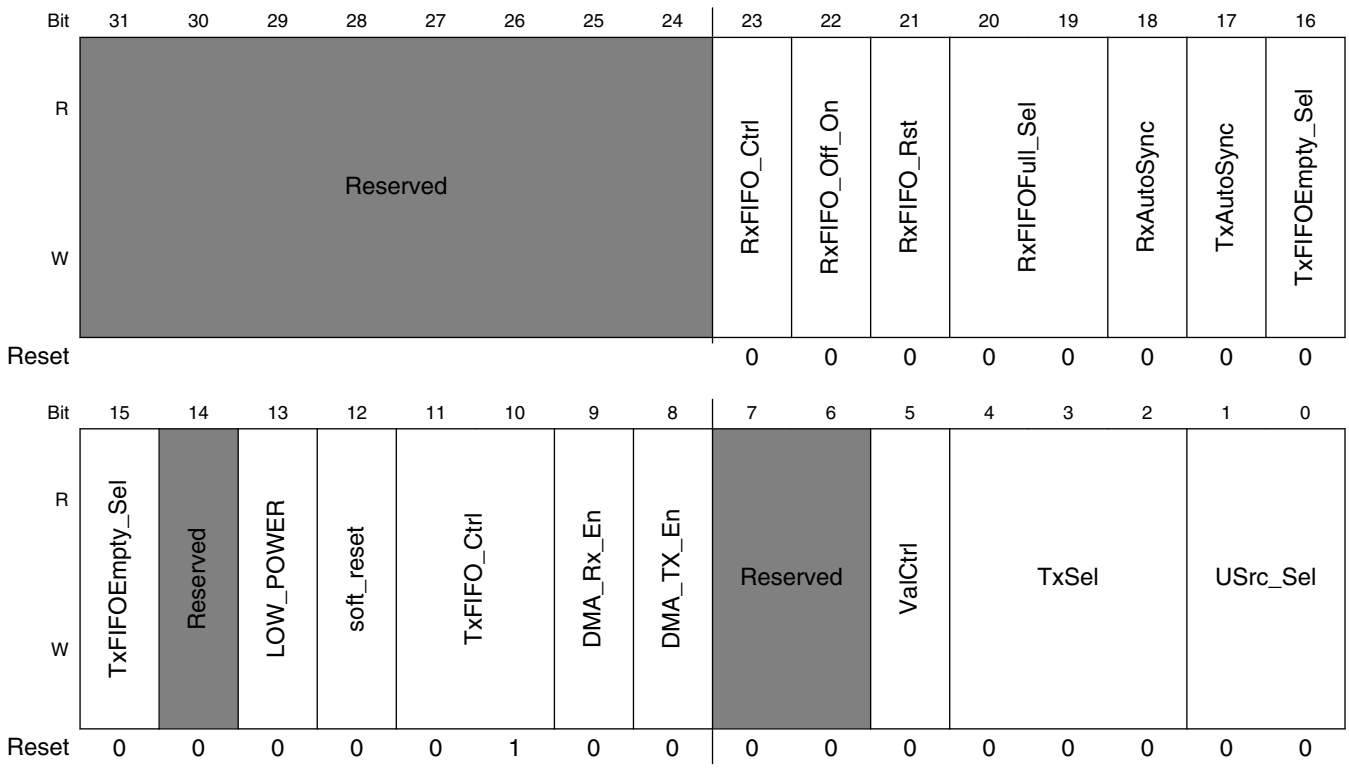
59.5 SPDIF Memory Map/Register Definition

SPDIF memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
200_4000	SPDIF Configuration Register (SPDIF_SCR)	32	R/W	0000_0400h	59.5.1/5000
200_4004	CDText Control Register (SPDIF_SRCD)	32	R/W	0000_0000h	59.5.2/5002
200_4008	PhaseConfig Register (SPDIF_SRPC)	32	R/W	0000_0000h	59.5.3/5003
200_400C	InterruptEn Register (SPDIF_SIE)	32	R/W	0000_0000h	59.5.4/5004
200_4010	InterruptStat Register (SPDIF_SIS)	32	R	0000_0002h	59.5.5/5006
200_4010	InterruptClear Register (SPDIF_SIC)	32	W	0000_0000h	59.5.6/5008
200_4014	SPDIFRxLeft Register (SPDIF_SRL)	32	R	0000_0000h	59.5.7/5009
200_4018	SPDIFRxRight Register (SPDIF_SRR)	32	R	0000_0000h	59.5.8/5010
200_401C	SPDIFRxCChannel_h Register (SPDIF_SRCSH)	32	R	0000_0000h	59.5.9/5010
200_4020	SPDIFRxCChannel_l Register (SPDIF_SRCSL)	32	R	0000_0000h	59.5.10/ 5011
200_4024	UchannelRx Register (SPDIF_SRU)	32	R	0000_0000h	59.5.11/ 5011
200_4028	QchannelRx Register (SPDIF_SRQ)	32	R	0000_0000h	59.5.12/ 5012
200_402C	SPDIFTxLeft Register (SPDIF_STL)	32	W	0000_0000h	59.5.13/ 5012
200_4030	SPDIFTxRight Register (SPDIF_STR)	32	W	0000_0000h	59.5.14/ 5013
200_4034	SPDIFTxCChannelCons_h Register (SPDIF_STCSCH)	32	R/W	0000_0000h	59.5.15/ 5013
200_4038	SPDIFTxCChannelCons_l Register (SPDIF_STCSCL)	32	R/W	0000_0000h	59.5.16/ 5014
200_4044	FreqMeas Register (SPDIF_SRFM)	32	R	0000_0000h	59.5.17/ 5014
200_4050	SPDIFTxClk Register (SPDIF_STC)	32	R/W	0002_0F00h	59.5.18/ 5015

59.5.1 SPDIF Configuration Register (SPDIF_SCR)

Address: 200_4000h base + 0h offset = 200_4000h



SPDIF_SCR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23 RxFIFO_Ctrl	0 Normal operation 1 Always read zero from Rx data register
22 RxFIFO_Off_On	0 SPDIF Rx FIFO is on 1 SPDIF Rx FIFO is off. Does not accept data from interface
21 RxFIFO_Rst	0 Normal operation 1 Reset register to 1 sample remaining
20–19 RxFIFOFull_Sel	00 Full interrupt if at least 1 sample in Rx left and right FIFOs 01 Full interrupt if at least 4 sample in Rx left and right FIFOs 10 Full interrupt if at least 8 sample in Rx left and right FIFOs 11 Full interrupt if at least 16 sample in Rx left and right FIFO
18 RxAutoSync	0 Rx FIFO auto sync off 1 RxFIFO auto sync on
17 TxAutoSync	0 Tx FIFO auto sync off 1 Tx FIFO auto sync on

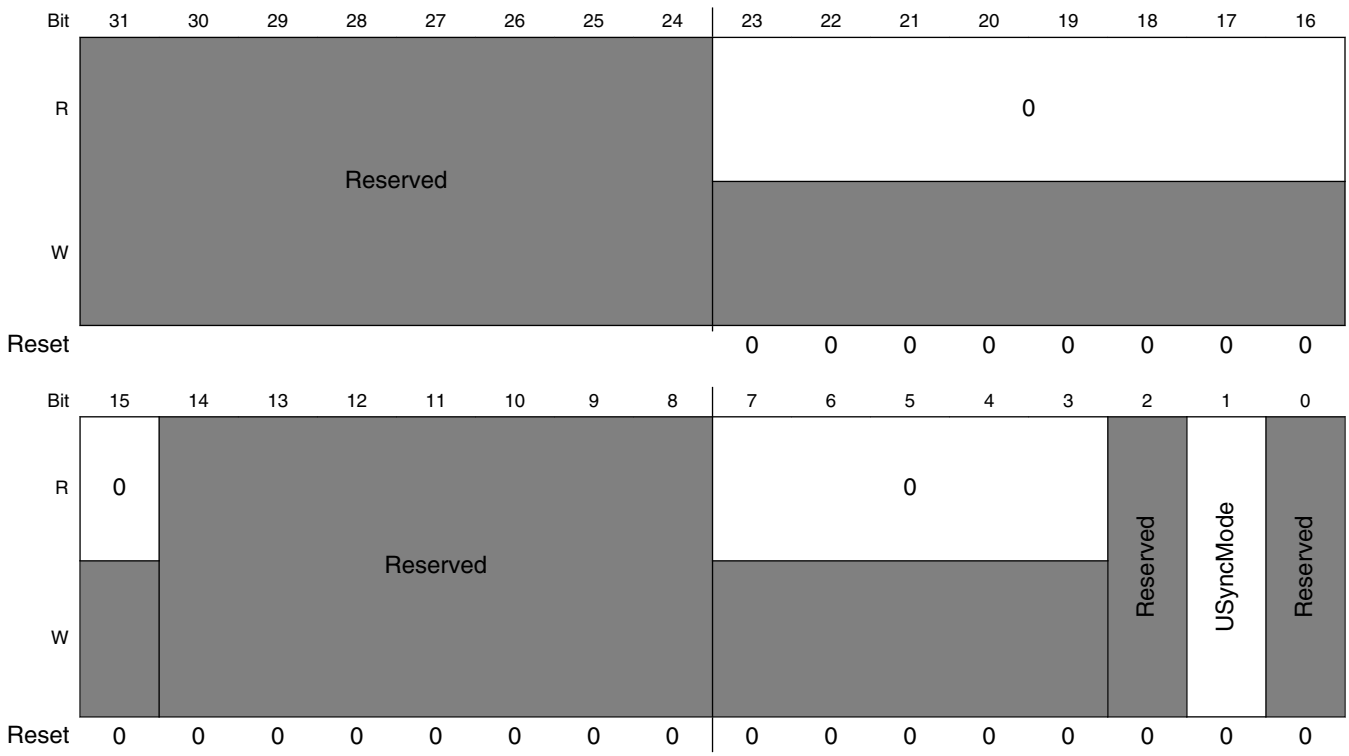
Table continues on the next page...

SPDIF_SCR field descriptions (continued)

Field	Description
16–15 TxFIFOEmpty_Sel	00 Empty interrupt if 0 sample in Tx left and right FIFOs 01 Empty interrupt if at most 4 sample in Tx left and right FIFOs 10 Empty interrupt if at most 8 sample in Tx left and right FIFOs 11 Empty interrupt if at most 12 sample in Tx left and right FIFOs
14 -	This field is reserved. Reserved
13 LOW_POWER	When write 1 to this bit, it will cause SPDIF enter low-power mode. return 1 when SPDIF in Low-Power mode.
12 soft_reset	When write 1 to this bit, it will cause SPDIF software reset. The software reset will last 8 cycles. When in the reset process, return 1 when read. else return 0 when read.
11–10 TxFIFO_Ctrl	00 Send out digital zero on SPDIF Tx 01 Tx Normal operation 10 Reset to 1 sample remaining 11 Reserved
9 DMA_Rx_En	DMA Receive Request Enable (RX FIFO full)
8 DMA_TX_En	DMA Transmit Request Enable (Tx FIFO empty)
7–6 -	This field is reserved. Reserved
5 ValCtrl	0 Outgoing Validity always set 1 Outgoing Validity always clear
4–2 TxSel	000 Off and output 0 001 Feed-through SPDIFIN 101 Tx Normal operation Others Reserved
USrc_Sel	00 No embedded U channel 01 U channel from SPDIF receive block (CD mode) 10 Reserved 11 U channel from on chip transmitter

59.5.2 CDText Control Register (SPDIF_SRCD)

Address: 200_4000h base + 4h offset = 200_4004h

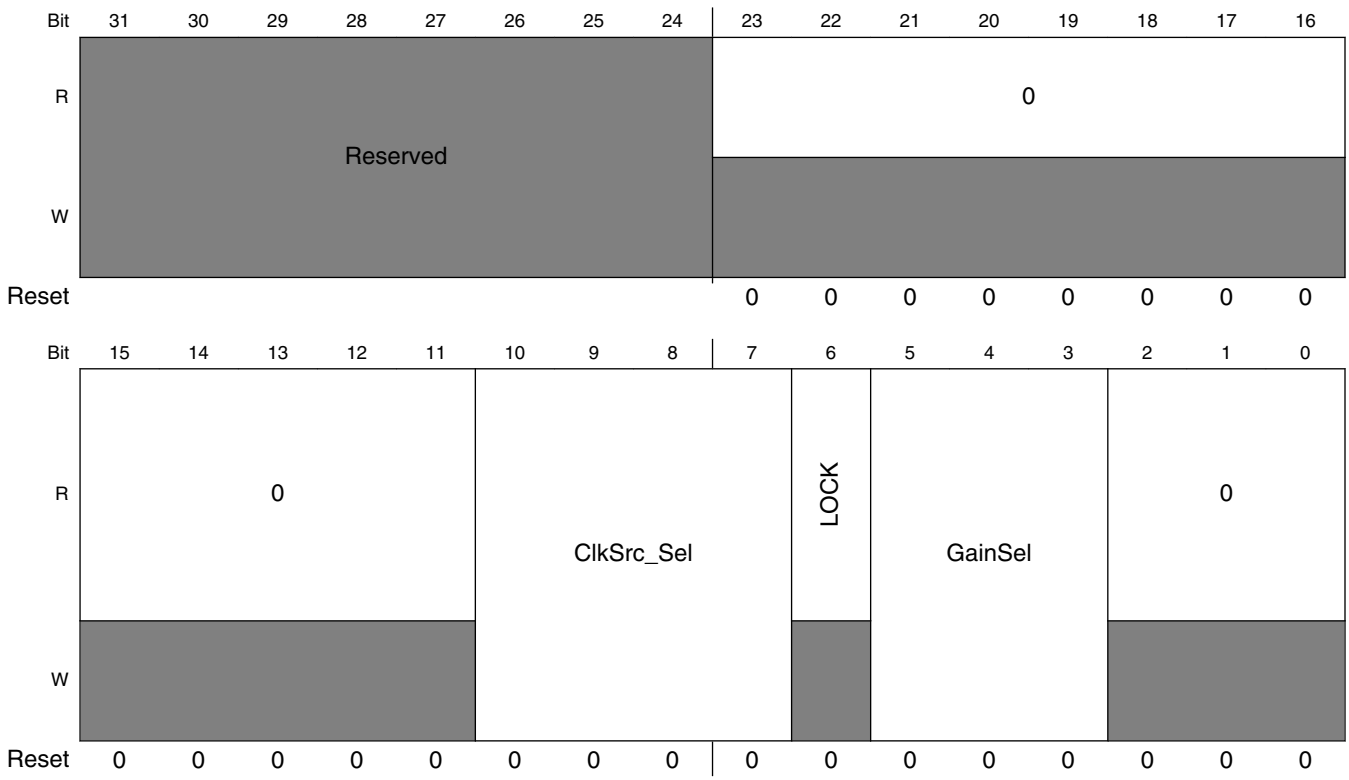


SPDIF_SRCD field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–15 Reserved	This read-only field is reserved and always has the value 0.
14–8 -	This field is reserved. Reserved. set to zero.
7–3 Reserved	This read-only field is reserved and always has the value 0.
2 -	This field is reserved. Reserved.
1 USyncMode	0 Non-CD data 1 CD user channel subcode
0 -	This field is reserved. Reserved.

59.5.3 PhaseConfig Register (SPDIF_SRPC)

Address: 200_4000h base + 8h offset = 200_4008h



SPDIF_SRPC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–11 Reserved	This read-only field is reserved and always has the value 0.
10–7 ClkSrc_Sel	Clock source selection, all other settings not shown are reserved: 0000 if (DPLL Locked) SPDIF_RxClk else REF_CLK_32K (XTALOSC) 0001 if (DPLL Locked) SPDIF_RxClk else tx_clk (SPDIF0_CLK_ROOT) 0010 if (DPLL Locked) SPDIF_RxClk else ASRC_EXT_CLK 0011 if (DPLL Locked) SPDIF_RxClk else SPDIF_EXT_CLK 0100 if (DPLL Locked) SPDIF_Rxclk else ESAI_HCKT 0101 REF_CLK_32K (XTALOSC) 0110 tx_clk (SPDIF0_CLK_ROOT) 0111 ASRC_CLK 1000 SPDIF_EXT_CLK 1001 ESAI_HCKT 1010 if (DPLL Locked) SPDIF_RxClk else MLB Clock

Table continues on the next page...

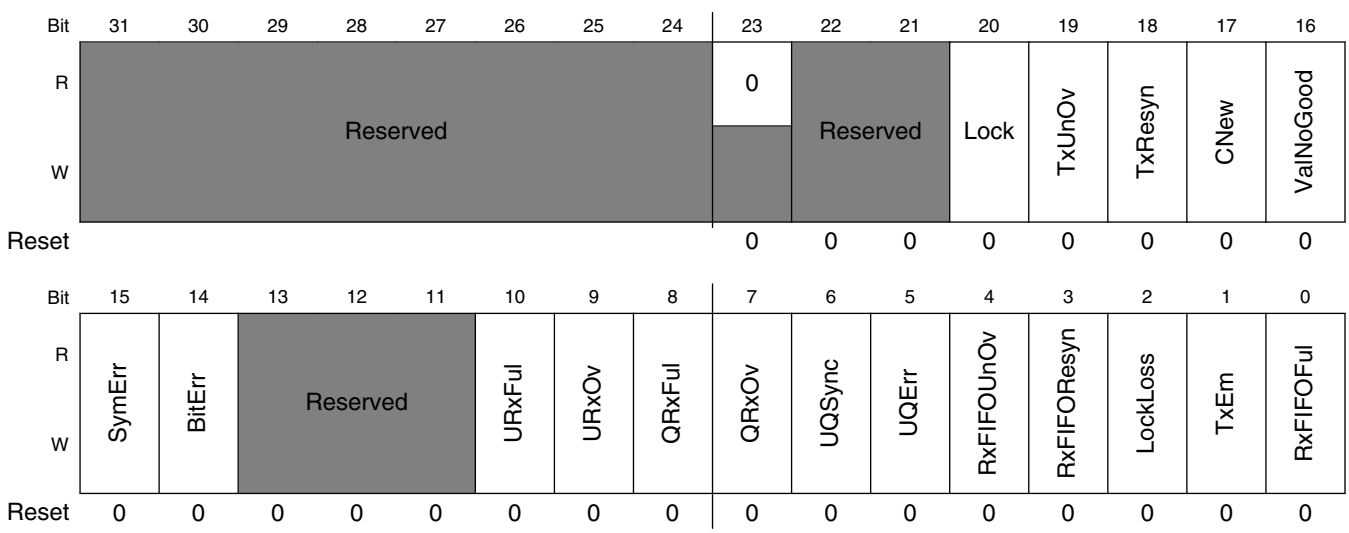
SPDIF_SRPC field descriptions (continued)

Field	Description
	1011 if (DPLL Locked) SPDIF_RxClk else MLB PHY Clock 1100 MLB Clock 1101 MLB PHY Clock
6 LOCK	LOCK bit to show that the internal DPLL is locked, read only
5–3 GainSel	Gain selection: 000 24*(2**10) 001 16*(2**10) 010 12*(2**10) 011 8*(2**10) 100 6*(2**10) 101 4*(2**10) 110 3*(2**10)
Reserved	This read-only field is reserved and always has the value 0.

59.5.4 InterruptEn Register (SPDIF_SIE)

The InterruptEn register (SPDIF_SIE) provides control over the enabling of interrupts.

Address: 200_4000h base + Ch offset = 200_400Ch



SPDIF_SIE field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.

Table continues on the next page...

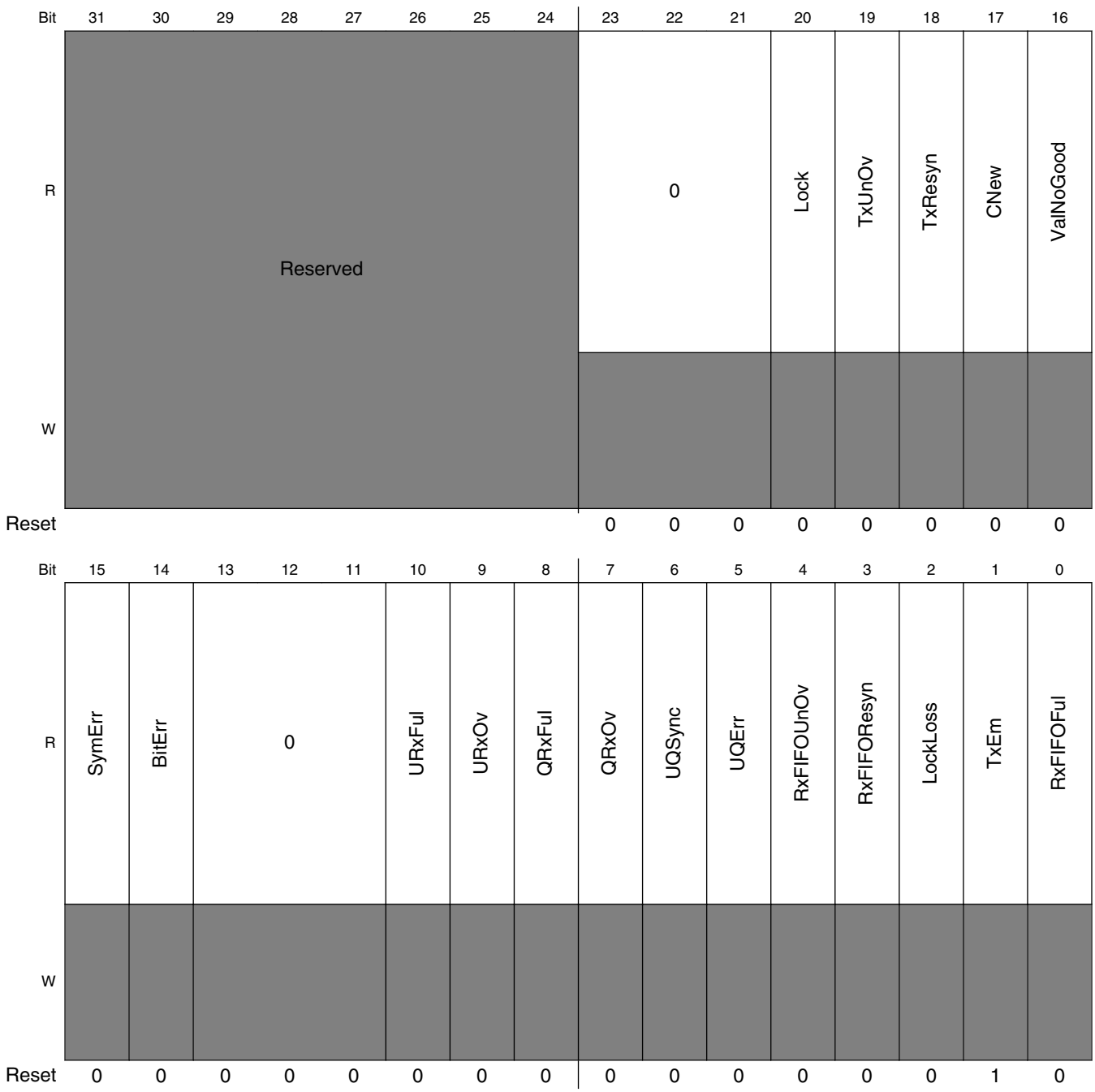
SPDIF_SIE field descriptions (continued)

Field	Description
	This field is reserved.
23 Reserved	This read-only field is reserved and always has the value 0.
22–21 -	This field is reserved. Reserved. set to zero.
20 Lock	SPDIF receiver's DPLL is locked
19 TxUnOv	SPDIF Tx FIFO under/overrun
18 TxResyn	SPDIF Tx FIFO resync
17 CNew	SPDIF receive change in value of control channel
16 ValNoGood	SPDIF validity flag no good
15 SymErr	SPDIF receiver found illegal symbol
14 BitErr	SPDIF receiver found parity bit error
13–11 -	This field is reserved. Reserved. set to zero.
10 URxFul	U Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from U Rx reg.
9 URxOv	U Channel receive register overrun
8 QRxFul	Q Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from Q Rx reg.
7 QRxOv	Q Channel receive register overrun
6 UQSync	U/Q Channel sync found
5 UQErr	U/Q Channel framing error
4 RxFIFOUnOv	Rx FIFO underrun/overrun
3 RxFIFOResyn	Rx FIFO resync
2 LockLoss	SPDIF receiver loss of lock
1 TxEm	SPDIF Tx FIFO empty, can't be cleared with reg. IntClear. To clear it, write toTx FIFO.
0 RxFIFOFul	SPDIF Rx FIFO full, can't be cleared with reg. IntClear. To clear it, read from Rx FIFO.

59.5.5 InterruptStat Register (SPDIF_SIS)

The InterruptStat (SPDIF_SIS) register is a read only register that provides the status on interrupt operations.

Address: 200_4000h base + 10h offset = 200_4010h



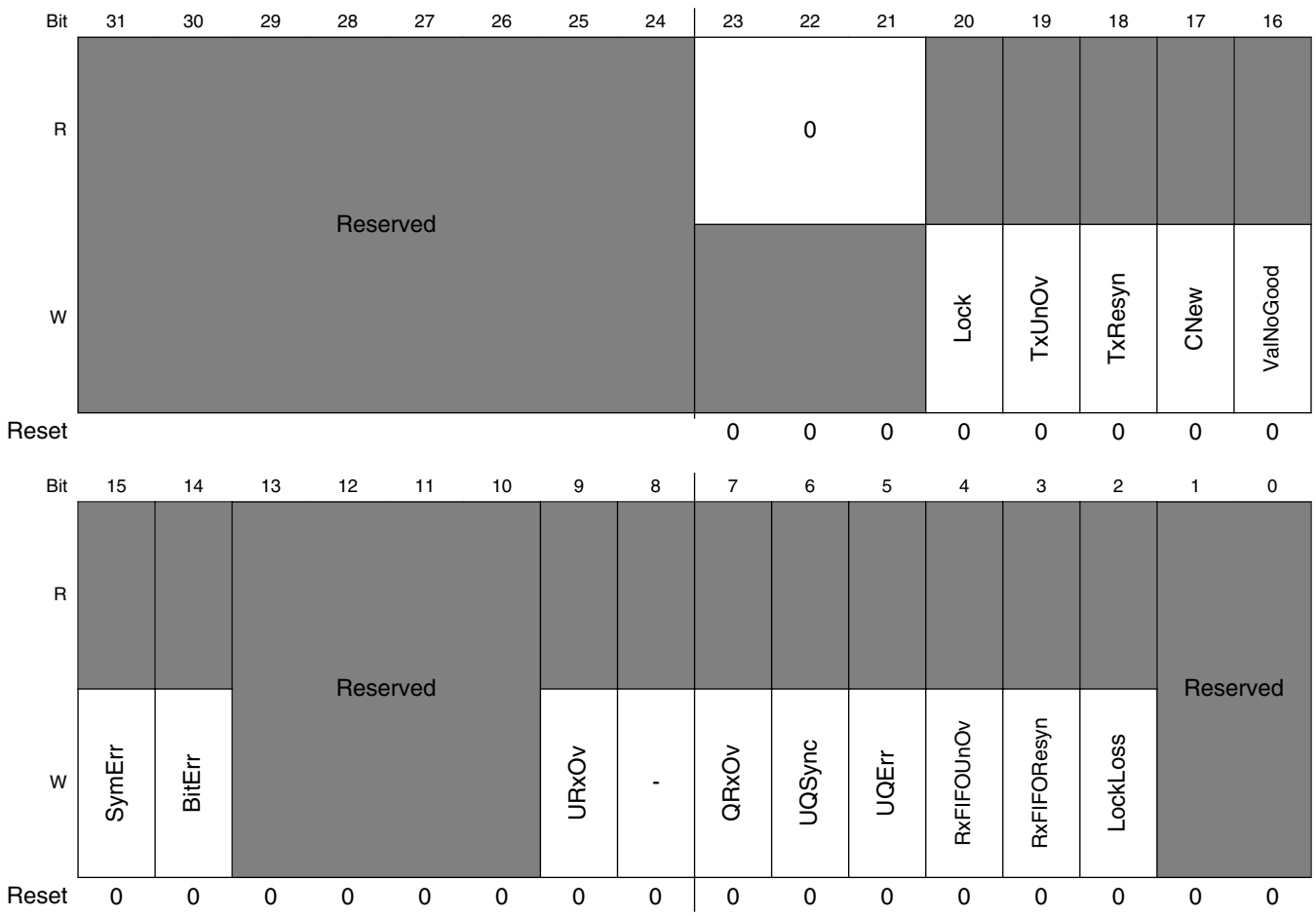
SPDIF_SIS field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–21 Reserved	This read-only field is reserved and always has the value 0.
20 Lock	SPDIF receiver's DPLL is locked
19 TxUnOv	SPDIF Tx FIFO under/overrun
18 TxResyn	SPDIF Tx FIFO resync
17 CNew	SPDIF receive change in value of control channel
16 ValNoGood	SPDIF validity flag no good
15 SymErr	SPDIF receiver found illegal symbol
14 BitErr	SPDIF receiver found parity bit error
13–11 Reserved	This read-only field is reserved and always has the value 0.
10 URxFul	U Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from U Rx reg.
9 URxOv	U Channel receive register overrun
8 QRxFul	Q Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from Q Rx reg.
7 QRxOv	Q Channel receive register overrun
6 UQSync	U/Q Channel sync found
5 UQErr	U/Q Channel framing error
4 RxFIFOUnOv	Rx FIFO underrun/overrun
3 RxFIFOResyn	Rx FIFO resync
2 LockLoss	SPDIF receiver loss of lock
1 TxEm	SPDIF Tx FIFO empty, can't be cleared with reg. IntClear. To clear it, write toTx FIFO.
0 RxFIFOFul	SPDIF Rx FIFO full, can't be cleared with reg. IntClear. To clear it, read from Rx FIFO.

59.5.6 InterruptClear Register (SPDIF_SIC)

The InterruptClear (SPDIF_SIC) register is a write only register and is used to clear interrupts.

Address: 200_4000h base + 10h offset = 200_4010h



SPDIF_SIC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–21 Reserved	This read-only field is reserved and always has the value 0.
20 Lock	SPDIF receiver's DPLL is locked
19 TxUnOv	SPDIF Tx FIFO under/overrun

Table continues on the next page...

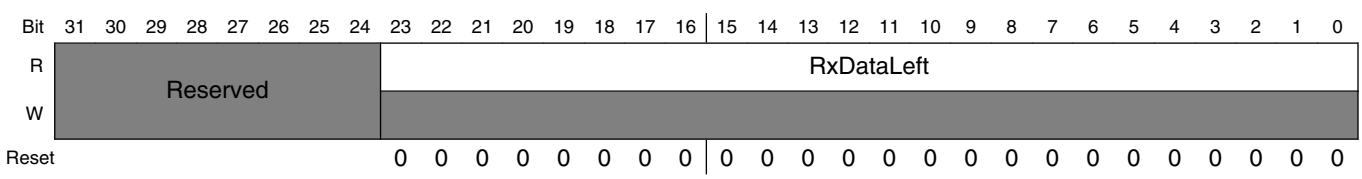
SPDIF_SIC field descriptions (continued)

Field	Description
18 TxResyn	SPDIF Tx FIFO resync
17 CNew	SPDIF receive change in value of control channel
16 ValNoGood	SPDIF validity flag no good
15 SymErr	SPDIF receiver found illegal symbol
14 BitErr	SPDIF receiver found parity bit error
13–10 -	This field is reserved. Reserved.
9 URxOv	U Channel receive register overrun
8 -	Reserved
7 QRxOv	Q Channel receive register overrun
6 UQSync	U/Q Channel sync found
5 UQErr	U/Q Channel framing error
4 RxFIFOUnOv	Rx FIFO underrun/overrun
3 RxFIFOResyn	Rx FIFO resync
2 LockLoss	SPDIF receiver loss of lock
-	This field is reserved. Reserved.

59.5.7 SPDIFRxLeft Register (SPDIF_SRL)

SPDIFRxLeft register is an audio data reception register.

Address: 200_4000h base + 14h offset = 200_4014h



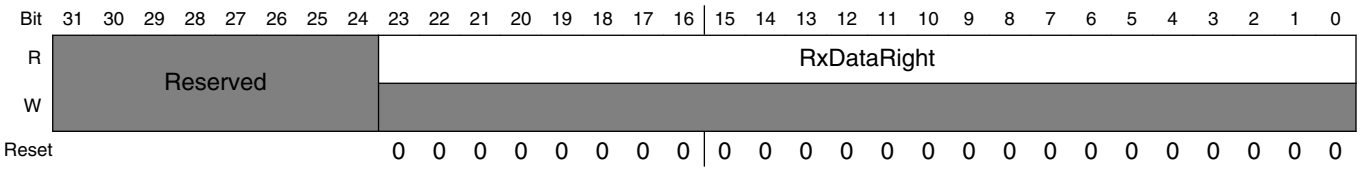
SPDIF_SRL field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
RxDataLeft	Processor receive SPDIF data left

59.5.8 SPDIFRxRight Register (SPDIF_SRR)

SPDIFRxRight register is an audio data reception register.

Address: 200_4000h base + 18h offset = 200_4018h



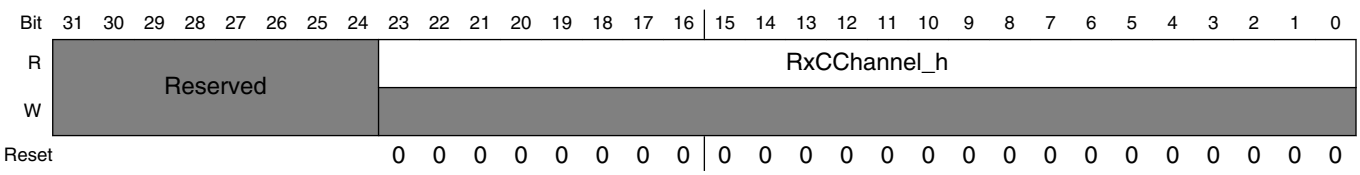
SPDIF_SRR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
RxDataRight	Processor receive SPDIF data right

59.5.9 SPDIFRxCChannel_h Register (SPDIF_SRCSH)

SPDIFRxCChannel_h register is a channel status reception register.

Address: 200_4000h base + 1Ch offset = 200_401Ch



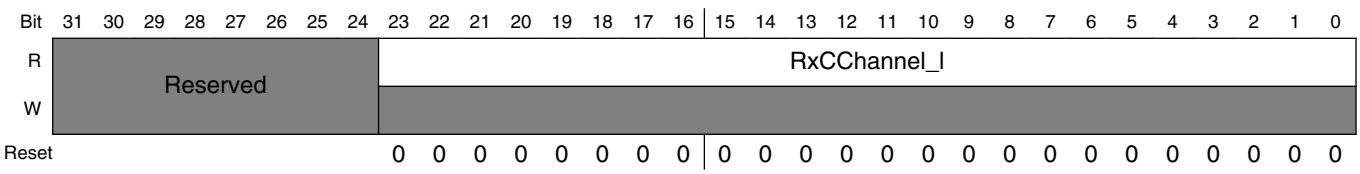
SPDIF_SRCSH field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
RxCChannel_h	SPDIF receive C channel register, contains first 24 bits of C channel without interpretation

59.5.10 SPDIFRxCChannel_l Register (SPDIF_SRCSL)

SPDIFRxCChannel_l register is a channel status reception register.

Address: 200_4000h base + 20h offset = 200_4020h



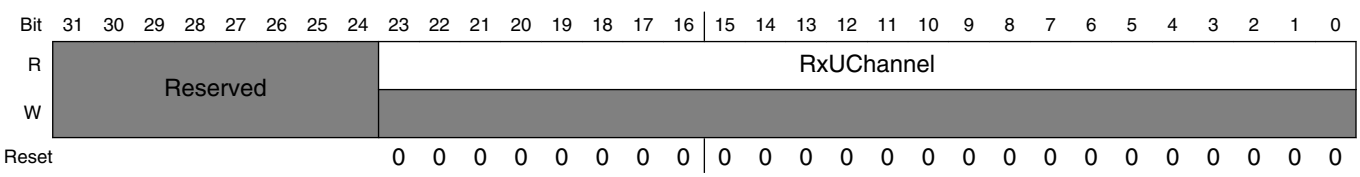
SPDIF_SRCSL field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
RxCChannel_l	SPDIF receive C channel register, contains next 24 bits of C channel without interpretation

59.5.11 UchannelRx Register (SPDIF_SRU)

UChannelRx register is a user bits reception register.

Address: 200_4000h base + 24h offset = 200_4024h



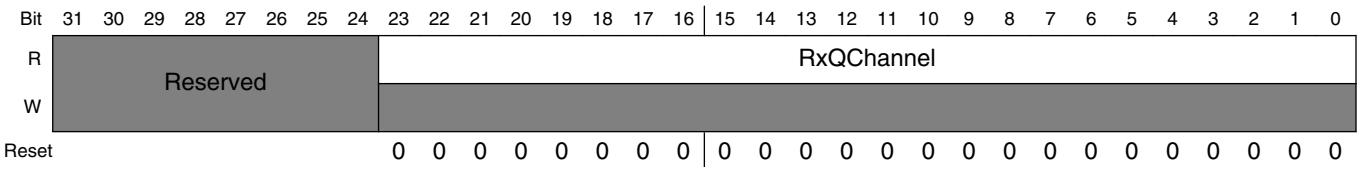
SPDIF_SRU field descriptions

Field	Description
31–24 [unimplemented]	This field is reserved. This is a 24-bit register the upper byte is unimplemented.
RxUChannel	SPDIF receive U channel register, contains next 3 U channel bytes

59.5.12 QchannelRx Register (SPDIF_SRQ)

QChannelRx register is a user bits reception register.

Address: 200_4000h base + 28h offset = 200_4028h



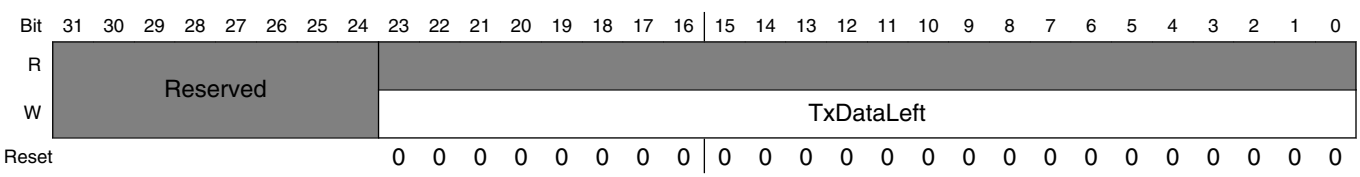
SPDIF_SRQ field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
RxQChannel	SPDIF receive Q channel register, contains next 3 Q channel bytes

59.5.13 SPDIFTxLeft Register (SPDIF_STL)

SPDIFTxLeft register is an audio data transmission register.

Address: 200_4000h base + 2Ch offset = 200_402Ch



SPDIF_STL field descriptions

Field	Description
31–24 [unimplemented]	This field is reserved. This is a 24-bit register the upper byte is unimplemented.
TxDataLeft	SPDIF transmit left channel data. It is write-only, and always returns zeros when read

59.5.14 SPDIFTxRight Register (SPDIF_STR)

SPDIFTxRight register is an audio data transmission register.

Address: 200_4000h base + 30h offset = 200_4030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TxDataRight																							
W																																
Reset									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIF_STR field descriptions

Field	Description
31–24 [unimplemented]	This field is reserved. This is a 24-bit register the upper byte is unimplemented.
TxDataRight	SPDIF transmit right channel data. It is write-only, and always returns zeros when read

59.5.15 SPDIFTxCChannelCons_h Register (SPDIF_STCSCH)

SPDIFTxCChannelCons_h register is a channel status transmission register.

Address: 200_4000h base + 34h offset = 200_4034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TxCChannelCons_h																							
Reset									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIF_STCSCH field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.

Table continues on the next page...

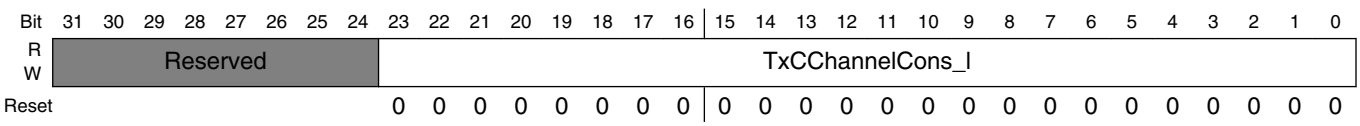
SPDIF_STCSCH field descriptions (continued)

Field	Description
TxCChannelCons_h	SPDIF transmit Cons. C channel data, contains first 24 bits without interpretation. When read, it returns the latest data written by the processor

59.5.16 SPDIFTxCChannelCons_l Register (SPDIF_STCSCL)

SPDIFTxCChannelCons_l register is a channel status transmission register.

Address: 200_4000h base + 38h offset = 200_4038h

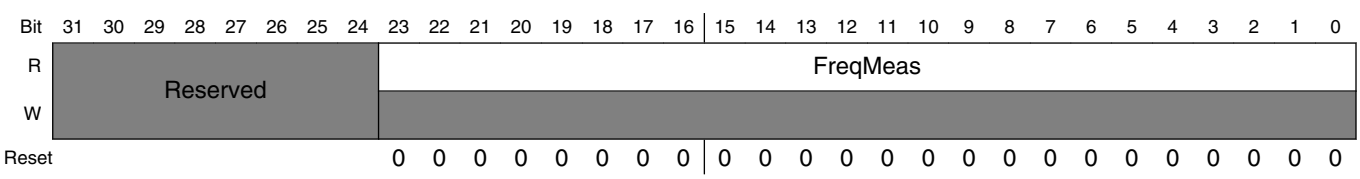


SPDIF_STCSCL field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
TxCChannelCons_l	SPDIF transmit Cons. C channel data, contains next 24 bits without interpretation. When read, it returns the latest data written by the processor

59.5.17 FreqMeas Register (SPDIF_SRFM)

Address: 200_4000h base + 44h offset = 200_4044h



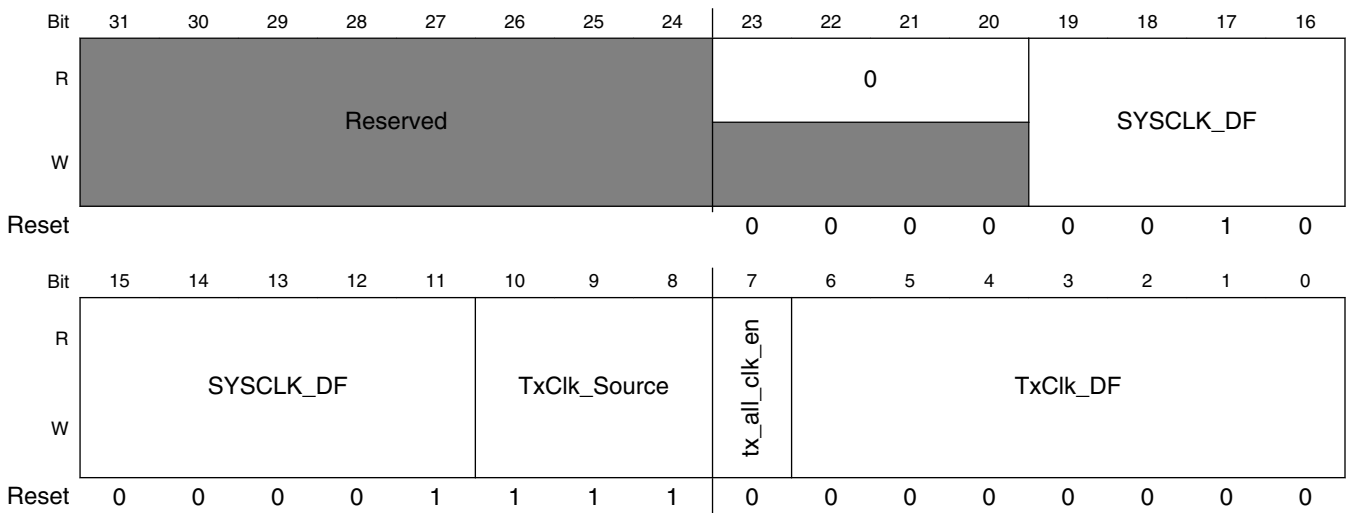
SPDIF_SRFM field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
FreqMeas	Frequency measurement data

59.5.18 SPDIFTxClk Register (SPDIF_STC)

The SPDIFTxClk Control register includes the means to select the transmit clock and frequency division.

Address: 200_4000h base + 50h offset = 200_4050h



SPDIF_STC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–11 SYSCLK_DF	system clock divider factor, 2~512. 0 no clock signal 1 divider factor is 2 ... ... 511 divider factor is 512
10–8 TxClk_Source	000 REF_CLK_32K input (XTALOSC 32 kHz clock) 001 tx_clk input (from SPDIF0_CLK_ROOT. See CCM.) 010 ASRC_EXT_CLK input 011 SPDIF_EXT_CLK, from pads 100 ESAI_HCKT input 101 ipg_clk input (frequency divided) 110 MLB clock input 111 MLB PHY clock input
7 tx_all_clk_en	Spdif transfer clock enable. When data is going to be transfered, this bit should be set to1.

Table continues on the next page...

SPDIF_STC field descriptions (continued)

Field	Description
	0 disable transfer clock. 1 enable transfer clock.
TxClk_DF	Divider factor (1-128) 0 divider factor is 1 1 divider factor is 2 ... ... 127 divider factor is 128

Chapter 60
System Reset Controller (SRC)

60.1 SRC Overview

The System Reset Controller (SRC) controls the reset and boot operation of the SoC.

It is responsible for the generation of all reset signals and boot decoding.

The reset controller determines the source and the type of reset, such as POR, WARM, COLD, and performs the necessary reset qualification and stretching sequences. Based on the type of reset, the reset logic generates the reset sequence for the entire IC. Whenever the chip is powered on, the reset is issued through SRC_ONOFF signal and the entire chip is reset.

60.1.1 Features

The SRC includes the following features.

- Receives and handles the resets from all the reset sources
- Resets the appropriate domains based upon the resets sources and the nature of the reset
- Latches the SRC_BOOT_MODE pins and common configuration signals from the internal fuse

60.2 External Signals

The following table describes the external signals of SRC.

Table 60-1. SRC External Signals

Signal	Description	Pad	Mode	Direction
SRC_BOOT_CFG00	Boot configuration signal	EIM_DA0	ALT7	I
SRC_BOOT_CFG01	Boot configuration signal	EIM_DA1	ALT7	I
SRC_BOOT_CFG02	Boot configuration signal	EIM_DA2	ALT7	I
SRC_BOOT_CFG03	Boot configuration signal	EIM_DA3	ALT7	I
SRC_BOOT_CFG04	Boot configuration signal	EIM_DA4	ALT7	I
SRC_BOOT_CFG05	Boot configuration signal	EIM_DA5	ALT7	I
SRC_BOOT_CFG06	Boot configuration signal	EIM_DA6	ALT7	I
SRC_BOOT_CFG07	Boot configuration signal	EIM_DA7	ALT7	I
SRC_BOOT_CFG08	Boot configuration signal	EIM_DA8	ALT7	I
SRC_BOOT_CFG09	Boot configuration signal	EIM_DA9	ALT7	I
SRC_BOOT_CFG10	Boot configuration signal	EIM_DA10	ALT7	I
SRC_BOOT_CFG11	Boot configuration signal	EIM_DA11	ALT7	I
SRC_BOOT_CFG12	Boot configuration signal	EIM_DA12	ALT7	I
SRC_BOOT_CFG13	Boot configuration signal	EIM_DA13	ALT7	I
SRC_BOOT_CFG14	Boot configuration signal	EIM_DA14	ALT7	I
SRC_BOOT_CFG15	Boot configuration signal	EIM_DA15	ALT7	I
SRC_BOOT_CFG16	Boot configuration signal	EIM_A16	ALT7	I
SRC_BOOT_CFG17	Boot configuration signal	EIM_A17	ALT7	I
SRC_BOOT_CFG18	Boot configuration signal	EIM_A18	ALT7	I
SRC_BOOT_CFG19	Boot configuration signal	EIM_A19	ALT7	I
SRC_BOOT_CFG20	Boot configuration signal	EIM_A20	ALT7	I
SRC_BOOT_CFG21	Boot configuration signal	EIM_A21	ALT7	I
SRC_BOOT_CFG22	Boot configuration signal	EIM_A22	ALT7	I
SRC_BOOT_CFG23	Boot configuration signal	EIM_A23	ALT7	I
SRC_BOOT_CFG24	Boot configuration signal	EIM_A24	ALT7	I
SRC_BOOT_CFG25	Boot configuration signal	EIM_WAIT	ALT7	I
SRC_BOOT_CFG26	Boot configuration signal	EIM_LBA	ALT7	I
SRC_BOOT_CFG27	Boot configuration signal	EIM_EB0	ALT7	I
SRC_BOOT_CFG28	Boot configuration signal	EIM_EB1	ALT7	I
SRC_BOOT_CFG29	Boot configuration signal	EIM_RW	ALT7	I
SRC_BOOT_CFG30	Boot configuration signal	EIM_EB2	ALT7	I
SRC_BOOT_CFG31	Boot configuration signal	EIM_EB3	ALT7	I
SRC_BOOT_MODE0	Boot mode signal	BOOT_MODE0	No muxing	I
SRC_BOOT_MODE1	Boot mode signal	BOOT_MODE1	No muxing	I
SRC_ONOFF	ONOFF signal	ONOFF	No muxing	I
SRC_POR_B	Power on reset signal	POR_B	No muxing	I

60.3 Clocks

The table found here describes the clock sources for SRC.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 60-2. SRC Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

60.4 Top-level resets, power-up sequence and external supply integration

Information found here defines chip resets, power-up sequence, and external supply integration.

60.4.1 Reset and Power-up Flow

The chip presumes the following reset and power-up flow:

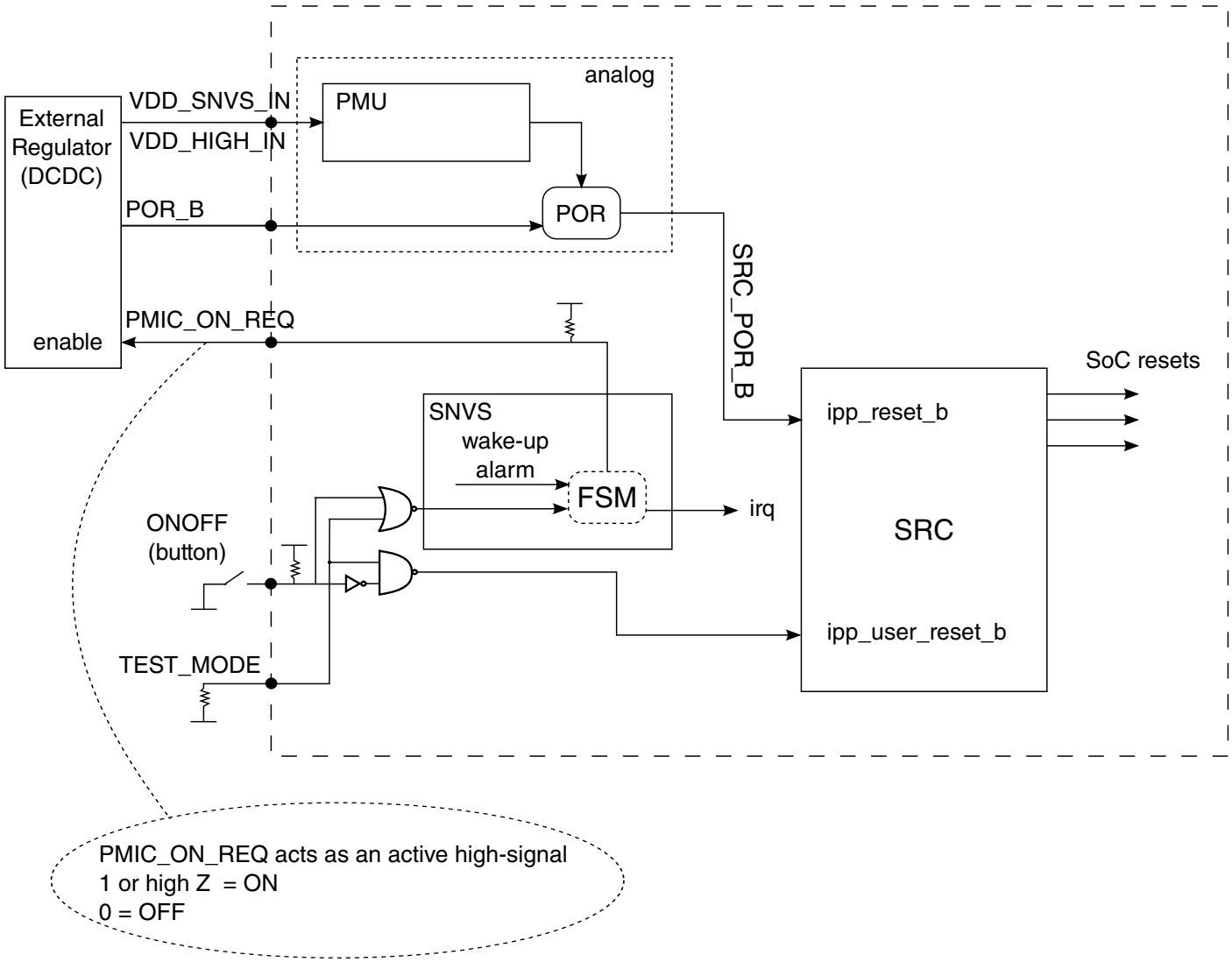


Figure 60-1. Chip reset scheme under PMU control

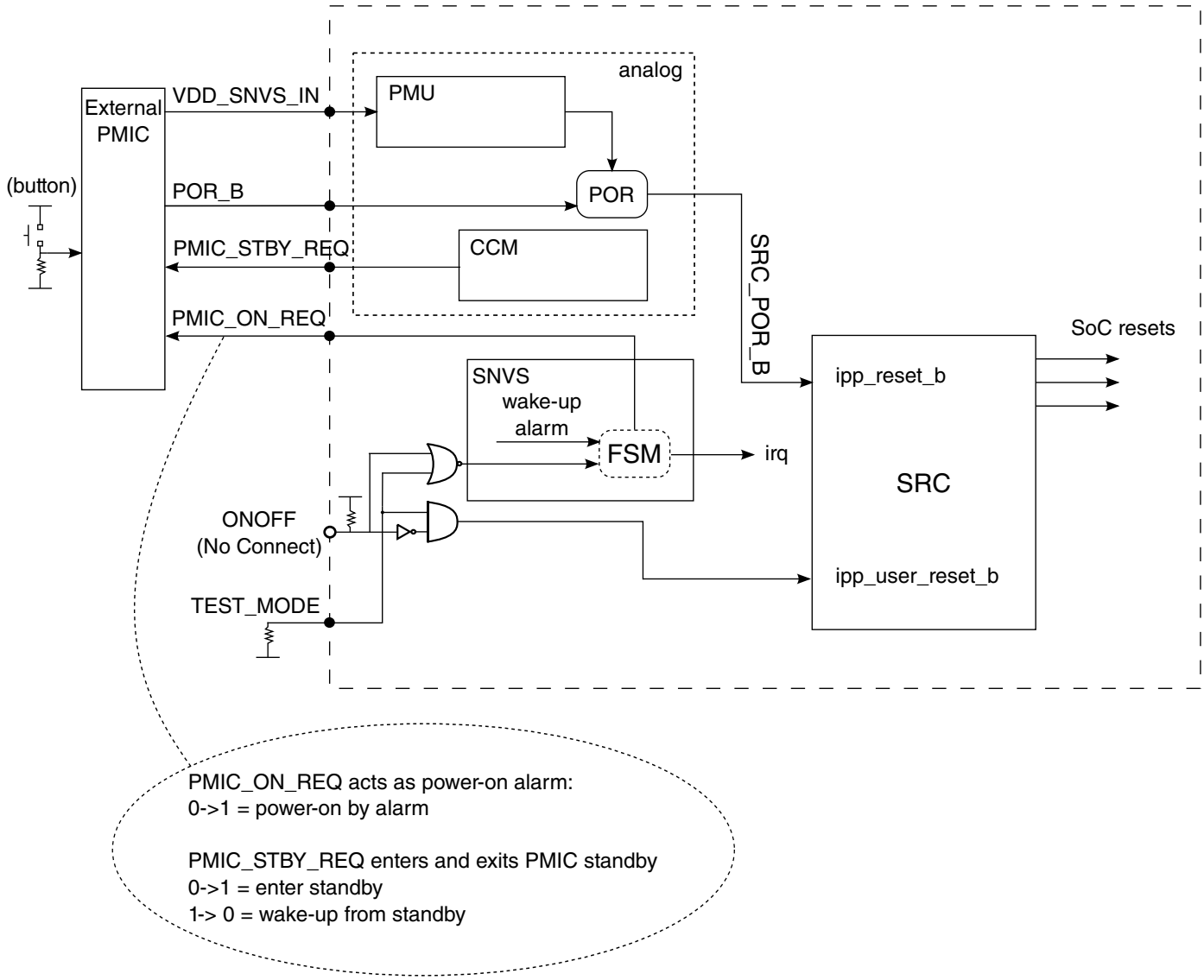


Figure 60-2. Chip reset scheme under external PMIC control

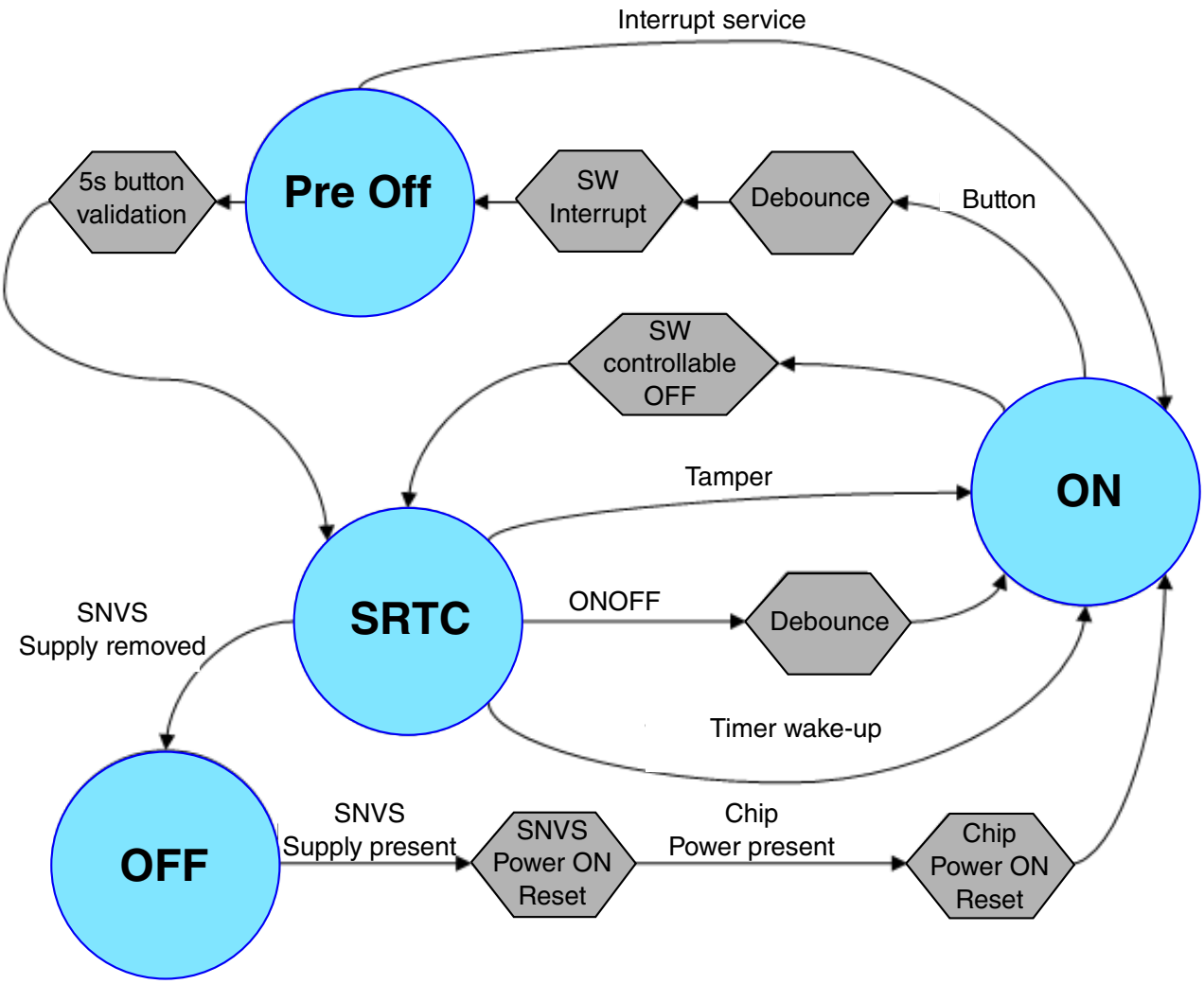


Figure 60-3. Chip on/off state flow diagram

60.4.2 Finite-State Machine (FSM)

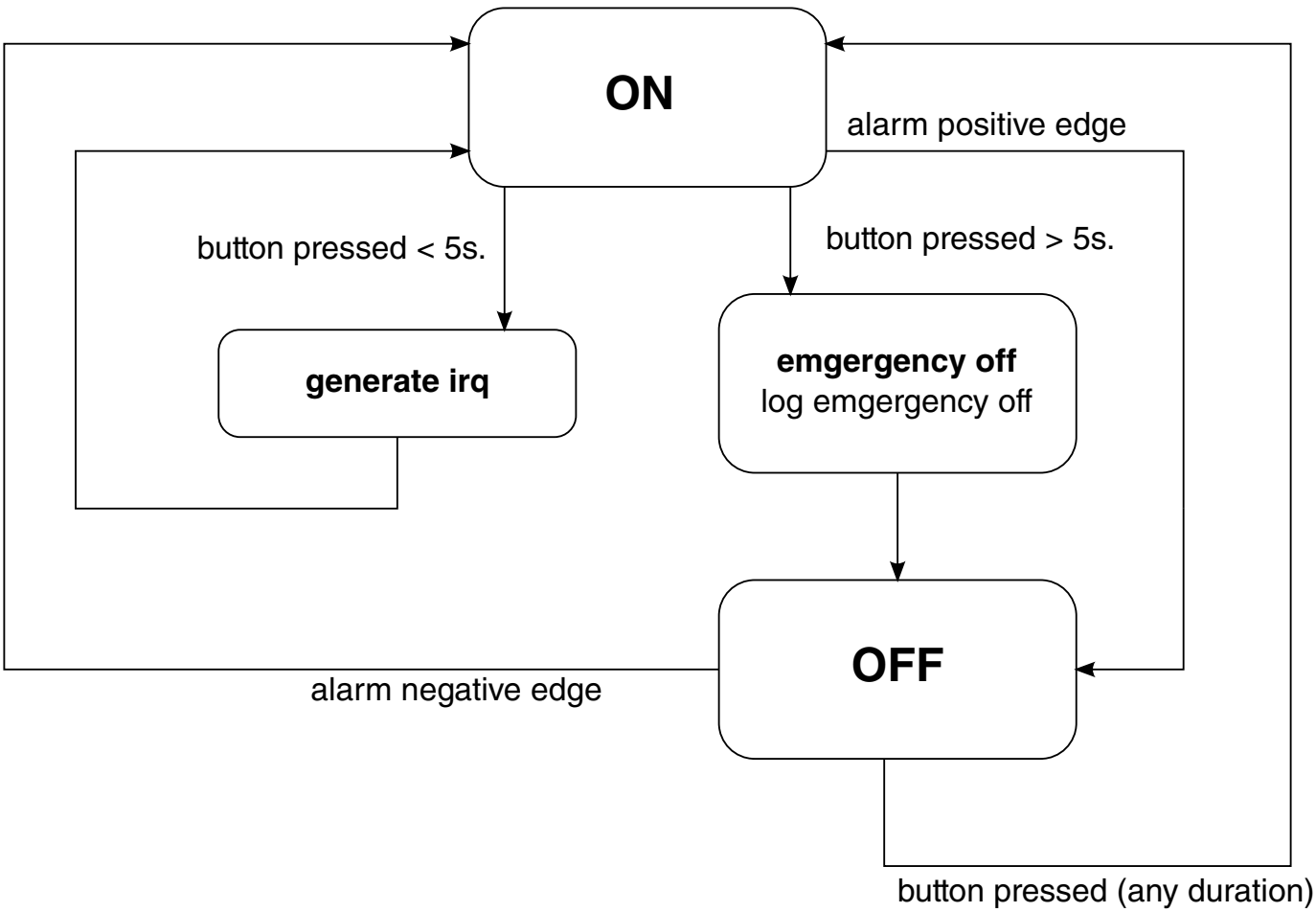


Figure 60-4. FSM

60.4.3 Power mode transitions

Table 60-3. Power mode transitions

Power mode	Configuration with external PMIC	Configuration with internal PMIC
ON, first time	1. Either coin cell or SoC power supply is connected to SNVS. 2. When button is pressed, PMIC powers on.	1. Either coin cell or SoC power supply is connected to SNVS. 2. When button is pressed, 'state' goes ON, PMIC_ON_REQ goes '1'. 3. External regulator is enabled.
Normal ON to OFF, by button	1. Button is pressed for a short duration on the external PMIC. 2. Interrupt request (irq) is sent to SoC from external PMIC.	1. SoC button is pressed for a short duration. 2. Interrupt request (irq) is sent to SoC from FSM. 3. Alarm timer is set up by software routine and started.

Table continues on the next page...

Table 60-3. Power mode transitions (continued)

Power mode	Configuration with external PMIC	Configuration with internal PMIC
	3. SoC is programming PMIC for power off when standby is asserted. 4. In CCM STOP mode, Standby is asserted, PMIC gates SoC supplies.	4. Upon alarm_in assertion to '1', PMIC_ON_REQ goes '0'. 5. External regulator goes OFF.
Emergency ON to OFF, by button	1. Button is pressed for an extended time on the external PMIC. 2. PMIC is powering off.	1. Button is pressed for longer than 5 seconds on the SoC. 2. FSM validates button pressed for 5 seconds. 3. Emergency power off is logged, PMIC_ON_REQ goes '0', alarm_mask goes '1'. 4. External regulator goes OFF.
OFF to ON, by button	1. Button is pressed on the external PMIC. 2. PMIC powers ON.	1. Button is pressed on the SoC. 2. PMIC_ON_REQ goes '1', alarm_mask goes '0'. 3. External regulator powers ON.
OFF to ON, by timer alarm	1. Timer alarm in SNVS is programmed by software before SoC goes OFF. 2. SoC enters OFF mode. 3. Upon timer limit, wake up alarm goes '0'. PMIC_ON_REQ goes '1'. 4. PMIC receives assertion of PMIC_ON_REQ and wakes up.	1. Timer alarm in SNVS is programmed by software before SoC goes OFF. 2. SoC enters OFF mode. 3. Upon timer limit, wake up alarm goes '0'. PMIC_ON_REQ goes '1'. 4. External regulator is enabled by PMIC_ON_REQ = 1.

60.5 Power-On Reset and power sequencing

This module generates an internal POR_B signal that is logically AND'ed with any externally applied SRC_POR_B signal. The internal POR_B signal will be held low until all of the following conditions are met:

- 4ms after the external power supply VDDHIGH_IN is valid
- 1ms after the VDD_SOC_CAP supply is valid

The 4ms and 1ms delays are derived from counting the 32 kHz RTC clock cycles; the accuracy depends on the accuracy of the RTC.

60.5.1 External POR using SRC_POR_B

If the external SRC_POR_B signal is used to control the processor POR, SRC_POR_B must remain low (asserted) until the VDD_ARM_CAP and VDD_SOC_CAP supplies are stable.

60.5.2 Internal POR

If the external SRC_POR_B signal is not used (always held high or left unconnected), the processor defaults to the internal POR function (PMU controls generation of the POR based on the power supplies).

If the internal POR function is used, the following power supply requirements must be met:

- VDD_ARM_IN and VDD_SOC_IN may be supplied from the same source, or
- VDD_SOC_IN can be supplied before VDD_ARM_IN with a maximum delay of 1 ms.

60.6 Functional Description

60.6.1 Reset Control

This section details the reset control of this device.

NOTE

SNVS resets are discussed in the *Multimedia Applications Processor Security Reference Manual for i.MX 6Dual/6Quad and i.MX 6Solo/6DualLite*.

60.6.1.1 Reset inputs and outputs

The reset control logic receives reset requests from all potential reset sources. All the immediate sources of reset are directly passed to the reset stretching block, whereas the resets requiring qualification are passed on to the reset qualification logic before they are sent to the reset stretching block.

All reset inputs and outputs are described in the following figure:

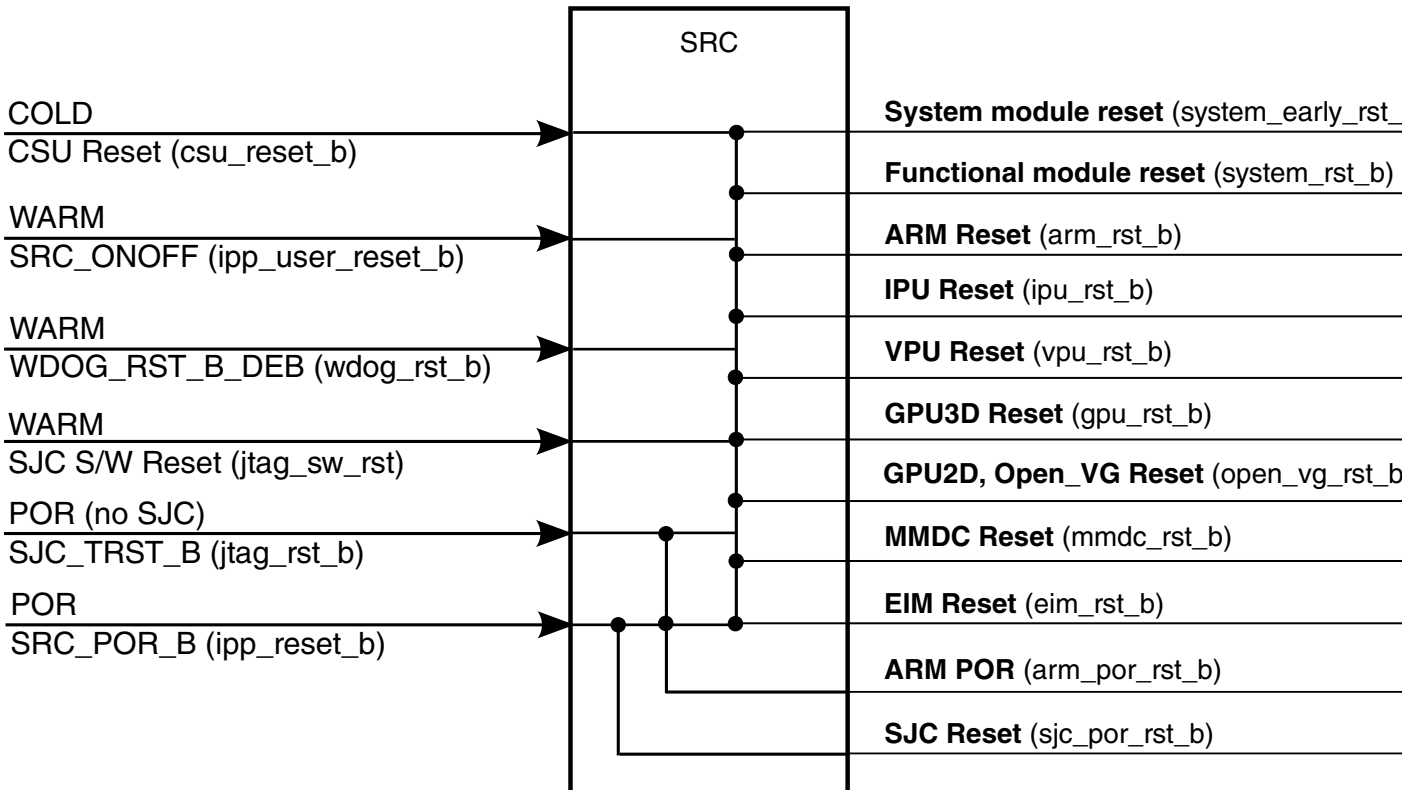


Figure 60-5. SRC inputs and outputs

The reset types and modules they affect are shown in Table 60-4. As there is no chip POR, the POR_B is used to reset the entire chip including test logic and JTAG modules.

NOTE

All resets are expected to be active low except jtag_sw_rst.

Table 60-4. SRC reset functionality

SoC Modules	POR	COLD	WARM
System modules (PLLs, fuses, etc)	yes	yes	yes
Functional modules	yes	yes	yes
Arm	yes	yes	yes
Arm SoC	yes	yes	yes
IPU	yes	yes	yes
VPU	yes	yes	yes
GPU3D	yes	yes	yes
MMDC	yes	yes	yes
Arm POR	yes	no	no
Arm debug	yes	no	no
SJC	yes	no	no

The reset priorities are POR (strongest), COLD and WARM (weakest). If a stronger reset is asserted during the sequence of a weaker reset, then the weaker sequence will be overridden, and the stronger reset sequence will commence. There is no priority within a reset type (POR, etc). If a reset is asserted during the reset sequence of the same type, the reset sequence will be interrupted and restarted.

The following lists the functionality of each of these reset outputs:

- system_early_rst_b - Resets the system modules that need to start first as CCM, OCOTP_CTRL, FUSEBOX, MMDC, etc.
- system_rst_b - Resets functional modules
- arm_rst_b - Resets Arm module (on regular system reset)
- ipu1_rst_b - Resets IPU1 module (on regular system reset)
- vpu_rst_b - Resets VPU module (on regular system reset)
- gpu_rst_b - Resets GPU3D module (on regular system reset)
- open_vg_rst_b - Resets Open_VG and GPU2D modules (on regular system reset)
- mmdc_rst_b - Resets MMDC
- arm_por_rst_b - Resets Arm POR input
- arm_soc_rst_b - Reset for Arm SOC
- arm_dbg_rst_b - Reset debug logic of Arm
- test_logic_rst_b - Reset test logic (IOMUXC, DAP)
- sjc_por_rst_b - Reset to SJC

NOTE

It is assumed that each reset source will deassert after its assertion, either due to reset generated to the system from SRC, or by negation of the reset source (if it came from an external source to the chip). In the latter case, the reset source is assumed to be held for at least 2 XTALI clocks so it can be sampled by SRC.

60.6.1.2 Reset Handling

60.6.1.2.1 Reset Qualification

The reset qualification logic qualifies the reset source before sending it out to the chip as a valid reset. WARM resets are in this category. All remaining reset sources are immediate resets and are acknowledged by the reset circuitry the moment they are asserted.

WARM resets are not immediate resets. WARM resets do reset the CCM, the source of MMDC clock. So, if a WARM reset were to immediately reset the CCM, then the MMDC clock would be shut off and this may cause the MMDC data to be lost. During

normal mode of operation of the chip, the protocol that is followed before shutting off the MMDC clock is that an mmdc_dvfs_req signal is sent to MMDC and only after the MMDC sends an acknowledge signal, mmdc_dvfs_ack, is the clock to the MMDC gated off.

However, the implication here is that a valid WARM reset source condition will not be able to cause a chip reset until the MMDC sends the acknowledge signal (mmdc_dvfs_ack). For example, a JTAG reset event has occurred but the JTAG reset will not be serviced until the mmdc_dvfs_ack signal is received. So, essentially all WARM reset sources depend on the MMDC providing the mmdc_dvfs_ack acknowledge signal before the reset is performed. When the MMDC is not used, mmdc_dvfs_ack is defaulted high.

The occurrence of WARM reset results in the assertion of warm_reset signal before the system resets are asserted to indicate to the MMDC that the reset occurred is a WARM reset.

A reset source is updated in the Reset Status Register (SRC_SRSR) when it becomes valid, provided it is asserted for the minimum amount of time after asserting. So, all immediate resets are immediately updated in the Status Register (SRC_SRSR). WARM resets would be updated when the mmdc_dvfs_ack signal is received from the MMDC.

After the reset is qualified, depending on the source of the reset, internal resets are asserted appropriately.

60.6.1.2.2 Reset Sequence and De-Assertion

The SRC_ONOFF will assert immediately after any reset source is recognized (except for the case of WARM reset when MMDC needs to answer mmdc_dvfs_ack first). After all of the reset sources are released, the SRC will start the following set of events depending on the type of reset that occurred.

60.6.1.2.3 POR (SRC_POR_B)

SRC_POR_B is an external reset signal. When the chip is powered up, the reset signal is passed through the POR_B pin indicating power-up sequence. The SRC resets the entire chip including the JTAG (SJC) module. All SRC registers will be reset during the POR sequence.

As soon as SRC_POR_B occurs, all resets are asserted and the entire chip is reset by SRC. The SRC_POR_B is stretched for 2 XTALI cycles and the stretching sequence takes place after 2 XTALI clocks of POR_B pin deassertion.

The sjc_por_rst_b signal is deasserted together with SRC_POR_B signal. The output is also deasserted after the stretching of SRC_POR_B has deasserted.

After the above resets deassert, system_early_rst_b reset is deasserted after 2 XTALI clocks. The system_early_rst_b is used for the CCM and PLL-IPs to start generating PLL clock ouputs and the system root clocks.

When the system root clocks are ready, the CCM will assert system_clk_ready signal. This signal is generated during the start sequence in the CCM and it involves the preparation of the PLLs to generate clock roots for functional operation.

SRC then enables OCOTP_CTRL and fusebox clocks, so that fuses can be loaded to OCOTP_CTRL.

- SRC will prepare the boot information and then de assert mmdc_rst_b.
- SRC will wait 8 ipg clock cycles, and then SRC will enable MMDC clocks.
- SRC will wait 8 XTALI clocks to allow MMDC to generate fixed external clock to external memory SDRAM.
- SRC will enable VPU and GPU clocks.
- After 8 ipg cycles, SRC will disable VPU, Open_VG and GPU clocks.
- After 8 ipg cycles, resets to all modules will be de-asserted (system_rst_b, vpu_rst_b, gpu_rst_b, open_vg_rst_b, ipu_rst_b).
- After 8 ipg cycles, system clocks will be enabled (en_system_clk).

60.6.1.2.4 COLD RESET

The sequence is similar to SRC_POR_B except the memory repair operation is not performed.

After the reset source deasserts, system_early_rst_b reset is deasserted after at least 2 XTALI clocks. The system_early_rst_b is used for the CCM and PLL-IPs to start generating PLL clock outputs and the system root clocks.

After the system root clocks are ready, the CCM will assert system_clk_ready signal. This signal is generated during the start sequence in the CCM and it involves the preparation of the PLLs to generate clock roots for functional operation. See CCM for more information.

After system_clk_ready arrives at the SRC, it will enable OCOTP_CTRL and fusebox clocks, so that fuses can be loaded to OCOTP_CTRL. OCOTP_CTRL will notify with iim_ready_flag when the fusebox loading finishes.

- SRC will prepare the boot information and then deassert mmdc_rst_b.
- SRC will wait 8 ipg clock cycles, and then SRC will enable MMDC clocks.

- SRC will wait 8 XTALI clocks to allow MMDC to generate fixed external clock to external memory SDRAM.
- SRC will enable VPU and GPU clocks so that reset will penetrate this module.
- After 8 ipg cycles, SRC will disable VPU, Open_VG and GPU clocks.
- After 8 ipg cycles resets to all modules will be deasserted (system_rst_b, vpu_rst_b, gpu_rst_b, open_vg_rst_b, ipu_rst_b).
- After 8 ipg cycles, system clocks will be enabled (en_system_clk).

60.6.1.2.5 WARM RESET

WARM reset will be enabled only if SRC_SCR[warm_reset_enable] bit is programmed. Otherwise, all WARM reset sources will generate a COLD reset. This bit will be reset only by a POR.

A WARM reset is similar to a COLD reset except that before the reset is sent, a signal to MMDC is asserted mmdc_dvfs_req (generates DVFS assertion to MMDC) to request to prepare MMDC to a WARM reset, finishing the transactions placing MMDC in self-refresh. Another signal will be asserted to MMDC (warm_reset) that will wrap the WARM reset sequence and will notify MMDC that a WARM reset is in process.

One of the sources of the WARM reset is SRC_ONOFF. If this is the case, it is qualified for 4 XTALI edges. The WARM reset is initiated immediately after 4 XTALI edges. The system does not come out of reset until the SRC_ONOFF is released.

In case the handshake mechanism with MMDC is stuck, meaning that no mmdc_dvfs_ack is received, COLD reset will be generated after a number of XTALI clocks. The number of XTALI clocks is defined in register the SRC_SCR[warm_rst_bypass_count] bitfield (default of this bitfield is 16 XTALI counts.)

The following is a basic description of the WARM reset sequence:

1. Arm sets SRC_SCR[warm_reset_enable] bit to enable the WARM Reset functionality. If this bit is not set, all WARM reset sources will result in COLD reset.
2. Assertion of one of the WARM reset sources.
3. The reset source is qualified in the SRC.
4. If mmdc_dvfs_ack signal is low, then SRC triggers the MMDC to switch to self-refresh mode using mmdc_dvfs_req signal. This is done through the CCM to combine with the DVFS sent from the CCM in case of frequency change of MMDC.
5. Wait for mmdc_dvfs_ack signal from the MMDC. If no ack is received during warm_rst_bypass_count number of XTALI clocks, COLD reset will be generated.
6. Assert warm_reset signal to MMDC.
7. SRC asserts system resets

The deassertion sequence is exactly the same as in the Cold Reset except waiting for 8 XTALIs for MMDC to generate fixed external clock to external memory MMDC. This stage is not needed in WARM reset since MMDC is held in self-refresh in WARM reset and there is no need to reconfigure it when exiting WARM reset.

WARM BOOT

Software can save any needed information in the memory before initiating a WARM reset. In this case, software will set SRC_SRSR[warm_boot] bit before initiating WARM reset. After the system returns to run mode, the warm_boot bit will still be set, indicating the software that data was saved in memory and can be reused.

NOTE

mmdc_dvfs_req and acknowledge during WARM reset can be masked in the CCM by configuration of register CCDR[17:16].

60.6.2 Parallel Reset Requests

SRC will follow the following rules in the case of parallel reset requests:

1. The order of strength of resets is POR - strongest, cold - medium, warm - weakest.
2. If a stronger reset is asserted during weaker reset sequence, then the stronger reset will take over and the stronger reset process will commence. The following cases fall into this category:
   - POR reset request in the middle of cold or warm reset process - the cold or warm reset process will be stopped and the POR sequence will start.
   - COLD reset request in the middle of warm reset process - the warm reset process will be stopped and the cold sequence will start.
3. If a weaker reset is asserted during stronger reset sequence, then the stronger reset sequence will continue without interference. If at the end of the stronger reset process the weaker request is still asserted then the weaker sequence will commence. The following cases fall into this category:
   - COLD or WARM reset requests in the middle of POR reset process - the POR process will continue without interference.
   - WARM reset request in the middle of COLD reset process - the COLD process will continue without interference.
4. If a similar reset request is asserted during the process of reset handling, then the process of reset handling will start over (with the same process). The following cases fall into this category:

- POR reset request in the middle of POR reset process - the POR process will start over.
- COLD reset request in the middle of COLD reset process - the COLD process will start over.

There is one exception to this category: WARM reset request in the middle of WARM reset process. In this case, the new WARM reset process cannot restart because MMDC is reset and there can't be a handshake with MMDC if it is reset. In this case the first WARM reset will continue, and only if the second WARM reset is still asserted after the first one has finished, the WARM sequence will start again.

60.6.3 Boot Mode Control

60.6.3.1 BOOT_MODE Pin Latching

The exact boot sequence is controlled by the values of the BOOT_MODE pins on this device.

The value of the BOOT_MODE pins will be latched after the OCOTP_CTRL asserts the fuse read completion flag. After latching, the values of the BOOT_MODE pins are used to determine the booting options of the core as described in the SRC_SBMRx registers.

The boot mode general purpose bits can be provided to the SRC from either e-fuses or GPIO signals. The gpio_bt_sel e-fuse defines the source to be used to derive the boot information. When gpio_bt_sel is set, e-fuses are used. When cleared, GPIO signals are used.

The boot information is provided in SRC_SBMR1 register. The figure below shows the selection of boot mode information.

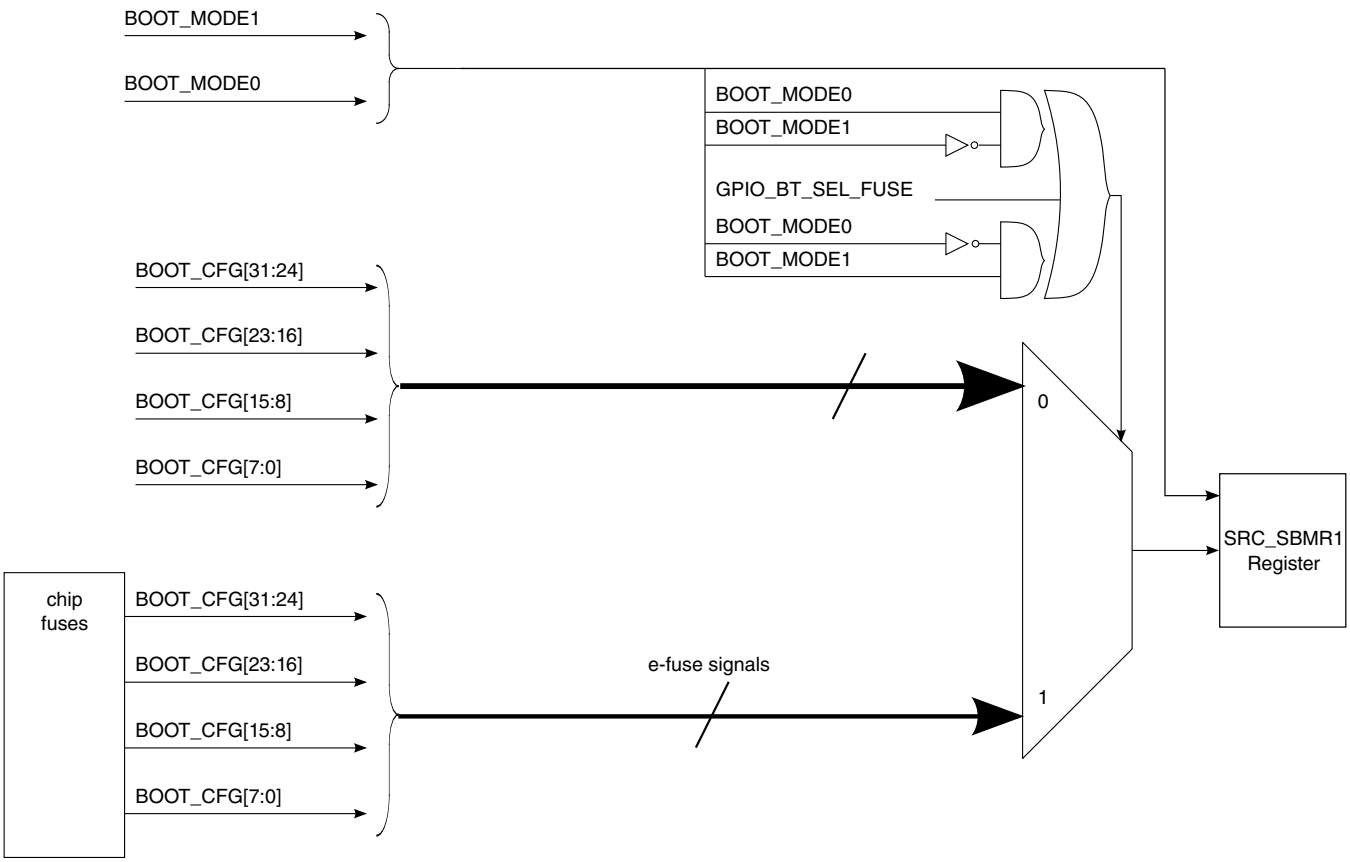


Figure 60-6. Boot mode information

60.7 SRC Memory Map/Register Definition

SRC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20D_8000	SRC Control Register (SRC_SCR)	32	R/W	0000_0521h	60.7.1/5035
20D_8004	SRC Boot Mode Register 1 (SRC_SBMR1)	32	R	0000_0000h	60.7.2/5038
20D_8008	SRC Reset Status Register (SRC_SRSR)	32	R/W	0000_0001h	60.7.3/5039
20D_8014	SRC Interrupt Status Register (SRC_SISR)	32	R	0000_0000h	60.7.4/5041
20D_8018	SRC Interrupt Mask Register (SRC_SIMR)	32	R/W	0000_001Fh	60.7.5/5043
20D_801C	SRC Boot Mode Register 2 (SRC_SBMR2)	32	R	0000_0000h	60.7.6/5045
20D_8020	SRC General Purpose Register 1 (SRC_GPR1)	32	R/W	0000_0000h	60.7.7/5046
20D_8024	SRC General Purpose Register 2 (SRC_GPR2)	32	R/W	0000_0000h	60.7.8/5047
20D_8028	SRC General Purpose Register 3 (SRC_GPR3)	32	R/W	0000_0000h	60.7.9/5047

Table continues on the next page...

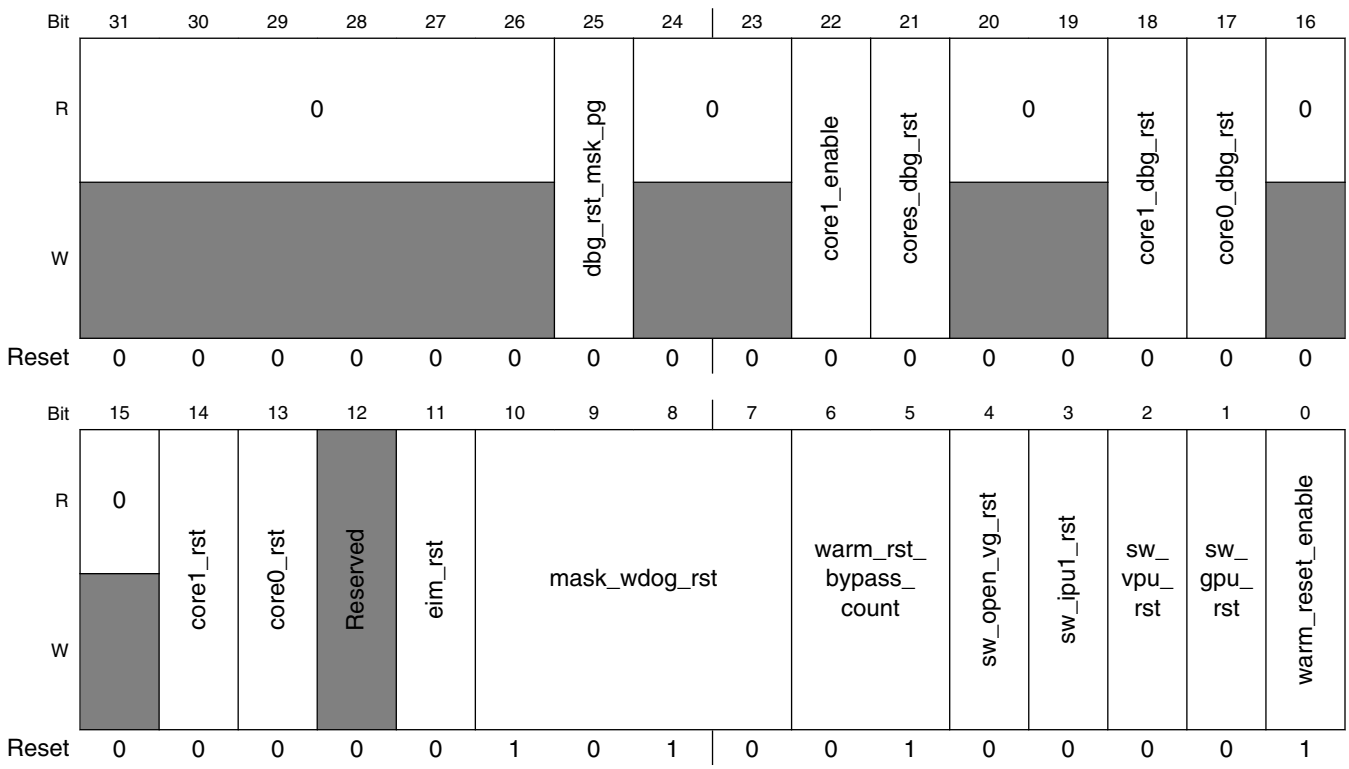
SRC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20D_802C	SRC General Purpose Register 4 (SRC_GPR4)	32	R/W	0000_0000h	60.7.10/ 5048
20D_8030	SRC General Purpose Register 5 (SRC_GPR5)	32	R/W	0000_0000h	60.7.11/ 5048
20D_8034	SRC General Purpose Register 6 (SRC_GPR6)	32	R/W	0000_0000h	60.7.12/ 5049
20D_8038	SRC General Purpose Register 7 (SRC_GPR7)	32	R/W	0000_0000h	60.7.13/ 5049
20D_803C	SRC General Purpose Register 8 (SRC_GPR8)	32	R/W	0000_0000h	60.7.14/ 5050
20D_8040	SRC General Purpose Register 9 (SRC_GPR9)	32	R/W	0000_0000h	60.7.15/ 5050
20D_8044	SRC General Purpose Register 10 (SRC_GPR10)	32	R/W	0000_0000h	60.7.16/ 5051

60.7.1 SRC Control Register (SRC_SCR)

The Reset control register (SCR), contains bits that control operation of the reset controller.

Address: 20D_8000h base + 0h offset = 20D_8000h



SRC_SCR field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.
25 dbg_rst_msk_pg	Do not assert debug resets after power gating event of core 0 do not mask core debug resets (debug resets will be asserted after power gating event) 1 mask core debug resets (debug resets won't be asserted after power gating event)
24–23 Reserved	This read-only field is reserved and always has the value 0.
22 core1_enable	core1 enable. **NOTE:** core0 cannot be disabled 0 core1 is disabled 1 core1 is enabled

Table continues on the next page...

SRC_SCR field descriptions (continued)

Field	Description
21 cores_dbg_rst	Software reset for debug of arm platform only. **NOTE:** This is a self clearing bit. After it is set to 1, the reset process will begin, and when it finishes, this bit will be self cleared. 0 do not assert arm platform debug reset 1 assert arm platform debug reset
20–19 Reserved	This read-only field is reserved and always has the value 0.
18 core1_dbg_rst	Software reset for core1 debug only. **NOTE:** This is a self clearing bit. After it is set to 1, the reset process will begin, and when it finishes, this bit will be self cleared. 0 do not assert core1 debug reset 1 assert core1 debug reset
17 core0_dbg_rst	Software reset for core0 debug only. **NOTE:** This is a self clearing bit. After it is set to 1, the reset process will begin, and when it finishes, this bit will be self cleared. 0 do not assert core0 debug reset 1 assert core0 debug reset
16–15 Reserved	This read-only field is reserved and always has the value 0.
14 core1_rst	Software reset for core1 only. **NOTE:** This is a self clearing bit. After it is set to 1, the reset process will begin, and when it finishes, this bit will be self cleared. 0 do not assert core1 reset 1 assert core1 reset
13 core0_rst	Software reset for core0 only. **NOTE:** This is a self clearing bit. After it is set to 1, the reset process will begin, and when it finishes, this bit will be self cleared. 0 do not assert core0 reset 1 assert core0 reset
12 -	This field is reserved. Reserved
11 eim_rst	EIM reset is needed in order to reconfigure the eim chip select. The software reset bit must de-asserted. The eim chip select configuration should be updated. The software bit must be re-asserted since this is not self-refresh.
10–7 mask_wdog_rst	Mask wdog_rst_b source. If these 4 bits are coded from A to 5 then, the wdog_rst_b input to SRC will be masked and the wdog_rst_b will not create a reset to the chip. **NOTE:** During the time the WDOG event is masked using SRC logic, it is likely that the WDOG Reset Status Register (WRSR) bit 1 (which indicates a WDOG timeout event) will get asserted. software / OS developer must prepare for this case. Re-enabling the WDOG is possible, by un-

Table continues on the next page...

SRC_SCR field descriptions (continued)

Field	Description
	masking it in SRC, though it must be preceded by servicing the WDOG. However, for the case that the event has been asserted, the status bit (WRSR bit-1) will remain asserted, regardless of servicing the WDOG module. (Hardware reset is the only way to cause the de-assertion of that bit). any other code will be coded to 1010 i.e. wdog_rst_b is not masked 0101 wdog_rst_b is masked 1010 wdog_rst_b is not masked (default)
6–5 warm_rst_ bypass_count	Defines the XTALI cycles to count before bypassing the MMDC acknowledge for WARM reset. If the MMDC acknowledge will not be asserted before this counter has elapsed, then a COLD reset will be initiated. 00 Counter not to be used - system will wait until MMDC acknowledge until it is asserted. 01 Wait 16 XTALI cycles before changing WARM reset to a COLD reset. 10 Wait 32 XTALI cycles before changing WARM reset to a COLD reset. 11 Wait 64 XTALI cycles before changing WARM reset to a COLD reset
4 sw_open_vg_rst	Software reset for open_vg **NOTE:** This is a self clearing bit. After it is set to 1, the reset process will begin, and when it finishes, this bit will be self cleared. Software can determine that the reset has finished after this bit is cleared. Software can also configure SRC to generate interrupt after the software has finished. Please refer to SRC_SISR register for details. **NOTE:** The reset process will involve 8 open_vg cycles before negating the open_vg reset, to allow reset assertion to propagate into open_vg. 0 do not assert open_vg reset 1 assert open_vg reset
3 sw_ipu1_rst	Software reset for IPU1 **NOTE:** This is a self clearing bit. After it is set to 1, the reset process will begin, and when it finishes, this bit will be self cleared. Software can determine that the reset has finished after this bit is cleared. Software can also configure SRC to generate interrupt after the software has finished. Please refer to SRC_SISR register for details. 0 do not assert IPU1 reset 1 assert IPU1 reset
2 sw_vpu_rst	Software reset for VPU **NOTE:** This is a self clearing bit. After it is set to 1, the reset process will begin, and when it finishes, this bit will be self cleared. Software can determine that the reset has finished after this bit is cleared. Software can also configure SRC to generate interrupt after the software has finished. Please refer to SRC_SISR register for details. **NOTE:** The reset process will involve 8 VPU cycles before negating the VPU reset, to allow reset assertion to propagate into VPU. 0 do not assert VPU reset 1 assert VPU reset
1 sw_gpu_rst	Software reset for GPU **NOTE:** This is a self clearing bit. After it is set to 1, the reset process will begin, and when it finishes, this bit will be self cleared. Software can determine that the reset has finished after this bit is cleared.

Table continues on the next page...

SRC_SCR field descriptions (continued)

Field	Description
	Software can also configure SRC to generate interrupt after the software has finished. Please refer to SRC_SISR register for details. **NOTE:** The reset process will involve 8 GPU cycles before negating the GPU reset, to allow reset assertion to propagate into GPU. 0 do not assert GPU reset 1 assert GPU reset
0 warm_reset_ enable	WARM reset enable bit. WARM reset will be enabled only if warm_reset_enable bit is set. Otherwise all WARM reset sources will generate COLD reset. 0 WARM reset disabled 1 WARM reset enabled

60.7.2 SRC Boot Mode Register 1 (SRC_SBMR1)

The Boot Mode register (SBMR) contains bits that reflect the status of Boot Mode Pins of the chip. The reset value is configuration dependent (depending on boot/fuses/IO pads).

If SRC_GPR10[28] bit is set, this bit instructs the ROM code to use the SRC_GPR9 register as if it is SBMR1. This allows software to override the fuse bits and boot from an alternate boot source.

Address: 20D_8000h base + 4h offset = 20D_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BOOT_CFG4[7:0]								BOOT_CFG3[7:0]								BOOT_CFG2[7:0]								BOOT_CFG1[7:0]							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_SBMR1 field descriptions

Field	Description
31–24 BOOT_ CFG4[7:0]	Refer to fusemap.
23–16 BOOT_ CFG3[7:0]	Refer to fusemap.
15–8 BOOT_ CFG2[7:0]	Refer to fusemap.
BOOT_ CFG1[7:0]	Refer to fusemap.

60.7.3 SRC Reset Status Register (SRC_SRSR)

The SRSR is a write to one clear register which records the source of the reset events for the chip. The SRC reset status register will capture all the reset sources that have occurred. This register is reset on ipp_reset_b. This is a read-write register.

For bit[6-0] - writing zero does not have any effect. Writing one will clear the corresponding bit. The individual bits can be cleared by writing one to that bit. When the system comes out of reset, this register will have bits set corresponding to all the reset sources that occurred during system reset. Software has to take care to clear this register by writing one after every reset that occurs so that the register will contain the information of recently occurred reset.

Address: 20D_8000h base + 8h offset = 20D_8008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															warm_boot
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

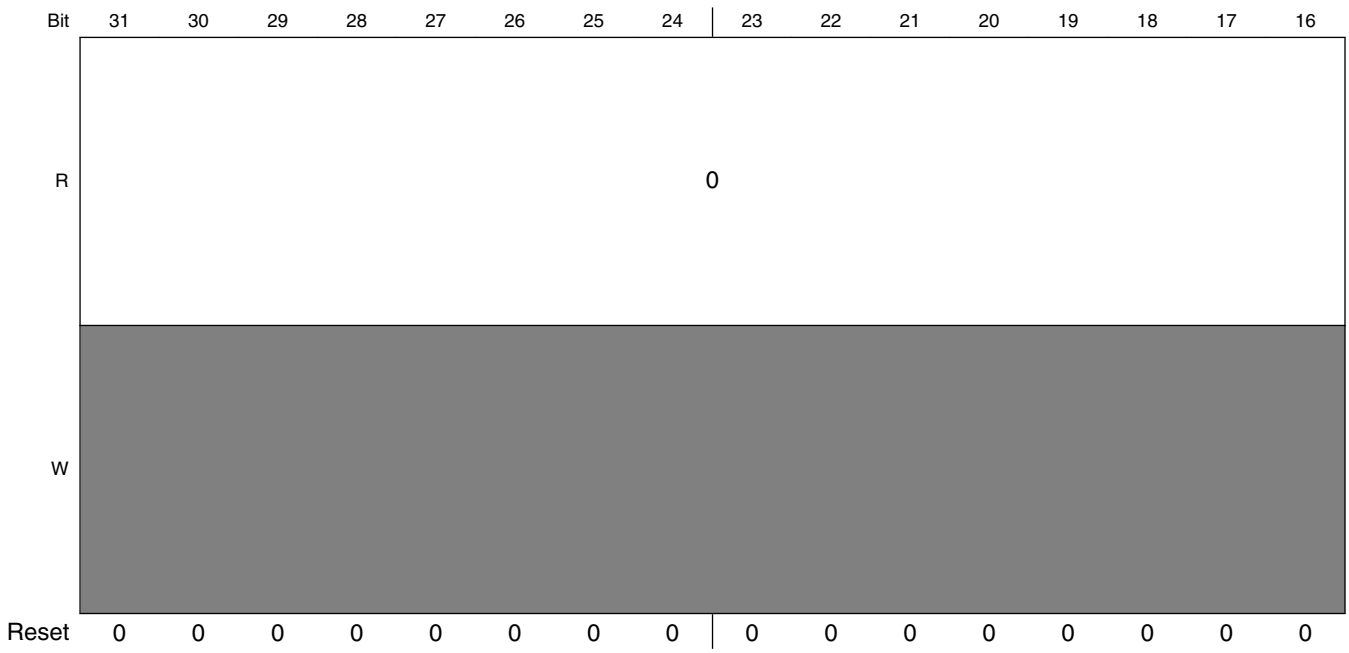
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									jtag_sw_rst	jtag_rst_b	wdog_rst_b	ipp_user_reset_b	csu_reset_b	0	ipp_reset_b
W										w1c	w1c	w1c	w1c	w1c		w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

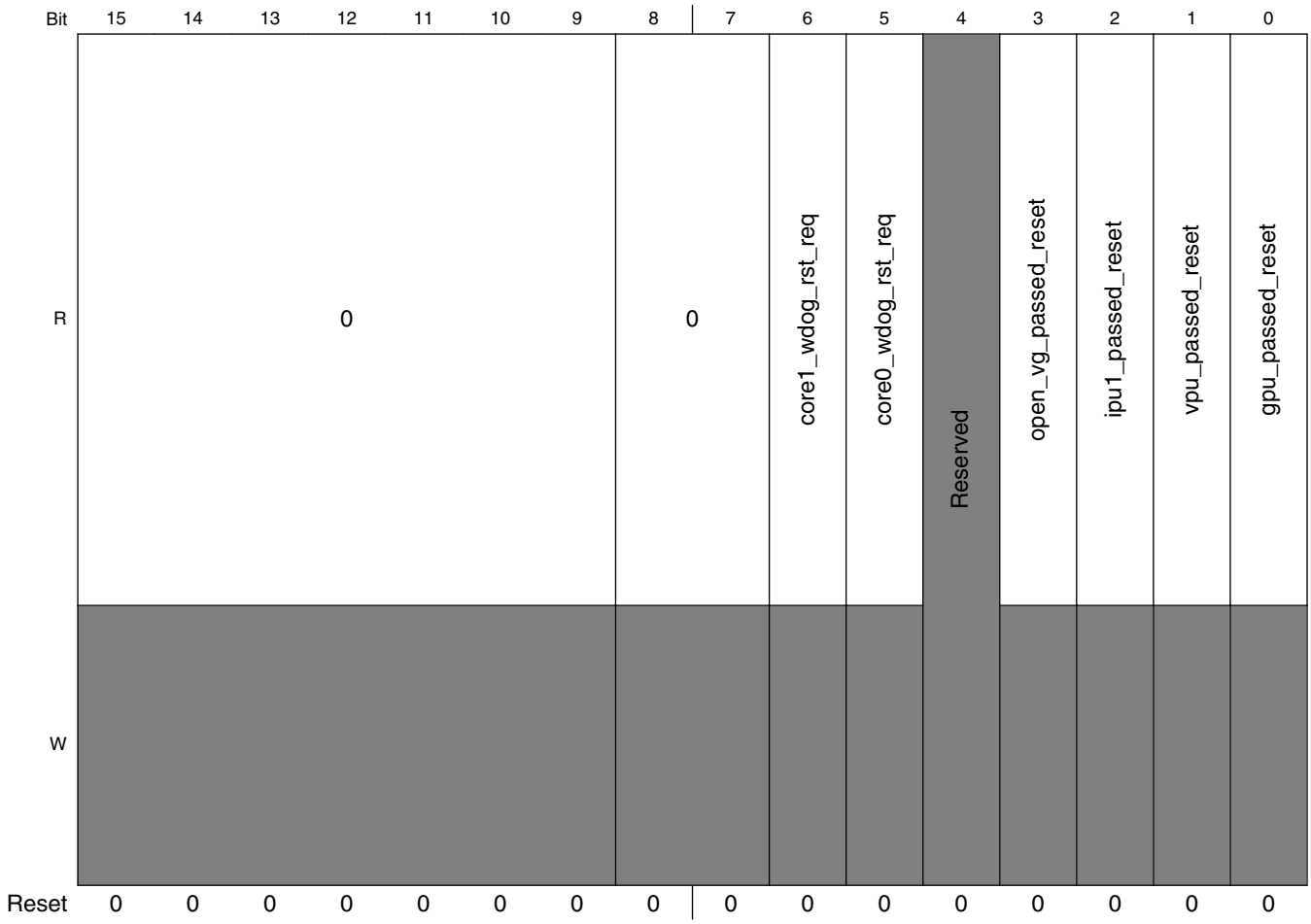
SRC_SRSR field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 warm_boot	WARM boot indication shows that WARM boot was initiated by software. This indicates to the software that it saved the needed information in the memory before initiating the WARM reset. In this case, software will set this bit to '1', before initiating the WARM reset. The warm_boot bit should be used as indication only after a warm_reset sequence. Software should clear this bit after warm_reset to indicate that the next warm_reset is not performed with warm_boot. Please refer to Reset Sequence and De-Assertion for details on warm_reset. 0 WARM boot process not initiated by software. 1 WARM boot initiated by software.
15–7 Reserved	This read-only field is reserved and always has the value 0.
6 jtag_sw_rst	JTAG software reset. Indicates whether the reset was the result of software reset from JTAG. 0 Reset is not a result of the mentioned case. 1 Reset is a result of the mentioned case.
5 jtag_rst_b	HIGH - Z JTAG reset. Indicates whether the reset was the result of HIGH-Z reset from JTAG. 0 Reset is not a result of HIGH-Z reset from JTAG. 1 Reset is a result of HIGH-Z reset from JTAG.
4 wdog_rst_b	IC Watchdog Time-out reset. Indicates whether the reset was the result of the watchdog time-out event. 0 Reset is not a result of the watchdog time-out event. 1 Reset is a result of the watchdog time-out event.
3 ipp_user_reset_b	Indicates whether the reset was the result of the ipp_user_reset_b qualified reset. 0 Reset is not a result of the ipp_user_reset_b qualified as COLD reset event. 1 Reset is a result of the ipp_user_reset_b qualified as COLD reset event.
2 csu_reset_b	Indicates whether the reset was the result of the csu_reset_b input. **NOTE:** If case the csu_reset_b occurred during a WARM reset process, during the phase that ipg_clk is not available yet, then the occurrence of CSU reset will not be reflected in this bit. 0 Reset is not a result of the csu_reset_b event. 1 Reset is a result of the csu_reset_b event.
1 Reserved	This read-only field is reserved and always has the value 0.
0 ipp_reset_b	Indicates whether reset was the result of ipp_reset_b pin (Power-up sequence) 0 Reset is not a result of ipp_reset_b pin. 1 Reset is a result of ipp_reset_b pin.

60.7.4 SRC Interrupt Status Register (SRC_SISR)

Address: 20D_8000h base + 14h offset = 20D_8014h





SRC_SISR field descriptions

Field	Description
31–9 Reserved	This read-only field is reserved and always has the value 0.
8–7 Reserved	This read-only field is reserved and always has the value 0.
6 core1_wdog_rst_req	WDOG reset request from core1. Read-only status bit.
5 core0_wdog_rst_req	WDOG reset request from core0. Read-only status bit.
4 -	This field is reserved. Reserved
3 open_vg_passed_reset	Interrupt generated to indicate that open_vg passed software reset and is ready to be used 0 interrupt generated not due to open_vg passed reset 1 interrupt generated due to open_vg passed reset
2 ipu1_passed_reset	Interrupt generated to indicate that ipu passed software reset and is ready to be used

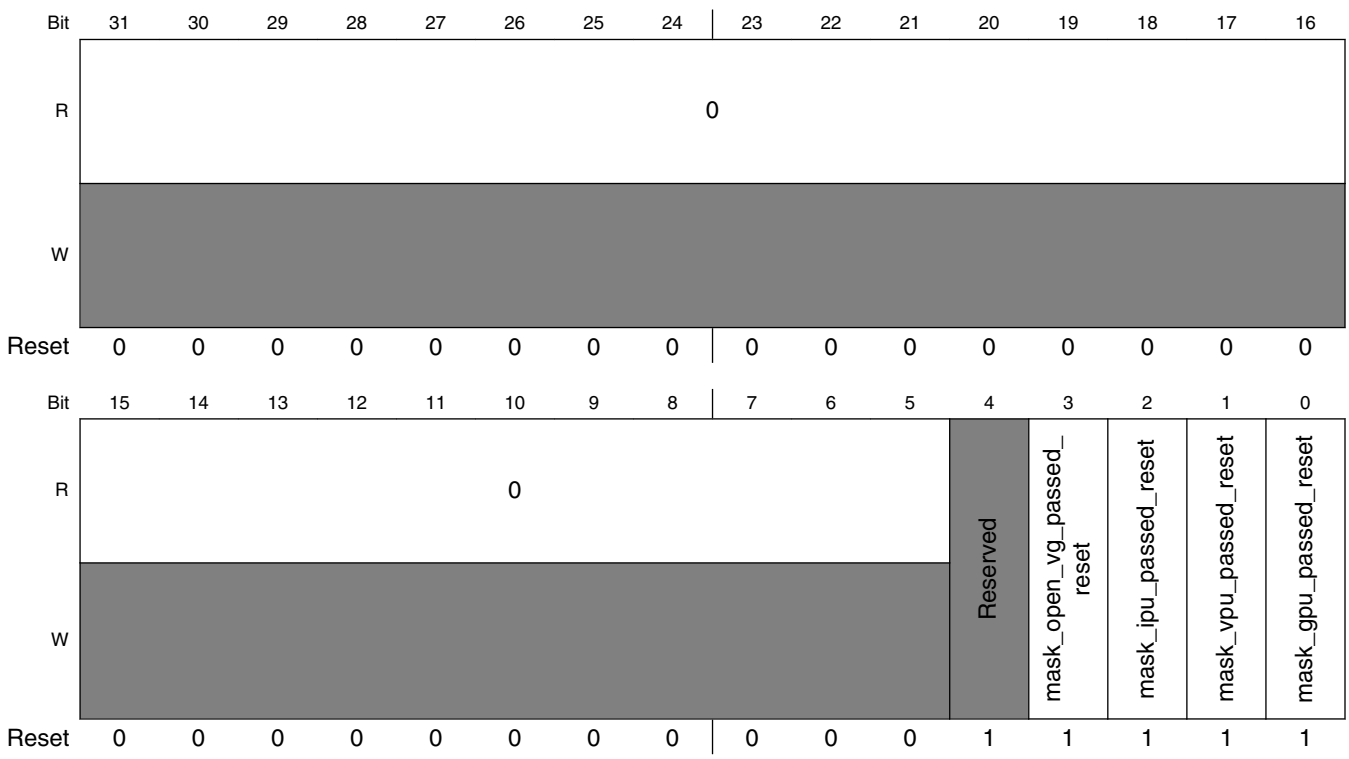
Table continues on the next page...

SRC_SISR field descriptions (continued)

Field	Description
	0 interrupt generated not due to ipu passed reset 1 interrupt generated due to ipu passed reset
1 vpu_passed_ reset	Interrupt generated to indicate that VPU passed software reset and is ready to be used 0 interrupt generated not due to VPU passed reset 1 interrupt generated due to VPU passed reset
0 gpu_passed_ reset	Interrupt generated to indicate that GPU passed software reset and is ready to be used 0 interrupt generated not due to GPU passed reset 1 interrupt generated due to GPU passed reset

60.7.5 SRC Interrupt Mask Register (SRC_SIMR)

Address: 20D_8000h base + 18h offset = 20D_8018h



SRC_SIMR field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 -	This field is reserved. Reserved

Table continues on the next page...

SRC_SIMR field descriptions (continued)

Field	Description
3 mask_open_vg_ passed_reset	mask interrupt generation due to open_vg passed reset 0 do not mask interrupt due to open_vg passed reset - interrupt will be created 1 mask interrupt due to open_vg passed reset
2 mask_ipu_ passed_reset	mask interrupt generation due to ipu passed reset 0 do not mask interrupt due to ipu passed reset - interrupt will be created 1 mask interrupt due to ipu passed reset
1 mask_vpu_ passed_reset	mask interrupt generation due to VPU passed reset 0 do not mask interrupt due to VPU passed reset - interrupt will be created 1 mask interrupt due to VPU passed reset
0 mask_gpu_ passed_reset	mask interrupt generation due to GPU passed reset 0 do not mask interrupt due to GPU passed reset - interrupt will be created 1 mask interrupt due to GPU passed reset

60.7.6 SRC Boot Mode Register 2 (SRC_SBMR2)

The Boot Mode register (SBMR), contains bits that reflect the status of Boot Mode Pins of the chip. The default values for those bits depends on the values of pins/fuses during reset sequence, hence the question mark on their default value.

Address: 20D_8000h base + 1Ch offset = 20D_801Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						BMOD[1:0]		0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											BT_FUSE_SEL	DIR_BT_DIS	0	SEC_CONFIG[1:0]	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_SBMR2 field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.
25–24 BMOD[1:0]	BMOD[1:0] shows the latched state of the BOOT_MODE1 and BOOT_MODE0 signals on the rising edge of POR_B. See the Boot mode pin settings section of System Boot.
23–5 Reserved	This read-only field is reserved and always has the value 0.
4 BT_FUSE_SEL	BT_FUSE_SEL (connected to gpio bt_fuse_sel) shows the state of the BT_FUSE_SEL fuse. See Fusemap for additional information on this fuse.
3 DIR_BT_DIS	DIR_BT_DIS shows the state of the DIR_BT_DIS fuse. See the Fusemap chapter for additional information on this fuse.
2 Reserved	This read-only field is reserved and always has the value 0.
SEC_ CONFIG[1:0]	SECONFIG[1] shows the state of the SECONFIG[1] fuse. See Fusemap for additional information on this fuse. SECONFIG[0] shows the state of the SECONFIG[0] fuse. This fuse is shown as reserved in Fusemap (address 0x440[1]) because it does not have a user-relevant function.

60.7.7 SRC General Purpose Register 1 (SRC_GPR1)

NOTE

This register is used by the ROM code and should not be used by application software.

Address: 20D_8000h base + 20h offset = 20D_8020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PERSISTENT_ENTRY0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR1 field descriptions

Field	Description
PERSISTENT_ ENTRY0	Holds entry function for core0 for waking-up from low power mode. The SRC ensures that the register value will persist across system resets.

60.7.8 SRC General Purpose Register 2 (SRC_GPR2)

NOTE

This register is used by the ROM code and should not be used by application software.

Address: 20D_8000h base + 24h offset = 20D_8024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PERSISTENT_ARG0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR2 field descriptions

Field	Description
PERSISTENT_ARG0	Holds argument of entry function for core0 for waking-up from low power mode. The SRC ensures that the register value will persist across system resets.

60.7.9 SRC General Purpose Register 3 (SRC_GPR3)

NOTE

This register is used by the ROM code and should not be used by application software.

Address: 20D_8000h base + 28h offset = 20D_8028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PERSISTENT_ENTRY1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR3 field descriptions

Field	Description
PERSISTENT_ENTRY1	Holds entry function for core1. The SRC ensures that the register value will persist across system resets.

60.7.10 SRC General Purpose Register 4 (SRC_GPR4)

NOTE

This register is used by the ROM code and should not be used by application software.

Address: 20D_8000h base + 2Ch offset = 20D_802Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PERSISTENT_ARG1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR4 field descriptions

Field	Description
PERSISTENT_ARG1	Holds argument of entry function for core1. The SRC ensures that the register value will persist across system resets.

60.7.11 SRC General Purpose Register 5 (SRC_GPR5)

NOTE

This register is used by the ROM code and should not be used by application software.

Address: 20D_8000h base + 30h offset = 20D_8030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR5 field descriptions

Field	Description
-	Read/write general purpose bits used to store an arbitrary value.

60.7.12 SRC General Purpose Register 6 (SRC_GPR6)

NOTE

This register is used by the ROM code and should not be used by application software.

Address: 20D_8000h base + 34h offset = 20D_8034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR6 field descriptions

Field	Description
-	Read/write general purpose bits used to store an arbitrary value.

60.7.13 SRC General Purpose Register 7 (SRC_GPR7)

NOTE

This register is used by the ROM code and should not be used by application software.

Address: 20D_8000h base + 38h offset = 20D_8038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR7 field descriptions

Field	Description
-	Read/write general purpose bits used to store an arbitrary value.

60.7.14 SRC General Purpose Register 8 (SRC_GPR8)

NOTE

This register is used by the ROM code and should not be used by application software.

Address: 20D_8000h base + 3Ch offset = 20D_803Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR8 field descriptions

Field	Description
-	Read/write general purpose bits used to store an arbitrary value.

60.7.15 SRC General Purpose Register 9 (SRC_GPR9)

NOTE

This register is used by the ROM code and should not be used by application software.

Address: 20D_8000h base + 40h offset = 20D_8040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR9 field descriptions

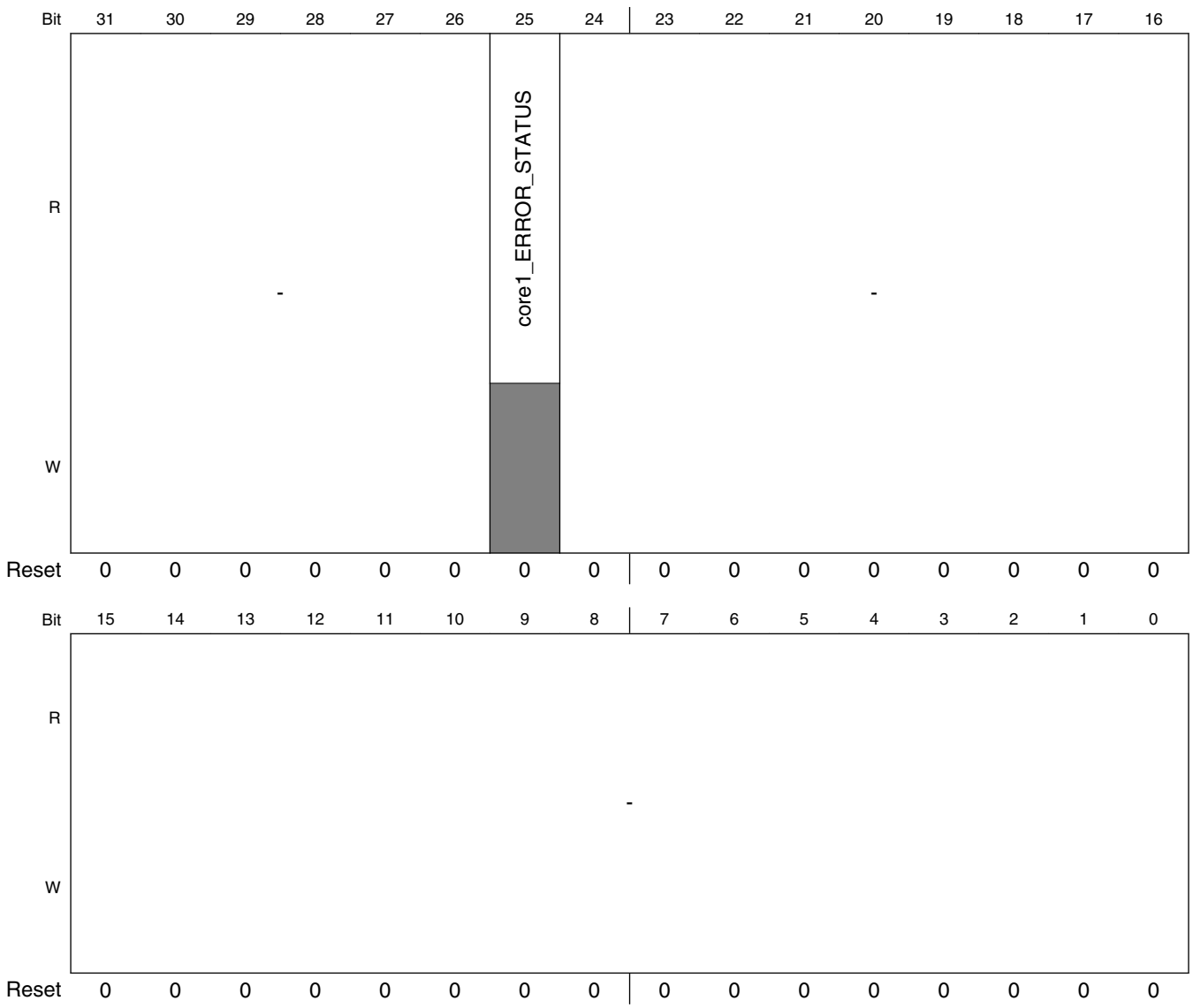
Field	Description
-	This field is reserved. Reserved.

60.7.16 SRC General Purpose Register 10 (SRC_GPR10)

NOTE

This register is used by the ROM code and should not be used by application software.

Address: 20D_8000h base + 44h offset = 20D_8044h



SRC_GPR10 field descriptions

Field	Description
31–26 -	Read/write bits, for general purpose **NOTE:** Reset only by POR
25 core1_ERROR_ STATUS	core1 error status bit.
-	Read/write bits, for general purpose **NOTE:** Reset only by POR

Chapter 61
Synchronous Serial Interface (SSI)

61.1 Overview

This block guide presents the Synchronous Serial Interface (SSI), and discusses the architecture, the programming model, the operating modes, and initialization of SSI.

The SSI is a full-duplex, serial port that allows the chip to communicate with a variety of serial devices. These serial devices can be standard CODer-DECoder (CODECs), Digital Signal Processors (DSPs), microprocessors, peripherals, and popular industry audio codecs that implement the inter-IC sound bus standard (I2S) and Intel AC97 standard.

SSI is typically used to transfer samples in a periodic manner. The SSI consists of independent transmitter and receiver sections with independent clock generation and frame synchronization.

The Figure 61-1 illustrates the organization of the SSI. It consists of control registers to set up the port, status register, separate transmit and receive circuits with FIFO registers, and separate serial clock and frame sync generation for the transmit and receive sections. The second set of Tx and Rx FIFOs, replicates the logic used for the first set of FIFOs.

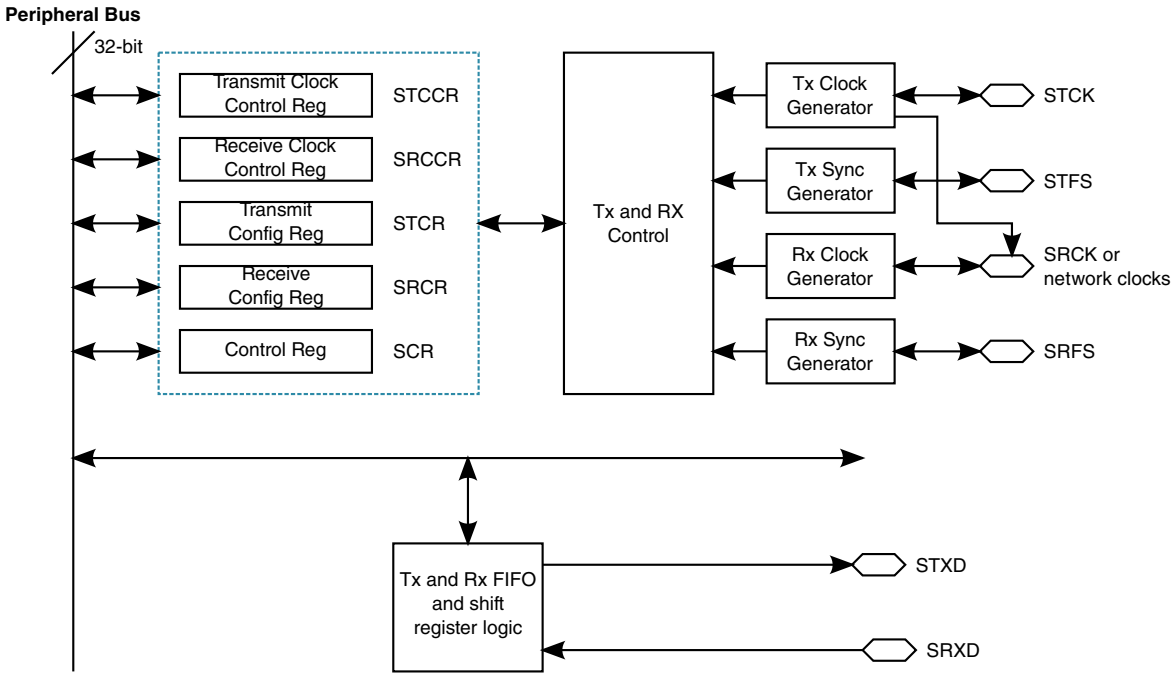


Figure 61-1. SSI Block Diagram

61.1.1 Features

The SSI includes the following features:

- Independent (asynchronous) or shared (synchronous) transmit and receive sections with separate or shared internal/external clocks and frame syncs, operating in Master or Slave mode.
- Normal mode operation using frame sync
- Network mode operation allowing multiple devices to share the port with as many as thirty-two time slots
- Gated Clock mode operation requiring no frame sync
- 2 sets of Transmit and Receive FIFOs. Each of the four FIFOs is 15x32 bits. The two sets of Tx/Rx FIFOs can be used in Network mode to provide 2 independent channels for transmission and reception (this mode is named as two-channel mode in the following descriptions)
- Programmable data interface modes such as I2S, LSB, MSB aligned
- Programmable word length (8, 10, 12, 16, 18, 20, 22 or 24 bits)
- Program options for frame sync and clock generation
- Programmable I2S modes (Master, Slave or Normal). Maximum audio sampling rate is 196kHz. (Note that maximum sampling rate depends on IPG frequency.) Minimum audio sampling rate is 8kHz. Network clock (as an oversampling clock to external device) available as output from SRCK in I2S Master mode

- AC97 support. Max frame rate is 48kHz. Min frame rate is 8kHz.
- Completely separate clock and frame sync selections for the receive and transmit sections. In AC97 standard, the clock is taken from an external source and frame sync is generated internally.
- SSI's system clock (generated inside CCM) can be used in I2S Transmitter Master mode. This system clock is also available as source clock for output SRCK in master mode, when operated in sync mode.
- Programmable internal clock divider
- Time Slot Mask Registers for reduced ARM platform overhead (for both Tx and Rx)
- SSI power-down feature

61.1.2 Modes of Operation

SSI has the following basic operating modes.

- Normal Mode : Asynchronous protocol, Synchronous protocol
  - Normal Mode Transmit
  - Normal Mode Receive
- Network Mode : Asynchronous protocol, Synchronous protocol
  - Network Mode Transmit
  - Network Mode Receive
- Gated Clock Mode : Synchronous protocol only
- I2S Mode
- AC97 Mode
  - AC97 Fixed Mode (SSI.SACNT[1]=0)
  - AC97 Variable Mode (SSI.SACNT[1]=1)

61.2 External Signal Description

61.2.1 Signals Overview

The Synchronous Serial Interface (SSI) is connected through the Digital Audio Multiplexer (AUDMUX). Refer to the AUDMUX chapter for programming details of the various multiplexing options.

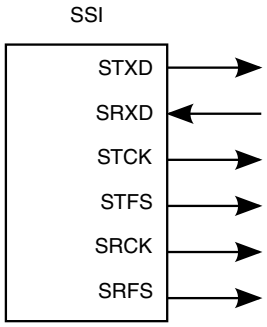
Table 61-1. Off-Chip Block Signals

Name	I/O	Function	Reset State	Pull up
SRCK	I/O	Serial Receive Clock. SRCK can be used as either an input or an output. This clock signal is used by the receiver in asynchronous mode and is always continuous. During synchronous mode, the STCK port is used instead for clocking in data. In SSI synchronous modes, this port can be used as an output for the network clock (oversampling clock) . In I2S master mode, this signal can be used to output the network clock to an external CODEC.	0	Passive
SRFS	I/O	Serial Receive Frame Sync. The SRFS port can be used as either an input or an output. The frame sync is used by the receiver to synchronize the transfer of data. The frame sync signal can be one bit or one word in length and can occur one bit before the transfer of data or right at the transfer of data. If SRFS is configured as an input, the external device should drive SRFS during the rising edge of STCK or SRCK.	0	Passive
SRXD	I	Serial Receive Data. The SRXD port is an input and is used to bring serial data into the Receive Data Shift Register.	-	-
STCK	I/O	Serial Transmit Clock. The STCK port can be used as either an input or an output. This clock signal is used by the transmitter and can be either continuous or gated. During Gated Clock mode, data on the STCK port is valid only during the transmission of data, otherwise it is pulled to the inactive state. In Synchronous mode, this port is used by both the transmit and receive sections.	0	Passive
STFS	I/O	Serial Transmit Frame Sync. The STFS port can be used as either an input or an output. The frame sync is used by the transmitter to synchronize the transfer of data. The frame sync signal can be one bit or one word in length and can occur one bit before the transfer of data or right at the transfer of data. In Synchronous mode, this port is used by both the transmit and receive sections. In Gated Clock mode, frame sync signals are not used. If STFS is configured as an input, the external device should drive STFS during the rising edge of STCK if TSCKP is positive edge triggered. The external device should drive STFS during the falling edge of STCK if TSCKP is negative edge triggered.	0	Passive
STXD	O	Serial Transmit Data. The STXD port is an output and transmits data from the Serial Transmit Shift Register. The STXD port is an output port when data is being transmitted and is disabled between data word transmissions and on the trailing edge of the bit clock after the last bit of a word is transmitted.	0	Passive

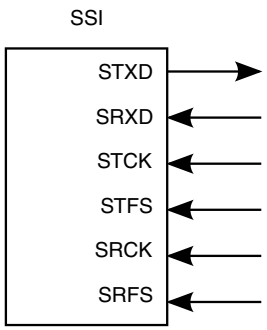
The following figure shows the main SSI configurations. These ports support all transmit and receive functions with continuous or gated clock as shown.

NOTE

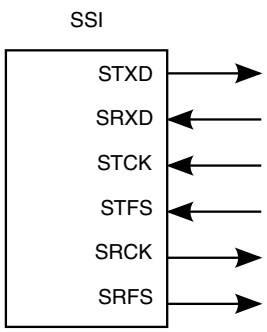
Gated clock implementations do not require the use of the frame sync ports (STFS and SRFS).



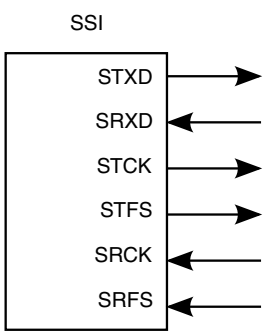
SSI Internal Continuous Clock for TX/RX (RXDIR = 1, TXDIR = 1, RFDIR = 1, TFDIR = 1, SYN = 0)



SSI External Continuous Clock for TX/RX (RXDIR = 0, TXDIR = 0, RFDIR = 0, TFDIR = 0, SYN = 0)



SSI Internal Continuous Clock for RX (RXDIR = 1, TXDIR = 0, RFDIR = 1, TFDIR = 0, SYN = 0)
SSI External Continuous Clock for TX



SSI Internal Continuous Clock for TX (RXDIR = 0, TXDIR = 1, RFDIR = 0, TFDIR = 1, SYN = 0)
SSI External Continuous Clock for RX

Figure 61-2. Asynchronous (SYN=0) SSI Configurations-Continuous Clock

See the following figure for an example of the port signals for an 8-bit data transfer. Continuous and gated clock signals are shown, as well as the bit-length frame sync signal and the word-length frame sync signal.

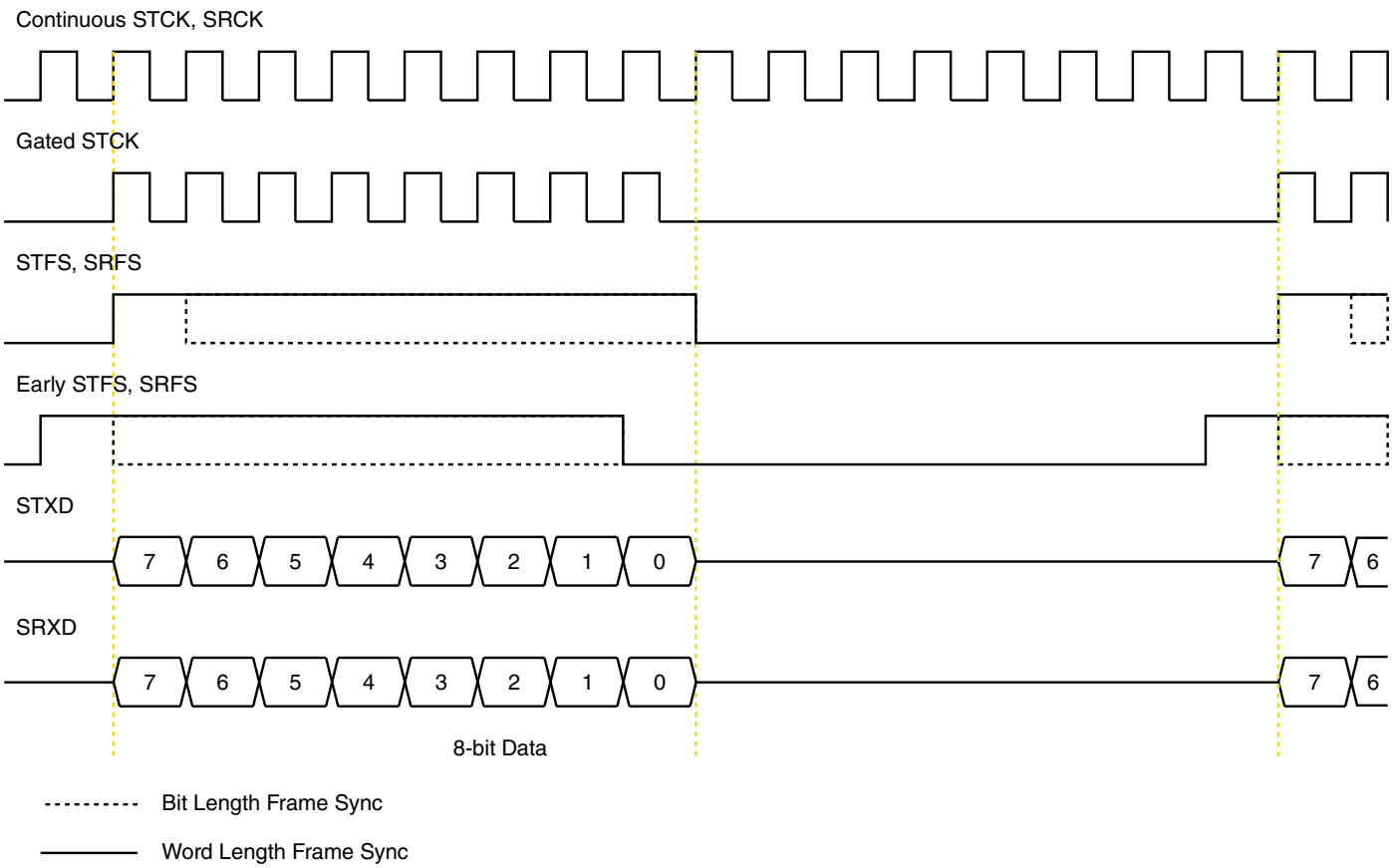


Figure 61-3. Serial Clock and Frame Sync Timing

See the table below for list of clock pin configurations.

Table 61-2. Clock Pin Configurations

SYN	RXDIR	TXDIR	RFDIR	TFDIR	SRCK	STCK	SRFS	STFS
Asynchronous Mode								
0	0	0	0	0	RCK in	TCK in	RFS in	TFS in
0	0	0	0	1	RCK in	TCK in	RFS in	TFS out
0	0	0	1	0	RCK in	TCK in	RFS out	TFS in
0	0	0	1	1	RCK in	TCK in	RFS out	TFS out
0	0	1	0	0	RCK in	TCK out	RFS in	TFS in
0	0	1	0	1	RCK in	TCK out	RFS in	TFS out
0	0	1	1	0	RCK in	TCK out	RFS out	TFS in
0	0	1	1	1	RCK in	TCK out	RFS out	TFS out
0	1	0	0	0	RCK out	TCK in	RFS in	TFS in

Table continues on the next page...

Table 61-2. Clock Pin Configurations (continued)

SYN	RXDIR	TXDIR	RFDIR	TFDIR	SRCK	STCK	SRFS	STFS
0	1	0	0	1	RCK out	TCK in	RFS in	TFS out
0	1	0	1	0	RCK out	TCK in	RFS out	TFS in
0	1	0	1	1	RCK out	TCK in	RFS out	TFS out
0	1	1	0	0	RCK out	TCK out	RFS in	TFS in
0	1	1	0	1	RCK out	TCK out	RFS in	TFS out
0	1	1	1	0	RCK out	TCK out	RFS out	TFS in
0	1	1	1	1	RCK out	TCK out	RFS out	TFS out
Synchronous Mode								
1	0	0	x	0	-	CK in	-	FS in
1	0	0	x	1	-	CK in	-	FS out
1	0	1	x	0	-	CK out	-	FS in
1	0	1	x	1	-	CK out	-	FS out
1	1	0	x	x	-	Gated in	-	-
1	1	1	x	x	-	Gated out	-	-

61.3 Clocks

The table found here describes the clock sources for SSI.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 61-3. SSI Clocks

Clock name	Clock Root	Description
ccm_ssi_clk	ssi_clk_root	Module / system clock for bit clock generation
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

61.4 SSI Transmit FIFO 0 & 1 Registers

The SSI Transmit FIFO registers are 15x32-bit registers. These registers are not directly accessible by the end user. Transmit Shift Register (TXSR) receives its values from these FIFO registers. Transmitted data is first-in-first-out.

When the Transmit Interrupt Enable (TIE) bit in the SIER and either of the Transmit FIFO Empty Enable (TFE0 or 1) bits in the SIER are set, an interrupt is asserted whenever the number of empty slots exceed or are equal to the selected threshold value of corresponding Tx-FIFO. The threshold value is contained in the corresponding Transmit FIFO Watermark (TFWM0 or 1) field in the SSI FIFO Control/Status Register (SFCSR).

61.5 SSI Transmit Shift Register (TXSR)

The SSI Transmit Shift Register (TXSR) is a 24-bit shift register that contains the data being transmitted.

This register is not directly accessible by the end user. When a continuous clock is used, data is shifted out to the Serial Transmit Data (STXD) port by the selected (internal/external) bit clock when the associated (internal/external) frame sync is asserted. When a gated clock is used, data is shifted out to the STXD port by the selected (internal/external) gated clock. The Word Length control bits (WL[3:0]) in the SSI Transmit and Receive Clock Control Register (STCCR) determine the number of bits to be shifted out of the TXSR before it is considered empty and can be written to again. This word length can be 8, 10, 12, 16, 18, 20, 22 or 24 bits. The data to be transmitted occupies the most significant portion of the shift register if TXBIT0 is '0', otherwise it occupies the least significant portion. The unused portion of the register is ignored. Data is always shifted out of this register with the Most Significant Bit (MSB) first when the SHFD bit of the STCR is cleared. If this bit is set, the Least Significant Bit (LSB) is shifted out first. The figures below show the transmitter loading and shifting operation. The figures show the working for some WL values, the same can be extended for other values.

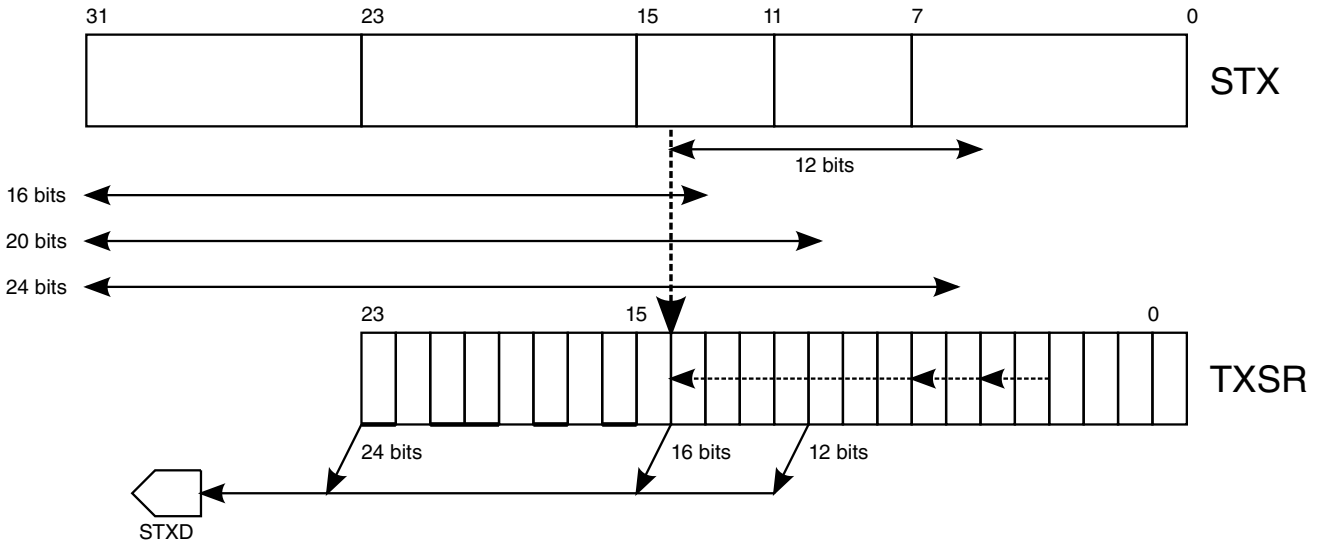


Figure 61-4. Transmit Data Path (TXBIT0=0, TSHFD=0) (MSB Alignment)

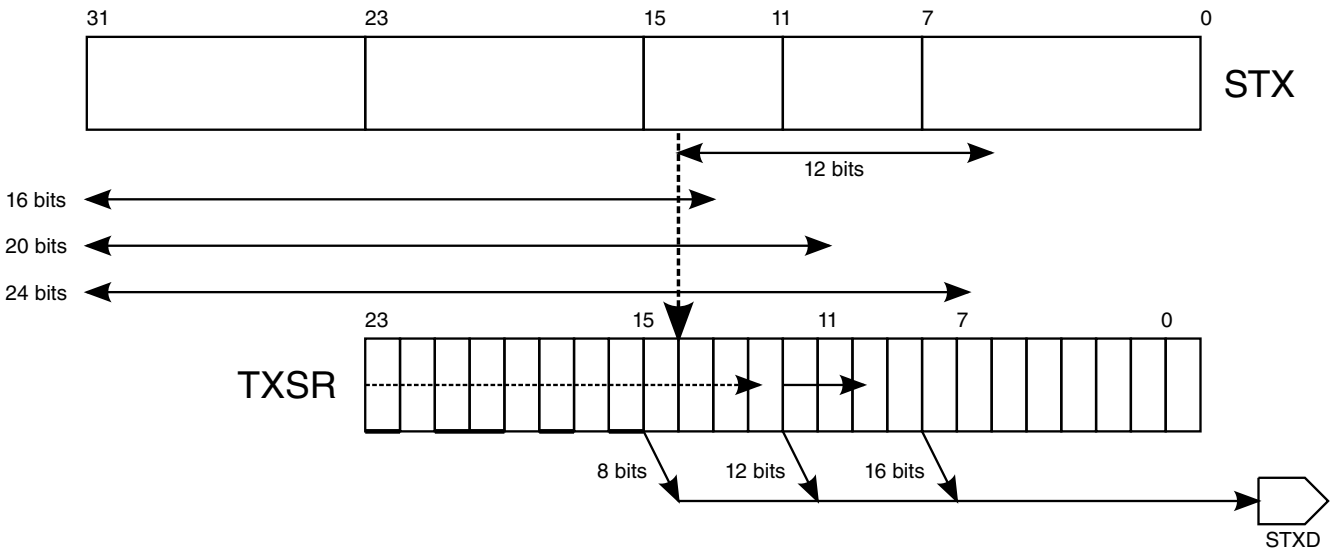


Figure 61-5. Transmit Data Path (TXBIT0=0, TSHFD=1) (MSB Alignment)

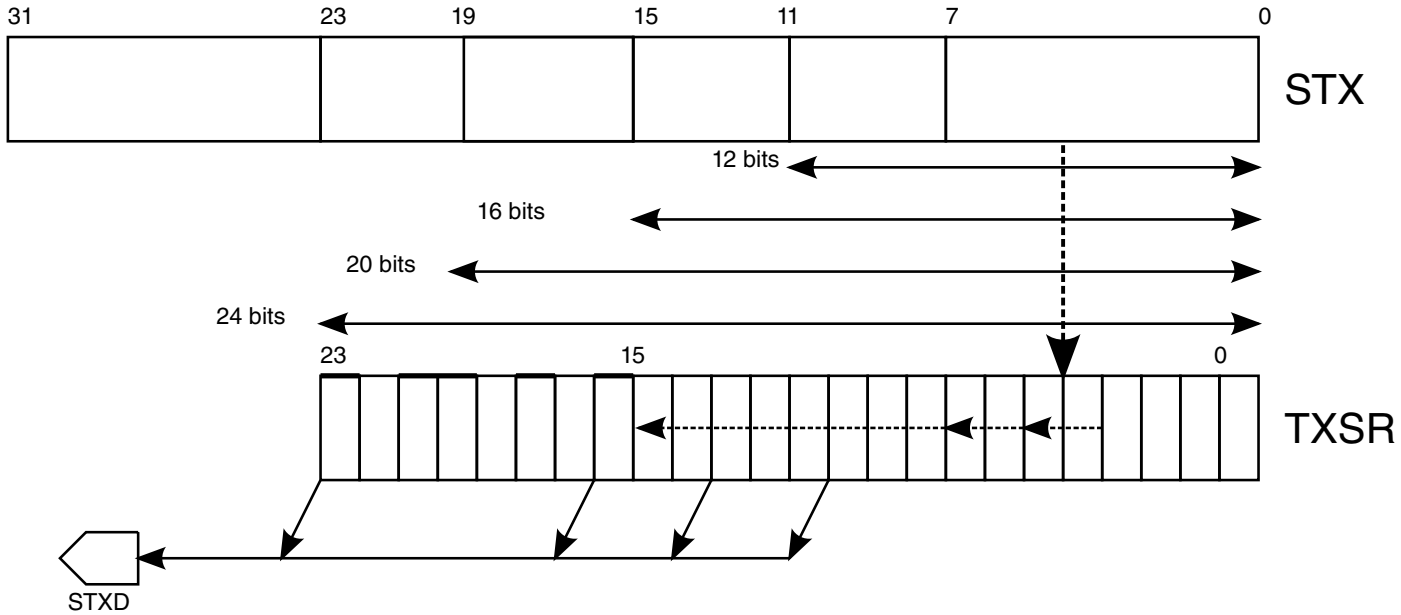


Figure 61-6. Transmit Data Path (TXBIT0=1, TSHFD=0) (LSB Alignment)

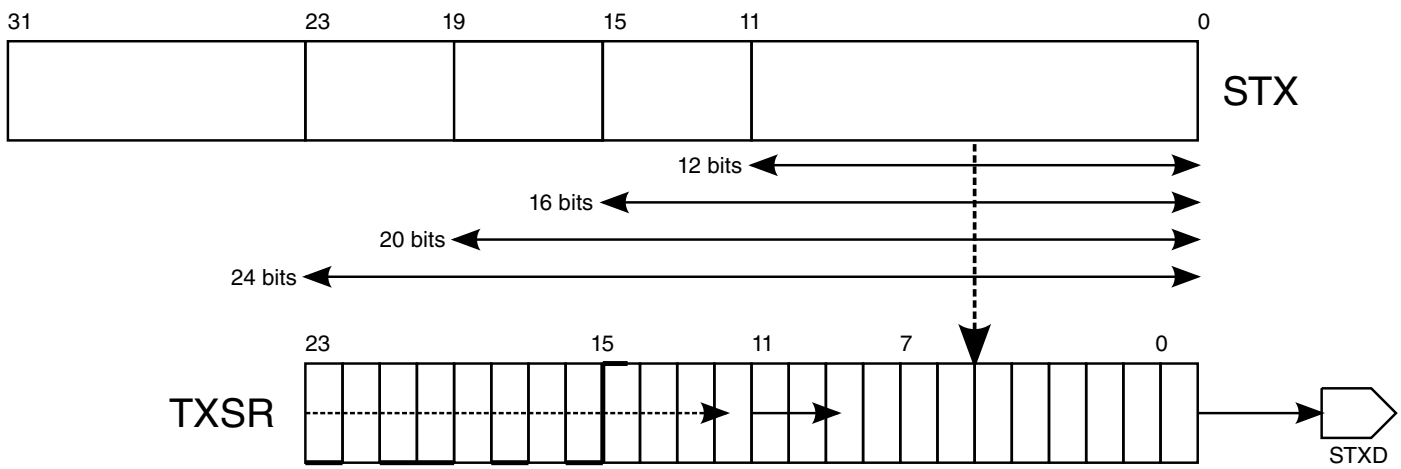


Figure 61-7. Transmit Data Path (TXBIT0=1, TSHFD=1) (LSB Alignment)

61.6 SSI Receive FIFO 0 and 1 Registers

The SSI Receive FIFO registers are 15x32-bit registers. These registers are not directly accessible by the end user. These FIFO registers receive data from the Receive Shift Register (RXSR). Received data is first-in-first-out.

When the Receive Interrupt Enable (RIE) bit in the SIER and either of the Receive FIFO Full Enable (RFF0_EN or RFF1_EN) bits in the SIER are set, an interrupt is asserted whenever the number of full slots exceeds or is equal to the selected threshold value of corresponding Rx-FIFO. The threshold value is contained in the corresponding Receive FIFO Watermark (RFWM0 or 1) field in the SSI FIFO Control/Status Register (SFCSR).

61.7 SSI Receive Shift Register (RXSR)

The SSI Receive Shift Register (RXSR) is a 24-bit, shift register that receives incoming data from the serial receive data SRXD port.

This register is not directly accessible by the end user. When a continuous clock is used, data is shifted in by the selected (internal/external) bit clock when the associated (internal/external) frame sync is asserted. When a gated clock is used, data is shifted in by the selected (internal/external) gated clock. Data is assumed to be received MSB first if the SHFD bit of the SSI.SRCR is cleared. If this bit is set, the data is received LSB first. Data is transferred to the appropriate SSI Receive Data Register (SRX0/1) or Receive FIFOs (if the receive FIFO is enabled and the corresponding SRX is full) after 8, 10, 12, 16, 18, 20, 22 or 24 bits have been shifted in depending on the WL[3:0] control bits. For receiving less than 24 bits of data, LSB bits are appended with zero. The figures below show the receiver loading and shifting operation. These figures show the operation for several values of WL and the same can be extended for other values.

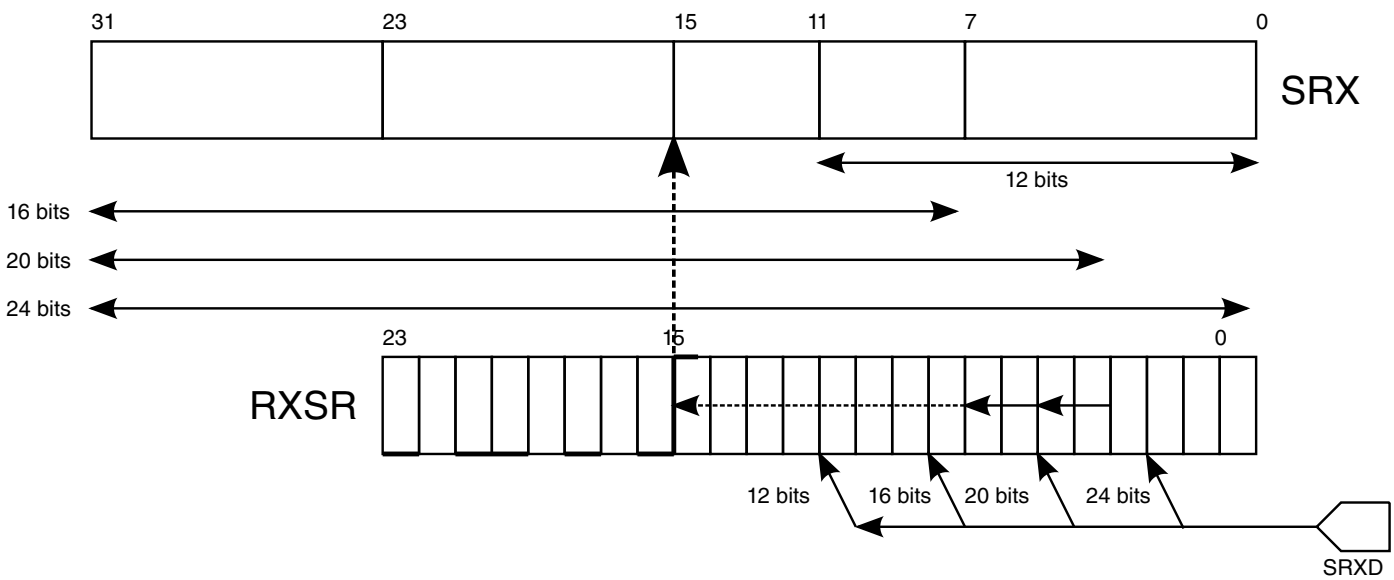


Figure 61-8. Receive Data Path (RXBIT0=0, RSHFD=0) (MSB Alignment)

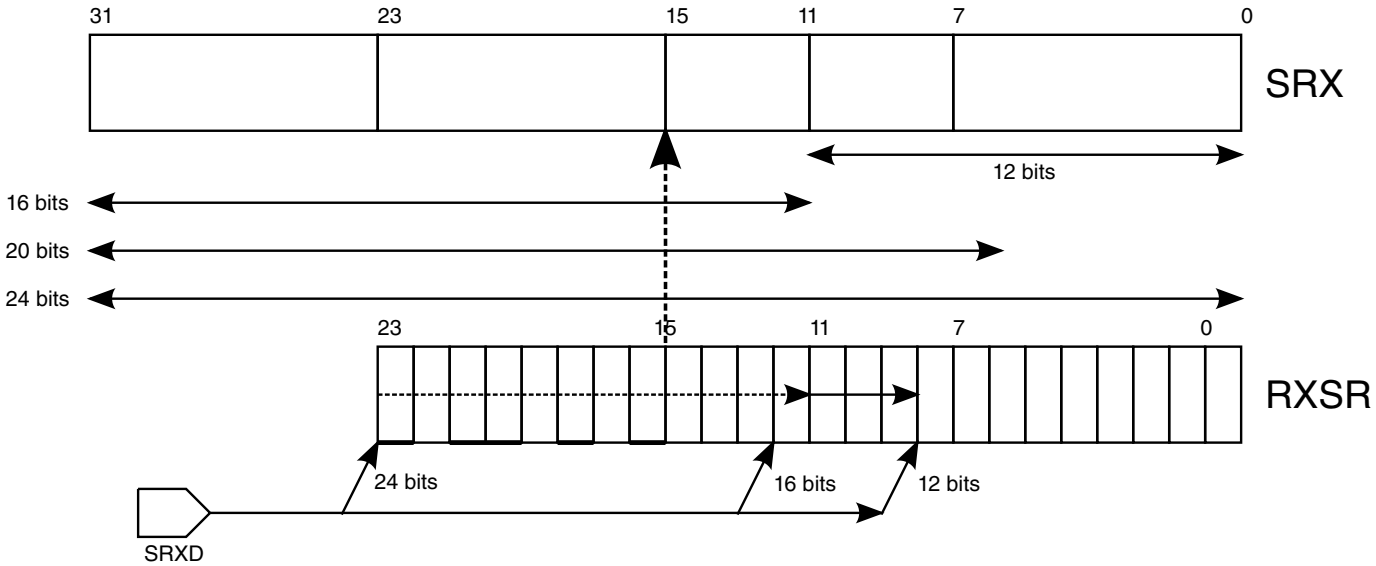


Figure 61-9. Receive Data Path (RXBIT0=0, RSHFD=1) (MSB Alignment)

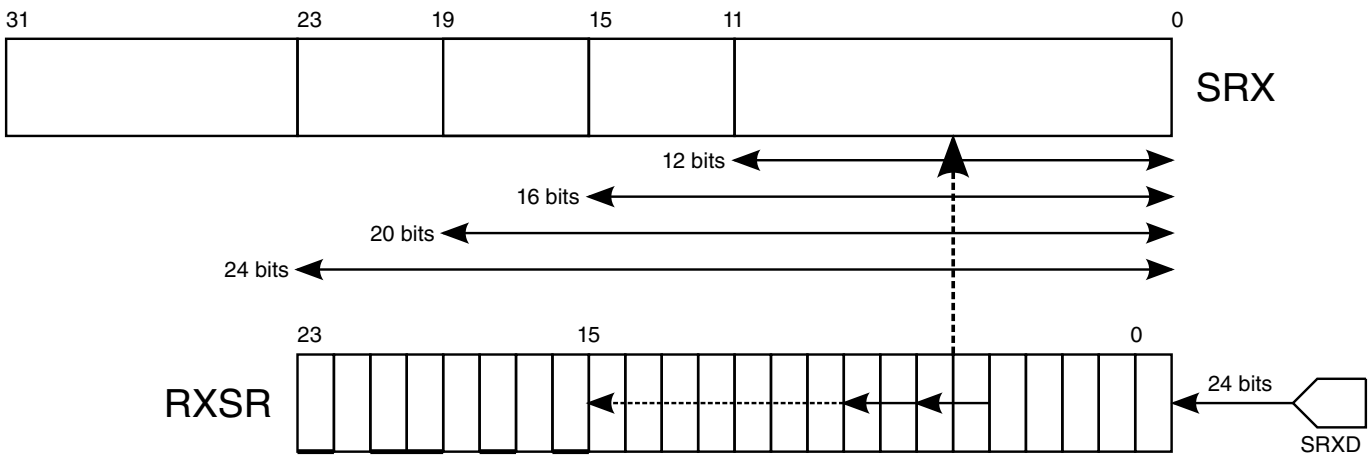


Figure 61-10. Receive Data Path (RXBIT0=1, RSHFD=0) (LSB Alignment)

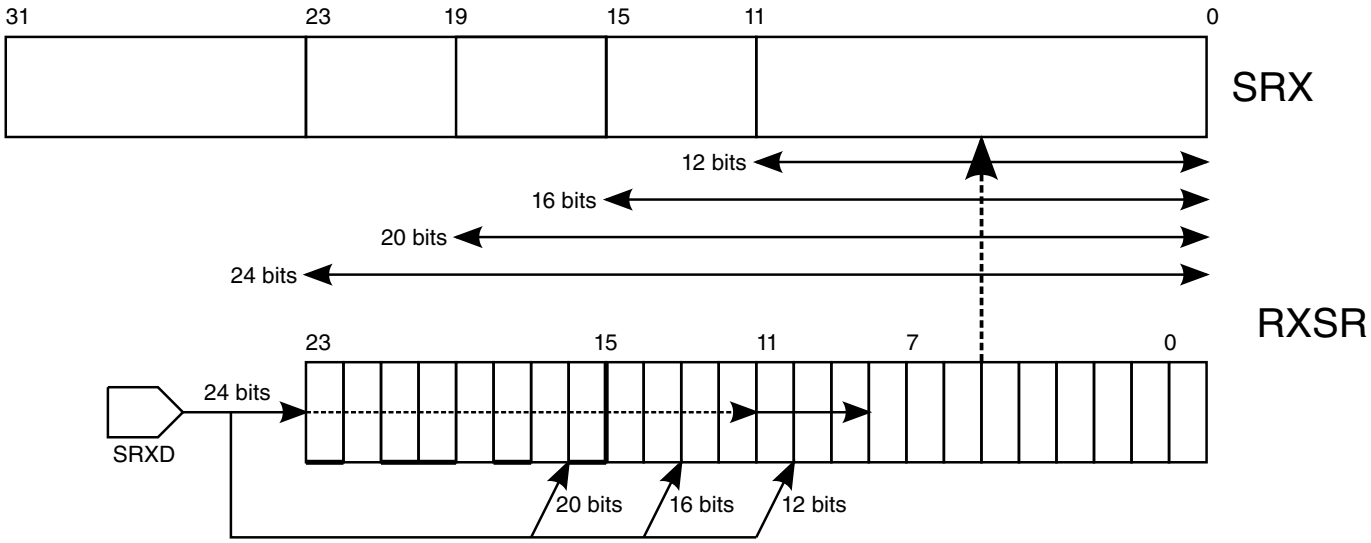


Figure 61-11. Receive Data Path (RXBIT0=1, RSHFD=1) (LSB Alignment)

61.8 Functional Description

This section provides a complete functional description of the block.

61.8.1 Operating Modes

Different modes can be programmed by several bits in the SSI control registers.

See the table below for the list of SSI operating modes and some of the typical applications in which they can be used:

Table 61-4. SSI Operating Modes

TX, RX Sections	Serial Clock	Mode	Typical Application
Asynchronous	Continuous	Normal	Multiple synchronous CODECs
Asynchronous	Continuous	Network	TDM CODEC or DSP networks
Synchronous	Continuous	Normal	Multiple synchronous CODECs
Synchronous	Continuous	Network	TDM CODEC or DSP network
Synchronous	Gated	Normal	SPI-type devices; DSP to ARM platform

The transmit and receive sections of the SSI can be synchronous or asynchronous. In Synchronous mode, the transmitter and the receiver use a common clock and frame synchronization signal. Masking of slots for Transmit and Receive section can differ in

synchronous mode. Also the shifting of data in independent and for receive section depends on RXBIT0 and RSHFD bits in SSI.SRCR register. In Asynchronous mode, the transmitter and receiver each has its own clock and frame synchronization signals.

Normal or Network mode can be selected. In Normal mode, the SSI functions with one data word of I/O per frame. In Network mode, any number from two to thirty-two data words of I/O per frame can be used. Network mode is typically used in time division multiplex networks with other processors or CODECs, allowing interface to time division multiplexed networks without additional logic. Gated clock mode option can be selected in Normal synchronous mode only. During Gated clock mode the clock is not-continuous and runs only during data-transmission. These distinctions result in the basic operating modes that allow the SSI to communicate with a wide variety of devices.

The SSI supports both Normal and Network modes, and these can be selected independently of whether the transmitter and receiver are synchronous or asynchronous. Typically these protocols are used in a periodic manner, where data is transferred at regular intervals, such as at the sampling rate of an external CODEC. Both modes use the concept of a frame. The beginning of the frame is marked with a frame sync when programmed with continuous clock. The frame sync occurs at a periodic interval. The length of the frame is determined by the DC[4:0] bits in either the SSI.SRCCR or SSI.STCCR register, depending on whether data is being transmitted or received. The number of words transferred per frame depends on the mode of the SSI.

In Normal mode, one data word is transferred per frame. In Network mode, the frame is divided into anywhere between two and thirty-two time slots, where in each time slot one data word can optionally be transferred.

Apart from the above basic modes of operation, SSI supports the following modes which require some specific programming.

- I2S mode
- AC97 mode
    - AC97 Fixed mode
    - AC97 Variable mode

In (non-I2S) slave modes (external frame sync), the programmed word length setting of the SSI should be equal to the word length setting of the master. In I2S slave mode, the programmed word length setting of the SSI can be lesser than or equal to the word length setting of the I2S master (external CODEC).

In slave modes, the programmed frame length setting (DC bits) of the SSI can be lesser than or equal to the frame length setting of the master (external CODEC).

The following sections provide detailed descriptions of the above modes.

61.8.1.1 Normal Mode

Normal mode is the simplest mode of the SSI. It is used to transfer data in one time slot per frame.

A time slot is a unit of data and the WL[3:0] bits define the number of bits in a time slot. In Continuous Clock mode, a frame sync occurs at the beginning of each frame. The length of the frame is determined by the following factors:

- The period of the Serial Bit Clock (DIV2, PSR, PM[7:0] bits for internal clock or the frequency of the external clock on the STCK port)
- The number of bits per time slot (WL[3:0] bits)
- The number of time slots per frame (DC[4:0] bits)

If Normal mode is configured with more than one time slot per frame, data is transferred only in the first time slot. No data is transferred in subsequent time slots. In Normal mode, DC[4:0] values corresponding to more than a single time slot in a frame, only results in lengthening of the frame.

61.8.1.1.1 Normal Mode Transmit

The conditions for data transmission from the SSI in Normal mode are:

- SSI Enabled (SSIEN = 1)
- Write data to Transmit Data Register (STX)
- Transmitter Enabled (TE = 1)
- Frame sync active (for continuous clock case)
- Bit clock begins

When the above conditions occur in Normal mode, the next data word is transferred into the Transmit Shift Register (SSI.TXSR) from the Transmit Data Register 0 (SSI.STX0), or from the Transmit FIFO 0 Register, if transmit FIFO 0 is enabled. The new data word is transmitted on arrival of frame-sync preceded by clocks in continuous clock mode. In gated-external mode, data word is transmitted on arrival of frame-sync. In gated-internal mode, data word is transmitted whenever data is available in Tx-FIFO.

If Transmit FIFO 0 is not enabled and the transmit data register empty enable (TDE0_EN) and transmit interrupt enable (TIE) bits are set, transmit interrupt occurs when the word in SSI_STX0 is shifted to Transmit Shift (SSI.TXSR) register for shifting.

If Transmit FIFO 0 is enabled and the transmit fifo empty enable (TFE0_EN) and transmit interrupt enable (TIE) bits are set, transmit interrupt occurs when the number of empty slots in Transmit Fifo 0 exceed or are equal to the selected threshold value i.e.

Transmit Fifo 0 Watermark (TFWM0) value. If transmit FIFO 0 is enabled and filled with data, 15 data words can be transferred before the core must write new data to the SSI.STX0 register.

The STXD port is disabled except during the data transmission period. For a continuous clock, the optional frame sync output and clock outputs are not disabled, even if both receiver and transmitter are disabled.

61.8.1.1.2 Normal Mode Receive

The conditions for data reception from the SSI are:

- SSI enabled (SSIEN = 1)
- Receiver enabled (RE = 1)
- Frame sync active (for continuous clock case)
- Bit clock begins

With the above conditions in Normal mode with a continuous clock, each time the frame sync signal is generated (or detected) a data word is clocked in. With the above conditions and a gated clock, each time the clock begins, a data word is clocked in.

If Receive FIFO 0 is not enabled and Receive Interrupt enable (RIE) and Received Data 0 Ready enable (RDR0_EN) bits are set, receive interrupt occurs when received data word is transferred from the Receive Shift Register (SSI.RXSR) to the Receive Data Register 0 (SSI.SRX0), thus setting the Receive Data Ready 0 (RDR0) flag.

If Receive FIFO 0 is enabled, and Receive Interrupt enable (RIE) and Received Fifo 0 full enable (RFF0_EN) bits are set, receive interrupt occurs when the received data word is transferred to the Receive FIFO 0 and Receive FIFO 0 reaches the selected threshold and results in Receive FIFO Full 0 (RFF0) flag to get set.

The core program has to read the data from the Receive Data Register 0 (SSI.SRX0) (in case Receive FIFO0 is disabled) before a new data word is transferred from the Receive Shift Register (SSI.RXSR), otherwise the Receive Overrun Error 0 (ROE0) bit is set. If receive FIFO 0 is enabled, the Receive Overrun Error 0 (ROE0) bit is set when the Receive FIFO 0 data level reaches the selected threshold and a new data word is ready to be transferred to the Receive FIFO 0.

See the following figure for an illustration of transmitter and receiver timing for an 8-bit word in the first time slot in Normal mode, continuous clock with a late word length frame sync. The Tx Data register is loaded with the data to be transmitted. On arrival of the clock, this data is transferred to the Transmit Shift Register, which gets transmitted on arrival of the frame-sync on the STXD output. Simultaneously, the Receive Shift Register shifts in the received data available on the SRXD input and, at the end of the time slot, this data is transferred to the Rx Data Register.

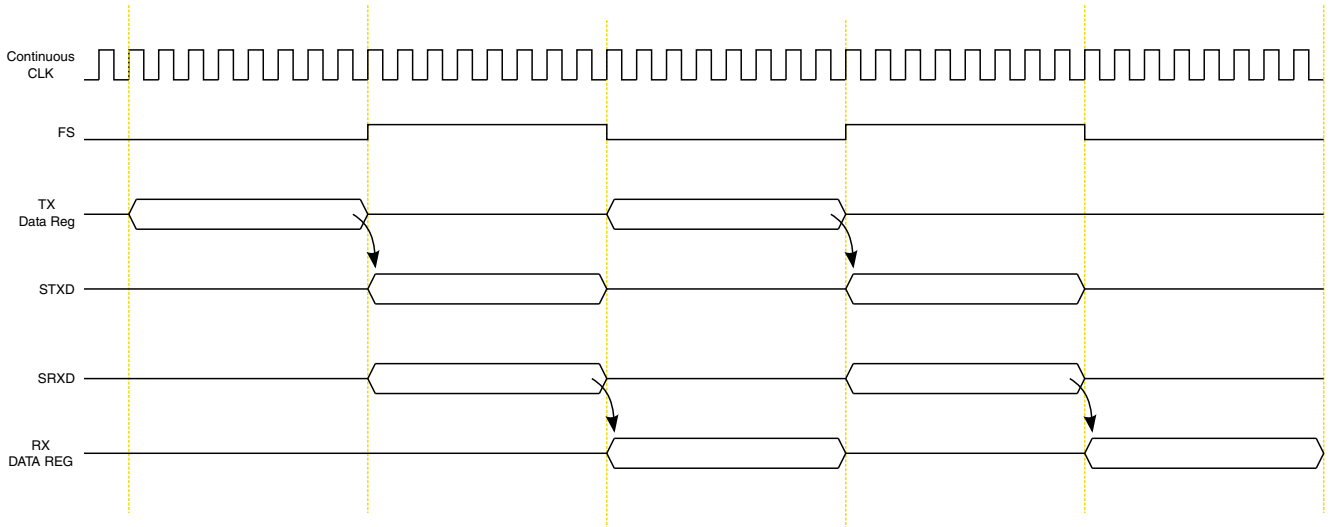


Figure 61-12. Normal Mode Timing - Continuous Clock

The following figure shows a similar case for internal (SSI generates clock) gated clock mode and Figure 61-14 shows a case for external (SSI receives clock) gated clock mode.

NOTE

A pull-down resistor is required in the gated clock case because the clock port is disabled between transmissions.

The Tx Data register is loaded with the data to be transmitted. On arrival of the clock, this data is transferred to the Transmit Shift Register, which gets transmitted on arrival of the frame-sync on the STXD output. Simultaneously, the Receive Shift Register shifts in the received data available on the SRXD input and, at the end of the time slot, this data is transferred to the Rx Data Register. In case of Internal Gated clock mode, the Tx Data line and clock output port are put in the high-impedance state at the end of transmission of the last bit (at the completion of the complete clock cycle), whereas, in External Gated clock mode, the Tx Data line is tri-stated at the last inactive edge of the incoming bit clock (during the last bit in a data word).

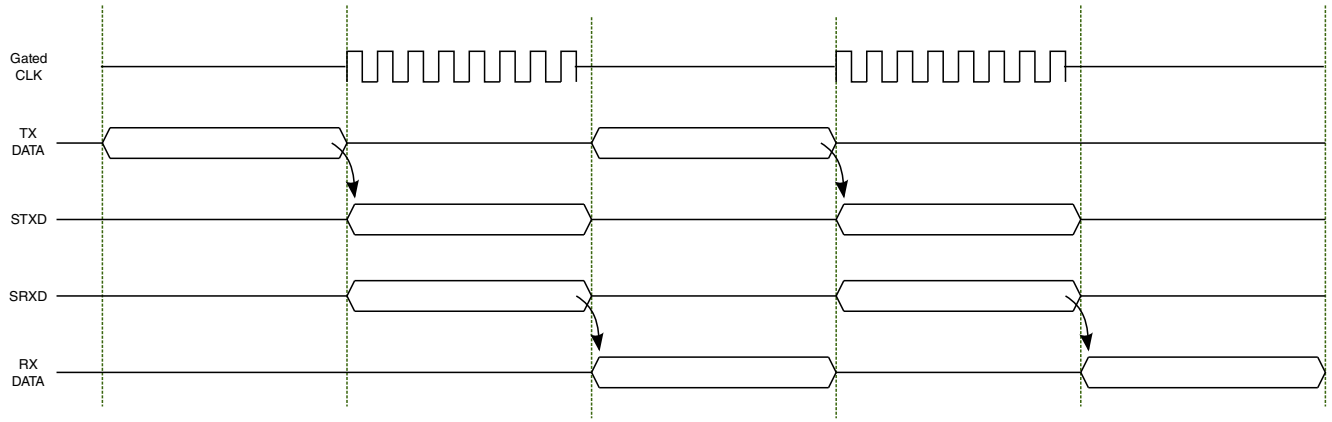


Figure 61-13. Normal Mode Timing - Internal Gated Clock

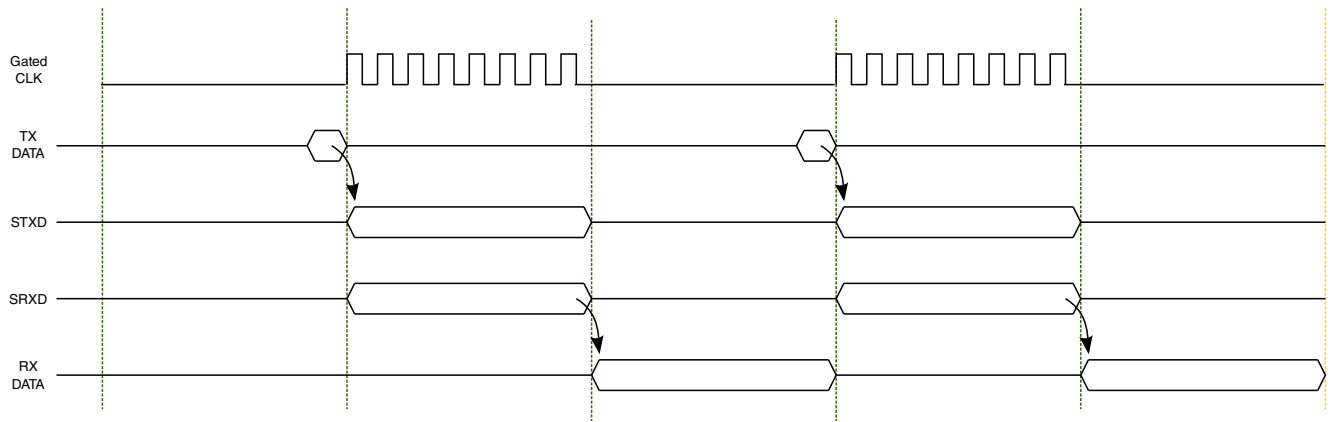


Figure 61-14. Normal Mode Timing - External Gated Clock

61.8.1.2 Network Mode

Network mode is used for creating a Time Division Multiplexed (TDM) network, such as a TDM CODEC network or a network of DSPs.

In Continuous Clock mode, a frame sync occurs at the beginning of each frame. In this mode, the frame is divided into more than one time slot. During each time slot, one data word can be transferred. Each time slot is then assigned to an appropriate CODEC or DSP on the network. The DSP can be a master device that controls its own private network, or a slave device that is connected to an existing TDM network and occupies a few time slots.

The frame sync signal indicates the beginning of a new data frame. Each data frame is divided into time slots and transmission and/or reception of one data word can occur in each time slot (rather than in just the frame sync time slot as in Normal mode). The frame rate dividers, controlled by the DC[4:0] bits, select two to thirty-two time slots per frame. The length of the frame is determined by the following factors:

- The period of the serial bit clock (PSR, PM[7:0] bits for internal clock, or the frequency of the external clock on the STCK port)
- The number of bits per sample (WL[3:0] bits)
- The number of time slots per frame (DC[4:0] bits)

In Network mode, data can be transmitted in any time slot. The distinction of the Network mode is that each time slot is identified with respect to the frame sync (data word time). This time slot identification allows the option of transmitting data during the time slot by writing to the STX registers or ignoring the time slot as determined by SSI.STMSK register bits. The receiver is treated in the same manner and received data is only transferred to the receive data register/fifo if the corresponding time slot is enabled (through SSI.SRMSK).

By utilizing the SSI.STMSK and SSI.SRMSK registers, software only has to service the SSI during valid time slots. This eliminates any overhead associated with unused time slots. See SSI Memory Map/Register Definition for more information on SSI.STMSK and SSI.SRMSK.

In the Two-Channel mode of operation, the second set of Transmit and Receive FIFOs and Data Registers are used to create two separate channels. These channels are completely independent, with a their own set of Core interrupts and DMA requests, which are identical to the ones available for the default channel. In this mode, data is transmitted/received in enabled time slots alternately from/to FIFO 0 and FIFO 1, starting from FIFO 0. The first data word is taken from FIFO 0 and transmitted in the first enabled time slot and subsequently, data is loaded from FIFO 1 and FIFO 0 alternately and transmitted. Similarly, the first received data is sent to FIFO 0 and subsequent data is sent to FIFO 1 and FIFO 0 alternately. Time slots can be selected through the Transmit and Receive Time Slot Mask registers (SSI.STMSK and SSI.SRMSK). For using this mode of operation, the TCH_EN bit (SCR[8]) needs to be set.

61.8.1.2.1 Network Mode Transmit

The transmit portion of SSI is enabled when the SSIEN and the TE bits in the SSI.SCR are both set. However, for continuous clock, when the TE bit is set, the transmitter is enabled only after detection of a new frame sync (transmission starts from the next frame boundary).

Normal start-up sequence for transmission is to perform the following:

1. Enable Network Mode.
2. Enable SSI
3. Write the data to be transmitted to the SSI.STX register. This clears the TDE flag
4. Set the TE bit to enable the transmitter on the next frame boundary (for continuous clock case).
5. Enable transmit interrupts.

(Alternatively, the programmer may decide not to transmit in a time slot by configuring the STMSK.) TDE flag is set as data is shifted from SSI.STX register to TXSR, but the STXD port remains disabled during the time slots. When the next frame sync is detected or generated (continuous clock), the data word in TXSR and is shifted out (transmitted). When the SSI.STX register is empty, the TDE bit is set, which causes a transmitter interrupt (in case the FIFO is disabled) to be sent if the TIE bit is set. Software can poll the TDE bit or use interrupts to reload the STX register with new data for the next time slot. Failing to reload the SSI.STX register before the TXSR is finished shifting (empty) causes a transmitter underrun and the TUE error bit is set. In case the FIFO is enabled, the TFE flag is set in accordance with the watermark setting and this flag causes the transmitter interrupt to occur.

The operation of clearing the TE bit disables the transmitter after completion of transmission of the current frame. Setting the TE bit enables transmission from the next frame. During that time the STXD port is disabled. The TE bit should be cleared after the TDE bit is set to ensure that all pending data is transmitted.

To summarize, the Network mode transmitter generates interrupts every enabled time slot (when FIFO is disabled) and requires the core program to respond to each enabled time slot. These responses from the core are one of the following:

- Write data in data register to enable transmission in the next time slot.
- Configure the time slot register to disable transmission in the next time slot (unless time slot is already masked by SSI.STMSK register bit).
- Do nothing-transmit underrun occurs at the beginning of the next time slot and the previous data is re-transmitted.

In the Two-Channel mode of operation, both the channels (Data Registers, FIFOs, Interrupts and DMA requests) operate in the same manner, as described above. The only difference in case of the second channel is that the Interrupts related to this channel are generated only in case this mode of operation is selected (TDE1 is low by default).

61.8.1.2.2 Network Mode Receive

The receiver portion of the SSI is enabled when both the SSIEN and the RE bits in the SSI.SCR are set. However, the receive enable only takes place during that time slot if RE is enabled before the second to last bit of the word. If the RE bit is cleared, the receiver is disabled at the end of the current frame.

SSI is capable of finding the start of the next frame automatically. When the word is completely received, it is transferred to the SRX register, which sets the RDR bit (Receive Data Ready). Setting the RDR bit causes a receive interrupt to occur if the receiver interrupt is enabled (the RIE bit is set) and (Receive data ready enable) RDR_EN bit is set. The second data word (second time slot in the frame), begins shifting in immediately after the transfer of the first data word to the SSI.SRX register. The core program has to read the data from the Receive Data Register (which clears RDR) before the second data word is completely received (ready to transfer to RX data register) or a receive overrun error occurs (the ROE bit is set).

An interrupt can occur after the reception of each enabled data word or the programmer can poll the RDR flag. The core program response can be one of the following:

- Read RX and use the data.
- Read RX and ignore the data.
- Do nothing-the receiver overrun exception occurs at the end of the current time slot.

NOTE

For a continuous clock, the optional frame sync output and clock output signals are not affected, even if the transmitter or receiver is disabled. TE and RE do not disable the bit clock or the frame sync generation. To disable the bit clock and the frame sync generation, the SSIEN bit in the SSI.SCR can be cleared or TFR_CLK_DIS/RFR_CLK_DIS bits can be set, or the port control logic external to the SSI (for example, in the IOMUXC) can be re configured.

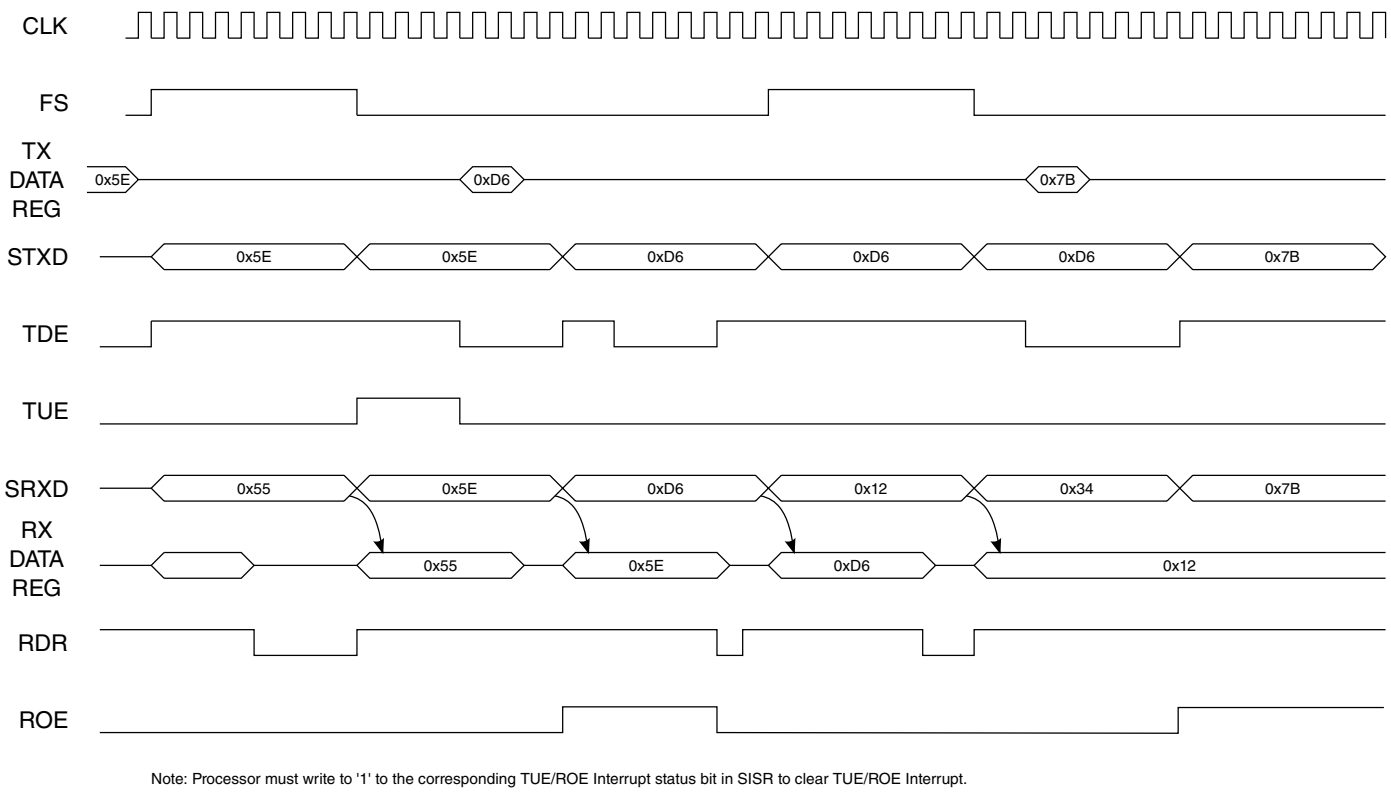
In the Two-Channel mode of operation, both the channels (Data Registers, FIFOs, Interrupts and DMA requests) operate in the same manner, as described above. The only difference in case of the second channel is that the Interrupts related to this channel are generated only in case this mode of operation is selected.

The transmitter and receiver timing for an 8-bit word with continuous clock, FIFO disabled, three words per frame sync in Network mode is shown in the figure below).

NOTE

The transmitter repeats the value 0x5E because of an underrun condition

For the receive section, data received on the SRXD pin gets transferred to the Rx Data register at the end of each time slot. If the FIFO is disabled, the RDR flag gets set and causes a receiver interrupt if RE, RIE and RDR_EN bits are set. If the FIFO is enabled, then the RFF flag is used for interrupt generation (this flag is set in accordance with the watermark settings). Here all time slots are enabled. The receive data ready flag is set after reception of the first data (0x55). Since the flag is not cleared (Rx Data Register is not read by core), the Receive Overrun Error (ROE) flag is set on reception of the next data (0x5E). ROE flag is cleared on writing '1' to the corresponding interrupt status bit in SSI Status Register.



Note: Processor must write to '1' to the corresponding TUE/ROE Interrupt status bit in SISR to clear TUE/ROE Interrupt.

Figure 61-15. Network Mode Timing - Continuous Clock

61.8.1.3 Gated Clock Mode

Gated Clock mode is often used to connect to SPI-type interfaces on Micro controller Units (MCUs) or external peripheral chips. In Gated Clock mode, the presence of the clock indicates that valid data is on the STXD or SRXD ports.

For this reason, no frame sync is needed in this mode. Once transmission of data has completed, the clock is pulled to the inactive state. Gated clocks are allowed for both the transmit and receive sections with either internal or external clock in Normal mode. Gated clocks are not allowed in Network mode. See Table 61-2 ("Clock Pin Configurations") for SSI configuration for gated-mode operation.

The clock runs when the TE bit and/or the RE bit are appropriately enabled. For the case of internally generated clock, all internal bit clocks, word clocks, and frame clocks continue to operate. When a valid time slot occurs (such as the first time slot in Normal mode), the internal bit clock is enabled onto the appropriate clock port. This allows data to be transferred out in periodic intervals in Gated Clock mode. With an external clock, the SSI waits for a clock signal to be received. Once the clock begins, valid data is shifted in. Care should be taken to clear all DC bits (0x00000) when SSI is used in Gated mode. In gated mode of operation the TFS, RFS, TLS, RLS, TFRC and RFRC bits of SSI.AISR register are not generated.

For Gated clock operated in external clock mode, a proper clock signalling must be applied to the SSI STCK in order for it to function properly. When TSCKP is 0, CLK_IST value should be 1. When TSCKP is 1, CLK_IST value should be 0. If the SSI uses rising edge transition to clock data (TSCKP=0) and the falling edge transition to latch data (RSCKP=0), the clock must be in an active low state when idle. If the SSI uses falling edge transition to clock data (TSCKP=1) and the rising edge transition to latch data (RSCKP=1), the clock must be in a active high state when idle. The figures below illustrate the different edge clocking/latching.

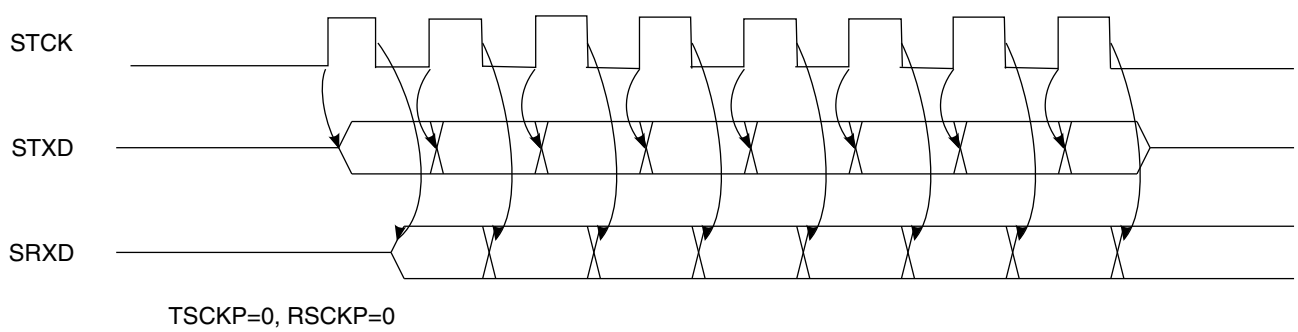


Figure 61-16. Internal Gated Mode Timing - Rising Edge Clocking / Falling Edge Latching

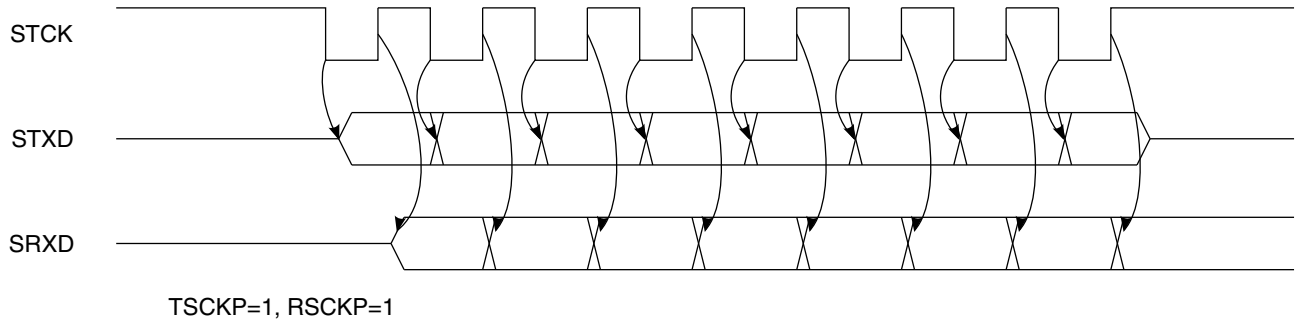


Figure 61-17. Internal Gated Mode Timing - Falling Edge Clocking / Rising Edge Latching

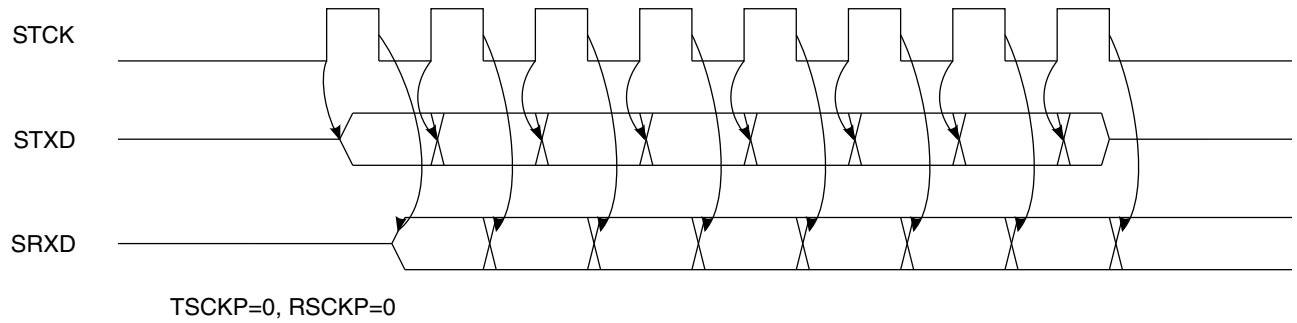


Figure 61-18. External Gated Mode Timing - Rising Edge Clocking / Falling Edge Latching

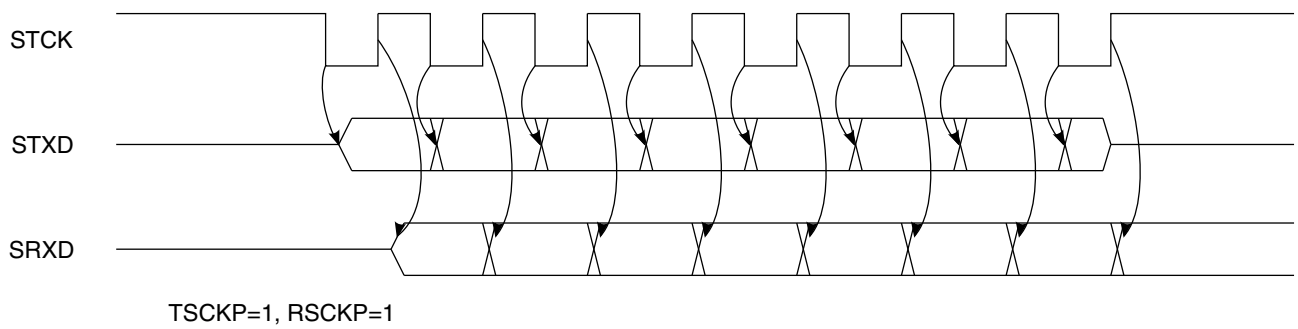


Figure 61-19. External Gated Mode Timing - Falling Edge clocking / Rising Edge Latching

The bit clock ports must be kept free of timing glitches. If a single glitch occurs, all ensuing transfers will be out of synchronization.

In case of External Gated Mode, even though the Tx Data line is put in the high-impedance state at the last non-active edge of the bit clock, the round trip delay should be sufficient to take care of hold time requirements at the external receiver.

61.8.1.4 I2S Mode

The SSI is compliant to the Inter-IC Sound (I2S) bus specification from Philips Semiconductors (February 1986, Revised June 5, 1996). For more information on I2S, refer to the latest version of NXP Semiconductor's I2S specification.

See the following figure for an illustration of the basic I2S protocol timing.

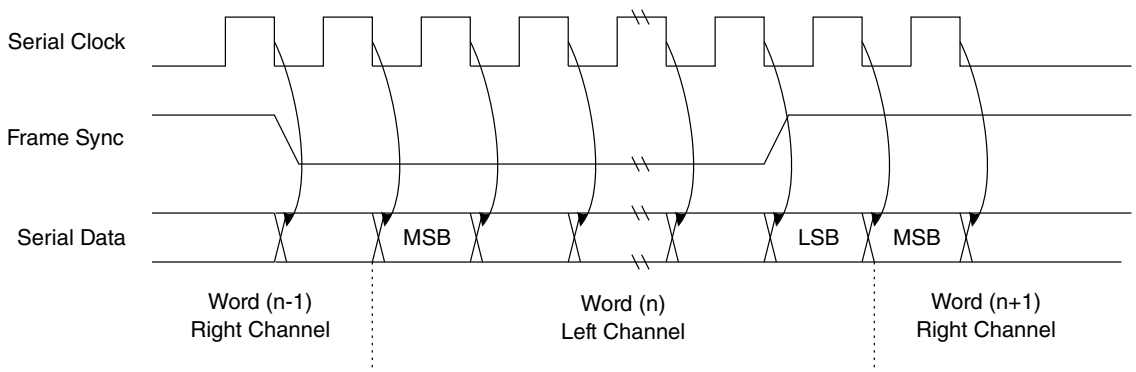


Figure 61-20. I2S Mode Timing - Serial Clock, Frame Sync and Serial Data

Select I2S mode using the options listed in the table below.

Table 61-5. I2S Mode Selection

I2S_MODE[1]	I2S_MODE[0]	Mode Type
0	0	Normal mode
0	1	I2S master mode
1	0	I2S slave mode
1	1	Normal mode

In normal mode operation, no register bits are forced to any particular state internally and the user can program the SSI to work in any operating condition.

When I2S modes are entered (I2S master (01) or I2S slave (10)), the following settings are recommended:

- Sync mode (SSI_SCR[4] =1)
- Tx shift direction: MSB transmitted first (SSI_STCR[4]=0)
- Rx shift direction: MSB received first (SSI_SRCR[4]=0)
- Tx data clocked at falling edge of the clock (SSI_STCR[3]=1)
- Rx data latched at rising edge of the clock (SSI_SRCR[3]=1)
- Tx frame sync active low (SSI_STCR[2]=1)
- Rx frame sync active low (SSI_SRCR[2]=1)

- Tx frame sync initiated one bit before data is transmitted (SSI_STCR[0]=1)
- Rx frame sync initiated one bit before data is received (SSI_SRCR[0]=1)
- TX Frame Rate should be 2 (SSI_STCCR[12:8] = 1)
- RX Frame Rate should be 2 (SSI_SRCCR[12:8] = 1)

In I2S master mode (SSI_SCR[6:5]=01), the following additional settings are recommended:

- TXDIR bit (SSI_STCR[5]) set to 1 to select internal generated bit clock
- TFDIR bit (SSI_STCR[6]) set to 1 to select internal generated frame sync

In I2S master mode(SSI.SCR[6:5]=01), the following settings are internally overridden by the hardware:

- Network mode is selected (SSI_SCR[3]=1)
- Tx frame sync length set to one-word-long-frame (SSI_STCR[1]=0)
- Rx frame sync length set to one-word-long-frame (SSI_SRCR[1]=0)
- Tx shifting with respect to bit 0 of TXSR (SSI.STCR[9]=1)
- Rx shifting with respect to bit 0 of RXSR (SSI.SRCR[9]=1)

The user needs to set the following control bits to configure the bit clock and frame sync:

- PM (SSI_STCCR[7:0])
- PSR (SSI_STCCR[17])
- DIV2(SSI_STCCR[18])
- WL (SSI_STCCR[16:13])
- DC (SSI_STCCR[12:8])

The word length is fixed to 32 in I2S Master mode and the WL bits determine the number of bits that will contain valid data (out of the 32 transmitted/received bits in each channel).

In I2S slave mode(SSI_SCR[6:5]=10), the following additional settings are recommended:

- TXDIR bit(SSI_STCR[5]) set to 0 to select external generated bit clock
- TFDIR bit(SSI_STCR[6]) set to 0 to select external generated frame sync

In I2S slave mode(SSI_SCR[6:5]=10), the following settings are internally overridden by the hardware:

- Normal mode is selected (SSI_SCR[3]=0)
- Tx frame sync length set to one-bit-long-frame (SSI_STCR[1]=1)
- Rx frame sync length set to one-bit-long-frame (SSI_SRCR[1]=1)
- Tx shifting with respect to bit 0 of TXSR (SSI_STCR[9]=1)
- Rx shifting with respect to bit 0 of RXSR (SSI_SRCR[9]=1)

The user needs to set the following control bits to configure the data transmission:

- WL (SSI_STCCR[16:13])
- DC (SSI_STCCR[12:8])

The word length is variable in I2S slave mode and the WL bits determine the number of bits that will contain valid data. The actual word length is determined by the external CODEC. The external I2S Master still sends frame sync according to the I2S protocol (early, word wide and active low), the SSI internally operates so that each frame sync transition is the start of a new frame (the WL bits determine the number of bits to be transmitted/received). After one data word has been transferred, the SSI waits for the next frame sync transition to start operation in the next time slot. Transmit (STMSK) and receive (SRMSK) mask bits should not be used in I2S Slave mode of operation. Masking is supported only for network mode of operation.

61.8.1.5 AC97 Mode

In AC97 mode of operation, the SSI transmits a 16-bit Tag Slot at the start of a frame and the rest of the slots (in that frame) are all 20-bits wide.

The same sequence is followed while receiving data. Refer to the AC97 specification for details regarding transmit and receive sequences and data formats.

NOTE

The Audio Codec specification released in 1997 [AC '97] defines the Architecture and Digital Interface, specifically designed for implementing audio and modem I/O functionality in personal computers. Companion specifications include the Modem Codec [MC '97], and the combined Audio/Modem Codec standard [AMC '97]. The current version of AC '97 was produced in 2002. The AC-97 specification defines a recommended 48-pin QFP IC package.

Note that the SSI only has one RxDATA pin so the SSI can only support one codec. Secondary codecs are not supported.

When AC97 mode is enabled, the following settings are internally overridden by the hardware. The programmed register values are not changed by entering AC97 mode but they no longer apply to the block's operation. Writing to the programmed register fields will update their values; these updates can be seen by reading back the register fields. However, these settings will not take effect until AC97 mode is turned off.

The register bits within the bracket are the equivalent settings:

- Sync mode is entered (SSI.SCR[4] =1)
- Network mode is selected (SSI.SCR[3]=1)
- Tx shift direction is MSB transmitted first (SSI.STCR[4]=0)
- Rx shift direction is MSB received first (SSI.SRCR[4]=0)
- Tx data is clocked at rising edge of the clock (SSI.STCR[3]=0)
- Rx data is latched at falling edge of the clock (SSI.SRCR[3]=0)
- Tx frame sync is active high (SSI.STCR[2]=0)
- Rx frame sync is active high (SSI.SRCR[2]=0)
- Tx frame sync length is one-word-long-frame (SSI.STCR[1]=0)
- Rx frame sync length is one-word-long-frame (SSI.SRCR[1]=0)
- Tx frame sync initiated one bit before data is transmitted (SSI.STCR[0]=1)
- Rx frame sync initiated one bit before data is received (SSI.SRCR[0]=1)
- Tx shifting with respect to bit 0 of TXSR (SSI.STCR[9]=1)
- Rx shifting with respect to bit 0 of RXSR (SSI.SRCR[9]=1)
- Tx FIFO is enabled (SSI.STCR[7]=1)
- Rx FIFO is enabled (SSI.SRCR[7]=1)
- TFDIR bit (SSI.STCR[6]) is forced to 1 internally to select internal generated frame sync
- TXDIR bit (SSI.STCR[5]) is forced to 0 internally to select external generated bit clock

Any alteration of these bits individually will not affect the operational conditions of the SSI unless AC97 mode is deselected.

Hence, the only control bits needed to be set by the user to configure the data transmission/reception are the WL (SSI.STCCR[16:13]) and DC (SSI.STCCR[12:8]) bits. In AC97 mode, the WL bits can only legally take the values corresponding to 16-bit (truncated data) or 20-bit time slots. In case WL bits are set to select 16-bit time slots, the SSI pads the transmit data (four least significant bits) with zeros and while receiving, stores only the most significant 16 bits in the Rx FIFO.

Follow the sequence for programming the SSI to work in AC97 mode:

1. Program the WL bits to a value corresponding to either 16 or 20 bits. The WL bit setting is only for the data portion of the AC97 frame (Slots #3 through #12). The Tag slot (Slot #0) is always 16 bits wide and the Command Address and Command Data slots (Slots #1 and #2) are always 20 bits wide.
2. Select the number of time slots by programming the DC bits. For AC97 operation, DC bits should be set to a value of '0xC', resulting in 13 time slots per frame.
3. Write data to be transmitted, in Tx FIFO 0 (through Tx Data Register 0) and Tx FIFO 1 while using Two-Channel Mode (TCH_EN = 1).
4. Program the FV, TIF, RD, WR and FRDIV bits in SSI.SACNT register

5. Update the contents of SSI.SACADD, SSI.SACDAT and SSI.SATAG (for Fixed mode only) registers
6. Enable the AC97 mode of operation (AC97EN bit in SSI.SACNT register)

Once the SSI starts transmitting and receiving data (after being configured in AC97 mode), the programmer needs to service the interrupts, as and when they are raised (updates to command address/data or tag registers, reading of received data and writing more data for transmission). Further details regarding fixed and variable mode implementation are provided in the following sections.

While using AC97 in Two-Channel Mode (TCH_EN=1), it is recommended that the received tag is not stored in the Rx FIFO (TIF=0). In case the programmer needs to update the SSI.SATAG register and also issue a RD/WR command (in a single frame), it is recommended that the SSI.SATAG register be updated prior to issuing a RD/WR command.

61.8.1.5.1 AC97 Fixed Mode (SSI.SACNT[1]=0)

In fixed mode of operation, SSI transmits in accordance with the AC97 Frame Rate Divider bits (i.e. FRDIV in SACNT) which decides the number of frames for which the SSI should be idle, after operating for one frame.

In a valid frame, TAG Value (written by Core) will be transmitted in Slot #0, Command Address will be transmitted in Slot #1 in case of RD/WR Command, and Command Data will be transmitted in Slot #2 in case of a WR Command. The data from TX-FIFO is transmitted in Slot #3 - Slot #12 depending on the valid slots indicated by the TAG value.

While receiving, bit 15 of the TAG Value (Slot #0) is checked to see if the CODEC is ready. If this bit is set, the frame is received. The received TAG provides the information about Slots containing valid data. The the corresponding TAG bit is valid, the Command Address (Slot #1) and Command Data (Slot #2) values are stored in the corresponding registers. The received data (Slot #3 - Slot #12) is then stored in the Rx-FIFO (for valid slots).

61.8.1.5.2 AC97 Variable Mode (SSI.SACNT[1]=1)

In Variable Mode, the transmit slots which should contain data in the current frame are determined by SLOTREQ bits received in the previous frame. While receiving, if the CODEC is ready, the frame is received and the SLOTREQ bits (contained in Slot #1) are stored for scheduling transmission in the next frame.

The SSI.SACCST, SSI.SACCEN and SSI.SACCDIS registers helps in determining which transmit slots are active. This information is used to ensure that SSI does not transmit data for powered-down/inactive channels.

61.8.2 External Frame and Clock Operation

When applying external frame sync and clock signals to SSI, there should be at least 4-bit clock cycles between the enabling of the transmit or receive section and the rising edge of the corresponding frame sync signal.

The transition of STFS or SRFS should be synchronized with the rising edge of external clock signal, STCK or SRCK.

61.8.2.1 Data Alignment Formats Supported

The SSI supports three data formats in order to provide flexibility with handling data. These formats dictate how data is written to (and read from) the data registers. Therefore, data can appear in different places in SSI.STX0/1 and SSI.SRX0/1 based on the data format and the number of bits per word. Independent data formats are supported for both the transmitter and receiver (that is, the transmitter and receiver can use different data formats).

The supported data formats are:

- MSB alignment
- LSB alignment
  - Zero-extended (receive data only)
  - Sign-extended (receive data only)

With MSB alignment, the most significant byte is bits 31 through 24 of the data register if the word length is larger than or equal to 16 bits. If the word length is less than 16 bits and MSB alignment is chosen, the most significant byte is bits 15 through 8. With LSB alignment, the least significant byte is bits 7 through 0. Data alignment is controlled by the TXBIT0 bit in the SSI.STCR and the RXBIT0 bit in the SSI.SRCR. See the table below for the bit assignment for all the data formats supported by the SSI.

Table 61-6. Data Alignment

Format	Bit Number																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8-bit LSB Aligned																									7	6	5	4	3	2	1	0
8-bit MSB Aligned																	7	6	5	4	3	2	1	0								
10-bit LSB Aligned																							9	8	7	6	5	4	3	2	1	0
10-bit MSB Aligned																	9	8	7	6	5	4	3	2	1	0						

Table continues on the next page...

Table 61-6. Data Alignment (continued)

12-bit LSB Aligned																					11	10	9	8	7	6	5	4	3	2	1	0
12-bit MSB Aligned																	11	10	9	8	7	6	5	4	3	2	1	0				
16-bit LSB Aligned																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit MSB Aligned	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
18-bit LSB Aligned															17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
18-bit MSB Aligned	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
20-bit LSB Aligned													19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
20-bit MSB Aligned	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
22-bit LSB Aligned											21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
22-bit MSB Aligned	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
24-bit LSB Aligned									23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24-bit MSB Aligned	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								

In addition, receive data can either be zero-extended or sign-extended if LSB alignment is selected. With zero-extension, all bits above the most significant bit are 0's. This format is useful when data is stored in a pure integer format. With sign-extension, all bits above the most significant bit are equal to the most significant bit. This format is useful when data is stored in a fixed-point integer format (which implies fractional values). Receive data extension is controlled by the RXEXT bit in the SSI.SRCR. Transmit data used with LSB alignment has no concept of sign/zero-extension. Unused bits above the most significant bit are simply ignored.

When configured in I2S or AC97 mode, the SSI forces the selection of LSB alignment. However, RXEXT still permits a choice between zero-extension and sign-extension.

See the SSI Memory Map/Register Definition section for more details on the relevant bits in the SSI.STCR and SSI.SRCR registers.

61.8.3 SSI Architecture

The Synchronous Serial Interface (SSI) is connected to chip pads through the Digital Audio Mux (AUDMUX) block. The AUDMUX can be configured to connect the SSI to the chip pads in various ways.

Refer to Figure 61-1 for a block diagram of the SSI.

61.8.4 SSI Clocking

The SSI uses the following clocks:

- Bit clock - Used to serially clock the data bits in and out of the SSI port. This clock is either generated internally (from SSI's sys clock) or taken from external clock source (through the Tx/Rx clock ports).
- Word clock - Used to count the number of data bits per word (8, 10, 12, 16, 18, 20, 22 or 24 bits). This clock is generated internally from the bit clock.
- Frame clock (Frame Sync) - Used to count the number of words in a frame. This signal can be generated internally from the bit clock, or taken from external source (from the Tx/Rx frame sync ports).
- Network clock -In master mode, this is an integer multiple of frame clock. This is oversampling clock. It is used in cases when SSI has to provide the clock.

Care should be taken to ensure that the bit clock frequency (either internally generated by dividing the SSI's sys clock or sourced from external device through Tx/Rx clock ports) is never greater than 1/5 of the ipg_clk (from CCM) frequency.

In Normal mode (SCR[6:5]=00), the bit clock, used to serially clock the data, is visible on the Serial Transmit Clock (STCK) and Serial Receive Clock (SRCK) ports. The word clock is an internal clock used to determine when transmission of an 8, 10, 12, 16, 18, 20, 22 or 24 bit word has completed. The word clock in turn then clocks the frame clock, which counts the number of words in the frame. The frame clock can be viewed on the STFS and SRFS frame sync ports, because a frame sync is generated after the correct number of words in the frame have passed. In master and synchronous mode, the unused port SRCK is used as network clock (oversampling clock) enabled by the SCR register bit 15, SYS_CLK_EN. This network clock is an oversampling clock of the frame sync clock (STFS). In this mode, the word length (WL), Prescaler Range (PSR), Prescaler Modulus (PM) and Frame rate (DC) selects the ratio of network clock to sampling clock STFS. In case of I2S mode, the network clock (oversampling clock) can be made available on this port if the SYS_CLK_EN bit is set. The relationship between the clocks and the dividers is shown in the figure below ("SSI Clocking"). The bit clock can be received from an SSI clock port or can be generated from the network clock through a divider, as shown in Figure 61-22 ("SSI Transmit Clock Generator Block Diagram").

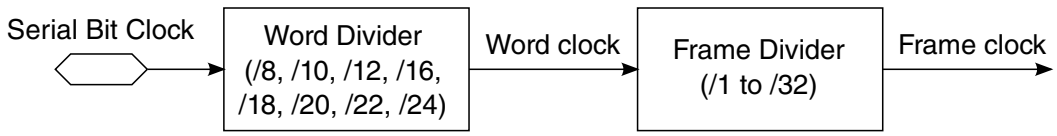


Figure 61-21. SSI Clocking

61.8.4.1 SSI Clock and Frame Sync Generation

Data clock and frame sync signals can be generated internally, or can be obtained from external sources. If internally generated, the SSI clock generator is used to derive bit clock and frame sync signals from the SSI's sys clock. The SSI clock generator consists of a selectable, fixed prescaler and a programmable prescaler for bit rate clock generation.

In Gated Clock mode, the data clock is valid only when data is being transmitted. Otherwise the clock port is pulled to the inactive state. A programmable frame rate divider and a word length divider are used for frame rate sync signal generation.

The following figure shows a block diagram of the clock generator for the transmit section. The serial bit clock can be internal or external, depending on the Transmit Direction (TXDIR) bit in the SSI Transmit Configuration Register (SSI.STCR). The receive section contains an equivalent clock generator circuit.

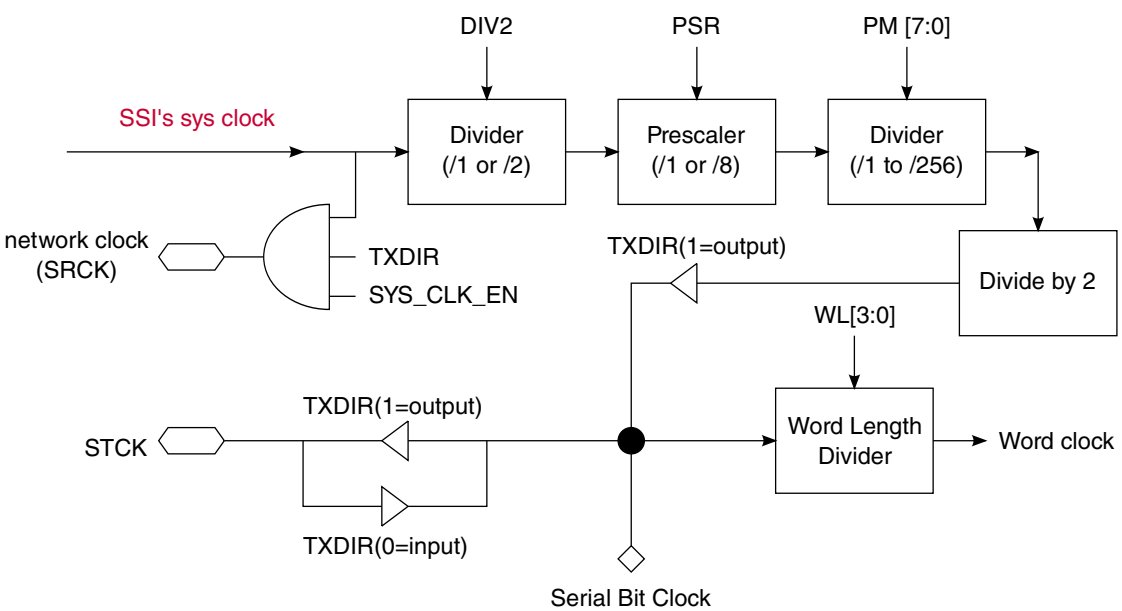


Figure 61-22. SSI Transmit Clock Generator Block Diagram

The figure below shows the Frame Sync Generator block for the transmit section. When internally generated, both receive and transmit frame sync are generated from the word clock and are defined by the Frame Rate Divider (DC[4:0]) bits and the Word Length (WL[3:0]) bits of the SSI Transmit Clock Control Register (SSI.STCCR). The receive section contains an equivalent circuit for the Frame Sync Generator.

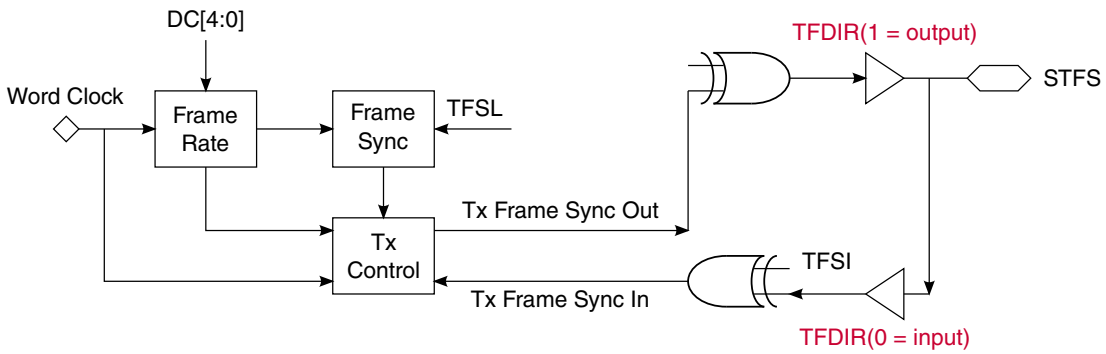


Figure 61-23. SSI Transmit Frame Sync Generator Block Diagram

61.8.4.2 DIV2, PSR and PM Bit Description

The bit clock frequency can be calculated from the SSI's sys clock using the equation in the following figure.

NOTE

You must ensure that the bit-clock frequency must be never greater than 1/5 of the peripheral clock frequency. The oversampling clock frequency can go up to peripheral clock frequency. Bits DIV2, PSR and PM should not be all set to zero at the same time.

$$f_{INT_BIT_CLK} = f_{SSI's\ sys\ clock} / [(DIV2 + 1) \times (7 \times PSR + 1) \times (PM + 1) \times 2]$$

where PM = PM[7:0]

$$f_{FRAME_SYN_CLK} = (F_{INT_BIT_CLK}) / [(DC + 1) \times WL]$$

where DC = DC[4:0] and WL = 8, 10, 12, 16, 18, 20, 22, 24

Figure 61-24. SSI Bit Clock Equation

For example, if the SSI's sys clock is 12.288 MHz, in 8-bit word Normal mode with DC[4:0] set to 0(00000), PM[7:0] set to 47 (0010 1111), the PSR bit cleared, DIV2 bit set to 1, a bit clock rate of 12.288 Mhz / [1 x 4 x 48] = 64 kHz is generated. Since the 8-bit word rate is equal to one (i.e. normal mode), the sampling rate (FS rate) would then be 64 kHz / [1 * 8] = 8 kHz.

In the next example, SSI's sys clock is 11.2896 Mhz. A 16-bit word Network mode with DC[4:0] set to 1 (00001), PM[7:0] set to 3 (0000 0011), the PSR bit is set to 0, DIV2 bit set to 0, and a 11.2896 MHz oversampling clock, a bit clock rate of 11.2896 Mhz / [1 x 2 x 4] = 1.4112 MHz is generated. Since the 16-bit word rate is equal to two, the sampling rate (FS rate) would be 1.4112 MHz / [2 * 16] = 44.1 kHz.

The table below shows programming examples for the clock dividers in the CCM and the SSI to support various bit clock (STCK) frequencies.

Table 61-7. SSI Bit Clock and Frame Rate as a Function of PSR, PM, and DIV2

Bits/ Word	Words / Frame	Ideal Frame Rate (kHz)	PLL Freq (Mhz)	SSIDIV (in CCM)	SSI's sys clock Freq (Mhz)	DIV2	PSR	PM	WL	DC	Actual Serial bit clock Freq (kHz) STCK	Target Serial bit clock Freq (kHz) STCK	Error (Hz)
16	1	8	688.128	56	12.288	0	0	47	7	0	128	128	0
16	2	8	688.128	56	12.288	0	0	23	7	1	256	256	0
16	4	8	688.128	56	12.288	0	0	11	7	3	512	512	0
16	1	12	688.128	56	12.288	0	0	31	7	0	192	192	0
16	2	12	688.128	56	12.288	0	0	15	7	1	384	384	0
16	4	12	688.128	56	12.288	0	0	7	7	3	768	768	0
16	1	16	688.128	56	12.288	0	0	23	7	0	256	256	0
16	2	16	688.128	56	12.288	0	0	11	7	1	512	512	0
16	4	16	688.128	56	12.288	0	0	5	7	3	1024	1024	0
16	1	24	688.128	56	12.288	0	0	15	7	0	384	384	0
16	2	24	688.128	56	12.288	0	0	7	7	1	768	768	0
16	4	24	688.128	56	12.288	0	0	3	7	3	1536	1536	0
16	1	32	688.128	56	12.288	0	0	11	7	0	512	512	0
16	2	32	688.128	56	12.288	0	0	5	7	1	1024	1024	0
16	4	32	688.128	56	12.288	0	0	2	7	3	2048	2048	0
16	1	48	688.128	56	12.288	0	0	15	7	0	768	768	0
16	2	48	688.128	56	12.288	0	0	3	7	1	1536	1536	0
16	4	48	688.128	56	12.288	0	0	1	7	3	3072	3072	0
16	1	11.025	632.2176	56	11.2896	0	0	31	7	0	176.4	176.4	0

Table continues on the next page...

Table 61-7. SSI Bit Clock and Frame Rate as a Function of PSR, PM, and DIV2 (continued)

Bits/ Word	Words / Frame	Ideal Frame Rate (kHz)	PLL Freq (Mhz)	SSIDIV (in CCM)	SSI's sys clock Freq (Mhz)	DIV2	PSR	PM	WL	DC	Actual Serial bit clock Freq (kHz) STCK	Target Serial bit clock Freq (kHz) STCK	Error (Hz)
16	2	11.025	632.2176	56	11.2896	0	0	15	7	1	352.8	352.8	0
16	4	11.025	632.2176	56	11.2896	0	0	7	7	3	705.6	705.6	0
16	1	22.05	632.2176	56	11.2896	0	0	15	7	0	352.8	352.8	0
16	2	22.05	632.2176	56	11.2896	0	0	7	7	1	705.6	705.6	0
16	4	22.05	632.2176	56	11.2896	0	0	3	7	3	1411.2	1411.2	0
16	1	44.1	632.2176	56	11.2896	0	0	7	7	0	705.6	705.6	0
16	2	44.1	632.2176	56	11.2896	0	0	3	7	1	1411.2	1411.2	0
16	4	44.1	632.2176	56	11.2896	0	0	1	7	3	2822.4	2822.4	0

NOTE

The table above describes how various frame rates can be achieved with the PLLs supplying a frequency of 688.128 MHz and 633.2176 MHz (with WL and DC settings as shown). These clocks are recommended as convenient starting points but the system allows for other input clock frequencies as well.

Table 61-7 shows programming of the CCM and SSI dividers in order to generate the appropriate network clock and serial bit clock frequencies for various sampling rates. In these examples, the master mode is selected either by setting I2S master bit (SCR[6:5]=01) or individually programming the SSI in network, synchronous, transmit internal mode (the table specifically illustrates the I2S mode frequencies/sample rates). The network clock is oversampling clock.

Note that the I2S master mode requires that a word length of 32 bits be used (regardless of the actual data type). Consequently, the fixed I2S frame rate of 64 bits per frame (word length (WL) can be any value) and DC of 1 are assumed.

61.8.5 Receive Interrupt Enable Bit Description

When the RIE and RE bit are set, the processor is interrupted when either of the SSI Receive FIFO Full (RFF0/1) bits in SSI.SISR is set (if the corresponding Receive FIFO is enabled).

If the Receive FIFO is not enabled, the interrupt is generated when the corresponding SSI Receive Data Ready (RDR0/1) bit in the SSI.SISR is set. When the receive FIFO is enabled, a maximum of 15 values are available to be read (15 values per channel in Two-Channel mode). If not enabled, then one value can be read from the SRX register (one each in case of Two-Channel mode). If the RIE bit is cleared, these interrupts are disabled. However, the RFF0/1 and RDR0/1 bits still indicate the receive data register full condition. Reading the SSI.SRX registers clears the RDR bits, thus clearing the pending interrupt. Two receive data interrupts (two per channel in case of Two-Channel mode) are available: receive data with exception status and receive data without exception. The tables below show the conditions under which these interrupts are generated.

Table 61-8. SSI Receive Data 1 Interrupts

Interrupt	RIE	ROE0	RFF0/RDR0
Receive Data 1(with Exception Status)	1	1	1
Receive Data 1(without exception)	1	0	1

Table 61-9. SSI Receive Data 0 Interrupts

Interrupt	RIE	ROE1	RFF1/RDR1
Receive Data 0 (with Exception Status)	1	1	1
Receive Data 0 (without exception)	1	0	1

61.8.6 Transmit Interrupt Enable Bit Description

The SSI Transmit Interrupt Enable (TIE) control bit determines whether the processor is interrupted when the SSI transmitter needs to be serviced.

When the TIE and TE bits are set, the program controller is interrupted when either of the SSI Transmit FIFO Empty (TFE0/1) flags in SISR are set (if corresponding Transmit FIFO is enabled). If the corresponding Transmit FIFO is not enabled, an interrupt is generated when the corresponding SSI Transmit Data Register Empty (TDE0/1) flag in the SISR is set and Transmit Enable (TE) bit is set.

When Transmit FIFO 0 is enabled, a maximum of 15 values can be written to the SSI (15 per channel in case of Two-Channel mode, using Tx FIFO 1). If not enabled, then one value can be written to the SSI.STX0 register (one per channel in case of Two-Channel mode using SSI.STX1). When the TIE bit is cleared, all transmit interrupts are disabled. However, the TDE0/1 bits always indicate the corresponding SSI.STX register empty condition, even when the transmitter is disabled by the Transmit Enable (TE) bit (in the SSI.SCR). Writing data to the STX clears the corresponding TDE bit, thus clearing the interrupt. Two transmit data interrupts are available (four in case of Two-Channel mode, two per channel): transmit data with exception status and transmit data without exceptions. The tables below show the conditions under which these interrupts are generated.

Table 61-10. SSI Transmit Data 1 Interrupts

Interrupt	TIE	TUE1	TFE1/TDE1
Transmit Data 1 (with Exception Status)	1	1	1
Transmit Data 1 (without exception)	1	0	1

Table 61-11. SSI Transmit Data 0 Interrupts

Interrupt	TIE	TUE0	TFE0/TDE0
Transmit Data 0 (with Exception Status)	1	1	1
Transmit Data 0 (without exception)	1	0	1

61.8.7 Internal Frame and Clock Shutdown

During transmit/receive operation, disabling TE/RE will ensure that data transmission/ reception stops after current frame ends following which TFRC/RFRC Status bits will get set to indicate the Frame Completion State.

If TE is disabled 4 clock cycles before the next frame, extra frame generated are invalid frames. TFR_CLK_DIS/RFR_CLK_DIS bit is set in the current or any of the previous frames, SSI will stop driving the STFS/SRFS and STCK/SRCK signals after the current frame ends.

If TFR_CLK_DIS/RFR_CLK_DIS bit is not set, SSI will continue generating STFS/ SRFS and STCK/SRCK signals (in case direction is from SSI), which then can be disabled by writing '1' to TFR_CLK_DIS/RFR_CLK_DIS bit. SSI will then stop driving these signals after end of frame is reached following which TFRC/RFRC status bits will get set to indicate the Frame Completion State.

The following figure is an illustration of transmission case where TXDIR and TFDIR are both set to '1'. In this case TE is disabled with TFR_CLK_DIS bit set in current or any of the previous frames.

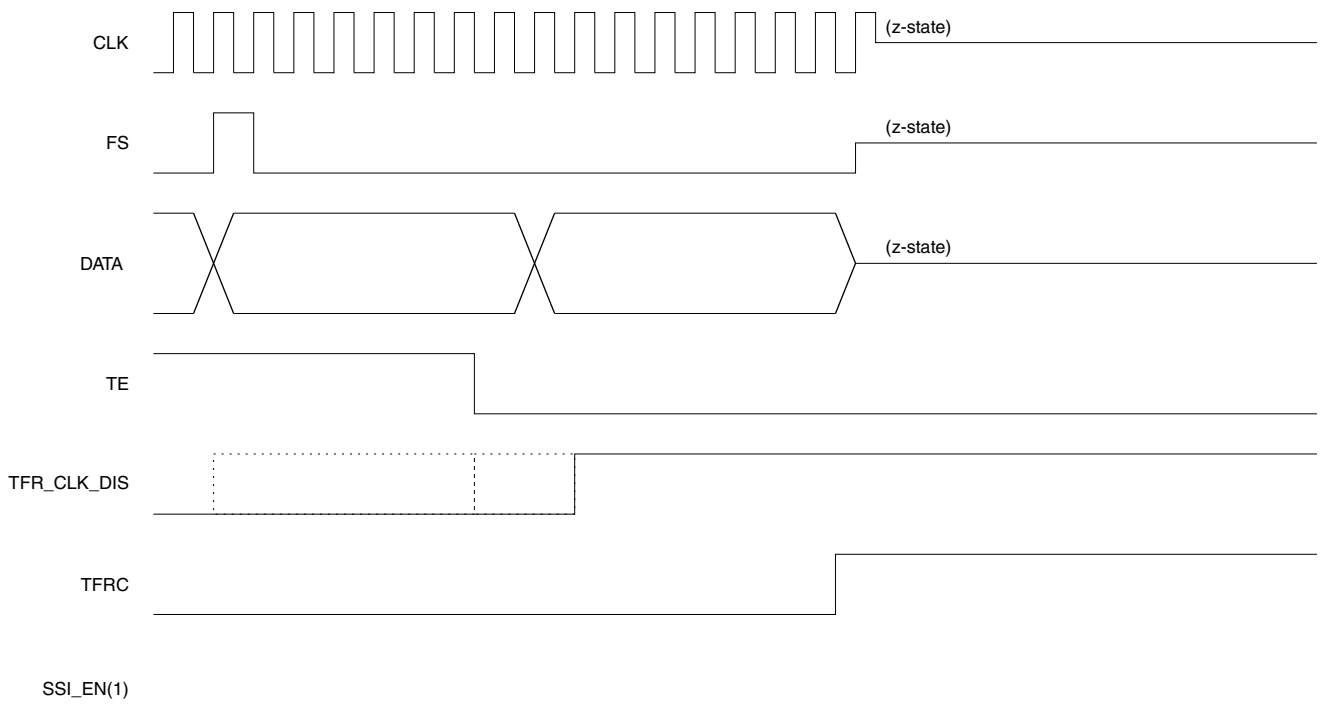


Figure 61-25. TFR_CLK_DIS assertion in current or previous frame as TE disable

The figure below is an illustration of transmission case where TXDIR and TFDIR are both set to '1'. In this case TFR_CLK_DIS bit is set after few frames of disabling TE. TFRC (Transmit Frame Complete) is set at frame boundary after TE is cleared. Once software services this interrupt and sets TFR_CLK_DIS bit later, TFRC bit is again set at next frame boundary.

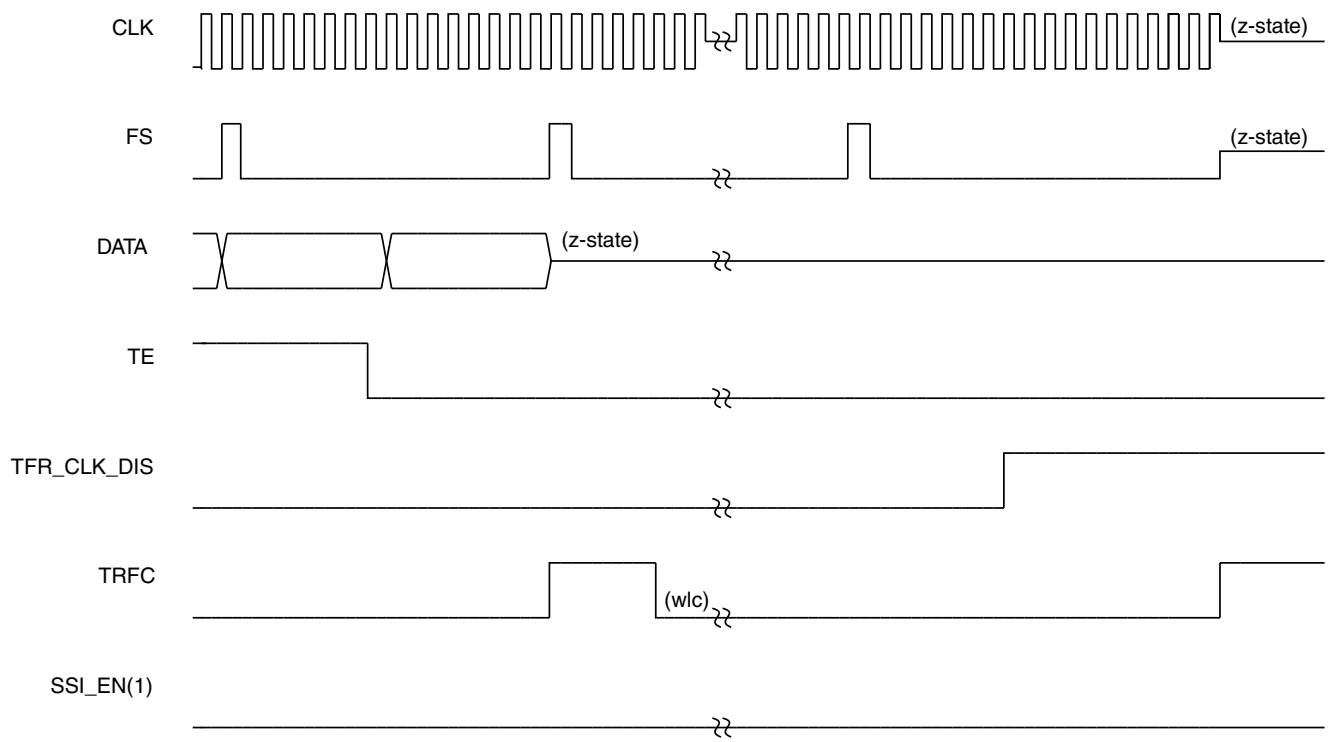


Figure 61-26. TFR_CLK_DIS assertion in subsequent frame after disabling TE

61.8.8 Peripheral Bus Interface

The SSI has a Peripheral Bus interface to provide a control and data interface. This interface is used by both the processor and DMA controller.

61.8.8.1 Transfer Lengths Supported

The Peripheral Bus interface of the SSI only supports 32-bit transfers with all SSI registers other than SSI.STX0, SSI.STX1, SSI.SRX0, and SSI.SRX1 (that is, the data registers).

With the exception of the data registers, using 8-bit and 16-bit transactions could result in undesired behavior but will not result in a transfer bus error. The data registers (SSI.STX0, SSI.STX1, SSI.SRX0, and SSI.SRX1) support 8-bit, 16-bit, and 32-bit transfer lengths without restrictions.

61.8.8.2 Transfer Bus Errors

Transfer bus errors are generated upon response to the following:

- Write transfer to a read-only register.
- Read or write access to a register space beyond the last populated register of the SSI in its memory map (up until the end of the allocated memory address range of the SSI).

61.8.8.3 Clock Rate

The Peripheral Bus clock frequency must be at least five times the serial bit clock frequency.

61.8.9 Reset

The SSI is affected by the following types of reset:

- Power-on Reset-The Power-on reset clears the SSIEN bit in SSI.SCR, which disables the SSI. All other status and control bits in the SSI are affected as described in SSI Programming Model in the "Memory Map and Register Definition section".
- SSI Reset-The SSI reset is generated when the SSIEN bit in the SSI.SCR is cleared. The SSI status bits are preset to the same state produced by the Power-on reset. The SSI control bits are unaffected. The control bits in the SSI.SCR are also unaffected. The SSI reset is useful for selective reset of the SSI without changing the present SSI control bits and without affecting the other peripherals.

61.9 SSI Memory Map/Register Definition

SSI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
202_8000	SSI Transmit Data Register n (SSI1_STX0)	32	R/W	0000_0000h	61.9.1/5096
202_8004	SSI Transmit Data Register n (SSI1_STX1)	32	R/W	0000_0000h	61.9.1/5096
202_8008	SSI Receive Data Register n (SSI1_SRX0)	32	R	0000_0000h	61.9.2/5096
202_800C	SSI Receive Data Register n (SSI1_SRX1)	32	R	0000_0000h	61.9.2/5096
202_8010	SSI Control Register (SSI1_SCR)	32	R/W	0000_0000h	61.9.3/5097
202_8014	SSI Interrupt Status Register (SSI1_SISR)	32	w1c	0000_3003h	61.9.4/5099
202_8018	SSI Interrupt Enable Register (SSI1_SIER)	32	R/W	0000_3003h	61.9.5/5105
202_801C	SSI Transmit Configuration Register (SSI1_STCR)	32	R/W	0000_0200h	61.9.6/5109
202_8020	SSI Receive Configuration Register (SSI1_SRCR)	32	R/W	0000_0200h	61.9.7/5111

Table continues on the next page...

SSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
202_8024	SSI Transmit Clock Control Register (SSI1_STCCR)	32	R/W	0004_0000h	61.9.8/5113
202_8028	SSI Receive Clock Control Register (SSI1_SRCCR)	32	R/W	0004_0000h	61.9.9/5115
202_802C	SSI FIFO Control/Status Register (SSI1_SFCSR)	32	R/W	0081_0081h	61.9.10/ 5116
202_8038	SSI AC97 Control Register (SSI1_SACNT)	32	R/W	0000_0000h	61.9.11/ 5120
202_803C	SSI AC97 Command Address Register (SSI1_SACADD)	32	R/W	0000_0000h	61.9.12/ 5121
202_8040	SSI AC97 Command Data Register (SSI1_SACDAT)	32	R/W	0000_0000h	61.9.13/ 5121
202_8044	SSI AC97 Tag Register (SSI1_SATAG)	32	R/W	0000_0000h	61.9.14/ 5122
202_8048	SSI Transmit Time Slot Mask Register (SSI1_STMSK)	32	R/W	0000_0000h	61.9.15/ 5122
202_804C	SSI Receive Time Slot Mask Register (SSI1_SRMSK)	32	R/W	0000_0000h	61.9.16/ 5123
202_8050	SSI AC97 Channel Status Register (SSI1_SACCST)	32	R	0000_0000h	61.9.17/ 5123
202_8054	SSI AC97 Channel Enable Register (SSI1_SACCEN)	32	W	0000_0000h	61.9.18/ 5124
202_8058	SSI AC97 Channel Disable Register (SSI1_SACCDIS)	32	W	0000_0000h	61.9.19/ 5124
202_C000	SSI Transmit Data Register n (SSI2_STX0)	32	R/W	0000_0000h	61.9.1/5096
202_C004	SSI Transmit Data Register n (SSI2_STX1)	32	R/W	0000_0000h	61.9.1/5096
202_C008	SSI Receive Data Register n (SSI2_SRX0)	32	R	0000_0000h	61.9.2/5096
202_C00C	SSI Receive Data Register n (SSI2_SRX1)	32	R	0000_0000h	61.9.2/5096
202_C010	SSI Control Register (SSI2_SCR)	32	R/W	0000_0000h	61.9.3/5097
202_C014	SSI Interrupt Status Register (SSI2_SISR)	32	w1c	0000_3003h	61.9.4/5099
202_C018	SSI Interrupt Enable Register (SSI2_SIER)	32	R/W	0000_3003h	61.9.5/5105
202_C01C	SSI Transmit Configuration Register (SSI2_STCR)	32	R/W	0000_0200h	61.9.6/5109
202_C020	SSI Receive Configuration Register (SSI2_SRCR)	32	R/W	0000_0200h	61.9.7/5111
202_C024	SSI Transmit Clock Control Register (SSI2_STCCR)	32	R/W	0004_0000h	61.9.8/5113
202_C028	SSI Receive Clock Control Register (SSI2_SRCCR)	32	R/W	0004_0000h	61.9.9/5115
202_C02C	SSI FIFO Control/Status Register (SSI2_SFCSR)	32	R/W	0081_0081h	61.9.10/ 5116
202_C038	SSI AC97 Control Register (SSI2_SACNT)	32	R/W	0000_0000h	61.9.11/ 5120
202_C03C	SSI AC97 Command Address Register (SSI2_SACADD)	32	R/W	0000_0000h	61.9.12/ 5121
202_C040	SSI AC97 Command Data Register (SSI2_SACDAT)	32	R/W	0000_0000h	61.9.13/ 5121

Table continues on the next page...

SSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
202_C044	SSI AC97 Tag Register (SSI2_SATAG)	32	R/W	0000_0000h	61.9.14/ 5122
202_C048	SSI Transmit Time Slot Mask Register (SSI2_STMSK)	32	R/W	0000_0000h	61.9.15/ 5122
202_C04C	SSI Receive Time Slot Mask Register (SSI2_SRMSK)	32	R/W	0000_0000h	61.9.16/ 5123
202_C050	SSI AC97 Channel Status Register (SSI2_SACCST)	32	R	0000_0000h	61.9.17/ 5123
202_C054	SSI AC97 Channel Enable Register (SSI2_SACCEN)	32	W	0000_0000h	61.9.18/ 5124
202_C058	SSI AC97 Channel Disable Register (SSI2_SACCDIS)	32	W	0000_0000h	61.9.19/ 5124
203_0000	SSI Transmit Data Register n (SSI3_STX0)	32	R/W	0000_0000h	61.9.1/5096
203_0004	SSI Transmit Data Register n (SSI3_STX1)	32	R/W	0000_0000h	61.9.1/5096
203_0008	SSI Receive Data Register n (SSI3_SRX0)	32	R	0000_0000h	61.9.2/5096
203_000C	SSI Receive Data Register n (SSI3_SRX1)	32	R	0000_0000h	61.9.2/5096
203_0010	SSI Control Register (SSI3_SCR)	32	R/W	0000_0000h	61.9.3/5097
203_0014	SSI Interrupt Status Register (SSI3_SISR)	32	w1c	0000_3003h	61.9.4/5099
203_0018	SSI Interrupt Enable Register (SSI3_SIER)	32	R/W	0000_3003h	61.9.5/5105
203_001C	SSI Transmit Configuration Register (SSI3_STCR)	32	R/W	0000_0200h	61.9.6/5109
203_0020	SSI Receive Configuration Register (SSI3_SRCR)	32	R/W	0000_0200h	61.9.7/5111
203_0024	SSI Transmit Clock Control Register (SSI3_STCCR)	32	R/W	0004_0000h	61.9.8/5113
203_0028	SSI Receive Clock Control Register (SSI3_SRCCR)	32	R/W	0004_0000h	61.9.9/5115
203_002C	SSI FIFO Control/Status Register (SSI3_SFCSR)	32	R/W	0081_0081h	61.9.10/ 5116
203_0038	SSI AC97 Control Register (SSI3_SACNT)	32	R/W	0000_0000h	61.9.11/ 5120
203_003C	SSI AC97 Command Address Register (SSI3_SACADD)	32	R/W	0000_0000h	61.9.12/ 5121
203_0040	SSI AC97 Command Data Register (SSI3_SACDAT)	32	R/W	0000_0000h	61.9.13/ 5121
203_0044	SSI AC97 Tag Register (SSI3_SATAG)	32	R/W	0000_0000h	61.9.14/ 5122
203_0048	SSI Transmit Time Slot Mask Register (SSI3_STMSK)	32	R/W	0000_0000h	61.9.15/ 5122
203_004C	SSI Receive Time Slot Mask Register (SSI3_SRMSK)	32	R/W	0000_0000h	61.9.16/ 5123
203_0050	SSI AC97 Channel Status Register (SSI3_SACCST)	32	R	0000_0000h	61.9.17/ 5123
203_0054	SSI AC97 Channel Enable Register (SSI3_SACCEN)	32	W	0000_0000h	61.9.18/ 5124
203_0058	SSI AC97 Channel Disable Register (SSI3_SACCDIS)	32	W	0000_0000h	61.9.19/ 5124

61.9.1 SSI Transmit Data Register n (SSIx_STX*n*)

NOTE

Enable SSI (SSIEN=1) before writing to SSI Transmit Data Registers.

Address: Base address + 0h offset + (4d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STXn																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSIx_STX*n* field descriptions

Field	Description
STXn	SSI Transmit Data. These bits store the data to be transmitted by the SSI. These are implemented as the first word of their respective Tx FIFOs. Data written to these registers is transferred to the Transmit Shift Register (TXSR), when shifting of the previous data is complete. If both FIFOs are in use, data is alternately transferred from STX0 and STX1, to TXSR. Multiple writes to the STX registers will not result in the previous data being over-written by the subsequent data. STX1 can only be used in Two-Channel mode of operation. Protection from over-writing is present irrespective of whether the transmitter is enabled or not. Example 1: If Tx FIFO0 is in use and user writes Data1...Data16 to STX0,Data16 will not over-write Data1. Data1...Data15 are stored in the FIFO whileData16 is discarded. Example 2: If Tx FIFO0 is not in use and user writes Data1, Data2 to STX0, then Data2 will not over-write Data1 and will be discarded.

61.9.2 SSI Receive Data Register n (SSIx_SRX*n*)

Address: Base address + 8h offset + (4d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SRXn																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSIx_SRX*n* field descriptions

Field	Description
SRXn	SSI Receive Data. These bits store the data received by the SSI. These are implemented as the first word of their respective Rx FIFOs. These bits receive data from the RXSR depending on the mode of operation. In case both FIFOs are in use, data is transferred to each data register alternately. SRX1 can only be used in Two-Channel mode of operation.

61.9.3 SSI Control Register (SSIx_SCR)

The SSI Control Register (SSI_SCR) sets up the SSI reset is controlled by bit 0 in the SSI_SCR. SSI operating modes are also selected in this register (except AC97 mode which is selected in the SSI_SACNT register).

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SYNC_TX_FS	RFR_CLK_DIS	TFR_CLK_DIS	CLK_IST	TCH_EN	SYS_CLK_EN	I2S_MODE	SYN	NET	RE	TE	SSIEN	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSIx_SCR field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value 0.
12 SYNC_TX_FS	SYNC_FS_TX bit provides a safe window for TE to be visible to the internal circuit which is just after FS occurrence. When SYNC_TX_FS is set, TE(SCR[1]) gets latched on FS occurrence & latched TE is used to enable/disable SSI transmitter. TE needs setup of 2 bit-clock cycles before occurrence of FS. If TE is changed within 2 bit-clock cycles of FS occurrence, there is high probability that TE will be latched on next FS. Note: With TFR_CLK_DIS feature on, TE is used directly to enable transmitter in following cases (i) Sync mode & Rx disabled (ii) Async Mode. Latched-TE is used to disable the transmitter. This bit has no relevance in gated mode and AC97 mode. 0 **TE_NOT_LATCHED** — TE not latched with FS occurrence & used directly for transmitter enable/ disable. 1 **TE_LATCHED** — TE latched with FS occurrence & latched-TE used for transmitter enable/disable.
11 RFR_CLK_DIS	Receive Frame Clock Disable. This bit provides the option to keep the Frame-sync and Clock enabled or to disable them after the receive frame in which the receiver is disabled. Writing to this bit has effect only when RE is disabled.The receiver is disabled by clearing the RE bit.

Table continues on the next page...

SSIx_SCR field descriptions (continued)

Field	Description
	0 **CONTINUE** — Continue Frame-sync/Clock generation after current frame during which RE is cleared. This may be required when Frame-sync and Clocks are required from SSI, even when no data is to be received. 1 **STOP** — Stop Frame-sync/Clock generation at next frame boundary. This will be effective also in case where receiver is already disabled in current or previous frames.
10 TFR_CLK_DIS	Transmit Frame Clock Disable. This bit provide option to keep the Frame-sync and Clock enabled or disabled after current transmit frame, in which transmitter is disabled by clearing TE bit. Writing to this bit has effect only when SSI is enabled TE is disabled. 0 **CONTINUE** — Continue Frame-sync/Clock generation after current frame during which TE is cleared. This may be required when Frame-sync and Clocks are required from SSI, even when no data is to be received. 1 **STOP** — Stop Frame-sync/Clock generation at next frame boundary. This will be effective also in case where transmitter is already disabled in current or previous frames.
9 CLK_IST	Clock Idle State. This bit controls the idle state of the transmit clock port during SSI internal gated mode. Note: When Clock idle state is '1' the clock polarity should always be negedge triggered and when Clock idle = '0' the clock polarity should always be positive edge triggered. 0 **IDLE_0** — Clock idle state is '0'. 1 **IDLE_1** — Clock idle state is '1'.
8 TCH_EN	Two-Channel Operation Enable. This bit allows SSI to operate in the two-channel mode.In this mode while receiving, the RXSR transfers data to SRX0 and SRX1 alternately and while transmitting, data is alternately transferred from STX0 and STX1 to TXSR. For an even number of slots, Two-Channel Operation can be enabled to optimize usage of both FIFOs or disabled as in the case of odd number of active slots. This feature is especially useful in I2S mode, where data for Left Speaker can be placed in Tx-FIFO0 and for Right speaker in Tx-FIFO1. 0 **DISABLED** — Two-channel mode disabled. 1 **ENABLED** — Two-channel mode enabled.
7 SYS_CLK_EN	Network Clock (Oversampling Clock) Enable. When set, this bit allows the SSI to output the network clock at the SRCK port, provided that synchronous mode, and transmit internal clock mode are set. The relationship between bit clock and network clock is determined by DIV2, PSR, and PM bits. This feature is especially useful in I2S Master mode to output network clock (oversampling clock) on SRCK port. 0 **NOT_OUTPUT** — network clock not output on SRCK port. 1 **OUTPUT** — network clock output on SRCK port.
6–5 I2S_MODE	I2S Mode Select. These bits allow the SSI to operate in Normal, I2S Master or I2S Slave mode. Refer to I2S Mode for a detailed description of I2S Mode of operation. Refer to Table 61-5 for details regarding settings.
4 SYN	Synchronous Mode. This bit controls whether SSI is in synchronous mode or not. In synchronous mode, the transmit and receive sections of SSI share a common clock port (STCK) and frame sync port (STFS). 0 **ASYNC_MODE** — Asynchronous mode selected. 1 **SYNC_MODE** — Synchronous mode selected.
3 NET	Network Mode. This bit controls whether SSI is in network mode or not. 0 **DISABLED** — Network mode not selected. 1 **ENABLED** — Network mode selected.

Table continues on the next page...

SSIx_SCR field descriptions (continued)

Field	Description
2 RE	Receive Enable. This control bit enables the receive section of the SSI. When this bit is enabled, data reception starts with the arrival of the next frame sync. If data is being received when this bit is cleared, data reception continues until the end of the current frame and then stops. If this bit is set again before the second to last bit of the last time slot in the current frame, then reception continues without interruption. RE should not be toggled in the same frame. 0 **DISABLED** — Receive section disabled. 1 **ENABLED** — Receive section enabled.
1 TE	Transmit Enable. This control bit enables the transmit section of the SSI. It enables the transfer of the contents of the STX registers to the TXSR and also enables the internal transmit clock. The transmit section is enabled when this bit is set and a frame boundary is detected. When this bit is cleared, the transmitter continues to send data until the end of the current frame and then stops. Data can be written to the STX registers with the TE bit cleared (the corresponding TDE bit will be cleared). If the TE bit is cleared and then set again before the second to last bit of the last time slot in the current frame, data transmission continues without interruption. The normal transmit enable sequence is to write data to the STX register(s) and then set the TE bit. The normal disable sequence is to clear the TE and TIE bits after the TDE bit is set. In gated clock mode, clearing the TE bit results in the clock stopping after the data currently in TXSR has shifted out. When the TE bit is set, the clock starts immediately (for internal gated clock mode). TE should not be toggled in the same frame. After enabling/disabling transmission, SSI expects 4 setup clock cycles before arrival of frame-sync for frame-sync to be accepted/rejected by In case of fewer clock cycles, there is high probability of the frame-sync to get missed. Note: If continuous clock is not provided, SSI expects 6 clock cycles before arrival of frame-sync for frame-sync to be accepted by the SSI. 0 **DISABLED** — Transmit section disabled. 1 **ENABLED** — Transmit section enabled.
0 SSIEN	SSIEN - SSI Enable This bit is used to enable/disable the SSI. When disabled, all SSI status bits are preset to the same state produced by the power-on reset, all control bits are unaffected, the contents of Tx and Rx FIFOs are cleared. When SSI is disabled, all internal clocks are disabled (except register access clock). 0 **DISABLED** — SSI is disabled. 1 **ENABLED** — SSI is enabled.

61.9.4 SSI Interrupt Status Register (SSIx_SISR)

The SSI Interrupt Status Register (SSI_SISR) is used to monitor the SSI. This register is used by the core to interrogate the status of the SSI. In gated mode of operation the TFS, RFS, TLS, RLS, TFRC and RFRC bits of AISR register are not generated. The status bits are described in the following table.

- SSI Status flags are valid when SSI is enabled.

- See Receive Interrupt Enable Bit Description and Transmit Interrupt Enable Bit Description for interrupt source mapping.
- All the flags in the SSI_SISR are updated after the first bit of the next SSI word has completed transmission or reception. Certain status bits (ROE0/1 and TUE0/1) are cleared by writing 1 to the corresponding interrupt status bit in SSI_SISR.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							RFRC	TFRC	0				CMDAU	CMDDU	RXT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDR1	RDR0	TDE1	TDE0	ROE1	ROE0	TUE1	TUE0	TFS	RFS	TLS	RLS	RFF1	RFF0	TFE1	TFE0
W					w1c	w1c	w1c	w1c								
Reset	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1

SSIx_SISR field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24 RFRC	Receive Frame Complete. This flag is set at the end of the frame during which Receiver is disabled. If Receive Frame & Clock are not disabled in the same frame, this flag is also set at the end of the frame in which Receive Frame & Clock are disabled. See the description of RFR_CLK_DIS bit for more details on how to disable Receiver Frame & Clock or keep them enabled after receiver is disabled.

Table continues on the next page...

SSIx_SISR field descriptions (continued)

Field	Description
	0 End of Frame not reached 1 End of frame reached after disabling RE or disabling RFR_CLK_DIS, when receiver is already disabled.
23 TFRC	Transmit Frame Complete. This flag is set at the end of the frame during which Transmitter is disabled. If Transmit Frame & Clock are not disabled in the same frame, this flag is also set at the end of the frame in which Transmit Frame & Clock are disabled. See description of TFR_CLK_DIS bit for more details on how to disable Transmit Frame & Clock or keep them enabled after transmitter is disabled. 0 End of Frame not reached 1 End of frame reached after disabling TE or disabling TFR_CLK_DIS, when transmitter is already disabled.
22–19 Reserved	This read-only field is reserved and always has the value 0.
18 CMDAU	Command Address Register Updated. This bit causes the Command Address Updated interrupt (when CMDAU_EN bit is set). This status bit is set each time there is a difference in the previous and current value of the received Command Address. This bit is cleared on reading the SACADD register. 0 No change in SACADD register. 1 SACADD register updated with different value.
17 CMDDU	Command Data Register Updated. This bit causes the Command Data Updated interrupt (when CMDDU_EN bit is set). This status bit is set each time there is a difference in the previous and current value of the received Command Data. This bit is cleared on reading the SACDAT register. 0 No change in SACDAT register. 1 SACDAT register updated with different value.
16 RXT	Receive Tag Updated. This status bit is set each time there is a difference in the previous and current value of the received tag. It causes the Receive Tag Interrupt (if RXT_EN bit is set). This bit is cleared on reading the SATAG register. 0 No change in SATAG register. 1 SATAG register updated with different value.
15 RDR1	Receive Data Ready 1. This flag bit is set when SRX1 or Rx FIFO 1 is loaded with a new value and Two-Channel mode is selected. RDR1 is cleared when the Core reads the SRX1 register. If Rx FIFO 1 is enabled, RDR1 is cleared when the FIFO is empty. If RIE and RDR1_EN are set, a Receive Data 1 interrupt request is issued on setting of RDR1 bit in case Rx FIFO1 is disabled, if the FIFO is enabled, the interrupt is issued on RFF1 assertion. The RDR1 bit is cleared by POR and SSI reset. 0 No new data for Core to read. 1 New data for Core to read.
14 RDR0	Receive Data Ready 0. This flag bit is set when SRX0 or Rx FIFO 0 is loaded with a new value. RDR0 is cleared when the Core reads the SRX0 register. If Rx FIFO 0 is enabled, RDR0 is cleared when the FIFO is empty. If RIE and RDR0_EN are set, a Receive Data 0 interrupt request is issued on setting of RDR0 bit in case Rx FIFO0 is disabled, if the FIFO is enabled, the interrupt is issued on RFF0 assertion. The RDR0 bit is cleared by POR and SSI reset. 0 No new data for Core to read. 1 New data for Core to read.

Table continues on the next page...

SSIx_SISR field descriptions (continued)

Field	Description
13 TDE1	Transmit Data Register Empty 1. This flag is set whenever data is transferred to TXSR from STX1 register and Two-Channel mode is selected. If Tx FIFO1 is enabled, this occurs when there is at least one empty slot in STX1 or Tx FIFO1. If Tx FIFO1 is not enabled, this occurs when the contents of STX1 are transferred to TXSR. The TDE1 bit is cleared when the Core writes to STX1. If TIE and TDE1_EN are set, an SSI Transmit Data 1 interrupt request is issued on setting of TDE1 bit. The TDE1 bit is cleared by POR and SSI reset. 0 Data available for transmission. 1 Data needs to be written by the Core for transmission.
12 TDE0	Transmit Data Register Empty 0. This flag is set whenever data is transferred to TXSR from STX0 register. If Tx FIFO 0 is enabled, this occurs when there is at least one empty slot in STX0 or Tx FIFO 0. If Tx FIFO 0 is not enabled, this occurs when the contents of STX0 are transferred to TXSR. The TDE0 bit is cleared when the Core writes to STX0. If TIE and TDE0_EN are set, an SSI Transmit Data 0 interrupt request is issued on setting of TDE0 bit. The TDE0 bit is cleared by POR and SSI reset. 0 Data available for transmission. 1 Data needs to be written by the Core for transmission.
11 ROE1	Receiver Overrun Error 1. This flag is set when the RXSR is filled and ready to transfer to SRX1 register or to Rx FIFO 1 (when enabled) and these are already full and Two-Channel mode is selected. If Rx FIFO 1 is enabled, this is indicated by RFF1 flag, else this is indicated by the RDR1 flag. The RXSR is not transferred in this case. The ROE1 flag causes an interrupt if RIE and ROE1_EN are set. The ROE1 bit is cleared by POR and SSI reset. It is also cleared by writing '1' to this bit. Clearing the RE bit does not affect the ROE1 bit. 0 Default interrupt issued to the Core. 1 Exception interrupt issued to the Core.
10 ROE0	Receiver Overrun Error 0. This flag is set when the RXSR is filled and ready to transfer to SRX0 register or to Rx FIFO 0 (when enabled) and these are already full. If Rx FIFO 0 is enabled, this is indicated by RFF0 flag, else this is indicated by the RDR0 flag. The RXSR is not transferred in this case. The ROE0 flag causes an interrupt if RIE and ROE0_EN are set. The ROE0 bit is cleared by POR and SSI reset. It is also cleared by writing '1' to this bit. Clearing the RE bit does not affect the ROE0 bit. 0 Default interrupt issued to the Core. 1 Exception interrupt issued to the Core.
9 TUE1	Transmitter Underrun Error 1. This flag is set when the TXSR is empty (no data to be transmitted), the TDE1 flag is set, a transmit time slot occurs and the SSI is in Two-Channel mode. When a transmit underrun error occurs, the previous data is retransmitted. In Network mode, each time slot requires data transmission (unless masked through STMSK register), when the transmitter is enabled (TE is set). The TUE1 flag causes an interrupt if TIE and TUE1_EN are set. The TUE1 bit is cleared by POR and SSI reset. It is also cleared by writing '1' to this bit. 0 Default interrupt issued to the Core. 1 Exception interrupt issued to the Core.
8 TUE0	Transmitter Underrun Error 0. This flag is set when the TXSR is empty (no data to be transmitted), the TDE0 flag is set and a transmit time slot occurs. When a transmit underrun error occurs, the previous data

Table continues on the next page...

SSIx_SISR field descriptions (continued)

Field	Description
	is retransmitted. In Network mode, each time slot requires data transmission (unless masked through STMSK register), when the transmitter is enabled (TE is set). The TUE0 flag causes an interrupt if TIE and TUE0_EN are set. The TUE0 bit is cleared by POR and SSI reset. It is also cleared by writing '1' to this bit. 0 Default interrupt issued to the Core. 1 Exception interrupt issued to the Core.
7 TFS	Transmit Frame Sync. This flag indicates the occurrence of transmit frame sync. Data written to the STX registers during the time slot when the TFS flag is set, is sent during the second time slot (in Network mode) or in the next first time slot (in Normal mode). In Network mode, the TFS bit is set during transmission of the first time slot of the frame and is then cleared when starting transmission of the next time slot. In Normal mode, this bit is high for the first time slot. This flag causes an interrupt if TIE and TFS_EN are set. The TFS bit is cleared by POR and SSI reset. 0 No Occurrence of Transmit frame sync. 1 Transmit frame sync occurred during transmission of last word written to STX registers.
6 RFS	Receive Frame Sync. This flag indicates the occurrence of receive frame sync. In Network mode, the RFS bit is set when the first slot of the frame is being received. It is cleared when the next slot begins to be received. In Normal mode, this bit is always high (When DC = 0). This flag causes an interrupt if RIE and RFS_EN are set. The RFS bit is cleared by POR and SSI reset. 0 No Occurrence of Receive frame sync. 1 Receive frame sync occurred during reception of next word in SRX registers.
5 TLS	Transmit Last Time Slot. This flag indicates the last time slot in a frame. When set, it indicates that the current time slot is the last time slot of the frame. TLS is set at the start of the last transmit time slot and causes the SSI to issue an interrupt (if TIE and TLS_EN are set). TLS is not generated when frame rate is 1 in normal mode of operation. TLS is cleared when the SISR is read with this bit set. The TLS bit is cleared by POR and SSI reset. 0 Current time slot is not last time slot of frame. 1 Current time slot is the last transmit time slot of frame.
4 RLS	Receive Last Time Slot. This flag indicates the last time slot in a frame. When set, it indicates that the current time slot is the last receive time slot of the frame. RLS is set at the end of the last time slot and causes the SSI to issue an interrupt (if RIE and RLS_EN are set). RLS is cleared when the SISR is read with this bit set. The RLS bit is cleared by POR and SSI reset. 0 Current time slot is not last time slot of frame. 1 Current time slot is the last receive time slot of frame.
3 RFF1	Receive FIFO Full 1. This flag is set when Rx FIFO1 is enabled, the data level in Rx FIFO1 reaches the selected Rx FIFO WaterMark 1 (RFWM1) threshold and the SSI is in Two-Channel mode. The setting of RFF1 only causes an interrupt when RIE and RFF1_EN are set, Rx FIFO1 is enabled and the Two-Channel mode is selected. RFF1 is automatically cleared when the amount of data in Rx FIFO1 falls below the threshold. The RFF1 bit is cleared by POR and SSI reset. When Rx FIFO1 contains15 words, the maximum it can hold, all further data received (for storage in this FIFO) is ignored until the FIFO contents are read. 0 **NOT_FULL** — Space available in Receive FIFO1. 1 **FULL** — Receive FIFO1 is full.
2 RFF0	Receive FIFO Full 0. This flag is set when Rx FIFO0 is enabled and the data level in Rx FIFO0 reaches the selected Rx FIFO WaterMark 0 (RFWM0) threshold. The setting of RFF0 only causes an interrupt

Table continues on the next page...

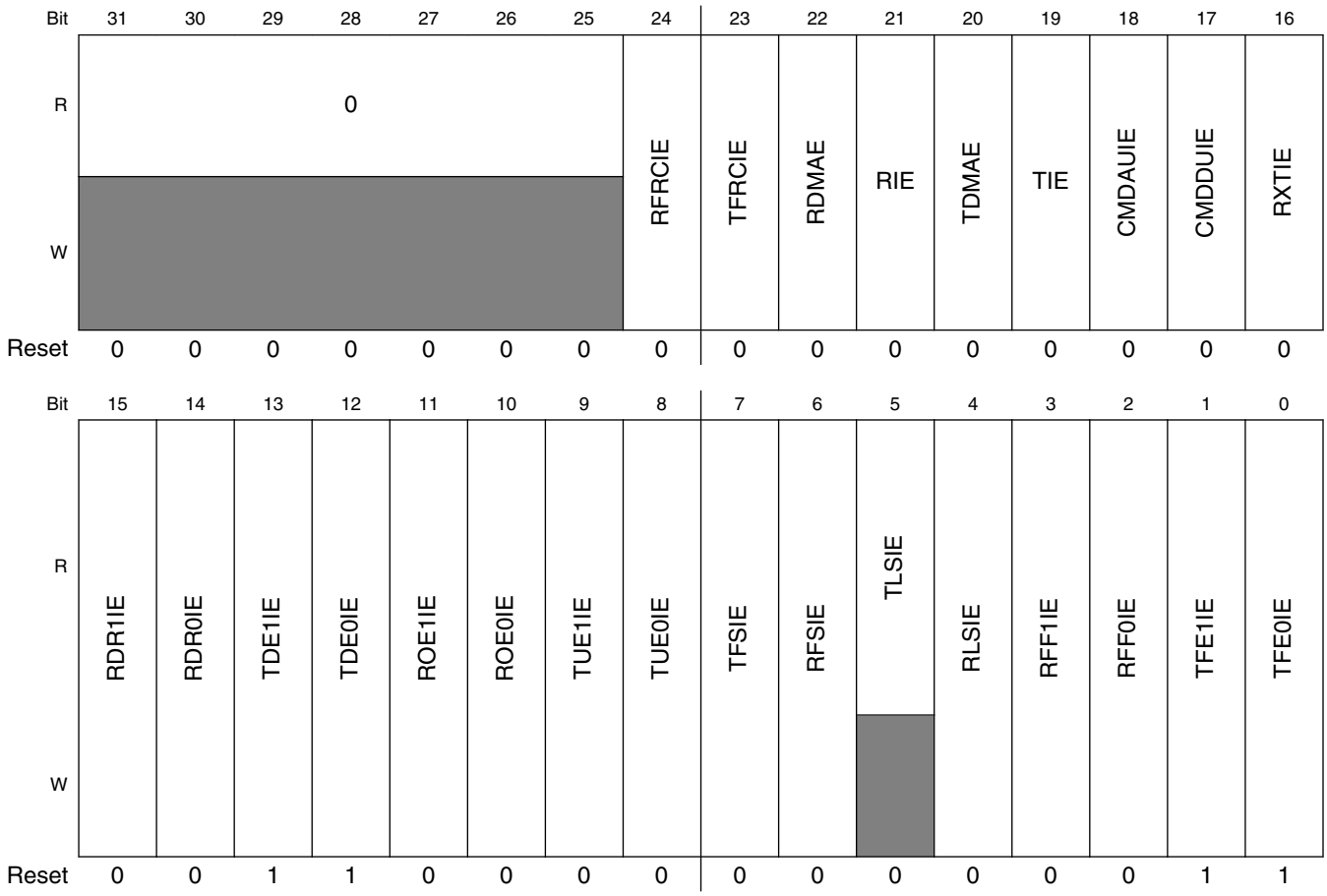
SSIx_SISR field descriptions (continued)

Field	Description
	when RIE and RFF0_EN are set and Rx FIFO0 is enabled. RFF0 is automatically cleared when the amount of data in Rx FIFO0 falls below the threshold. The RFF0 bit is cleared by POR and SSI reset. When Rx FIFO0 contains15 words, the maximum it can hold, all further data received (for storage in this FIFO) is ignored until the FIFO contents are read. 0 **NOT_FULL** — Space available in Receive FIFO0. 1 **FULL** — Receive FIFO0 is full.
1 TFE1	Transmit FIFO Empty 1. This flag is set when the empty slots in Tx FIFO exceed or are equal to the selected Tx FIFO WaterMark 1 (TFWM1) threshold and the Two-Channel mode is selected. The setting of TFE1 only causes an interrupt when TIE and TFE1_EN are set, Tx FIFO1 is enabled and Two-Channel mode is selected. The TFE1 bit is automatically cleared when the data level in Tx FIFO1 becomes more than the amount specified by the watermark bits. The TFE1 bit is set by POR and SSI reset. 0 **HAS_DATA** — Transmit FIFO1 has data for transmission. 1 **EMPTY** — Transmit FIFO1 is empty.
0 TFE0	Transmit FIFO Empty 0. This flag is set when the empty slots in Tx FIFO exceed or are equal to the selected Tx FIFO WaterMark 0 (TFWM0) threshold. The setting of TFE0 only causes an interrupt when TIE and TFE0_EN are set and Tx FIFO0 is enabled. The TFE0 bit is automatically cleared when the data level in Tx FIFO0 becomes more than the amount specified by the watermark bits. The TFE0 bit is set by POR and SSI reset. 0 **HAS_DATA** — Transmit FIFO0 has data for transmission. 1 **EMPTY** — Transmit FIFO0 is empty.

61.9.5 SSI Interrupt Enable Register (SSIx_SIER)

The SSI Interrupt Enable Register (SIER) is a 25-bit register used to set up the SSI interrupts and DMA requests.

Address: Base address + 18h offset



SSIx_SIER field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24 RFRCIE	Receive Frame Complete Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
23 TFRCIE	Transmit Frame Complete Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not.

Table continues on the next page...

SSIx_SIER field descriptions (continued)

Field	Description
	0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
22 RDMAE	Receive DMA Enable. This bit allows SSI to request for DMA transfers. When enabled, DMA requests are generated when any of the RFF0/1 bits in the SISR are set and if the corresponding RFEN bit is also set. If the corresponding FIFO is disabled, a DMA request is generated when the corresponding RDR bit is set. 0 SSI Receiver DMA requests disabled. 1 SSI Receiver DMA requests enabled.
21 RIE	Receive Interrupt Enable. This control bit allows the SSI to issue receiver related interrupts to the Core. Refer to Receive Interrupt Enable Bit Description for a detailed description of this bit. 0 SSI Receiver Interrupt requests disabled. 1 SSI Receiver Interrupt requests enabled.
20 TDMAE	Transmit DMA Enable. This bit allows SSI to request for DMA transfers. When enabled, DMA requests are generated when any of the TFE0/1 bits in the SISR are set and if the corresponding TFEN bit is also set. If the corresponding FIFO is disabled, a DMA request is generated when the corresponding TDE bit is set. 0 SSI Transmitter DMA requests disabled. 1 SSI Transmitter DMA requests enabled.
19 TIE	Transmit Interrupt Enable. This control bit allows the SSI to issue transmitter data related interrupts to the Core. Refer to Transmit Interrupt Enable Bit Description for a detailed description of this bit. 0 SSI Transmitter Interrupt requests disabled. 1 SSI Transmitter Interrupt requests enabled.
18 CMDAUIE	Command Address Register Updated Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
17 CMDDUIE	Command Data Register Updated Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
16 RXTIE	Receive Tag Updated Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
15 RDR1IE	Receive Data Ready 1 Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
14 RDR0IE	Receive Data Ready 0 Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not.

Table continues on the next page...

SSIx_SIER field descriptions (continued)

Field	Description
	0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
13 TDE1IE	Transmit Data Register Empty 1 Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
12 TDE0IE	Transmit Data Register Empty 0 Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
11 ROE1IE	Receiver Overrun Error 1 Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
10 ROE0IE	Receiver Overrun Error 0 Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
9 TUE1IE	Transmitter Underrun Error 1 Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
8 TUE0IE	Transmitter Underrun Error 0 Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
7 TFSIE	Transmit Frame Sync Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
6 RFSIE	Receive Frame Sync Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
5 TLSIE	Transmit Last Time Slot Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not.

Table continues on the next page...

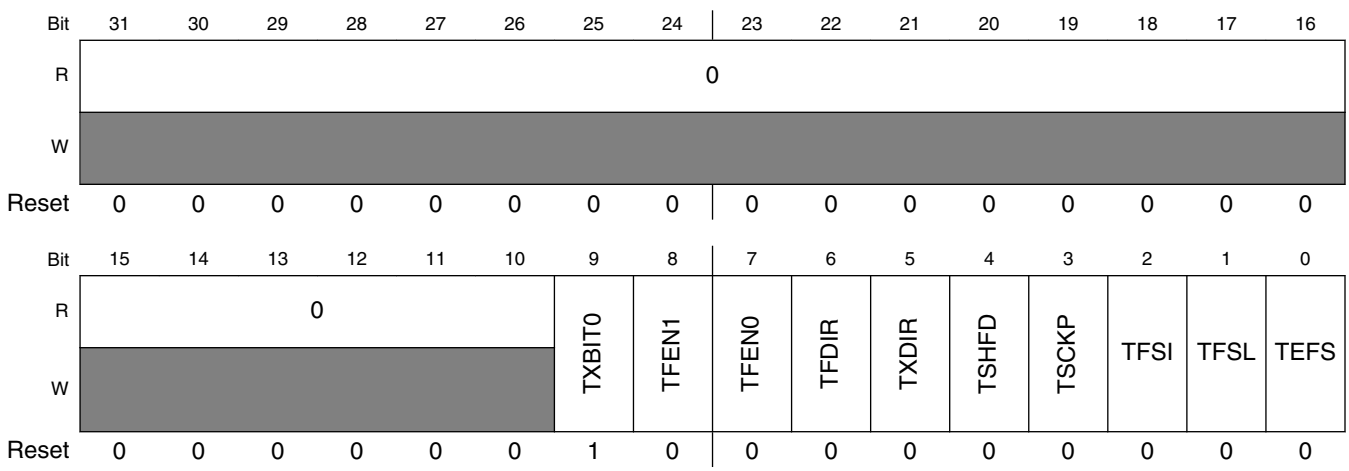
SSIx_SIER field descriptions (continued)

Field	Description
	0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
4 RLSIE	Receive Last Time Slot Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
3 RFF1IE	Receive FIFO Full 1 Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
2 RFF0IE	Receive FIFO Full 0 Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
1 TFE1IE	Transmit FIFO Empty 1 Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
0 TFE0IE	Transmit FIFO Empty 0 Interrupt Enable. Enable Bit. Controls whether the corresponding status bit in SISR can issue an interrupt to the core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.

61.9.6 SSI Transmit Configuration Register (SSIx_STCR)

The SSI Transmit Configuration Register (SSI_STCR) is a read/write control register used to direct the transmit operation of the STCR controls the direction of the bit clock and frame sync ports, STCK and STFS. Interrupt enable bit for the transmit sections is provided in this control register. The Power-on reset clears all SSI_STCR bits. However, SSI reset does not affect the SSI_STCR bits. The SSI_STCR bits are described in the following paragraphs. See the Programmable Registers section for the programming model of the SSI. The SSI Control Register (SSI_SCR) must first be set to enable interrupts. Next, the SSI interrupt bit in the Interrupt Enable Register (SSI_SIER) must be set to enable the interrupt. Finally, the interrupt can be enabled from within the SSI.

Address: Base address + 1Ch offset



SSIx_STCR field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value 0.
9 TXBIT0	Transmit Bit 0. This control bit allows SSI to transmit the data word from bit position 0 or 15/31 in the transmit shift register. The shifting data direction can be MSB or LSB first, controlled by the TSHFD bit. 0 **MSB_ALIGNED** — Shifting with respect to bit 31 (if word length = 16, 18, 20, 22 or 24) or bit 15 (if word length = 8, 10 or 12) of transmit shift register (MSB aligned). 1 **LSB_ALIGNED** — Shifting with respect to bit 0 of transmit shift register (LSB aligned).
8 TFEN1	Transmit FIFO Enable 1. This bit enables transmit FIFO 1. When enabled, the FIFO allows15 samples to be transmitted by the SSI (per channel) (a 9th sample can be shifting out) before TDE1 bit is set. When the FIFO is disabled, an interrupt is generated when a single sample is transferred to the transmit shift register (provided the interrupt is enabled). 0 Transmit FIFO 1 disabled. 1 Transmit FIFO 1 enabled.

Table continues on the next page...

SSIx_STCR field descriptions (continued)

Field	Description
7 TFEN0	Transmit FIFO Enable 0. This bit enables transmit FIFO 0. When enabled, the FIFO allows15 samples to be transmitted by the SSI per channel (a 9th sample can be shifting out) before TDE0 bit is set. When the FIFO is disabled, an interrupt is generated when a single sample is transferred to the transmit shift register (provided the interrupt is enabled). 0 Transmit FIFO 0 disabled. 1 Transmit FIFO 0 enabled.
6 TFDIR	Transmit Frame Direction. This bit controls the direction and source of the transmit frame sync signal. Internally generated frame sync signal is sent out through the STFS port and external frame sync is taken from the same port. 0 **EXTERNAL** — Frame Sync is external. 1 **INTERNAL** — Frame Sync generated internally.
5 TXDIR	Transmit Clock Direction. This bit controls the direction and source of the clock signal used to clock the TXSR. Internally generated clock is output through the STCK port. External clock is taken from this port. Refer to Table 61-2 for details of clock pin configurations. 0 **EXTERNAL** — Transmit Clock is external. 1 **INTERNAL** — Transmit Clock generated internally.
4 TSHFD	Transmit Shift Direction. This bit controls whether the MSB or LSB will be transmitted first in a sample. **NOTE:** The CODEC device labels the MSB as bit 0, whereas the Core labels the LSB as bit 0. Therefore, when using a standard CODEC, Core MSB (CODEC LSB) is shifted in first (TSHFD cleared). 0 **MSB_FIRST** — Data transmitted MSB first. 1 **LSB_FIRST** — Data transmitted LSB first.
3 TSCKP	Transmit Clock Polarity. This bit controls which bit clock edge is used to clock out data for the transmit section. Note: TSCKP is 0 CLK_IST = 0; TSCKP is 1CLK_IST = 1 0 **RISING_EDGE** — Data clocked out on rising edge of bit clock. 1 **FALLING_EDGE** — Data clocked out on falling edge of bit clock.
2 TFSI	Transmit Frame Sync Invert. This bit controls the active state of the frame sync I/O signal for the transmit section of SSI. 0 **ACTIVE_HIGH** — Transmit frame sync is active high. 1 **ACTIVE_LOW** — Transmit frame sync is active low.
1 TFSL	Transmit Frame Sync Length. This bit controls the length of the frame sync signal to be generated or recognized for the transmit section. The length of a word-long frame sync is same as the length of the data word selected by WL[3:0]. 0 **ONE_WORD** — Transmit frame sync is one-word long. 1 **ONE_CLOCK_BIT** — Transmit frame sync is one-clock-bit long.
0 TEFS	Transmit Early Frame Sync. This bit controls when the frame sync is initiated for the transmit section. The frame sync signal is deasserted after one bit-for-bit length frame sync and after one word-for-word length frame sync. In case of synchronous operation, the frame sync can also be initiated on receiving the first bit of data. 0 **FIRST_BIT** — Transmit frame sync initiated as the first bit of data is transmitted. 1 **ONE_BIT_BEFORE** — Transmit frame sync is initiated one bit before the data is transmitted.

61.9.7 SSI Receive Configuration Register (SSIx_SRCR)

The SSI Receive Configuration Register (SSI_SRCR) is a read/write control register used to direct the receive operation of the SSI. SSI.SRCR controls the direction of the bit clock and frame sync ports, SRCK and SRFS. Interrupt enable bit for the transmit sections is provided in this control register. The Power-on reset clears all SSI_SRCR bits. However, SSI reset does not affect the SSI_SRCR bits.

Address: Base address + 20h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					RXEXT	RXBIT0	RFEN1	RFEN0	RFDIR	RXDIR	RSHFD	RSCKP	RFSI	RFSL	REFS
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

SSIx_SRCR field descriptions

Field	Description
31–11 Reserved	This read-only field is reserved and always has the value 0.
10 RXEXT	Receive Data Extension. This control bit allows SSI to store the received data word in sign extended form. This bit affects data storage only in case received data is LSB aligned (SRCR[9]=1) 0 **OFF** — Sign extension turned off. 1 **ON** — Sign extension turned on.
9 RXBIT0	Receive Bit 0. This control bit allows SSI to receive the data word at bit position 0 or 15/31 in the receive shift register. The shifting data direction can be MSB or LSB first, controlled by the RSHFD bit. 0 **MSB_ALIGNED** — Shifting with respect to bit 31 (if word length = 16, 18, 20, 22 or 24) or bit 15 (if word length = 8, 10 or 12) of receive shift register (MSB aligned). 1 **LSB_ALIGNED** — Shifting with respect to bit 0 of receive shift register (LSB aligned).
8 RFEN1	Receive FIFO Enable 1. This bit enables receive FIFO 1. When enabled, the FIFO allows15 samples to be received by the SSI per channel (a 16th sample can be shifting in) before RDR1 bit is set. When the FIFO is disabled, an interrupt is generated when a single sample is received by the SSI (provided the interrupt is enabled). 0 Receive FIFO 1 disabled. 1 Receive FIFO 1 enabled.

Table continues on the next page...

SSIx_SRCR field descriptions (continued)

Field	Description
7 RFEN0	Receive FIFO Enable 0. This bit enables receive FIFO 0. When enabled, the FIFO allows15 samples to be received by the SSI (per channel) (a 16th sample can be shifting in) before RDR0 bit is set. When the FIFO is disabled, an interrupt is generated when a single sample is received by the SSI (provided the interrupt is enabled). 0 Receive FIFO 0 disabled. 1 Receive FIFO 0 enabled.
6 RFDIR	Receive Frame Direction. This bit controls the direction and source of the receive frame sync signal. Internally generated frame sync signal is sent out through the SRFS port and external frame sync is taken from the same port. 0 **EXTERNAL** — Frame Sync is external. 1 **INTERNAL** — Frame Sync generated internally.
5 RXDIR	Receive Clock Direction. This bit controls the direction and source of the clock signal used to clock the RXSR. Internally generated clock is output through the SRCK port. External clock is taken from this port. Refer to Table 61-2 for details on clock pin configurations. 0 **EXTERNAL** — Receive Clock is external. 1 **INTERNAL** — Receive Clock generated internally.
4 RSHFD	Receive Shift Direction. This bit controls whether the MSB or LSB will be received first in a sample. **NOTE:** The CODEC device labels the MSB as bit 0, whereas the Core labels the LSB as bit 0. Therefore, when using a standard CODEC, Core MSB (CODEC LSB) is shifted in first (RSHFD cleared). 0 **MSB_FIRST** — Data received MSB first. 1 **LSB_FIRST** — Data received LSB first.
3 RSCKP	Receive Clock Polarity. This bit controls which bit clock edge is used to latch in data for the receive section. 0 **FALLING_EDGE** — Data latched on falling edge of bit clock. 1 **RISING_EDGE** — Data latched on rising edge of bit clock.
2 RFSI	Receive Frame Sync Invert. This bit controls the active state of the frame sync I/O signal for the receive section of SSI. 0 **ACTIVE_HIGH** — Receive frame sync is active high. 1 **ACTIVE_LOW** — Receive frame sync is active low.
1 RFSL	Receive Frame Sync Length. This bit controls the length of the frame sync signal to be generated or recognized for the receive section. The length of a word-long frame sync is same as the length of the data word selected by WL[3:0]. 0 **ONE_WORD** — Receive frame sync is one-word long. 1 **ONE_CLOCK_BIT** — Receive frame sync is one-clock-bit long.
0 REFS	Receive Early Frame Sync. This bit controls when the frame sync is initiated for the receive section. The frame sync is disabled after one bit-for-bit length frame sync and after one word-for-word length frame sync. 0 **FIRST_BIT** — Receive frame sync initiated as the first bit of data is received. 1 **ONE_BIT_BEFORE** — Receive frame sync is initiated one bit before the data is received.

61.9.8 SSI Transmit Clock Control Register (SSIx_STCCR)

The SSI Transmit and Receive Control (SSI_STCCR and SSI_SRCCR) registers are 19-bit, read/write control registers used to direct the operation of the SSI. The Clock Controller Module (CCM) can source the SSI clock (SSI's sys clock from CCM's ssi_clk_root) from multiple sources and perform fractional division to support commonly used audio bit rates. The CCM can maintain the SSI's sys clock frequency at a constant rate even in cases where the ipg_clk (from CCM) frequency changes. These registers control the SSI clock generator, bit and frame sync rates, word length, and number of words per frame for the serial data. The SSI.STCCR register is dedicated to the transmit section, and the SSI_SRCCR register is dedicated to the receive section except in Synchronous mode, in which the SSI_STCCR register controls both the receive and transmit sections. Power-on reset clears all SSI_STCCR and SSI_SRCCR bits. SSI reset does not affect the SSI_STCCR and SSI_SRCCR bits. The control bits are described in the following paragraphs. Although the bit patterns of the SSI_STCCR and SSI_SRCCR registers are the same, the contents of these two registers can be programmed differently.

Table 61-12. SSI Data Length

WL3	WL2	WL1	WL0	Number of Bits/Word	Supported in Implementation
0	0	0	0	2	No
0	0	0	1	4	No
0	0	1	0	6	No
0	0	1	1	8	Yes
0	1	0	0	10	Yes
0	1	0	1	12	Yes
0	1	1	0	14	No
0	1	1	1	16	Yes
1	0	0	0	18	Yes
1	0	0	1	20	Yes
1	0	1	0	22	Yes
1	0	1	1	24	Yes
1	1	0	0	26	No
1	1	0	1	28	No
1	1	1	0	30	No
1	1	1	1	32	No

Address: Base address + 24h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													DIV2	PSR	WL3_ WL0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WL3_WL0			DC4_DC0					PM7_PM0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSIx_STCCR field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value 0.
18 DIV2	Divide By 2. This bit controls a divide-by-two divider in series with the rest of the prescalers. 0 Divider bypassed. 1 Divider used to divide clock by 2.
17 PSR	Prescaler Range. This bit controls a fixed divide-by-eight prescaler in series with the variable prescaler. It extends the range of the prescaler for those cases where a slower bit clock is required. 0 Prescaler bypassed. 1 Prescaler used to divide clock by 8.
16–13 WL3_WL0	Word Length Control. These bits are used to control the length of the data words being transferred by the SSI. These bits control the Word Length Divider in the Clock Generator. They also control the frame sync pulse length when the FSL bit is cleared. In I2S Master mode, the SSI works with a fixed word length of 32, and the WL bits are used to control the amount of valid data in those 32 bits. In AC97 Mode of operation, if word length is set to any value other than 16 bits, it will result in a word length of 20 bits.
12–8 DC4_DC0	Frame Rate Divider Control. These bits are used to control the divide ratio for the programmable frame rate dividers. The divide ratio works on the word clock. In Normal mode, this ratio determines the word transfer rate. In Network mode, this ratio sets the number of words per frame. The divide ratio ranges from 1 to 32 in Normal mode and from 2 to 32 in Network mode. In Normal mode, a divide ratio of 1 (DC=00000) provides continuous periodic data word transfer. A bit-length frame sync must be used in this case. These bits can be programmed with values ranging from "00000" to "11111" to control the number of words in a frame.
PM7_PM0	Prescaler Modulus Select. These bits control the prescale divider in the clock generator. This prescaler is used only in Internal Clock mode to divide the internal clock. The bit clock output is available at the clock port. A divide ratio from 1 to 256 (PM[7:0] = 0x00 to 0xFF) can be selected. Refer to DIV2, PSR and PM Bit Description for details regarding settings.

61.9.9 SSI Receive Clock Control Register (SSIx_SRCCR)

The SSI Transmit and Receive Control (SSI_STCCR and SSI_SRCCR) registers are 19-bit, read/write control registers used to direct the operation of the SSI. The Clock Controller Module (CCM) can source the SSI clock (SSI's sys clock-from CCM's ssi_clk_root) from multiple sources and perform fractional division to support commonly used audio bit rates. The CCM can maintain the SSI's sys clock frequency at a constant rate even in cases where the ipg_clk from CCM frequency changes. These registers control the SSI clock generator, bit and frame sync rates, word length, and number of words per frame for the serial data. The SSI_STCCR register is dedicated to the transmit section, and the SSI_SRCCR register is dedicated to the receive section except in Synchronous mode, in which the SSI_STCCR register controls both the receive and transmit sections. Power-on reset clears all SSI_STCCR and SSI_SRCCR bits. SSI reset does not affect the SSI_STCCR and SSI_SRCCR bits. The control bits are described in the following paragraphs. Although the bit patterns of the SSI_STCCR and SSI_SRCCR registers are the same, the contents of these two registers can be programmed differently.

Table 61-13. SSI Data Length

WL3	WL2	WL1	WL0	Number of Bits/Word	Supported in Implementation
0	0	0	0	2	No
0	0	0	1	4	No
0	0	1	0	6	No
0	0	1	1	8	Yes
0	1	0	0	10	Yes
0	1	0	1	12	Yes
0	1	1	0	14	No
0	1	1	1	16	Yes
1	0	0	0	18	Yes
1	0	0	1	20	Yes
1	0	1	0	22	Yes
1	0	1	1	24	Yes
1	1	0	0	26	No
1	1	0	1	28	No
1	1	1	0	30	No
1	1	1	1	32	No

Address: Base address + 28h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													DIV2	PSR	WL3_ WL0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WL3_WL0			DC4_DC0					PM7_PM0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSIx_SRCCR field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value 0.
18 DIV2	Divide By 2. This bit controls a divide-by-two divider in series with the rest of the prescalers. 0 Divider bypassed. 1 Divider used to divide clock by 2.
17 PSR	Prescaler Range. This bit controls a fixed divide-by-eight prescaler in series with the variable prescaler. It extends the range of the prescaler for those cases where a slower bit clock is required. 0 Prescaler bypassed. 1 Prescaler used to divide clock by 8.
16–13 WL3_WL0	Word Length Control. These bits are used to control the length of the data words being transferred by the SSI. These bits control the Word Length Divider in the Clock Generator. They also control the frame sync pulse length when the FSL bit is cleared. In I2S Master mode, the SSI works with a fixed word length of 32, and the WL bits are used to control the amount of valid data in those 32 bits. In AC97 Mode of operation, if word length is set to any value other than 16 bits, it will result in a word length of 20 bits.
12–8 DC4_DC0	Frame Rate Divider Control. These bits are used to control the divide ratio for the programmable frame rate dividers. The divide ratio works on the word clock. In Normal mode, this ratio determines the word transfer rate. In Network mode, this ratio sets the number of words per frame. The divide ratio ranges from 1 to 32 in Normal mode and from 2 to 32 in Network mode. In Normal mode, a divide ratio of 1 (DC=00000) provides continuous periodic data word transfer. A bit-length frame sync must be used in this case. These bits can be programmed with values ranging from "00000" to "11111" to control the number of words in a frame.
PM7_PM0	Prescaler Modulus Select. These bits control the prescale divider in the clock generator. This prescaler is used only in Internal Clock mode to divide the internal clock. The bit clock output is available at the clock port. A divide ratio from 1 to 256 (PM[7:0] = 0x00 to 0xFF) can be selected. Refer to DIV2, PSR and PM Bit Description for details regarding settings.

61.9.10 SSI FIFO Control/Status Register (SSIx_SFCSR)

The SSI FIFO Control / Status Register indicates the status of the Transmit FIFO Empty flag, with different settings of the Transmit FIFO WaterMark bits and varying amounts of data in the Tx FIFO

.

Table 61-14. Status of Transmit FIFO Empty Flag

Transmit FIFO Watermark (TFWM)	Number of data in Tx-Fifo														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
2	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
3	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
5	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
6	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
7	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
8	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
9	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
10	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
11	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
12	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
13	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address: Base address + 2Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RFCNT1				TFCNT1				RFWM1				TFWM1				RFCNT0				TFCNT0				RFWM0				TFWM0			
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

SSIx_SFCSR field descriptions

Field	Description
31–28 RFCNT1	Receive FIFO Counter1. These bits indicate the number of data words in Receive FIFO 1. 0000 0 data word in receive FIFO 0001 1 data word in receive FIFO 0010 2 data word in receive FIFO 0011 3 data word in receive FIFO 0100 4 data word in receive FIFO 0101 5 data word in receive FIFO 0110 6 data word in receive FIFO 0111 7 data word in receive FIFO 1000 8 data word in receive FIFO 1001 9 data word in receive FIFO 1010 10 data word in receive FIFO 1011 11 data word in receive FIFO 1100 12 data word in receive FIFO 1101 13 data word in receive FIFO

Table continues on the next page...

SSIx_SFCSR field descriptions (continued)

Field	Description
	1110 14 data word in receive FIFO 1111 15 data word in receive FIFO
27–24 TFCNT1	Transmit FIFO Counter1. These bits indicate the number of data words in Transmit FIFO. 0000 0 data word in transmit FIFO 0001 1 data word in transmit FIFO 0010 2 data word in transmit FIFO 0011 3 data word in transmit FIFO 0100 4 data word in transmit FIFO 0101 5 data word in transmit FIFO 0110 6 data word in transmit FIFO 0111 7 data word in transmit FIFO 1000 8 data word in transmit FIFO 1001 9 data word in transmit FIFO 1010 10 data word in transmit FIFO 1011 11 data word in transmit FIFO 1100 12 data word in transmit FIFO 1101 13 data word in transmit FIFO 1110 14 data word in transmit FIFO 1111 15 data word in transmit FIFO
23–20 RFWM1	Receive FIFO Full WaterMark 1. These bits control the threshold at which the RFF1 flag will be set. The RFF1 flag is set whenever the data level in Rx FIFO 1 reaches the selected threshold. 0000 Reserved 0001 RFF set when at least one data word has been written to the Receive FIFO. Set when RxFIFO = 1,2.....15 data words 0010 RFF set when 2 or more data words have been written to the Receive FIFO. Set when RxFIFO = 2,3.....15 data words 0011 RFF set when 3 or more data words have been written to the Receive FIFO. Set when RxFIFO = 3,4.....15 data words 0100 RFF set when 4 or more data words have been written to the Receive FIFO. Set when RxFIFO = 4,5.....15 data words 0101 RFF set when 5 or more data words have been written to the Receive FIFO. Set when RxFIFO = 5,6.....15 data words 0110 RFF set when 6 or more data words have been written to the Receive.. Set when RxFIFO = 6,7.....15 data words 0111 RFF set when 7 or more data words have been written to the Receive FIFO. Set when RxFIFO = 7,8.....15 data words 1000 RFF set when 8 or more data words have been written to the Receive FIFO. Set when RxFIFO =8,9.....15 data words 1001 RFF set when 9 or more data words have been written to the Receive FIFO. Set when RxFIFO = 9,10.....15 data words 1010 RFF set when 10 or more data words have been written to the Receive FIFO. Set when RxFIFO = 10,11.....15 data words 1011 RFF set when 11 or more data words have been written to the Receive FIFO. Set when RxFIFO = 11,12.....15 data words 1100 RFF set when 12 or more data words have been written to the Receive FIFO. Set when RxFIFO = 12,13.....15 data words

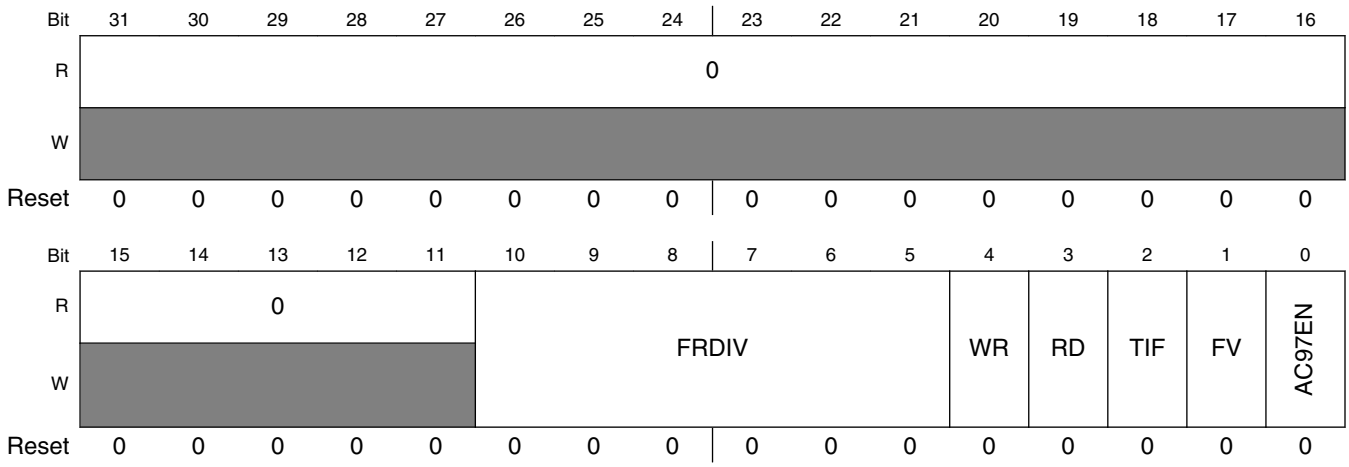
Table continues on the next page...

SSIx_SFCSR field descriptions (continued)

Field	Description
	1101 RFF set when 13 or more data words have been written to the Receive FIFO. Set when RxFIFO = 13,14,15data words 1110 RFF set when 14 or more data words have been written to the Receive FIFO. Set when RxFIFO = 14,15 data words 1111 RFF set when 15 data words have been written to the Receive FIFO (default). Set when RxFIFO = 15 data words
19–16 TFWM1	Transmit FIFO Empty WaterMark 1. These bits control the threshold at which the TFE1 flag will be set. The TFE1 flag is set whenever the empty slots in Tx FIFO exceed or are equal to the selected threshold. 0000 Reserved 0001 TFE set when there are more than or equal to 1 empty slots in Transmit FIFO (default). Transmit FIFO empty is set when TxFIFO <= 14 data. 0010 TFE set when there are more than or equal to 2 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <=13 data. 0011 TFE set when there are more than or equal to 3 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <=12 data. 0100 TFE set when there are more than or equal to 4 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <=11 data. 0101 TFE set when there are more than or equal to 5 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <=10 data. 0110 TFE set when there are more than or equal to 6 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <=9 data. 0111 TFE set when there are more than or equal to 7 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <=8 data. 1000 TFE set when there are more than or equal to 8 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <=7 data. 1001 TFE set when there are more than or equal to 9 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <= 6 data. 1010 TFE set when there are more than or equal to 10 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <= 5 data. 1011 TFE set when there are more than or equal to 11 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <= 4 data. 1100 TFE set when there are more than or equal to 12 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <= 3 data. 1101 TFE set when there are more than or equal to 13 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <= 2 data. 1110 TFE set when there are more than or equal to 14 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <= 1 data. 1111 TFE set when there are 15 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO = 0 data.
15–12 RFCNT0	Receive FIFO Counter 0. These bits indicate the number of data words in Receive FIFO 0. See SSI_SFCSR[RFCNT1] for details regarding settings for receive FIFO counter bits.
11–8 TFCNT0	Transmit FIFO Counter 0. These bits indicate the number of data words in Transmit FIFO 0. See SSI_SFCSR[TFCNT1] for details regarding settings for transmit FIFO counter bits.
7–4 RFWM0	Receive FIFO Full WaterMark 0. These bits control the threshold at which the RFF0 flag will be set. The RFF0 flag is set whenever the data level in Rx FIFO 0 reaches the selected threshold. See SSI_SFCSR[RFWM1] for details regarding settings for receive FIFO watermark bits.
TFWM0	Transmit FIFO Empty WaterMark 0. These bits control the threshold at which the TFE0 flag will be set. The TFE0 flag is set whenever the empty slots in Tx FIFO exceed or are equal to the selected threshold. See SSI_SFCSR[TFWM0] for details regarding settings for transmit FIFO watermark bits.

61.9.11 SSI AC97 Control Register (SSIx_SACNT)

Address: Base address + 38h offset



SSIx_SACNT field descriptions

Field	Description
31–11 Reserved	This read-only field is reserved and always has the value 0.
10–5 FRDIV	Frame Rate Divider. These bits control the frequency of AC97 data transmission/reception. They are programmed with the number of frames for which the SSI should be idle, after operating in one frame. Through these bits, AC97 frequency of operation, from 48 KHz (000000) to 1 KHz (101111) can be achieved. Sample Value: 001010 (10 Decimal) = SSI will operate once every 11 frames.
4 WR	Write Command. This bit specifies whether the next frame will carry an AC97 Write Command or not. The programmer should take care that only one of the bits (WR or RD) is set at a time. When this bit is set, the corresponding tag bits (corresponding to Command Address and Command Data slots of the next Tx frame) are automatically set. This bit is automatically cleared by the SSI after completing transmission of a frame. 0 Next frame will not have a Write Command. 1 Next frame will have a Write Command.
3 RD	Read Command. This bit specifies whether the next frame will carry an AC97 Read Command or not. The programmer should take care that only one of the bits (WR or RD) is set at a time. When this bit is set, the corresponding tag bit (corresponding to Command Address slot of the next Tx frame) is automatically set. This bit is automatically cleared by the SSI after completing transmission of a frame. 0 Next frame will not have a Read Command. 1 Next frame will have a Read Command.
2 TIF	Tag in FIFO. This bit controls the destination of the information received in AC97 tag slot (Slot #0). 0 **SATAG_REGISTER** — Tag info stored in SATAG register. 1 **RX_FIFO0** — Tag info stored in Rx FIFO 0.
1 FV	Fixed/Variable Operation. This bit selects whether the SSI is in AC97 Fixed mode or AC97 Variable mode.

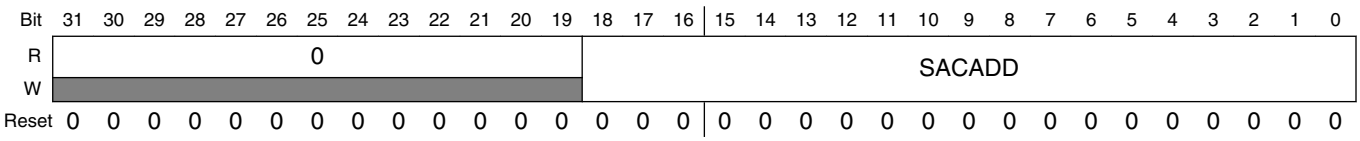
Table continues on the next page...

SSIx_SACNT field descriptions (continued)

Field	Description
	0 **FIXED** — AC97 Fixed Mode. 1 **VARIABLE** — AC97 Variable Mode.
0 AC97EN	AC97 Mode Enable. This bit is used to enable SSI AC97 operation. Refer to AC97 Mode for details of AC97 operation. 0 AC97 mode disabled. 1 SSI in AC97 mode.

61.9.12 SSI AC97 Command Address Register (SSIx_SACADD)

Address: Base address + 3Ch offset

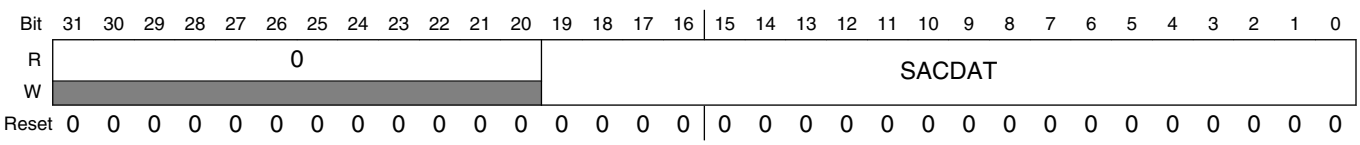


SSIx_SACADD field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value 0.
SACADD	AC97 Command Address. These bits store the Command Address Slot information (bit 19 of the slot is sent in accordance with the Read and Write Command bits in SSI_SACNT register). These bits can be updated by a direct write from the Core. They are also updated with the information received in the incoming Command Address Slot. If the contents of these bits change due to an update, the CMDAU bit in SISR is set.

61.9.13 SSI AC97 Command Data Register (SSIx_SACDAT)

Address: Base address + 40h offset



SSIx_SACDAT field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value 0.
SACDAT	AC97 Command Data. The outgoing Command Data Slot carries the information contained in these bits. These bits can be updated by a direct write from the Core. They are also updated with the information received in the incoming Command Data Slot. If the contents of these bits change due to an update, the

Table continues on the next page...

SSIx_SACDAT field descriptions (continued)

Field	Description
	CMDDU bit in SISR is set. These bits are transmitted only during AC97 Write Command. During AC97 Read Command, 0x00000 is transmitted in time slot #2.

61.9.14 SSI AC97 Tag Register (SSIx_SATAG)

Address: Base address + 44h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																SATAG															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSIx_SATAG field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
SATAG	AC97 Tag Value. Writing to this register (by the Core) sets the value of the Tx-Tag in AC97 fixed mode of operation. On a read, the Core gets the Rx-Tag Value received (in the last frame) from the Codec. If TIF bit in SSI_SACNT register is set, the TAG value is also stored in Rx-FIFO in addition to SATAG register. When the received Tag value changes, the RXT bit in SISR register is set. Bits SATAG[1:0] convey the Codec -ID. In current implementation only Primary Codecs are supported. Thus writing value 2'b00 to this field is mandatory.

61.9.15 SSI Transmit Time Slot Mask Register (SSIx_STMSK)

Address: Base address + 48h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STMSK																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSIx_STMSK field descriptions

Field	Description
STMSK	Transmit Mask. These bits indicate which slot has been masked in the current frame. The Core can write to this register to control the time slots in which the SSI transmits data. Each bit has info corresponding to the respective time slot in the frame. Transmit mask bits should not be used in I2S Slave mode of operation. SSI_STMSK register value must be set before enabling Transmission. 0 Valid Time Slot. 1 Time Slot masked (no data transmitted in this time slot).

61.9.16 SSI Receive Time Slot Mask Register (SSIx_SRMSK)

Address: Base address + 4Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SRMSK																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSIx_SRMSK field descriptions

Field	Description
SRMSK	Receive Mask. These bits indicate which slot has been masked in the current frame. The Core can write to this register to control the time slots in which the SSI receives data. Each bit has info corresponding to the respective time slot in the frame. SSI_SRMSK register value must be set before enabling Receiver. Receive mask bits should not be used in I2S Slave mode of operation. 0 Valid Time Slot. 1 Time Slot masked (no data received in this time slot).

61.9.17 SSI AC97 Channel Status Register (SSIx_SACCST)

Address: Base address + 50h offset

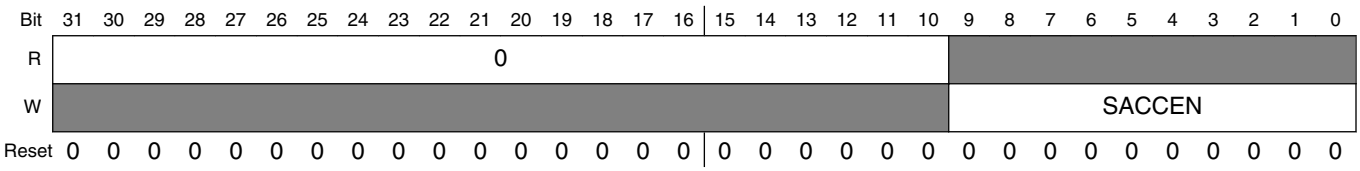
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																						SACCST									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSIx_SACCST field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value 0.
SACCST	AC97 Channel Status. These bits indicate which data slot has been enabled in AC97 variable mode operation. This register is updated in case the core enables/disables a channel through a write to SSI_SACCEN/SSI_SACCDIS register or the external codec enables a channel by sending a '1' in the corresponding SLOTREQ bit. Bit [0] corresponds to the first data slot in an AC97 frame (Slot #3) and Bit [9] corresponds to the tenth data slot (slot #12). The contents of this register only have relevance while the SSI is operating in AC97 variable mode. Writes to this register result in an error response on the block interface. 0 Data channel disabled. 1 Data channel enabled.

61.9.18 SSI AC97 Channel Enable Register (SSIx_SACCEN)

Address: Base address + 54h offset

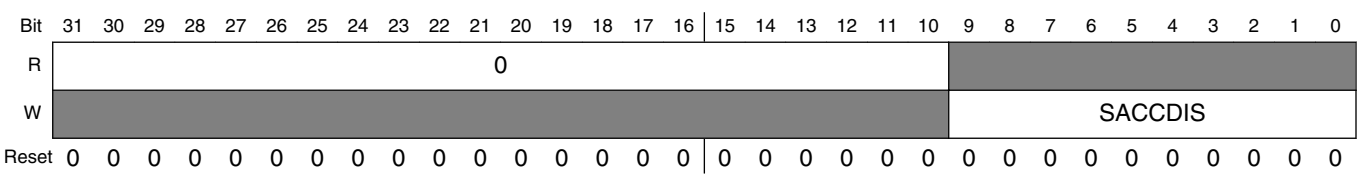


SSIx_SACCEN field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value 0.
SACCEN	AC97 Channel Enable. The Core writes a '1' to these bits to enable an AC97 data channel. Writing a '0' has no effect. Bit [0] corresponds to the first data slot in an AC97 frame (Slot #3) and Bit [9] corresponds to the tenth data slot (slot #12). Writes to these bits only have effect in the AC97 Variable mode of operation. These bits are always read as '0' by the Core. 0 Write Has no effect. 1 Write Enables the corresponding data channel.

61.9.19 SSI AC97 Channel Disable Register (SSIx_SACCDIS)

Address: Base address + 58h offset



SSIx_SACCDIS field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value 0.
SACCDIS	AC97 Channel Disable. The Core writes a '1' to these bits to disable an AC97 data channel. Writing a '0' has no effect. Bit [0] corresponds to the first data slot in an AC97 frame (Slot #3) and Bit [9] corresponds to the tenth data slot (slot #12). Writes to these bits only have effect in the AC97 Variable mode of operation. These bits are always read as '0' by the Core. 0 Write Has no effect. 1 Write Disables the corresponding data channel.

Chapter 62
Temperature Monitor (TEMPMON)

62.1 Overview

The temperature sensor module implements a temperature sensor/conversion function based on a temperature-dependent voltage to time conversion.

The module features an alarm function that can raise an interrupt signal if the temperature is above a specified threshold. A self-repeating mode can also be programmed which executes a temperature sensing operation based on a programmed delay.

Software can use this module to monitor the on-die temperature and take appropriate actions such as throttling back the core frequency when a temperature interrupt is set.

The high-level implementation of the temperature sensor is shown in the figure below.

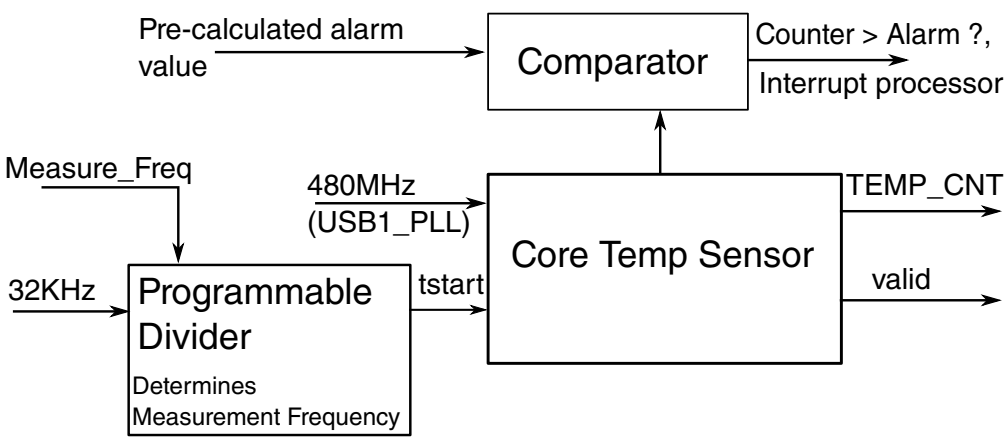


Figure 62-1. High Level Temp Sensor System Diagram

As shown in the figure above, the temperature sensor uses and assumes that the bandgap reference, 480MHz PLL and 32KHz RTC modules are properly programmed and fully settled for correct operation.

62.2 Software Usage Guidelines

During normal system operation software can use the temperature sensor counter output (TEMP_CNT) in conjunction with the fused temperature calibration data to determine the on-die operational temperature or to set an over-temperature interrupt alarm to within a couple of °C.

Based on calibration, two sets of temperature and counter values will be available via fuses on the device. These data points will correspond to the points (N_1, T_1) and (N_2, T_2) in the curve below.

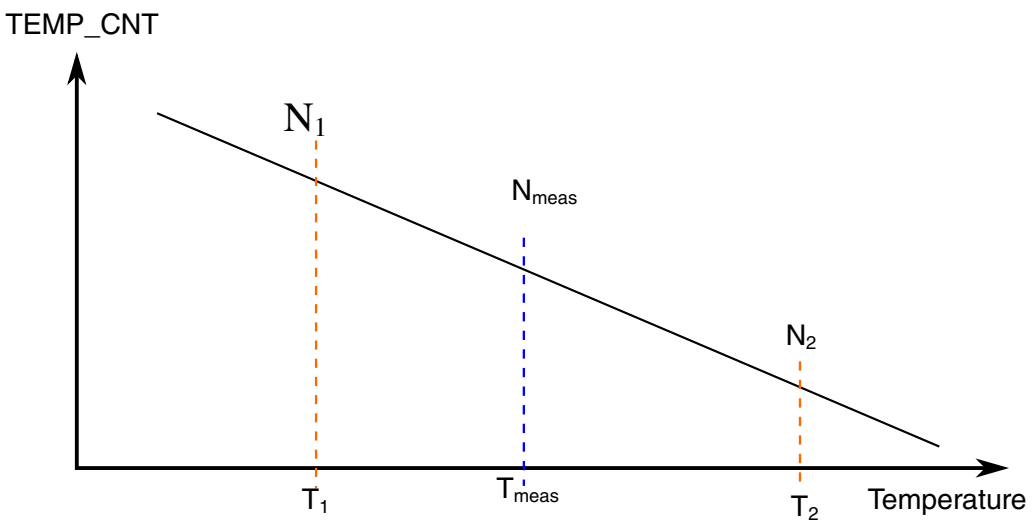


Figure 62-2. Temperature Measurement Cycle

After a temperature measurement cycle, software should use the calibration points in conjunction with the temperature code value in the TEMPMON_TEMPSENSE0[TEMP_CNT] bitfield to calculate the temperature for the device using the following equation:

$$T_{meas} = T_2 - (N_{meas} - N_2) * ((T_2 - T_1) / (N_1 - N_2))$$

Likewise, to determine the alarm counter value to be written in the TEMPMON_TEMPSENSE0 register for a temperature based interrupt, the above equation can be solved for the N_{meas} value that should be used based on the desired temperature trigger.

The temperature calibration point fuse values are available in the OCOTP_ANA1 register. The temperature calibration values are fused individually for each part in the product testing process. The fields of this register are described in the following table.

Table 62-1. OCOTP_ANA1 Temperature Sensor Calibration Data

Bit Range	Bit Mask	Name	Description
[31:20]	FFF0_0000h	ROOM_COUNT	Value of TEMPMON_TEMPSENSE0[TEMP_VALUE] after a measurement cycle at room temperature (25.0 °C).
[19:8]	000F_FF00h	HOT_COUNT	Value of TEMPMON_TEMPSENSE0[TEMP_VALUE] after a measurement cycle at the hot temperature, i.e. HOT_TEMP.
[7:0]	0000_00FFh	HOT_TEMP	The hot temperature test point. Each LSB equals 1 °C.

The points on the calibration curve are as follows.

- (N_1, T_1) = (ROOM_COUNT, 25.0)
- (N_2, T_2) = (HOT_COUNT, HOT_TEMP)
- (N_{meas}, T_{meas}) = (TEMP_CNT, T_{meas})

Substituting the fields from OCOTP_ANA1 into the earlier equation results in the following:

T_{meas} = HOT_TEMP - (N_{meas} - HOT_COUNT) * ((HOT_TEMP - 25.0) / (ROOM_COUNT – HOT_COUNT))

62.3 TEMPMON Memory Map/Register Definition

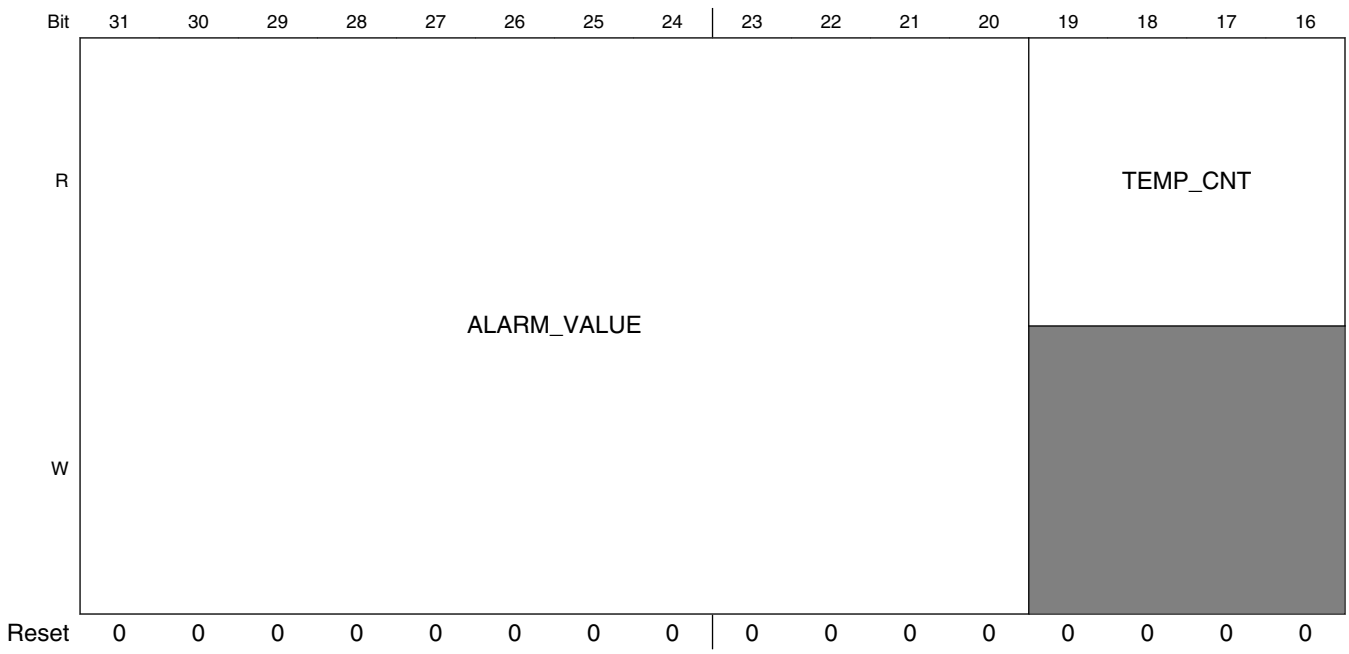
TEMPMON memory map

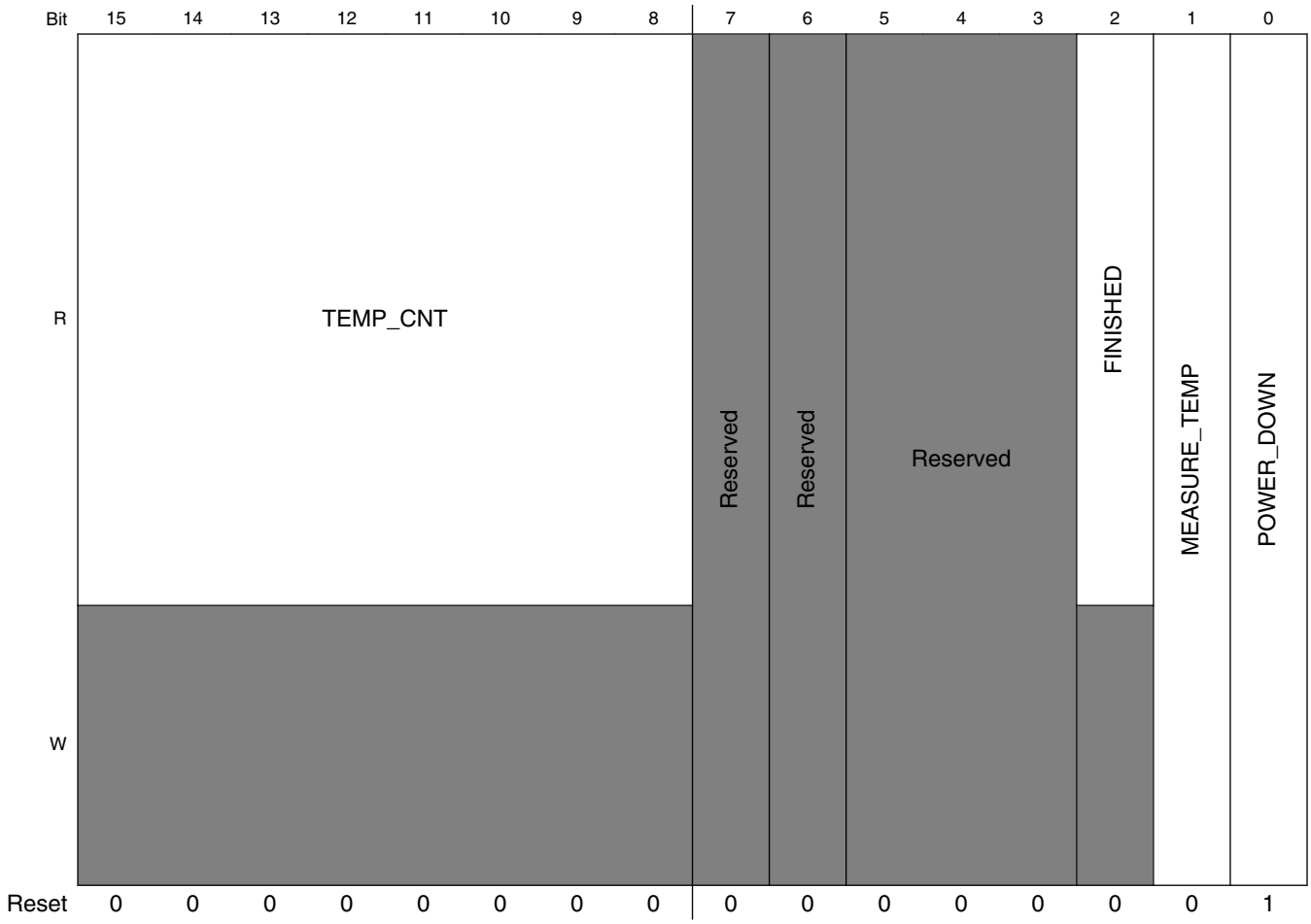
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_8180	Tempsensor Control Register 0 (TEMPMON_TEMPSENSE0)	32	R/W	0000_0001h	62.3.1/5128
20C_8184	Tempsensor Control Register 0 (TEMPMON_TEMPSENSE0_SET)	32	R/W	0000_0001h	62.3.1/5128
20C_8188	Tempsensor Control Register 0 (TEMPMON_TEMPSENSE0_CLR)	32	R/W	0000_0001h	62.3.1/5128
20C_818C	Tempsensor Control Register 0 (TEMPMON_TEMPSENSE0_TOG)	32	R/W	0000_0001h	62.3.1/5128
20C_8190	Tempsensor Control Register 1 (TEMPMON_TEMPSENSE1)	32	R/W	0000_0001h	62.3.2/5130
20C_8194	Tempsensor Control Register 1 (TEMPMON_TEMPSENSE1_SET)	32	R/W	0000_0001h	62.3.2/5130
20C_8198	Tempsensor Control Register 1 (TEMPMON_TEMPSENSE1_CLR)	32	R/W	0000_0001h	62.3.2/5130
20C_819C	Tempsensor Control Register 1 (TEMPMON_TEMPSENSE1_TOG)	32	R/W	0000_0001h	62.3.2/5130

62.3.1 Tempsensor Control Register 0 (TEMPMON_TEMPSENSE0*n*)

This register defines the basic controls for the temperature sensor minus the frequency of automatic sampling which is defined in the tempsensor.

Address: 20C_8000h base + 180h offset + (4d × i), where i=0d to 3d





TEMPMON_TEMPSENSE0*n* field descriptions

Field	Description
31–20 ALARM_VALUE	This bit field contains the temperature count (raw sensor output) that will generate a high alarm when TEMP_CNT is smaller than this field.
19–8 TEMP_CNT	This bit field contains the last measured temperature count.
7 -	This field is reserved. Reserved.
6 -	This field is reserved. Reserved.
5–3 -	This field is reserved. Reserved
2 FINISHED	Indicates that the latest temp is valid. This bit should be cleared by the sensor after the start of each measurement. 0 **INVALID** — Last measurement is not ready yet. 1 **VALID** — Last measurement is valid.
1 MEASURE_ TEMP	Starts the measurement process. If the measurement frequency is zero in the TEMPSENSE1 register, this results in a single conversion.

Table continues on the next page...

TEMPMON_TEMPSENSE0*n* field descriptions (continued)

Field	Description
	0 **STOP** — Do not start the measurement process. 1 **START** — Start the measurement process.
0 POWER_DOWN	This bit powers down the temperature sensor. 0 **POWER_UP** — Enable power to the temperature sensor. 1 **POWER_DOWN** — Power down the temperature sensor.

62.3.2 Tempsensor Control Register 1 (TEMPMON_TEMPSENSE1*n*)

This register defines the automatic repeat time of the temperature sensor.

Address: 20C_8000h base + 190h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																MEASURE_FREQ															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

TEMPMON_TEMPSENSE1*n* field descriptions

Field	Description
31–16 -	This field is reserved. Reserved.
MEASURE_ FREQ	This bits determines how many RTC clocks to wait before automatically repeating a temperature measurement. The pause time before remeasuring is the field value multiplied by the RTC period. 0x0000 Defines a single measurement with no repeat. 0x0001 Updates the temperature value at a RTC clock rate. 0x0002 Updates the temperature value at a RTC/2 clock rate. . . . — 0xFFFF Determines a two second sample period with a 32.768KHz RTC clock. Exact timings depend on the accuracy of the RTC clock.

Chapter 63
TrustZone Address Space Controller (TZASC)

63.1 Overview

The TrustZone Address Space Controller (TZASC) protects security-sensitive SW and data in a trusted execution environment against potentially compromised SW running on the platform.

The TZASC block diagram is shown in figure below.

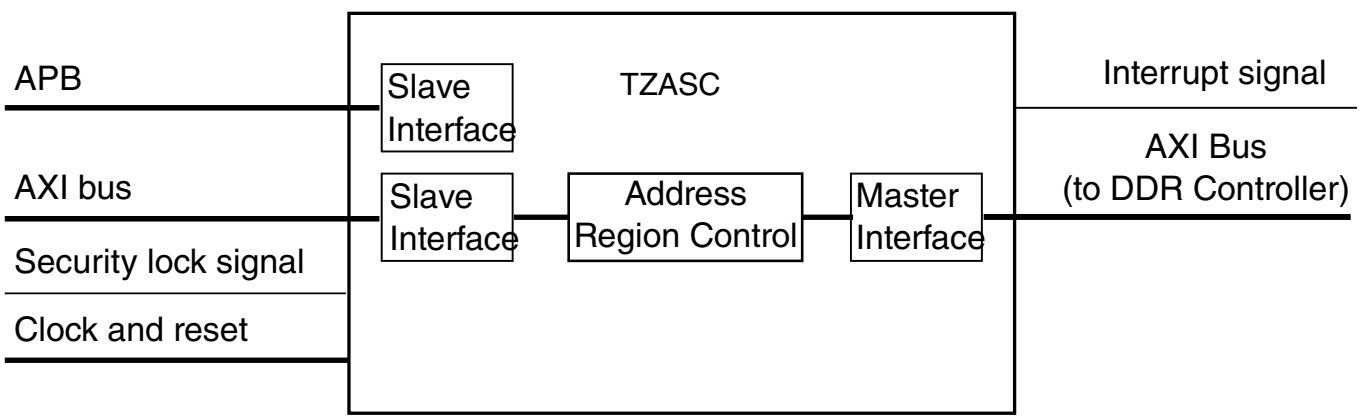


Figure 63-1. TZASC Block Diagram

The TZASC is an IP by Arm ("CoreLink™ TrustZone Address Space Controller TZC-380"), designed to provide configurable protection over program (SW) memory space.

The main features of TZASC are:

- Supports 16 independent address regions
- Access controls are independently programmable for each address region
- Sensitive registers may be locked
- Host interrupt may be programmed to signal attempted access control violations

- AXI master/slave interfaces for transactions
- APB slave interface for configuration and status reporting

63.2 Clocks

The table found here describes the clock sources for TZASC.

See Clock Controller Module (CCM) for clock setting, configuration, and gating information.

Table 63-1. TZASC Clocks

Clock name	Clock Root	Description
aclk	mmdc_ch0_axi_clk_root	Module clock

63.3 i.MX 6Solo/6DualLite Specific Configuration

The i.MX 6Solo/6DualLite uses two TZASC instances, one on each of the two DDR controllers (MMDC0, MMDC1 modules), to provide address protection based on access security level.

The i.MX 6Solo/6DualLite utilizes bus muxing logic to route DDR memory traffic through or bypass the TZASC modules. By default, the TZASC modules are bypassed, and their clocks are gated off.

TZASC-1 provides protection on memory accessible via MMDC0, while TZASC-2 provides protection for memory accessible via MMDC1.

Enabling TZASCs is expected to have a slight impact on memory performance. Exact value cannot be stated, since varies, depending on specific application software.

The proper and preferred method of enabling the TZASCs, is by burning the TZASC_ENABLE fuse.

For every power-up cycle, with TZASC_ENABLE fuse burned, the Boot ROM code will seamlessly handle the enable and engage of the TZASC modules and their clocks, leaving them in their active state, (i.e. enabled, and not bypassed). From this state and on, it is the responsibility of OS image, to configure the memory regions protection, per a specific application / use-case needs.

A configuration lock bits ("TZASC1_BOOT_LOCK", "TZASC2_BOOT_LOCK" in GPR3 register of IOMUXC) once set, will block any attempts to change the security settings, past the OS image configuration code settings. The configuration locking is in place, until the next hardware reset cycle.

NOTE

Engaging the TZASC functionality (i.e. - not bypassed), has to be done while DDR bus is guaranteed to be idle, with no pending transactions.

Enabling TZASC, w/o use of the associated fuse, is possible, but not trivial, since has to be done while traffic to DDR is guaranteed to be stopped. A typical way of achieving this, is by:

- Ensuring no other master can issue accesses to DDR.
- Protect against program flow change, to DDR space, (Disable interrupts, ans such).
- Run switching code from internal RAM.

TZASC enabling code has to handle the data-path mux control (via TZASC1_BYPASS, TZASC2_BYPASS, bits in GPR9, in IOMUXC module) as well as clock enabling (in LPCG module), and configuration locking, if desired (via set of TZASC1_BOOT_LOCK, TZASC2_BOOT_LOCK bits in GPR3, IOMUXC module).

The TZASC_BYPASS bit(s) in GPR9 register, once set, preserve their values until the next power-up cycle ("Sticky" type), in order to protect against unauthorized 'disable' operation.

63.4 Address Mapping in various memory mapping modes

The address configured to the TZASC controller(s) must match the "local addresses" as being passed on to the DDR controller(s).

The DDR controllers "local addresses" are referred to, as the addresses seen by the DDR controllers. In the single channel (single controller - x32 or x64) case, these addresses are identical to the physical addresses used in the system (0-4GB range), and are accessible via MMDC0.

However, for other, memory maps, such as 2x32 fixed and 2x32 Interleaved options, each MMDC and associated TZASC may be seeing different addresses. For SoC specific options, refer to DDR mapping to MMDC controller ports in the “Memory Maps” chapter.

For the 2x32 Fixed map (LPDDR2 only) , each MMDC (and TZASC) sees a 0-2GB address ranges.

For the "4KB address interleaved" scheme (i.e. LPDDR2 Dual channel 2x 32-bit, Interleaved mapping), addresses are split between MMDC (and TZASC) such that odd 4K Bytes regions are mapped to MMDC0, while even ones are mapped to MMDC1. Both regions are allocated in the 0-2GB space.

The memory regions configured in TZASC controllers ("base_address_low", "base_address_high" fields in "Region Setup Low/High" registers) must take into consideration the address translation done in 4KB interleaving scheme. In practice, the "local addresses" are composed of the original address, with address bit-12 (bit index 13) omitted from the address value, while higher bits "[31:13]" are shifter to the right one place, in place of bit-12.

Memory "aliasing" implications on TZASC settings - in systems which does not utilize the maximal supported DDR space the controller is designed for, the whole DDR memory map becomes "aliased" (replicated) by the size of the physical memory used. In such cases, the TZASC must be configured to protect all aliased regions as well (i.e. effectively reducing the number of available TZASC regions, since all aliased regions must be handled, for each "real" space needing protection).

For complete details on TZASC functionality and the programming model, see the Arm document, “CoreLink™ TrustZone Address Space Controller TZC-380 Technical Reference Manual, (Rev r0p1 or newer)”, available at **http://infocenter.arm.com**.

Chapter 64
Universal Asynchronous Receiver/Transmitter (UART)

64.1 Overview

Universal Asynchronous Receiver/Transmitter (UART) provides serial communication capability with external devices through a level converter and an RS-232 cable or through use of external circuitry that converts infrared signals to electrical signals (for reception) or transforms electrical signals to signals that drive an infrared LED (for transmission) to provide low speed IrDA compatibility.

UART supports NRZ encoding format , RS485 compatible 9 bit data format and IrDA-compatible infrared slow data rate (SIR) format.

The following figure is the UART block diagram.

The "Module Clock" is the UART_CLK which comes from CCM. The "Peripheral Clock" is the IPG_CLK which comes from CCM.

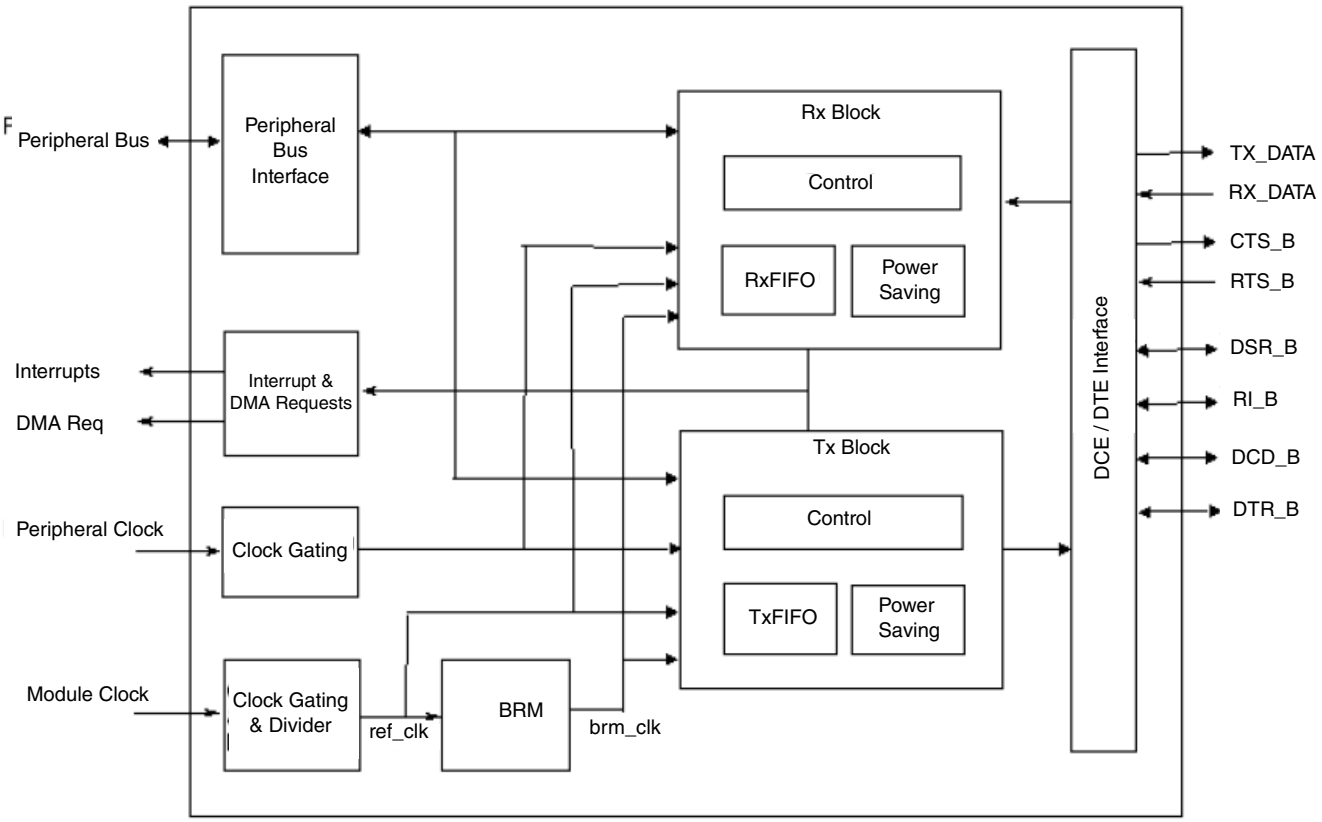


Figure 64-1. UART Block Diagram

64.1.1 Features

The UART includes the following features:

- High-speed TIA/EIA-232-F compatible, up to 5.0 Mbit/s
- Serial IR interface low-speed, IrDA-compatible (up to 115.2 Kbit/s)
- 9-bit or Multidrop mode (RS-485) support (automatic slave address detection)
- 7 or 8 data bits for RS-232 characters, or 9 bit RS-485 format
- 1 or 2 stop bits
- Programmable parity (even, odd, and no parity)
- Hardware flow control support for request to send (RTS_B) and clear to send (CTS_B) signals
- RS-485 driver direction control via CTS_B signal
- Edge-selectable RTS_B and edge-detect interrupts
- Status flags for various flow control and FIFO states
- Voting logic for improved noise immunity (16x oversampling)
- Transmitter FIFO empty interrupt suppression
- UART internal clocks enable/disable

- Auto baud rate detection (up to 115.2 Kbit/s)
- Receiver and transmitter enable/disable for power saving
- RX_DATA input and TX_DATA output can be inverted respectively in RS-232/ RS-485 mode
- DCE/DTE capability
- RTS_B, IrDA asynchronous wake (AIRINT), receive asynchronous wake (AWAKE), RI_B (DTE only), DCD_B (DTE only), DTR_B (DCE only) and DSR_B (DTE only) interrupts wake the processor from STOP mode
- Maskable interrupts
- Two DMA Requests (TxFIFO DMA Request and RxFIFO DMA Request)
- Escape character sequence detection
- Software reset (SRST_B)
- Two independent, 32-entry FIFOs for transmit and receive
- The peripheral clock can be totally asynchronous with the module clock. The module clock determines baud rate. This allows frequency scaling on peripheral clock (such as during DVFS mode) while remaining the module clock frequency and baud rate.

64.1.2 Modes of operation

- Serial RS-232NRZ mode
- 9-bit RS-485 mode
- IrDA mode

To set UART in different modes, see the table below.

Table 64-1. UART mode definition

MDEN (UMCR[0])	IREN (UCR1[7])	UART Mode	Descriptioin
0	0	RS-232	RXD/TXD data is serial RS-232 NRZ format
0	1	IrDA (Interface)	RXD/TXD data is IrDA-compatible infrared slow data rate (SIR) format
1	0	RS-485	RXD/TXD data is RS485 compatible 9 bit data format
1	1	Undefined	Undefined

64.2 External Signals

Tables that lists the conventions for representing signals and describing all UART signals that connect off-chip can be found here.

The chip-level IOMUX modifies the direction and routing of the UART signals based on whether the UART is operating in DCE mode (UARTn_UFCR[DCEDTE]=0) or DTE mode (UARTn_UFCR[DCEDTE]=1). The routing of the external signals to the UART module is shown in the figure below.

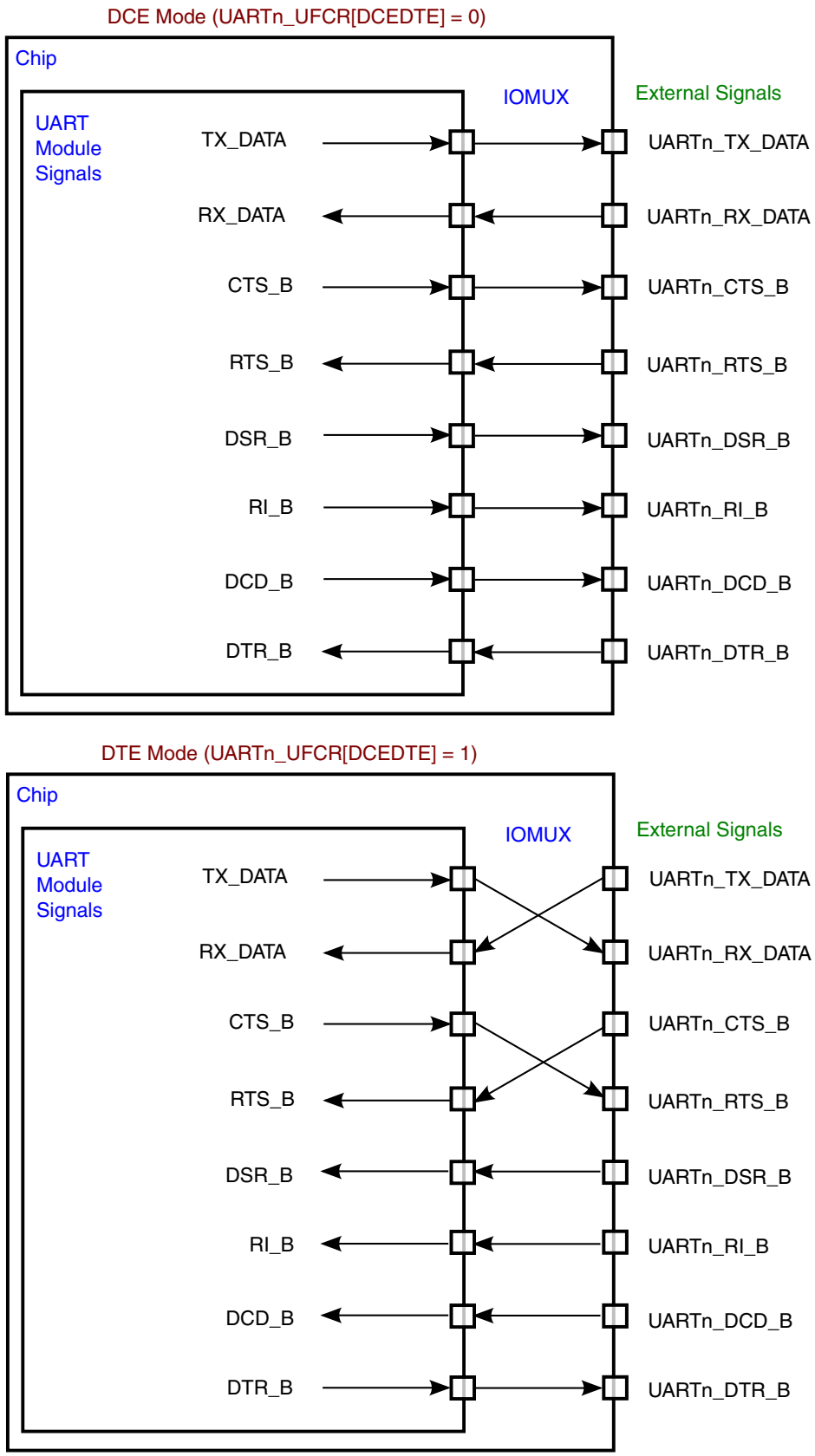


Figure 64-2. UART external signals to module signals routing with respect to DCE/DTE mode

The following tables describes the external signals of UART:

Table 64-2. UART1 External Signals

Signal	Description	Pad	Mode	Direction (DCE Mode)	Direction (DTE Mode)
UART1_CTS_B	Clear to send	EIM_D19	ALT4	O	I
		SD3_DAT0	ALT1		
UART1_DCD_B	Data carrier detected	EIM_D23	ALT3	O	I
UART1_DSR_B	Data set ready	EIM_D25	ALT7	O	I
UART1_DTR_B	Data terminal ready	EIM_D24	ALT7	I	O
UART1_RI_B	Ring indicator	EIM_EB3	ALT3	O	I
UART1_RTS_B	Request to send	EIM_D20	ALT4	I	O
		SD3_DAT1	ALT1		
UART1_RX_DATA	Serial / infrared data receive	CSI0_DAT11	ALT3	I	O
		SD3_DAT6	ALT1		
UART1_TX_DATA	Serial/infrared data transmit	CSI0_DAT10	ALT3	O	I
		SD3_DAT7	ALT1		

Table 64-3. UART2 External Signals

Signal	Description	Pad	Mode	Direction (DCE Mode)	Direction (DTE Mode)
UART2_CTS_B	Clear to send	EIM_D28	ALT4	O	I
		SD3_CMD	ALT1		
		SD4_DAT6	ALT2		
UART2_RTS_B	Request to send	EIM_D29	ALT4	I	O
		SD3_CLK	ALT1		
		SD4_DAT5	ALT2		
UART2_RX_DATA	Serial/infrared data receive	EIM_D27	ALT4	I	O
		GPIO_8	ALT4		
		SD3_DAT4	ALT1		
		SD4_DAT4	ALT2		
UART2_TX_DATA	Serial/infrared data transmit	EIM_D26	ALT4	O	I
		GPIO_7	ALT4		
		SD3_DAT5	ALT1		
		SD4_DAT7	ALT2		

Table 64-4. UART3 External Signals

Signal	Description	Pad	Mode	Direction (DCE Mode)	Direction (DTE Mode)
UART3_CTS_B	Clear to send	EIM_D23	ALT2	O	I
		EIM_D30	ALT4		
		SD3_DAT3	ALT1		
UART3_RTS_B	Request to send	EIM_D31	ALT4	I	O
		EIM_EB3	ALT2		
		SD3_RST	ALT1		
UART3_RX_DATA	Serial / infrared data receive	EIM_D25	ALT2	I	O
		SD4_CLK	ALT2		
UART3_TX_DATA	Serial/infrared data transmit	EIM_D24	ALT2	O	I
		SD4_CMD	ALT2		

Table 64-5. UART4 External Signals

Signal	Description	Pad	Mode	Direction (DCE Mode)	Direction (DTE Mode)
UART4_CTS_B	Clear to send	CSI0_DAT17	ALT3	O	I
UART4_RTS_B	Request to send	CSI0_DAT16	ALT3	I	O
UART4_RX_DATA	Serial / infrared data receive	CSI0_DAT13	ALT3	I	O
		KEY_ROW0	ALT4		
UART4_TX_DATA	Serial/infrared data transmit	CSI0_DAT12	ALT3	O	I
		KEY_COL0	ALT4		

Table 64-6. UART5 External Signals

Signal	Description	Pad	Mode	Direction (DCE Mode)	Direction (DTE Mode)
UART5_CTS_B	Clear to send	CSI0_DAT19	ALT3	O	I
		KEY_ROW4	ALT4		
UART5_RTS_B	Request to send	CSI0_DAT18	ALT3	I	O
		KEY_COL4	ALT4		
UART5_RX_DATA	Serial / infrared data receive	CSI0_DAT15	ALT3	I	O
		KEY_ROW1	ALT4		
UART5_TX_DATA	Serial / infrared data transmit	CSI0_DAT14	ALT3	O	I
		KEY_COL1	ALT4		

The user must configure the input path to the UART by properly configuring the DAISY bits in the IOMUXC_UARTn_RX_DATA_INPUT and the IOMUXC_UARTn_UART_RTS_B_SELECT_INPUT registers.

For IOMUXC_UARTn_UART_RTS_B_SELECT_INPUT[DAISY]:
- Configurations that select UARTn_RTS_B for the pad are only valid when UARTn_UFCR[DCEDTE]=0 (DCE mode)
- Configurations that select UARTn_CTS_B for the pad are only valid when UARTn_UFCR[DCEDTE]=1 (DTE mode)

For IOMUXC_UARTn_UART_RX_DATA_B_SELECT_INPUT[DAISY]:
- Configurations that select UARTn_RX_DATA for the pad are only valid when UARTn_UFCR[DCEDTE]=0 (DCE mode)
- Configurations that select UARTn_TX_DATA for the pad are only valid when UARTn_UFCR[DCEDTE]=1 (DTE mode)

64.2.1 Detailed Signal Descriptions

64.2.1.1 Interrupt Signals

64.2.1.1.1 *interrupt_uart* - UART Interrupt

Output interrupt request.

64.2.1.2 DMA Request Signals

64.2.1.2.1 *dma_req_rx* - Receiver DMA Request

Output DMA Request signal for receiver interface.

64.2.1.2.2 *dma_req_tx* - Transmitter DMA Request

Output DMA Request signal for transmitter interface. Set at 0 when TXDMAEN (UCR1[3]) is at 1 and TRDY (USR1[13]) is also at 1.

64.2.1.3 Special Signals

64.2.1.3.1 *stop_req* - Stop Mode

Input stop mode. Indicates to UART that Arm platform is going to enter in Stop Mode and clocks are going to stop running.

See Low Power Modes for more information about Stop Mode.

64.2.1.3.2 *doze_req* - Doze Mode

Input doze mode. Arm platform requests UART to switch in doze mode (power saving mode).

See Low Power Modes for more information about Doze Mode.

64.2.1.3.3 *debug_req* - Debug Mode

Input debug mode. Indicates UART it has to enter in debug mode.

See UART Operation in System Debug State, for more information about Debug Mode.

64.3 Clocks

The table found here describes the clock sources for UART.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 64-7. UART Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_s	ipg_clk_root	Peripheral access clock
ipg_perclk	uart_clk_root	Module clock

64.4 Functional Description

This section provides a complete functional description of the block.

64.4.1 Interrupts and DMA Requests

See the following table for the lists of all interrupt and DMA signals and associated interrupt and DMA sources of the UART. See register description section for explanation of interrupt/DMA enable and status.

Table 64-8. Interrupts and DMA

Interrupt/DMA Output	Interrupt/DMA Enable	Enable Register Location	Interrupt/DMA Flag	Flag Register Location
interrupt_uart	RRDYEN	UCR1 (bit 9)	RRDY	USR1 (bit 9)
	IDEN	UCR1 (bit 12)	IDLE	USR2 (bit 12)
	DREN	UCR4 (bit 0)	RDR	USR2 (bit 0)
	RXDSEN	UCR3 (bit 6)	RXDS	USR1 (bit 6)
	ATEN	UCR2 (bit 3)	AGTIM	USR1 (bit 8)
interrupt_uart	TXMPTYEN	UCR1 (bit 6)	TXFE	USR2 (bit 14)
	TRDYEN	UCR1 (bit 13)	TRDY	USR1 (bit 13)
	TCEN	UCR4 (bit 3)	TXDC	USR2 (bit 3)
interrupt_uart	OREN	UCR4 (bit 1)	ORE	USR2 (bit 1)
	BKEN	UCR4 (bit 2)	BRCD	USR2 (bit 2)
	WKEN	UCR4 (bit 7)	WAKE	USR2 (bit 7)
	ADEN	UCR1 (bit 15)	ADET	USR2 (bit 15)
	ACIEN	UCR3 (bit 0)	ACST	USR2 (bit 11)
	ESCI	UCR2 (bit 15)	ESCF	USR1 (bit 11)
	ENIRI	UCR4 (bit 8)	IRINT	USR2 (bit 8)
	AIRINTEN	UCR3 (bit 5)	AIRINT	USR1 (bit 5)
	AWAKEN	UCR3 (bit 4)	AWAKE	USR1 (bit 4)
	FRAERREN	UCR3 (bit 11)	FRAERR	USR1 (bit 10)
	PARERREN	UCR3 (bit 12)	PARITYERR	USR1 (bit 15)
	RTSDEN	UCR1 (bit 5)	RTSD	USR1 (bit 12)
	RTSEN	UCR2 (bit 4)	RTSF	USR2 (bit 4)
	DTREN (DCE)	UCR3 (bit 13)	DTRF	USR2 (bit 13)
	RI (DTE)	UCR3 (bit 8)	RIDELT	USR2 (bit 10)
	DCD (DTE)	UCR3 (bit 9)	DCDDELT	USR2 (bit 6)
	DTRDEN	UCR3 (bit 3)	DTRD	USR1 (bit 7)
	SADEN	UMCR (bit 3)	SAD	USR1 (bit 3)
dma_req_rx	RXDMAEN	UCR1 (bit 8)	RRDY	USR1 (bit 9)
	ATDMAEN	UCR1 (bit 2)	AGTIM	USR1 (bit 8)
	IDDMAEN	UCR4 (bit 6)	IDLE	USR2 (bit 12)
dma_req_tx	TXDMAEN	UCR1 (bit 3)	TRDY	USR1 (bit 13)

64.4.2 Clocks

This section describes clocks and special clocking requirements of the UART.

64.4.2.1 Clock requirements

UART module receives 2 clocks, *peripheral_clock* and *module_clock*. The *peripheral_clock* is used as write clock of the TxFIFO, read clock of the RxFIFO and synchronization of the modem control input pins. It must always be running when UART is enabled. There is an exception in stop mode (see Clocking in Low-Power Modes).

The *module_clock* is for all the state machines, writing RxFIFO, reading TxFIFO, etc. It must always be running when UART is sending or receiving characters.This clock is used in order to allow frequency scaling on *peripheral_clock* without changing configuration of baud rate (*module_clock* staying at a fixed frequency).

The constraints on *peripheral_clock* and *module_clock* are as follows:

- *peripheral_clock* and *module_clock* can totally be asynchronous. They can also be synchronous.
- Due to the 16x oversampling of the incoming characters, *module_clock* frequency must always be greater or equal to 16x the maximum baud rate. For example, if max baud rate is 4 Mbit/s, *module_clock* must be greater or equal to 4 M x 16 = 64 MHz.

NOTE

The restriction that *peripheral_clock* frequency must be higher or equal to 16x baud rate has been removed. There is no limitation on *peripheral_clock* frequency to baud rate.

64.4.2.2 Maximum Baud Rate

The max baud rate the UART can support is determined by the max frequency of the module_clock.

For example, if the SoC can provide the fastest module_clock 66.5 MHz, the UART can transmit and receive serial data with the maximum baud rate 66.5M/16 = 4.15 Mbit/s.

The UART supports serial IR interface low speed. In the low speed IrDA mode, the max baud rate is 115.2 Kbit/s. To support the 115.2 Kbit/s, module_clock frequency must be higher or equal to 1.8432 MHz.

64.4.2.3 Clocking in Low-Power Modes

The UART supports 2 low-power modes: DOZE and STOP.

In STOP mode (input pin *stop_req* is at '1'), the UART doesn't need any clock. In this mode the UART can wake-up the Arm platform with the asynchronous interrupts (see Low Power Modes).

- If before entering in STOP mode the software has enabled RTSDEN interrupt, when RTS will change state (put at '0' by external device started to send), the asynchronous interrupt will wake-up the system, *peripheral_clock* and *module_clock* will be provided to the UART before first start bit, so that no data will be lost.
- If RTS doesn't change state (already at '0' before entering in STOP mode), then wake-up interrupt (AWAKE) will be sent at the arrival of first Start bit (on falling edge). In this case, the UART must receive the *peripheral_clock* and *module_clock* during the first half of start bit to correctly receive this character (for example, at 115.2 Kbit/s, UART must receive *peripheral_clock* and *module_clock* at maximum 4.3 microseconds after falling edge of Start bit). If the UART receives *peripheral_clock* and *module_clock* too late, first character will be lost, and so should be dropped. Also, if autobaud detection is enabled, the first character won't be correctly received and another autobaud detection will need to be initiated.

In Doze mode, UART behavior is programmable through DOZE bit (UCR1[1]). If DOZE bit is set to '1', then UART is disabled in Doze mode, and in consequence, UART clocks can be switched-off (after being sure UART is not transmitting nor receiving). On the contrary, if DOZE bit is set to '0', UART is enabled and it must receive *peripheral_clock* and *module_clock* .

64.4.3 General UART Definitions

Definitions of terms that occurs the following discussions are given in this section.

- Bit Time-The period of time required to serially transmit or receive 1 bit of data (1 cycle of the baud rate frequency).
- Start bit-The bit time of a logic 0 that indicates the beginning of a data frame. A start bit begins with a 1-to-0 transition, and is preceded by at least 1 bit time of logic 1.
- Stop bit-1 bit time of logic 1 that indicates the end of a data frame.
- BREAK-A frame in which all of the data bits, including the stop bit, are logic 0. This type of frame is usually sent to signal the end of a message or the beginning of a new message.

- Mark - When no data is being sent, the serial port's transmit pin's voltage is 1 and is said to be in a MARK state.
- Space - The serial port can also be forced to keep the transmit pin at a 0 and is said to be the SPACE or BREAK state.
- Frame-A start bit followed by a specified number of data or information bits and terminated by a stop bit. The number of data or information bits depends on the format specified and must be the same for the transmitting device and the receiving device. The most common frame format is 1 start bit followed by 8 data bits (least significant bit first) and terminated by 1 stop bit. An additional stop bit and a parity bit also can be included.
- Framing Error-An error condition that occurs when the stop bit of a received frame is missing, usually when the frame boundaries in the received bit stream are not synchronized with the receiver bit counter. Framing errors can go undetected if a data bit in the expected stop bit time happens to be a logic 1. A framing error is always present on the receiver side when the transmitter is sending BREAKs. However, when the UART is programmed to expect 2 stop bits and only the first stop bit is received, this is not a framing error by definition.
- Parity Error-An error condition that occurs when the calculated parity of the received data bits in a frame does not match the parity bit received on the RX_DATA input. Parity error is calculated only after an entire frame is received.
- Idle-One in NRZ encoding format and selectable polarity in IrDA mode.
- Overrun Error-An error condition that occurs when the latest character received is ignored to prevent overwriting a character already present in the UART receive buffer (RxFIFO). An overrun error indicates that the software reading the buffer (RxFIFO) is not keeping up with the actual reception of characters on the RX_DATA input.

64.4.3.1 RTS_B - UART Request To Send

The UART Request To Send input controls the transmitter. The modem or other terminal equipment signals the UART when it is ready to receive by setting '0' on the RTS_B pin.

Normally, the transmitter waits until this signal is active (low) before transmitting a character, however when the Ignore RTS (IRTS) bit is set, the transmitter sends a character as soon as it is ready to transmit. An interrupt (RTSD) can be posted on any transition of this pin and can wake the Arm platform from STOP mode on its assertion. When RTS_B is set to '1' during a transmission, the UART transmitter finishes transmitting the current character and shuts off. The contents of the TxFIFO (characters to be transmitted) remain undisturbed. The operation of this input is the same regardless of whether the UART is in DTE or DCE mode.

64.4.3.2 RTS Edge Triggered Interrupt

The input to the RTS_B pin can be programmed to generate an interrupt on a selectable edge.

See the table below for summary of the operation of the RTS edge triggered interrupt (RTSF).

To enable the RTS_B pin to generate an interrupt, set the request to send interrupt enable (RTSEN) bit (UCR2[4]) to 1. Writing 1 to the RTS_B edge triggered interrupt flag (RTSF) bit (USR2[4]) clears the interrupt flag. The interrupt can occur on the rising edge, falling edge, or either edge of the RTS_B input. The request to send edge control (RTEC) field (UCR2[10:9]) programs the edge that generates the interrupt. When RTEC is set to 0x00 and RTSEN = 1, the interrupt occurs on the rising edge (default). When RTEC is set to 0x01 and RTSEN = 1, the interrupt occurs on the falling edge. When RTEC is set to 0x1X and RTSEN = 1, the interrupt occurs on either edge. This is a synchronous interrupt. The RTSF bit is cleared by writing 1 to it. Writing 0 to RTSF has no effect.

Table 64-9. RTS_B Edge Triggered Interrupt Truth Table

RTS_B	RTSEN	RTEC [1]	RTEC [0]	RTSF	Interrupt Occurs On…	interrupt_uart
X	0	X	X	0	Interrupt disabled	1
1->0	1	0	0	0	Rising edge	1
0->1	1	0	0	1	Rising edge	0
1->0	1	0	1	1	Falling edge	0
0->1	1	0	1	0	Falling edge	1
1->0	1	1	X	1	Either edge	0
0->1	1	1	X	1	Either edge	0

There is another RTS_B interrupt that is not programmable. The status bit RTSD asserts the *interrupt_uart* interrupt when the RTS_B delta interrupt enable = 1. This is an asynchronous interrupt. The RTSD bit is cleared by writing 1 to it. Writing 0 to the RTSD bit has no effect.

64.4.3.3 DTR_B - Data Terminal Ready

This signal indicates the general readiness of the Data Terminal Equipment (DTE). This signal is an input in DCE mode and an output in DTE mode. If the connection between the DCE and the DTE is established once, the DTR_B signal must remain active throughout the whole connection time.

In general the DTR_B and DSR_B signals are responsible for establishing the connection. RTS_B and CTS_B are responsible for the data transfer and the transfer direction in the case of a half-duplex configuration. The DTR_B signal is like a "main switch". If the DTR_B signal is inactive the RTS_B and CTS_B signals have no effect. In DCE mode, an interrupt (DTRD) can be posted on any transition of this pin and can wake the ARM platform from STOP mode on its assertion.

64.4.3.4 DSR_B - Data Set Ready

This signal indicates the general readiness of the DCE. This signal is an output in DCE mode and an input in DTE mode. The DCE uses this signal to inform the DTE that it is switched on, has completed all preparations and can communicate with the DTE.

In DTE mode, an interrupt (DTRD) can be posted on any transition of this pin and can wake the ARM platform from STOP mode on its assertion.

64.4.3.5 DTR_B/DSR_B Edge Triggered Interrupt

The DTR_B input pin (DCE mode) or DSR_B input pin (DTE mode) can be configured to cause an interrupt on a selectable edge.

See the table below for summary of the operation of the DTR/DSR edge triggered interrupt. To enable the interrupt, set the DTREN bit (UCR3[13]) to '1'. Write a "one" to the DTRF bit (USR2[13]) to clear the interrupt flag.

The interrupt can be configured to occur on either the rising, falling, or either edge of the DTR_B/DSR_B input. Write to the DPEC[1:0] bits (UCR3[15:14]) to program which edge will cause an interrupt. If the bits are set to 00b and DTREN = 1, the interrupt will occur on the rising edge (default). If the bits are set to 01b and DTREN = 1, the interrupt will occur on the falling edge. If the bits are set to 1Xb and DTREN = 1, the interrupt will occur on either edge.

Table 64-10. DTR/DSR_B Edge Triggered Interrupt Truth Table

DTR_B / DSR_B	DTREN	DPEC[1]	DPEC[0]	DTRF	Interrupt occurs on:	interrupt_uart
X	0	X	X	0	turned off	1
1->0	1	0	0	0	rising edge	1
0->1	1	0	0	1	rising edge	0
1->0	1	0	1	1	falling edge	0
0->1	1	0	1	0	falling edge	1

Table continues on the next page...

Table 64-10. DTR/DSR_B Edge Triggered Interrupt Truth Table (continued)

DTR_B / DSR_B	DTREN	DPEC[1]	DPEC[0]	DTRF	Interrupt occurs on:	interrupt_uart
1->0	1	1	X	1	either edge	0
0->1	1	1	X	1	either edge	0

64.4.3.6 DCD_B - Data Carrier Detect

This signal is an output in DCE mode and an input in DTE mode. If used, the DCE device uses this signal to inform the DTE it has detected the carrier signal and the connection will be set up. This signal remains active while the connection remains established.

In DTE mode this input can trigger an interrupt on changing state. This is achieved by setting to '1' the interrupt enable bit (DCD, UCR3[9]). The change state is reflected in DCDDELT (USR2[6]). Also, the state of the Data Carrier Detect input is mirrored in the status register DCDIN (USR2[5]).

64.4.3.7 RI_B - Ring Indicator

This signal is an output in DCE mode and an input in DTE mode. If used, the DCE device uses this signal to inform the DTE that a ring just occurred.

In DTE mode this input can trigger an interrupt on changing state. This is achieved by setting to '1' the interrupt enable bit (RI, UCR3[8]). The change state is reflected in RIDELT (USR2[10]). Also, the state of the Ring Indicator input is mirrored in the status register RIIN (USR2[9]).

64.4.3.8 CTS_B - Clear To Send

This output pin serves two purposes. Normally, the receiver indicates that it is ready to receive data by asserting this pin (low). When the CTS_B trigger level is programmed to trigger at 32 characters received and the receiver detects the valid start bit of the 33 character, it de-asserts this pin. The operation of this output is the same regardless of whether the UART is in DTE or DCE mode.

64.4.3.9 Programmable CTS_B Deassertion

The CTS_B output can also be programmed to deassert when the RxFIFO reaches a certain level. Setting the CTS trigger level (UCR4[15:10]) at any value less than 32 deasserts the CTS_B pin on detection of the valid start bit of the N + 1 character (where N is the trigger level setting). However, the receiver continues to receive characters until the RxFIFO is full.

64.4.3.10 TX_DATA - UART Transmit

This is the transmitter serial output. When operating in RS-232/RS-485 mode, NRZ encoded data is transmitted, and the data can be inverted (controlled by INVT (UCR3[1])) before transmitted. When operating in infrared mode, a 3/16 bit-period pulse is output for each 0 bit transmitted, and no pulse is output for each 1 bit transmitted.

For RS-232/RS-485 applications, this pin must be connected to an RS-232/RS-485 transmitter. The operation of this output is the same regardless of whether the UART is in DTE or DCE mode. See Figure 64-3.

64.4.3.11 RX_DATA - UART Receive

This is the receiver serial input. When operating in RS-232/RS-485 mode, NRZ encoded data is expected, and the data can be inverted (controlled by INVR (UCR4[9])) before sampled. When operating in infrared mode, a narrow pulse is expected for each 0 bit received and no pulse is expected for each 1 bit received.

External circuitry must convert the IR signal to an electrical signal. RS-232/RS-485 applications require an external RS-232/RS-485 receiver to convert voltage levels. The operation of this input is the same regardless of whether the UART is in DTE or DCE mode. See the figure below.

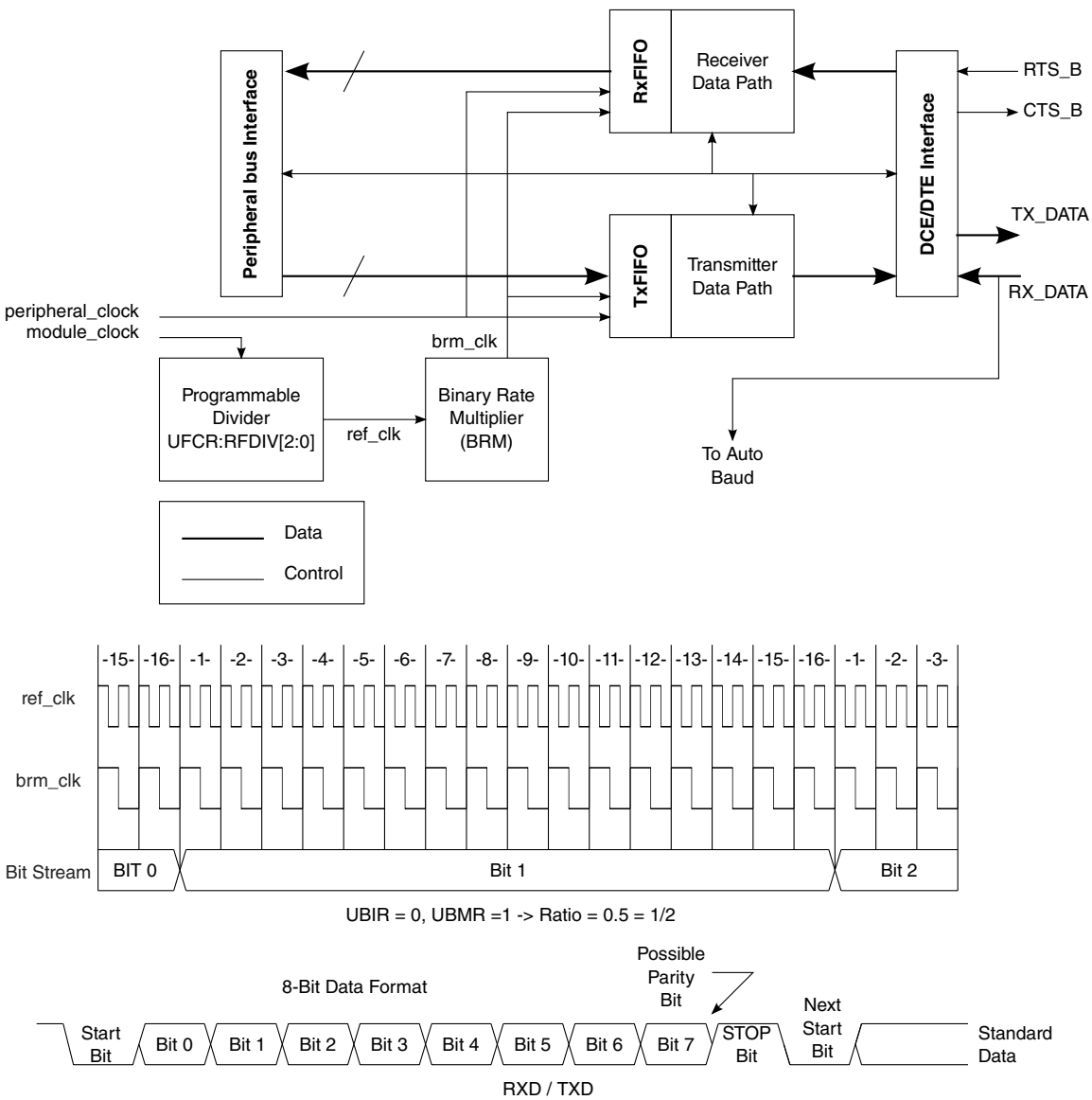


Figure 64-3. UART Simplified Block and Clock Generation Diagrams

64.4.4 Transmitter

The transmitter accepts a parallel character from the Arm platform and transmits it serially. The start, stop, and parity (when enabled) bits are added to the character.

When the ignore RTS bit (IRTS) is set, the transmitter sends a character as soon as it is ready to transmit. RTS_B can be used to provide flow-control of the serial data. When RTS_B is set to '1', the transmitter finishes sending the character in progress (if any), stops, and waits for RTS_B to be set to '0' again. Generation of BREAK characters and parity errors (for debugging purposes) is supported. The transmitter operates from the clock provided by the Binary Rate Multiplier(BRM). Normal NRZ encoded data is transmitted when the IR interface is disabled.

The transmitter FIFO (TxFIFO) contains 32 bytes. The data is written to TxFIFO by writing to the UTXD register with the byte data to the [7:0] bits. The data is written consecutively if the TxFIFO is not full. It is read (internally) consecutively if the TxFIFO is not empty. TXFULL bit (UTS[4]) can be used to control whether TXFIFO is full or not. The TxFIFO can be written regardless of the transmitter is disabled or enabled. If the UART is disabled, user can still write data into the TxFIFO correctly. But in this case the write access will yield to a transfer error.

64.4.4.1 Transmitter FIFO Empty Interrupt Suppression

The transmitter FIFO empty interrupt suppression logic suppresses the TXFE interrupt between writes to the TxFIFO.

When TxFIFO is empty, the software can either send one or several characters. If the software sends one character, it would write the character into the UTXD register, then that character is immediately transferred to the transmitter shift register, assuming the transmitter is already enabled. Without interrupt suppression logic, the TXFE interrupt flag would be set immediately. But, with this logic, the interrupt flag is set when the last bit of the character has been transmitted, for example, before the transmission of the parity bit (if exists) and the stop bit(s).

So, the suppression logic doesn't immediately send the TXFE interrupt flag. It allows the software to write another character to the TxFIFO before the interrupt flag is asserted.

When the transmitter shift register empties before another character is written to the TxFIFO, the interrupt flag is asserted. Writing data to the TxFIFO would release the interrupt flag. The interrupt flag is asserted on the following conditions:

- System Reset

- UART software reset
- When a single character has been written to Transmitter FIFO and then the Transmitter FIFO and the Transmitter Shift Register become empty until another character is written to the Transmitter FIFO
- The last character in the TxFIFO is transferred to the shift register, when TxFIFO contains two or more characters. See the figure below.

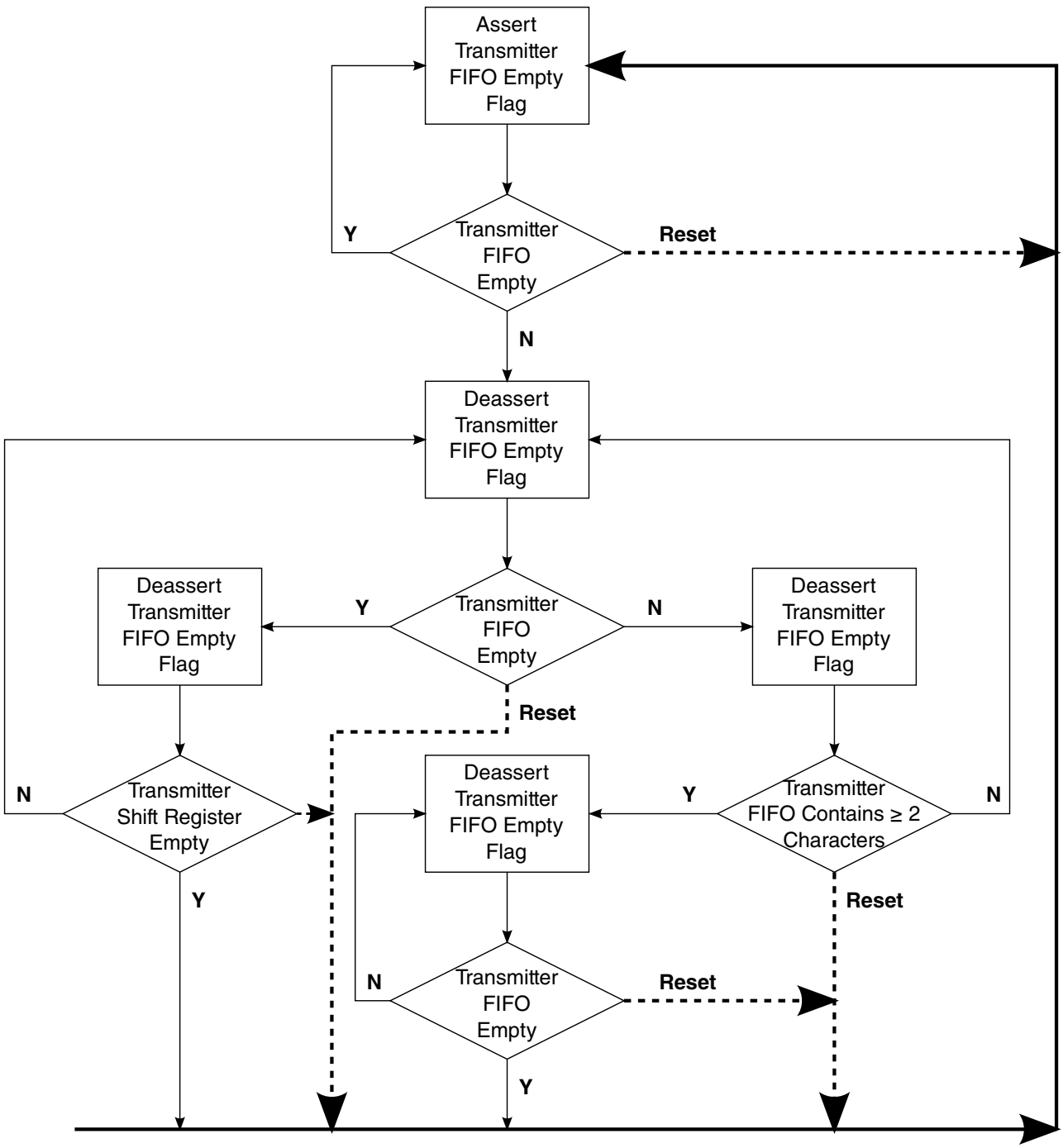


Figure 64-4. Transmitter FIFO Empty Interrupt Suppression Flow Chart

64.4.4.2 Transmitting a Break Condition

Asserting SNDBRK bit of the UCR1 Register forces the transmitter to send a break character (continuous zeros). The transmitter will finish sending the character in progress (if any) before sending break until this bit is reset.

The user is responsible to ensure that this bit is high for long enough to generate a valid BREAK. The transmitter samples SNDBRK after every bit is transmitted. Following completion of the BREAK transmission, the UART will transmit two mark bits. The user can continue to fill the FIFO and any character remaining will be transmitted when the break is terminated.

64.4.5 Receiver

See the figure below for the receiver flow chart.

The receiver accepts a serial data stream and converts it into parallel characters. When enabled, it searches for a start bit, qualifies it, and samples the following data bits at the bit-center.

Jitter tolerance and noise immunity are provided by sampling at a 16x rate and using voting techniques to clean up the samples. Once the start bit is found, the data bits, parity bit (if enabled), and stop bits (either 1 or 2 depending on user selection) are shifted in. Parity is checked and its status reported in the URXD register when parity is enabled. Frame errors and BREAKs are also checked and reported. When a new character is ready to be read by the Arm platform from the RxFIFO, the receive data ready (RDR = USR2[0]) bit is asserted and an interrupt is posted (if DREN = UCR4[0] = 1). If the receiver trigger level is set to 2 (RXTL[5:0] = UFCR[5:0] = 2), and 2 chars have been received into RxFIFO, the receiver ready interrupt flag (RRDY = USR1[9]) is asserted and an interrupt is posted if the receiver ready interrupt enable bit is set (RRDYEN = UCR1[9] = 1). If the UART Receiver Register (URXD) is read once, and in consequence there is only 1 character in the RxFIFO, the interrupt generated by the RDR bit is automatically cleared. The RRDY bit is cleared when the data in the RxFIFO falls below the programmed trigger level.

Normal NRZ encoded data is expected when the IR interface is disabled. The RxFIFO contains 32 half-word entries. Characters received are written consecutively into this FIFO. If the FIFO is full and a 33rd characters is received, this character will be ignored and the USR2[ORE] bit will be set.

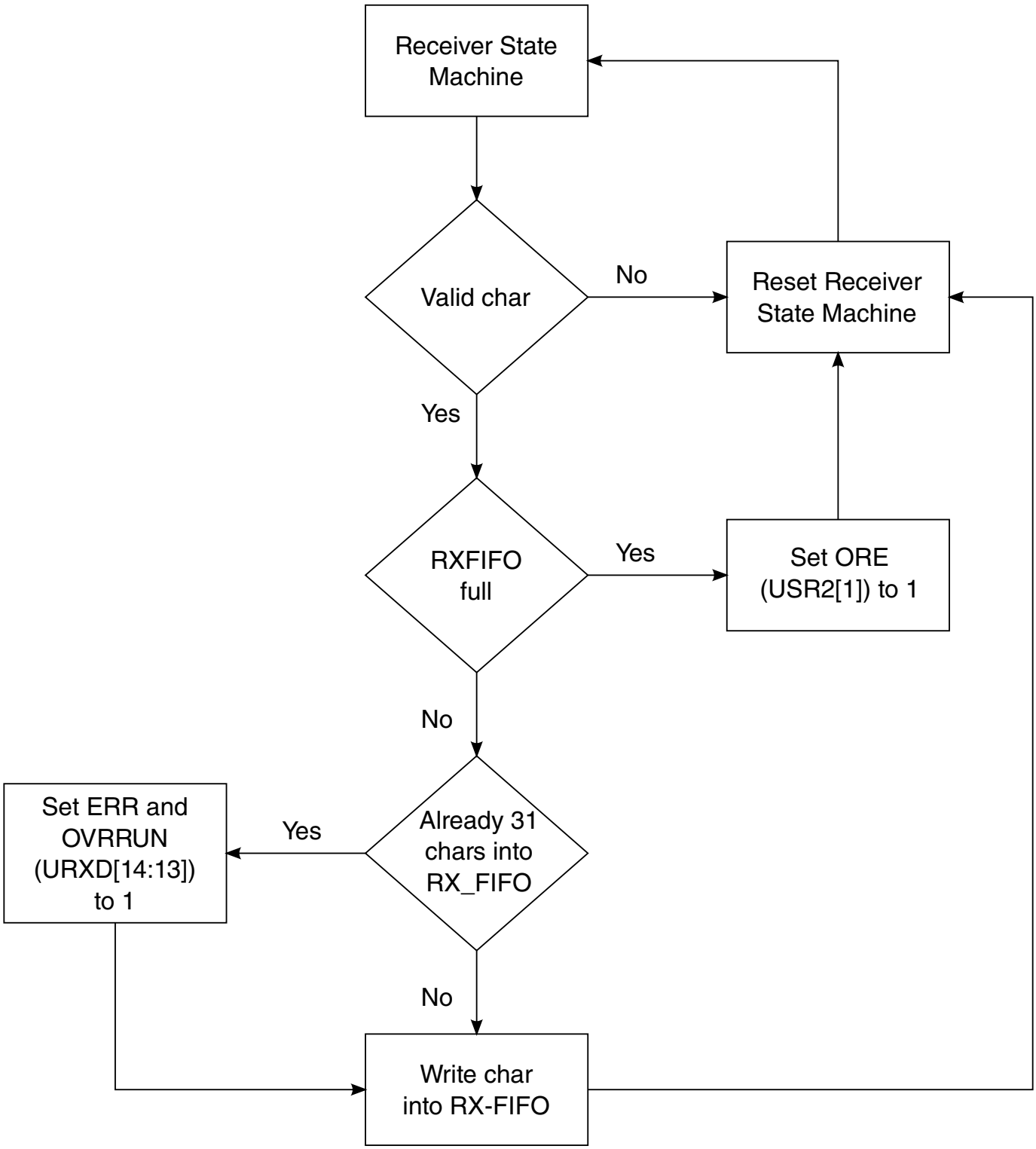


Figure 64-5. Receiver Flow Chart

64.4.5.1 Idle Line Detect

The receiver logic block includes the ability to detect an idle line. Idle lines indicate the end or the beginning of a message.

For an idle condition to occur:

- RxFIFO must be empty and
- RX_DATA pin must be idle for more than a configured number of frames (ICD[1:0] = UCR1[11:10]).

When the idle condition detected interrupt enable (IDEN = UCR1[12]) is set and the line is idle for 4 (default), 8, 16, or 32 (maximum) frames, the detection of an idle condition flags an interrupt (see the table below). When an idle condition is detected, the IDLE (USR2[12]) bit is set. Clear the IDLE bit by writing 1 to it. Writing 0 to the IDLE bit has no effect.

Table 64-11. Detection Truth Table

IDEN	ICD [1]	ICD [0]	IDLE	*interrupt_uart*
0	X	X	0	1
1	0	0	asserted after 4 idle frames	asserted after 4 idle frames
1	0	1	asserted after 8 idle frames	asserted after 8 idle frames
1	1	0	asserted after 16 idle frames	asserted after 16 idle frames
1	1	1	asserted after 32 idle frames	asserted after 32 idle frames
NOTE: This table assumes that no other interrupt is set at the same time this interrupt is set for the *interrupt_uart* signal. This table shows how this interrupt affects the *interrupt_uart* signal.				

During a normal message there is no idle time between frames. When all of the information bits in a frame are logic 1s, the start bit ensures that at least one logic 0 bit time occurs for each frame so that the IDLE bit is not asserted.

64.4.5.2 Aging Character Detect

The receiver block also includes the possibility to detect when at least one character has been sitting into the RxFIFO for a time corresponding to 8 characters. This aging character capability allows the UART to inform the Arm platform that there is less character into the RxFIFO than the Rx trigger and, no new character has been detected on the RXD line.

The aging capability is a timer which starts to count as soon as the RxFIFO is not empty and its trigger level is not reached (RRDY=0). This counter is reset when either a RxFIFO read is performed or another character starts to present on the RXD line. If none of those two events occurs, the bit AGTIM (USR1[8]) is set when the counter has

measured a time corresponding to 8 characters. AGTIM is cleared by writing a 1 to it. AGTIM can flag an interrupt to Arm platform on *interrupt_uart* if ATEN (UCR2[3]) has been set.

To summarize, AGTIM is set when:

- There is at least one character into RxFIFO.
- No read has occurred on RxFIFO and RXD line has stayed high, for a time corresponding to 8 characters.
- The RxFIFO trigger is not reached (RRDY=0)

64.4.5.3 Receiver Wake

The WAKE bit (USR2[7]) is set when the receiver detects a qualified Start bit. For this, two conditions must be fulfilled, firstly a falling edge on RX_DATA line must be detected and secondly the RX_DATA line must stays at low level for more than a half-bit duration.

When the wake interrupt enable WKEN (UCR4[7]) bit is enabled, the receiver flags an interrupt (*interrupt_uart*) if the WAKE status bit is set. The WAKE bit is cleared by writing 1 to it. Writing 0 to the WAKE bit has no effect. The WAKE status bit can be asserted in either serial RS-232 mode or IR mode. The generation of the WAKE interrupt needs the clock *module_clock* .

When the asynchronous wake interrupt (AWAKE) is enabled (AWAKEN = UCR3[4] = 1), and the Arm platform is in STOP mode, and UART clocks have been shut-off, then a falling edge detected on the receive pin (RX_DATA) asserts the AWAKE bit (USR1[4]) and the *interrupt_uart* interrupt to wake the Arm platform from STOP mode. Re-enable UART clocks and clear the AWAKE bit by writing 1 to it. Writing 0 to the AWAKE bit has no effect. When IR interface is enabled (UCR1[7]=1), the AWAKE bit is always not asserted. The generation of the asynchronous AWAKE interrupt does not need any clocks.

In IR mode, if the asynchronous IR WAKE interrupt is enabled (AIRINTEN = UCR3[5] = 1), and if the Arm platform is in STOP mode (UART clocks are off when Arm platform in STOP mode), then the detection of a falling edge on the receive pin (RXD_IR), asserts the AIRINT bit (USR1[5]), and the *interrupt_uart* interrupt. This interrupt wakes the Arm platform from STOP mode. Software re-enables UART clocks and clear the AIRINT bit by writing 1 to it. Writing 0 to the AIRINT bit has no effect. When IR interface is disabled (UCR1[7]=0), the AIRINT bit is always not asserted. The generation of the asynchronous AIRINT interrupt does not need any clocks.

Recommended procedure for programming the asynchronous interrupts is to first clear them by writing 1 to the appropriate bit in the UART Status Register 1 (USR1). Poll or enable the interrupt for the Receiver IDLE Interrupt Flag (RXDS) in the USR1. When asserted, the RXDS bit indicates to the software that the receiver state machine is in the idle state, the next state is idle, and the RX_DATA pin is idle (high). After following this procedure, enable the asynchronous interrupt and enter STOP mode.

64.4.5.4 Receiving a BREAK Condition

A BREAK condition is received when the receiver detects all 0s (including a 0 during the bit time of the stop bit) in a frame. The BREAK condition asserts the BRCD bit (USR2[2]) and writes only the first BREAK character to the RxFIFO. Clear the BRCD bit by writing 1 to it. Writing 0 to the BRCD bit has no effect.

Asserting BRCD would generate an interrupt on *interrupt_uart*. The interrupt generation can be masked using the control bit BKEN (UCR4[2]). Receiving a break condition will also effect the following bits in the receiver register URXD:

URXD(11) = BRK. While high this bit indicates that the current char was detected as a break.

URXD(12) = FRMERR. The frame error bit will always be set when BRK is set.

URXD(10) = PRERR. If odd parity was selected the parity error bit will also be set when BRK is set.

URXD(14) = ERR. The error detect bit indicates that the character present in the rx data field has an error status. This can be asserted by a break.

64.4.5.5 Vote Logic

The vote logic block provides jitter tolerance and noise immunity by sampling with respect to a 16x clock (*brm_clk*) and using voting techniques to clean up the samples. The voting is implemented by sampling the incoming signal constantly on the rising edge of the *brm_clk*.

See Figure 64-6. The receiver is provided with the majority vote value, which is 2 out of the 3 samples. For examples of the majority vote results of the vote logic, see the following table.

Table 64-12. Majority Vote Results

Samples	Vote
000	0
101	1
001	0
111	1

The vote logic captures a sample on every rising edge of *brm_clk*, however the receiver uses 16x oversampling to take its value in the middle of the sample character.

The receiver starts to count when the Start bit is set however it does not capture the contents of the RxFIFO at the time the Start bit is set. The start bit is validated when 0s are received for 7 consecutive 1/16 of bit times following the 1-to-0 transition. Once the counter reaches 0xF, it starts counting on the next bit and captures it in the middle of the sampling frame (see Table 64-12). All data bits are captured in the same manner. Once the stop bit is detected, the receiver shift register (SIPO_OUT) data is parallel shifted to the RxFIFO.

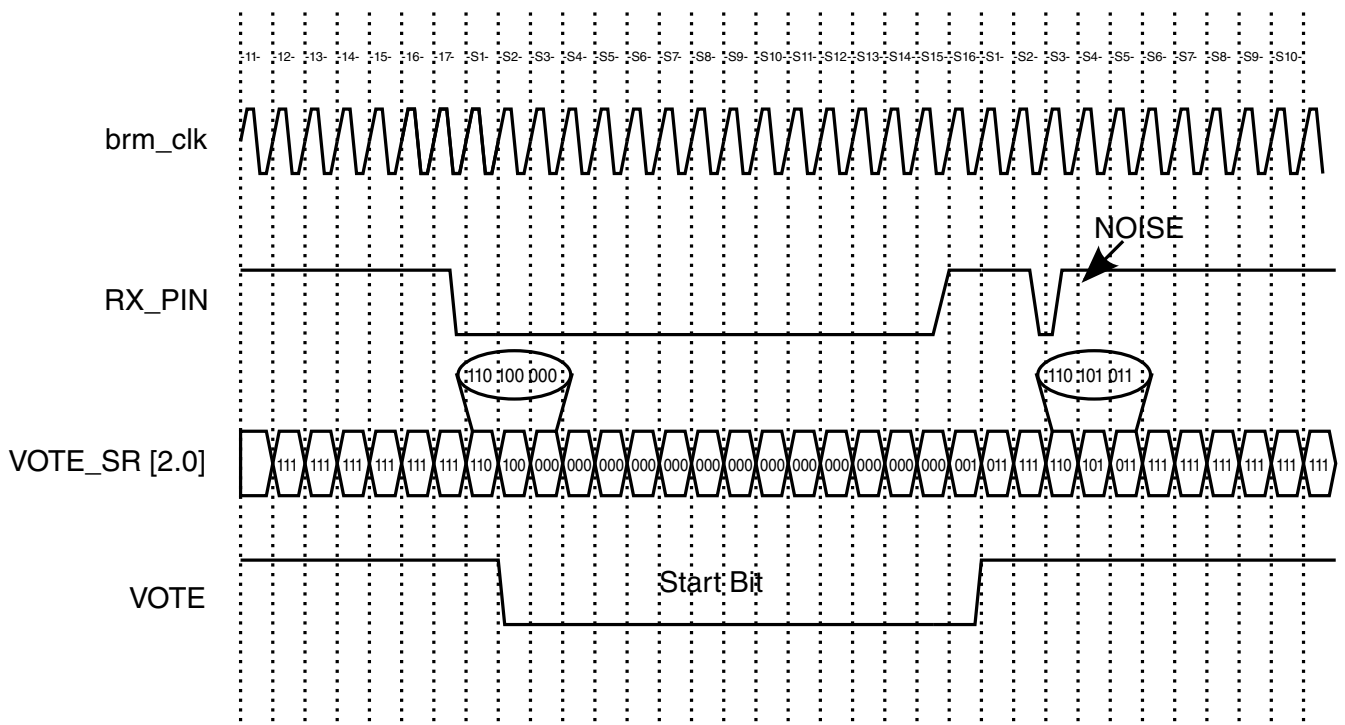


Figure 64-6. Majority Vote Results

A new feature has been recently implemented, it allows to re-synchronize the counter on each edge of RX_DATA line. This is automatic and allows to improve the immunity of UART against signal distortion.

There is a special case when the *brm_clk* frequency is too low and is unable to capture a 0 pulse in IrDA. In this case, the software must set the IRSC (UCR4[5]) bit so that the reference clock (after internal divider) is used for the voting logic. The pulse is validated by counting the length of the pulse.

Refer to Infrared Interface for more details.

64.4.5.6 Baud Rate Automatic Detection Logic

When the baud rate automatic detection logic is enabled, the UART locks onto the incoming baud rate. To enable this feature, set the automatic detection of baud rate bit (ADBR = UCR1[14] = 1) and write 1 to the ADET bit (USR2[15]) to clear it.

When ADET=0 and ADBR =1, the detection starts. Then, once the beginning of start bit (transition from 1-to-0 of RX_DATA) has been detected, UART starts a counter (UBRC) working at reference frequency. Once the end of start bit is detected (transition from 0-to-1 of RX_DATA), the value of UBRC - 1 is directly copied into UBMR register. UBIR register is filled with 0x000F.

So, at the end of start bit, registers gets following values:

```
UBRC = number of reference clock periods (after divider) during Start bit.
UBIR = 0x000F
UBMR = UBRC - 1
```

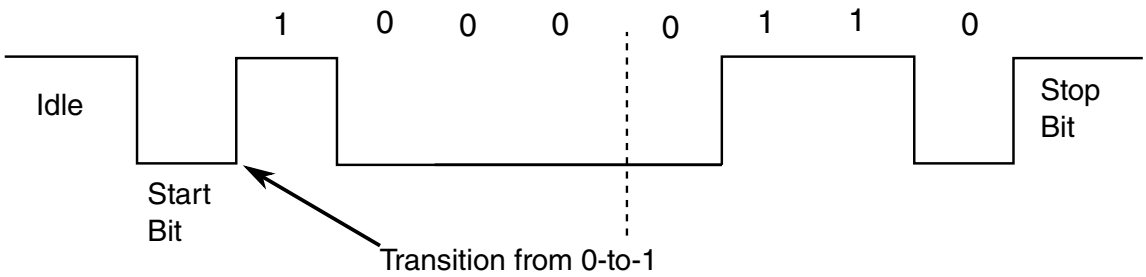
The updated values of the 3 registers can be read.

See Table 64-13 for list of parameters for baud rate detection and Figure 64-7 for baud rate detection protocol diagram.

If any of the UART BRM registers are simultaneously written by the baud rate automatic detection logic and by the peripheral data bus, the peripheral data bus would have lower priority.

Table 64-13. Baud Rate Automatic Detection

ADBR	ADET	Baud Rate Detection	*interrupt_uart*
0	X	Manual Configuration	1
1	0	Auto Detection Started	1
1	1	Auto Detection Complete	0
NOTE: This table assumes that no other interrupt is set at the same time this interrupt is set for the *interrupt_uart* signal.			



Note: LSB Transmitted first.

Figure 64-7. Baud Rate Detection Protocol Diagram

64.4.5.6.1 Baud Rate Automatic Detection Protocol

The receiver must receive an ASCII character "A" or "a" to verify proper detection of the incoming baud rate. When an ASCII character "A" (0x41) or "a" (0x61) is received and no error occurs, the Automatic Detect baud rate bit is set (ADET=1) and if the interrupt is enabled (ADEN=UCR1[15]=1), an interrupt *interrupt_uart* is generated.

When an ASCII character "A" or "a" is not received (because of a bit error or the reception of another character), the auto detection sequence restarts and waits for another 1-to-0 transition.

As long as ADET = 0 and ADBR = 1, the UART continues to try to lock onto the incoming baud rate. Once the ASCII character "A" or "a" is detected and the ADET bit is set, the receiver ignores the ADBR bit and continues normal operation with the calculated baud rate.

The UART interrupt is active (*interrupt_uart* = 0) as long as ADET = 1 and ADBR = 1. This can be disabled by clearing the automatic baud rate detection interrupt enable bit (ADEN = 0). Before starting an automatic baud rate detection sequence, set ADET = 0 and ADBR = 1.

The RxFIFO must contain the ASCII character "A" or "a" following the automatic baud rate detection interrupt.

The 16-bit UART Baud Rate Count Register (UBRC) is reset to 4 and stays at 0xFFFF when an overflow occurs. The UBRC register counts (measures) the duration of start bit. When the start bit is detected and counted, the UART Baud Rate Count Register retains its value until the next automatic baud rate detection sequence is initiated.

The Baud Rate Count Register counts only when auto detection is enabled.

64.4.5.6.2 New Baud Rate Determination

In order to fight against the problems caused by the distortion and the noise on the RX_DATA line, the duration of the baud rate measurement has been extended.

Previously, as described above, this determination was based on the measurement of the START bit duration. Now, this measurement is based on the duration of START bit + bit0. Bit0 is the first bit following the START bit. In fact, the counter which is started at the falling edge of START bit is no longer stopped at next rising edge (end of START bit), but it is stopped at the next falling edge (end of bit0). As the character sent is always a "A" (41h) or a "a" (61h), this second falling edge will always be present and it will indicate the end of bit0. Once this counter is stopped, the result is divided by 2 and used by the BRM to determine the incoming baud rate.

NOTE

UBRC register contains the result of this division by two, in consequence it reflects the measurement of the duration of one bit.

64.4.5.6.2.1 New Autobaud Counter Stopped bit and Interrupt

A new bit has been added in USR2 register: ACST (USR2[11]). This bit is set immediately after the determination of the baud rate.

So,

- if ADNIMP is not set (default), ACST is set to 1 after the end of bit0,
- If ADNIMP is set to 1, ACST is set to 1 at the end of START bit.

If ACIEN (UCR3[0]) is set to 1, ACST will flag an interrupt on *interrupt_uart* signal. This interrupt informs the Arm platform that the BRM has just been set with the result of the bit length measurement. If needed, the Arm platform can perform a read of UBMR (or UBRC) register and determine by itself the baud rate measured. Then the Arm platform has the possibility to correct the BRM registers with the nearest standardized baud rate.

NOTE

ACST is set only if ADBR is set to 1, for example, the UART is autobauding.

Clear the ACST bit by writing 1 to it. Writing 0 to the ACST bit has no effect.

64.4.6 Escape Sequence Detection

An escape sequence typically consists of 3 characters entered in rapid succession (such as +++). Because these are valid characters by themselves, the time between characters determines if it is a valid escape sequence.

Too much time between two of the "+" characters is interpreted as two "+" characters, and not part of an escape sequence.

The software chooses the escape character and writes its value to the UART Escape Character Register (UESC). The software must also enable escape detection feature by setting ESCEN (UCR2[11]) to 1. The hardware compares this value to incoming characters in the RxFIFO. When an escape character is detected, the internal escape timer starts to count. The software specifies a time-out value for the maximum allowable time between 2 successive escape characters (see the table below). The escape timer is programmable in intervals of 2 ms to a maximum interval of 8.192 seconds.

Table 64-14. Escape Timer Scaling

UTIM Register	Maximum Time Between Specified Escape Characters
0x000	2 ms
0x001	4 ms
0x002	6 ms
0x003	8 ms
0x004	10 ms
...	...
0F8	498 ms
0F9	500 ms
...	...
9C3	5 s
...	...
FFD	8.188 s
FFE	8.190 s
FFF	8.192 s
NOTE: To calculate the time interval: (UTIM_Value + 1) x 0.002 = Time_Interval Example: (09C3 + 1) x 0.002 = 5 s.	

The escape sequence detection feature is available for all the reference frequencies. Before using Escape Sequence Detection, the user must fill the ONEMS register. This 24-bit register must contain the value of the UART internal frequency divided by 1000. The internal frequency is obtained after the UART internal divider which is applied on *module_clock* clock.

Example I:

- If the input clock *module_clock* frequency is 66.5 MHz.
- And if the input clock *module_clock* is divided by 2 with the internal divider: UFCR[9:7] = 3'b100

$$ONEMS = \frac{66.5 \times 10^6}{2 \times 1000} = 33250 = 81E2h$$

Figure 64-8. Calculation of Frequency for ONEMS Register

Example II:

- If the input clock *module_clock* frequency is 66.5 MHz.
- And if the input clock *module_clock* is divided by 1with the internal divider: UFCR[9:7] = 3'b101

$$ONEMS = \frac{66.5 \times 10^6}{1000} = 66500 = 103C4h$$

Figure 64-9. Calculation of Frequency for ONEMS Register

The escape sequence detection interrupt is asserted when the escape sequence interrupt enable (ESCI) bit is set and an escape sequence is detected (ESCF set). Clear the ESCF bit by writing 1 to it. Writing 0 to the ESCF bit has no effect.

64.5 Binary Rate Multiplier (BRM)

The BRM sub-block receives *ref_clk* (*module_clock* clock after divider). From this clock, and with integer and non-integer division, BRM generates a 16x baud rate clock .

The UART transmitte will shift data out based on this 16x baud rate clock. The UART receiver will sample the serial data line based on this 16x baud rate clock. The input and output frequency ratio is programmed in the UART BRM Incremental Register (UBIR) and UART BRM MOD Register (UBMR). The output frequency is divided by the input frequency to produce this ratio. For integer division, set the UBIR = 0x000F and write the divisor to the UBMR register. All values written to these registers must be one less than the actual value to eliminate division by 0 (undefined), and to increase the maximum range of the registers.

Updating the BRM registers requires writing to both registers. The UBIR register must be written before writing to the UBMR register. If only one register is written to by the software, the BRM continues to use the previous values.

The following examples show how to determine what values are to be programmed into UBIR and UBMR for a given reference frequency and desired baud rate. The following equation can be used to help determine these values:

$$\text{BaudRate} = \frac{\text{Ref Freq}}{\left(16 \times \frac{\text{UBMR} + 1}{\text{UBIR} + 1} \right)}$$

Figure 64-10. Frequency and Baud Rate for UBIR and UBMR

With:

Reference Frequency (Hz): UART Reference Frequency (*module_clock* after RFDIV divider)

Baud Rate (bit/s): Desired baud rate.

Integer Division ÷ 21

Reference Frequency = 19.44 MHz

UBIR = 0x000F

UBMR = 0x0014

Baud Rate = 925.7 kbit/s

NOTE

Observe that each value written to the registers is one less than the actual value.

Non-Integer Division

```
Reference Frequency = 16 MHz
Desired Baud Rate = 920 Kbits/s
```

$$\frac{UBMR + 1}{UBIR + 1} = \frac{RefFreq}{16 \times BaudRate} = \frac{16 \times 10^6}{16 \times 920 \times 10^3} = 1.087$$

```
Ratio = 1.087 = 1087 / 1000
UBIR = 999 (decimal)= 0x3E7
UBMR = 1086 (decimal)= 0x43E
Non-Integer Division
Reference Frequency = 25 MHz
Desired Baud Rate = 920 kbit/s
Ratio = 1.69837 = 625 / 368
UBIR = 367 (decimal)= 0x16F
UBMR = 624 (decimal)= 0x270
```

Non-Integer Division

```
Reference Frequency: 30 MHz
Desired Baud Rate = 115.2 kbit/s
Ratio = 16.276043 = 65153 / 4003
UBIR = 4002 (decimal) = 0x0FA2
UBMR = 65152 (decimal) = 0xFE80
```

64.6 Infrared Interface

64.6.1 Generalities-Infrared

The Infrared interface is selected when IREN (UCR1[7]) is set to 1.

The Infrared Interface is compatible with IrDA Serial Infrared Physical Layer Specification. In this specification, a "zero" is represented by a positive pulse, and a "one" is represented by no pulse (line remains low).

In the UART:

In TX: For each "zero" to be transmitted, a narrow positive pulse which is 3/16 of a bit time is generated. For each "one" to be transmitted no pulse is generated (output is low). External circuitry has to be provided to drive an Infrared LED.

In RX: When receiving, a narrow negative pulse is expected for each "zero" transmitted while no pulse is expected for each "one" transmitted (input is high).

NOTE

Rx part of IR block expects to receive an inverted signal compared to IrDA specification. Circuitry external to the IC transforms the Infrared signal to an electrical signal.

The IR interface has an edge triggered interrupt (IRINT). This interrupt validates a zero bit being received. This interrupt is enabled by writing a "one" to ENIRI bit.

The behavior of Infrared Interface is determined by 3 bits INVT (UCR3[1]), INVR (UCR4[9]) and IRSC (UCR4[5]).

64.6.2 Inverted Transmission and Reception bits (INVT & INVR)

The values of INVT and INVR depend of the IrDA transceiver connected on the TXD_IR and RXD_IR pins of the UART. If this transceiver is not inverting on both paths Tx and Rx, a Zero is represented by a positive pulse and a One is represented by no pulse (line remains low). In this case, the bit INVT must be set to 0 and the bit INVR must be set to 1 (because Rx IR block expects an inverted signal).

On the contrary user must set INVT=1 and INVR=0 if both paths of the transceiver are inverting, that is, a Zero is represented as a negative pulse and a One is represented by no pulse (line remains high). The transceiver can also be inverting on only one path (Tx or Rx), in this case INVT and INVR must be together equal to 1 or to 0, depending on which path is inverted.

64.6.3 InfraRed Special Case (IRSC) Bit

The value to apply to IRSC bit is based on 2 parameters: the baud rate and the Minimum Pulse Duration (MPD) of the transceiver.

According to IrDA Standard Specification, for SIR (Serial IR) baud rates from 2.4 Kbit/s to 115.2 Kbit/s this nominal pulse duration is equal to 3/16 of a bit duration (at the selected baud rate). But, for all the baud rates a Minimum Pulse Duration is also specified. According to IrDA Standard, a Zero is represented by a light pulse, so the IrDA transceiver can't emit a light pulse shorter than the MPD. For SIR, the MPD is constant and equal to 1.41 µs.

But user must take into account the electrical MPD associated with the transceiver on the receiver path. Typically this value is 2.0 µs, but for some manufacturers MPD can go down to 1.0 µs.

In order to understand the meaning of IRSC bit, one must understand how the RX path works in IrDA mode.

When the UART is in IrDA mode, a Zero is not only detected by the state of the RXD_IR line, but also with the duration of the pulse. This pulse duration can be measured with 2 different clocks. In this case, clock is selected with the IRSC bit.

- If IRSC = 0, the clock used is the BRM clock.
- If IRSC = 1, the clock used is the UART internal clock (UART clock after the divider (RFDIV)).

In normal operation, IRSC=0. This means that at any time, the user must ensure that the frequency of BRM_clock is high enough to measure the pulse. The pulse must last at least 2 BRM clock cycles. If this condition is not fulfilled, IRSC must be set to 1.

Let's examine two examples, for a Minimum Pulse Duration equal to the MPD from the IrDA SIR specification (i.e., 1.41 µs).

1: Calculation of BRM Clock Period (Clock Period < 1.41 µs)

The user wants to receive IrDA data at 115.2 Kbit/s. The UBIR and UBMR registers are set in order to create the BRM_clock with a frequency of 16*baud rate = 16 * 115.2K = 1.843 MHz. But at the same time, in order to correctly detect the pulse, the user must be sure that 2* BRM_clock period is lower than 1.41 µs. Lets check:

BRM_clock period = 1/1843000 = 542 ns

So 2*BRM_clock period = 1.09 µs < 1.41 µs. It is fine.

2: Calculation of BRM Clock Period (Clock Period > 1.41 µs)

This time the user wants to receive at 19.2 Kbit/s. So, the BRM_clock is set to 16*19200 = 307.2 kHz. Let's check if 2* BRM_clock period < 1.41 µs:

1. BRM_clock period =1/307200 = 3.25 µs

   So 2*BRM_clock period = 6.50 µs >> 1.41 µs. It doesn't work.

   So, in this case, the BRM clock can't be used to measure the pulse duration and the user must select the UART internal clock by setting IRSC =1.

NOTE

Like for Escape character detection, when IR Special Case is enabled (IRSC=1), the UART must measure a duration. In order to do that, the user must fill the ONEMS register. Refer to Escape Sequence Detection.

64.6.4 IrDA interrupt

Serial infrared mode (SIR) uses an edge triggered interrupt flag IRINT (USR2[8]). When INVR =0, detection of a falling edge on the RXD pin asserts the IRINT bit. When INVR=1, detection of a rising edge on the RXD pin asserts the IRINT bit. When IRINT and ENIRI bits are both asserted, the *interrupt_uart* interrupt is asserted. Clear the IRINT bit by writing 1 to it. Writing 0 to the IRINT bit has no effect.

64.6.5 Conclusion about IrDA

Before using the UART in IrDA, the baud rate limit must be calculated. This baud rate limit will inform the user if IRSC bit has to be set or not.

Let's determine this limit:

As already described, if IRSC = 0, the following condition must always be fulfilled

$$2 \times BRMClockPeriod < MinPulseDuration$$

Figure 64-11. Calculation of Baud Rate

So,

$$BRMClockFrequency > \frac{2}{MPD}$$

So, knowing BRM_clock frequency = 16 * Baud Rate, we get:

$$BaudRate > \frac{1}{8 \times MinPulseDuration}$$

So, the user needs to set IRSC = 0 when:

- If Minimum Pulse Duration = 2.5 us and Baud Rate > 50 Kbit/s.
- If Minimum Pulse Duration = 2.0 us and Baud Rate > 62.5 Kbit/s.
- If Minimum Pulse Duration = 1.41 us and Baud Rate > 88.6 Kbit/s.

NOTE

For baud rates lower than the limit, IRSC must be set to1.

64.6.6 Programming IrDA Interface

64.6.6.1 High Speed

As an example, the following sequence can be used to program the IrDA interface in order to send and receive characters at 115.2 Kbit/s.

Assumptions:

- Input UART clock = 90 MHz
- Internal clock divider = 3 (divide Input UART clock by 3)
- Baud rate = 115.2 Kbit/s
- IrDA transceiver is not inverting on both channels: for Tx and Rx, a Zero is represented by a positive pulse, and a One is represented by no pulse (line stays low).
- Interrupt: Sent to Arm platform when 1 char is received into the Rx FIFO (RDR)

Registers values and Programming orders:

```
UCR1 = 0x0085
UCR1[7] = IREN = 1: Enable IR interface
UCR1[0] = UARTEN = 1: Enable UART
UTS = 0x0000
UFCR = 0x0981
TXTL[5:0] = 0x02: Default value
RFDIV[2:0] = 0x3: Divide Input UART clock by 3 (resulting internal clock  is 30 MHz)
RXTL[5:0] = 0x01: Default value
UBIR = 0x0202
UBMR = 0x20BE Baud rate = 115.2 kbit/s with internal clock = 30 MHz
UCR2 = 0x4027
UCR2[14] = IRTS = 1: Ignore level of RTS input signal
UCR2[5] = WS = 1: Characters are 8-bit length
UCR2[2] = TXEN = 1: Enable Rx path
UCR2 [1] = RXEN = 1: Enable Tx path
UCR2[0] = SRST_B = 1: No software reset
UCR3 = 0x0000

         UCR4 = 0x8201
CTSTL[5:0] = 0x20: Default value
UCR4[9] = INVR = 1: Inverted Infrared Reception (because IrDA transceiver is not inverting)
UCR4[1] = DREN = 1: To enable RDR interrupt (sent when one char is received)
```

The UART is ready to send a character as soon as there is a write into UTXD register. And an interrupt will be sent to Arm platform when a character is received.

64.6.6.2 Low Speed

This time, we keep the same assumptions but the speed is now 9.6 Kbit/s. So, this baud rate is below the limit (even with a Min. Pulse Duration of 2.5 us) and thus IRSC must be set to 1.

Assumptions:

- Input UART clock = 90 MHz
- Internal clock divider = 3 (divide Input UART clock by 3)
- Baud rate = 9.6 Kbit/s
- IrDA transceiver is not inverting on both channels: for Tx and Rx, a Zero is represented by a positive pulse, and a One is represented by no pulse (line stays low).
- Interrupt: Sent to Arm platform when 1 char is received into the Rx FIFO (RDR).

Registers values and Programming orders:

```
UCR1 = 0x0085
UCR1[7] = IREN = 1: Enable IR interface
UCR1[0] = UARTEN = 1: Enable UART
UFCR = 0x0981
UFCR[15:10] = TXTL[5:0] = 0x02: Default value
RFDIV[2:0] = 0x3: Divide Input UART clock by 3 (resulting internal clock  is 30 MHz)
UFCR[5:0] = RXTL[5:0] = 0x01: Default value
UBIR = 0x00FF
UBMR = 0xC354 Baud rate = 9.6 kbit/s with internal clock = 30 MHz
UCR2 = 0x4027
UCR2[14] = IRTS = 1: Ignore level of RTS input signal
UCR2[5] = WS = 1: Characters are 8-bit length
UCR2[2] = TXEN = 1: Enable Rx path
UCR2 [1] = RXEN = 1: Enable Tx path
UCR2[0] = SRST_B = 1: No software reset
UCR3 = 0x0000
UCR3[1] = INVT = 0: Positive pulse represents 0.
UCR4 = 0x8221
UCR4[15:10] = CTSTL[5:0] = 0x20: Default value
UCR4[9] = INVR = 1: Inverted Infrared Reception (because IrDA transceiver is not inverting)
UCR4[5] = IRSC = 1: Because data rate is below the limit and thus the UART internal clock is
used to measure the pulse duration.
UCR4[1] = DREN = 1: To enable RDR interrupt (sent when one char is received)
```

The UART is now ready to send a character as soon as there is a write into UTXD register. An interrupt will be sent to Arm platform when a character is received.

64.7 9-bit RS-485 Mode

64.7.1 Generalities

The UART provides a 9-bit mode to facilitate multidrop (RS-485) network communication. To enable this mode, set MDEN bit in the UMCR register to 1. When 9-bit RS-485 mode is enabled, UART transmitter can transmit the ninth bit (9^{th} bit) set by TXB8, and UART receiver can differentiate between data frames (9^{th} bit = 0) and address frames (9^{th} bit = 1).

The CTS_B pin can be used to control RS-485 output driver outside the chip.

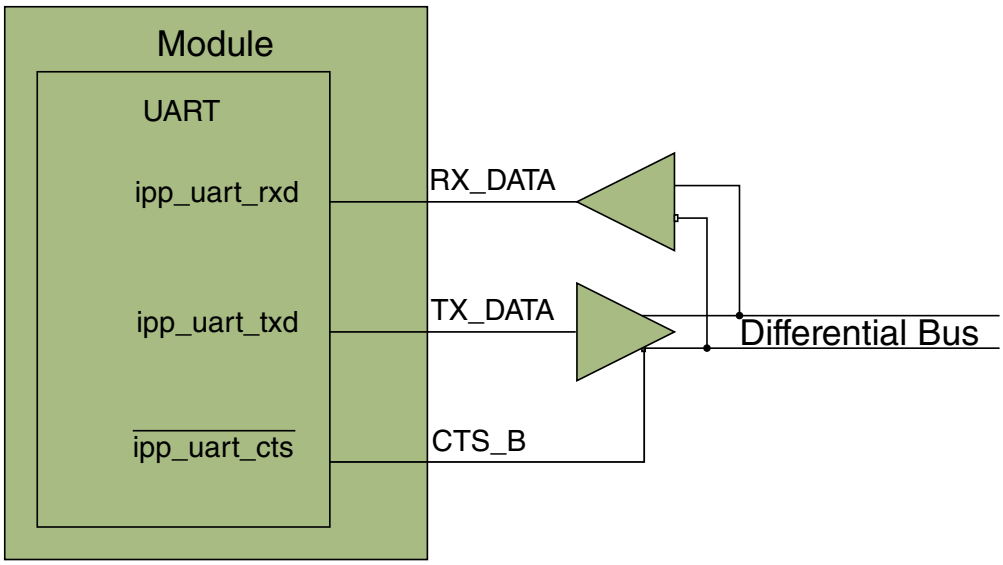


Figure 64-12. RS-485 driver connection (UART in DCE mode)

64.7.2 Transmit 9-bit RS-485 frames

To transmit 9-bit RS-485 frames, user need to enable parity (PREN=1) to enable trasmitting the ninth data bit, set 8-bit data word size (WS=1), and write TXB8 (UMCR[2]) as the 9th bit (bit [8]) to be transmitted (write '0' to TXB8 to transmit a data frame, write '1' to transmit a address frame). The other data bit [7:0] is written to TxFIFO by writing to the UTXD same as normal RS-232 operation.

64.7.3 Receive 9-bit RS-485 frames

To receive 9-bit RS-485 frames, user need to enable parity (PREN=1) to enable receiving the ninth data bit, set 8-bit data word size (WS=1). The receiver will save the 9-bit data to RxFIFO, and user should read the 9th databit (bit [8]) by reading the PRERR (URXD[10]) bit, and read data bit [7:0] by reading the RX_DATA (URXD[7:0]).

There are two slave address detect modes, normal detect mode and automatic detect mode, and can be selected by SLAM (UMCR[1]).

64.7.3.1 RS-485 Slave Address Normal Detect Mode

To enable Normal Detect mode, clear SLAM (UMCR[1] to 0). The receiver ignores all data frames (9^{th} bit = 0) until an address frame is received (9^{th} bit = 1). At that time, the slave address detected (SAD = USR1[3]) bit is asserted and the *interrupt_uart* interrupt is generated (if SADEN = UMCR[3] = 1). The address byte and sebsequent bytes are all put into RxFIFO along with their 9^{th} bit. The UART will also generate DMA request *dma_req_rx* when the RxFIFO reaches the selected threshold (controlled by RXTL) if receive ready DMA (RXDMAEN = UCR1[8]) request is enabled.

User should read the 9^{th} databit (bit [8]) by reading the PRERR (URXD[10]) bit, and read data bit [7:0] by reading the RX_DATA (URXD[7:0]).

In this mode, once the UART has detected a 9^{th} bit is equal to '1', it will always save the subsequent frames to RxFIFO. So the software must decide whether the address and data in RxFIFO are needed or not.

64.7.3.2 RS-485 Slave Address Automatic Detect Mode

To enable Automatic Detect Mode, set SLAM (UMCR[1]) to 1. The receiver tries to detect an address byte (frame 9^{th} bit = 1) that matches the programmed SLADDR (UMCR[15:8]) character. If the received byte is a data or an address byte that does not match the programmed SLADDR character, the receiver will discard these data.

Once the UART receives a matching address byte, it will assert the slave address detected (SAD = USR1[3]) bit and the *interrupt_uart* interrupt will be generated (if SADEN = UMCR[3] = 1). The address byte and sebsequent bytes are all put into RxFIFO along with their 9^{th} bit. If receive ready DMA(RXDMAEN = UCR1[8]) request is enabled, the UART will also generate DMA request *dma_req_rx* when the RxFIFO reaches the selected threshold (controlled by RXTL).

If another address byte is received and this address byte does not match SLADDR character, the receiver will discard the address byte and subsequent data byte. If the address byte again matches SLADDR character, the receiver will put this address byte and subsequent data byte in the RxFIFO along with their 9^{th} bit.

User should read the 9^{th} databit (bit [8]) by reading the PRERR (URXD[10]) bit, and read data bit [7:0] by reading the RX_DATA (URXD[7:0]).

See Initialization for 9-bit RS-485 programming guide.

64.8 Low Power Modes

These modes are controlled by the signals *doze_req* and *stop_req*. The control/status/data registers won't change when getting in/out of low power modes.

Table 64-15. UART Low Power State Operation

	Normal State (*doze_req* = 1'b0 & *stop_req* = 1'b0)	Doze State (*doze_req* = 1'b1)		Stop State (*stop_req* = 1'b1)
		DOZE bit = 0	DOZE bit = 1	
UART-Clock	ON	ON	ON	OFF
UART Serial / IrDA	ON	ON	OFF	OFF

64.8.1 UART Operation in System Doze Mode

While in Doze State (when *doze_req* input pin is set to 1'b1), the UART behavior depends on the DOZE (UCR1[1]) control bit.

While the DOZE bit is negated, the UART serial interface is enabled. While the system is in the Doze State, and the DOZE bit is asserted, the UART is disabled. If the Doze State is entered with the DOZE bit asserted while the UART serial interface was receiving or transmitting data, it will complete the receive/transmit of the current character and signal to the far-end transmitter/receiver to stop sending/receiving.

64.8.2 UART Operation in System Stop Mode

The internal baud rate clocks of the transmitter and receiver are gated off if the *stop_req* signal to UART is asserted. Even though the clocks at the input of the UART continue to run during system Stop mode, the UART will not do any transmission or reception.

The following UART interrupts wake the Arm platform processor from STOP mode:

- RTS (RTSD)
- IrDA Asynchronous WAKE (AIRINT)
- Asynchronous WAKE (AWAKE)
- RI (RIDELT in DTE mode only)
- DCD (DCDDELT in DTE mode only)
- DTR (DTRD in DCE mode only)
- DSR (DTRD in DTE mode only)

When an asynchronous WAKE (awake) interrupt exits the Arm platform from STOP mode, make sure that a dummy character is sent first because the first character may not be received correctly.

64.8.3 Power Saving Method in UART

The RXEN (UCR2[1]), TXEN (UCR2[2]) and UARTEN (UCR1[0]) bits are set by the user and provide software control of low-power modes.

Setting the UARTEN (UCR1[0]) bit to 0 shuts off the receiver and transmitter logic and the associated clocks.

If the UART is used only in transmit mode, UARTEN and TXEN must be set to 1. If the UART is used only in receive mode, UARTEN and RXEN must be set to 1. Setting TXEN or RXEN to 0 allows to save a lot of power.

64.9 UART Operation in System Debug State

The bit UTS [11] controls whether the UART will respond to the input signal *debug_req*, or whether it will continue to run as normal.

If the UART is programmed to respond to *debug_req*:

1. The UART will halt all operations upon detecting the *debug_req* input.
2. A transfer in progress, either to/from a core (using the IP Bus interface) or to/from an external device, will be completed before halting. This means a single byte/word transfer, not an entire FIFO. Reception of any further data from an external device will be disabled.
3. Internal registers will continue to be writable and readable using the IP Bus interface. A read will leave the contents unaffected.
4. The RX FIFO is affected in debug mode in the following way:
   - All writes into the RX FIFO are prevented.
   - The bit RXDBG (UTS[9]) is used to select the readability of the RX FIFO during debug mode:

     RXDBG = 0: hold the read pointer at the location it had upon entering debug mode, and URXD register returns only the data value at that location, no matter how many reads attempted.

RXDBG = 1, selectable at any time: Allow to read the characters received in Rx FIFO. It will not be possible to re-read previously read locations, nor will it be possible to readjust the read pointer to the value it had prior to entering debug mode.

64.10 Reset

This section describes how to reset the block and explains special requirements related to reset.

64.10.1 Hardware reset

All of registers, FIFOs, state machines and sequential elements can be reset to their initial values by hardware reset or power on reset.

64.10.2 Software reset

The status registers USR1 and USR2, BRM registers UBIR and UBMR, TxFIFO and RxFIFO, and transmitter and receiver state machines can be reset by software reset. Internal logic will keep the software reset asserted for about 4 *module_clock* cycles.

Programmer can follow the following software reset sequence:

1. Clear the SRST_B bit (UCR2[0])
2. Wait for software reset complete: poll SOFTRST bit (UTS[0]) until it is 0.
3. Re-program baud rate registers: Re-write UBIR and UBMR.

64.11 Transfer Error

The UART can generate a transfer error on the peripheral bus in the following cases:

- Core is writing into a read-only register.
- Core is accessing (read or write) an unused location within the assigned address space reserved to UART.

- Core is writing into UTXD register with transmit interface disabled (TXEN=0 or UARTEN=0)
- Core is reading URXD register with receive interface disabled (RXEN=0 or UARTEN=0)

64.12 Functional Timing

This section includes timing diagrams for functional signaling.

64.12.1 IrDA Mode

According to IrDA specification, the low speed (115.2Kbit/s and below) IR frame format is compatible with UART frame.

In this figure, an example data 0x65 is used.

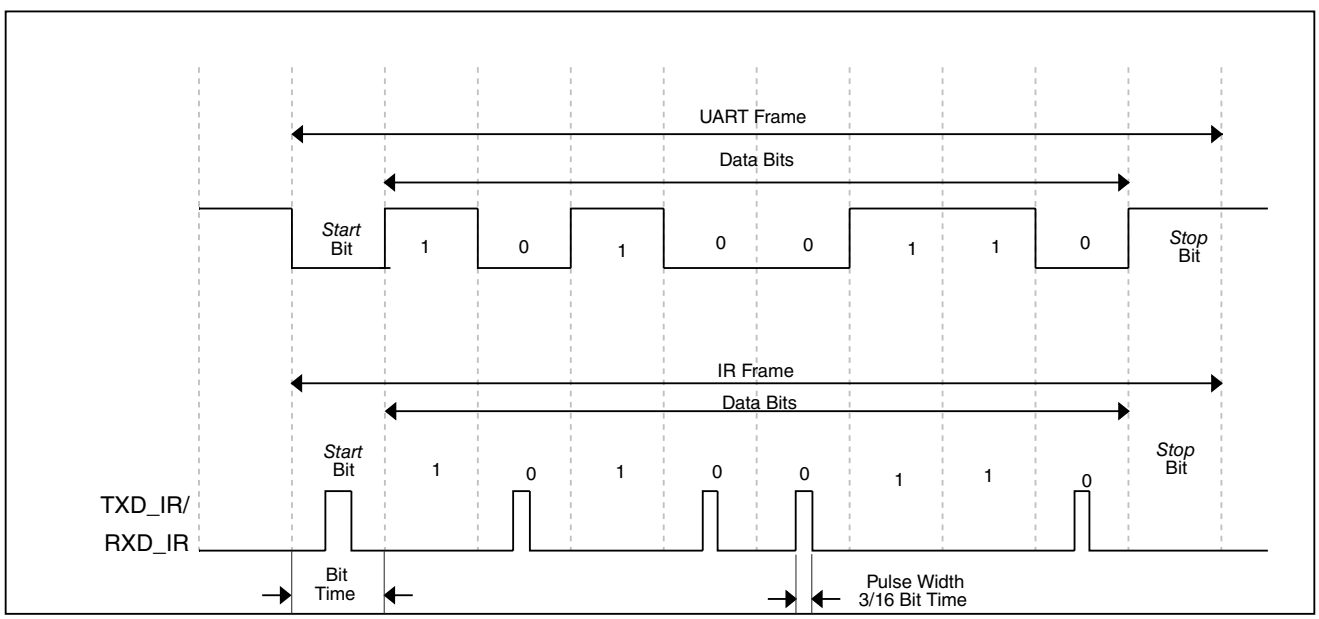


Figure 64-13. Timing diagram of Low Speed IR (<=115.2 Kbit/s) Data Line

64.13 Initialization

64.13.1 Programming the UART in RS-232 mode

As an example, the following sequence can be used to program the UART in order to send and receive characters in RS-232 mode.

Assumptions:

- Input uart clock = 100 MHz
- Baud rate = 921.6 Kbps
- Data bits = 8 bits
- Parity = Even
- Stop bits = 1 bit
- Flow control = Hardware

Main program:

1. UCR1 = 0x0001

   Enable the UART.

2. UCR2 = 0x2127

   Set hardware flow control, data format and enable transmitter and receiver.

3. UCR3 = 0x0704

   Set UCR3[RXDMUXSEL] = 1.

4. UCR4 = 0x7C00

   Set CTS trigger level to 31,

5. UFCR = 0x089E

   Set internal clock divider = 5 (divide input uart clock by 5). So the reference clock is 100 MHz/5 = 20 MHz.

   Set TXTL = 2 and RXTL = 30.

6. UBIR = 0x08FF
7. UBMR = 0x0C34

   In the above two steps, set baud rate to 921.6Kbps based on the 20MHz reference clock.

8. UCR1 = 0x2201

   Enable the TRDY and RRDY interrupts.

9. UMCR = 0x0000

UMCR stay at default value 0x0000

Interrupt service routine for the transmitter:

- Write characters into UTXD

The TRDY interrupt will be automatically de-asserted when the data level of the TxFIFO exceeds the TXTL=2. Note: For the first time the interrupt may be de-asserted after 4 characters are written into the TxFIFO because of the shift register.

Interrupt service routine for the receiver:

- Read characters from URXD

The RRDY interrupt will be automatically de-asserted when the data level of the RxFIFO is below the RXTL=30.

64.13.2 Programming the UART in 9-bit RS-485 mode

As an example, the following sequence can be used to program the UART in order to send and receive frames in RS-485 mode.

Assumptions:

- Input uart clock = 100 MHz
- Baud rate = 5 Mbps

Main program:

1. UCR1 = 0x0001

   Enable the UART.

2. UCR2 = 0x4127

   Set software flow control ($\overline{\text{CTS}}$ pin is controlled by UCR2[12]), enable parity(enable 9^{th} bit rxd/txd), 8-bit word size , and enable transmitter and receiver.

3. UCR4 = 0x7C00

   Set CTS trigger level to 31,

4. UFCR = 0x0A9E

   Set RFDIV = 5 (divide input uart clock by 1), so the reference clock is 100 MHz. Set UART in DCE mode (RS-485 driver connection outside the chip is the same as Figure 64-12)

Set TXTL = 2 and RXTL = 30.

5. UBIR = 0x0003
6. UBMR = 0x0004

   In the above two steps, set baud rate to 5 Mbps based on the 100 MHz reference clock.

7. UCR1 = 0x2001 when UART as a master ,

   or UCR1 = 0x0201 (or 0x0101) when UART as a slave.

   Enable TRDY interrupt when UART as a master, enable RRDY interrupt or DMA request when UART as a slave.

8. UMCR = 0xA50B

   Enable 9-bit RS-485 mode, enable SAD interrupt, set automatic slave address detect mode, set slave address is 0xA5.

Interrupt service routine for the transmitter:

- Transmit data: write its ninth bit (bit[8]) to UMCR[2], write its bit [7:0] into UTXD[7:0]

The TRDY interrupt will be automatically de-asserted when the data level of the TxFIFO exceeds the TXTL=2.

Note: For the first time the interrupt may be de-asserted after 4 characters are written into the TxFIFO because of the shift register.

Interrupt service routine for the receiver:

- Receive data: read its ninth bit (bit[8]) from URXD[10] , read its bit [7:0] from URXD[7:0].

Note: in RS-485 mode, URXD[10] bit is not the parity error, instead it holds the ninth bit (bit[8]) of the received data.

The SAD interrupt can not de-assert automatically, it needs MCU write 1 to USR1[3] to clear it . The RRDY interrupt or DMA request will be automatically de-asserted when the data level of the RxFIFO is below the RXTL=30.

64.14 References

- EIA/TIA-232-F Interface Standard

http://www.eia.org, http://www.tiaonline.org/standards

- IrDA Standard

  http://www.irda.org

64.15 UART Memory Map/Register Definition

UART supports 8-bit, 16-bit and 32-bit accesses to 32-bit memory-mapped addresses. Any access to unmapped memory location will yield a transfer error.

All registers except the ONEMS described in this section are 16-bit registers. The ONEMS register is a 24-bit register.

- For 32-bit write accesses, the upper two bytes will not be taken into account.
- For 32-bit read accesses the upper two bytes will return 0.

The ONEMS register is expanded from 16 bits to 24 bits in order to support the high frequency of the BRM internal clock *ref_clk* (*module_clock* after divider). The ONEMS register can be accessed as 8 bits, 16 bits or 32 bits.

- For 32-bit write accesses, the most significant byte of the ONEMS will be discarded.
- For 32-bit read accesses, the most significant byte of the ONEMS will be read as 0.

UART memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
202_0000	UART Receiver Register (UART1_URXD)	32	R	0000_0000h	64.15.1/ 5187
202_0040	UART Transmitter Register (UART1_UTXD)	32	W	0000_0000h	64.15.2/ 5189
202_0080	UART Control Register 1 (UART1_UCR1)	32	R/W	0000_0000h	64.15.3/ 5190
202_0084	UART Control Register 2 (UART1_UCR2)	32	R/W	0000_0001h	64.15.4/ 5192
202_0088	UART Control Register 3 (UART1_UCR3)	32	R/W	0000_0700h	64.15.5/ 5195
202_008C	UART Control Register 4 (UART1_UCR4)	32	R/W	0000_8000h	64.15.6/ 5197
202_0090	UART FIFO Control Register (UART1_UFCR)	32	R/W	0000_0801h	64.15.7/ 5199

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
202_0094	UART Status Register 1 (UART1_USR1)	32	R/W	0000_2040h	64.15.8/ 5201
202_0098	UART Status Register 2 (UART1_USR2)	32	R/W	0000_4028h	64.15.9/ 5204
202_009C	UART Escape Character Register (UART1_UESC)	32	R/W	0000_002Bh	64.15.10/ 5206
202_00A0	UART Escape Timer Register (UART1_UTIM)	32	R/W	0000_0000h	64.15.11/ 5207
202_00A4	UART BRM Incremental Register (UART1_UBIR)	32	R/W	0000_0000h	64.15.12/ 5207
202_00A8	UART BRM Modulator Register (UART1_UBMR)	32	R/W	0000_0000h	64.15.13/ 5208
202_00AC	UART Baud Rate Count Register (UART1_UBRC)	32	R	0000_0004h	64.15.14/ 5208
202_00B0	UART One Millisecond Register (UART1_ONEMS)	32	R/W	0000_0000h	64.15.15/ 5209
202_00B4	UART Test Register (UART1_UTS)	32	R/W	0000_0060h	64.15.16/ 5210
202_00B8	UART RS-485 Mode Control Register (UART1_UMCR)	32	R/W	0000_0000h	64.15.17/ 5211
21E_8000	UART Receiver Register (UART2_URXD)	32	R	0000_0000h	64.15.1/ 5187
21E_8040	UART Transmitter Register (UART2_UTXD)	32	W	0000_0000h	64.15.2/ 5189
21E_8080	UART Control Register 1 (UART2_UCR1)	32	R/W	0000_0000h	64.15.3/ 5190
21E_8084	UART Control Register 2 (UART2_UCR2)	32	R/W	0000_0001h	64.15.4/ 5192
21E_8088	UART Control Register 3 (UART2_UCR3)	32	R/W	0000_0700h	64.15.5/ 5195
21E_808C	UART Control Register 4 (UART2_UCR4)	32	R/W	0000_8000h	64.15.6/ 5197
21E_8090	UART FIFO Control Register (UART2_UFCR)	32	R/W	0000_0801h	64.15.7/ 5199
21E_8094	UART Status Register 1 (UART2_USR1)	32	R/W	0000_2040h	64.15.8/ 5201
21E_8098	UART Status Register 2 (UART2_USR2)	32	R/W	0000_4028h	64.15.9/ 5204
21E_809C	UART Escape Character Register (UART2_UESC)	32	R/W	0000_002Bh	64.15.10/ 5206
21E_80A0	UART Escape Timer Register (UART2_UTIM)	32	R/W	0000_0000h	64.15.11/ 5207
21E_80A4	UART BRM Incremental Register (UART2_UBIR)	32	R/W	0000_0000h	64.15.12/ 5207

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21E_80A8	UART BRM Modulator Register (UART2_UBMR)	32	R/W	0000_0000h	64.15.13/ 5208
21E_80AC	UART Baud Rate Count Register (UART2_UBRC)	32	R	0000_0004h	64.15.14/ 5208
21E_80B0	UART One Millisecond Register (UART2_ONEMS)	32	R/W	0000_0000h	64.15.15/ 5209
21E_80B4	UART Test Register (UART2_UTS)	32	R/W	0000_0060h	64.15.16/ 5210
21E_80B8	UART RS-485 Mode Control Register (UART2_UMCR)	32	R/W	0000_0000h	64.15.17/ 5211
21E_C000	UART Receiver Register (UART3_URXD)	32	R	0000_0000h	64.15.1/ 5187
21E_C040	UART Transmitter Register (UART3_UTXD)	32	W	0000_0000h	64.15.2/ 5189
21E_C080	UART Control Register 1 (UART3_UCR1)	32	R/W	0000_0000h	64.15.3/ 5190
21E_C084	UART Control Register 2 (UART3_UCR2)	32	R/W	0000_0001h	64.15.4/ 5192
21E_C088	UART Control Register 3 (UART3_UCR3)	32	R/W	0000_0700h	64.15.5/ 5195
21E_C08C	UART Control Register 4 (UART3_UCR4)	32	R/W	0000_8000h	64.15.6/ 5197
21E_C090	UART FIFO Control Register (UART3_UFCR)	32	R/W	0000_0801h	64.15.7/ 5199
21E_C094	UART Status Register 1 (UART3_USR1)	32	R/W	0000_2040h	64.15.8/ 5201
21E_C098	UART Status Register 2 (UART3_USR2)	32	R/W	0000_4028h	64.15.9/ 5204
21E_C09C	UART Escape Character Register (UART3_UESC)	32	R/W	0000_002Bh	64.15.10/ 5206
21E_C0A0	UART Escape Timer Register (UART3_UTIM)	32	R/W	0000_0000h	64.15.11/ 5207
21E_C0A4	UART BRM Incremental Register (UART3_UBIR)	32	R/W	0000_0000h	64.15.12/ 5207
21E_C0A8	UART BRM Modulator Register (UART3_UBMR)	32	R/W	0000_0000h	64.15.13/ 5208
21E_C0AC	UART Baud Rate Count Register (UART3_UBRC)	32	R	0000_0004h	64.15.14/ 5208
21E_C0B0	UART One Millisecond Register (UART3_ONEMS)	32	R/W	0000_0000h	64.15.15/ 5209
21E_C0B4	UART Test Register (UART3_UTS)	32	R/W	0000_0060h	64.15.16/ 5210
21E_C0B8	UART RS-485 Mode Control Register (UART3_UMCR)	32	R/W	0000_0000h	64.15.17/ 5211

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21F_0000	UART Receiver Register (UART4_URXD)	32	R	0000_0000h	64.15.1/ 5187
21F_0040	UART Transmitter Register (UART4_UTXD)	32	W	0000_0000h	64.15.2/ 5189
21F_0080	UART Control Register 1 (UART4_UCR1)	32	R/W	0000_0000h	64.15.3/ 5190
21F_0084	UART Control Register 2 (UART4_UCR2)	32	R/W	0000_0001h	64.15.4/ 5192
21F_0088	UART Control Register 3 (UART4_UCR3)	32	R/W	0000_0700h	64.15.5/ 5195
21F_008C	UART Control Register 4 (UART4_UCR4)	32	R/W	0000_8000h	64.15.6/ 5197
21F_0090	UART FIFO Control Register (UART4_UFCR)	32	R/W	0000_0801h	64.15.7/ 5199
21F_0094	UART Status Register 1 (UART4_USR1)	32	R/W	0000_2040h	64.15.8/ 5201
21F_0098	UART Status Register 2 (UART4_USR2)	32	R/W	0000_4028h	64.15.9/ 5204
21F_009C	UART Escape Character Register (UART4_UESC)	32	R/W	0000_002Bh	64.15.10/ 5206
21F_00A0	UART Escape Timer Register (UART4_UTIM)	32	R/W	0000_0000h	64.15.11/ 5207
21F_00A4	UART BRM Incremental Register (UART4_UBIR)	32	R/W	0000_0000h	64.15.12/ 5207
21F_00A8	UART BRM Modulator Register (UART4_UBMR)	32	R/W	0000_0000h	64.15.13/ 5208
21F_00AC	UART Baud Rate Count Register (UART4_UBRC)	32	R	0000_0004h	64.15.14/ 5208
21F_00B0	UART One Millisecond Register (UART4_ONEMS)	32	R/W	0000_0000h	64.15.15/ 5209
21F_00B4	UART Test Register (UART4_UTS)	32	R/W	0000_0060h	64.15.16/ 5210
21F_00B8	UART RS-485 Mode Control Register (UART4_UMCR)	32	R/W	0000_0000h	64.15.17/ 5211
21F_4000	UART Receiver Register (UART5_URXD)	32	R	0000_0000h	64.15.1/ 5187
21F_4040	UART Transmitter Register (UART5_UTXD)	32	W	0000_0000h	64.15.2/ 5189
21F_4080	UART Control Register 1 (UART5_UCR1)	32	R/W	0000_0000h	64.15.3/ 5190
21F_4084	UART Control Register 2 (UART5_UCR2)	32	R/W	0000_0001h	64.15.4/ 5192
21F_4088	UART Control Register 3 (UART5_UCR3)	32	R/W	0000_0700h	64.15.5/ 5195

Table continues on the next page...

UART memory map (continued)

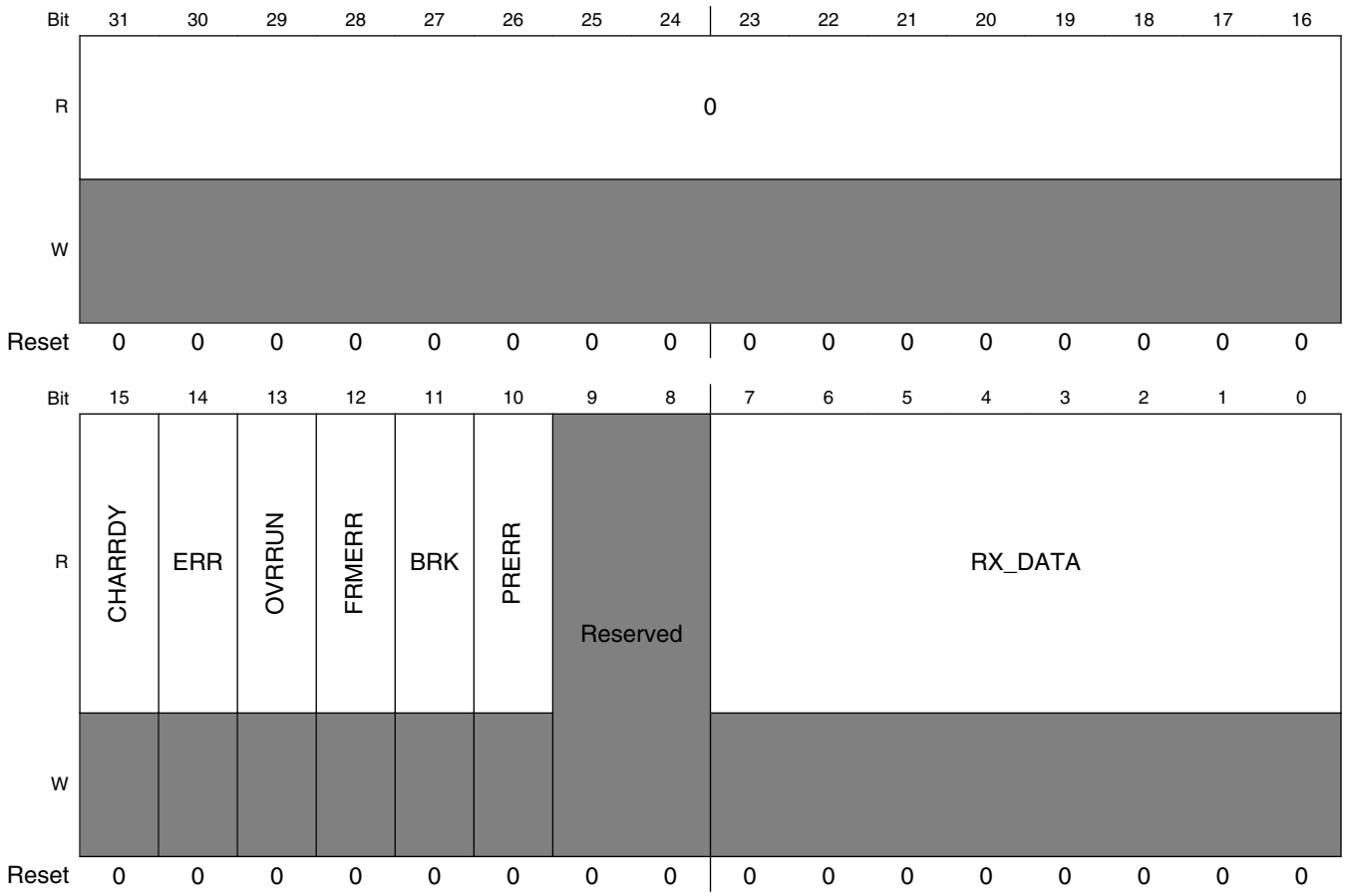
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21F_408C	UART Control Register 4 (UART5_UCR4)	32	R/W	0000_8000h	64.15.6/ 5197
21F_4090	UART FIFO Control Register (UART5_UFCR)	32	R/W	0000_0801h	64.15.7/ 5199
21F_4094	UART Status Register 1 (UART5_USR1)	32	R/W	0000_2040h	64.15.8/ 5201
21F_4098	UART Status Register 2 (UART5_USR2)	32	R/W	0000_4028h	64.15.9/ 5204
21F_409C	UART Escape Character Register (UART5_UESC)	32	R/W	0000_002Bh	64.15.10/ 5206
21F_40A0	UART Escape Timer Register (UART5_UTIM)	32	R/W	0000_0000h	64.15.11/ 5207
21F_40A4	UART BRM Incremental Register (UART5_UBIR)	32	R/W	0000_0000h	64.15.12/ 5207
21F_40A8	UART BRM Modulator Register (UART5_UBMR)	32	R/W	0000_0000h	64.15.13/ 5208
21F_40AC	UART Baud Rate Count Register (UART5_UBRC)	32	R	0000_0004h	64.15.14/ 5208
21F_40B0	UART One Millisecond Register (UART5_ONEMS)	32	R/W	0000_0000h	64.15.15/ 5209
21F_40B4	UART Test Register (UART5_UTS)	32	R/W	0000_0060h	64.15.16/ 5210
21F_40B8	UART RS-485 Mode Control Register (UART5_UMCR)	32	R/W	0000_0000h	64.15.17/ 5211

64.15.1 UART Receiver Register (UARTx_URXD)

NOTE

The UART will yield a transfer error on the peripheral bus when core is reading URXD register with receive interface disabled (RXEN=0 or UARTEN=0).

Address: Base address + 0h offset



UARTx_URXD field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 CHARRDY	Character Ready. This read-only bit indicates an invalid read when the FIFO becomes empty and software tries to read the same old data. This bit should not be used for polling for data written to the RX FIFO. 0 Character in RX_DATA field and associated flags are invalid. 1 Character in RX_DATA field and associated flags valid and ready for reading.

Table continues on the next page...

UARTx_URXD field descriptions (continued)

Field	Description
14 ERR	**Error Detect.** Indicates whether the character present in the RX_DATA field has an error (OVRRUN, FRMERR, BRK or PRERR) status. The ERR bit is updated and valid for each received character. 0 No error status was detected 1 An error status was detected
13 OVRRUN	**Receiver Overrun.** This read-only bit, when HIGH, indicates that the corresponding character was stored in the last position (32nd) of the Rx FIFO. Even if a 33rd character has not been detected, this bit will be set to '1' for the 32nd character. 0 No RxFIFO overrun was detected 1 A RxFIFO overrun was detected
12 FRMERR	**Frame Error.** Indicates whether the current character had a framing error (a missing stop bit) and is possibly corrupted. FRMERR is updated for each character read from the RxFIFO. 0 The current character has no framing error 1 The current character has a framing error
11 BRK	**BREAK Detect.** Indicates whether the current character was detected as a BREAK character. The data bits and the stop bit are all 0. The FRMERR bit is set when BRK is set. When odd parity is selected, PRERR is also set when BRK is set. BRK is valid for each character read from the RxFIFO. 0 The current character is not a BREAK character 1 The current character is a BREAK character
10 PRERR	**In RS-485 mode, it holds the ninth data bit (bit [8]) of received 9-bit RS-485 data** **In RS232/IrDA mode, it is the Parity Error flag**. Indicates whether the current character was detected with a parity error and is possibly corrupted. PRERR is updated for each character read from the RxFIFO. When parity is disabled, PRERR always reads as 0. 0 = No parity error was detected for data in the RX_DATA field 1 = A parity error was detected for data in the RX_DATA field
9–8 -	This field is reserved. **Reserved**
RX_DATA	**Received Data**. Holds the received character. In 7-bit mode, the most significant bit (MSB) is forced to 0. In 8-bit mode, all bits are active.

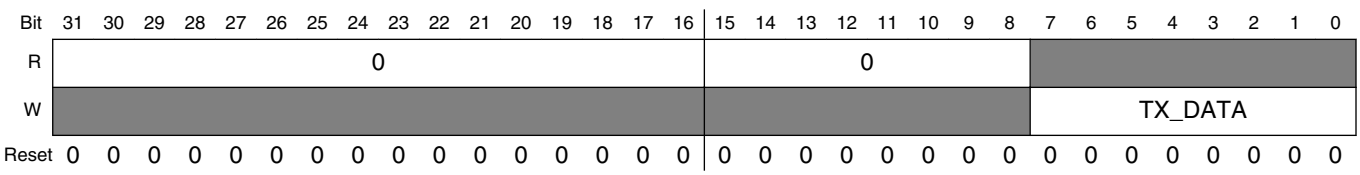
64.15.2 UART Transmitter Register (UARTx_UTXD)

NOTE

The UART will yield a transfer error on the peripheral bus when core is writing into UART_URXD register with transmit interface disabled (TXEN=0 or UARTEN=0).

Memory space between UART_URXD and UART_UTXD registers is reserved. Any read or write access to this space will be considered as an invalid access and yield a transfer error.

Address: Base address + 40h offset



UARTx_UTXD field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 Reserved	This read-only field is reserved and always has the value 0.
TX_DATA	**Transmit Data**. Holds the parallel transmit data inputs. In 7-bit mode, D7 is ignored. In 8-bit mode, all bits are used. Data is transmitted least significant bit (LSB) first. A new character is transmitted when the TX_DATA field is written. The TX_DATA field must be written only when the TRDY bit is high to ensure that corrupted data is not sent.

64.15.3 UART Control Register 1 (UARTx_UCR1)

Address: Base address + 80h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADEN	ADBR	TRDYEN	IDEN	ICD		RRDYEN	RXDMAEN	IREN	TXMPTYEN	RTSDEN	SNDBRK	TXDMAEN	ATDMAEN	DOZE	UARTEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UCR1 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 ADEN	**Automatic Baud Rate Detection Interrupt Enable**. Enables/Disables the automatic baud rate detect complete (ADET) bit to generate an interrupt (*interrupt_uart* = 0). 0 Disable the automatic baud rate detection interrupt 1 Enable the automatic baud rate detection interrupt
14 ADBR	**Automatic Detection of Baud Rate**. Enables/Disables automatic baud rate detection. When the ADBR bit is set and the ADET bit is cleared, the receiver detects the incoming baud rate automatically. The ADET flag is set when the receiver verifies that the incoming baud rate is detected properly by detecting an ASCII character "A" or "a" (0x41 or 0x61). 0 Disable automatic detection of baud rate 1 Enable automatic detection of baud rate
13 TRDYEN	**Transmitter Ready Interrupt Enable**. Enables/Disables the transmitter Ready Interrupt (TRDY) when the transmitter has one or more slots available in the TxFIFO. The fill level in the TXFIFO at which an interrupt is generated is controlled by TxTL bits. When TRDYEN is negated, the transmitter ready interrupt is disabled. **NOTE:** An interrupt will be issued as long as TRDYEN and TRDY are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the TRDY interrupt. 0 Disable the transmitter ready interrupt 1 Enable the transmitter ready interrupt
12 IDEN	**Idle Condition Detected Interrupt Enable**. Enables/Disables the IDLE bit to generate an interrupt (*interrupt_uart* = 0). 0 Disable the IDLE interrupt 1 Enable the IDLE interrupt

Table continues on the next page...

UARTx_UCR1 field descriptions (continued)

Field	Description
11–10 ICD	**Idle Condition Detect**. Controls the number of frames RXD is allowed to be idle before an idle condition is reported. 00 Idle for more than 4 frames 01 Idle for more than 8 frames 10 Idle for more than 16 frames 11 Idle for more than 32 frames
9 RRDYEN	**Receiver Ready Interrupt Enable**. Enables/Disables the RRDY interrupt when the RxFIFO contains data. The fill level in the RxFIFO at which an interrupt is generated is controlled by the RXTL bits. When RRDYEN is negated, the receiver ready interrupt is disabled. 0 Disables the RRDY interrupt 1 Enables the RRDY interrupt
8 RXDMAEN	**Receive Ready DMA Enable**. Enables/Disables the receive DMA request *dma_req_rx* when the receiver has data in the RxFIFO. The fill level in the RxFIFO at which a DMA request is generated is controlled by the RXTL bits. When negated, the receive DMA request is disabled. 0 Disable DMA request 1 Enable DMA request
7 IREN	**Infrared Interface Enable**. Enables/Disables the IR interface. See the IR interface description in Infrared Interface, for more information. Note: MDEN(UMCR[0]) must be cleared to 0 when using IrDA interface. See Table 64-1 0 Disable the IR interface 1 Enable the IR interface
6 TXMPTYEN	**Transmitter Empty Interrupt Enable**. Enables/Disables the transmitter FIFO empty (TXFE) interrupt. *interrupt_uart*. When negated, the TXFE interrupt is disabled. **NOTE:** An interrupt will be issued as long as TXMPTYEN and TXFE are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the TXFE interrupt. 0 Disable the transmitter FIFO empty interrupt 1 Enable the transmitter FIFO empty interrupt
5 RTSDEN	**RTS Delta Interrupt Enable**. Enables/Disables the RTSD interrupt. The current status of the RTS_B pin is read in the RTSS bit. 0 Disable RTSD interrupt 1 Enable RTSD interrupt
4 SNDBRK	**Send BREAK**. Forces the transmitter to send a BREAK character. The transmitter finishes sending the character in progress (if any) and sends BREAK characters until SNDBRK is reset. Because the transmitter samples SNDBRK after every bit is transmitted, it is important that SNDBRK is asserted high for a sufficient period of time to generate a valid BREAK. After the BREAK transmission completes, the UART transmits 2 mark bits. The user can continue to fill the TxFIFO and any characters remaining are transmitted when the BREAK is terminated. 0 Do not send a BREAK character 1 Send a BREAK character (continuous 0s)
3 TXDMAEN	**Transmitter Ready DMA Enable**. Enables/Disables the transmit DMA request *dma_req_tx* when the transmitter has one or more slots available in the TxFIFO. The fill level in the TxFIFO that generates the *dma_req_tx* is controlled by the TXTL bits.

Table continues on the next page...

UARTx_UCR1 field descriptions (continued)

Field	Description
	NOTE: A DMA request will be issued as long as TXDMAEN and TRDY are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the transmit DMA request. 0 Disable transmit DMA request 1 Enable transmit DMA request
2 ATDMAEN	**Aging DMA Timer Enable**. Enables/Disables the receive DMA request *dma_req_rx* for the aging timer interrupt (triggered with AGTIM flag in USR1[8]). 0 Disable AGTIM DMA request 1 Enable AGTIM DMA request
1 DOZE	**DOZE**. Determines the UART enable condition in the DOZE state. When *doze_req* input pin is at '1', (the Arm Platform executes a doze instruction and the system is placed in the Doze State), the DOZE bit affects operation of the UART. While in the Doze State, if this bit is asserted, the UART is disabled. See the description in Low Power Modes. 0 The UART is enabled when in DOZE state 1 The UART is disabled when in DOZE state
0 UARTEN	**UART Enable**. Enables/Disables the UART. If UARTEN is negated in the middle of a transmission, the transmitter stops and pulls the TXD line to a logic 1. UARTEN must be set to 1 before any access to UTXD and URXD registers, otherwise a transfer error is returned. This bit can be set to 1 along with other bits in this register. There is no restriction to the sequence of programing this bit and other control registers. 0 Disable the UART 1 Enable the UART

64.15.4 UART Control Register 2 (UARTx_UCR2)

Address: Base address + 84h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ESCI	IRTS	CTSC	CTS	ESCEN	RTEC		PREN	PROE	STPB	WS	RTSEN	ATEN	TXEN	RXEN	SRST
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

UARTx_UCR2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 ESCI	**Escape Sequence Interrupt Enable**. Enables/Disables the ESCF bit to generate an interrupt. 0 Disable the escape sequence interrupt 1 Enable the escape sequence interrupt
14 IRTS	**Ignore RTS Pin**. Forces the RTS input signal presented to the transmitter to always be asserted (set to low), effectively ignoring the external pin. When in this mode, the RTS pin serves as a general purpose input. 0 Transmit only when the RTS pin is asserted 1 Ignore the RTS pin
13 CTSC	CTS **Pin Control**. Controls the operation of the CTS_B module output. When CTSC is asserted, the CTS_B module output is controlled by the receiver. When the RxFIFO is filled to the level of the programmed trigger level and the start bit of the overflowing character (TRIGGER LEVEL + 1) is validated, the CTS_B module output is negated to indicate to the far-end transmitter to stop transmitting. When the trigger level is programmed for less than 32, the receiver continues to receive data until the RxFIFO is full. When the CTSC bit is negated, the CTS_B module output is controlled by the CTS bit. On reset, because CTSC is cleared to 0, the CTS_B pin is controlled by the CTS bit, which again is cleared to 0 on reset. This means that on reset the CTS_B signal is negated. 0 The CTS_B pin is controlled by the CTS bit 1 The CTS_B pin is controlled by the receiver
12 CTS	**Clear to Send**. Controls the CTS_B pin when the CTSC bit is negated. CTS has no function when CTSC is asserted. 0 The CTS_B pin is high (inactive) 1 The CTS_B pin is low (active)
11 ESCEN	**Escape Enable**. Enables/Disables the escape sequence detection logic. 0 Disable escape sequence detection 1 Enable escape sequence detection
10–9 RTEC	**Request to Send Edge Control**. Selects the edge that triggers the RTS interrupt. This has no effect on the RTS delta interrupt. RTEC has an effect only when RTSEN = 1 (see Table 64-9). 00 Trigger interrupt on a rising edge 01 Trigger interrupt on a falling edge 1X Trigger interrupt on any edge
8 PREN	**Parity Enable**. Enables/Disables the parity generator in the transmitter and parity checker in the receiver. When PREN is asserted, the parity generator and checker are enabled, and disabled when PREN is negated. 0 Disable parity generator and checker 1 Enable parity generator and checker
7 PROE	**Parity Odd/Even**. Controls the sense of the parity generator and checker. When PROE is high, odd parity is generated and expected. When PROE is low, even parity is generated and expected. PROE has no function if PREN is low. 0 Even parity 1 Odd parity

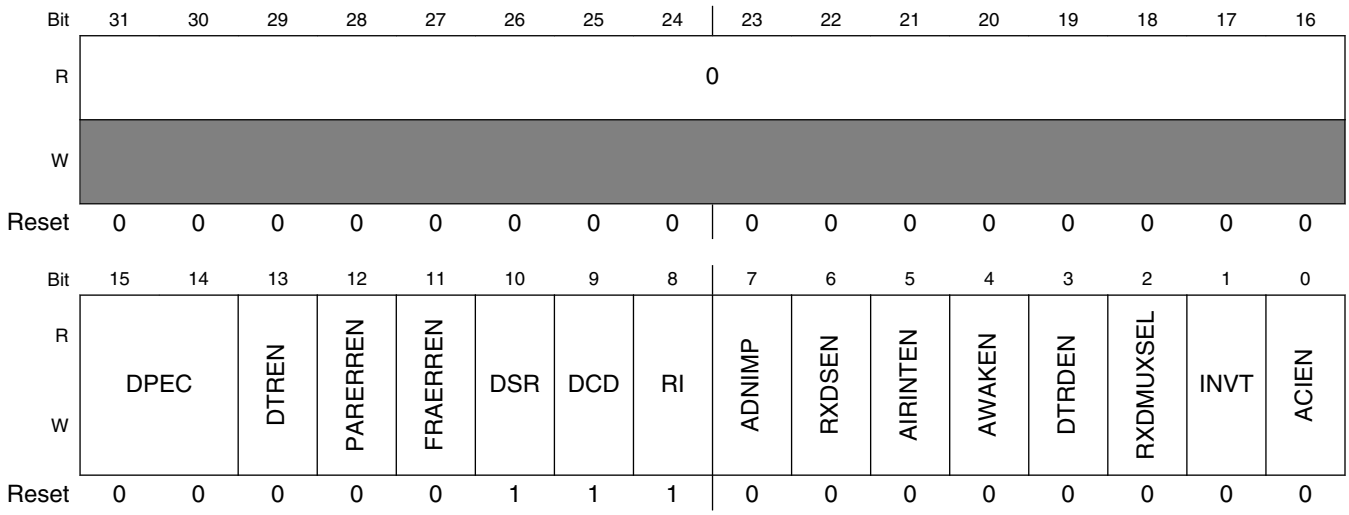
Table continues on the next page...

UARTx_UCR2 field descriptions (continued)

Field	Description
6 STPB	**Stop**. Controls the number of stop bits after a character. When STPB is low, 1 stop bit is sent. When STPB is high, 2 stop bits are sent. STPB also affects the receiver. 0 The transmitter sends 1 stop bit. The receiver expects 1 or more stop bits. 1 The transmitter sends 2 stop bits. The receiver expects 2 or more stop bits.
5 WS	**Word Size**. Controls the character length. When WS is high, the transmitter and receiver are in 8-bit mode. When WS is low, they are in 7-bit mode. The transmitter ignores bit 7 and the receiver sets bit 7 to 0. WS can be changed in-between transmission (reception) of characters, however not when a transmission (reception) is in progress, in which case the length of the current character being transmitted (received) is unpredictable. 0 7-bit transmit and receive character length (not including START, STOP or PARITY bits) 1 8-bit transmit and receive character length (not including START, STOP or PARITY bits)
4 RTSEN	**Request to Send Interrupt Enable.** Controls the RTS edge sensitive interrupt. When RTSEN is asserted and the programmed edge is detected on the RTS_B pin (the RTSF bit is asserted), an interrupt will be generated on the *interrupt_uart* pin. (See Table 64-9.) 0 Disable request to send interrupt 1 Enable request to send interrupt
3 ATEN	**Aging Timer Enable**. This bit is used to enable the aging timer interrupt (triggered with AGTIM) 0 AGTIM interrupt disabled 1 AGTIM interrupt enabled
2 TXEN	**Transmitter Enable**. Enables/Disables the transmitter. When TXEN is negated the transmitter is disabled and idle. When the UARTEN and TXEN bits are set the transmitter is enabled. If TXEN is negated in the middle of a transmission, the UART disables the transmitter immediately, and starts marking 1s. The transmitter FIFO cannot be written when this bit is cleared. 0 Disable the transmitter 1 Enable the transmitter
1 RXEN	**Receiver Enable**. Enables/Disables the receiver. When the receiver is enabled, if the RXD input is already low, the receiver does not recognize BREAK characters, because it requires a valid 1-to-0 transition before it can accept any character. 0 Disable the receiver 1 Enable the receiver
0 SRST	**Software Reset**. Once the software writes 0 to SRST_B, the software reset remains active for 4 *module_clock* cycles before the hardware deasserts SRST_B. The software can only write 0 to SRST_B. Writing 1 to SRST_B is ignored. 0 Reset the transmit and receive state machines, all FIFOs and register USR1, USR2, UBIR, UBMR, UBRC , URXD, UTXD and UTS[6-3]. 1 No reset

64.15.5 UART Control Register 3 (UARTx_UCR3)

Address: Base address + 88h offset



UARTx_UCR3 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–14 DPEC	**DTR/DSR Interrupt Edge Control**. These bits control the edge of DTR_B (DCE) or DSR_B (DTE) on which an interrupt will be generated. An interrupt will only be generated if the DTREN bit is set. 00 interrupt generated on rising edge 01 interrupt generated on falling edge 1X interrupt generated on either edge
13 DTREN	**Data Terminal Ready Interrupt Enable**. When this bit is set, it will enable the status bit DTRF (USR2 [13]) (DTR/DSR edge sensitive interrupt) to cause an interrupt. 0 Data Terminal Ready Interrupt Disabled 1 Data Terminal Ready Interrupt Enabled
12 PARERREN	**Parity Error Interrupt Enable. Enables/Disables the interrupt. When asserted, PARERREN causes the PARITYERR bit to generate an interrupt.** 0 Disable the parity error interrupt 1 Enable the parity error interrupt
11 FRAERREN	**Frame Error Interrupt Enable. Enables/Disables the interrupt. When asserted, FRAERREN causes the FRAMERR bit to generate an interrupt.** 0 Disable the frame error interrupt 1 Enable the frame error interrupt
10 DSR	**Data Set Ready**. This bit is used by software to control the DSR/DTR module output for the modem interface. In DCE mode it applies to DSR_B and in DTE mode it applies to DTR_B. 0 DSR/ DTR pin is logic zero 1 DSR/ DTR pin is logic one

Table continues on the next page...

UARTx_UCR3 field descriptions (continued)

Field	Description
9 DCD	**Data Carrier Detect**. In DCE mode this bit is used by software to control the DCD_B module output for the modem interface. In DTE mode, when this bit is set, it will enable the status bit DCDDELT (USR2 (6)) to cause an interrupt. 0 DCD_B pin is logic zero (DCE mode) 1 DCD_B pin is logic one (DCE mode) 0 DCDDELT interrupt disabled (DTE mode) 1 DCDDELT interrupt enabled (DTE mode)
8 RI	**Ring Indicator**. In DCE mode this bit is used by software to control the RI_B module output for the modem interface. In DTE mode, when this bit is set, it will enable the status bit RIDELT (USR2 (10)) to cause an interrupt. 0 RI_B pin is logic zero (DCE mode) 1 RI_B pin is logic one (DCE mode) 0 RIDELT interrupt disabled (DTE mode) 1 RIDELT interrupt enabled (DTE mode)
7 ADNIMP	**Autobaud Detection Not Improved-. Disables new features of autobaud detection (See** Baud Rate Automatic Detection Protocol**, for more details).** 0 Autobaud detection new features selected 1 Keep old autobaud detection mechanism
6 RXDSEN	Receive Status Interrupt Enable. Controls the receive status interrupt (*interrupt_uart*). When this bit is enabled and RXDS status bit is set, the interrupt *interrupt_uart* will be generated. 0 Disable the RXDS interrupt 1 Enable the RXDS interrupt
5 AIRINTEN	Asynchronous IR WAKE Interrupt Enable. Controls the asynchronous IR WAKE interrupt. An interrupt is generated when AIRINTEN is asserted and a pulse is detected on the RXD pin. 0 Disable the AIRINT interrupt 1 Enable the AIRINT interrupt
4 AWAKEN	Asynchronous WAKE Interrupt Enable. Controls the asynchronous WAKE interrupt. An interrupt is generated when AWAKEN is asserted and a falling edge is detected on the RXD pin. 0 Disable the AWAKE interrupt 1 Enable the AWAKE interrupt
3 DTRDEN	**Data Terminal Ready Delta Enable**. Enables / Disables the asynchronous DTRD interrupt. When DTRDEN is asserted and an edge (rising or falling) is detected on DTR_B (in DCE mode) or on DSR_B (in DTE mode), then an interrupt is generated. 0 Disable DTRD interrupt 1 Enable DTRD interrupt
2 RXDMUXSEL	RXD Muxed Input Selected. Selects proper input pins for serial and Infrared input signal. **NOTE:** In this chip, UARTs are used in MUXED mode, so that this bit should always be set.
1 INVT	Invert TXD output in RS-232/RS-485 mode, set TXD active level in IrDA mode. In RS232/RS-485 mode(UMCR[0] = 1), if this bit is set to 1, the TXD output is inverted before transmitted. In **IrDA mode**, when INVT is cleared, the infrared logic block transmits a positive IR 3/16 pulse for all 0s and 0s are transmitted for 1s. When INVT is set (INVT = 1), the infrared logic block transmits an active low or negative infrared 3/16 pulse for all 0s and 1s are transmitted for 1s.

Table continues on the next page...

UARTx_UCR3 field descriptions (continued)

Field	Description
	0 **TXD is not inverted** 1 **TXD is inverted** 0 **TXD** Active low transmission 1 **TXD** Active high transmission
0 ACIEN	**Autobaud Counter Interrupt Enable.** This bit is used to enable the autobaud counter stopped interrupt (triggered with ACST (USR2[11]). 0 ACST interrupt disabled 1 ACST interrupt enabled

64.15.6 UART Control Register 4 (UARTx_UCR4)

Address: Base address + 8Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CTSTL						INVR	ENIRI	WKEN	IDDMAEN	IRSC	LPBYP	TCEN	BKEN	OREN	DREN
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UCR4 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–10 CTSTL	**CTS Trigger Level**. Controls the threshold at which the CTS_B pin is deasserted by the RxFIFO. After the trigger level is reached and the CTS_B pin is deasserted, the RxFIFO continues to receive data until it is full. The CTSTL bits are encoded as shown in the Settings column. Settings 0 to 32 are in use. All other settings are Reserved. 000000 0 characters received 000001 1 characters in the RxFIFO ... — ... — 100000 32 characters in the RxFIFO (maximum)
9 INVR	**Invert RXD input in RS-232/RS-485 Mode, d**etermine **RXD input** logic level being sampled in **In IrDA mode.**

Table continues on the next page...

UARTx_UCR4 field descriptions (continued)

Field	Description
	In RS232/RS-485 Mode(UMCR[0] = 1), if this bit is set to 1, the RXD input is inverted before sampled. **In IrDA mode**,when cleared, the infrared logic block expects an active low or negative IR 3/16 pulse for 0s and 1s are expected for 1s. When INVR is set (INVR 1), the infrared logic block expects an active high or positive IR 3/16 pulse for 0s and 0s are expected for 1s. 0 **RXD input is not inverted** 1 **RXD input is inverted** 0 **RXD** active low detection 1 **RXD** active high detection
8 ENIRI	**Serial Infrared Interrupt Enable**. Enables/Disables the serial infrared interrupt. 0 Serial infrared Interrupt disabled 1 Serial infrared Interrupt enabled
7 WKEN	**WAKE Interrupt Enable**. Enables/Disables the WAKE bit to generate an interrupt. The WAKE bit is set at the detection of a start bit by the receiver. 0 Disable the WAKE interrupt 1 Enable the WAKE interrupt
6 IDDMAEN	**DMA IDLE Condition Detected Interrupt Enable** Enables/Disables the receive DMA request *dma_req_rx* for the IDLE interrupt (triggered with IDLE flag in USR2[12]). 0 DMA IDLE interrupt disabled 1 DMA IDLE interrupt enabled
5 IRSC	**IR Special Case**. Selects the clock for the vote logic. When set, IRSC switches the vote logic clock from the sampling clock to the UART reference clock. The IR pulses are counted a predetermined amount of time depending on the reference frequency. See InfraRed Special Case (IRSC) Bit. 0 The vote logic uses the sampling clock (16x baud rate) for normal operation 1 The vote logic uses the UART reference clock
4 LPBYP	**Low Power Bypass**. Allows to bypass the low power new features in UART. To use during debug phase. 0 Low power features enabled 1 Low power features disabled
3 TCEN	**TransmitComplete Interrupt Enable**. Enables/Disables the TXDC bit to generate an interrupt (*interrupt_uart* = 0) **NOTE:** An interrupt will be issued as long as TCEN and TXDC are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the TXDC interrupt. 0 Disable TXDC interrupt 1 Enable TXDC interrupt
2 BKEN	**BREAK Condition Detected Interrupt Enable**. Enables/Disables the BRCD bit to generate an interrupt. 0 Disable the BRCD interrupt 1 Enable the BRCD interrupt
1 OREN	**Receiver Overrun Interrupt Enable**. Enables/Disables the ORE bit to generate an interrupt. 0 Disable ORE interrupt 1 Enable ORE interrupt

Table continues on the next page...

UARTx_UCR4 field descriptions (continued)

Field	Description
0 DREN	**Receive Data Ready Interrupt Enable**. Enables/Disables the RDR bit to generate an interrupt. 0 Disable RDR interrupt 1 Enable RDR interrupt

64.15.7 UART FIFO Control Register (UARTx_UFCR)

Address: Base address + 90h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TXTL						RFDIV			DCEDTE	RXTL					
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

UARTx_UFCR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–10 TXTL	**Transmitter Trigger Level**. Controls the threshold at which a maskable interrupt is generated by the TxFIFO. A maskable interrupt is generated whenever the data level in the TxFIFO falls below the selected threshold. The bits are encoded as shown in the Settings column. Settings 0 to 32 are in use. All other settings are Reserved. 000000 Reserved 000001 Reserved 000010 TxFIFO has 2 or fewer characters ... — ... — 011111 TxFIFO has 31 or fewer characters 100000 TxFIFO has 32 characters (maximum)
9–7 RFDIV	Reference Frequency Divider. Controls the divide ratio for the reference clock. The input clock is *module_clock*. The output from the divider is *ref_clk* which is used by BRM to create the 16x baud rate oversampling clock (*brm_clk*). 000 Divide input clock by 6 001 Divide input clock by 5

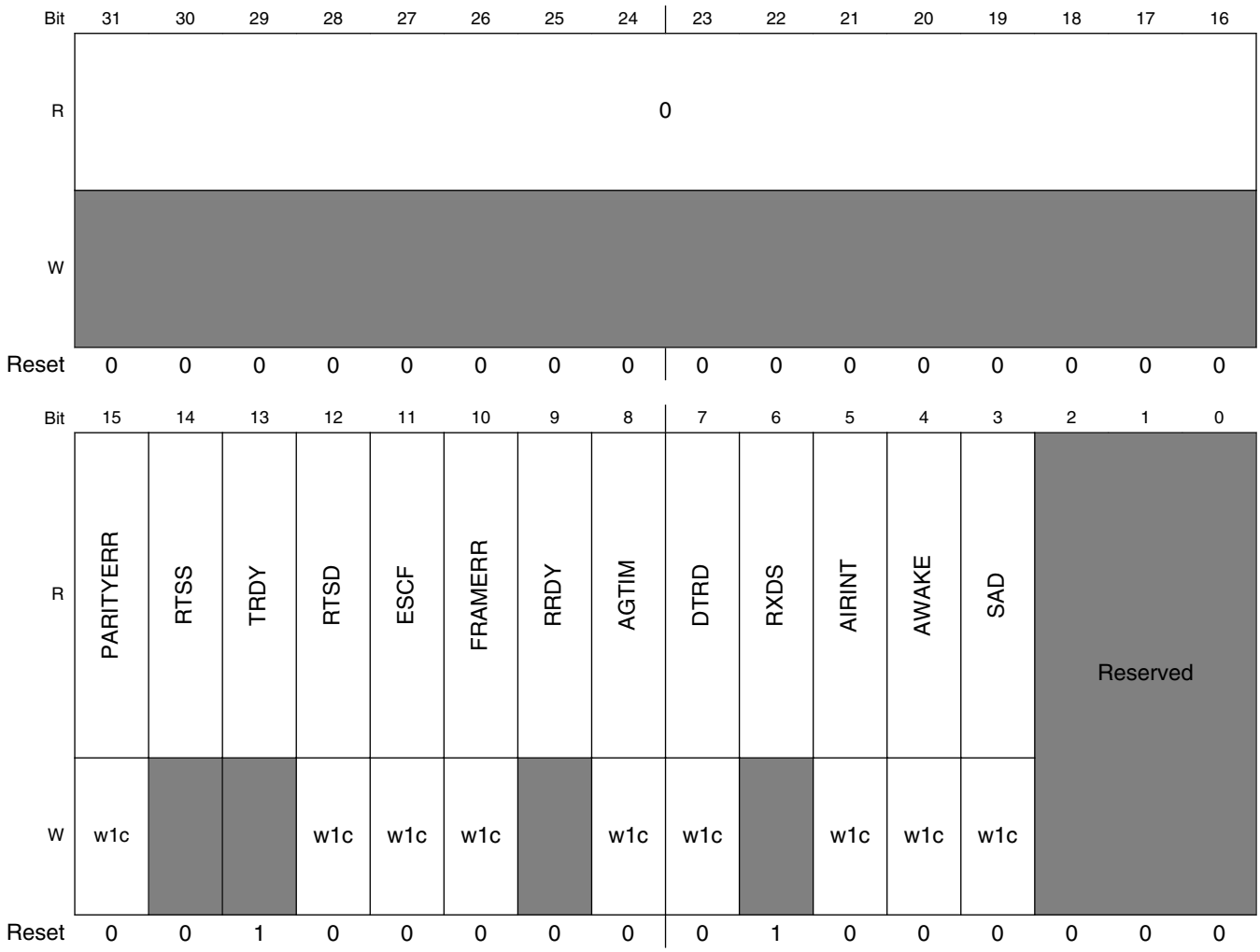
Table continues on the next page...

UARTx_UFCR field descriptions (continued)

Field	Description
	010 Divide input clock by 4 011 Divide input clock by 3 100 Divide input clock by 2 101 Divide input clock by 1 110 Divide input clock by 7 111 Reserved
6 DCEDTE	**DCE/DTE mode select**. Select UART as data communication equipment (DCE mode) or as data terminal equipment (DTE mode). 0 DCE mode selected 1 DTE mode selected
RXTL	**Receiver Trigger Level**. Controls the threshold at which a maskable interrupt is generated by the RxFIFO. A maskable interrupt is generated whenever the data level in the RxFIFO reaches the selected threshold. The RXTL bits are encoded as shown in the Settings column. Setting 0 to 32 are in use. All other settings are Reserved. 000000 0 characters received 000001 RxFIFO has 1 character ... — ... — 011111 RxFIFO has 31 characters 100000 RxFIFO has 32 characters (maximum)

64.15.8 UART Status Register 1 (UARTx_USR1)

Address: Base address + 94h offset



UARTx_USR1 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 PARITYERR	**Parity Error Interrupt Flag**. Indicates a parity error is detected. PARITYERR is cleared by writing 1 to it. Writing 0 to PARITYERR has no effect. When parity is disabled, PARITYERR always reads 0. At reset, PARITYERR is set to 0. 0 No parity error detected 1 Parity error detected (write 1 to clear)
14 RTSS	RTS_B **Pin Status**. Indicates the current status of the RTS_B pin. A "snapshot" of RTS_B is taken immediately before RTSS is presented to the data bus. RTSS cannot be cleared because all writes to RTSS are ignored. At reset, RTSS is set to 0.

Table continues on the next page...

UARTx_USR1 field descriptions (continued)

Field	Description
	0 The RTS_B module input is high (inactive) 1 The RTS_B module input is low (active)
13 TRDY	**Transmitter Ready Interrupt / DMA Flag**. Indicates that the TxFIFO emptied below its target threshold and requires data. TRDY is automatically cleared when the data level in the TxFIFO exceeds the threshold set by TXTL bits. At reset, TRDY is set to 1. 0 The transmitter does not require data 1 The transmitter requires data (interrupt posted)
12 RTSD	**RTS Delta.** Indicates whether the RTS_B pin changed state. It (RTSD) generates a maskable interrupt. When in STOP mode, RTS assertion sets RTSD and can be used to wake the processor. The current state of the RTS_B pin is available on the RTSS bit. Clear RTSD by writing 1 to it. Writing 0 to RTSD has no effect. At reset, RTSD is set to 0. 0 RTS_B pin did not change state since last cleared 1 RTS_B pin changed state (write 1 to clear)
11 ESCF	**Escape Sequence Interrupt Flag**. Indicates if an escape sequence was detected. ESCF is asserted when the ESCEN bit is set and an escape sequence is detected in the RxFIFO. Clear ESCF by writing 1 to it. Writing 0 to ESCF has no effect. 0 No escape sequence detected 1 Escape sequence detected (write 1 to clear).
10 FRAMERR	**Frame Error Interrupt Flag**. Indicates that a frame error is detected. The *interrupt_uart* interrupt will be generated if a frame error is detected and the interrupt is enabled. Clear FRAMERR by writing 1 to it. Writing 0 to FRAMERR has no effect. 0 No frame error detected 1 Frame error detected (write 1 to clear)
9 RRDY	**Receiver Ready Interrupt / DMA Flag**. Indicates that the RxFIFO data level is above the threshold set by the RXTL bits. (See the RXTL bits description in UART FIFO Control Register (UART_UFCR) for setting the interrupt threshold.) When asserted, RRDY generates a maskable interrupt or DMA request. RRDY is automatically cleared when data level in the RxFIFO goes below the set threshold level. At reset, RRDY is set to 0. 0 No character ready 1 Character(s) ready (interrupt posted)
8 AGTIM	Ageing Timer Interrupt Flag. Indicates that data in the RxFIFO has been idle for a time of 8 character lengths (where a character length consists of 7 or 8 bits, depending on the setting of the WS bit in UCR2, with the bit time corresponding to the baud rate setting) and FIFO data level is less than RxFIFO threshold level (RXTL in the UFCR). Clear by writing a 1 to it. 0 AGTIM is not active 1 AGTIM is active (write 1 to clear)
7 DTRD	**DTR Delta.** Indicates whether DTR_B (in DCE mode) or DSR_B (in DTE mode) pins changed state. DTRD generates a maskable interrupt if DTRDEN (UCR3[3]) is set. Clear DTRD by writing 1 to it. Writing 0 to DTRD has no effect. 0 DTR_B (DCE) or DSR_B (DTE) pin did not change state since last cleared 1 DTR_B (DCE) or DSR_B (DTE) pin changed state (write 1 to clear)
6 RXDS	Receiver IDLE Interrupt Flag. Indicates that the receiver state machine is in an IDLE state, the next state is IDLE, and the receive pin is high. RXDS is automatically cleared when a character is received. RXDS is active only when the receiver is enabled.

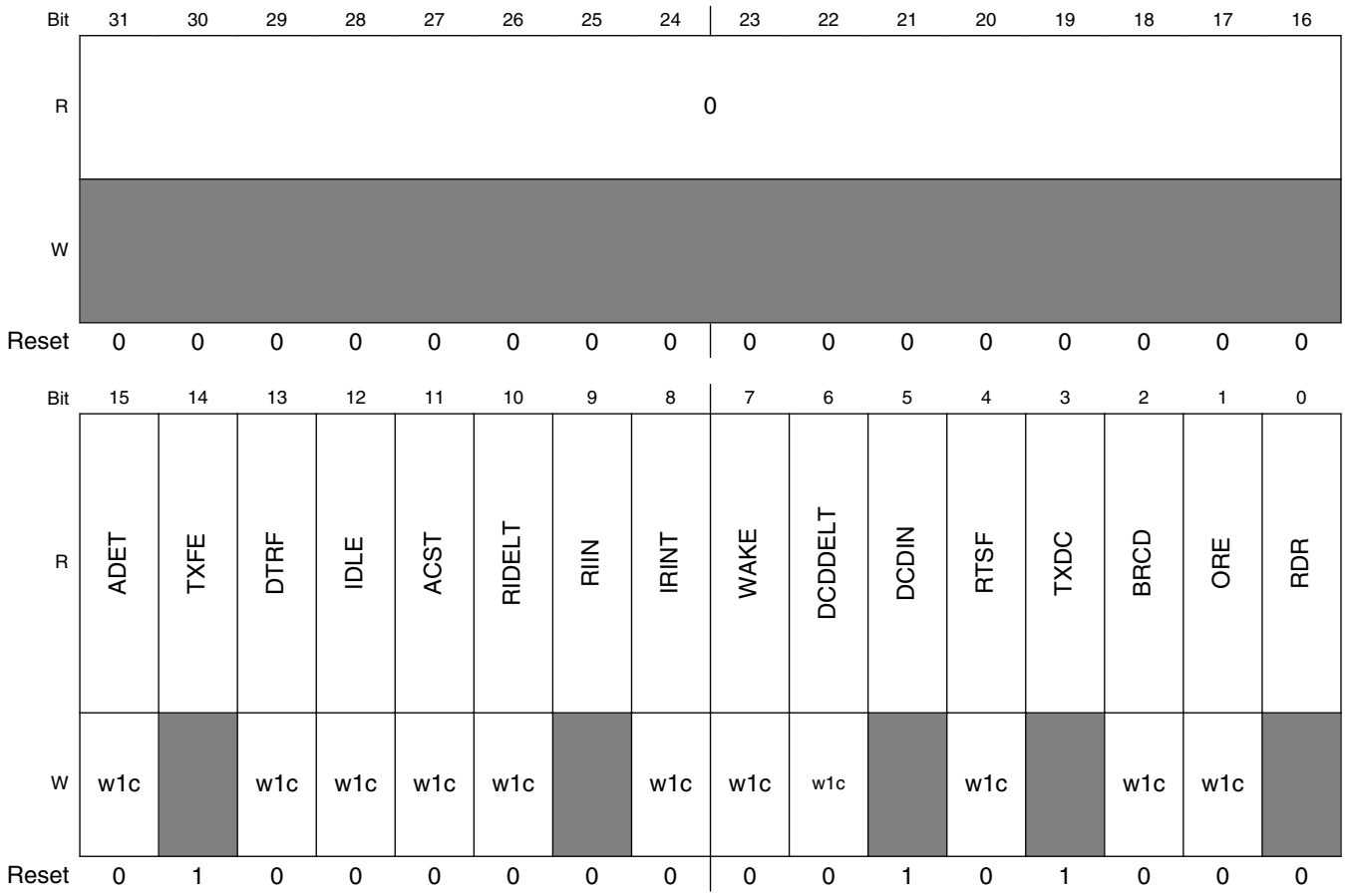
Table continues on the next page...

UARTx_USR1 field descriptions (continued)

Field	Description
	0 Receive in progress 1 Receiver is IDLE
5 AIRINT	Asynchronous IR WAKE Interrupt Flag. Indicates that the IR WAKE pulse was detected on the RXD pin. Clear AIRINT by writing 1 to it. Writing 0 to AIRINT has no effect. 0 No pulse was detected on the RXD IrDA pin 1 A pulse was detected on the RXD IrDA pin
4 AWAKE	Asynchronous WAKE Interrupt Flag. Indicates that a falling edge was detected on the RXD pin. Clear AWAKE by writing 1 to it. Writing 0 to AWAKE has no effect. 0 No falling edge was detected on the RXD Serial pin 1 A falling edge was detected on the RXD Serial pin
3 SAD	RS-485 Slave Address Detected Interrupt Flag. Indicates if RS-485 Slave Address was detected . SAD was asserted in RS-485 mode when the SADEN bit is set and Slave Address is detected in RxFIFO (in Nomal Address Detect Mode, the 9^{th} data bit = 1; in Automatic Address Detect Mode, the received charater matches the programmed SLADDR). 0 No slave address detected 1 Slave address detected
-	This field is reserved. Reserved

64.15.9 UART Status Register 2 (UARTx_USR2)

Address: Base address + 98h offset



UARTx_USR2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 ADET	**Automatic Baud Rate Detect Complete**. Indicates that an "A" or "a" was received and that the receiver detected and verified the incoming baud rate. Clear ADET by writing 1 to it. Writing 0 to ADET has no effect. 0 ASCII "A" or "a" was not received 1 ASCII "A" or "a" was received (write 1 to clear)
14 TXFE	**Transmit Buffer FIFO Empty**. Indicates that the transmit buffer (TxFIFO) is empty. TXFE is cleared automatically when data is written to the TxFIFO. Even though TXFE is high, the transmission might still be in progress. 0 The transmit buffer (TxFIFO) is not empty 1 The transmit buffer (TxFIFO) is empty

Table continues on the next page...

UARTx_USR2 field descriptions (continued)

Field	Description
13 DTRF	**DTR edge triggered interrupt flag**. This bit is asserted, when the programmed edge is detected on the DTR_B pin (DCE mode) or on DSR_B (DTE mode). This flag can cause an interrupt if DTREN (UCR3[13]) is enabled. 0 Programmed edge not detected on DTR/DSR 1 Programmed edge detected on DTR/DSR (write 1 to clear)
12 IDLE	**Idle Condition**. Indicates that an idle condition has existed for more than a programmed amount frame (see Idle Line Detect. An interrupt can be generated by this IDLE bit if IDEN (UCR1[12]) is enabled. IDLE is cleared by writing 1 to it. Writing 0 to IDLE has no effect. 0 No idle condition detected 1 Idle condition detected (write 1 to clear)
11 ACST	**Autobaud Counter Stopped**. In autobaud detection (ADBR=1), indicates the counter which determines the baud rate was running and is now stopped. This means either START bit is finished (if ADNIMP=1), or Bit 0 is finished (if ADNIMP=0). See New Autobaud Counter Stopped bit and Interrupt, for more details. An interrupt can be flagged on *interrupt_uart* if ACIEN=1. 0 Measurement of bit length not finished (in autobaud) 1 Measurement of bit length finished (in autobaud). (write 1 to clear)
10 RIDELT	**Ring Indicator Delta**. This bit is used in DTE mode to indicate that the Ring Indicator input (RI_B) has changed state. This flag can generate an interrupt if RI (UCR3[8]) is enabled. RIDELT is cleared by writing 1 to it. Writing 0 to RIDELT has no effect. 0 Ring Indicator input has not changed state 1 Ring Indicator input has changed state (write 1 to clear)
9 RIIN	**Ring Indicator Input**. This bit is used in DTE mode to reflect the status if the Ring Indicator input (RI_B). The Ring Indicator input is used to indicate that a ring has occurred. In DCE mode this bit is always zero. 0 Ring Detected 1 No Ring Detected
8 IRINT	**Serial Infrared Interrupt Flag.** When an edge is detected on the RXD pin during SIR Mode, this flag will be asserted. This flag can cause an interrupt which can be masked using the control bit ENIRI: UCR4 [8]. 0 no edge detected 1 valid edge detected (write 1 to clear)
7 WAKE	**Wake**. Indicates the start bit is detected. WAKE can generate an interrupt that can be masked using the WKEN bit. Clear WAKE by writing 1 to it. Writing 0 to WAKE has no effect. 0 start bit not detected 1 start bit detected (write 1 to clear)
6 DCDDELT	**Data Carrier Detect Delta**. This bit is used in DTE mode to indicate that the Data Carrier Detect input (DCD_B) has changed state. This flag can cause an interrupt if DCD (UCR3[9]) is enabled. When in STOP mode, this bit can be used to wake the processor. In DCE mode this bit is always zero. 0 Data Carrier Detect input has not changed state 1 Data Carrier Detect input has changed state (write 1 to clear)
5 DCDIN	**Data Carrier Detect Input**. This bit is used in DTE mode reflect the status of the Data Carrier Detect input (DCD_B). The Data Carrier Detect input is used to indicate that a carrier signal has been detected. In DCE mode this bit is always zero.

Table continues on the next page...

UARTx_USR2 field descriptions (continued)

Field	Description
	0 Carrier signal Detected 1 No Carrier signal Detected
4 RTSF	**RTS Edge Triggered Interrupt Flag. Indicates if** a programmed edge is detected on the RTS_B pin. The RTEC bits select the edge that generates an interrupt (see Table 64-9). RTSF can generate an interrupt that can be masked using the RTSEN bit. Clear RTSF by writing 1 to it. Writing 0 to RTSF has no effect. 0 Programmed edge not detected on RTS_B 1 Programmed edge detected on RTS_B (write 1 to clear)
3 TXDC	**Transmitter Complete**. Indicates that the transmit buffer (TxFIFO) and Shift Register is empty; therefore the transmission is complete. TXDC is cleared automatically when data is written to the TxFIFO. 0 Transmit is incomplete 1 Transmit is complete
2 BRCD	**BREAK Condition Detected**. Indicates that a BREAK condition was detected by the receiver. Clear BRCD by writing 1 to it. Writing 0 to BRCD has no effect. 0 No BREAK condition was detected 1 A BREAK condition was detected (write 1 to clear)
1 ORE	**Overrun Error**. When set to 1, ORE indicates that the receive buffer (RxFIFO) was full (32 chars inside), and a 33rd character has been fully received. This 33rd character has been discarded. Clear ORE by writing 1 to it. Writing 0 to ORE has no effect. 0 No overrun error 1 Overrun error (write 1 to clear)
0 RDR	**Receive Data Ready**-Indicates that at least 1 character is received and written to the RxFIFO. If the URXD register is read and there is only 1 character in the RxFIFO, RDR is automatically cleared. 0 No receive data ready 1 Receive data ready

64.15.10 UART Escape Character Register (UARTx_UESC)

Address: Base address + 9Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																ESC_CHAR															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1

UARTx_UESC field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.
ESC_CHAR	**UART Escape Character**. Holds the selected escape character that all received characters are compared against to detect an escape sequence.

64.15.11 UART Escape Timer Register (UARTx_UTIM)

Address: Base address + A0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TIM															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UTIM field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value 0.
TIM	UART Escape Timer. Holds the maximum time interval (in ms) allowed between escape characters. The escape timer register is programmable in intervals of 2 ms. See Escape Sequence Detection and Table 64-14 for more information on the UART escape sequence detection. Reset value 0x000 = 2 ms up to 0xFFF = 8.192 s.

64.15.12 UART BRM Incremental Register (UARTx_UBIR)

This register can be written by both software and hardware. When enabling the automatic baud rate detection feature hardware can write 0x000F value into the UBIR after finishing detecting baud rate. Hardware has higher priority when both software and hardware try to write it at the same cycle[3].

Please note software reset will reset the register to its reset value.

Address: Base address + A4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																INC															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UBIR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
INC	Incremental Numerator. Holds the numerator value minus one of the BRM ratio (see Binary Rate Multiplier (BRM)). The UBIR register MUST be updated before the UBMR register for the baud rate to be updated correctly. If only one register is written to by software, the BRM will ignore this data until the other register is written to by software. Updating this field using byte accesses is not recommended and is undefined.

3. Note: The write priority in the new design is not same as the original UART. In the orginal design, software has higher priority than hardware when writing this register at the same time.

64.15.13 UART BRM Modulator Register (UARTx_UBMR)

This register can be written by both software and hardware. When enabling the automatic baud rate detection feature hardware can write a proper value into the UBMR based on detected baud rate. Hardware has higher priority when both software and hardware try to write it at the same cycle[4].

Please note software reset will reset the register to its reset value.

Address: Base address + A8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																MOD															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UBMR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
MOD	**Modulator Denominator.** Holds the value of the denominator minus one of the BRM ratio (see Binary Rate Multiplier (BRM)). The UBIR register MUST be updated before the UBMR register for the baud rate to be updated correctly. If only one register is written to by software, the BRM will ignore this data until the other register is written to by software. Updating this register using byte accesses is not recommended and undefined.

64.15.14 UART Baud Rate Count Register (UARTx_UBRC)

Address: Base address + ACh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																BCNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

UARTx_UBRC field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
BCNT	**Baud Rate Count Register.** This read only register is used to count the start bit of the incoming baud rate (if ADNIMP=1), or start bit + bit0 (if ADNIMP=0). When the measurement is done, the Baud Rate Count Register contains the number of UART internal clock cycles (clock after divider) present in an incoming bit. BCNT retains its value until the next Automatic Baud Rate Detection sequence has been initiated. The 16 bit Baud Rate Count register is reset to 4 and stays at hex FFFF in the case of an overflow.

4. Note: The write priority in the new design is not same as the original UART. In the orginal design, software has higher priority than hardware when writing this register at the same time.

64.15.15 UART One Millisecond Register (UARTx_ONEMS)

NOTE

This register has been expanded from 16 bits to 24 bits. In previous versions, the 16-bit ONEMS can only support the maximum 65.535MHz (0xFFFFx1000) *ref_clk*. To support 4Mbps Bluetooth application with 66.5MHz *module_clock*, the value 0x103C4 (66.5M/1000) should be written into this register. In this case, the 16 bits are not enough to contain the 0x103C4. So this register was expanded to 24 bits to support high frequency of the *ref_clk*.

Address: Base address + B0h offset

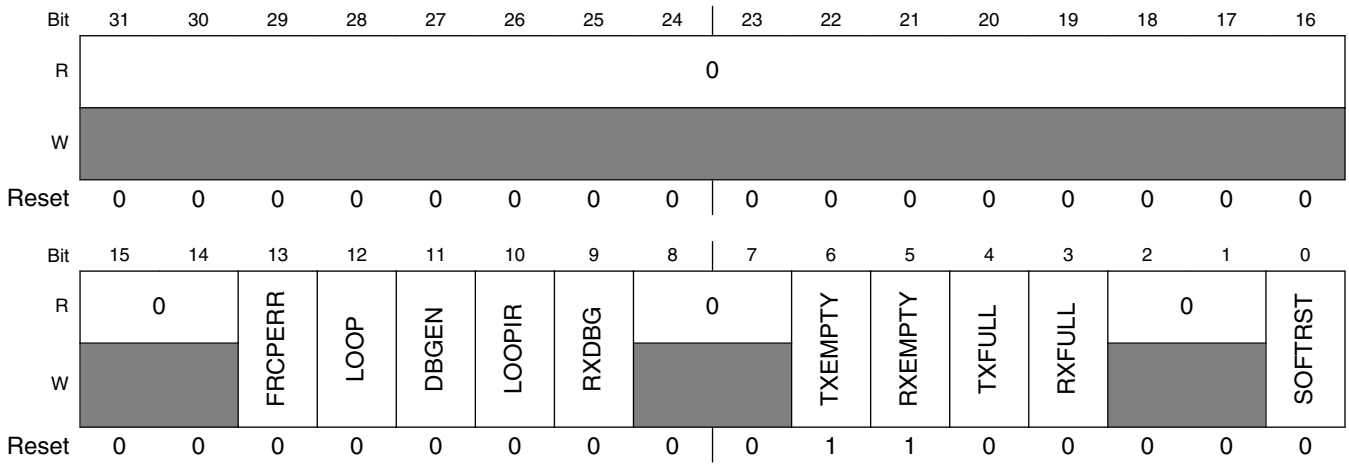
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ONEMS																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_ONEMS field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value 0.
ONEMS	**One Millisecond Register.** This 24-bit register must contain the value of the UART internal frequency (*ref_clk* in Figure 64-1) divided by 1000. The internal frequency is obtained after the UART BRM internal divider (F (*ref_clk*) = F(*module_clock*) / RFDIV). In fact this register contains the value corresponding to the number of UART BRM internal clock cycles present in one millisecond. The ONEMS (and UTIM) registers value are used in the escape character detection feature (Escape Sequence Detection) to count the number of clock cycles left between two escape characters. The ONEMS register is also used in infrared special case mode (IRSC = UCR4[5] = 1'b1), see InfraRed Special Case (IRSC) Bit.

64.15.16 UART Test Register (UARTx_UTS)

Address: Base address + B4h offset



UARTx_UTS field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
13 FRCPERR	Force Parity Error. Forces the transmitter to generate a parity error if parity is enabled. FRCPERR is provided for system debugging. 0 Generate normal parity 1 Generate inverted parity (error)
12 LOOP	Loop TX and RX for Test. Controls loopback for test purposes. When LOOP is high, the receiver input is internally connected to the transmitter and ignores the RXD pin. The transmitter is unaffected by LOOP. If RXDMUXSEL (UCR3[2]) is set to 1, the loopback is applied on serial and IrDA signals. If RXDMUXSEL is set to 0, the loopback is only applied on serial signals. 0 Normal receiver operation 1 Internally connect the transmitter output to the receiver input
11 DBGEN	debug_enable_B**.** This bit controls whether to respond to the *debug_req* input signal. 0 UART will go into debug mode when *debug_req* is HIGH 1 UART will not go into debug mode even if *debug_req* is HIGH
10 LOOPIR	**Loop TX and RX for IR Test (LOOPIR)**. This bit controls loopback from transmitter to receiver in the InfraRed interface. 0 No IR loop 1 Connect IR transmitter to IR receiver
9 RXDBG	**RX_fifo_debug_mode.** This bit controls the operation of the RX fifo read counter when in debug mode. 0 rx fifo read pointer does not increment 1 rx_fifo read pointer increments as normal

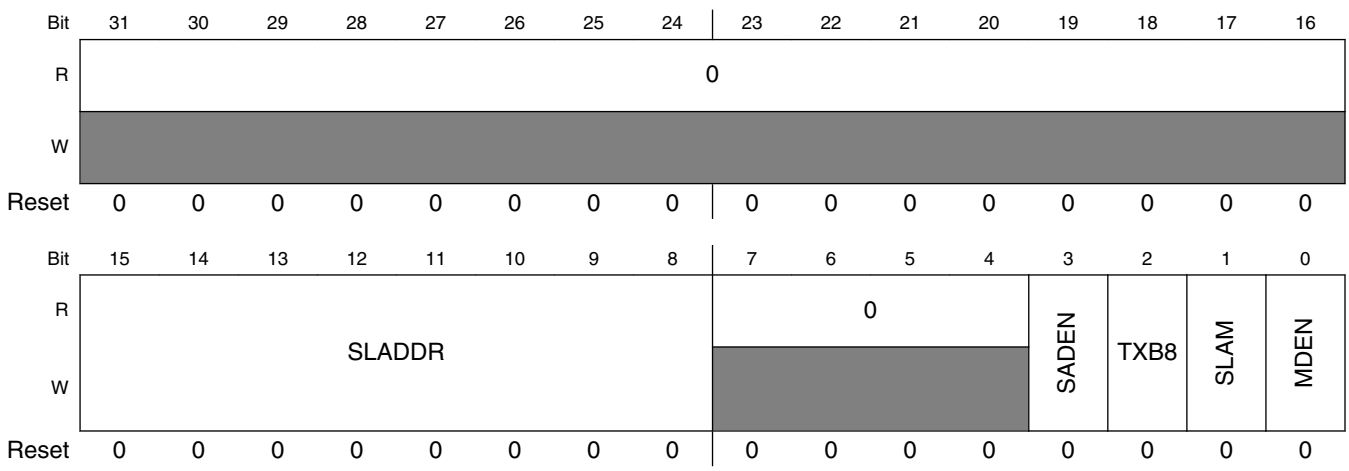
Table continues on the next page...

UARTx_UTS field descriptions (continued)

Field	Description
8–7 Reserved	This read-only field is reserved and always has the value 0.
6 TXEMPTY	TxFIFO Empty. Indicates that the TxFIFO is empty. 0 The TxFIFO is not empty 1 The TxFIFO is empty
5 RXEMPTY	RxFIFO Empty. Indicates the RxFIFO is empty. 0 The RxFIFO is not empty 1 The RxFIFO is empty
4 TXFULL	TxFIFO FULL. Indicates the TxFIFO is full. 0 The TxFIFO is not full 1 The TxFIFO is full
3 RXFULL	RxFIFO FULL. Indicates the RxFIFO is full. 0 The RxFIFO is not full 1 The RxFIFO is full
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SOFTRST	Software Reset. Indicates the status of the software reset (SRST_B bit of UCR2). 0 Software reset inactive 1 Software reset active

64.15.17 UART RS-485 Mode Control Register (UARTx_UMCR)

Address: Base address + B8h offset



UARTx_UMCR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 SLADDR	RS-485 Slave Address Character. Holds the selected slave adress character that the receiver wil try to detect.
7–4 Reserved	This read-only field is reserved and always has the value 0.
3 SADEN	RS-485 Slave Address Detected Interrupt Enable. 0 Disable RS-485 Slave Address Detected Interrupt 1 Enable RS-485 Slave Address Detected Interrupt
2 TXB8	Transmit RS-485 bit 8 (the ninth bit or 9th bit). In RS-485 mode, software writes TXB8 bit as the 9th data bit to be transmitted. 0 0 will be transmitted as the RS485 9th data bit 1 1 will be transmitted as the RS485 9th data bit
1 SLAM	RS-485 Slave Address Detect Mode Selection. 0 Select Normal Address Detect mode 1 Select Automatic Address Detect mode
0 MDEN	9-bit data or Multidrop Mode (RS-485) Enable. 0 Normal RS-232 or IrDA mode, see Table 64-1 for detail. 1 Enable RS-485 mode, see Table 64-1 for detail

Chapter 65
Universal Serial Bus Controller (USB)

65.1 Overview

The USB controller block provides high performance USB functionality that conforms to the *Universal Serial Bus Specification*, Rev. 2.0 (Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips; 2000), and the *On-The-Go and Embedded Host Supplement to the USB Revision* 2.0 Specification (Hewlett-Packard Company, Intel Corporation, LSI Corporation, Microsoft Corporation, Renesas Electronics Corporation, ST-Ericsson; 2012).

The USB controller consists of four independent USB controller cores: one On-The-Go (OTG) controller core, and three host-only controller cores. Each controller core can support UTMI or HSIC interfaces according to its feature. All four controller cores are single-port cores. For the OTG core, there is only one port, which can be used as either a downstream or an upstream port. For the host-only core, there is also only one port, which is used as a downstream port. See Features for more details.

The following figure is a block diagram of USB.

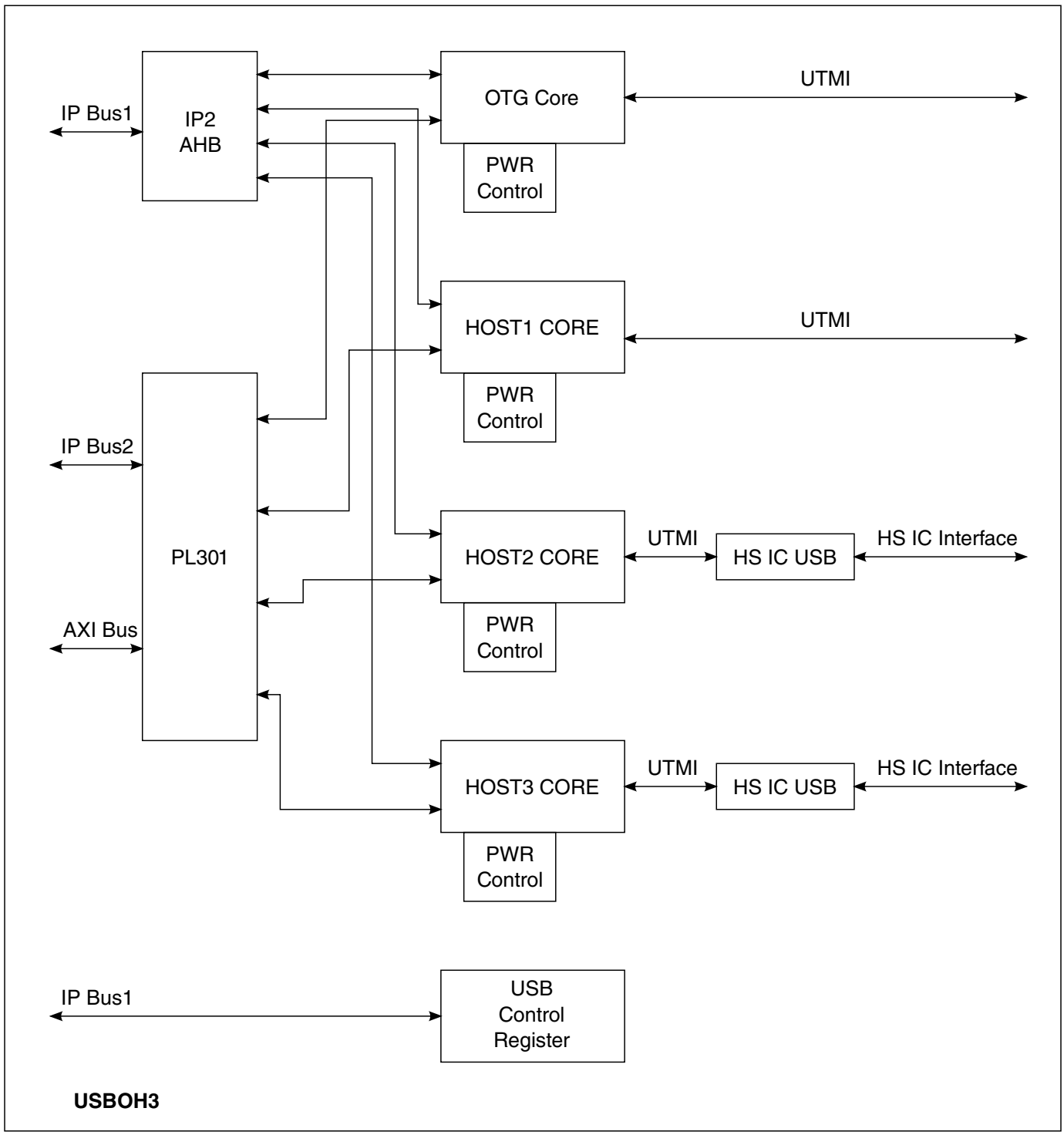


Figure 65-1. USB block diagram

65.1.1 Features

There are four USB 2.0 controller cores in this chip:

- Controller Core 0 is also named 'OTG Core'; its connected port is named 'OTG port'.
- Controller Core 1 is also named 'Host1 Core'; its connected port is named 'Host1 port'.
- Controller Core 2 is also named 'Host2 Core'; its connected port is named 'Host2 port'.
- Controller Core 3 is also named 'Host3 Core'; its connected port is named 'Host3 port'.

The following list provides features of each controller core.

- USB 2.0 Controller Core 0
  - High-Speed/Full-Speed/Low-Speed OTG core
  - HS/FS/LS UTMI compliant interface
  - High Speed, Full Speed and Low Speed operation in HOST mode (with UTMI transceiver)
  - High Speed, and Full Speed operation in Peripheral mode (with UTMI transceiver)
  - Hardware support for OTG signaling, session request protocol, and host negotiation protocol
  - Up to 8 bidirectional endpoints
  - Support charger detection CHD_B pin and register interface
- USB 2.0 Controller Core 1
  - High-Speed/Full-Speed/Low-Speed Host-Only core
  - HS/FS/LS UTMI compliant interface
  - High Speed, Full Speed and Low Speed operation in Host mode (with UTMI transceiver)
  - High Speed, and Full Speed operation in Peripheral mode (with UTMI transceiver)
  - Hardware support for OTG signaling, session request protocol, and host negotiation protocol
  - Up to 8 bidirectional endpoints
  - Supports charger detection with register interface only
- USB 2.0 Controller Core 2
  - High-Speed/Full-Speed/Low-Speed Host-Only core
  - High Speed Inter-Chip USB compliant interface (HSIC)
- USB 2.0 Controller Core 3
  - High-Speed/Full-Speed/Low-Speed Host-Only core
  - High Speed Inter-Chip USB compliant interface (HSIC)
- Low-power mode with local and remote wake-up capability

- Serial PHY interfaces configurable for bidirectional/unidirectional and differential/ single ended
- Embedded DMA controller in each core

65.1.2 Modes of Operation

The USB has two main modes of operation: normal mode and low power mode.

Each USB OTG controller core can operate in High Speed operation (480 Mbps), Full Speed operation (12 Mbps) and Low Speed operation (1.5 Mbps).

This chapter explains the operation modes.

65.1.2.1 Normal Mode

The OTG controller core can operate in Host mode and Device (Peripheral) mode. The Host-only controller core can operate in Host mode only.

Each USB controller core has its corresponding port, which can work in one or more interface modes.

NOTE

Each controller supports only the interface type listed below. Selecting a different interface type in the PORTSC_PTS field results in unpredictable behavior and may cause the system to hang.

- OTG port
  - This port supports on-chip UTMI transceivers only.
- Host1 Port
  - This port supports on-chip UTMI transceivers only.
- Host2 Port
  - This port supports HSIC interfaces only. Interface for onboard HSIC compatible USB peripherals.
- Host3 Port
  - This port supports HSIC interfaces only. Interface for onboard HSIC compatible USB peripherals.

NOTE

HSIC is an inter-chip interface that is optimized for circuit board layouts.

65.1.2.2 Low-Power Mode

Each USB controller core has a low-power mode (Suspend mode) to save power consumption.

As described in the USB 2.0 specification, the device can go into the Suspend state after it sees a constant Idle state on the upstream facing port. The OTG controller core enters Suspend mode after 3 ms of inactivity on the port when it is in Device Operation mode. Host controllers, including the OTG controller in Host mode, do not suspend automatically but can be placed in Suspend mode by software.

Either the local Arm platform or the remote USB Host/Peripheral can initiate a wake-up sequence to resume USB communication. For details about Suspend/Resume, see USB Power Control.

65.2 External Signals

The table found here describes the external signals of USB.

Table 65-1. USB External Signals

Signal	Description	Pad	Mode	Direction
USB_H1_DN	DN Host 1 Signal	USB_H1_DN	No muxing	IO
USB_H1_DP	DP Host 1 Signal	USB_H1_DP	No muxing	IO
USB_H1_OC	Host 1 External input for VBUS overcurrent detection	EIM_D30	ALT6	I
		GPIO_3	ALT6	
USB_H1_PWR	To control PMIC to supply VBUS voltage	EIM_D31	ALT6	O
		GPIO_0	ALT6	
USB_H2_DATA	Data signal	RGMII_TXC	ALT0	IO
USB_H2_STROBE	Strobe signal	RGMII_TX_CTL	ALT0	IO
USB_H3_DATA	Data signal	RGMII_RX_CTL	ALT0	IO
USB_H3_STROBE	Strobe signal	RGMII_RXC	ALT0	IO
USB_OTG_CHD_B	Charge detect signal	USB_OTG_CHD_B	No muxing	IO
USB_OTG_DN	DN OTG Signal	USB_OTG_DN	No muxing	IO
USB_OTG_DP	DP OTG Signal	USB_OTG_DP	No muxing	IO
USB_OTG_ID	ID signal	ENET_RX_ER	ALT0	I
		GPIO_1	ALT3	
USB_OTG_OC	OTG External input for VBUS overcurrent detection	EIM_D21	ALT4	I
		KEY_COL4	ALT2	
USB_OTG_PWR	To control PMIC to supply VBUS voltage	EIM_D22	ALT4	O
		KEY_ROW4	ALT2	

65.3 Functional Description

These sections describe the functionality of the various building blocks of the USB.

65.3.1 USB 2.0 Controller Core 0

The USB 2.0 Controller 0 is an instantiation of an EHCI-compatible core which supports high-, full-, and low-speed operation.

In Host mode, this controller core supports high-, full-, and low-speed operation. In Device mode, it supports high- and full-speed operation.

65.3.1.1 Host Mode

The controller supports direct connection of .

Although there is no separate Transaction Translator block in the system, the transaction translator function normally associated with a USB 2.0 high speed hub has been implemented within the DMA and protocol engine blocks to support connection to full and low speed devices.

65.3.1.2 Peripheral (Device) Mode

- Up to eight bidirectional endpoints
- High/full-speed operation
- Support of HNP, and SRP
- Remote wake-up capability

65.3.1.3 Pins Used for OTG Controller

The required power supplies:

- 1.1-V supply from LDO 1P1 regulator
- 2.5-V supply from LDO 2P5 regulator
- 3.0-V supply from USB LDO regulator

USB OTG PHY pins:

- USB_OTG_VBUS
- USB_OTG_DN
- USB_OTG_DP
- USB_OTG_CHD_B, see Universal Serial Bus 2.0 Integrated PHY (USB-PHY).

The following external signals are multiplexed with other pins. For the pin mapping, see External Signals. For the IOMUXC register setting, see IOMUX Controller (IOMUXC)

- USB_OTG_ID
- USB_OTG_PWR
- USB_OTG_OC

65.3.2 USB 2.0 Controller Core 1

USB 2.0 Controller Core 1 is an instantiation of EHCI-compatible core which supports High Speed / Full Speed / Low Speed operation

65.3.2.1 Pins Used for Host Controller 1

The required power supplies:

- 1.1V voltage supply from LDO 1P1 regulator
- 2.5V voltage supply from LDO 2P5 regulator
- 3.0V voltage supply from USB LDO regulator

USB Host 1 PHY pins:

- USB_H1_VBUS
- USB_H1_DN
- USB_H1_DP

The following external signals are multiplexed with other pins. For the pin mapping, see External Signals. For the IOMUXC register setting, see IOMUX Controller (IOMUXC)

- USB_H1_PWR
- USB_H1_OC

65.3.3 USB 2.0 Controller Core 2

USB 2.0 Controller Core 2 is an instantiation of the EHCI-compatible core which supports High Speed / Full Speed / Low Speed operation.

This USB core's signals connect directly to I/O pins (HSIC interface).

65.3.4 USB 2.0 Controller Core 3

Host Controller 3 is an instantiation of EHCI-compatible core which supports High Speed / Full Speed / Low Speed operation.

This USB core's signals connect directly to I/O pins (HSIC interface).

65.3.5 USB Power Control

The USB controller supports suspend and wake-up functionality.

The power control block allows for placing the transceiver in USB low power mode when USB bus is IDLE, and supports local and remote wake-up to bring the transceiver out of USB low power mode when needed. Additionally, the power control block can wake-up the Arm platform from core sleep mode by generating an interrupt.

65.3.5.1 Entering Low Power Suspend Mode

In Host operation mode, low power suspend mode is entered as follows:

1. Clear the ASE and PSE bits in USB_USBCMD, and wait until the AS and PS bits in USB_USBSTS become "0".
2. Set the "SUSPEND" bit in USB_PORTSC1
3. Set the "PHCD" bit in USB_PORTSC1
4. Set all PWD bits in USBPHYx_PWD
5. Set CLKGATE in USBPHYx_CTRL

NOTE

Step 3,4,5 shall be done in atomic operation. That is, interrupt should be disabled during these three steps.

For device operation mode, low power suspend mode is entered as follows:

1. After Host drive is IDLE for 3ms, an SLI interrupt is issued (the "DCSUSPEND" or "SLI" bit in USB_USBSTS)
2. Set the "PHCD" bit on USB_PORTSC1
3. Set all PWD bits in USBPHYx_PWD
4. Set CLKGATE in USBPHYx_CTRL

NOTE

Step 2,3,4 shall be done in atomic operation. That is, interrupt should be disabled during these three steps.

65.3.5.2 Wake-Up Events

The power control block monitors the USB bus when the USB core is in the USB suspend state.

Depending on whether the core is on Host or Device mode, a number of wake-up conditions are monitored. Upon detection of a wake-up condition, an interrupt will be generated to Arm platform if the related wake-up interrupt enable bit is set.

USB wake-up interrupt also re-activates the Arm platform clocks if they were stopped during the suspend.

65.3.5.2.1 Host Mode Events

The host controller wakes up on the following events:

- Remote Wake-up Request

A peripheral can request the host to reactivate the bus by driving wake-up signaling on the DM/DP lines. The power control block sends a wake-up request to the USB core when a J-K transition on DM/DP line is detected.

- Wake-Up On Overcurrent

If Wake-Up On Overcurrent is enabled (WKOC bit in the USB core register PORTSC1 is set '1'), the power control block sends a wake-up request to the USB core when an overcurrent event is detected.

- Wake-Up On Disconnect

If Wake-Up On Disconnect is enabled (WKDC bit in the USB core register PORTSC1 is set '1'), the power control block sends a wake-up request to the USB core when a disconnection event is detected (J-SE0/K-SE0 transition on DM/DP line).

- Wake-Up On Connect

If a Wake-Up On Connect is enabled (WKCN bit in the USB core register PORTSC1 is set '1'), the power control sub-block sends a wake-up request to the USB core when the connection event is detected (SE0-J/SE0-K transition on DM/DP line).

For a detailed description of register bits WKOC, WKDC, WKCN, please see Port Status & Control (USB_*n*PORTSC1).

65.3.6 Interrupts

65.3.6.1 USB Core Interrupts

Each USB core uses one dedicated vector in the Interrupt Table. The vector numbers associated with each of the cores can be found in the Interrupt section.

With the exception of the wake-up interrupts, all of the interrupt sources are controlled in the USB Cores. Refer to the Interrupt Enable Register (USB_*n*USBINTR) for details.

65.3.6.2 USB Wake-Up Interrupts

Each USB Core has an associated wake-up interrupt. The wake-up interrupts are generated outside of the USB controller cores, but using the same vector as the corresponding USB controller cores interrupt.

These interrupts are generated by the Power Control blocks which run on the 32 KHz standby clock. The wake-up interrupt is designed to work even when the USB and Arm platform clocks are disabled, such that a wake-up condition on the USB bus can re-activate the Arm platform clocks.

Because the wake-up interrupt is generated and cleared on a 32 KHz clock, this interrupt request responds very slowly to clear actions. For this reason, the software must disable the wake-up interrupt to clear the request flag. Disabling the interrupt masks the request instantaneously as this is clocked by the Arm platform clock. The software should wait for at least three 32 KHz clock cycles before re-enabling this interrupt to allow sufficient time for the request flag to clear. Because this interrupt is only used during low power modes of the USB, it is sufficient to enable the wake-up interrupt just prior to entering the USB suspend mode.

65.4 USB Operation Model

This section describes the detailed application knowledge for Host1, Host2, Host3 and OTG ports. It can be generally divided in two parts, one is for Host and the other is for Device. Host port applies to all host ports, and to OTG port when operating in Host mode. Device part only applies to OTG port when operating in Device mode.

65.4.1 Register Interface

Configuration, control and status registers are divided into three categories, identification, capability and operational registers.

NOTE

USB controller registers support only DWORD (32-bit) access.

- Identification registers are used to declare the slave interface presence along with the complete set of the hardware configuration parameters.
- Static, read only capability registers define the software limits, restrictions, and capabilities of the host/device controller.
- Operational registers are dynamic control or status registers that may be read only, read/write, or read/write-to-clear. The following sections define the use of these registers.

EHCI registers are listed alongside device registers to show the complementary nature of host and device control.

The following table describes the Interface register sets.

Table 65-2. Interface Register Sets

Offset	Register Set	Explanation
000h-07Ch	Identification Registers	Identification registers are used to declare the slave interface presence and include a table of the hardware configuration parameters.
100h-124h	Capability Registers	Capability registers specify the limits, restrictions, and capabilities of a host/device controller implementation. These values are used as parameters to the host/device controller driver.
080h-0FCh 140h-1FCh	Operational Registers	Operational registers are used by the system software to control and monitor the operational state of the host/device controller.

65.4.1.1 Configuration, Control and Status Register Set

The following table describes the Device/Host capability registers.

NOTE

Depending on implementation, "*x*" can have the following values: UOG, UH1, UH2 or UH3.

Table 65-3. Device/Host Capability Registers

Offset	Size (Bytes)	Mnemonic	Register Name	Device Mode	Host Mode
000h	4	USB_*x*_ID	Identification Register	O	O
004h	4	USB_*x*_HWGENERAL	General Hardware Parameters	O	O
008h	4	USB_*x*_HWHOST	Host Hardware Parameters	X	O
00Ch	4	USB_*x*_HWDEVICE	Device Hardware Parameters	O	X
010h	4	USB_*x*_HWTXBUF	TX Buffer Hardware Parameters	O	O
014h	4	USB_*x*_HWRXBUF	RX Buffer Hardware Parameters	O	O
018-07Fh		-	Reserved		
080h	4	USB_*x*_GPTIMER0LD	General Purpose Timer #0 Load Register	O	O
084h	4	USB_*x*_GPTIMER0CTRL	General Purpose Timer #0 Control Register	O	O
088h	4	USB_*x*_GPTIMER1LD	General Purpose Timer #1 Load Register	O	O
08Ch	4	USB_*x*_GPTIMER1CTRL	General Purpose Timer #1 Control Register	O	O
090h	4	USB_*x*_SBUSCFG	System Bus Interface Configuration Register	O	O
094-09Fh		-	Reserved		
100h	1	USB_*x*_CAPLENGTH	Capability Register Length	O	O
101h		-	Reserved		
102h	2	USB_*x*_HCIVERSION	Host Controller Interface Version Number	X	O
104h	4	USB_*x*_HCSPARAMS	Host Controller Structural Parameters	X	O
108h	4	USB_*x*_HCCPARAMS	Host Controller Capability Parameters	X	O
10C-11Fh		-	Reserved		
120h	2	USB_*x*_DCIVERSION	Device Controller Interface Version Number	O	X
122h	2	-	Reserved		
124h	4	USB_*x*_DCCPARAMS	Device Controller Capability Parameters	O	X
128-13Fh		-	Reserved		
140h	4	USB_*x*_USBCMD	USB Command Register	O	O
144h	4	USB_*x*_USBSTS	USB Status Register	O	O
148h	4	USB_*x*_USBINTR	USB Interrupt Enable Register	O	O
14Ch	4	USB_*x*_FRINDEX	USB Frame Index	O	O
150h	4	-	Reserved		
154h	4	USB_*x*_PERIODICLISTBASE	Frame List Base Address	X	O
		USB_*x*_DEVICEADDR	USB Device Address	O	X
158h	4	USB_*x*_ASYNCLISTADDR	Next Asynchronous List Address	X	O
	4	USB_*x*_ENDPOINTLISTADDR	Address at Endpoint list in memory	O	X
15Ch	4	-	Reserved		
160h	4	USB_*x*_BURSTSIZE	Programmable Burst Size	O	O
164h	4	USB_*x*_TXFILLTUNING	Host Transmit Pre-Buffer Packet Tuning	X	O

Table continues on the next page...

Table 65-3. Device/Host Capability Registers (continued)

Offset	Size (Bytes)	Mnemonic	Register Name	Device Mode	Host Mode
168h	4	-	Reserved		
16Ch	4	USB_*x*_IC_USB	IC_USB enable and voltage negotiation	O	O
170h	4	-	Reserved		
178h	4	USB_*x*_ENDPTNAK	Endpoint NAK register	O	X
17Ch	4	USB_*x*_ENDPTNAKEN	Endpoint NAK Enable register	O	X
180h	4	USB_*x*_CONFIGFLAG	Configured Flag Register	X	O
184h	4	USB_*x*_PORTSC1	Port Status/Control Register 1	O	O
188-1A3h		-	Reserved		
1A4h	4	USB_*x*_OTGSC	On-The-Go Status/Control Register (OTG only)	O	O
1A8h	4	USB_*x*_USBMODE	USB Controller Operating Mode	O	O
1ACh	4	USB_*x*_ENDPTSETUPSTAT	Endpoint Setup Status	O	X
1B0h	4	USB_*x*_ENDPTPRIME	Endpoint Initialization	O	X
1B4h	4	USB_*x*_ENDPTFLUSH	Endpoint De-Initialization	O	X
1B8h	4	USB_*x*_ENDPTSTATUS	Endpoint Status	O	X
1BCh	4	USB_*x*_ENDPTCOMPLETE	Endpoint Complete	O	X
1C0 1C4 ... 1DCh	64	USB_*x*_ENDPTCTRL0 USB_*x*_ENDPTCTRL1 .... USB_*x*_ENDPTCTRL7	Endpoint Control Register 0-7	O	X

NOTE

"O" means the register is available in host/device operation mode;

"X" means the register is reserved in host/device operation mode

65.4.1.2 Identification Registers

Identification registers are used to declare the slave interface presence and include a table of the hardware configuration parameters.

65.4.1.3 OTG Operations

65.4.1.3.1 Register Bits

In the previous section, the Register interface has behaviors described for device mode and behaviors described for host mode. However, for OTG operations it is necessary to perform tasks independent of the controller mode.

NOTE

The only way to transit the controller mode out of host or device mode is with the controller reset bit. Therefore, it is also necessary for the OTG tasks to be performed independent of a controller reset as well as independent of the controller mode.

The following figure shows the controller mode.

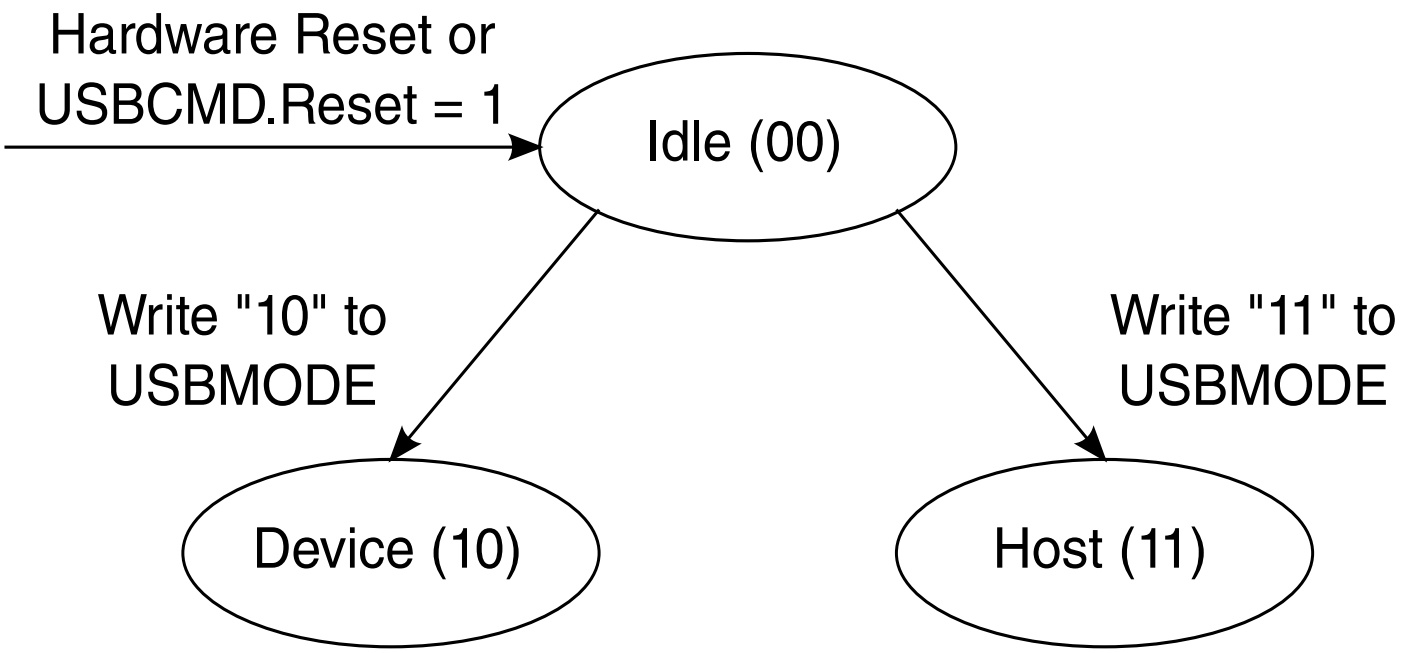


Figure 65-2. Controller Mode

To this end, listed below are the register bits that are used for OTG operations, which are independent of the controller mode and are also not affected by a write to the reset bit in the USBCMD register:

All Identification Registers

All Device/Host Capability Registers

OTGSC: All bits

PORTSC1:

Physical Interface Select

Physical Interface Serial Select

Physical Interface Data Width

Physical Interface Low Power

Physical Interface Wake Signals

Port Indicators

Port Power

65.4.2 Host Data Structures

This section defines the interface data structures used to communicate control, status, and data between HCD (software) and the Enhanced Host Controller (hardware).

The data structure definitions in this chapter support a 32-bit memory buffer address space. The interface consists of a Periodic Schedule, Periodic Frame List, Asynchronous Schedule, Isochronous Transaction Descriptors, Split-transaction Isochronous Transfer Descriptors, Queue Heads, and Queue Element Transfer Descriptors.

The periodic frame list is the root of all periodic (isochronous and interrupt transfer type) transfers for the host controller. The asynchronous list is the root for all the bulk and control transfers. Isochronous data streams are managed using Isochronous Transaction Descriptors. Isochronous split-transaction data streams are managed with Split-transaction Isochronous Transfer Descriptors. All Interrupt, Control, and Bulk data streams are managed via queue heads and Queue Element Transfer Descriptors. These data structures are optimized to reduce the total memory footprint of the schedule and to reduce (on average) the number of memory accesses needed to execute a USB transaction.

Note that software must ensure that no interface data structure reachable by the EHCI host controller spans a 4 K-page boundary.

The data structures defined in this section are (from the host controller's perspective) a mix of read-only and read/writeable fields. The host controller must preserve the read-only fields on all data structure writes.

65.4.2.1 Periodic Frame List

This schedule is for all periodic transfers (isochronous and interrupt). The periodic schedule is referenced from the operational registers space using the USB_PERIODICLISTBASE address register and the USB_FRINDEX register.

The periodic schedule is based on an array of pointers called the Periodic Frame List.

The USB_PERIODICLISTBASE address register is combined with the USB_FRINDEX register to produce a memory pointer into the frame list. The Periodic Frame List implements a sliding window of work over time.

The following figure shows the organization of periodic schedule.

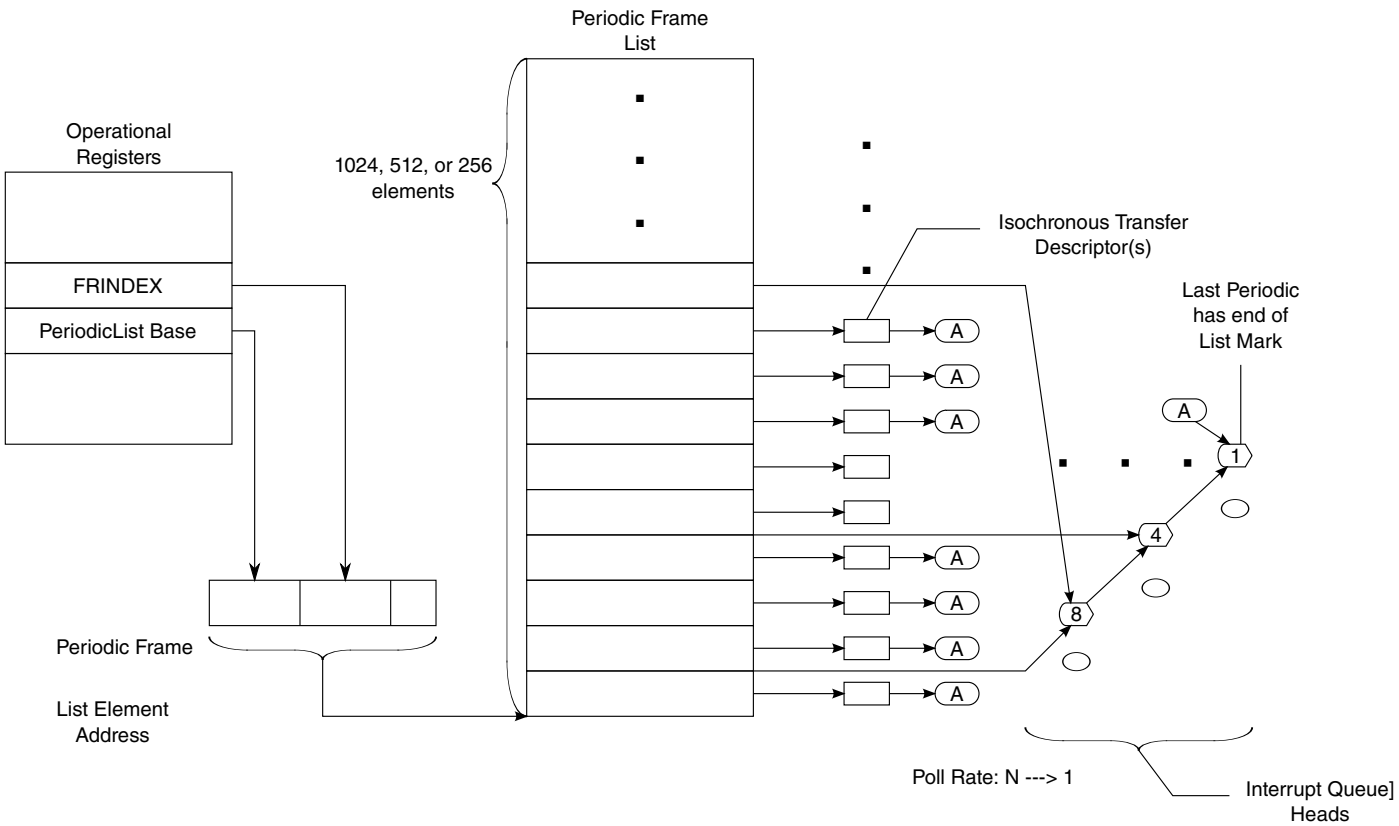


Figure 65-3. Periodic Schedule Organization

Split transaction Interrupt, Bulk and Control are also managed using queue heads and queue element transfer descriptors.

The periodic frame list is a 4 K-page aligned array of Frame List Link pointers. The length of the frame list may be programmable. The programmability of the periodic frame list is exported to system software via the USB_HCCPARAMS register. If non-programmable, the length is 1024 elements. If programmable, the length can be selected by system software as one of 256, 512, or 1024 elements. An implementation must support all three sizes. Programming the size (that is, the number of elements) is accomplished by system software writing the appropriate value into Frame List Size field in the USB_USBCMD register.

Frame List Link pointers direct the host controller to the first work item in the frame's periodic schedule for the current micro-frame. The link pointers are aligned on DWord boundaries within the Frame List.

The table below illustrates the format of the Frame list element pointer.

Table 65-4. Format of Frame List Element Pointer

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Frame List Link Pointer																											0		Typ			03-00H

Frame List Link pointers always reference memory objects that are 32-byte aligned. The referenced object may be an isochronous transfer descriptor for high-speed devices, a split-transaction isochronous transfer descriptor (for full-speed isochronous endpoints), or a queue head (used to support high-, full- and low-speed interrupt). System software should not place non-periodic schedule items into the periodic schedule. The least significant bits in a frame list pointer are used to key the host controller as to the type of object the pointer is referencing.

The least significant bit is the T-Bit (bit 0). When this bit is set to a one, the host controller never uses the value of the frame list pointer as a physical memory pointer. The Typ field is used to indicate the exact type of data structure being referenced by this pointer. The value encodings are.

Table 65-5. Typ Field Value Definitions

Value	Meaning
00b	Isochronous Transfer Descriptor
01b	Queue Head
10b	Split Transaction Isochronous Transfer Descriptor.
11b	Frame Span Traversal Node.

65.4.2.2 Asynchronous List Queue Head Pointer

The Asynchronous Transfer List (based at the USB_ASYNCLISTADDR register) is where all of the control and bulk transfers are managed.

Host controllers use this list only when it reaches the end of the periodic list, the periodic list is disabled, or the periodic list is empty.

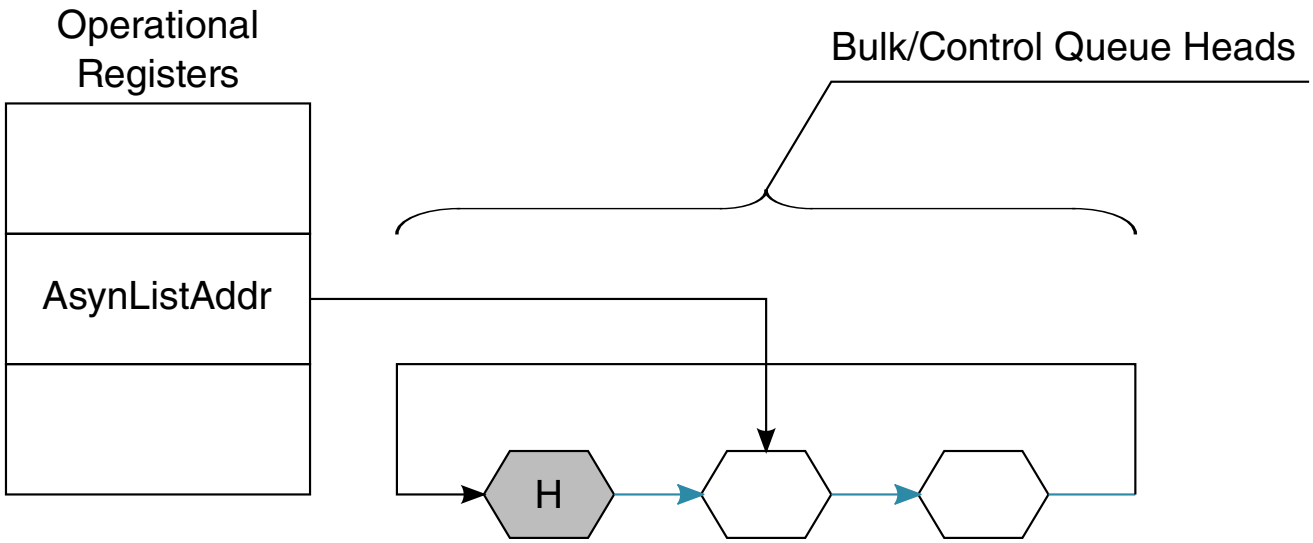


Figure 65-4. Asynchronous Schedule Organization

The Asynchronous list is a simple circular list of queue heads. The USB_ASYNCLISTADDR register is simply a pointer to the next queue head. This implements a pure round-robin service for all queue heads linked into the asynchronous list.

65.4.2.3 Isochronous (High-Speed) Transfer Descriptor (iTD)

The format of an isochronous transfer descriptor is shown in the table below.

This structure is used only for high-speed isochronous endpoints. All other transfer types should use queue structures. Isochronous TDs must be aligned on a 32-byte boundary.

Table 65-6. Isochronous Transaction Descriptor (iTD)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Next Link Pointer																											0		Typ		T	03-00 H
Status				Transaction 0 Length												IOC	PG*																07-04 H
Status				Transaction 1 Length												IOC	PG*																0B-08H
Status				Transaction 2 Length												IOC	PG*																0F-0CH
Status				Transaction 3 Length												IOC	PG*																13-10 H

Table continues on the next page...

Table 65-6. Isochronous Transaction Descriptor (iTD) (continued)

Status	Transaction 4 Length			IOC	PG*	Transaction 4 Offset*				17-14H
Status	Transaction 5 Length			IOC	PG*	Transaction 5 Offset*				1B-18H
Status	Transaction 6 Length			IOC	PG*	Transaction 6 Offset*				1F-1CH
Status	Transaction 7 Length			IOC	PG*	Transaction 7 Offset*				23-20H
Buffer Pointer (Page 0)						EndPt	R	Device Address		27-24H
Buffer Pointer (Page 1)						I/O	Maximum Packet Size			2B-28H
Buffer Pointer (Page 2)						-			Mult	2F-2CH
Buffer Pointer (Page 3)						-				33-30H
Buffer Pointer (Page 4)						-				37-34H
Buffer Pointer (Page 5)						-				3B-38H
Buffer Pointer (Page 6)						-				3F-3CH

Host Controller Read/Write

Host Controller Read Only

These fields may be modified by the host controller if the I/O field indicates an OUT.

65.4.2.3.1 Next Link Pointer

The first DWord of an iTD is a pointer to the next schedule data structure.

The following table describes the Next Schedule Element pointer field.

Table 65-7. Next Schedule Element Pointer

Bit	Description
31-5 Link Pointer (LP)	These bits correspond to memory address signals [31:5], respectively. This field points to another Isochronous Transaction Descriptor (iTD/siTD) or Queue Head (QH).
4-3	These bits are reserved and their value has no effect on operation. Software should initialize this field to zero.

Table continues on the next page...

Table 65-7. Next Schedule Element Pointer (continued)

Reserved	
2-1 QH/(s)iTD Select (Typ)	This field indicates to the Host Controller whether the item referenced is an iTD, siTD or a QH. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor 11b FSTN (frame span traversal node)
0 Terminate (T)	1= Link Pointer field is not valid. 0= Link Pointer field is valid.

65.4.2.3.2 iTD Transaction Status and Control List

DWords 1 through 8 are eight slots of transaction control and status.

Each transaction description includes:

- Status results field
- Transaction length (bytes to send for OUT transactions and bytes received for IN transactions).
- Buffer offset. The PG and Transaction X Offset fields are used with the buffer pointer list to construct the starting buffer address for the transaction.

The host controller uses the information in each transaction description plus the endpoint information contained in the first three DWords of the Buffer Page Pointer list, to execute a transaction on the USB.

The following table describes iTD Transaction Status and Control fields.

Table 65-8. iTD Transaction Status and Control

Bit	Description	
31-28 Status	This field records the status of the transaction executed by the host controller for this slot. This field is a bit vector with the following encoding:	
	Bit	Definition
	31	Active. Set to one by software to enable the execution of an isochronous transaction by the Host Controller. When the transaction associated with this descriptor is completed, the Host Controller sets this bit to zero indicating that a transaction for this element should not be executed when it is next encountered in the schedule.
	30	Data Buffer Error. Set to a one by the Host Controller during status update to indicate that the Host Controller is unable to keep up with the reception of incoming data (overrun) or is unable to supply data fast enough during transmission (under run). If an overrun condition occurs, no action is necessary.
	29	Babble Detected. Set to one by the Host Controller during status update when" babble" is detected during the transaction generated by this descriptor.

Table continues on the next page...

Table 65-8. iTD Transaction Status and Control (continued)

Bit	Description	
	28	Transaction Error (XactErr). Set to one by the Host Controller during status update in the case where the host did not receive a valid response from the device (Timeout, CRC, Bad PID, etc.). This bit may only be set for isochronous IN transactions.
27-16 Transaction X Length	For an OUT, this field is the number of data bytes the host controller sends during the transaction. The host controller is not required to update this field to reflect the actual number of bytes transferred during the transfer. For an IN, the initial value of the endpoint to deliver. During the status update, the host controller writes back the field is the number of bytes the host expects the number of bytes successfully received. The value in this register is the actual byte count (0‡zero length data, 1‡one byte, 2‡two bytes, etc.). The maximum value this field may contain is 0xC00 (3072).	
15 Interrupt On Complete (IOC)	If this bit is set to one, it specifies that when this transaction completes, the Host Controller should issue an interrupt at the next interrupt threshold.	
14-12 Page Select (PG)	These bits are set by software to indicate which of the buffer page pointers the offset field in this slot should be concatenated to produce the starting memory address for this transaction. The valid range of values for this field is 0 to 6.	
11-0 Transaction X Offset	This field is a value that is an offset, expressed in bytes, from the beginning of a buffer. This field is concatenated onto the buffer page pointer indicated in the adjacent *PG* field to produce the starting buffer address for this transaction.	

65.4.2.3.3 iTD Buffer Page Pointer List (Plus)

DWords 9-15 of an isochronous transaction descriptor are nominally page pointers (4 K aligned) to the data buffer for this transfer descriptor. This data structure requires the associated data buffer to be contiguous (relative to virtual memory), but allows the physical memory pages to be non-contiguous.

Seven page pointers are provided to support the expression of eight isochronous transfers. The seven pointers allow for 3 (transactions) * 1024 (maximum packet size) * 8 (transaction records) (24576 bytes) to be moved with this data structure, regardless of the alignment offset of the first page.

Because each pointer is a 4 K aligned page pointer, the least significant 12 bits in several of the page pointers are used for other purposes.

The tables below illustrate the field descriptions.

Table 65-9. iTD Buffer Pointer Page 0 (Plus)

Bit	Description
31-12 Buffer Pointer (Page 0)	This is a 4 K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11-8 Endpoint Number (Endpt)	This 4-bit field selects the particular endpoint number on the device serving as the data source or sink.

Table continues on the next page...

Table 65-9. iTD Buffer Pointer Page 0 (Plus) (continued)

Bit	Description
7 Reserved	Bit reserved for future use and should be initialized by software to zero.
6-0 Device Address	This field selects the specific device serving as the data source or sink.

Table 65-10. iTD Buffer Pointer Page 1 (Plus)

Bit	Description
31-12 Buffer Pointer (Page 1)	This is a 4K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11 Direction (I/O)	0 = OUT; 1 = IN. This field encodes whether the high-speed transaction should use an IN or OUT PID.
10-0 Maximum Packet Size	This directly corresponds to the maximum packet size of the associated endpoint *(wMaxPacketSize)*. This field is used for high-bandwidth endpoints where more than one transaction is issued per transaction description (per micro-frame). This field is used with the *Multi* field to support high-bandwidth pipes. This field is also used for all IN transfers to detect packet babble. Software should not set a value larger than 1024 (400h). Any value larger yields undefined results.

Table 65-11. iTD Buffer Pointer Page 2 (Plus)

Bit	Description
31-12 Buffer Pointer	This is a 4K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11-2 Reserved	This bit reserved for future use and should be set to zero.
1-0 Multi	This field is used to indicate to the host controller the number of transactions that should be executed per transaction description (per micro-frame). The valid values are: Value Meaning 00b Reserved. A zero in this field yields undefined results. 01b One transaction to be issued for this endpoint per micro- frame. 10b Two transactions to be issued for this endpoint per micro- frame. 11b Three transactions to be issued for this endpoint per micro- frame.

Table 65-12. iTD Buffer Pointer Page 3-6

Bit	Description
31-12 Buffer Pointer	This is a 4 K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11-0 Reserved	These bits reserved for future use and should be set to zero.

65.4.2.4 Split Transaction Isochronous Transfer Descriptor (siTD)

All Full-speed isochronous transfers through the internal transaction translator are managed using the siTD data structure. This data structure satisfies the operational requirements for managing the split transaction protocol.

The following table shows the Split Transaction Isochronous Transfer Descriptor (siTD).

Table 65-13. Split Transaction Isochronous Transfer Descriptor

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Addr
Next Link Pointer																											0		Typ		T	03-00
I/O	Port Number							-	Hub Addr							Reserved				EndPt				-	Device Address							07-04[1]
Reserved																µFrame C-mask								µFrame S-mask								0B-08[1]
ioc	P	Reserved				Total Bytes to Transfer										µFrame C-prog-mask								Status								0F-0C[2]
Buffer Pointer (Page 0)																				Current Offset												13-10[2]
Buffer Pointer (Page 1)																				Reserved							TP		T-count			17-14[2]
Back Pointer																											0				T	1B-18

1. 04-0B: Static Endpoint State
2. 0C-13: Transfer results

Host Controller Read/Write

Host Controller Read Only

65.4.2.4.1 Next Link Pointer

DWord0 of a siTD is a pointer to the next schedule data structure.

The following table describes the Next Link Pointer fields.

Table 65-14. Next Link Pointer

Bit	Description
31-5	Next Link Pointer (LP). This field contains the address of the next data object to be processed in the periodic list and corresponds to memory address signals [31:5], respectively.

Table continues on the next page...

Table 65-14. Next Link Pointer (continued)

Bit	Description
4-3	Reserved. These bits must be written as zeros.
2-1	QH/(s)iTD Select (Typ). This field indicates to the Host Controller whether the item referenced is an iTD/siTD or a QH. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor 11b FSTN (frame span traversal node)
0	Terminate (T). 1 = Link Pointer field is not valid. 0 = Link Pointer is valid.

65.4.2.4.2 siTD Endpoint Capabilities/Characteristics

DWords 1 and 2 specify static information about the full-speed endpoint, the addressing of the parent Companion Controller, and micro-frame scheduling control.

The tables below describe the Endpoint and transaction translator characteristics and micro-frame schedule control fields.

Table 65-15. Endpoint and Transaction Translator Characteristics

Bit	Description
31	Direction (I/O).0 = OUT; 1 = IN. This field encodes whether the full-speed transaction should be an IN or OUT.
30-24	Port Number. This field is the port number of the recipient Transaction Translator.
23	Reserved. Bit reserved and should be set to zero.
22-16	Hub Address. This field holds the device address of the Companion Controllers' hub.
15-12	Reserved. Field reserved and should be set to zero.
11-8	Endpoint Number (Endpt). This 4-bit field selects the particular endpoint number on the device serving as the data source or sink.
7	Reserved. Bit is reserved for future use. It should be set to zero.
6-0	Device Address. This field selects the specific device serving as the data source or sink.

Table 65-16. Micro-frame Schedule Control

Bit	Description
31-16	Reserved. This field reserved for future use. It should be set to zero.
15-8	Split Completion Mask (mFrame C-Mask). This field (along with the *Active* and *SplitX- state* fields in the *Status* byte) is used to determine during which micro-frames the host controller should execute complete-split transactions. When the criteria for using this field is met, an all zeros value has undefined behavior. The host

Table continues on the next page...

Table 65-16. Micro-frame Schedule Control (continued)

Bit	Description
	controller uses the value of the three low-order bits of the FRINDEX register to index into this bit field. If the FRINDEX register value indexes to a position where the m*Frame C-Mask* field is a one, then this siTD is a candidate for transaction execution. There may be more than one bit in this mask set.
7-0	Split Start Mask (mFrame S-mask). This field (along with the *Active* and *SplitX-state* fields in the *Status* byte) is used to determine during which micro-frames the host controller should execute start-split transactions. The host controller uses the value of the three low-order bits of the FRINDEX register to index into this bit field. If the FRINDEX register value indexes to a position where the m*Frame S-mask* field is a one, then this siTD is a candidate for transaction execution. An all zeros value in this field, in combination with existing periodic frame list has undefined results.

65.4.2.4.3 siTD Transfer State

DWords 3-6 are used to manage the state of the transfer.

The following table describes siTD transfer state fields.

Table 65-17. siTD Transfer Status and Control

Bit	Description
31	Interrupt On Complete (ioc). 0 = Do not interrupt when transaction is complete. 1 = Do interrupt when transaction is complete. When the host controller determines that the split transaction has completed it asserts a hardware interrupt at the next interrupt threshold.
30	Page Select (P). Used to indicate which data page pointer should be concatenated with the *CurrentOffset* field to construct a data buffer pointer (0 selects *Page 0* pointer and 1 selects *Page 1*). The host controller is not required to write this field back when the siTD is retired (*Active* bit transitioned from a one to a zero).
29-26	Reserved. This field reserved for future use and should be set to zero.
25-16	Total Bytes To Transfer. This field is initialized by software to the total number of bytes expected in this transfer. Maximum value is 1023 (3FFh)
15-8	µFrame Complete-split Progress Mask (C-prog-Mask). This field is used by the host controller to record which split-completes has been executed.
7-0: Status—This field records the status of the transaction executed by the host controller for this slot. It is a bit vector with the encoding shown in the following rows.	
7	Active. Set to one by software to enable the execution of an isochronous split transaction by the Host Controller.
6	ERR. Set to a one by the Host Controller when an ERR response is received from the Companion Controller.
5	Data Buffer Error. Set to a one by the Host Controller during status update to indicate that the Host Controller is unable to keep up with the reception of incoming data (overrun) or is unable to supply data fast enough during transmission (under run). In the case of an under run, the Host Controller transmits an incorrect CRC (thus invalidating the data at the endpoint). If an overrun condition occurs, no action is necessary.
4	Babble Detected. Set to a one by the Host Controller during status update when" babble" is detected during the transaction generated by this descriptor.
3	Transaction Error (XactErr). Set to a one by the Host Controller during status update in the case where the host did not receive a valid response from the device (Timeout, CRC, Bad PID, etc.). This bit is set only for IN transactions.
2	Missed Micro-Frame. The host controller detected that a host-induced hold- off caused the host controller to miss a required complete-split transaction.

Table continues on the next page...

Table 65-17. siTD Transfer Status and Control (continued)

Bit	Description
1	Split Transaction State (SplitXstate). The bit encodings are: Value Meaning 00b Do Start Split. This value directs the host controller to issue a Start split transaction to the endpoint when a match is encountered in the S-mask. 01b Do Complete Split. This value directs the host controller to issue a Complete split transaction to the endpoint when a match is encountered in the C-mask.
0	Reserved. Bit reserved for future use and should be set to zero.

65.4.2.4.4 siTD Buffer Pointer List (plus)

DWords 4 and 5 are the data buffer page pointers for the transfer. This structure supports one physical page cross. The most significant 20 bits of each DWord in this section are the 4 K (page) aligned buffer pointers.

The least significant 12 bits of each DWord are used as additional transfer state. The following table describes the siTD buffer pointer fields.

Table 65-18. Buffer Page Pointer List (plus)

Bit	Description
31-12	Buffer Pointer List. Bits [31:12] of DWords 4 and 5 are 4 K page aligned physical memory addresses. These bits correspond to physical address bits [31:12] respectively. The lower 12 bits in each pointer are defined and used as specified below. The field *P* (see siTD Transfer State) specifies the *current* active pointer.
Bits 11-0 (Page 0)	Current Offset—The 12 least significant bits of the Page 0 pointer are the current byte offset for the current page pointer (as selected with the page indicator bit (*P* field). The host controller is not required to write this field back when the siTD is retired (*Active* bit transitioned from a one to a zero).
Bits 11-0 (Page 1)—The least significant bits of the Page 1 pointer are split into three subfields as shown in the following rows.	
11-5 (Page 1)	Reserved
4-3 (Page 1)	Transaction position (TP). This field is used with T-count to determine whether to send *all*, *first*, *middle*, or *last* with each outbound transaction payload. System software must initialize this field with the appropriate starting value. The host controller must correctly manage this state during the lifetime of the transfer. The bit encodings are: Value Meaning 00b All. The entire full-speed transaction data payload is in this transaction (that is, less than or equal to 188 bytes). 01b Begin. This is the first data payload for a full- speed that is greater than 188 bytes. 10B Mid. This is the *middle* payload for a full-speed OUT transaction that is larger than 188 bytes.

Table continues on the next page...

Table 65-18. Buffer Page Pointer List (plus) (continued)

Bit	Description
	11b End. This is the *last* payload for a full-speed OUT transaction that was larger than 188 bytes.
2-0 (Page 1)	Transaction count (T-Count). Software initializes this field with the number of OUT start-splits this transfer requires. Any value larger than 6 is undefined.

65.4.2.4.5 siTD Back Link Pointer

DWord 6 of a siTD is simply another schedule link pointer. This pointer is always zero, or references a siTD, and it cannot reference any other schedule data structure.

The following table describes the siTD back link pointer fields.

Table 65-19. siTD Back Link Pointer

Bit	Description
31-5	siTD Back Pointer. This field is a physical memory pointer to a siTD.
4-1	Reserved. This field is reserved for future use. It should be set to zero.
0	Terminate (T). 1 = siTD Back Pointer field is not valid. 0 = siTD Back Pointer field is valid.

65.4.2.5 Queue element transfer descriptor (qTD)

This data structure is only used with a queue head. It describes one or more USB transactions to transfer up to 20480 (5*4096) bytes.

The structure contains two structure pointers used for queue advancement, a DWord of transfer state, and a five-element array of data buffer pointers.

It is 32 bytes and must be physically contiguous.

The buffer associated with this transfer must be virtually contiguous. The buffer may start on any byte boundary; however, for optimal utilization of on-chip busses it is recommended to align the buffers on a 32-byte boundary. A separate buffer pointer list element must be used for each physical page in the buffer, regardless of whether the buffer is physically contiguous.

Host controller updates (host controller writes) to stand-alone qTDs only occur during transfer retirement. References in the following bit field definitions of updates to the qTD are to the qTD portion of a queue head.

The following table shows the queue element transfer descriptor data structure.

Table 65-20. Queue element transfer descriptor data structure

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Addr
Next qTD Pointer																											0				T	03-00
Alternate Next qTD Pointer																											0				T	07-04
dt	Total Bytes to Transfer															ioc	C_Page			Cerr		PID Code		Status								0B-08[1]
Buffer Pointer (page 0)																				Current Offset												0F-0C[1]
Buffer Pointer (page 1)																				Reserved												13-10
Buffer Pointer (page 2)																				Reserved												17-14
Buffer Pointer (page 3)																				Reserved												1B-18
Buffer Pointer (page 4)																				Reserved												1F-1C

1. 08-0F: Transfer Results

Host Controller Read/Write

Host Controller Read Only

Queue Element Transfer Descriptors must be aligned on 32-byte boundaries.

65.4.2.5.1 Next qTD Pointer

The first DWord of an element transfer descriptor is a pointer to another transfer element descriptor.

The following table describes Next qTD pointer fields.

Table 65-21. qTD Next Element Transfer Pointer (DWord 0)

Bit	Description
31-5	Next Transfer Element Pointer. This field contains the physical memory address of the next qTD to be processed. The field corresponds to memory address signals[31:5], respectively.
4-1	Reserved

Table continues on the next page...

Table 65-21. qTD Next Element Transfer Pointer (DWord 0) (continued)

0	Terminate (T). 1= pointer is invalid. 0=Pointer is valid (points to a valid Transfer Element Descriptor). This bit indicates to the Host Controller that there are no more valid entries in the queue.

65.4.2.5.2 Alternate Next qTD Pointer

The second DWord of a queue element transfer descriptor is used to support hardware-only advance of the data stream to the next transfer descriptor on short packet. To be more explicit the host controller always uses this pointer when the current qTD is retired due to short packet.

The following table describes the TD Alternate Next Element Transfer Pointer field descriptions.

Table 65-22. TD Alternate Next Element Transfer Pointer (DWord 1)

Bit	Description
31-5	Alternate Next Transfer Element Pointer. This field contains the physical memory address of the next qTD to be processed in the event that the current qTD execution encounters a short packet (for an IN transaction). The field corresponds to memory address signals [31:5], respectively.
4-1	Reserved
0	Terminate (T). 1= pointer is invalid. 0=Pointer is valid (points to a valid Transfer Element Descriptor). This bit indicates to the Host Controller that there are no more valid entries in the queue.

65.4.2.5.3 qTD Token

The third DWord of a queue element transfer descriptor contains most of the information the host controller requires to execute a USB transaction (the remaining endpoint-addressing information is specified in the queue head).

NOTE

The field descriptions forward reference fields defined in the queue head. Where necessary, these forward references are preceded with a QH notation.

The following table describes the TD Token fields.

Table 65-23. TD Token (DWord 2)

Bit	Description
31 Data Toggle	This is the data toggle sequence bit. The use of this bit depends on the setting of the *Data Toggle Control* bit in the queue head.

Table continues on the next page...

Table 65-23. TD Token (DWord 2) (continued)

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>30-16
Total Bytes to Transfer</td><td>This field specifies the total number of bytes to be moved with this transfer descriptor. This field is decremented by the number of bytes actually moved during the transaction, only on the successful completion of the transaction. The maximum value software may store in this field is 5 * 4K (5000H). This is the maximum number of bytes 5 page pointers can access. If the value of this field is zero when the host controller fetches this transfer descriptor (and the active bit is set), the host controller executes a zero-length transaction and retires the transfer descriptor. It is not a requirement for OUT transfers that Total Bytes To Transfer be an even multiple of QHD.Maximum Packet Length. If software builds such a transfer descriptor for an OUT transfer, the last transaction is always less than QHD.Maximum Packet Length.

Although it is possible to create a transfer up to 20K this assumes the 1st offset into the first page is 0. When the offset cannot be predetermined, crossing past the 5th page can be guaranteed by limiting the total bytes to 16K**. Therefore, the maximum recommended transfer is 16 K(4000H).</td></tr>
<tr><td>15
Interrupt On Complete (IOC)</td><td>If this bit is set to a one, it specifies that when this qTD is completed, the Host Controller should issue an interrupt at the next interrupt threshold.</td></tr>
<tr><td>14-12
Current Page (C_Page)</td><td>This field is used as an index into the qTD buffer pointer list. Valid values are in the range 0H to 4H. The host controller is not required to write this field back when the qTD is retired.</td></tr>
<tr><td>11-10
Error Counter (CERR)</td><td>This field is a 2-bit down counter that keeps track of the number of consecutive Errors detected while executing this qTD. If this field is programmed with a non-zero value during set-up, the Host Controller decrements the count and writes it back to the qTD if the transaction fails. If the counter counts from one to zero, the Host Controller marks the qTD inactive, sets the Halted bit to a one, and error status bit for the error that caused CERR to decrement to zero. An interrupt is generated if the USB Error Interrupt Enable bit in the USBINTR register is set to a one. If HCD programs this field to zero during set-up, the Host Controller does not count errors for this qTD and there is no limit on the retries of this qTD. Note that write-backs of intermediate execution state are to the queue head overlay area, not the qTD.

Transaction Error - Decrement

Data Buffer Error - No Decrement[3]

Stalled - No Decrement[1]

Babble Detected - No Decrement[1]

No Error - No Decrement[2]
<table>
<tr><th>Error</th><th>Decrement Counter</th></tr>
<tr><td>1</td><td>Detection of Babble or Stall automatically halts the queue head. Thus, count is not decremented</td></tr>
<tr><td>2</td><td>If the EPS field indicates a HS device or the queue head is in the Asynchronous Schedule (and PIDCode indicates an IN or OUT) and a bus transaction completes and the host controller does not detect a transaction error, then the host controller should reset CERR to extend the total number of errors for this transaction. For example, CERR should be reset with maximum value (3) on each successful completion of a transaction. The host controller must never reset this field if the value at the start of the transaction is 00b.

See Split Transaction Interrupt for CERR adjustment rules when the EPS field indicates a FS or LS device and the queue head is in the Periodic Schedule. See Asynchronous - Do Complete Split for CERR adjustment rules when the EPS field indicates a FS or LS device, the queue head is in the Asynchronous schedule and the PIDCode indicates a SETUP.</td></tr>
</table></td></tr>
</table>

Table continues on the next page...

Table 65-23. TD Token (DWord 2) (continued)

Bit	Description	
	3	Data buffer errors are host problems. They don't count against the device's retries.
	NOTE: Software must not program CERR to a value of zero when the EPS field is programmed with a value indicating a Full- or Low-speed device. This combination could result in undefined behavior.	
9-8 PID Code	This field is an encoding of the token, which should be used for transactions associated with this transfer descriptor. Encodings are:	
	00b	OUT Token generates token (E1H)
	01b	IN Token generates token (69H)
	10b	SETUP Token generates token (2DH) (undefined if endpoint is an interrupt, the queue head is non-zero) transfer type, for example, µ*Frame S-mask* field in.
	11b	Reserved
7-0 Status	This field is used by the Host Controller to communicate individual command execution states back to HCD. This field contains the status of the last transaction performed on this qTD. The bit encodings are:	
	Bit	Status Field Description
	7	Active. Set to one by software to enable the execution of transactions by the Host Controller.
	6	Halted. Set to one by the Host Controller during status updates to indicate that a serious error has occurred at the device/endpoint addressed by this qTD. This can be caused by babble, the error counter counting down to zero, or reception of the STALL handshake from the device during a transaction. Any time that a transaction results in the Halted bit being set to a one, the Active bit is also set to zero.
	5	Data Buffer Error. Set to a one by the Host Controller during status update to indicate that the Host Controller is unable to keep up with the reception of incoming data (overrun) or is unable to supply data fast enough during transmission (under run). If an overrun condition occurs, the Host Controller forces a timeout condition on the USB, invalidating the transaction at the source. If the host controller sets this bit to a one, then it remains a one for the duration of the transfer.
	4	Babble Detected. Set to a one by the Host Controller during status update when" babble" is detected during the transaction. In addition to setting this bit, the Host Controller also sets the *Halted* bit to a one. Because "babble" is considered a fatal error for the transfer, setting the Halted bit to a one insures that no more transactions occur because of this descriptor.
	3	Transaction Error (XactErr). Set to a one by the Host Controller during status update in the case where the host did not receive a valid response from the device (Timeout, CRC, Bad PID, etc.). If the host controller sets this bit to a one, then it remains a one for the duration of the transfer.
	2	Missed Micro-Frame. This bit is ignored unless the *QH.EPS* field indicates a full- or low-speed endpoint and the queue head is in the periodic list. This bit is set when the host controller detected that a host-induced hold-off caused the host controller to miss a required complete-split transaction. If the host controller sets this bit to a one, then it remains a one for the duration of the transfer.
	1	Split Transaction State (SplitXstate). This bit is ignored by the host controller unless the *QH.EPS* field indicates a full- or low-speed endpoint. When a Full- or Low-speed device, the host controller uses this bit to track the state of the split-

Table continues on the next page...

Table 65-23. TD Token (DWord 2) (continued)

Bit	Description	
		transaction. The functional requirements of the host controller for managing this state bit and the split transaction protocol depends on whether the endpoint is in the periodic or asynchronous schedule. The bit encodings are: Value Meaning 0b Do Start Split. This value directs the host controller to issue a Start split transaction to the endpoint. 1b Do Complete Split. This value directs the host controller to issue a Complete split transaction to the endpoint.
	0	Ping State (P)/ERR. If the *QH.EPS* field indicates a High-speed device and the *PID_Code* indicates an OUT endpoint, then this is the state bit for the Ping protocol. The bit encodings are: Value Meaning 0b Do OUT. This value directs the host controller to issue an OUT PID to the endpoint. 1b Do Ping. This value directs the host controller to issue a PING PID to the endpoint. If the *QH.EPS* field does not indicate a High-speed device, then this field is used as an error indicator bit. It is set to a one by the host controller whenever a periodic split-transaction receives an ERR handshake.

65.4.2.5.4 qTD Buffer Page Pointer List

The last five DWords of a queue element transfer descriptor is an array of physical memory address pointers. These pointers reference the individual pages of a data buffer.

System software initializes Current Offset field to the starting offset into the current page, where current page is selected through the value in the C_Page field.

The following table describes the qTD Buffer Pointer(s) (DWords 3-7) fields.

Table 65-24. qTD Buffer Pointer(s) (DWords 3-7)

Bit	Description
31-12	Buffer Pointer List. Each element in the list is a 4 K page aligned physical memory address. The lower 12 bits in each pointer are reserved (except for the first one), as each memory pointer must reference the start of a 4 K page. The field C_Page specifies the current active pointer. When the transfer element descriptor is fetched, the starting buffer address is selected using C_Page (similar to an array index to select an array element). If a transaction spans a 4K buffer boundary, the host controller must detect the page-span boundary in the data stream, increment C_Page and advance to the next buffer pointer in the list, and conclude the transaction through the new buffer pointer.
11-0	Current Offset (Reserved). This field is reserved in all pointers except the first one (for example Page 0). The host controller should ignore all reserved bits. For the page 0 current offset interpretation, this field is the byte offset into the current page (as selected by C_Page). The host controller is not required to write this field back when the qTD is retired. Software should ensure the Reserved fields are initialized to zero.

65.4.2.6 Queue Head

The table located in this section shows the Queue Head structure layout.

The following table shows the queue head structure layout.

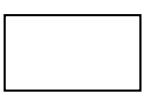
Table 65-25. Queue Head Structure Layout

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Addr
Queue Head Horizontal Link Pointer																											0		Typ		T	03-00
RL				C	Maximum Packet Length											H	dtc	EP		EndPt				I	Device Address							07-04[1]
Mult		Port Number[2]							Hub Addr[2]							μFrame C-mask[2]								μFrame S-mask								0B-08[1]
Current qTD Pointer																											0					0F-0C
Next qTD Pointer																											0				T	13-10[3]
Alternate Next qTD pointer																											NakCnt				T	17-14[4]
dt	Total Bytes to Transfer															ioc	C_Page			Cerr		PID Code		Status								1B-18
Buffer Pointer (Page 0)																				Current Offset												1F-1C
Buffer Pointer (Page 1)																				Reserved				C-prog-mask[2]								23-20
Buffer Pointer (Page 2)																				S-bytes[2]										FrameTag[2]		27-24[4]
Buffer Pointer (Page 3)																				Reserved												2B-28
Buffer Pointer (Page 4)																				Reserved												2F-2C[3]

1. 04-0B: Static endpoint state.
2. These fields are used exclusively to support split transactions to USB 2.0 hubs
3. 10-2F: Transfer overlay.
4. 14-27: Transfer results.



Host Controller Read/Write



Host Controller Read Only

65.4.2.6.1 Queue Head Horizontal Link Pointer

The first DWord of a Queue Head contains a link pointer to the next data object to be processed after any required processing in this queue has been completed, as well as the control bits defined below.

This pointer may reference a queue head or one of the isochronous transfer descriptors. It must not reference a queue element transfer descriptor.

The following table describes the Queue head DWord 0 fields.

Table 65-26. Queue Head DWord 0

Bit	Description
31-5	Queue Head Horizontal Link Pointer (QHLP). This field contains the address of the next data object to be processed in the horizontal list and corresponds to memory address signals [31:5], respectively.
4-3	Reserved
2-1	QH/(s)iTD Select (Typ). This field indicates to the hardware whether the item referenced by the link pointer is an iTD, siTD or a QH. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor) 11b FSTN (frame span traversal node)
0	Terminate (T). 1=Last QH (pointer is invalid). 0=Pointer is valid. If the queue head is in the context of the periodic list, a one bit in this field indicates to the host controller that this is the end of the periodic list. This bit is ignored by the host controller when the queue head is in the Asynchronous schedule. Software must ensure that queue heads reachable by the host controller always have valid horizontal link pointers.

65.4.2.6.2 Queue Head Endpoint Capabilities/Characteristics

The second and third DWords of a Queue Head specifies static information about the endpoint. This information does not change over the lifetime of the endpoint.

There are three types of information in this region:

- Endpoint Characteristics. These are the USB endpoint characteristics including addressing, maximum packet size, and endpoint speed.
- Endpoint Capabilities. These are adjustable parameters of the endpoint. They effect how the endpoint data stream is managed by the host controller.
- Split Transaction Characteristics. This data structure is used to manage full- and low-speed data streams for bulk, control, and interrupt via split transactions to USB2.0 Hub Transaction Translator. There are additional fields used for addressing the hub and scheduling the protocol transactions (for periodic).

The host controller must not modify the bits in this region.

The following table describes the Endpoint characteristics: Queue head DWord 1 fields.

Table 65-27. Endpoint Characteristics: Queue Head DWord 1

Bit	Description	
31-28	Nak Count Reload (RL). This field contains a value, which is used by the host controller to reload the Nak Counter field.	
27	Control Endpoint Flag (C). If the *QH.EPS* field indicates the endpoint is not a high-speed device, and the endpoint is a control endpoint, then software must set this bit to a one. Otherwise, it should always set this bit to zero.	
26-16	Maximum Packet Length. This directly corresponds to the maximum packet size of the associated endpoint (*wMaxPacketSize*). The maximum value this field may contain is 0x400 (1024).	
15	Head of Reclamation List Flag (H). This bit is set by System Software to mark a queue head as being the head of the reclamation list.	
14	Data Toggle Control (DTC). This bit specifies where the host controller should get the initial data toggle on an overlay transition. 0b Ignore DT bit from incoming qTD. Host controller preserves DT bit in the queue head. 1b Initial data toggle comes from incoming qTD DT bit. Host controller replaces DT bit in the queue head from the DT bit in the qTD.	
13-12	Endpoint Speed (EPS). This is the speed of the associated endpoint. Bit combinations are:	
	Value	Meaning
	00b	Full-Speed (12 Mbits/sec)
	01b	Low-Speed (1.5 Mbits/sec)
	10b	High-Speed (480 Mbits/sec)
	11b	Reserved
	This field must not be modified by the host controller.	
11-8	Endpoint Number (Endpt). This 4-bit field selects the particular endpoint number on the device serving as the data source or sink.	
7	Inactivate on Next Transaction (I). This bit is used by system software to request that the host controller set the Active bit to zero. See Rebalancing the periodic schedule , for full operational details. This field is only valid when the queue head is in the Periodic Schedule and the *EPS* field indicates a Full or Low-speed endpoint. Setting this bit to one when the queue head is in the Asynchronous Schedule or the *EPS* field indicates a high-speed device yields undefined results.	
6-0	Device Address. This field selects the specific device serving as the data source or sink.	

The table below describes the Endpoint capabilities: Queue head DWord 2 field descriptions.

Table 65-28. Endpoint Capabilities: Queue Head DWord 2

Bit	Description
31-30	High-Bandwidth Pipe Multiplier (Mult). This field is a multiplier used to key the host controller as the number of successive packets the host controller may submit to the endpoint in the current execution. The host controller makes the simplifying assumption that software properly initializes this field (regardless of location of queue head in the schedules or other run time parameters). The valid values are: Value Meaning

Table continues on the next page...

Table 65-28. Endpoint Capabilities: Queue Head DWord 2 (continued)

	00b Reserved. A zero in this field yields undefined results. 01b One transaction to be issued for this endpoint per micro-frame. 10b Two transactions to be issued for this endpoint per micro-frame. 11b Three transactions to be issued for this endpoint per micro-frame.
29-23	Port Number. This field is ignored by the host controller unless the *EPS* field indicates a full- or low-speed device. The value is the port number identifier on the USB 2.0 Hub (for hub at device address *Hub Addr* below), below which the full- or low-speed device associated with this endpoint is attached. This information is used in the split-transaction protocol.
22-16	Hub Addr. This field is ignored by the host controller unless the *EPS* field indicates a full-or low-speed device. The value is the USB device address of the USB 2.0 Hub below which the full- or low-speed device associated with this endpoint is attached. This field is used in the split-transaction protocol.
15-8	Split Completion Mask (µFrame C-Mask). This field is ignored by the host controller unless the *EPS* field indicates this device is a low- or full-speed device and this queue head is in the periodic list. This field (along with the *Active* and *SplitX-state* fields) is used to determine during which micro-frames the host controller should execute a complete-split transaction. When the criteria for using this field are met, a zero value in this field has undefined behavior. This field is used by the host controller to match against the three low-order bits of the FRINDEX register. If the FRINDEX register bits decode to a position where the µ*Frame C- Mask* field is a one, then this queue head is a candidate for transaction execution. There may be more than one bit in this mask set.
7-0	Interrupt Schedule Mask (µFrame S-mask). This field is used for all endpoint speeds. Software should set this field to a zero when the queue head is on the asynchronous schedule. A non-zero value in this field indicates an interrupt endpoint. The host controller uses the value of the three low-order bits of the FRINDEX register as an index into a bit position in this bit vector. If the µ*Frame S-mask* field has a one at the indexed bit position then this queue head is a candidate for transaction execution. If the *EPS* field indicates the endpoint is a high-speed endpoint, then the transaction executed is determined by the *PID_Code* field contained in the execution area. This field is also used to support split transaction types: Interrupt (IN/OUT). This condition is true when this field is non-zero and the *EPS* field indicates this is either a full- or low-speed device. A zero value in this field, in combination with existing in the periodic frame list has undefined results.

65.4.2.6.3 Transfer Overlay-Queue Head

The nine DWords in this area represent a transaction working space for the host controller. The general operational model is that the host controller can detect whether the overlay area contains a description of an active transfer. If it does not contain an active transfer, then it follows the Queue Head Horizontal Link Pointer to the next queue head. The host controller will never follow the Next Transfer Queue Element or Alternate Queue Element pointers unless it is actively attempting to advance the queue. For the duration of the transfer, the host controller keeps the incremental status of the transfer in the overlay area. When the transfer is complete, the results are written back to the original queue element.

The DWord3 of a Queue Head contains a pointer to the source qTD currently associated with the overlay. The host controller uses this pointer to write back the overlay area into the source qTD after the transfer is complete.

The following table describes the current qTD link pointer field descriptions.

Table 65-29. Current qTD Link Pointer

Bit	Description
31-5	Current Element Transaction Descriptor Link Pointer. This field contains the address Of the current transaction being processed in this queue and corresponds to memory address signals [31:5], respectively.
4-0	Reserved (R). These bits are ignored by the host controller when using the value as an address to write data. The actual value may vary depending on the usage.

The DWords 4-11 of a queue head are the transaction overlay area. This area has the same base structure as a Queue Element Transfer Descriptor. The queue head utilizes the reserved fields of the page pointers to implement tracking the state of split transactions.

This area is characterized as an overlay because when the queue is advanced to the next queue element, the source queue element is merged onto this area. This area serves as execution cache for the transfer.

The table below describes the Host-controller rules for bits in overlay.

Table 65-30. Host-Controller Rules for Bits in Overlay (DWords 5, 6, 8 and 9)

DWord	Bit	Description
5	4-1	Nak Counter (NakCnt)µRW. This field is a counter the host controller decrements whenever a transaction for the endpoint associated with this queue head results in a Nak or Nyet response. This counter is reloaded from *RL* before a transaction is executed during the first pass of the reclamation list (relative to an Asynchronous List Restart condition). It is also loaded from RL during an overlay.
6	31	Data Toggle. The *Data Toggle Control* controls whether the host controller preserves this bit when an overlay operation is performed.
6	15	Interrupt On Complete (IOC). The IOC control bit is always inherited from the source qTD when the overlay operation is performed.
6	11-10	Error Counter (C_ERR). This two-bit field is copied from the qTD during the overlay and written back during queue advancement.
6	0	Ping State (P)/ERR. If the *EPS* field indicates a high-speed endpoint, then this field should be preserved during the overlay operation.
8	7-0	Split-transaction Complete-split Progress (C-prog-mask). This field is initialized to zero during any overlay. This field is used to track the progress of an interrupt split-transaction.
9	4-0	Split-transaction Frame Tag (Frame Tag). This field is initialized to zero during any overlay. This field is used to track the progress of an interrupt split-transaction.
9	11-5	S-bytes. Software must ensure that the *S-bytes* field in a *qTD* is zero before activating the *qTD*. This field is used to keep track of the number of bytes sent or received during an IN or OUT split transaction.

65.4.2.7 Periodic Frame Span Traversal Node (FSTN)

This data structure is to be used only for managing Full- and Low-speed transactions that span a Host-frame boundary.

See Host Controller Operational Model for FSTNs for full operational details. Software must not use an FSTN in the Asynchronous Schedule. An FSTN in the Asynchronous schedule results in undefined behavior. Software must not use the FSTN feature with a host controller whose USB_HCIVERSION register indicates a revision implementation below 0096h. FSTNs are not defined for implementations before 0.96 and their use yields undefined results.

Table 65-31. Frame Span Traversal Node Structure Layout

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Addr
Normal Path Link Pointer																											0		Typ		T	03-00
Back Path Link Pointer																											0		Typ[1]		T	07-04

1. Must be set to indicate a queue head

Host Controller Read/Write

Host Controller Read Only

65.4.2.7.1 FSTN Normal Path Pointer

The first DWord of an FSTN contains a link pointer to the next schedule object. This object can be of any valid periodic schedule data type.

The following table describes the FSTN normal path pointer fields.

Table 65-32. FSTN Normal Path Pointer Field Descriptions

Bit	Description
31-5	Normal Path Link Pointer (NPLP). This field contains the address of the next data object to be processed in the periodic list and corresponds to memory address signals [31:5], respectively.
4-3	Reserved
2-1	QH/(s)iTD/FSTN Select (Typ). This field indicates to the Host Controller whether the item referenced is a iTD/siTD, a QH or an FSTN. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor) 11b FSTN (Frame Span Traversal Node)
0	Terminate (T). 1 = Link Pointer field is not valid. 0 = Link Pointer is valid.

65.4.2.7.2 FSTN Back Path Link Pointer

The second DWord of an FTSN node contains a link pointer to a queue head.

If the T-bit in this pointer is zero, then this FSTN is a Save-Place indicator. Its Typ field must be set by software to indicate the target data structure is a queue head. If the T-bit in this pointer is set to one, then this FSTN is the Restore indicator. When the T-bit is one, the host controller ignores the Typ field.

The following table describes the FSTN back path link pointer fields.

Table 65-33. FSTN Back Path Link Pointer Field Descriptions

Bit	Description
31-5	Back Path Link Pointer (BPLP). This field contains the address of a Queue Head. This field corresponds to memory address signals [31:5], respectively.
4-3	Reserved
2-1	Typ. Software must ensure this field is set to indicate the target data structure is a Queue Head. Any other value in this field yields undefined results.
0	Terminate (T). 1=Link Pointer field is not valid (that is the host controller must not use bits [31:5] as a valid memory address). This value also indicates that this FSTN is a Restore indicator. 0=Link Pointer is valid (that is the host controller may use bits [31:5] (in combination with the CTRLDSSEGMENT register if applicable) as a valid memory address). This value also indicates that this FSTN is a Save-Place indicator.

65.4.3 Host Operational Model

The general operational model is for the enhanced interface host controller hardware and enhanced interface host controller driver (generally referred to as system software).

Each significant operational feature of the EHCI host controller is discussed in a separate section. Each section presents the operational model requirements for the host controller hardware. Where appropriate, recommended system software operational models for features are also presented.

65.4.3.1 Host Controller Initialization

After initial power-on or HCReset (hardware or through HCReset bit in the USB_USBCMD register), all of the operational registers are at their default values. After a hardware reset, only the operational registers not contained in the Auxiliary power well are at their default values.

The following table describes the default values of operational registers.

.

Table 65-34. Default Values of Operational Register Space

Operational Register	Default Value (after Reset)
USB_USBCMD	00080000h (00080B00h, if *Asynchronous Schedule Park Capability is one*)
USB_USBSTS	00001000h
USB_USBINTR	00000000h
USB_FRINDEX	00000000h
USB_CTRLDSSEGMENT	00000000h
USB_PERIODICLISTBASE	Undefined
USB_ASYNCLISTADDR	Undefined
USB_CONFIGFLAG	00000000h
USB_PORTSC1	00002000h (w/*PPC* set to one); 00003000h (w/*PPC* set to zero)

To initialize the host controller, software should perform the following steps:

- Write the appropriate value to the USB_USBINTR register to enable the appropriate interrupts.
- Write the base address of the Periodic Frame List to the USB_PERIODICLIST BASE register. If no work items are in the periodic schedule, all elements of the Periodic Frame List should have their T-Bits set to one.
- Write the USB_USBCMD register to set the desired interrupt threshold, frame list size (if applicable) and turn the host controller ON through setting the Run/Stop bit.

At this point, the host controller is up and running and the port registers begin reporting device connects, and so on. System software can enumerate a port through the reset process (where the port is in the enabled state). At this point, the port is active with SOFs occurring down the enabled ports, but the schedules have not enabled. To communicate with devices through the asynchronous schedule, system software must write the USB_ASYNCLISTADDR register with the address of a control or bulk queue head. Software must then enable the asynchronous schedule by writing one to the Asynchronous Schedule Enable bit in the USB_USBCMD register. To communicate with devices through the periodic schedule, system software must enable the periodic schedule by writing one to the Periodic Schedule Enable bit in the USB_USBCMD register.

NOTE

The schedules can be turned on before the first port is reset (and enabled).

When the USB_USBCMD register is written, system software must ensure the appropriate bits are preserved, depending on the intended operation.

65.4.3.2 Port Routing and Control

The EHCI specification defines that a USB 2.0 Host controller is comprised of one high-speed host controller, which implements the EHCI programming interface and 0 to N USB 1.1 companion host controllers.

Companion host controllers (cHCs) may be implementations of either Universal or Open host controller specifications. This configuration is used to deliver the required full USB 2.0-defined port capability; for example, Low-, Full-, and High-speed capability for every port.

NOTE

The USB controllers on do not require nor support companion controllers to support Full and Low Speed device. Full and Low Speed devices are supported within the USB controller by emulating the functionality of a high-speed HUB. Therefore, no port routing is present in the controller. Please refer to Embedded Transaction Translator Function for details.

The following figure illustrates a simple block diagram of the port routing logic and its relationship to the high-speed and companion host controllers within a USB 2.0 host controller.

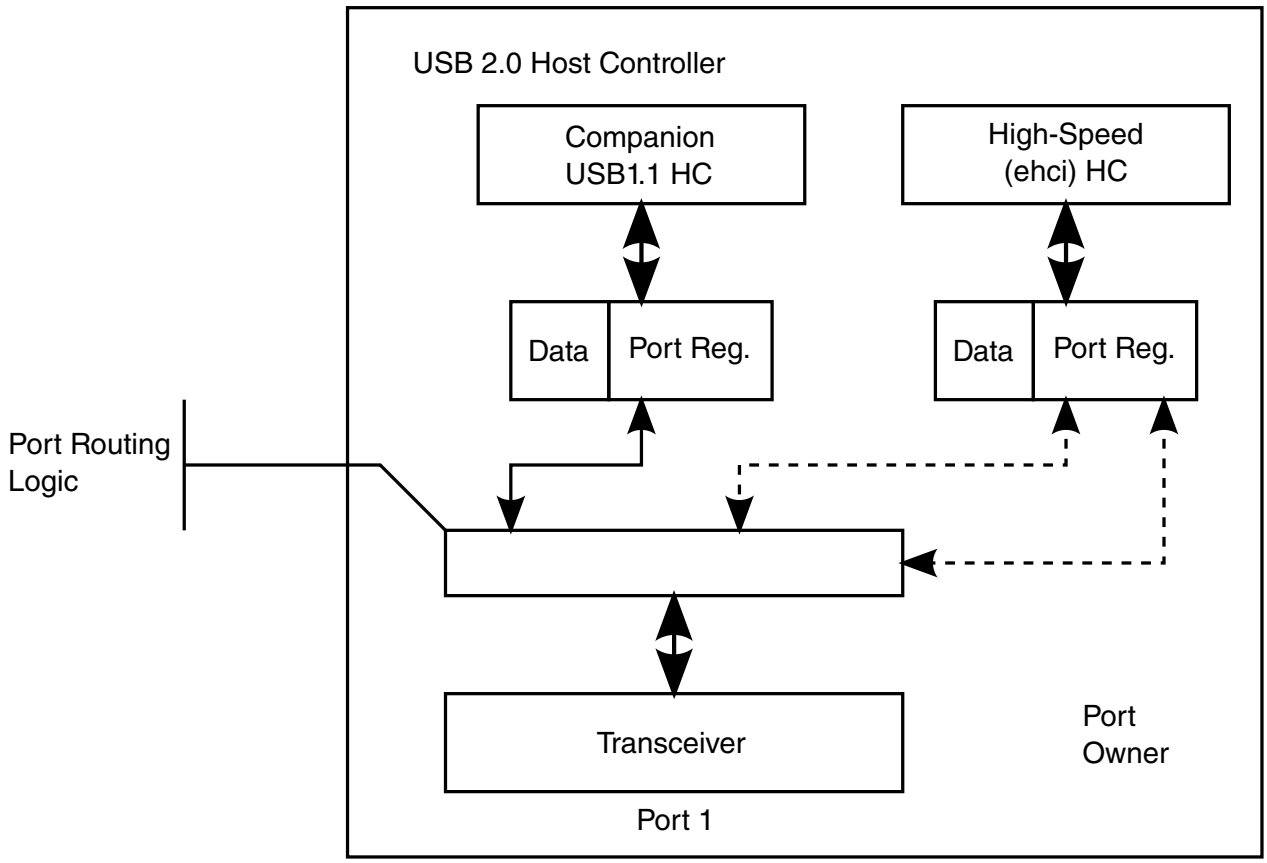


Figure 65-5. Example USB 2.0 Host Controller Port Routing Block Diagram

There exists one transceiver per physical port and each host controller block has its own port status and control registers. The EHCI controller has port status and control registers for every port. Each companion host controller has only the port control and status registers it is required to operate. Either the EHCI host controller or one companion host controller controls each transceiver. Routing logic lies between the transceiver, the port status and control registers.[1]

The port routing logic is controlled from signals originating in the EHCI host controller. The EHCI host controller has a global routing policy control field and per-port ownership control fields. The Configured Flag (CF) bit is the global routing policy control. At power-on or reset, the default routing policy is to the companion controllers (if they exist). If the system does not include a driver for the EHCI host controller and the host controller includes Companion Controllers, then the ports still work in Full- and Low-speed mode (assuming the system includes a driver for the companion controllers). In general, when the EHCI owns the ports, the companion host controllers' port registers do not see a connect indication from the transceiver. Similarly, when a companion host controller owns a port, the EHCI controller's port registers do not see a connect indication

1. The routing logic should not be implemented in the 480 MHz clock domain of the transceiver.

from the transceiver. The details on the rules for the port routing logic are described in the following sections. The USB 2.0 host controller must be implemented as a multi-function PCI device if the implementation includes companion controllers. The companion host controllers' function numbers must be less than the EHCI host controller function number. The EHCI host controller must be a larger function number with respect to the companion host controllers associated with this EHCI host controller. If a PCI device implementation contains only an EHCI controller (that is no companion controllers or other PCI functions), then the EHCI host controller must be function zero, in accordance with the PCI Specification. The N_CC field in the Structural Parameter register (HCSPARAMS) indicates whether the controller implementation includes companion host controllers. When N_CC has a non-zero value there exists companion host controllers. If N_CC has a value of zero, then the host controller implementation does not include companion host controllers. If the host controller root ports are exposed to attachment of full- or low-speed devices, the ports always fails the high-speed chirp during reset and the ports are not enabled. System software can notify the user of the illegal condition. This type of implementation requires a USB 2.0 hub be connected to a root port to provide full and low-speed device connectivity.

System software uses information in the host controller capability registers to determine how the ports are routed to the companion host controllers. See Host Controller Structural Parameters (USB_*n*HCSPARAMS)

65.4.3.2.1 Port Routing Control through EHCI Configured (CF) Bit

Each port in the USB 2.0 host controller are routed either to a single companion host controller or to the EHCI host controller.

The port routing logic is controlled by two mechanisms in the EHCI HC: a host controller global flag and per-port control. The Configured Flag (CF) bit, is used to globally set the policy of the routing logic. Each port register has a Port Owner control bit which allows the EHCI Driver to explicitly control the routing of individual ports. Whenever the CF bit transitions from zero to one (this transition is only available under program control) the port routing unconditionally routes all of the port registers to the EHCI HC (all Port Owner bits go to zero). While the CF-bit is one, the EHCI Driver controls individual ports' routing through the Port Owner control bit. Likewise, whenever the CF bit transitions from one to zero (as a result of Aux power application, HCRESET, or software writing zero to CF-bit), the port routing unconditionally routes all of the port registers to the appropriate companion HC. The default value for the EHCI HC's CF bit (after Aux power application or HCRESET) is zero.

The *view* of the port depends on the current owner. A Universal or Open companion host controller will see port register bits consistent with the appropriate specification. Port bit definitions that are required for EHCI host controllers are not visible to companion host controllers.

The following table summarizes the default routing for all the ports, based on the value of the EHCI HC's CF bit.

Table 65-35. Default Port Routing Depending on EHCI HC CF Bit

HS CF Bit	Default Port Ownership	Explanation
0B	Companion HCs	The companion host controllers own the ports and only Full- and Low-speed devices are supported in the system. The exact port assignments are implementation dependent. The ports behave only as Full- and Low-speed ports in this configuration
1B	EHCI HC	The EHCI host controller has default ownership over all of the ports. The routing logic inhibits device connect events from reaching the companion HCs' port status and control registers when the port owner is the EHCI HC.The EHCI HC has access to the additional port status and control bits defined in this specification (see Port Status & Control (USB_*n*PORTSC1)). The EHCI HC can temporarily release control of the port to a companion HC by setting the *PortOwner* bit in the PORTSC1 register to one.

65.4.3.2.2 Port Routing Control through PortOwner and Disconnect Event

Manipulating the port routing through the CF-bit is an extreme process and not intended to be used during normal operation.

The normal mode of port ownership transferal is on the granularity of individual ports using the Port Owner bit in the EHCI HC's USB_PORTSC1 register (for hand-offs from EHCI to companion host controllers). Individual port ownership is returned to the EHCI controller when the port registers a device disconnect. When the disconnect is detected, the port routing logic immediately returns the port ownership to the EHCI controller. The companion host controller port register detects the device disconnect and operates normally.

Under normal operating conditions (assuming all HC drivers loaded and operational and the EHCI *CF-bit* is set to one), the typical port enumeration sequence proceeds as illustrated below:

- Initial condition is that EHCI is port owner. A device is connected causing the port to detect a connect, set the port connect change bit and issue a port-change interrupt (if enabled).
- EHCI Driver identifies the port with the new connect change bit asserted and sends a change report to the hub driver. Hub driver issues a GetPortStatus() request and

identifies the connect change. It then issues a request to clear the connect change, followed by a request to reset and enable the port.

- When the EHCI Driver receives the request to reset and enable the port, it first checks the value reported by the LineStatus bits in the USB_PORTSC1 register. If they indicate the attached device is a full-speed device (for example, D+ is asserted), then the EHCI Driver sets the PortReset control bit to one (and sets the PortEnable bit to zero) which begins the reset-process. Software times the duration of the reset, then terminates reset signaling by writing zero to the port reset bit. The reset process is actually complete when software reads zero in the PortReset bit. The EHCI Driver checks the PortOwner bit in the USB_PORTSC1 register. If set to one, the connected device is a high-speed device and EHCI Driver (root hub emulator) issues a change report to the hub driver and the hub driver continues to enumerate the attached device.
- At the time the EHCI Driver receives the port reset and enable request the LineStatus bits might indicate a low-speed device. Additionally, when the port reset process is complete, the PortEnable field may indicate that a full-speed device is attached. In either case the EHCI driver sets the PortOwner bit in the USB_PORTSC1 register to one to release port ownership to a companion host controller.
- When the EHCI Driver sets PortOwner bit to one, the port routing logic makes the connection state of the transceiver available to the companion host controller port register and removes the connection state from the EHCI HC port. The EHCI USB_PORTSC1 register observes and reports a disconnect event through the disconnect change bit. The EHCI Driver detects the connection status change (either by polling or by port change interrupt) and then sends a change report to the hub driver. When the hub driver requests that port-state, the EHCI Driver responds with a reset complete change set to one, a connect change set to one and a connect status set to zero. This information is derived directly from the EHCI port register. This allows the hub driver to assume the device was disconnected during reset. It acknowledges the change bits and wait for the next change event. While the EHCI controller does not own the port, it simply remains in a state where the port reports no device connected. The device-connect evaluation circuitry of the companion HC activates and detects the device, the companion Driver detects the connection and enumerates the port.

When a port is routed to a companion HC, it remains under the control of the companion HC until the device is disconnected from the root port (ignoring for now the scenario where EHCI's CF-bit transitions from1b to 0b). When a disconnect occurs, the disconnect event is detected by both the companion HC port control and the EHCI port ownership control. On the event, the port ownership is returned immediately to the EHCI controller. The companion HC stack detects the disconnect and acknowledges as it would in an ordinary standalone implementation. Subsequent connects is detected by the EHCI port register and the process repeats.

65.4.3.2.3 Example Port Routing State Machine

The following figure illustrates an example of how the port ownership should be managed. The following sections describe the entry conditions to each state.

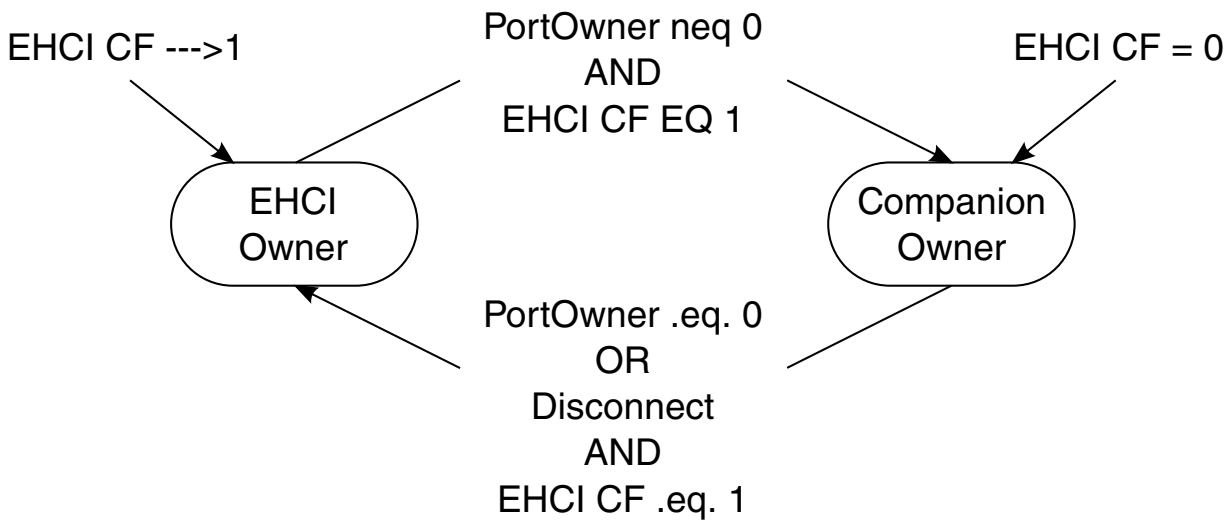


Figure 65-6. Port Owner Handoff State Machine

65.4.3.2.3.1 EHCI HC Owner

Entry to this state occurs when one of the following events occur:

- When the EHCI HC's Configure Flag (CF) bit in the USB_CONFIGFLAG register transitions from zero to one. This signals the fact that the system has a host controller driver for the EHCI HC and that all ports in the USB 2.0 host controller must default route to the EHCI controller.
- When the port is owned by a companion HC and the device is disconnected from the port. The EHCI port routing control logic is notified of the disconnect, and returns port routing to the EHCI controller. The connection state of the companion HC goes immediately to the disconnected state (with appropriate side effect to connect change, enable and enable change). The companion HC driver acknowledges the disconnect by setting the connect status change bit to zero. This allows the companion HC's driver to interact with the port completely through the disconnect process.
- When system software writes zero to the PortOwner bit in the USB_PORTSC1 register. This allows software to take ownership of a port from a companion host controller. When this occurs, the routing logic to the companion HC effectively signals a disconnect to the companion HC's port status and control register.

65.4.3.2.3.2 Companion HC Owner

Entry to this state occurs whenever one of the following events occur:

- When the PortOwner field transitions from zero to one.
- When the HS-mode HC's Configure Flag (CF) is equal to zero.

On entry to this state, the routing logic allows the companion HC port register to detect a device connect. Normal port enumeration proceeds.

65.4.3.2.4 Port Power

The Port Power Control (PPC) bit in the USB_HCSPARAMS register indicates whether the USB 2.0 host controller has port power control (see Host Controller Structural Parameters (USB_*n*HCSPARAMS)).

When this bit is zero, then the host controller does not support software control of port power switches. When in this configuration, the port power is always available and the companion host controllers must implement functionality consistent with port power always on. When the *PPC* bit is one, then the host controller implementation includes port power switches. Each available switch has an output enable, which is referred to in this discussion as PortPowerOutputEnable (PPE). PPE is controlled based on the state of the combination bits PPC bit, EHCI Configured (CF)-bit and individual Port Power (PP) bits.

The following table describes the summary behavioral model.

Table 65-36. Port Power Enable Control Rules

CF	CHC[1] (PP)	EHC[2] (PP)	Owner	PPE[3]	Description
0	0	X	CHC	0	When the EHCI controller is not configured, the port is owned by the companion host controller. When the companion HC's port power select is off, then the port power is off.
0	1	X	CHC	1	Similar to previous entry. When the companion HC's port power select is on, then the port power is on.
1	0	0	CHC	0	Port owner has port power turned off, the power to port is off.
1	0	0	EHC	0	Port owner has port power turned off, the power to port is off.
1	0	1	EHC	1	Port owner has port power on, so power to port is on.
1	0	1	CHC	1	If either HC has port power turned on, the power to the port is on.

Table continues on the next page...

Table 65-36. Port Power Enable Control Rules (continued)

1	1	0	EHC	1	If either HC has port power turned on, the power to the port is on.
1	1	0	CHC	1	Port owner has port power on, so power to port is on.
1	1	1	CHC	1	Port owner has port power on, so power to port is on.
1	1	1	EHC	1	Port owner has port power on, so power to port is on.

1. CHC (Companion Host Controller).
2. EHC (EHCI Host Controller).
3. PPE (Port Power Enable). This bit actually turns on the port power switch (if one exists).

65.4.3.2.5 Port Reporting Over-Current

Host controllers are by definition power providers on USB. Whether the ports are considered high- or low-powered is a platform implementation issue. Each EHCI USB_PORTSC1 register has an over-current status and over-current change bit.

The functionality of these bits are specified in the USB Specification Revision 2.0.

The over current detection and limiting logic usually resides outside the host controller logic. This logic may be associated with one or more ports. When this logic detects an over-current condition it is made available to both the companion and EHCI ports. The effect of an over-current status on a companion host controller port is beyond the scope of this document.

The over-current condition effects the following bits in the USB_PORTSC1 register on the EHCI port:

- Over-current Active bits are set to one. When the over-current condition goes away, the Over-current Active bit transitions from one to zero.
- Over-current Change bits are set to one. On every transition of the Over-current Active bit the host controller sets the Over-current Change bit to one. Software sets the Over-current Change bit to zero by writing one to this bit.
- Port Enabled/Disabled bit is set to zero. When this change bit gets set to one, then the Port Change Detect bit in the USB_USBSTS register is set to one.
- Port Power (PP) bits may optionally be set to zero. There is no requirement in USB that a power provider shut off power in an over current condition. It is sufficient to limit the current and leave power applied. When the Over-current Change bit transitions from zero to one, the host controller also sets the Port Change Detect bit in the USB_USBSTS register to one. In addition, if the Port Change Interrupt Enable bit in the USB_USBINTR register is one, then the host controller issues an interrupt to the system. Refer to Table 65-37 for summary behavior for over-current detection

when the host controller is halted (suspended from a device component point of view).

65.4.3.3 Suspend/Resume-Host Operational Model

The EHCI host controller provides an equivalent suspend and resume model as that defined for individual ports in a USB 2.0 Hub.

Control mechanisms are provided to allow system software to suspend and resume individual ports. The mechanisms allow the individual ports to be resumed completely through software initiation. Other control mechanisms are provided to parameterize the host controller's response (or sensitivity) to external resume events. In this discussion, host-initiated, or software initiated resumes are called Resume Events/Actions. Bus-initiated resume events are called wake-up events. The classes of wake-up events are:

- Remote-wake-up enabled device asserts resume signaling. In similar kind to USB 2.0 Hubs, EHCI controllers must always respond to explicit device resume signaling and wake-up the system (if necessary).
- Port connect and disconnect and over-current events. Sensitivity to these events can be turned on or off by using the per-port control bits in the USB_PORTSC1 registers.

Selective suspend is a feature supported by every USB_PORTSC1 register. It is used to place specific ports into a suspend mode. This feature is used as a functional component for implementing the appropriate power management policy implemented in a particular operating system. When system software intends to suspend the entire bus, it should selectively suspend all enabled ports, then shut off the host controller by setting the Run/Stop bit in the USB_USBCMD register to zero. The EHCI sub-block can then be placed into a lower device state through the PCI power management interface (see Appendix A, Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 0.95, November 2000, Intel Corporation. http://www.intel.com).

When a wake event occurs, the system resumes operation and system software eventually set the Run/Stop bit to one and resume the suspended ports. Software must not set the Run/Stop bit to one until it is confirmed that the clock to the host controller is stable. This is usually confirmed in a system implementation in that all of the clocks in the system are stable before the Arm platform is restarted. So, by definition, if software is running, clocks in the system are stable and the Run/Stop bit in the USB_USBCMD register can be set to one. Minimum system software delays are also defined in the PCI Power Management Specification. Refer to PCI Power Management Specification for more information.

65.4.3.3.1 Port Suspend/Resume

System software places individual ports into suspend mode by writing one into the appropriate USB_PORTSC1 Suspend bit. Software must only set the Suspend bit when the port is in the enabled state (Port Enabled bit is one) and the EHCI is the port owner (PortOwner bit is zero).

The host controller may evaluate the Suspend bit immediately or wait until a micro-frame or frame boundary occurs. If evaluated immediately, the port is not suspended until the current transaction (if one is executing) completes. Therefore, there may be several micro-frames of activity on the port until the host controller evaluates the Suspend bit. The host controller must evaluate the Suspend bit at least every frame boundary.

System software can initiate a resume on a selectively suspended port by writing one to the Force Port Resume bit. Software should not attempt to resume a port unless the port reports that it is in the suspended state (see Port Status & Control (USB_*n*PORTSC1)). If system software sets Force Port Resume bit to one when the port is not in the suspended state, the resulting behavior is undefined. In order to assure proper USB device operation, software must wait for at least 10 ms after a port indicates that it is suspended (Suspend bit is one) before initiating a port resume through the Force Port Resume bit. When Force Port Resume bit is one, the host controller sends resume signaling down the port. System software times the duration of the resume (nominally 20 ms) then sets the Force Port Resume bit to zero. When the host controller receives the write to transition Force Port Resume to zero, it completes the resume sequence as defined in the USB specification, and sets both the Force Port Resume and Suspend bits to zero. Software-initiated port resumes do not affect the Port Change Detect bit in the USB_USBSTS register nor do they cause an interrupt if the Port Change Interrupt Enable bit in the USB_USBINTR register is one. An external USB event may also initiate a resume. The wake events are defined above. When a wake event occurs on a suspended port, the resume signaling is detected by the port and the resume is reflected downstream within 100 µsec. The port's Force Port Resume bit is set to one and the Port Change Detect bit in the USB_USBSTS register is set to one. If the Port Change Interrupt Enable bit in the USB_USBINTR register is one the host controller issues a hardware interrupt.

System software observes the resume event on the port, delays a port resume time (nominally 20 ms), then terminates the resume sequence by writing zero to the Force Port Resume bit in the port. The host controller receives the write of zero to Force Port Resume, terminates the resume sequence and sets Force Port Resume and Suspend port bits to zero. Software can determine that the port is enabled (not suspended) by sampling the USB_PORTSC1 register and observing that the Suspend and Force Port Resume bits are zero. Software must ensure that the host controller is running (that is HCHalted bit in the USB_USBSTS register is zero), before terminating a resume by writing zero to a

port's Force Port Resume bit. If HCHalted is one when Force Port Resume is set to zero, then SOFs do not occur down the enabled port and the device returns to suspend mode in a maximum of 10 msec.

The table below summarizes the wake-up events. Whenever a resume event is detected, the Port Change Detect bit in the USB_USBSTS register is set to one. If the Port Change Interrupt Enable bit is one in the USB_USBINTR register, the host controller generates an interrupt on the resume event. Software acknowledges the resume event interrupt by clearing the Port Change Detect status bit in the USB_USBSTS register.

Table 65-37. Behavior During Wake-up Events

Port Status and Signaling Type	Signaled Port Response	Device State	
		D0	Not D0
Port disabled, resume K-State received	No Effect	N/A	N/A
Port suspended, resume K-State received	Resume reflected downstream on signaled port. Force Port Resume status bit in USB_PORTSC1 register is set to one. Port Change Detect bit in USB_USBSTS register set to one.	[1], [2]	[2]
Port is enabled, disabled or suspended, and the port's WKDSCNNT_E bit is one. A disconnect is detected.	Depending in the initial port state, theUSB_ PORTSC1 Connected Enable status bits are set to zero, and the Connect Change status bit is set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [2]	[2]
Port is enabled, disabled or suspended, and the port's WKDSCNNT_E bit is zero. A disconnect is detected.	Depending on the initial port state, the USB_PORTSC1 Connect and Enable status bits are set to zero, and the Connect Change status bit is set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [3]	[3]
Port is not connected and the port's WKCNNT_E bit is one. A connect is detected.	USB_PORTSC1 Connect Status and Connect Status Change bits are set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [2]	[2]
Port is not connected and the port's WKCNNT_E bit is zero. A connect is detected.	USB_PORTSC1 Connect Status and Connect Status Change bits are set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [3]	[3]
Port is connected and the port's WKOC_E bit is one. An over-current condition occurs.	USB_PORTSC1 Over-current Active, Over-current Change bits are set to one. If Port Enable/Disable bit is one, it is set to zero. Port Change Detect bit in the USB_USBSTS register is set to one	[1], [2]	[2]
Port is connected and the port's WKOC_E bit is zero. An over-current condition occurs.	USB_PORTSC1 Over-current Active, Over-current Change bits are set to one. If Port Enable/Disable bit is one, it is set to zero. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [3]	[3]

[1] Hardware interrupt issued if Port Change Interrupt Enable bit in the USB_USBINTR register is one.

[2] PME# asserted if enabled (Note: PME Status must always be set to one).

[3] PME# not asserted.

65.4.3.4 Schedule Traversal Rules

The host controller executes transactions for devices using a simple, shared-memory schedule.

The schedule is comprised of a few data structures, organized into two distinct lists. The data structures are designed to provide the maximum flexibility required by USB, minimize memory traffic and hardware / software complexity.

System software maintains two schedules for the host controller: a periodic schedule and an asynchronous schedule. The root of the periodic schedule is the USB_PERIODICLISTBASE register (see Frame List Base Address (USB_*n*PERIODICLISTBASE))/ Device Address (USB_*n*DEVICEADDR). The USB_PERIODICLISTBASE register is the physical memory base address of the periodic frame list. The periodic frame list is an array of physical memory pointers. The objects referenced from the frame list must be valid schedule data structures as defined in Host Data Structures. In each micro-frame, if the periodic schedule is enabled (see Periodic scheduling threshold) then the host controller must execute from the periodic schedule before executing from the asynchronous schedule. It only executes from the asynchronous schedule after it encounters the end of the periodic schedule. The host controller traverses the periodic schedule by constructing an array offset reference from the USB_PERIODICLISTBASE and the USB_FRINDEX registers (see the following figure). It fetches the element and begins traversing the graph of linked schedule data structures.

The end of the periodic schedule is identified by a next link pointer of a schedule data structure having its T-bit set to one. When the host controller encounters a T-Bit set to one during a horizontal traversal of the periodic list, it interprets this as an End-Of-Periodic-List mark. This causes the host controller to cease working on the periodic schedule and transitions immediately to traversing the asynchronous schedule. After the transition, the host controller executes from the asynchronous schedule until the end of the micro-frame.

The following figure illustrates the derivation of pointer into frame list array.

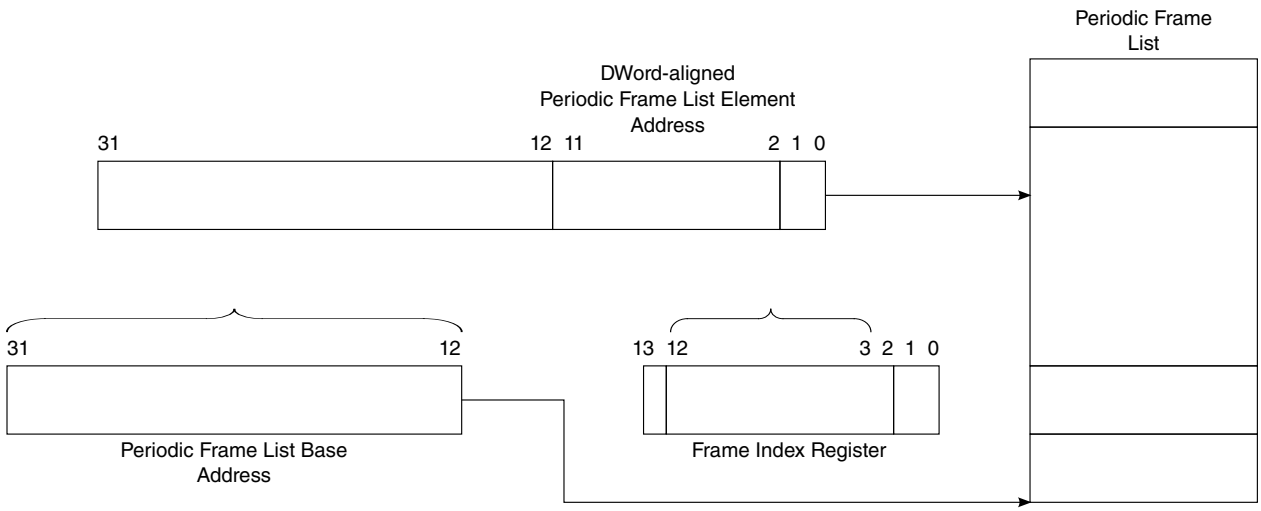


Figure 65-7. Derivation of Pointer into Frame List Array

When the host controller determines that it is the time to execute from the asynchronous list, it uses the operational register USB_ASYNCLISTADDR to access the asynchronous schedule, see the figure below.

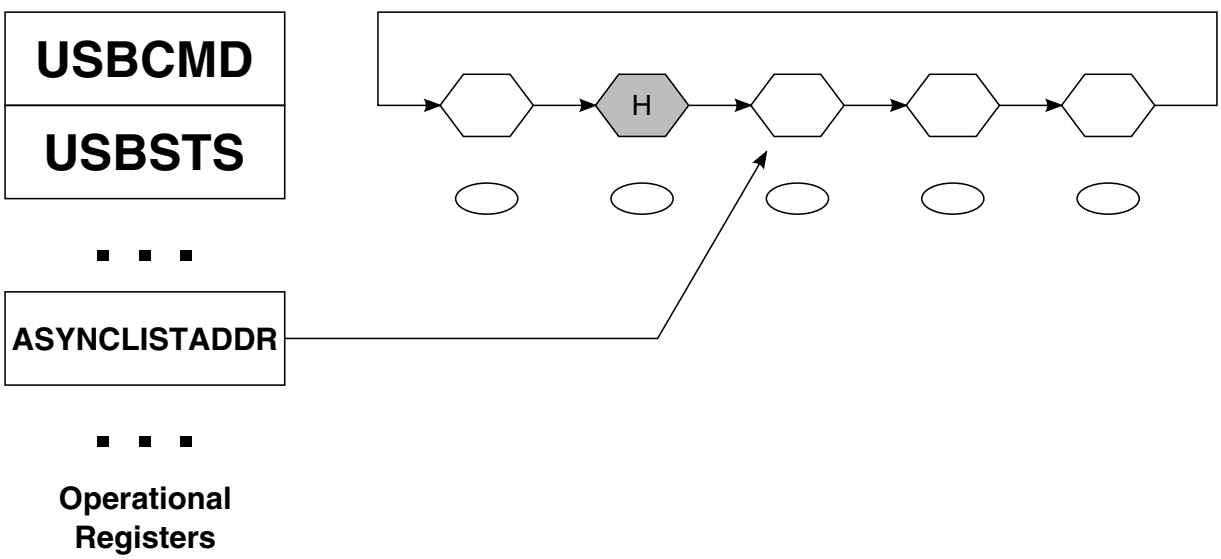


Figure 65-8. General Format of Asynchronous Schedule List

The USB_ASYNCLISTADDR register contains a physical memory pointer to the next queue head. When the host controller makes a transition to executing the asynchronous schedule, it begins by reading the queue head referenced by the USB_ASYNCLISTADDR register. Software must set queue head horizontal pointer T-bits to zero for queue heads in the asynchronous schedule. See Asynchronous Schedule for complete operational details.

65.4.3.4.1 Example - Preserving Micro-Frame Integrity

One of the requirements of a USB host controller is to maintain Frame Integrity. This means that the HC must preserve the micro-frame boundaries.

For example, SOF packets must be generated on time (within the specified allowable jitter), and High-speed EOF1,2 thresholds must be enforced. The end of micro-frame timing points EOF1 and EOF2 are clearly defined in the USB Specification Revision 2.0. One implication of this responsibility is that the HC must ensure that it does not start transactions that do not complete before the end of the micro-frame. More precisely, no transactions should be started by the host controller, which do not complete in their entirety before the EOF1 point. In order to enforce this rule, the host controller must check each transaction before it starts to ensure that it completes before the end of the micro-frame.

So, what exactly needs to be involved in this check? Fundamentally, the transaction data payload, plus bit stuffing, plus transaction overhead must be taken into consideration. It is possible to be extremely accurate on how much time the next transaction takes. Take OUTs for an example. The host controller must fetch all of the OUT data from memory in order to send it onto the USB bus. A host controller implementation could pre-fetch all of the OUT data, and pre-compute the actual number of bits in the token and data packets. In addition, the system knows the depth of the target endpoint, so it could closely estimate turnaround time for handshake. In addition, the host controller knows the size of a handshake packet. Pre-computing effects of bit stuffing and summing up the other overhead numbers can allow the host controller to know exactly whether there is enough bus time, before EOF1 to complete the OUT transaction. To accomplish this particular approach takes an inordinate amount of time and hardware complexity.

The alternative is to make a reasonable guess whether the next transaction can be started. An example approximation algorithm is described below. This example algorithm relies on the EHCI policy that periodic transactions are scheduled first in the micro-frame. It is a reasonable assumption that software never over-commits the micro-frame to periodic transactions greater than the specification allowable 80%. In the available remaining 20% bandwidth, the host controller has some ability (in this example) to decide whether or not to execute a transaction. The result of this algorithm is that sometimes, under some circumstances a transaction is not executed that could have been executed. However, under all circumstances, a transaction is never started unless there is enough time in the frame to complete the transaction.

65.4.3.4.1.1 Transaction Fit - A Best-Fit Approximation Algorithm

A curve is calculated which represents the latest start time for every packet size, at which software schedules the start of a periodic transaction.

This curve is the 80% bandwidth curve. Another curve is calculated which is the absolute, latest permitted start time for every packet size. This curve represents the absolute latest time, that a transaction of each packet size can be started and completed, in the micro-frame. A plot of these two curves are illustrated in Figure 65-9. The plot Y-axis represents the number of byte-times left in a frame.

The space between the 80% and the Last Start plots is bandwidth reclamation area. In this algorithm the host controller may skip transactions during this time if it is prudent.

The Best-Fit Approximation method plots a function ($f(x)$) between the 80% and Last Start curves. The function $f(x)$ adds a constant to every transaction's maximum packet size and the result compared with the number of bytes left in the frame. The constant represents an approximation of the effects of bit stuffing and protocol overhead. The host controller starts transactions whose results land above the function curve. The host controller will not start transactions whose results land below the function curve.

The following figure illustrates the Best-Fit Approximation.

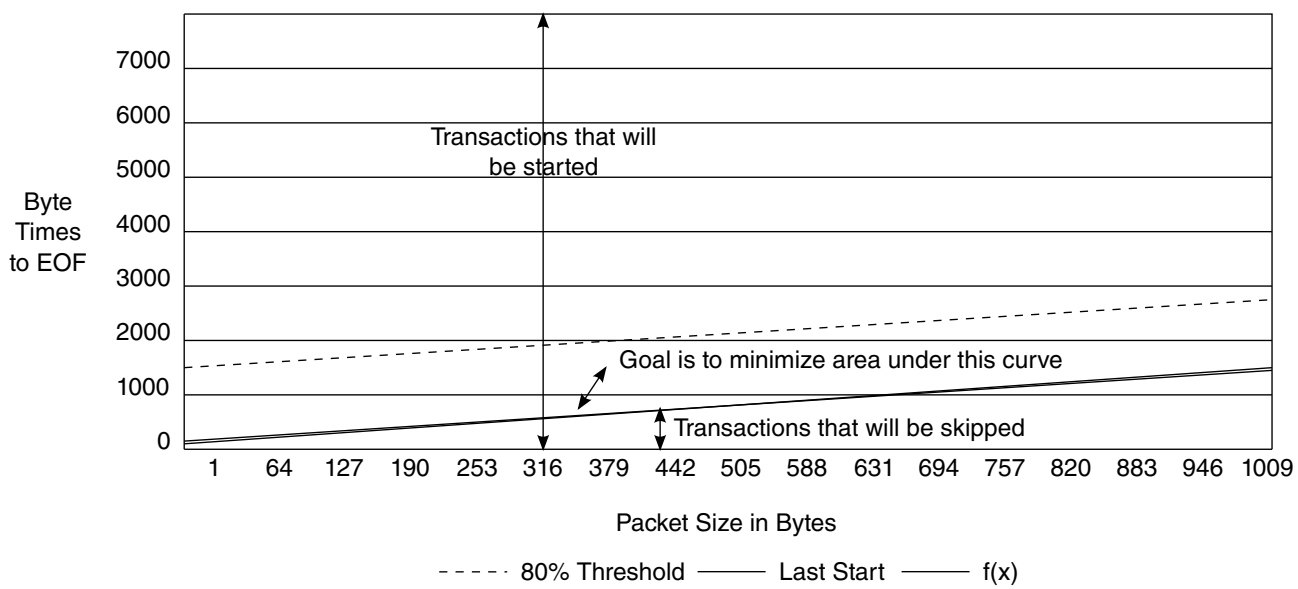


Figure 65-9. Best Fit Approximation

The LastStart line was calculated in this example to assume the absolute worst-case bus overhead per transaction. The particular transaction used is a start-split, zero-length OUT transaction with a handshake. Summaries of the component parts are listed in the table below. The component times were derived from the protocol timings defined in the USB Specification Revision 2.0.

Table 65-38. Example Worse-case Transaction Timing Components

Component	Bit time	Byte Time	Explanation

Table continues on the next page...

Table 65-38. Example Worse-case Transaction Timing Components (continued)

Split Token	76	9.5	Split token as defined in USB core specification. Includes sync, token, eop, and so on.
Host 2 Host IPG	88	11	Number of bit times required between consecutive host packets.
Token	67	8.375	Token as defined in USB core specification. Includes sync, token, eop, and so on.
Host 2 Host IPG	88	11	Token as defined in USB core specification. Includessync, token, eop, and so on.
Data Packet (0 data bytes)	66.7	8.34	Zero-length data packet. Includes sync, PID, crc16, eop, and so on.
Turnaround time	721	90.125	Time for packet initiator (Host) to see the beginning of a response to a transmitted packet.
Handshake packet	48	6	Handshake packet as defined in USB core specification. Includes sync, PID, eop, and so on.
		144	Total

The exact details of the function (f(x)) are up to the particular implementation. However, it should be obvious that the goal is to minimize the area under the curve between the approximation function and the Last Start curve, without dipping below the LastStart line, while at the same time keeping the check as simple as possible for hardware implementation. The f(x) in Figure 65-9 was constructed using the following pseudo-code test on each transaction size data point. This algorithm assumes that the host controller keeps track of the remaining bits in the frame.

```
Alorithm CheckTransactionWillFit (MaximumPacketSize, HC_BytesLeftInFrame)
Begin
Local Temp = MaximumPacketSize + 192
Local rvalue = TRUE
If MaximumPacketSize >= 128 then
     Temp += 128
End If
If Temp > HC_BytesLeftInFrame then
     Rvalue = FALSE
End If
Return rvalue
End
```

This algorithm takes two inputs, the current maximum packet size of the transaction and the hardware counter of the number of bytes left in the current micro-frame. It unconditionally adds a simple constant of 192 to the maximum packet size to account for a first-order effect of transaction overhead and bit stuffing. If the transaction size is greater than or equal to 128 bytes, then an additional constant of 128 is added to the running sum to account for the additional worst-case bit stuffing of payloads larger than 128. An inflection point was inserted at 128 because the $f(x)$ plot was getting close to the LastStart line.

65.4.3.5 Periodic Schedule Frame Boundaries vs Bus Frame Boundaries

The USB Specification Revision 2.0 requires that the frame boundaries (SOF frame number changes) of the high-speed bus and the full- and low-speed bus(s) below USB 2.0 Hubs be strictly aligned.

Super-imposed on this requirement is that USB 2.0 Hubs manage full- and low-speed transactions through a micro-frame pipeline (see start- (SS) and complete- (CS) splits illustrated in the following figure). A simple, direct projection of the frame boundary model into the host controller interface schedule architecture creates tension (complexity for both hardware and software) between the frame boundaries and the scheduling mechanisms required to service the full- and low-speed transaction translator periodic pipelines.

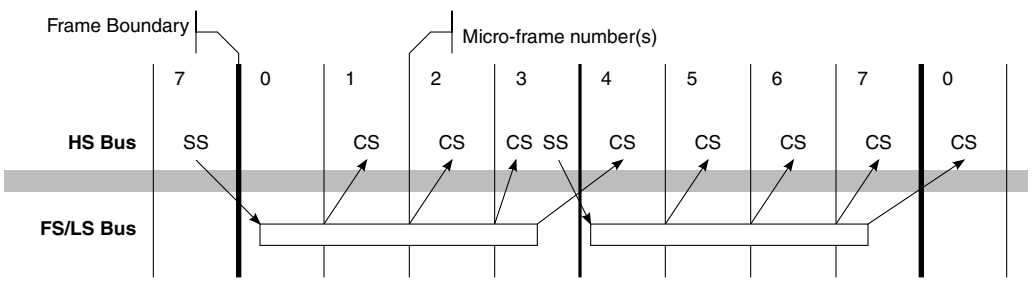


Figure 65-10. Frame Boundary Relationship between HS bus and FS/LS Bus

The simple projection, as the above figure illustrates, introduces frame-boundary wrap conditions for scheduling on both the beginning and end of a frame. In order to reduce the complexity for hardware and software, the host controller is required to implement one micro-frame phase shift for its view of frame boundaries. The phase shift eliminates the beginning of frame and frame-wrap scheduling boundary conditions.

The implementation of this phase shift requires that the host controller use one register value for accessing the periodic frame list and another value for the frame number value included in the SOF token. These two values are separate, but tightly coupled. The periodic frame list is accessed through the Frame List Index Register (USB_FRINDEX) documented in USB Frame Index (USB_*n*FRINDEX) and initially illustrated in Schedule Traversal Rules. Bits FRINDEX[2:0], represent the micro-frame number. The SOF value is coupled to the value of FRINDEX[13:3]. Both FRINDEX[13:3] and the SOF value are increment based on FRINDEX[2:0]. It is required that the SOF value be delayed from the FRINDEX value by one micro-frame. The one micro-frame delay yields host controller periodic schedule and bus frame boundary relationship as illustrated in the following figure. This adjustment allows software to trivially schedule the periodic start and

complete-split transactions for full-and low-speed periodic endpoints, using the natural alignment of the periodic schedule interface. The reasons for selecting this phase-shift are beyond the scope of this specification.

The following figure illustrates how periodic schedule data structures relate to schedule frame boundaries and bus frame boundaries. To aid the presentation, two terms are defined: The host controller's view of the 1 msec boundaries is called H-Frames. The high-speed bus's view of the 1 msec boundaries is called B-Frames.

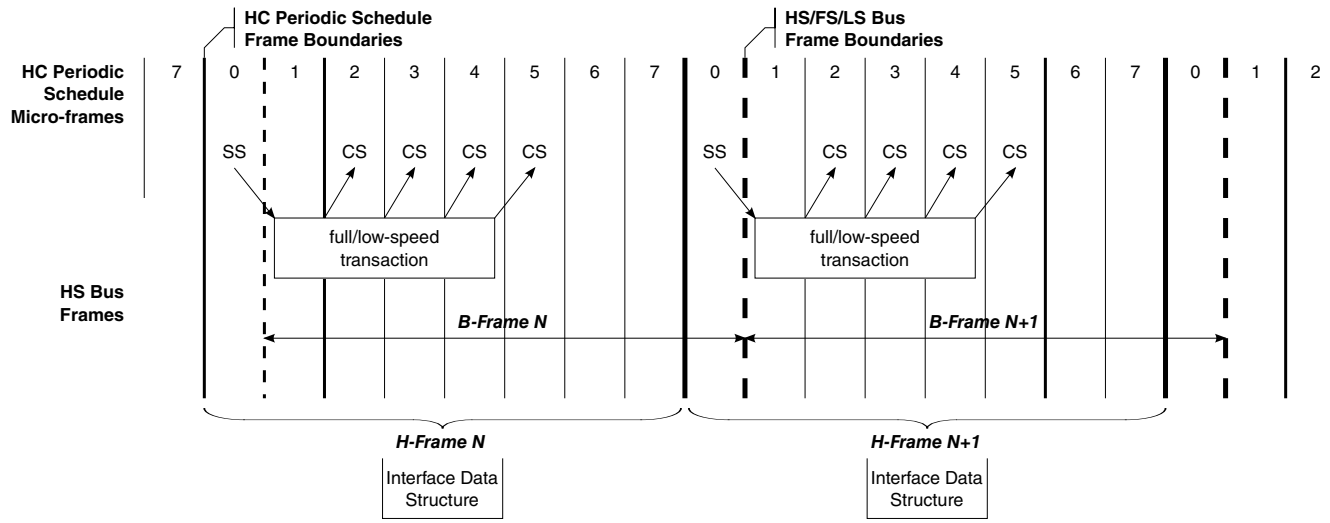


Figure 65-11. Relationship of Periodic Schedule Frame Boundaries to Bus Frame Boundaries

H-Frame boundaries for the host controller correspond to increments of FRINDEX[13:3]. Micro-frame numbers for the H-Frame are tracked by FRINDEX[2:0]. B-Frame boundaries are visible on the high-speed bus through changes in the SOF token's frame number. Micro-frame numbers on the high-speed bus are only derived from the SOF token's frame number (that is the high-speed bus sees eight SOFs with the same frame number value). H-Frames and B-Frames have the fixed relationship (that is B-Frames lag H-Frames by one micro-frame time) illustrated in the figure above. The host controller's periodic schedule is naturally aligned to H-Frames. Software schedules transactions for full- and low-speed periodic endpoints relative the H-Frames. The result is these transactions execute on the high-speed bus at exactly the right time for the USB 2.0 Hub periodic pipeline. As described in USB Frame Index (USB_*n*FRINDEX), the SOF Value can be implemented as a shadow register (in this example, called SOFV), which lags the FRINDEX register bits [13:3] by one micro-frame count. This lag behavior can be accomplished by incrementing FRINDEX[13:3] based on carry-out on the 7 to 0 increment of FRINDEX[2:0] and incrementing SOFV based on the transition of 0 to 1 of FRINDEX[2:0].

Software is allowed to write to FRINDEX. USB Frame Index (USB_*n*FRINDEX) provides the requirements that software should adhere when writing a new value in FRINDEX.

The table below illustrates the required relationship between the value of FRINDEX and the value of SOFV.

Table 65-39. Operation of FRINDEX and SOFV (SOF Value Register)

Current			Next		
FRINDEX[F]	SOFV	FRINDEX[mF]	FRINDEX[F]	SOFV	FRINDEX[mF]
N	N	111b	N+1	N	000b
N+1	N	000b	N+1	N+1	001b
N+1	N+1	001b	N+1	N+1	010b
N+1	N+1	010b	N+1	N+1	011b
N+1	N+1	011b	N+1	N+1	100b
N+1	N+1	100b	N+1	N+1	101b
N+1	N+1	101b	N+1	N+1	110b
N+1	N+1	110b	N+1	N+1	111b

NOTE

Where [F] = [13:3]; [µF] = [2:0]

65.4.3.6 Periodic Schedule

The periodic schedule traversal is enabled or disabled through the Periodic Schedule Enable bit in the USB_USBCMD register. If the Periodic Schedule Enable bit is set to zero, then the host controller simply does not try to access the periodic frame list through the USB_PERIODICLISTBASE register. Likewise, when the Periodic Schedule Enable bit is one, then the host controller does use the USB_PERIODICLISTBASE register to traverse the periodic schedule.

The host controller will not react to modifications to the Periodic Schedule Enable immediately. In order to eliminate conflicts with split transactions, the host controller evaluates the Periodic Schedule Enable bit only when FRINDEX[2:0] is zero. System software must not disable the periodic schedule if the schedule contains an active split transaction work item that spans the 000b micro-frame. These work items must be removed from the schedule before the Periodic Schedule Enable bit is written to zero.

The Periodic Schedule Status bit in the USB_USBSTS register indicates status of the periodic schedule. System software enables (or disables) the periodic schedule by writing one (or zero) to the Periodic Schedule Enable bit in the USB_USBCMD register. Software then can poll the Periodic Schedule Status bit to determine when the periodic

schedule has made the desired transition. Software must not modify the Periodic Schedule Enable bit unless the value of the Periodic Schedule Enable bit equals that of the Periodic Schedule Status bit.

The periodic schedule is used to manage all isochronous and interrupt transfer streams. The base of the periodic schedule is the periodic frame list. Software links schedule data structures to the periodic frame list to produce a graph of scheduled data structures. The graph represents an appropriate sequence of transactions.

The following figure illustrates isochronous transfers (using iTDs and siTDs) with a period of one are linked directly to the periodic frame list. Interrupt transfers (are managed with queue heads) and isochronous streams with periods other than one are linked following the period-one iTD/siTDs. Interrupt queue heads are linked into the frame list ordered by poll rate. Longer poll rates are linked first (for example, closest to the periodic frame list), followed by shorter poll rates, with queue heads with a poll rate of one, on the very end.

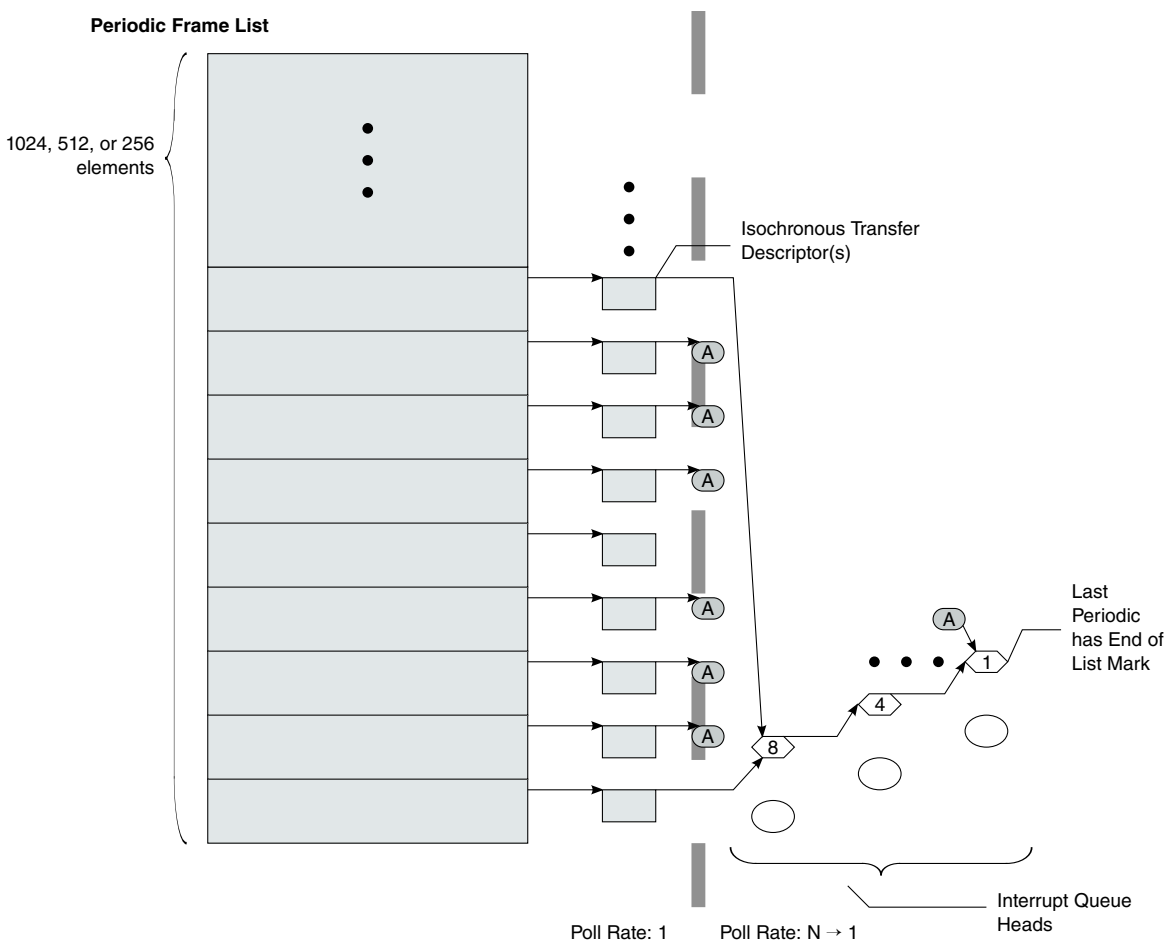


Figure 65-12. Example Periodic Schedule

65.4.3.7 Managing Isochronous Transfers Using iTDs

The structure of an iTD is presented in Isochronous (High-Speed) Transfer Descriptor (iTD). The four distinct sections to an iTD:

- The first field is the Next Link Pointer. This field is for schedule linkage purposes only.
- Transaction description array. This area is an eight-element array. Each element represents control and status information for one micro-frame's worth of transactions for a single high-speed isochronous endpoint.
- The buffer page pointer array is a 7-element array of physical memory pointers to data buffers. These are 4 K aligned pointers to physical memory.
- Endpoint capabilities. This area utilizes the unused low-order 12 bits of the buffer page pointer array. The fields in this area are used across all transactions executed for this iTD, including endpoint addressing, transfer direction, maximum packet size and high-bandwidth multiplier.

65.4.3.7.1 Host Controller Operational Model for iTDs

The host controller uses FRINDEX register bits [12:3] to index into the periodic frame list. This means that the host controller visits each frame list element eight consecutive times before incrementing to the next periodic frame list element. Each iTD contains eight transaction descriptions, which map directly to FRINDEX register bits [2:0]. Therefore, each transaction descriptor corresponds to one micro-frame. Each iTD can span 8 micro-frames worth of transactions.

When the host controller fetches an iTD, it uses FRINDEX register bits [2:0] to index into the transaction description array.

If the active bit in the Status field of the indexed transaction description is set to zero, the host controller ignores the iTD and follows the Next pointer to the next schedule data structure.

When the indexed active bit is one, the host controller continues to parse the iTD. It stores the indexed transaction description and the general endpoint information (device address, endpoint number, maximum packet size, and so on.). It also uses the Page Select (PG) field to index the buffer pointer array, storing the selected buffer pointer and the next sequential buffer pointer. For example, if PG field is 0, then the host controller stores Page 0 and Page 1.

The host controller constructs a physical data buffer address by concatenating the current buffer pointer (as selected using the current transaction description's PG field) and the transaction description's Transaction Offset field. The host controller uses the endpoint addressing information and I/O-bit to execute a transaction to the appropriate endpoint. When the transaction is complete, the host controller clears the active bit and writes back any additional status information to the Status field in the currently selected transaction description.

The data buffer associated with the iTD must be virtually contiguous memory. Seven page pointers are provided to support eight high-bandwidth transactions regardless of the starting packet's offset alignment into the first page. A starting buffer pointer (physical memory address) is constructed by concatenating the page pointer (for example, page 0 pointer) selected by the active transaction descriptions' PG (for example, value: 00B) field with the transaction offset field. As the transaction moves data, the host controller must detect when an increment of the current buffer pointer crosses a page boundary. When this occurs the host controller simply replaces the current buffer pointer's page portion with the next page pointer (for example, page 1 pointer) and continues to move data. The size of each bus transaction is determined by the value in the Maximum Packet Size field. An iTD supports high-bandwidth pipes through the Mult (multiplier) field. When the Mult field is 1, 2, or 3, the host controller executes the specified number of Maximum Packet sized bus transactions for the endpoint in the current micro-frame. In other words, the Mult field represents a transaction count for the endpoint in the current micro-frame. If the Mult field is zero, the operation of the host controller is undefined. The transfer description is used to service all transactions indicated by the Mult field.

For OUT transfers, the value of the Transaction X Length field represents the total bytes to be sent during the micro-frame. The Mult field must be set by software to be consistent with Transaction X Length and Maximum Packet Sixe. The host controller sends the bytes in Maximum Packet Size'd portions. After each transaction, the host controller decrements its local copy of Transaction X Length by Maximum Packet Size. The number of bytes the host controller sends is always Maximum Packet Size or Transaction X Length, whichever is less. The host controller advances the transfer state in the transfer description, updates the appropriate record in the iTD and moves to the next schedule data structure. The maximum sized transaction supported is 3 x 1024 bytes.

For IN transfers, the host controller issues Mult transactions. It is assumed that software has properly initialized the iTD to accommodate all of the possible data. During each IN transaction, the host controller must use Maximum Packet Size to detect packet babble errors. The host controller keeps the sum of bytes received in the Transaction X Length field. After all transactions for the endpoint have completed for the micro-frame, Transaction X Length contains the total bytes received. If the final value of Transaction X Length is less than the value of Maximum Packet Size, then less data than was allowed for was received from the associated endpoint. This short packet condition does not set

the USBINT bit in the USB_USBSTS register to one. The host controller will not detect this condition. If the device sends more than Transaction X Length or Maximum Packet Size bytes (whichever is less), then the host controller sets the Babble Detected bit to one and set the Active bit to zero. Note, that the host controller is not required to update the iTD field Transaction X Length in this error scenario. If the Mult field is greater than one, then the host controller automatically executes the value of Mult transactions. The host controller will not execute all Mult transactions if:

- The endpoint is an OUT and Transaction X Length goes to zero before all the Mult transactions have executed (ran out of data), or
- The endpoint is an IN and the endpoint delivers a short packet, or an error occurs on a transaction before Mult transactions have been executed. The end of micro-frame may occur before all of the transaction opportunities have been executed. When this happens, the transfer state of the transfer description is advanced to reflect the progress that was made, the result written back to the iTD and the host controller proceeds to processing the next micro-frame. Refer to Appendix D for a table summary of the host controller required behavior for all the high-bandwidth transaction cases.

65.4.3.7.2 Software Operational Model for iTDs

A client buffer request to an isochronous endpoint may span 1 to N micro-frames. When N is larger than one, system software may have to use multiple iTDs to read or write data with the buffer (if N is larger than eight, it must use more than one iTD).

The following figure illustrates the simple model of how a client buffer is mapped by system software to the periodic schedule (that is the periodic frame list and a set of iTDs). On the right is the client description of its request. The description includes a buffer base address plus additional annotations to identify which portions of the buffer should be used with each bus transaction. In the middle is the iTD data structures used by the system software to service the client request. Each iTD can be initialized to service up to 24 transactions, organized into eight groups of up to three transactions each. Each group maps to one micro-frame's worth of transactions. The EHCI controller does not provide per-transaction results within a micro-frame. It treats the per-micro-frame transactions as a single logical transfer. On the left is the host controller's frame list. System software establishes references from the appropriate locations in the frame list to each of the appropriate iTDs. If the buffer is large, then system software can use a small set of iTDs to service the entire buffer. System software can activate the transaction description records (contained in each iTD) in any pattern required for the particular data stream.

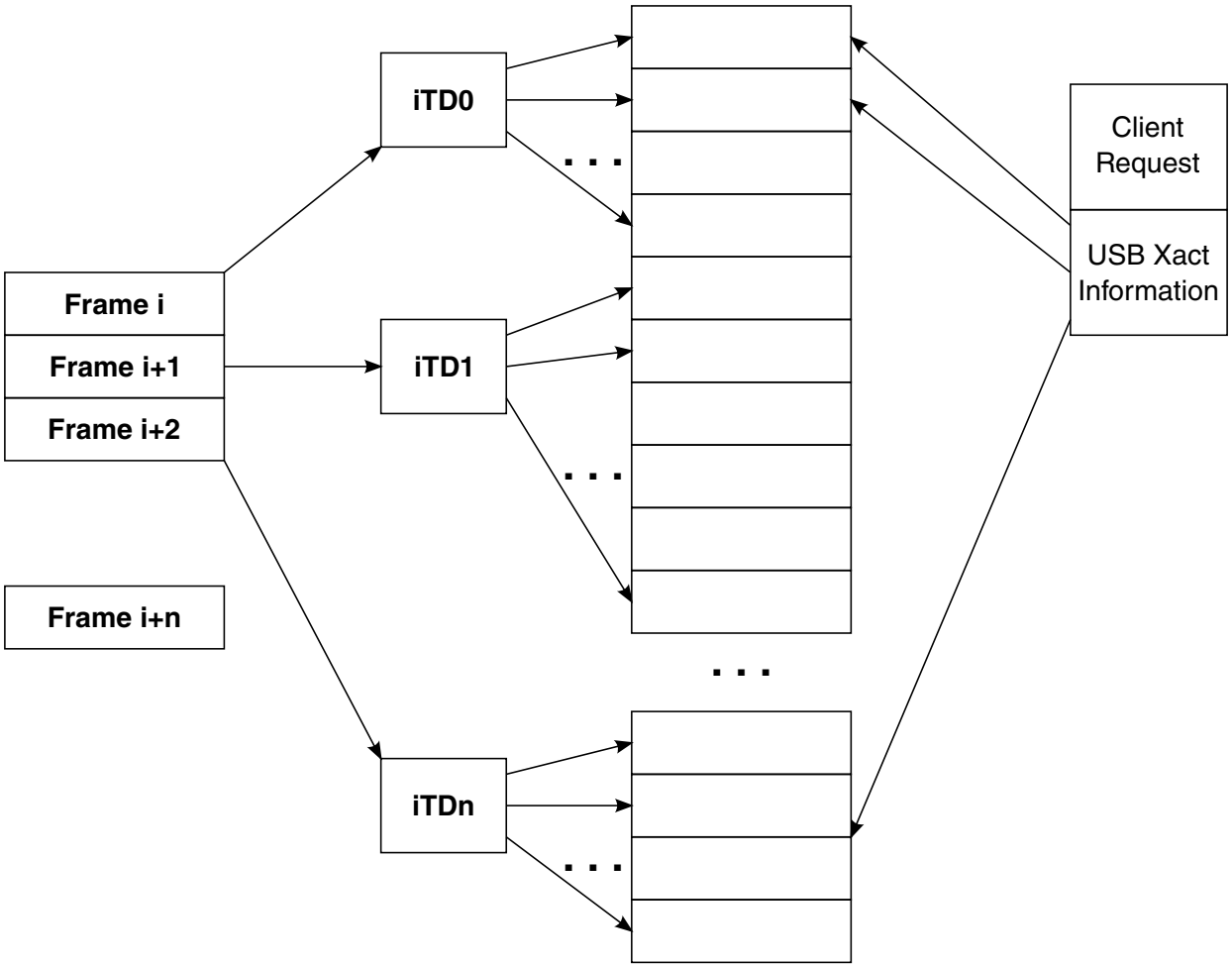


Figure 65-13. Example Association of iTDs to Client Request Buffer

As noted above, the client request includes a pointer to the base of the buffer and offsets into the buffer to annotate which buffer sections are to be used on each bus transaction that occurs on this endpoint. System software must initialize each transaction description in an iTD to ensure it uses the correct portion of the client buffer. For example, for each transaction description, the PG field is set to index the correct physical buffer page pointer and the Transaction Offset field is set relative to the correct buffer pointer page (for example, the same one referenced by the PG field). When the host controller executes a transaction it selects a transaction description record based on FRINDEX[2:0]. It then uses the current Page Buffer Pointer (as selected by the PG field) and concatenates to the transaction offset field. The result is a starting buffer address for the transaction. As the host controller moves data for the transaction, it must watch for a page wrap condition and properly advance to the next available Page Buffer Pointer. System software must not use the Page 6 buffer pointer in a transaction description where the length of the transfer wraps a page boundary. Doing so yields undefined behavior. The host controller hardware is not required to 'alias' the page selector to Page zero. USB 2.0 isochronous

endpoints can specify a period greater than one. Software can achieve the appropriate scheduling by linking iTDs into the appropriate frames (relative to the frame list) and by setting appropriate transaction description elements active bits to one.

65.4.3.7.2.1 Periodic scheduling threshold

The Isochronous Scheduling Threshold field in the USB_HCCPARAMS capability register is an indicator to system software as to how the host controller pre-fetches and effectively caches schedule data structures.

It is used by system software when adding isochronous work items to the periodic schedule. The value of this field indicates to system software the minimum distance it can update isochronous data (relative to the current location of the host controller execution in the periodic list) and still have the host controller process them.

The iTD and siTD data structures each describe 8 micro-frames worth of transactions. The host controller is allowed to cache one (or more) of these data structures in order to reduce memory traffic. Three basic caching models that account for the fact the isochronous data structures span 8 micro-frames. The three caching models are: no caching, micro-frame caching and frame caching.

When software is adding new isochronous transactions to the schedule, it always performs a read of the USB_FRINDEX register to determine the current frame and micro-frame the host controller is currently executing. Of course, there is no information about where in the micro-frame the host controller is, so a constant uncertainty-factor of one micro-frame has to be assumed. Combining the knowledge of where the host controller is executing with the knowledge of the caching model allows the definition of simple algorithms for how closely software can reliably work to the executing host controller.

No caching is indicated with a value of zero in the Isochronous Scheduling Threshold field. The host controller may pre-fetch data structures during a periodic schedule traversal (per micro-frame) but always dumps any accumulated schedule state at the end of the micro-frame. At the appropriate time relative to the beginning of every micro-frame, the host controller always begins schedule traversal from the frame list. Software can use the value of the USB_FRINDEX register (plus the constant 1 uncertainty-factor) to determine the approximate position of the executing host controller. When no caching is selected, software can add an isochronous transaction as near as 2 micro-frames in front of the current executing position of the host controller.

Frame caching is indicated with a non-zero value in bit [7] of the Isochronous Scheduling Threshold field. In the frame-caching model, system software assumes that the host controller caches one (or more) isochronous data structures for an entire frame (8 micro-frames). Software uses the value of the USB_FRINDEX register (plus the constant 1

uncertainty) to determine the current micro-frame/frame (assume modulo 8 arithmetic in adding the constant 1 to the micro-frame number). For any current frame N, if the current micro-frame is 0 to 6, then software can safely add isochronous transactions to Frame N + 1. If the current micro-frame is 7, then software can add isochronous transactions to Frame N + 2.

Micro-frame caching is indicated with a non-zero value in the least-significant 3 bits of the Isochronous Scheduling Threshold field. System software assumes the host controller caches one or more periodic data structures for the number of micro-frames indicated in the Isochronous Scheduling Threshold field. For example, if the count value were 2, then the host controller keeps a window of 2 micro-frames worth of state (current micro-frame, plus the next) on-chip. On each micro-frame boundary, the host controller releases the current micro-frame state and begins accumulating the next micro-frame state.

65.4.3.8 Asynchronous Schedule

The Asynchronous schedule traversal is enabled or disabled through the Asynchronous Schedule Enable bit in the USB_USBCMD register.

If the Asynchronous Schedule Enable bit is set to zero, then the host controller simply does not try to access the asynchronous schedule through the USB_ASYNCLISTADDR register. Likewise, when the Asynchronous Schedule Enable bit is one, then the host controller does use the USB_ASYNCLISTADDR register to traverse the asynchronous schedule. Modifications to the Asynchronous Schedule Enable bit are not necessarily immediate. Rather the new value of the bit is taken into consideration the next time the host controller needs to use the value of the USB_ASYNCLISTADDR register to get the next queue head.

The Asynchronous Schedule Status bit in the USB_USBSTS register indicates status of the asynchronous schedule. System software enables (or disables) the asynchronous schedule by writing one (or zero) to the Asynchronous Schedule Enable bit in the USB_USBCMD register. Software then can poll the Asynchronous Schedule Status bit to determine when the asynchronous schedule has made the desired transition. Software must not modify the Asynchronous Schedule Enable bit unless the value of the Asynchronous Schedule Enable bit equals that of the Asynchronous Schedule Status bit.

The asynchronous schedule is used to manage all Control and Bulk transfers. Control and Bulk transfers are managed using queue head data structures. The asynchronous schedule is based at the USB_ASYNCLISTADDR register. The default value of the USB_ASYNCLISTADDR register after reset is undefined and the schedule is disabled when the Asynchronous Schedule Enable bit is zero.

Software may only write this register with defined results when the schedule is disabled. For example, Asynchronous Schedule Enable bit in the USB_USBCMD and the Asynchronous Schedule Status bit in the USB_USBSTS register are zero. System software enables execution from the asynchronous schedule by writing a valid memory address (of a queue head) into this register. Then software enables the asynchronous schedule by setting the Asynchronous Schedule Enable bit to one. The asynchronous schedule is actually enabled when the Asynchronous Schedule Status bit is one.

When the host controller begins servicing the asynchronous schedule, it begins by using the value of the USB_ASYNCLISTADDR register. It reads the first referenced data structure and begins executing transactions and traversing the linked list as appropriate. When the host controller completes processing the asynchronous schedule, it retains the value of the last accessed queue head's horizontal pointer in the USB_ASYNCLISTADDR register. Next time the asynchronous schedule is accessed, this is the first data structure that is serviced. This provides round-robin fairness for processing the asynchronous schedule.

A host controller completes processing the asynchronous schedule when one of the following events occur:

- The end of a micro-frame occurs.
- The host controller detects an empty list condition (see Empty Asynchronous Schedule Detection)
- The schedule has been disabled through the Asynchronous Schedule Enable bit in the USB_USBCMD register.

The queue heads in the asynchronous list are linked into a simple circular list as shown in Figure 65-8. Queue head data structures are the only valid data structures that may be linked into the asynchronous schedule. An isochronous transfer descriptor (iTD or siTD) in the asynchronous schedule yields undefined results.

The maximum packet size field in a queue head is sized to accommodate the use of this data structure for all non-isochronous transfer types. The USB Specification, Revision 2.0 specifies the maximum packet sizes for all transfer types and transfer speeds. System software should always parameterize the queue head data structures according to the core specification requirements.

65.4.3.8.1 Adding Queue Heads to Asynchronous Schedule

This is a software requirement section.

There are two independent events for adding queue heads to the asynchronous schedule. The first is the initial activation of the asynchronous list. The second is inserting a new queue head into an activated asynchronous list.

Activation of the list is simple. System software writes the physical memory address of a queue head into the USB_ASYNCLISTADDR register, then enables the list by setting the Asynchronous Schedule Enable bit in the USB_USBCMD register to one.

When inserting a queue head into an active list, software must ensure that the schedule is always coherent from the host controllers' point of view. This means that the system software must ensure that all queue head pointer fields are valid. For example, qTD pointers have T-Bits set to one or reference valid qTDs and the Horizontal Pointer references a valid queue head data structure. The following algorithm represents the functional requirements:

```
InsertQueueHead (pQHeadCurrent, pQueueHeadNew)
          --
          -- Requirement: all inputs must be properly initialized.
          --
          -- pQHeadCurrent is a pointer to a queue head that is
          -- already in the active list
          -- pQHeadNew is a pointer to the queue head to be added
          --
          -- This algorithm links a new queue head into a existing
          -- list
          --
          pQueueHeadNew.HorizontalPointer  =  pQueueHeadCurrent.HorizontalPointer
          pQueueHeadCurrent.HorizontalPointer =  physicalAddressOf(pQueueHeadNew)
End InsertQueueHead
```

65.4.3.8.2 Removing Queue Heads from Asynchronous Schedule

This is a software requirement section.

There are two independent events for removing queue heads from the asynchronous schedule. The first is shutting down (deactivating) the asynchronous list. The second is extracting a single queue head from an activated list.

Software deactivates the asynchronous schedule by setting the Asynchronous Schedule Enable bit in the USB_USBCMD register to zero. Software can determine when the list is idle when the Asynchronous Schedule Status bit in the USB_USBSTS register is zero. The normal mode of operation is that software removes queue heads from the asynchronous schedule without shutting it down. Software must not remove an active queue head from the schedule. Software should first deactivate all active qTDs, wait for the queue head to go inactive, then remove the queue head from the asynchronous list. Software removes a queue head from the asynchronous list through the following algorithm. As illustrated, the unlinking is quite easy. Software merely must ensure all of the link pointers reachable by the host controller are kept consistent.

```
UnlinkQueueHead (pQHeadPrevious, pQueueHeadToUnlink, pQHeadNext)
          --
          -- Requirement: all inputs must be properly initialized.
          --
          -- pQHeadPrevious is a pointer to a queue head that
          -- references the queue head to remove
          -- pQHeadToUnlink is a pointer to the queue head to be
```

```
        -- removed
        -- pQheadNext is a pointer to a queue head still in the
        -- schedule. Software provides this pointer with the
        -- following strict rules:
        --          if the host software is one queue head, then
        --          pQHeadNext must be the same as
        --          QueueheadToUnlink.HorizontalPointer. If the host
        --          software is unlinking a consecutive series of
        --          queue heads, QHeadNext must be set by software to
        --          the  queue head remaining in the schedule.
        --
        -- This algorithm unlinks a queue head from a circular list
        --
        pQueueHeadPrevious.HorizontalPointer =  pQueueHeadToUnlink.HorizontalPointer
                          pQueueHeadToUnlink.HorizontalPointer = pQHeadNext
End UnlinkQueueHead
```

If software removes the queue head with the H-bit set to one, it must select another queue head still linked into the schedule and set its H-bit to one. This should be completed before removing the queue head. The requirement is that software keep one queue head in the asynchronous schedule, with its H-bit set to one. At the point software has removed one or more queue heads from the asynchronous schedule, it is unknown whether the host controller has a cached pointer to them. Similarly, it is unknown how long the host controller might retain the cached information, as it is implementation dependent and may be affected by the actual dynamics of the schedule load. Therefore, once software has removed a queue head from the asynchronous list, it must retain the coherency of the queue head (link pointers, and so on). It cannot disturb the removed queue heads until it knows that the host controller does not have a local copy of a pointer to any of the removed data structures.

The method software uses to determine when it is safe to modify a removed queue head is to handshake with the host controller. The handshake mechanism allows software to remove items from the asynchronous schedule, then execute a simple, lightweight handshake that is used by software as a key that it can free (or reuse) the memory associated the data structures it has removed from the asynchronous schedule.

The handshake is implemented with three bits in the host controller. The first bit is a command bit (Interrupt on Async Advance Doorbell bit in the USB_USBCMD register) that allows software to inform the host controller that something has been removed from its asynchronous schedule. The second bit is a status bit (Interrupt on Async Advance bit in the USB_USBSTS register) that the host controller sets after it has released all on-chip state that may potentially reference one of the data structures just removed. When the host controller sets this status bit to one, it also sets the command bit to zero. The third bit is an interrupt enable (Interrupt on Async Advance bit in the USB_USBINTR register) that is matched with the status bit. If the status bit is one and the interrupt enable bit is one, then the host controller asserts a hardware interrupt.

The figure below illustrates a general example. In this example, consecutive queue heads (B and C) are unlinked from the schedule using the algorithm above. Before the unlink operation, the host controller has a copy of queue head A.

The unlink algorithm requires that as software unlinks each queue head, the unlinked queue head is loaded with the address of a queue head that remains in the asynchronous schedule.

When the host controller observes that doorbell bit being set to one, it makes a note of the local reachable schedule information. In this example, the local reachable schedule information includes both queue heads (A and B). It is sufficient that the host controller can set the status bit (and clear the doorbell bit) as soon as it has traversed beyond current reachable schedule information (that is traversed beyond queue head (B) in this example). The following figure illustrates the generic queue head unlink scenario.

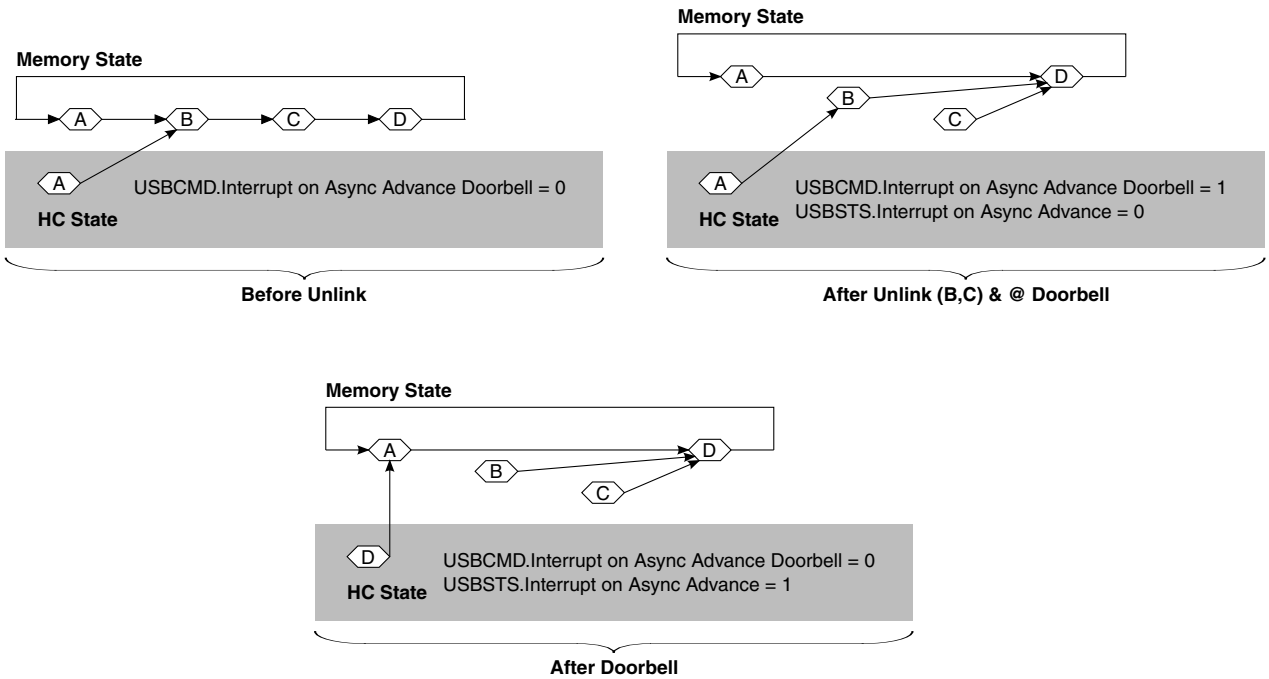


Figure 65-14. Generic Queue Head Unlink Scenario

Alternatively, a host controller implementation is allowed to traverse the entire asynchronous schedule list (for example, observed the head of the queue (twice)) before setting the Advance on Async status bit to one.

Software may re-use the memory associated with the removed queue heads after it observes the Interrupt on Async Advance status bit is set to one, following assertion of the doorbell. Software should acknowledge the Interrupt on Async Advance status as indicated in the USB_USBSTS register, before using the doorbell handshake again.

65.4.3.8.3 Empty Asynchronous Schedule Detection

The Enhanced Host Controller Interface uses two bits to detect when the asynchronous schedule is empty.

The queue head data structure (see Table 65-25) defines an *H-bit* in the queue head, which allows software to mark a queue head as being the *head* of the reclaim list. The Enhanced Host Controller Interface also keeps a 1-bit flag in the USB_USBSTS register (*Reclamation*) that is set to zero when the Enhanced Interface Host Controller observes a queue head with the H-bit set to one. The reclamation flag in the status register is set to one when any USB transaction from the asynchronous schedule is executed (or whenever the asynchronous schedule starts, see Asynchronous schedule traversal: *Start Event*).

If the Enhanced Host Controller Interface ever encounters an *H-bit* of one and a *Reclamation* bit of zero, the EHCI controller simply stops traversal of the asynchronous schedule.

An example illustrating the H-bit in a schedule is shown in the following figure.

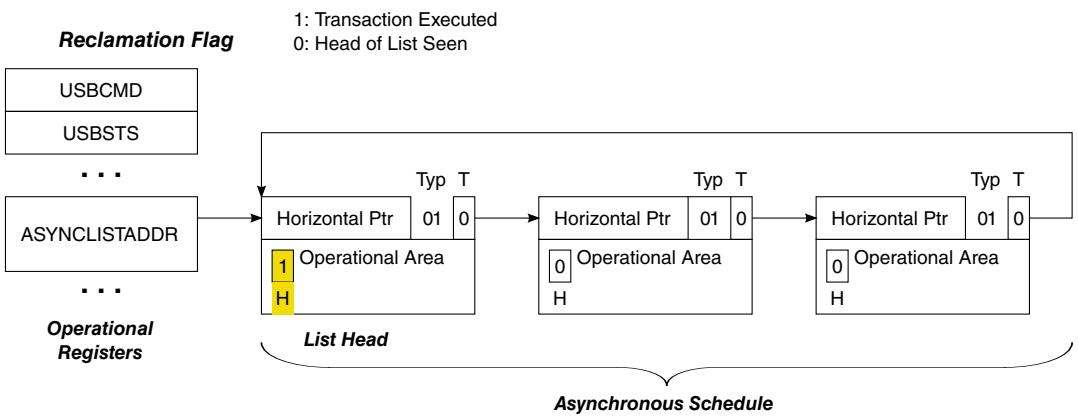


Figure 65-15. Asynchronous Schedule List w/Annotation to Mark Head of List

Software must ensure there is at most one queue head with the *H-bit* set to one, and that it is always coherent with respect to the schedule.

65.4.3.8.4 Restarting Asynchronous Schedule Before EOF

There are many situations where the host controller will detect an empty list *long* before the end of the micro-frame.

It is important to remember that under many circumstances the schedule traversal has stopped due to Nak/Nyet responses from all endpoints.

An example of particular interest is when a start-split for a bulk endpoint occurs early in the micro-frame. Given the EHCI simple traversal rules, the complete-split for that transaction may Nak/Nyet out very quickly. If it is the only item in the schedule, then the host controller ceases traversal of the Asynchronous schedule very early in the micro-frame. In order to provide reasonable service to this endpoint, the host controller should issue the complete-split before the end of the current micro-frame, instead of waiting

until the next micro-frame. When the reason for host controller idling asynchronous schedule traversal is because of empty list detection, it is mandatory the host controller implement a 'waking' method to resume traversal of the asynchronous schedule. An example method is described below.

65.4.3.8.4.1 Example Method for Restarting Asynchronous Schedule Traversal

The reason for idling the host controller when the list is empty is to keep the host controller from unnecessarily occupying too much memory bandwidth. The question is: *how long should the host controller stay idle before restarting*?

The answer in this example is based on deriving a manifest constant, which is the amount of time the host controller will stay idle before restarting traversal. In this example, the manifest constant is called *AsyncSchedSleepTime*, and has a value of 10 µsec. The value is derived based on the analysis in Example Derivation for *AsyncSchedSleepTime*, The traversal algorithm is simple:

- Traverse the Asynchronous schedule until the either an End-Of-micro-Frame event occurs, or an empty list is detected. If the event is an End-of-micro-Frame, go attempt to traverse the Periodic schedule. If the event is an empty list, then set a sleep timer and go to a *schedule sleep* state.
- When the sleep timer expires, set working context to the Asynchronous Schedule start condition and go to *schedule active* state. The start context allows the HC to reload *Nakcnt* fields, and so on. So the HC has a chance to run for more than one iteration through the schedule.

This process simply repeats itself each micro-frame. The figure below illustrates a sample state machine to manage the active and sleep states of the Asynchronous Schedule traversal policy. There are three states: Actively traversing the Asynchronous schedule, Sleeping, and Not Active. The last two are similar in terms of interaction with the Asynchronous schedule, but the Not Active state means that the host controller is busy with the Periodic schedule or the Asynchronous schedule is not enabled. The Sleeping state is specifically a special state where the host controller is just waiting for a period of time before resuming execution of the Asynchronous schedule.

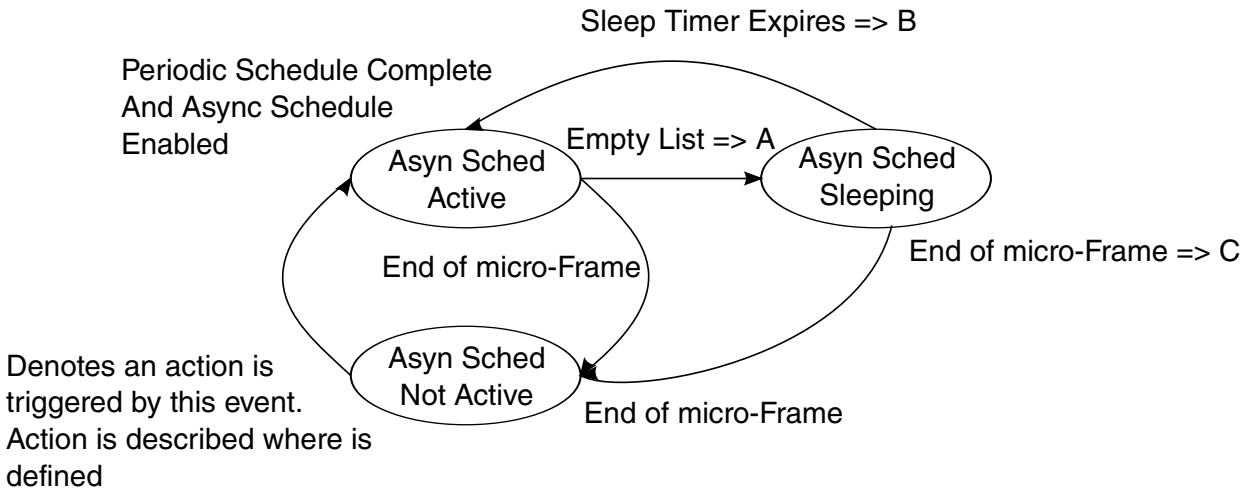


Figure 65-16. Example State Machine for Managing Asynchronous Schedule Traversal

The actions referred to in the figure above are defined in the following table.

Table 65-40. Asynchronous Schedule SM Transition Actions

Action	Action Description Label
A	On detection of the empty list, the host controller sets the *AsynchronousTraversalSleepTimer* to *AsyncSchedSleepTime*.
B	When the *AsynchronousTraversalSleepTimer* expires, the host controller sets the *Reclamation* bit in the USBSTS register to one and moves the Nak Counter reload state machine to WaitForListHead (see Nak Count Reload Control).
C	The host controller cancels the sleep timer (*AsynchronousTraversalSleepTimer*).

65.4.3.8.4.2 Async Sched Not Active

This is the initial state of the traversal state machine after a host controller reset. The traversal state machine does not leave this state when the *Asynchronous Schedule Enable* bit in the USB_USBCMD register is zero.

This state is entered from Async Sched Active or Async Sched Sleeping states when the end-of-micro-frame event is detected.

65.4.3.8.4.3 Async Sched Active

This state is entered from the Async Sched Not Active state when the periodic schedule is not active. It is also entered from the Async Sched Sleeping states when the *AsyncrhonousTraversalSleepTimer* expires. On every transition into this state, the host controller sets the *Reclamation* bit in the USB_USBSTS register to one.

While in this state, the host controller continually traverses the asynchronous schedule until either the end of micro-frame or an empty list condition is detected.

65.4.3.8.4.4 Async Sched Sleeping

The state is entered from the Async Sched Active state when a schedule empty condition is detected. On entry to this state, the host controller sets the *AsynchronousTraversalSleepTimer* to *AsyncSchedSleepTime*.

65.4.3.8.4.5 Example Derivation for *AsyncSchedSleepTime*

The derivation is based on analysis of what work the host controller could be doing next.

It assumes the host controller does not keep any state about what work is possibly pending in the asynchronous schedule. The schedule could contain any mix of the possible combinations of high- full- or low-speed control and bulk requests.

The table below summarizes some of the typical 'next transactions' that could be in the schedule, and the amount of time (for example *footprint*, or *wall clock*) the transaction takes to complete.

Table 65-41. Typical Low-/Full-speed Transaction Times

Transaction Attributes		Footprint (time)	Description
Speed	HS	11.9 ms	Maximum foot print for a worst-case, full-sized bulk data transaction.
Size	512	9.45 ms	Maximum footprint for an approximate best-case, full-sized bulk data transaction.
Type	Bulk		
Speed	FS	~50 ms	Approximate typical for full-sized bulk data. An 8-byte low-speed is about 2x, or between 90 and 100 ms.
Size	64		
Type	Bulk		
Speed	FS	~12 ms	Approximate typical for 8-byte bulk/control (that is setup)
Size	8		
Type	Cntrl		

A *AsyncSchedSleepTime* value of 10 µs provides a reasonable relaxation of the system memory load and still provides a good level of service for the various transfer types and payload sizes. For example, say we detect an empty list after issuing a start-split for a 64-byte full-speed bulk request. Assuming this is the only thing in the list, the host controller gets the results of the full-speed transaction from the hub during the fifth complete-split request. If the full-speed transaction was an IN and it nak'd, the 10 µs sleep period would allow the host controller to get the NAK results on the first complete-split.

65.4.3.8.5 Asynchronous schedule traversal: *Start Event*

Once the HC has *idled* itself through the empty schedule detection (Section 0), it will naturally *activate* and begin processing from the Periodic Schedule at the beginning of each micro-frame.

In addition, it may have idled itself early in a micro-frame. When this occurs (idles early in the micro-frame) the HC must occasionally *re-activate* during the micro-frame and traverse the asynchronous schedule to determine whether any progress can be made. The requirements and method for this restart are described in Restarting Asynchronous Schedule Before EOF . Asynchronous schedule *Start Event*s are defined to be:

- Whenever the host controller transitions from the periodic schedule to the asynchronous schedule. If the periodic schedule is disabled and the asynchronous schedule is enabled, then the beginning of the micro-frame is equivalent to the transition from the periodic schedule, or
- The asynchronous schedule traversal restarts from a sleeping state (see Restarting Asynchronous Schedule Before EOF).

65.4.3.8.6 Reclamation Status Bit (USBSTS Register)

The operation of the empty asynchronous schedule detection feature (see Empty Asynchronous Schedule Detection) depends on the proper management of the *Reclamation* bit in the USB_USBSTS register. The host controller tests for an empty schedule just after it fetches a new queue head while traversing the asynchronous schedule (see Fetch Queue Head).

The host controller is required to set the *Reclamation* bit to one whenever an asynchronous schedule traversal *Start Event*, as documented in Asynchronous schedule traversal: *Start Event*, occurs. The *Reclamation* bit is also set to one whenever the host controller executes a transaction while traversing the asynchronous schedule (see Execute Transaction). The host controller sets the *Reclamation* bit to zero whenever it finds a queue head with its *H-bit* set to one. Software should only set a queue head's *H-bit* if the queue head is in the asynchronous schedule. If software sets the *H-bit* in an interrupt queue head to one, the resulting behavior is undefined. The host controller may set the *Reclamation* bit to zero when executing from the periodic schedule.

65.4.3.9 Operational Model for Nak Counter

This section describes the operational model for the *NakCnt* field defined in a queue head.

See Queue Head Initialization for more information. Software should not use this feature for interrupt queue heads. This rule is not required to be enforced by the host controller.

USB protocol has built-in flow control through the Nak response by a device. There are several scenarios, beyond the Ping feature, where an endpoint may naturally Nak or Nyet the majority of the time. An example is the host controller management of the split transaction protocol for control and bulk endpoints. All bulk endpoints (High- or Full-speed) are serviced through the same asynchronous schedule. The time between the *Start-split* transaction and the first *Complete-split* transaction could be very short (that is like when the endpoint is the only one in the asynchronous schedule). The hub NYETs (effectively Naks) the *Complete-split* transaction until the classic transaction is complete. This could result in the host controller thrashing memory, repeatedly fetching the queue head and executing the transaction to the Hub, which does not complete until after the transaction on the classic bus completes.

The two component fields in a queue head to support the throttling feature: a counter field (*NakCnt*), and a counter reload field (*RL*). *NakCnt* is used by the host controller as one of the criteria to determine whether or not to execute a transaction to the endpoint. The two operational modes associated with this counter:

- Not Used- This mode is set when the *RL* field is zero. The host controller ignores the *NakCnt* field for any execution of transactions through a queue head with an *RL* field of zero. Software must use this selection for interrupt endpoints.
- Nak Throttle Mode- This mode is selected when the *RL* field is non-zero. In this mode, the value in the *NakCnt* field represents the maximum number of Nak or Nyet responses the host controller tolerates on each endpoint. In this mode, the HC decrements the *NakCnt* field based on the token/handshake criteria listed in the table below. The host controller must reload *NakCnt* when the endpoint successfully moves data (for example, policy to reward device for moving data).

The following table describes the NakCnt field adjustment rules.

Table 65-42. NakCnt Field Adjustment Rules

Token	Handshake	
	Handshake NAK	NYET
IN/PING	decrement *NakCnt*	N/A (protocol error)
OUT	decrement *NakCnt*	No Action[1] Start
Split	decrement *NakCnt*	N/A (protocol error)
Complete Split	No Action	Decrement *NakCnt*

1. Recommended behavior on this response is to reload *NakCnt*

In summary, system software enables the counter by setting the reload field (*RL*) to a non-zero value. The host controller may execute a transaction if *NakCnt* is non-zero. The host controller does not execute a transaction if *NakCnt* is zero. The reload mechanism is described in detail in Nak Count Reload Control .

NOTE

When all queue heads in the Asynchronous Schedule either exhausts all transfers or all NakCnt's go to zero, then the host controller detects an empty Asynchronous Schedule and idle schedule traversal (see Empty Asynchronous Schedule Detection).

Any time the host controller begins a new traversal of the Asynchronous Schedule, a *Start Event* is assumed, see Asynchronous schedule traversal: *Start Event*. Every time a Start-Event occurs, the Nak Count reload procedure is enabled.

65.4.3.9.1 Nak Count Reload Control

When the host controller reaches the *Execute Transaction* state for a queue head (meaning that it has an active operational state), it checks to determine whether the *NakCnt* field should be reloaded from *RL* (see Execute Transaction). If the answer is yes, then *RL* is copied into *NakCnt*. After the reload or if the reload is not active, the host controller evaluates whether to execute the transaction.

The host controller must reload nak counters (*NakCnt* see Table 65-25) in queue heads during the first pass through the reclamation list after an asynchronous schedule Start Event (see Asynchronous schedule traversal: *Start Event* for the definition of the Start Event). The Asynchronous Schedule should have at most one queue head marked as the head (see Figure 65-15).

The following figure illustrates an example state machine that satisfies the operational requirements of the host controller detecting the first pass through the Asynchronous Schedule. This state machine is maintained internal to the host controller and is only used to gate reloading of the nak counter during the queue head traversal state: Execute Transaction (see the figure below). The host controller does not perform the nak counter reload operation if the RL field (see Table 65-25) is set to zero.

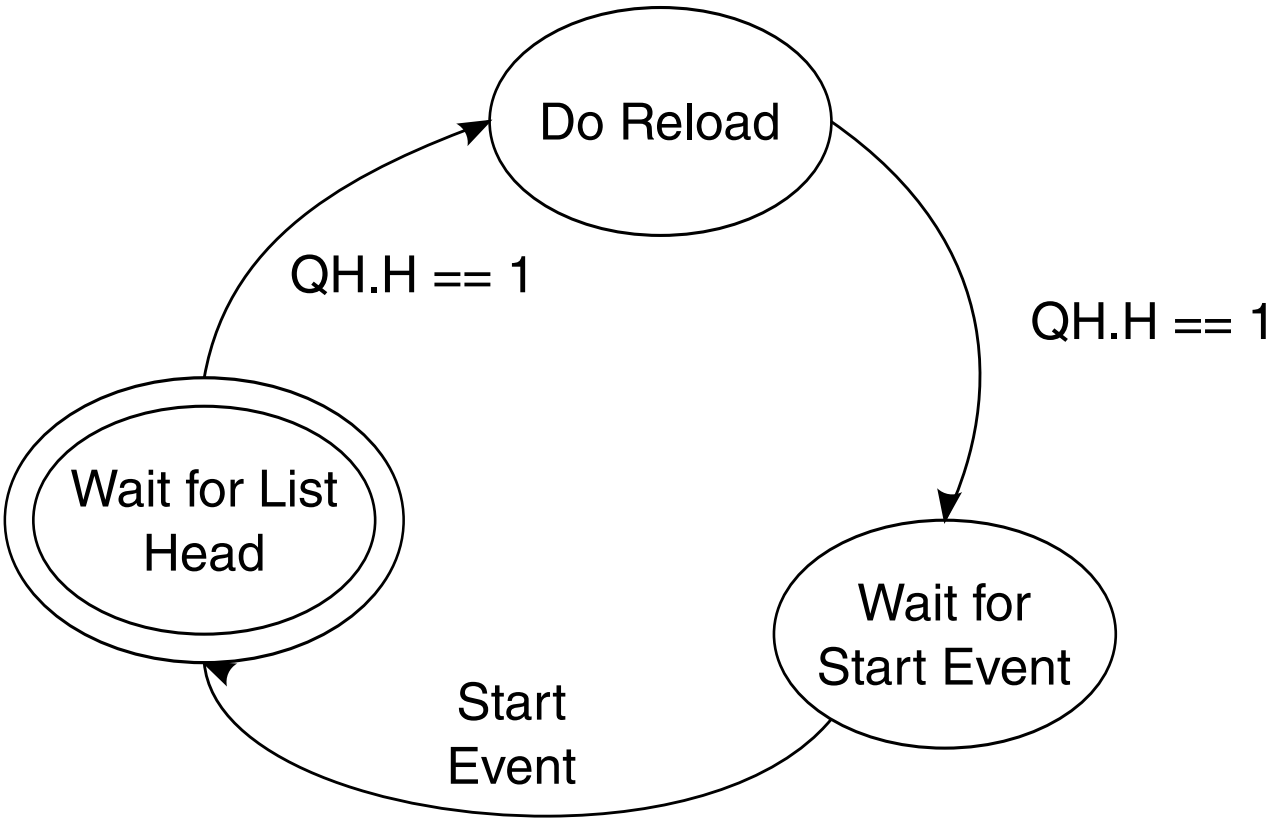


Figure 65-17. Example HC State Machine for Controlling Nak Counter Reloads

65.4.3.9.1.1 Wait for List Head

This is the initial state.

The state machine enters this state from Wait for Start Event when a start event as defined in Asynchronous schedule traversal: *Start Event* occurs.

The purpose of this state is to wait for the first observation of the head of the Asynchronous Schedule.

This occurs when the host controller fetches a queue head whose *H-bit* is set to one.

65.4.3.9.1.2 Do Reload

This state is entered from the Wait for List Head state when the host controller fetches a queue head with the *H-bit* set to one. While in this state, the host controller performs nak counter reloads for every queue head visited that has a non-zero nak reload value (*RL*) field.

65.4.3.9.1.3 Wait for Start Event

This state is entered from the *Do Reload* state when a queue head with the *H-bit* set to one is fetched. While in this state, the host controller does not perform nak counter reloads.

65.4.3.10 Managing Control/Bulk/Interrupt Transfers through Queue Heads

This section presents an overview of how the host controller interacts with queuing data structures.

Queue heads use the Queue Element Transfer Descriptor (qTD) structure. One queue head is used to manage the data stream for one endpoint. The queue head structure contains static endpoint characteristics and capabilities. It also contains a working area from where individual bus transactions for an endpoint are executed (see Overlay area defined in Table 65-25). Each qTD represents one or more bus transactions, which is defined in the context of this specification as a *transfer*.

The general processing model for the host controller's use of a queue head is simple:

- read a queue head,
- execute a transaction from the overlay area,
- write back the results of the transaction to the overlay area,
- move to the next queue head.

If the host controller encounters errors during a transaction, the host controller sets one (or more) of the error reporting bits in the queue head's *Status* field. The *Status* field accumulates all errors encountered during the execution of a qTD (for example, the error bits in the queue head *Status* field are 'sticky' until the transfer (qTD) has completed). This state is always written back to the source qTD when the transfer is complete. On transfer (for example, buffer or halt conditions) boundaries, the host controller must auto-advance (without software intervention) to the next qTD. Additionally, the hardware must be able to halt the queue so no additional bus transactions occurs for the endpoint and the host controller does not advance the queue.

An example host controller operational state machine of a queue head traversal is illustrated in the following figure. This state machine is a model for how a host controller should traverse a queue head. The host controller must be able to advance the queue from the *Fetch QH* state in order to avoid all hardware/software race conditions. This simple mechanism allows software to simply link qTDs to the queue head and *activate* them, then the host controller always *find* them if/when they are reachable. The figure below illustrates the Host Controller Queue Head Traversal State Machine.

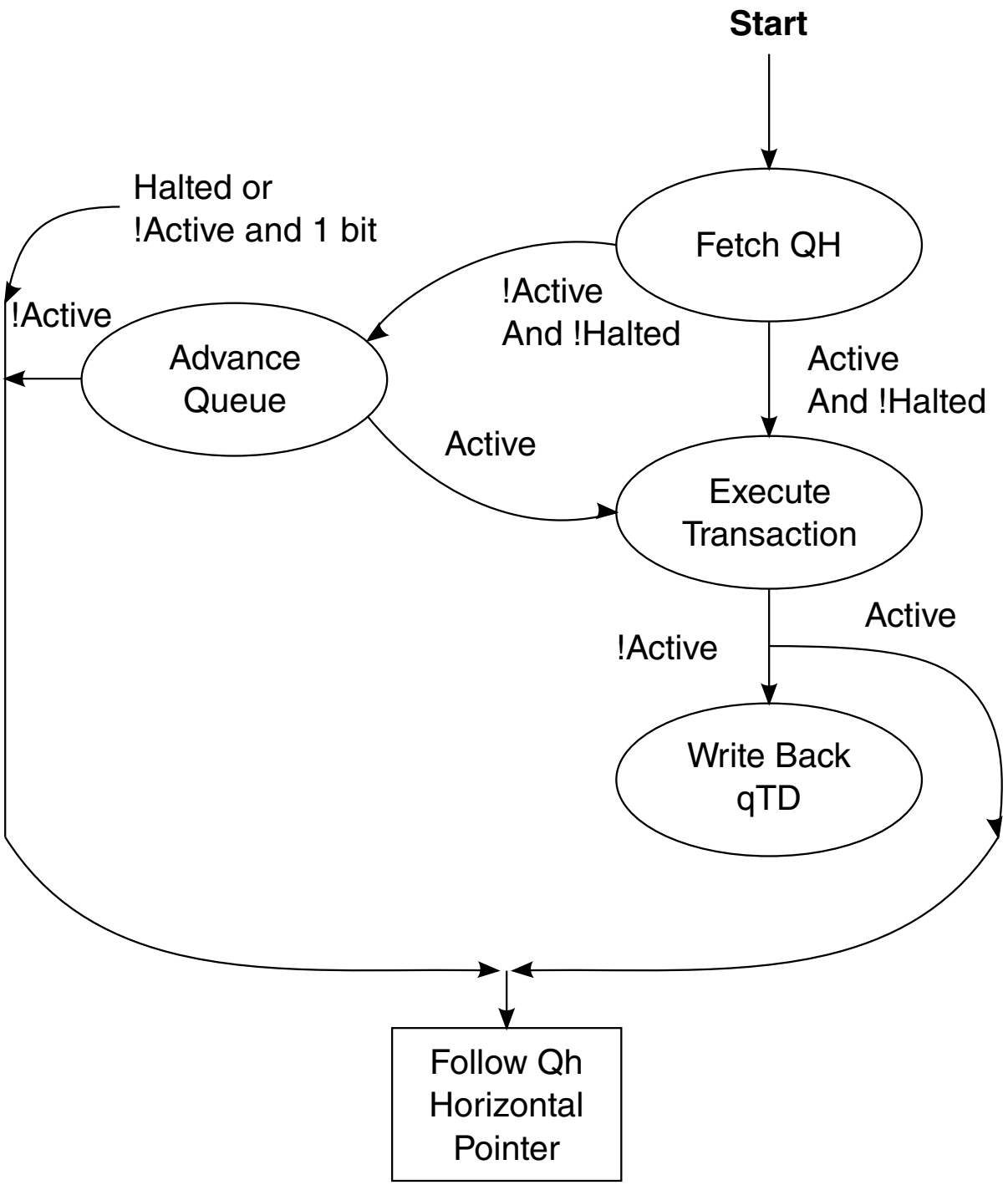


Figure 65-18. Host Controller Queue Head Traversal State Machine

This traversal state machine applies to all queue heads, regardless of transfer type or whether split transactions are required. The following sections describe each state. Each state description describes the entry criteria. The Execute Transaction state (see Execute

Transaction) describes the basic requirements for all endpoints. Split Transactions for Asynchronous Transfers and Split Transaction Interrupt describe details of the required extensions to the Execute Transaction state for endpoints requiring split transactions.

NOTE

Prior to software placing a queue head into either the periodic or asynchronous list, software must ensure the queue head is properly initialized. Minimally, the queue head should be initialized to the following (see Section Queue Head for layout of a queue head):

Valid static endpoint state.

- For the very first use of a queue head, software may zero-out the queue head transfer overlay, then set the *Next qTD Pointer* field value to reference a valid qTD.

65.4.3.10.1 Fetch Queue Head

A queue head can be referenced from the physical address stored in the ASYNCLISTADDR Register (see Next Asynch. Address (USB_*n*ASYNCLISTADDR))/ Endpoint List Address (USB_*n*ENDPTLISTADDR) Additionally, it may be referenced from the *Next LinkPointer* field of an iTD, siTD, FSTN or another Queue Head. If the referencing link pointer has the *Typ* field set to indicate a queue head, it is assumed to reference a queue head structure as defined in Table 65-25.

While in this state, the host controller performs operations to implement empty schedule detection (see Empty Asynchronous Schedule Detection) and Nak Counter reloads (see Operational Model for Nak Counter). After the queue head has been fetched, the host controller conducts the following queries for empty schedule detection:

- If queue head is not an interrupt queue head (that is *S-mask* is zero), and
- The *H-bit* is one, and
- The *Reclamation* bit in the USBSTS register is zero.

When these criteria are met, the host controller stops traversing the asynchronous list (as described in Empty Asynchronous Schedule Detection). When the criteria are not met, the host controller continues schedule traversal. If the queue head is not an interrupt and the *H-bit* is one and the *Reclamation* bit is one, then the host controller sets the *Reclamation* bit in the USBSTS register to zero before completing this state. The operations for reloading of the Nak Counter are described in detail in Operational Model for Nak Counter.

This state is complete when the queue head has been read on-chip.

65.4.3.10.2 Advance Queue

To advance the queue, the host controller must find the next qTD, adjust pointers, perform the overlay and write back the results to the queue head.

This state is entered from the FetchQHD state if the overlay *Active* and *Halt* bits are set to zero. On entry to this state, the host controller determines which next pointer to use to fetch a qTD, fetches a qTD and determines whether or not to perform an overlay.

NOTE

If the *I-bit* is one and the *Active* bit is zero, the host controller immediately skips processing of this queue head, exits this state and uses the horizontal pointer to the next schedule data structure. If the field *Bytes to Transfer* is not zero and the *T-bit* in the *Alternate Next qTD Pointer* is set to zero, then the host controller uses the *Alternate Next qTD Pointer*. Otherwise, the host controller uses the *NextqTD Pointer*. If *NextqTD Pointer's T-bit* is set to one, then the host controller exits this state and uses the horizontal pointer to the next schedule data structure.

Using the selected pointer the host controller fetches the referenced qTD. If the fetched qTD has its *Active* bit set to one, the host controller moves the pointer value used to reach the qTD (*Next* or *Alternate Next*) to the *Current qTD Pointer* field, then performs the overlay. If the fetched qTD has its *Active* bit set to zero, the host controller aborts the queue advance and follows the queue head's horizontal pointer to the next schedule data structure.

The host controller performs the overlay based on the following rules:

- The value of the data toggle (*dt*) field in the overlay area depends on the value of the *data toggle control* (*dtc*) bit (see Table 65-27).
- If the *EPS* field indicates the endpoint is a high-speed endpoint, the *Ping* state field is preserved by the host controller. The value of this field is not changed as a result of the overlay.
- *C-prog-mask* field is set to zero (field from incoming qTD is ignored, as is the current contents of the overlay area).
- *Frame Tag* field is set to zero (field from incoming qTD is ignored, as is the current contents of the overlay area).
- *NakCnt* field in the overlay area is loaded from the *RL* field in the queue head's Static Endpoint State.
- All other areas of the overlay are set by the incoming qTD.

The host controller exits this state when it has committed the write to the queue head.

65.4.3.10.3 Execute Transaction

The host controller enters this state from the Fetch Queue Head state only if the *Active* bit in *Status* field of the queue head is set to one.

On entry to this state, the host controller executes a few pre-operations, then checks some pre-condition criteria before committing to executing a transaction for the queue head.

The pre-operations performed and the pre-condition criteria depend on whether the queue head is an interrupt endpoint. The host controller can determine that a queue head is an interrupt queue head when the queue head's *S-mask* field contains a non-zero value. It is the responsibility of software to ensure the *S-mask* field is appropriately initialized based on the transfer type. There are other criteria that must be met if the *EPS* field indicates that the endpoint is a low- or full-speed endpoint, see Split Transactions for Asynchronous Transfers and Split Transaction Interrupt .

65.4.3.10.3.1 Interrupt Transfer Pre-condition Criteria

If the queue head is for an interrupt endpoint (for example, non-zero *S-mask* field), then the FRINDEX[2:0] field must identify a bit in the *S-mask* field that has one in it.

For example, an *S-mask* value of 00100000b would evaluate to true only when FRINDEX[2:0] is equal to 101b. If this condition is met then the host controller considers this queue head for a transaction.

65.4.3.10.3.2 Asynchronous Transfer Pre-operations and Pre-condition Criteria

If the queue head is not for an interrupt endpoint (for example, zero *S-mask* field), then the host controller performs one pre-operation and then evaluates one pre-condition criteria.

The pre-operation is:

Checks the Nak counter reload state (Operational Model for Nak Counter). It may be necessary for the host controller to reload the Nak Counter field. The reload is performed at this time.

The pre-condition evaluated is:

- Whether or not the *NakCnt* field has been reloaded, the host controller checks the value of the *NakCnt* field in the queue head. If *NakCnt* is non-zero, or if the *Reload Nak Counter* field is zero, then the host controller considers this queue head for a transaction.

65.4.3.10.3.3 Transfer Type Independent Pre-operations

Regardless of the transfer type, the host controller always performs at least one pre-operation and evaluates one pre-condition. The pre-operation is:

- A host controller internal transaction (down) counter *qHTransactionCounter* is loaded from the queue head's *Mult* field. A host controller implementation is allowed to ignore this for queue heads on the asynchronous list. It is mandatory for interrupt queue heads. Software should ensure that the *Mult* field is set appropriately for the transfer type.

The pre-conditions evaluated are:

- The host controller determines whether there is enough time in the micro-frame to complete this transaction (see Transaction Fit - A Best-Fit Approximation Algorithm for an example evaluation method). If there is not enough time to complete the transaction, the host controller exits this state.
- If the value of *qHTransactionCounter* for an interrupt endpoint is zero, then the host controller exits this state.

When the pre-operations are complete and pre-conditions are met, the host controller sets the *Reclamation* bit in the USBSTS register to one and then begins executing one or more transactions using the endpoint information in the queue head. The host controller iterates *qHTransactionCounter* times in this state executing transactions. After each transaction is executed, *qHTransactionCounter* is decremented by one. The host controller exits this state when one of the following events occurs:

- The *qHTransactionCounter* decrements to zero, or
- The endpoint responds to the transaction with any handshake other than an ACK,[4] or
- The transaction experiences a transaction error, or
- The *Active* bit in the queue head goes to zero, or
- There is not enough time in the micro-frame left to execute the next transaction(see Transaction Fit - A Best-Fit Approximation Algorithm) for example method for implementing the frame boundary test).

NOTE

For a high-bandwidth interrupt OUT endpoint, the host controller may optionally immediately retry the transaction if it fails.

The results of each transaction is recorded in the on-chip overlay area. If data was successfully moved during the transaction, the transfer state in the overlay area is advanced. To advance queue head's transfer state, the *Total Bytes to Transfer* field is decremented by the number of bytes moved in the transaction, the data toggle bit (*dt*) is toggled, the current page offset is advanced to the next appropriate value (for example,

advanced by the number of bytes successfully moved), and the *C_Page* field is updated to the appropriate value (if necessary). See Buffer Pointer List Use for Data Streaming with qTDs .

NOTE

The *Total Bytes To Transfer* field may be zero when all the other criteria for executing a transaction are met. When this occurs, the host controller executes zero-length transaction to the endpoint. If the *PID_Code* field indicates an IN transaction and the device delivers data, the host controller detects a packet babble condition, set the *babble* and *halted* bits in the *Status* field, set the *Active* bit to zero, write back the results to the source qTD, then exit this state.

In the event an IN token receives a data PID mismatch response, the host controller must ignore the received data (for example not advance the transfer state for the bytes received). Additionally, if the endpoint is an interrupt IN, then the host controller must record that the transaction occurred (for example, decrement *qHTransactionCounter*). It is recommended (but not required) the host controller continue executing transactions for this endpoint if the resultant value of *qHTransactionCounter* is greater than one.

If the response to the IN bus transaction is a Nak (or Nyet) and *RL* is non-zero, *NakCnt* is decremented by one. If *RL* is zero, then no write-back by the host controller is required (for a transaction receiving a Nak or Nyet response and the value of *CErr* did not change). Software should set the *RL* field to zero if the queue head is an interrupt endpoint. Host controller hardware is not required to enforce this rule or operation.

After the transaction has finished and the host controller has completed the post processing of the results (advancing the transfer state and possibly *NakCnt*, the host controller writes back the results of the transaction to the queue head's overlay area in main memory).

The number of bytes moved during an IN transaction depends on how much data the device endpoint delivers. The maximum number of bytes a device can send is *MaximumPacket Size*. The number of bytes moved during an OUT transaction is either *Maximum Packet Length* bytes or *Total Bytes to Transfer*, whichever is less.

If there was a transaction error during the transaction, the transfer state (as defined above) is not advanced by the host controller. The *CErr* field is decremented by one and the status field is updated to reflect the type of error observed. Transaction errors are summarized in Transaction Error .

The following events causes the host controller to clear the *Active* bit in the queue head's overlay status field. When the *Active* bit transitions from one to zero, the transfer in the overlay is considered complete. The reason for the transfer completion (clearing the *Active* bit) determines the next state.

- *CErr* field decrements to zero. When this occurs the *Halted* bit is set to one and *Active* is set to zero. This results in the hardware not advancing the queue and the pipe halts. Software must intercede to recover.
- The device responds to the transaction with a STALL PID. When this occurs, the *Halted* bit is set to one and the *Active* bit is set to zero. This results in the hardware not advancing the queue and the pipe halts. Software must intercede to recover.
- The *Total Bytes to Transfer* field is zero after the transaction completes.
  - For a zero length transaction, it was zero before the transaction was started. When this condition occurs, the *Active* bit is set to zero.
- The PID code is an IN, and the number of bytes moved during the transaction is less than the *Maximum Packet Length*. When this occurs, the *Active* bit is set to zero and a short packet condition exists. The short-packet condition is detected during the Advance Queue state. Refer to Split Transactions for additional rules for managing low- and full-speed transactions.

With the exception of a NAK response (when *RL* field is zero), the host controller always writes the results of the transaction back to the overlay area in main memory. This includes when the transfer completes. For a high-speed endpoint, the queue head information written back includes minimally the following fields: The *PID Code* field indicates an IN and the device sends more than the expected number of bytes (for example *Maximum Packet Length* or *Total Bytes to Transfer* bytes, whichever is less) (for example a packet babble). This results in the host controller setting the *Halted* bit to one.

- NakCnt, dt, Total Bytes to Transfer, C_Page, Status, CERR, and Current Offset

For a low- or full-speed device the queue head information written back also includes the fields:

- C-prog-mask, FrameTag and S-bytes.

The duration of this state depends on the time it takes to complete the transaction(s) and the status write to the overlay is committed.

65.4.3.10.3.4 Halting a Queue Head

A halted endpoint is defined only for the transfer types that are managed through queue heads (control, bulk and interrupt).

The following events indicate that the endpoint has reached a condition where no more activity can occur without intervention from the driver:

- An endpoint may return a STALL handshake during a transaction,
- A transaction had three consecutive error conditions, or
- A Packet Babble error occurs on the endpoint.

When any of these events occur (for a queue head) the Host Controller halts the queue head and set the USBERRINT status bit in the USB_n_USBSTS register to one. To halt the queue head, the *Active* bit is set to zero and the *Halted* bit is set to one. There may be other error status bits that are set when a queue is halted. The host controller always writes back the overlay area to the source qTD when the transfer is complete, regardless of the reason (normal completion, short packet or halt). The host controller does not advance the transfer state on a transaction that results in a *Halt* condition (for example no updates necessary for *Total Bytes to Transfer*, *C_Page*, *Current Offset,* and *dt*). The host controller must update *CErr* as appropriate. When a queue head is halted, the *USB Error Interrupt* bit in the USB_n_USBSTS register is set to one. If the *USB Error Interrupt Enable* bit in the USB_n_USBINTR register is set to one, a hardware interrupt is generated at the next interrupt threshold.

65.4.3.10.3.5 Asynchronous Schedule Park Mode

Asynchronous Schedule Park mode is a special execution mode that can be enabled by system software, where the host controller is permitted to execute more than one bus transaction from a high-speed queue head in the Asynchronous schedule before continuing horizontal traversal of the Asynchronous schedule.

This feature has no effect on queue heads or other data structures in the Periodic schedule. This feature is similar in intent as the *Mult* feature that is used in the Periodic schedule. Where-as the *Mult* feature is a characteristic that is tunable for each endpoint; park-mode is a policy that is applied to all high-speed queue heads in the asynchronous schedule. It is essentially the specification of an iterator for consecutive bus transactions to the same endpoint. All of the rules for managing bus transactions and the results of those as defined in Execute Transaction apply. This feature merely specifies how many consecutive times the host controller is permitted to execute from the same queue head before moving to the next queue head in the Asynchronous List. This feature should allow the host controller to attain better bus utilization for those devices that are capable of moving data at maximum rate, while at the same time providing a fair service to all endpoints.

A host controller exports its capability to support this feature to system software by setting the *Asynchronous Schedule Park Capability* bit in the USB_n_HCCPARAMs register to one. This information keys system software that the *Asynchronous Schedule Park Mode Enable* and *Asynchronous Schedule Park Mode Count* fields in the USB_n_USBCMD register are modifiable. System software enables the feature by writing a one to the *Asynchronous Schedule Park Mode Enable* bit.

When park-mode is not enabled (for example *Asynchronous Schedule Park Mode Enable* bit in the USB_n_USBCMD register is zero), the host controller must not execute more than one bus transaction per high-speed queue head, per traversal of the asynchronous schedule. When park-mode is enabled, the host controller must not apply the feature to a queue head whose *EPS* field indicates a Low/Full-speed device (for example only one bus transaction is allowed from each Low/Full-speed queue head per traversal of the asynchronous schedule). Park-mode may only be applied to queue heads in the Asynchronous schedule whose *EPS* field indicates that it is a high-speed device.

The host controller must apply park mode to queue heads whose *EPS* field indicates a high-speed endpoint. The maximum number of consecutive bus transactions a host controller may execute on a high-speed queue head is determined by the value in the *Asynchronous Schedule Park Mode Count* field in the USB_n_USBCMD register. Software must not set *Asynchronous Schedule Park Mode Enable* bit to one and also set *Asynchronous Schedule Park Mode Count* field to zero. The resulting behavior is not defined. An example behavioral example describes the operational requirements for the host controller implementing park-mode. This feature does not affect how the host controller handles the bus transaction as defined in Execute Transaction . It only effects how many consecutive bus transactions for the current queue head can be executed. All boundary conditions, error detection and reporting applies as usual. This feature is similar in concept to the use of the *Mult* field for high-bandwidth Interrupt for queue heads in the Periodic Schedule.

The host controller effectively loads an internal down-counter *PM-Count* from *Asynchronous Schedule Park Mode Count* when *Asyncrhonous Schedule Park Mode Enable* bit is one, and a high-speed queue head is first fetched and meets all the criteria for executing a bus transaction. After the bus transaction, *PM-Count* is decremented. The host controller may continue to execute bus transactions from the current queue head until *PM-Count* goes to zero, an error is detected, the buffer for the current transfer is exhausted or the endpoint responds with a flow-control or STALL handshake.

The following table summarizes the responses that effect whether the host controller continues with another bus transaction for the current queue head.

Table 65-43. Actions for Park Mode, based on Endpoint Response and Residual Transfer State

PID	Endpoint Response	Transfer State after Transaction		Action
		PM-Count	Bytes to Transfer	
IN	DATA[0,1] w/Maximum Packet sized data	Not zero	Not Zero	Allowed to perform another bus transaction.[1, 2]
		Not zero	Zero	Retire qTD and move to next QH
		Zero	Don't care	Move to next QH.

Table continues on the next page...

Table 65-43. Actions for Park Mode, based on Endpoint Response and Residual Transfer State (continued)

	DATA[0,1] w/short packet	Don't care	Don't care	Retire qTD and move to next QH.
	NAK	Don't care	Don't care	Move to next QH.
	STALL, XactErr	Don't care	Don't care	Move to next QH.
OUT	ACK	Not zero	Not Zero	Allowed to perform another bus transaction.[2]
		Not zero	Zero	Retire qTD and move to next QH
		Zero	Don't' care	Move to next QH.
	NYET, NAK	Don't care	Don't care	Move to next QH.
	STALL, XactErr	Don't care	Don't care	Move to next QH
PING	ACK	Not Zero	Not Zero	Allowed to perform another bus transaction. [2]
	NAK	Don't care	Don't care	Move to next QH
	STALL, XactErr	Don't care	Don't care	Move to next QH

1. The host controller may continue to execute bus transactions from the current high-speed queue head (if *PM-Count* is not equal to zero), if a PID mismatch is detected (for example expected DATA1 and received DATA0, or visa-versa).
2. This specification does not *require* that the host controller execute another bus transaction when *PM-Count* is non-zero. Implementations are encouraged to make appropriate complexity and performance trade-offs.

65.4.3.10.4 Write Back qTD

This state is entered from the Execute Transaction state when the *Active* bit is set to zero.

The source data for the write-back is the transfer results area of the queue head overlay area (see Table 65-43).

The host controller uses the *Current qTD Pointer* field as the target address for the qTD.

The queue head transfer result area is written back to the transfer result area of the target qTD. This state is also referred to as: qTD retirement. The fields that must be written back to the source qTD include *Total Bytes to Transfer*, *Cerr*, and *Status*.

The duration of this state depends on when the qTD write-back is committed.

65.4.3.10.5 Follow Queue Head Horizontal Pointer

The host controller must use the horizontal pointer in the queue head to the next schedule data structure when any of the following conditions exist:

- If the *Active* bit is one on exit from the Execute Transaction state, or
- When the host controller exits the Write Back qTD state, or
- If the Advance Queue state fails to advance the queue because the target qTD is not active, or
- If the *Halted* bit is one on exit from the Fetch QH state.

There is no functional requirement that the host controller wait until the current transaction is complete before using the horizontal pointer to read the next linked data structure. However, it must wait until the current transaction is complete before executing the next data structure.

65.4.3.10.6 Buffer Pointer List Use for Data Streaming with qTDs

A qTD has an array of buffer pointers, which is used to reference the data buffer for a transfer. This specification requires that the buffer associated with the transfer be *virtually contiguous*.

This means: if the buffer spans more than one physical page, it must obey the following rules (the figure below illustrates an example):

- The first portion of the buffer must begin at some offset in a page and extend through the end of the page.
- The remaining buffer cannot be allocated in small chunks scattered around memory. For each 4 K chunk beyond the first page, each buffer portion matches to a full 4 K page. The final portion, which may only be large enough to occupy a portion of a page, must start at the top of the page and be contiguous within that page.

The buffer pointer list in the qTD is long enough to support a maximum transfer size of 20 K bytes. This case occurs when all five buffer pointers are used and the first offset is zero. A qTD handles a 16 Kbyte buffer with any starting buffer alignment.

The host controller uses the field *C_Page* field as an index value to determine which buffer pointer in the list should be used to start the current transaction. The host controller uses a different buffer pointer for each physical page of the buffer. This is always true, even if the buffer is physically contiguous.

The host controller must detect when the current transaction spans a page boundary and automatically move to the next available buffer pointer in the page pointer list. The next available pointer is reached by incrementing *C_Page* and pulling the next page pointer from the list. Software must ensure there are sufficient buffer pointers to move the amount of data specified in the *Bytes to Transfer* field.

The following figure illustrates a nominal example of how System software would initialize the buffer pointers list and the *C_Page* field for a transfer size of 16383 bytes. *C_Page* is set to zero. The upper 20-bits of Page 0 references the start of the physical page. *Current Offset* (the lower 12-bits of queue head Dword 7) holds the offset in the page for example 2049 (for example 4096-2047). The remaining page pointers are set to reference the beginning of each subsequent 4 K page.

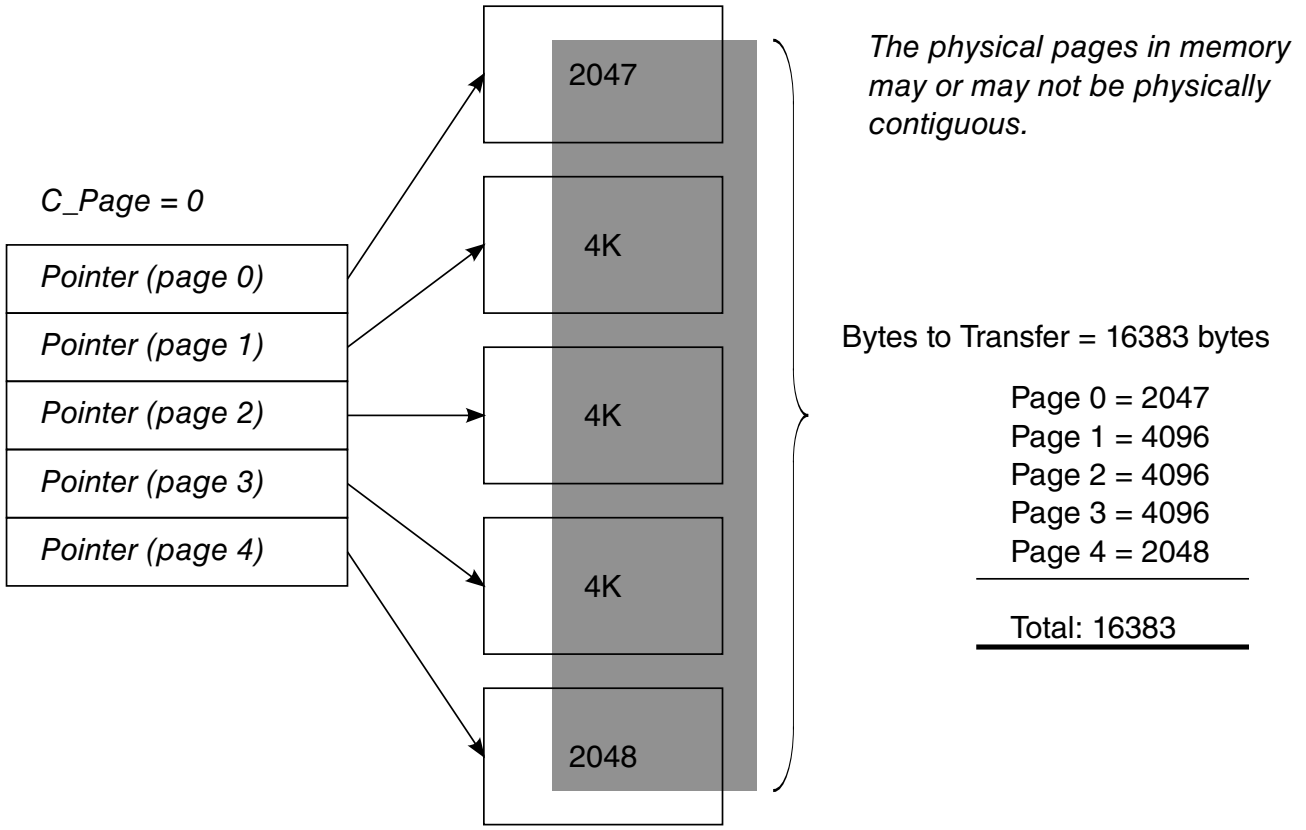


Figure 65-19. Example Mapping of qTD Buffer Pointers to Buffer Pages

For the first transaction on the qTD (assuming a 512-byte transaction), the host controller uses the first buffer pointer (page 0 because *C_Page* is set to zero) and concatenates the *Current Offset* field. The 512 bytes are moved during the transaction, the *Current Offset* and *Total Bytes to Transfer* are adjusted by 512 and written back to the queue head working area.

During the 4th transaction, the host controller needs 511 bytes in page 0 and one byte in page 1. The host controller increments *C_Page* (to 1) and use the page 1 pointer to move the final byte of the transaction. After the 4th transaction, the active page pointer is the page 1 pointer and *Current Offset* has rolled to one, and both are written back to the overlay area. The transactions continue for the rest of the buffer, with the host controller automatically moving to the next page pointer (that is *C_Page*) when necessary. The three conditions for how the host controller handles *C_Page*:

- The current transaction does not span a page boundary. The value of *C_Page* is not adjusted by the host controller.

- The current transaction does span a page boundary. The host controller must detect the page cross condition and advance to the next buffer while streaming data to/from the USB.
- The current transaction completes on a page boundary (that is the last byte moved for the current transaction is the last byte in the page for the current page pointer). The host controller must increment *C_Page* before writing back status for the transaction.

NOTE

The only valid adjustment the host controller may make to *C_Page* is to increment by one.

65.4.3.10.7 Adding Interrupt Queue Heads to the Periodic Schedule

The link path(s) from the periodic frame list to a queue head establishes in which frames a transaction can be executed for the queue head. Queue heads are linked into the periodic schedule so they are polled at the appropriate rate.

System software sets a bit in a queue head's *S-Mask* to indicate which micro-frame within 1 msec period a transaction should be executed for the queue head. Software must ensure that all queue heads in the periodic schedule have *S-Mask* set to a non-zero value. An *S-mask* with zero value in the context of the periodic schedule yields undefined results.

If the desired poll rate is greater than one frame, system software can use a combination of queue head linking and *S-Mask* values to spread interrupts of equal poll rates through the schedule so that the periodic bandwidth is allocated and managed in the most efficient manner possible. Some examples are illustrated in the following table.

Table 65-44. Example Periodic Reference Patterns for Interrupt Transfers with 2ms Poll Rate

Frame # Reference Sequence	Description
0, 2, 4, 6, 8, and so on *S-Mask* = 01h	A queue head for the *bInterval* of 2 msec (16 micro-frames) is linked into the periodic schedule so that it is reachable from the periodic frame list locations indicated in the previous column. In addition, the *S-Mask* field in the queue head is set to 01h, indicating that the transaction for the endpoint should be executed on the bus during micro-frame 0 of the frame.
0, 2, 4, 6, 8, and so on *S-Mask* = 02h	Another example of a queue head with a *bInterval* of 2 msec is linked into the periodic frame list at exactly the same interval as the previous example. However, the *S-Mask* is set to 02h indicating that the transaction for the endpoint should be executed on the bus during micro-frame 1 of the frame.

65.4.3.10.8 Managing Transfer Complete Interrupts from Queue Heads

The host controller sets an interrupt to be signaled at the next interrupt threshold when the completed transfer (qTD) has an *Interrupt on Complete* (*IOC*) bit set to one, or whenever a transfer (qTD) completes with a short packet.

If system software needs multiple qTDs to complete a client request (that is like a control transfer) the intermediate qTDs do not require interrupts. System software may only need a single interrupt to notify it that the complete buffer has been transferred. System software may set IOC's to occur more frequently. A motivation for this may be that it wants early notification so that interface data structures can be re-used in a timely manner.

65.4.3.11 Ping Control

USB 2.0 defines an addition to the protocol for high-speed devices called Ping. Ping is required for all USB 2.0 High-speed bulk and control endpoints.

Ping is not allowed for a split-transaction stream. This extension to the protocol eliminates the bad side-effects of Naking OUT endpoints. The *Status* field has a *Ping State* bit, which the host controller uses to determine the *next* actual PID it uses in the next transaction to the endpoint (see the table below).

The Ping State bit is only managed by the host controller for queue heads that meet the following criteria:

- Queue head is not an interrupt and
- *EPS* field equals High-Speed and
- *PIDCode* field equals OUT

The following table illustrates the state transition table for the host controller's responsibility for maintaining the PING protocol. Refer to Chapter 8 in the USB Specification Revision 2.0 for detailed description on the Ping protocol.

Table 65-45. Ping Control State Transition Table

Event			
Current	Host	Device	Next
Do Ping	PING	Nak	Do Ping
Do Ping	PING	Ack	Do OUT
Do Ping	PING	XactErr [1]	Do Ping
Do Ping	PING	Stall	N/C [2] Do
OUT	OUT	Nak	Do Ping
Do OUT	OUT	Nyet	Do Ping

Table continues on the next page...

Table 65-45. Ping Control State Transition Table (continued)

Do OUT	OUT	Ack	Do OUT
Do OUT	OUT	XactErr[1]	Do Ping
Do OUT	OUT	Stall	N/C [2]

1. Transaction Error (XactErr) is any time the host misses the handshake.
2. No transition change required for the Ping State bit. The Stall handshake results in the endpoint being halted (for example Active set to zero and Halt set to one). Software intervention is required to restart queue. 3 A Nyet response to an OUT means that the device has accepted the data, but cannot receive any more at this time. Host must advance the transfer state and additionally, transition the Ping State bit to Do Ping. The Ping State bit has the following encoding:

Table 65-46. Ping State bit Encoding

Value	Meaning
0B	Do OUT The host controller uses an OUT PID during the next bus transaction to this endpoint.
1B	Do Ping The host controller uses a PING PID during the next bus transaction to this endpoint.

The defined ping protocol (see USB 2.0 Specification, Chapter 8) allows the host to be *imprecise* on the initialization of the ping protocol (that is start in *Do OUT* when we don't know whether there is space on the device or not). The host controller manages the *Ping State* bit. System software sets the initial value in the queue head when it initializes a queue head. The host controller preserves the *Ping State* bit across all queue advancements. This means that when a new qTD is written into the queue head overlay area, the previous value of the *Ping State* bit is preserved.

65.4.3.12 Split Transactions

USB 2.0 defines extensions to the bus protocol for managing USB 1.x data streams through USB 2.0 Hubs.

This section describes how the host controller uses the interface data structures to manage data streams with full- and low-speed devices, connected below USB 2.0 hub, utilizing the split transaction protocol.

Refer to USB 2.0 Specification for the complete definition of the split transaction protocol. Full- and Low-speed devices are enumerated identically as high-speed devices, but the transactions to the Full- and Low-speed endpoints use the split-transaction protocol on the high-speed bus. The split transaction protocol is an encapsulation of (or wrapper around) the Full- or Low-speed transaction. The high-speed wrapper portion of the protocol is addressed to the USB 2.0 Hub and Transaction Translator below which the Full- or Low-speed device is attached.

The EHCI interface uses dedicated data structures for managing full-speed isochronous data streams (see Split Transaction Isochronous Transfer Descriptor (siTD)). Control, Bulk and Interrupt are managed using the queuing data structures (see Queue Head). The interface data structures need to be programmed with the device address and the Transaction Translator number of the USB 2.0 Hub operating as the Low-/Full-speed host controller for this link. The following sections describe the details of how the host controller must process and manage the split transaction protocol.

65.4.3.12.1 Split Transactions for Asynchronous Transfers

A queue head in the asynchronous schedule with an *EPS* field indicating a full-or low-speed device indicates to the host controller that it must use split transactions to stream data for this queue head.

All full-speed bulk and full-, low-speed control are managed through queue heads in the asynchronous schedule.

Software must initialize the queue head with the appropriate device address and port number for the transaction translator that is serving as the full/low-speed host controller for the links connecting the endpoint. Software must also initialize the split transaction state bit (*SplitXState*) to Do-Start-Split. Finally, if the endpoint is a control endpoint, then system software must set the *Control Transfer Type* (*C*) bit in the queue head to one. If this is not a control transfer type endpoint, the *C* bit must be initialized by software to be zero. This information is used by the host controller to properly set the Endpoint Type (ET) field in the split transaction bus token. When the *C* bit is zero, the split transaction token's ET field is set to indicate a bulk endpoint. When the *C* bit is one, the split transaction token's ET field is set to indicate a control endpoint. Refer to Chapter 8 of USB Specification Revision 2.0 for details.

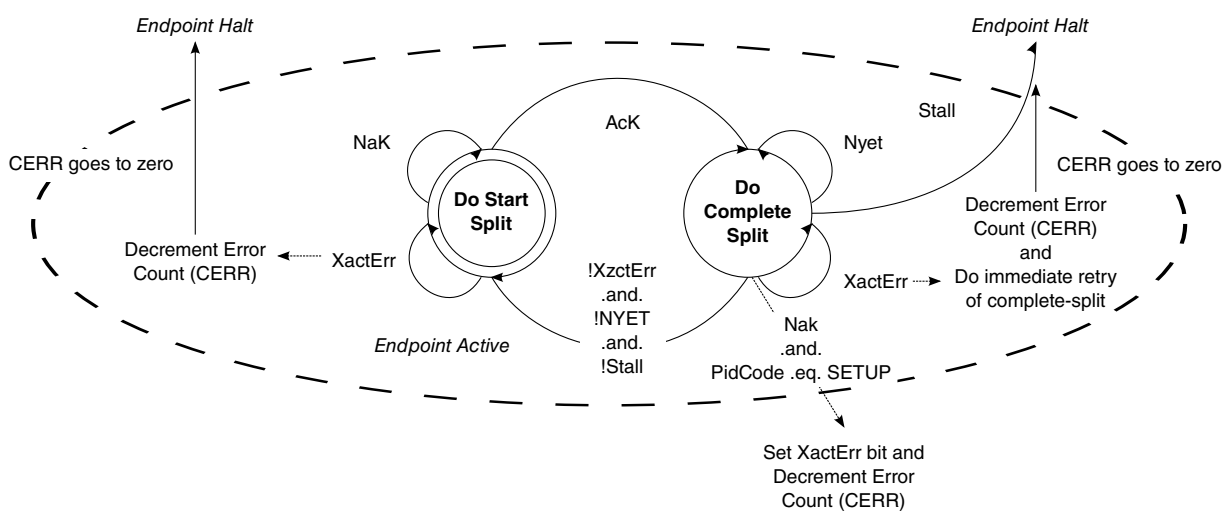


Figure 65-20. Host Controller Asynchronous Schedule Split-Transaction State Machine

65.4.3.12.1.1 Asynchronous - Do Start Split

This is the state which software must initialize a full- or low-speed asynchronous queue head. This state is entered from the Do Complete Split state only after a complete-split transaction receives a valid response from the transaction translator that is not a Nyet handshake.

For queue heads in this state, the host controller executes a start-split transaction to the appropriate transaction translator. If the bus transaction completes without an error and *PidCode* indicates an IN or OUT transaction, then the host controller reloads the error counter (*CErr*). If it is a successful bus transaction and the *PidCode* indicates a SETUP, the host controller does not reload the error counter. If the transaction translator responds with a Nak, the queue head is left in this state, and the host controller proceeds to the next queue head in the asynchronous schedule.

If the host controller times out the transaction (no response, or bad response) the host controller decrements *Cerr* and proceeds to the next queue head in the asynchronous schedule.

65.4.3.12.1.2 Asynchronous - Do Complete Split

This state is entered from the Do Start Split state only after a start-split transaction receives an Ack handshake from the transaction translator.

For queue heads in this state, the host controller executes a complete-split transaction to the appropriate transaction translator. If the transaction translator responds with a Nyet handshake, the queue head is left in this state, the error counter is reset and the host controller proceeds to the next queue head in the asynchronous schedule. When a Nyet handshake is received for a bus transaction where the queue head's *PidCode* indicates an IN or OUT, the host controller reloads the error counter (*CErr*). When a Nyet handshake is received for a complete-split bus transaction where the queue head's *PidCode* indicates a SETUP, the host controller must not adjust the value of *CErr*.

Independent of *PIDCode*, the following responses have the effects:

- Transaction Error (XactErr). Timeout or data CRC failure, and so on. The error counter (*Cerr*) is decremented by one and the complete split transaction is *immediately* retried (if possible). If there is not enough time in the micro-frame to execute the retry, the host controller MUST ensure that the next time the host controller begins executing from the Asynchronous schedule, it must begin executing from this queue head. If another start-split (for some other endpoint) is sent to the transaction translator before the complete-split is really completed, the transaction translator could dump the results (which were never delivered to the host). This is why the core specification states the retries must be immediate. A method to

accomplish this behavior is to not advance the asynchronous schedule. When the host controller returns to the asynchronous schedule in the next micro-frame, the first transaction from the schedule is the retry for this endpoint.

If *Cerr* went to zero, the host controller must halt the queue.

- NAK. The target endpoint Nak'd the full- or low-speed transaction. The state of the transfer is not advanced and the state is exited. If the *PidCode* is a SETUP, then the Nak response is a protocol error. The *XactErr* status bit is set to one and the *CErr* field is decremented.
- STALL. The target endpoint responded with a STALL handshake. The host controller sets the *halt* bit in the status byte, retires the qTD but does not attempt to advance the queue.

If the *PidCode* indicates an IN, then any of following responses are expected:

- DATA0/1. On reception of data, the host controller ensures the PID matches the expected data toggle and checks CRC. If the packet is *good*, the host controller advances the state of the transfer, for example move the data pointer by the number of bytes received, decrement *BytesToTransfer* field by the number of bytes received, and toggle the *dt* bit. The host controller then exit this state. The response and advancement of transfer may trigger other processing events, such as retirement of the qTD and advancement of the queue.

If the data sequence PID does not match the expected, the data is ignored, the transfer state is not advanced and this state is exited. If the *PidCode* indicates an OUT/SETUP, then any of following responses are expected:

- ACK. The target endpoint accepted the data, so the host controller must advance the state of the transfer. The *Current Offset* field is incremented by *Maximum Packet Length* or *Bytes to Transfer*, whichever is less. The field *Bytes To Transfer* is decremented by the same amount and the data toggle bit (*dt*) is toggled. The host controller then exit this state.
- Advancing the transfer state may cause other processing events such as retirement of the qTD and advancement of the queue (see Managing Control/Bulk/Interrupt Transfers through Queue Heads).

65.4.3.12.2 Split Transaction Interrupt

Split-transaction Interrupt-IN/OUT endpoints are managed through the same data structures used for high-speed interrupt endpoints. They both co-exist in the periodic schedule.

Queue heads/qTDs offer the set of features required for reliable data delivery, which is characteristic to interrupt transfer types. The split-transaction protocol is managed completely within this defined functional transfer framework. For example, for a high-speed endpoint, the host controller visits a queue head, execute a high-speed transaction (if criteria are met) and advance the transfer state (or not) depending on the results of the entire transaction. For low- and full-speed endpoints, the details of the *execution* phase are different (that is takes more than one bus transaction to complete), but the remainder of the operational framework is intact. This means that the transfer advancement, and so on, occurs as defined in Managing Control/Bulk/Interrupt Transfers through Queue Heads, but only occurs on the completion of a split transaction.

65.4.3.12.2.1 Split Transaction Scheduling Mechanisms for Interrupt

Full- and low-speed Interrupt queue heads have an *EPS* field indicating full- or low-speed and have a non-zero *S-mask* field.

The host controller can detect this combination of parameters and assume the endpoint is a periodic endpoint. Low- and full-speed interrupt queue heads require the use of the split transaction protocol. The host controller sets the Endpoint Type (ET) field in the split token to indicate the transaction is an interrupt. These transactions are managed through a transaction translator's periodic pipeline. Software should not set these fields to indicate the queue head is an interrupt unless the queue head is used in the periodic schedule.

System software manages the per/transaction translator periodic pipeline by budgeting and scheduling exactly during which micro-frames the start-splits and complete-splits for each endpoint occurs. The characteristics of the transaction translator are such that the high-speed transaction protocol must execute during explicit micro-frames, or the data or response information in the pipeline is lost.

The following figure illustrates the general scheduling boundary conditions that are supported by the EHCI periodic schedule and queue head data structure. The S and ^{C}X labels indicate micro-frames where software can schedule start-splits and complete splits (respectively).

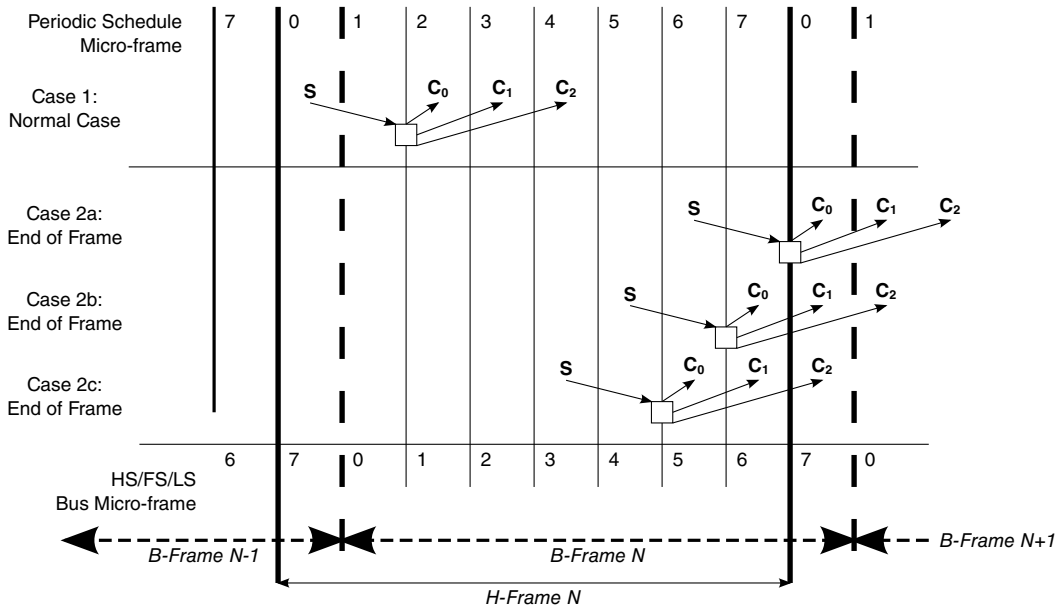


Figure 65-21. Split Transaction, Interrupt Scheduling Boundary Conditions

The scheduling cases are:

- Case 1: The normal scheduling case is where the entire split transaction is completely bounded by a frame (*H-Frame* in this case).
- Case 2a through Case 2c: The USB 2.0 Hub pipeline rules states clearly, when and how many complete-splits must be scheduled to account for earliest to latest execution on the full/low-speed link. The complete-splits may span the *H-Frame* boundary when the start-split is in micro-frame 4 or later. When this occurs, the *H-Frame* to *B-Frame* alignment requires that the queue head be reachable from consecutive periodic frame list locations. System software cannot build an efficient schedule that satisfies this requirement unless it uses FSTNs.

The figure below illustrates the general layout of the periodic schedule.

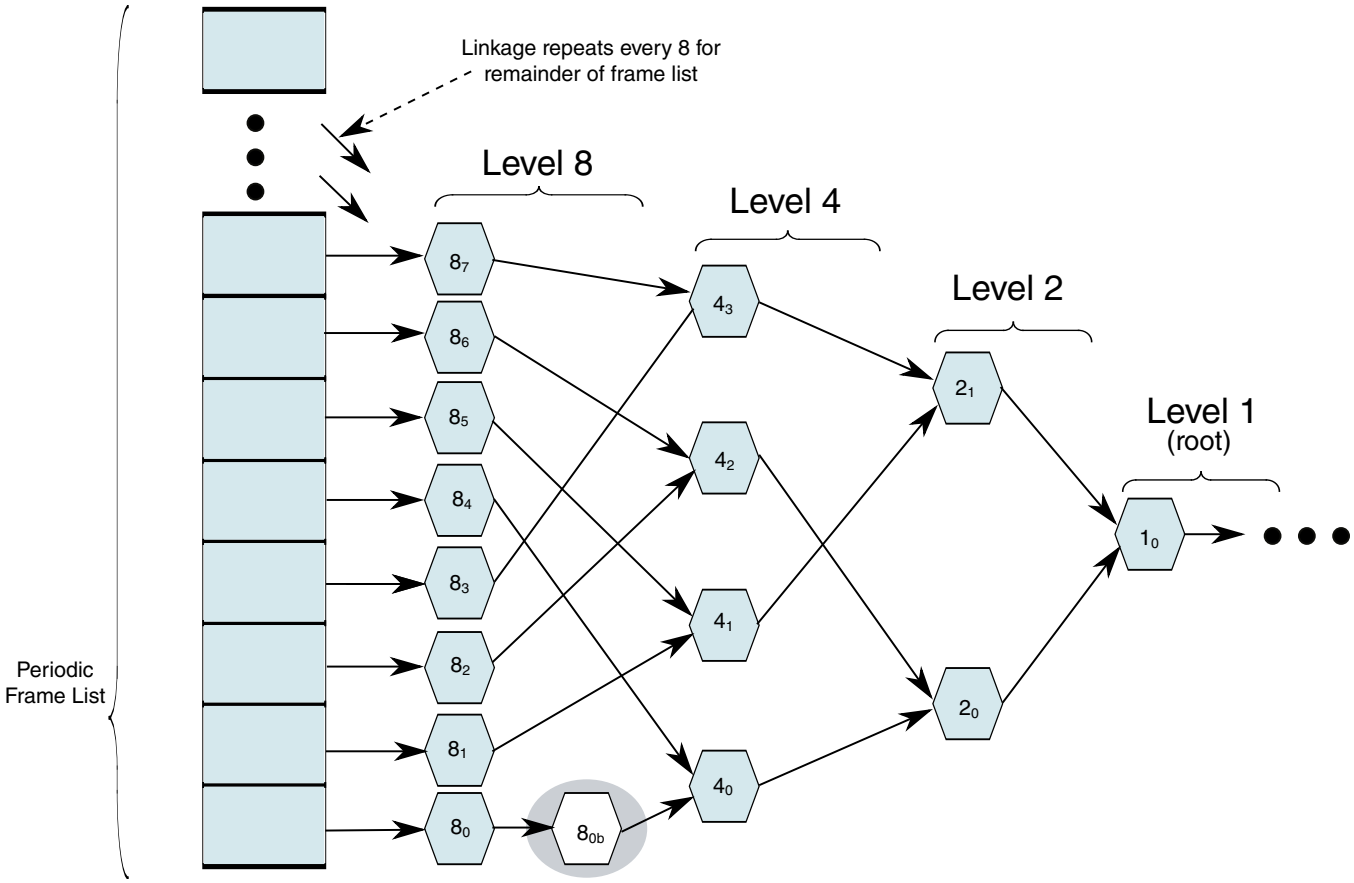


Figure 65-22. General Structure of EHCI Periodic Schedule Utilizing Interrupt Spreading

The periodic frame list is effectively the leaf level a binary tree, which is always traversed leaf to root. Each level in the tree corresponds to a 2^N poll rate. Software can efficiently manage periodic bandwidth on the USB by *spreading* interrupt queue heads that have the same poll rate requirement across all the available paths from the frame list. For example, system software can schedule eight poll rate 8 queue heads and account for them once in the high-speed bus bandwidth allocation.

When an endpoint is allocated an execution footprint that spans a frame boundary, the queue head for the endpoint must be reachable from consecutive locations in the frame list. An example would be if 8_{0b} where such an endpoint. Without additional support on the interface, to get 8_{0b} reachable at the correct time, software would have to link 8_1 to 8_{0b}. It would then have to move 4_1 and everything linked after into the same path as 4_0. This upsets the integrity of the binary tree and disallows the use of the spreading technique.

FSTN data structures are used to preserve the integrity of the binary-tree structure and enable the use of the spreading technique. Host Controller Operational Model for FSTNs defines the hardware and software operational model requirements for using FSTNs.

The following queue head fields are initialized by system software to instruct the host controller when to execute portions of the split-transaction protocol:

- *SplitXState*. This is single bit residing in the *Status* field of a queue head (see Table 65-23). This bit is used to track the current state of the split transaction.
- *Frame S-mask*. This is a bit-field where-in system software sets a bit corresponding to the micro-frame (within an *H-Frame*) that the host controller should execute a start-split transaction. This is always qualified by the value of the *SplitXState* bit in the *Status* field of the queue head. For example, referring to Figure 65-21, case one, the *S-mask* would have a value of 00000001b indicating that if the queue head is traversed by the host controller, and the *SplitXState* indicates Do_Start, and the current micro-frame as indicated by FRINDEX[2:0] is 0, then execute a start-split transaction.
- *Frame C-mask.* This is a bit-field where system software sets one or more bits corresponding to the micro-frames (within an *H-Frame*) that the host controller should execute complete-split transactions. The interpretation of this field is always qualified by the value of the *SplitXState* bit in the *Status* field of the queue head. For example, referring to Figure 65-21, case one, the *C-mask* would have a value of 00011100b indicating that if the queue head is traversed by the host controller, and the *SplitXState* indicates Do_Complete, and the current micro-frame as indicated by FRINDEX[2:0] is 2, 3, or 4, then execute a complete-split transaction. It is software's responsibility to ensure that the translation between *H-Frames* and *B-Frames* is correctly performed when setting bits in *S-mask* and *C-mask*

65.4.3.12.2.2 Host Controller Operational Model for FSTNs

The FSTN data structure is used to manage Low/Full-speed interrupt queue heads that need to be reached from consecutive frame list locations (that is boundary cases 2a through 2c).

An FSTN is essentially a *back pointer*, similar in intent to the back pointer field in the siTD data structure (see siTD Back Link Pointer).

This feature provides software a simple primitive to save a schedule position, redirect the host controller to traverse the necessary queue heads in the previous frame, then restore the original schedule position and complete normal traversal.

The four components to the use of FSTNs:

- FSTN data structure.
- A *Save Place* indicator. This is always an FSTN with its *Back Path Link Pointer.T-bit* set to zero.

- A *Restore* indicator. This is always an FSTN with its *Back Path Link Pointer.T-bit* set to one.
- Host controller FSTN traversal rules.

When the host controller encounters an FSTN during micro-frames 2 through 7 it simply follows the node's *Normal Path Link Pointer* to access the next schedule data structure.

NOTE

The FSTN's *Normal Path Link Pointer.T-bit* may set to one, which the host controller must interpret as the end of periodic list mark.

When the host controller encounters a *Save-Place* FSTN in micro-frames 0 or 1, it saves the value of the *Normal Path Link Pointer* and set an internal flag indicating that it is executing in *Recovery Path* mode. *Recovery Path* mode modifies the host controller's rules for how it traverses the schedule and limits which data structures is considered for execution of bus transactions. The host controller continues executing in *Recovery Path* mode until it encounters a *Restore* FSTN or it determines that it has reached the end of the micro-frame (see details in the list below).

The rules for schedule traversal and limited execution while in *Recovery Path* mode are:

- Always follow the *Normal Path Link Pointer* when it encounters an FSTN that is a *Save-Place* indicator. The host controller must not recursively follow *Save-Place* FSTNs. Therefore, while executing in *Recovery Path* mode, it must never follow an FSTN's *Back Path Link Pointer*.
- Do not process an siTD or, iTD data structure. Simply follow its *Next Link Pointer*.
- Do not process a QH (Queue Head) whose *EPS* field indicates a high-speed device. Simply follow its *Horizontal Link Pointer*.
- When a QH's *EPS* field indicates a Full/Low-speed device, the host controller considers only it for execution if its *SplitXState* is DoComplete (note: this applies whether the *PID Code* indicates an IN or an OUT). See Execute Transaction and Tracking Split Transaction Progress for Interrupt Transfers for a complete list of additional conditions that must be met in general for the host controller to issue a bus transaction.
  - The host controller must not execute a Start-split transaction while executing in *Recovery Path* mode. See Periodic Isochronous - Do Complete Split for special handling when in *Recovery Path* mode.
- Stop traversing the *recovery path* when it encounters an FSTN that is a *Restore* indicator. The host controller unconditionally uses the saved value of the *Save-Place* FSTN's *Normal Path Link Pointer* when returning to the normal path traversal. The

host controller must clear the context of executing a *Recovery Path* when it restores schedule traversal to the *Save-Place* FSTN's *Normal Path Link Pointer*.

- If the host controller determines that there is not enough time left in the micro-frame to complete processing of the periodic schedule, it abandons traversal of the recovery path, and clears the context of executing a recovery path. The result is that at the start of the next consecutive micro-frame, the host controller starts traversal at the frame list.

An example traversal of a periodic schedule that includes FSTNs is illustrated in the following figure.

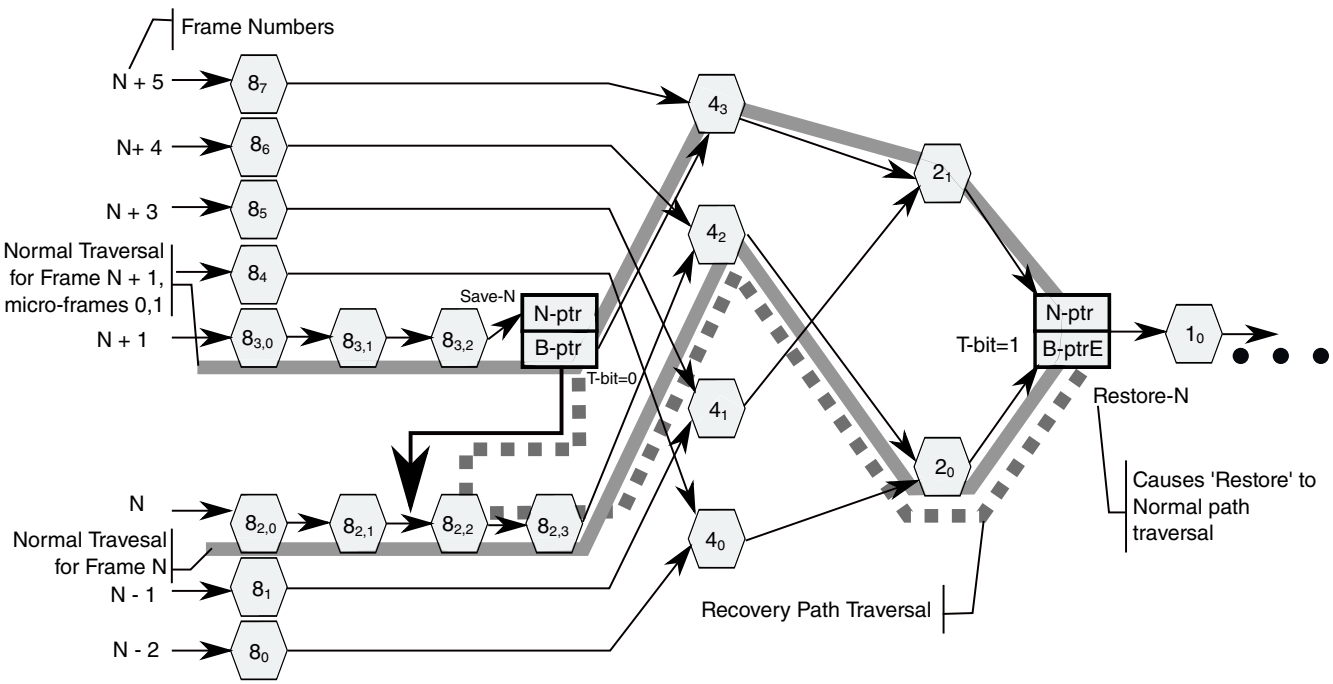


Figure 65-23. Example Host Controller Traversal of Recovery Path via FSTNs

In frame N+1 (micro-frames 0 and 1), when the host controller encounters Save-Path FSTN (Save-N), it observes that Save-N.Back Path Link Pointer.T-bit is zero (definition of a Save-Path indicator). The host controller saves the value of Save-N.Normal Path Link Pointer and follows Save-N.Back Path Link Pointer. At the same time, it sets an internal flag indicating that it is now in *Recovery Path* mode (the recovery path is annotated in the figure above with a large dashed line). The host controller continues traversing data structures on the recovery path and executing only those bus transactions as noted above, on the recovery path until it reaches Restore FSTN (Restore-N). Restore-N.Back Path Link Pointer.T-bit is set to one (definition of a Restore indicator), so the host controller exits *Recovery Path* mode by clearing the internal *Recovery Path* mode flag and commences (restores) schedule traversal using the saved value of the *Save-Place* FSTN's *Normal Path Link Pointer* (for example Save-N.Normal Path Link Pointer). The nodes traversed during these micro-frames include: {$8_{3.0}$, $8_{3.1}$, $8_{3.2}$, Save-A, $8_{2.2}$, $8_{2.3}$, 4_2,

2_0, Restore-N, 4_3, 2_1, Restore-N, 1_0 …}. The nodes on the recovery-path are in bold. In frame N (micro-frames 0-7), for this example, the host controller traverses all of the schedule data structures utilizing the *Normal Path Link Pointers* in any FSTNs it encounters. This is because the host controller has not yet encountered a *Save-Place* FSTN so it not executing in *Recovery Path* mode. When it encounters the *Restore* FSTN, (Restore-N), during micro-frames 0 and 1, it uses Restore-N.Normal Path Link Pointer to traverse to the next data structure (that is normal schedule traversal). This is because the host controller must use a Restore FSTN's *Normal Path Link Pointer* when not executing in a *Recovery-Path* mode. The nodes traversed during frame N include: {$8_{2.0}$, $8_{2.1}$, $8_{2.2}$, $8_{2.3}$, 4_2, 2_0, Restore-N, 1_0 …}.

In frame N+1 (micro-frames 2-7), when the host controller encounters Save-Path FSTN Save-N, it unconditionally follows Save-N.Normal Path Link Pointer. The nodes traversed during these micro-frames include: {$8_{3.0}$, $8_{3.1}$, $8_{3.2}$, Save-A, 4_3, 2_1, Restore-N, 1_0 …}.

65.4.3.12.2.3 Software Operational Model for FSTNs

Software must create a consistent, coherent schedule for the host controller to traverse.

When using FSTNs, system software must adhere to the following rules:

- Each *Save-Place* indicator requires a matching *Restore* indicator.
  - The *Save-Place* indicator is an FSTN with a valid *Back Path Link Pointer and T-bit* equal to zero.
    - *Back Path Link Pointer.Typ* field must be set to indicate the referenced data structure is a queue head. The *Restore* indicator is an FSTN with its *Back Path Link Pointer.T-bit* set to one.
  - A *Restore* FSTN may be matched to one or more *Save-Place* FSTNs. For example, if the schedule includes a poll-rate 1 level, then system software only needs to place a *Restore* FSTN at the beginning of this list in order to match all possible *Save-Place* FSTNs.
- If the schedule does not have elements linked at a poll-rate level of one, and one or more *Save-Place* FSTNs are used, then System Software must ensure the *Restore* FSTN's *Normal Path Link Pointer*'s *T-bit* is set to one, as this is used to mark the end of the periodic list.
- When the schedule does have elements linked at a poll rate level of one, a *Restore* FSTN must be the first data structure on the poll rate one list. All traversal paths from the frame list converge on the poll-rate one list. System software must ensure that *Recovery Path* mode is exited before the host controller is allowed to traverse the poll rate level one list.
- A *Save-Place* FSTN's *Back Path Link Pointer* must reference a queue head data structure. The referenced queue head must be reachable from the previous frame list

location. In other words, if the *Save-Place* FSTN is reachable from frame list offset N, then the FSTN's *Back Path Link Pointer* must reference a queue head that is reachable from frame list offset N-1.

Software should make the schedule as efficient as possible. What this means in this context is that software should have no more than one *Save-Place* FSTN reachable in any single frame. Note there is times when two (or more, depending on the implementation) could exist as full/low-speed footprints change with bandwidth adjustments. This could occur, for example when a bandwidth re-balance causes system software to move the *Save-Place* FSTN from one poll rate level to another. During the transition, software must preserve the integrity of the previous schedule until the new schedule is in place.

65.4.3.12.2.4 Tracking Split Transaction Progress for Interrupt Transfers

To correctly maintain the data stream, the host controller must be able to detect and report errors where data is lost.

For interrupt-IN transfers, data is lost when it makes it into the USB 2.0 hub, but the USB 2.0 host system is unable to get it from the USB 2.0 Hub and into the system before it expires from the transaction translator pipeline.

When a lost data condition is detected, the queue must be halted, thus signaling system software to recover from the error. A data-loss condition exists whenever a start-split is issued, accepted and successfully executed by the USB 2.0 Hub, but the complete-splits get unrecoverable errors on the high-speed link, or the complete-splits do not occur at the correct times. One reason complete-splits might not occur at the right time would be due to host-induced system hold-offs that cause the host controller to miss bus transactions because it cannot get timely access to the schedule in system memory.

The same condition can occur for an interrupt-OUT, but the result is not an endpoint halt condition, but rather effects only the progress of the transfer. The queue head has the following fields to track the progress of each split transaction. These fields are used to keep incremental state about which (and when) portions have been executed.

- *C-prog-mask*. This is an eight-bit bit-vector where the host controller keeps track of which complete-splits have been executed. Due to the nature of the Transaction Translator periodic pipeline, the complete-splits need to be executed in-order. The host controller needs to detect when the complete-splits have not been executed in order. This can only occur due to system hold-offs where the host controller cannot get to the memory-based schedule. *C-prog-mask* is a simple bit-vector that the host controller sets one of the *C-prog*-mask bits for each complete-split executed. The bit position is determined by the micro-frame number in which the complete-split was executed. The host controller always checks *C-prog-mask* before executing a

complete-split transaction. If the previous complete-splits have not been executed then it means one (or more) have been skipped and data has potentially been lost.
- *FrameTag*. This field is used by the host controller during the complete-split portion of the split transaction to tag the queue head with the frame number (*H-Frame* number) when the next complete split must be executed.
- *S-bytes*. This field can be used to store the number of data payload bytes sent during the start-split (if the transaction was an OUT). The *S-bytes* field must be used to accumulate the data payload bytes received during the complete-splits (for an IN).

65.4.3.12.2.5 Split Transaction Execution State Machine for Interrupt

In the following presentation, all references to micro-frame are in the context of a micro-frame within an *H-Frame.*

As with asynchronous Full- and Low-speed endpoints, a split-transaction state machine is used to manage the split transaction sequence.

Aside from the fields defined in the queue head for scheduling and tracking the split transaction, the host controller calculates one internal mechanism that is also used to manage the split transaction. The internal calculated mechanism is:

- *cMicroFrameBit* is a single-bit encoding of the current micro-frame number. It is an eight-bit value calculated by the host controller at the beginning of every micro-frame. It is calculated from the three least significant bits of the *FRINDEX* register (that is, *cMicroFrameBit* = (1 shifted-left(*FRINDEX*[2:0])). The *cMicroFrameBit* has at most one bit asserted, which always corresponds to the current micro-frame number. For example, if the current micro-frame is 0, then *cMicroFrameBit* will equal 00000001b. The variable *cMicroFrameBit* is used to compare against the *S-mask* and *C-mask* fields to determine whether the queue head is marked for a start- or complete-splt transaction for the current micro-frame.

The following figure illustrates the state machine for managing a complete interrupt split transaction. There are two phases to each split transaction. The first is a single start-split transaction, which occurs when the *SplitXState* is at Do_Start and the single bit in *cMicroFrameBit* has a corresponding bit active in *QH.S-mask*. The transaction translator does not acknowledge the receipt of the periodic start-split, so the host controller unconditionally transitions the state to Do_Complete. Due to the available jitter in the transaction translator pipeline, there will be more than one complete-split transaction scheduled by software for the Do_Complete state. This translates simply to the fact that there are multiple bits set to a one in the *QH.C-mask* field.

The host controller keeps the queue head in the Do_Complete state until the split transaction is complete (see definition below), or an error condition triggers the *three-strikes-rule* (for example, after the host tries the same transaction three times, and each

encounters an error, the host controller will stop retrying the bus transaction and halt the endpoint, thus requiring system software to detect the condition and perform system-dependent recovery).

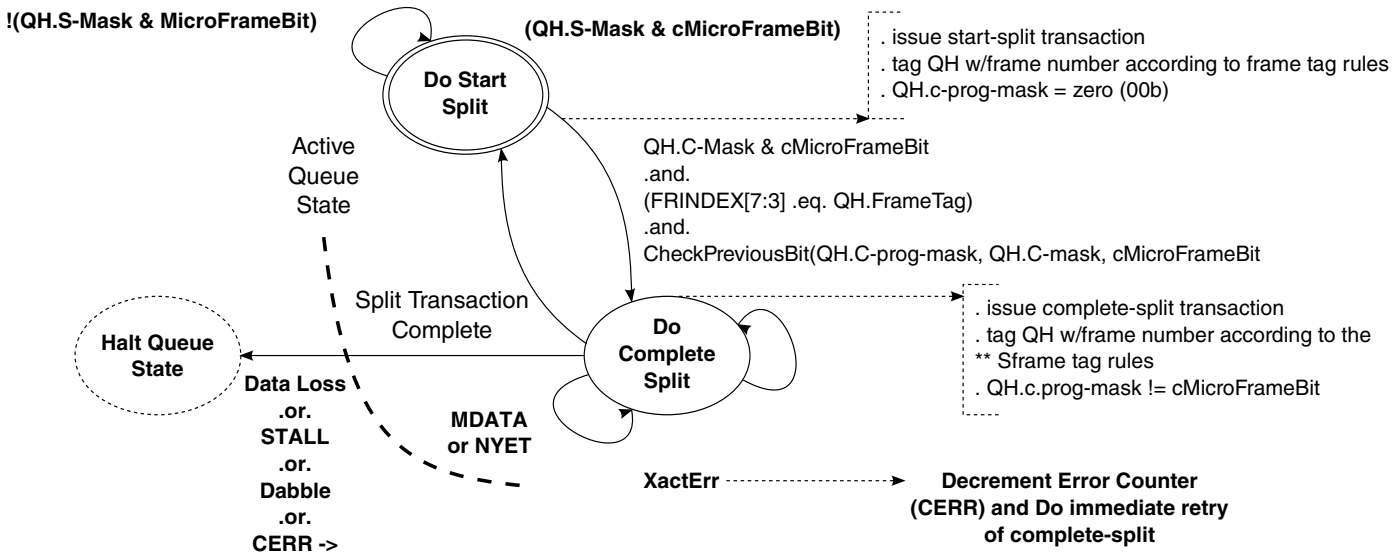


Figure 65-24. Split Transaction State Machine for Interrupt

See Previous Section for the frame tag management rules.

Periodic Interrupt - Do Start Split

This is the state software must initialize a full- or low-speed interrupt queue head *StartXState* bit. This state is entered from the Do_Complete Split state only after the split transaction is complete. This occurs when one of the following events occur: The transaction translator responds to a complete-split transaction with one of the following:

- NAK. A NAK response is a propagation of the full- or low-speed endpoint's NAK response.
- ACK. An ACK response is a propagation of the full- or low-speed endpoint's ACK response. Only occurs on an OUT endpoint.
- DATA 0/1. Only occurs for INs. Indicates that this is the last of the data from the endpoint for this split transaction.
- ERR. The transaction on the low-/full-speed link below the transaction translator had a failure (for example, timeout, bad CRC, etc.).
- NYET (and Last). The host controller issued the last complete-split and the transaction translator responded with a NYET handshake. This means that the start-split was not correctly received by the transaction translator, so it never executed a transaction to the full- or low-speed endpoint, see Section Periodic Isochronous - Do Complete Split for the definition of 'Last'.

Each time the host controller visits a queue head in this state (once within the Execute Transaction state), it performs the following test to determine whether to execute a start-split.

- *QH.S-mask* is bit-wise anded with *cMicroFrameBit*.

If the result is non-zero, then the host controller will issue a start-split transaction. If the *PIDCode* field indicates an IN transaction, the host controller must zero-out the *QH.S-bytes* field. After the split-transaction has been executed, the host controller sets up state in the queue head to track the progress of the complete-split phase of the split transaction. Specifically, it records the expected frame number into *QH.FrameTag* field (see Section), set *C-prog-mask* to zero (00h), and exits this state. Note that the host controller must not adjust the value of *CErr* as a result of completion of a start-split transaction.

Periodic Interrupt - Do Complete Split

This state is entered unconditionally from the Do Start Split state after a start-split transaction is executed on the bus. Each time the host controller visits a queue head in this state (once within the Execute Transaction state), it checks to determine whether a complete-split transaction should be executed now.

There are four tests to determine whether a complete-split transaction should be executed.

- Test A. *cMicroFrameBit* is bit-wise anded with *QH.C-mask* field. A non-zero result indicates that software scheduled a complete-split for this endpoint, during this micro-frame.
- Test B. *QH.FrameTag* is compared with the current contents of *FRINDEX*[7:3]. An equal indicates a match.
- Test C. The complete-split progress bit vector is checked to determine whether the previous bit is set, indicating that the previous complete-split was appropriately executed. An example algorithm for this test is provided below:

```
Algorithm Boolean CheckPreviousBit(QH.C-prog-mask, QH.C-mask, cMicroFrameBit)
Begin
-- Return values:
-- TRUE - no error
-- FALSE - error
--
Boolean rvalue = TRUE;
previousBit = cMicroframeBit logical-rotate-right(1)
-- Bit-wise anding previousBit with C-mask indicates
-- whether there was an intent
-- to send a complete split in the previous micro-frame. So,
-- if the
-- 'previous bit' is set in C-mask, check C-prog-mask to
-- make sure it
-- happened.
If (previousBit bitAND QH.C-mask)then
                                If not(previousBit bitAND QH.C-prog-mask) then
                      rvalue = FALSE;
          End if
End If
-- If the C-prog-mask already has a one in this bit position,
```

```
-- then an aliasing
-- error has occurred. It will probably get caught by the
-- FrameTag Test, but
-- at any rate it is an error condition that as detectable here
-- should not allow
-- a transaction to be executed.
If (cMicroFrameBit bitAND QH.C-prog-mask) then
 rvalue = FALSE;
End if
return (rvalue)
End Algorithm
```

- Test D. Check to see if a start-split should be executed in this micro-frame. Note this is the same test performed in the Do Start Split state (see Section Periodic Isochronous - Do Start Split). Whenever it evaluates to TRUE and the controller is NOT processing in the context of a *Recovery Path* mode, it means a start-split should occur in this micro-frame. Test D and Test A evaluating to TRUE at the same time is a system software error. Behavior is undefined.

If (A .and. B .and. C .and. not(D)) then the host controller will execute a complete-split transaction. When the host controller commits to executing the complete-split transaction, it updates *QH.C-prog-mask* by bit-ORing with *cMicroFrameBit*. On completion of the complete-split transaction, the host controller records the result of the transaction in the queue head and sets *QH.FrameTag* to the expected *H-Frame* number (see Section). The effect to the state of the queue head and thus the state of the transfer depends on the response by the transaction translator to the complete-split transaction. The following responses have the effects (note that any responses that result in decrementing of the *CErr* will result in the queue head being halted by the host controller if the result of the decrement is zero):

- NYET (and Last). On each NYET response, the host controller checks to determine whether this is the last complete-split for this split transaction. Last is defined in this context as the condition where all of the scheduled complete-splits have been executed. If it is the last complete-split (with a NYET response), then the transfer state of the queue head is not advanced (never received any data) and this state exited. The transaction translator must have responded to all the clompete-splits with NYETs, meaning that the start-split issued by the host controller was not received. The start-split should be retried at the next poll period.
- The test for whether this is the Last complete split can be performed by XOR *QH.C-mask* with *QH.C-prog-mask*. If the result is all zeros then all complete-splits have been executed. When this condition occurs, the *XactErr* status bit is set to a one and the *CErr* field is decremented.
- NYET (and not Last). See above description for testing for Last. The complete-split transaction received a NYET response from the transaction translator. Do not update any transfer state (except for *C-prog-mask* and *FrameTag*) and stay in this state. The host controller must not adjust *CErr* on this response.

- Transaction Error (XactErr). Timeout, data CRC failure, etc. The *CErr* field is decremented and the *XactErr* bit in the *Status* field is set to a one. The complete split transaction is *immediately* retried (if *Cerr* is non-zero).If there is not enough time in the micro-frame to complete the retry and the endpoint is an IN, or *CErr* is decremented to a zero from a one, the queue is halted. If there is not enough time in the micro-frame to complete the retry and the endpoint is an OUT and *CErr* is not zero, then this state is exited (that is, return to Do Start Split). This results in a retry of the entire OUT split transaction, at the next poll period. Refer to Chapter 11 Hubs (specifically the section full- and low-speed Interrupts) in the USB Specification Revision 2.0 for detailed requirements on why these errors must be immediately retried.
- ACK. This can only occur if the target endpoint is an OUT. The target endpoint ACK'd the data and this response is a propagation of the endpoint ACK up to the host controller. The host controller must advance the state of the transfer. The *Current Offset* field is incremented by *Maximum Packet Length* or *Bytes to Transfer*, whichever is less. The field *Bytes To Transfer* is decremented by the same amount. And the data toggle bit (*dt*) is toggled. The host controller will then exit this state for this queue head. The host controller must reload *CErr* with maximum value on this response. Advancing the transfer state may cause other process events such as retirement of the qTD and advancement of the queue (see Section Managing Control/Bulk/Interrupt Transfers through Queue Heads).
- MDATA. This response will only occur for an IN endpoint. The transaction translator responded with zero or more bytes of data and an MDATA PID. The incremental number of bytes received is accumulated in *QH.S-bytes*. The host controller must not adjust *CErr* on this response.
- DATA0/1. This response may only occur for an IN endpoint. The number of bytes received is added to the accumulated byte count in *QH.S-bytes*. The state of the transfer is advanced by the result and the host controller will exit this state for this queue head.
- Advancing the transfer state may cause other processing events such as retirement of the qTD and advancement of the queue (see Section Managing Control/Bulk/Interrupt Transfers through Queue Heads).
- If the data sequence PID does not match the expected, the entirety of the data received in this split transaction is ignored, the transfer state is not advanced and this state is exited.
- NAK. The target endpoint Nak'd the full- or low-speed transaction. The state of the transfer is not advanced, and this state is exited. The host controller must reload *CErr* with maximum value on this response.

- ERR. There was an error during the full- or low-speed transaction. The ERR status bit is set to a one, *Cerr* is decremented, the state of the transfer is not advanced, and this state is exited.
- STALL. The queue is halted (an exit condition of the Execute Transaction state). The status field bits: *Active* bit is set to zero and the *Halted* bit is set to a one and the qTD is retired. Responses which are not enumerated in the list or which are received out of sequence are illegal and may result in undefined host controller behavior. The other possible combinations of tests A, B, C, and D may indicate that data or response was lost. The table below lists the possible combinations and the appropriate action.

Table 65-47. Interrupt IN/OUT Do Complete Split State Execution Criteria

Condition	Action	Description
not(A) not(D)	Ignore QHD	Neither a start nor complete-split is scheduled for the current micro-frame.Host controller should continue walking the schedule.
A not(C)	If PIDCode = IN Halt QHD If PIDCode = OUT Retry start-split	Progress bit check failed. These means a complete-split has been missed. There is the possibility of lost data. If *PIDCode* is an IN, then the Queue head must be halted. If *PIDCode* is an OUT, then the transfer state is not advanced and the state exited (for example, start-split is retried). This is a host-induced error and does not effect *CER*R. In either case, set the *Missed Micro-frame* bit in the status field to a one.
A not(B) C	If PIDCode = IN Halt QHD If PIDCode = OUT Retry start-split	*QH.FrameTag* test failed. This means that exactly one or more *H-Frames* have been skipped. This means complete-splits and have missed. There is the possibility of lost data. If *PIDCode* is an IN, then the Queue head must be halted. If *PIDCode* is an OUT, then the transfer state is not advanced and the state exited (for example, start-split is retried). This is a host-induced error and does not effect *CER*R. In either case, set the *Missed Micro-frame* bit in the status field to a one.
A B C not(D)	Execute complete-split	This is the non-error case where the host controller executes a complete-split transaction.
D	If PIDCode = IN Halt QHD If PIDCode = OUT Retry start-split	This is a degenerate case where the start-split was issued, but all of the complete-splits were skipped and all possible intervening opportunities to detect the missed data failed to fire. If *PIDCode* is an IN, then the Queue head must be halted. If *PIDCode* is an OUT, then the transfer state is not advanced and the state exited (for example, start-split is retried). This is a host-induced error and does not effect *CER*R. In either case, set the *Missed Micro-frame* bit in the status field to a one. Note: When executing in the context of a *Recovery Path* mode, the host controller is allowed to process the queue head and take the actions indicated above, or it may wait until the queue head is visited in the

Table 65-47. Interrupt IN/OUT Do Complete Split State Execution Criteria

		normal processing mode. Regardless, the host controller must not execute a start-split in the context of a executing in a *Recovery Path* mode.

Managing QH.FrameTag Field

The *QH.FrameTag* field in a queue head is completely managed by the host controller. The rules for setting *QH.FrameTag* are simple:

- Rule 1: If transitioning from Do Start Split to Do Complete Split and the current value of *FRINDEX*[2:0] is 6 *QH.FrameTag* is set to *FRINDEX*[7:3] + 1. This accommodates split transactions whose start-split and complete-splits are in different *H-Frames* (case 2a, see Figure 65-21).
- Rule 2: If the current value of *FRINDEX*[2:0] is 7, *QH.FrameTag* is set to *FRINDEX*[7:3] + 1. This accommodates staying in Do Complete Split for cases 2a, 2b, and 2c (Figure 65-21).
- Rule 3: If transitioning from Do_Start Split to Do Complete Split and the current value of *FRINDEX*[2:0] is not 6, or currently in Do Complete Split and the current value of (*FRINDEX*[2:0]) is not 7, *FrameTag* is set to *FRINDEX*[7:3]. This accommodates all other cases (Figure 65-21).

65.4.3.12.2.6 Rebalancing the periodic schedule

System software must occasionally adjust a periodic queue head's S-mask and C-mask fields during operation.

This need occurs when adjustments to the periodic schedule create a new bandwidth budget and one or more queue head's are assigned new execution footprints (that is, new S-mask and C-mask values).

It is imperative that System software must not update these masks to new values in the midst of a split transaction. In order to avoid any race conditions with the update, the EHCI host controller provides a simple assist to system software. System software sets the *Inactivate-on-next-Transaction* (*I*) bit to a one to signal the host controller that it intends to update the S-mask and C-mask on this queue head. System software will then wait for the host controller to observe the *I-bit* is a one and transition the *Active* bit to a zero. The rules for how and when the host controller sets the *Active* bit to zero are enumerated below:

- If the *Active* bit is a zero, no action is taken. The host controller does not attempt to advance the queue when the *I-bit* is a one.
- If the *Active* bit is a one and the *SplitXState* is DoStart (regardless of the value of *S-mask*), the host controller will simply set *Active* bit to a zero. The host controller is

not required to write the transfer state back to the *current* qTD. Note that if the *S-mask* indicates that a start-split is scheduled for the current micro-frame, the host controller must not issue the start-split bus transaction. It must set the *Active* bit to zero.

System software must save transfer state before setting the *I-bit* to a one. This is required so that it can correctly determine what transfer progress (if any) occurred after the *I-bit* was set to a one and the host controller executed its final bus-transaction and set *Active* to a zero.

After system software has updated the S-mask and C-mask, it must then reactivate the queue head. Because the *Active* bit and the *I-bit* cannot be updated with the same write, system software needs to use the following algorithm to coherently re-activate a queue head that has been stopped via the *I-bit*.

1. Set the *Halted* bit to a one, then
2. Set the *I-bit* to a zero, then
3. Set the *Active* bit to a one and the *Halted* bit to a zero in the same write.

Setting the *Halted* bit to a one inhibits the host controller from attempting to advance the queue between the time the *I-bit* goes to a zero and the *Active* bit goes to a one.

65.4.3.12.3 Split Transaction Isochronous

Full-speed isochronous transfers are managed using the split-transaction protocol through a USB 2.0 transaction translator in a USB2.0 Hub. The EHCI controller utilizes siTD data structure to support the special requirements of isochronous split-transactions.

This data structure uses the scheduling model of isochronous TDs (iTD, Section Isochronous (High-Speed) Transfer Descriptor (iTD)) (see Section Managing Isochronous Transfers Using iTDs for the operational model of iTDs) with the contiguous data feature provided by queue heads. This simple arrangement allows a single isochronous scheduling model and adds the additional feature that all data received from the endpoint (per split transaction) must land into a contiguous buffer.

65.4.3.12.3.1 Split Transaction Scheduling Mechanisms for Isochronous

Full-speed isochronous transactions are managed through a transaction translator's periodic pipeline. As with full- and low-speed interrupt, system software manages each transaction translator's periodic pipeline by budgeting and scheduling exactly during which micro-frames the start-splits and complete-splits for each full-speed isochronous endpoint occur.

The requirements described in Section Split Transaction Scheduling Mechanisms for Interrupt apply. The following figure illustrates the general scheduling boundary conditions that are supported by the EHCI periodic schedule. The ^{S}X and ^{C}X labels indicate micro-frames where software can schedule start- and complete-splits (respectively). The *H-Frame* boundaries are marked with a large, solid bold vertical line. The *B-Frame* boundaries are marked with a large, bold, dashed line. The bottom of the figure illustrates the relationship of an siTD to the *H-Frame*.

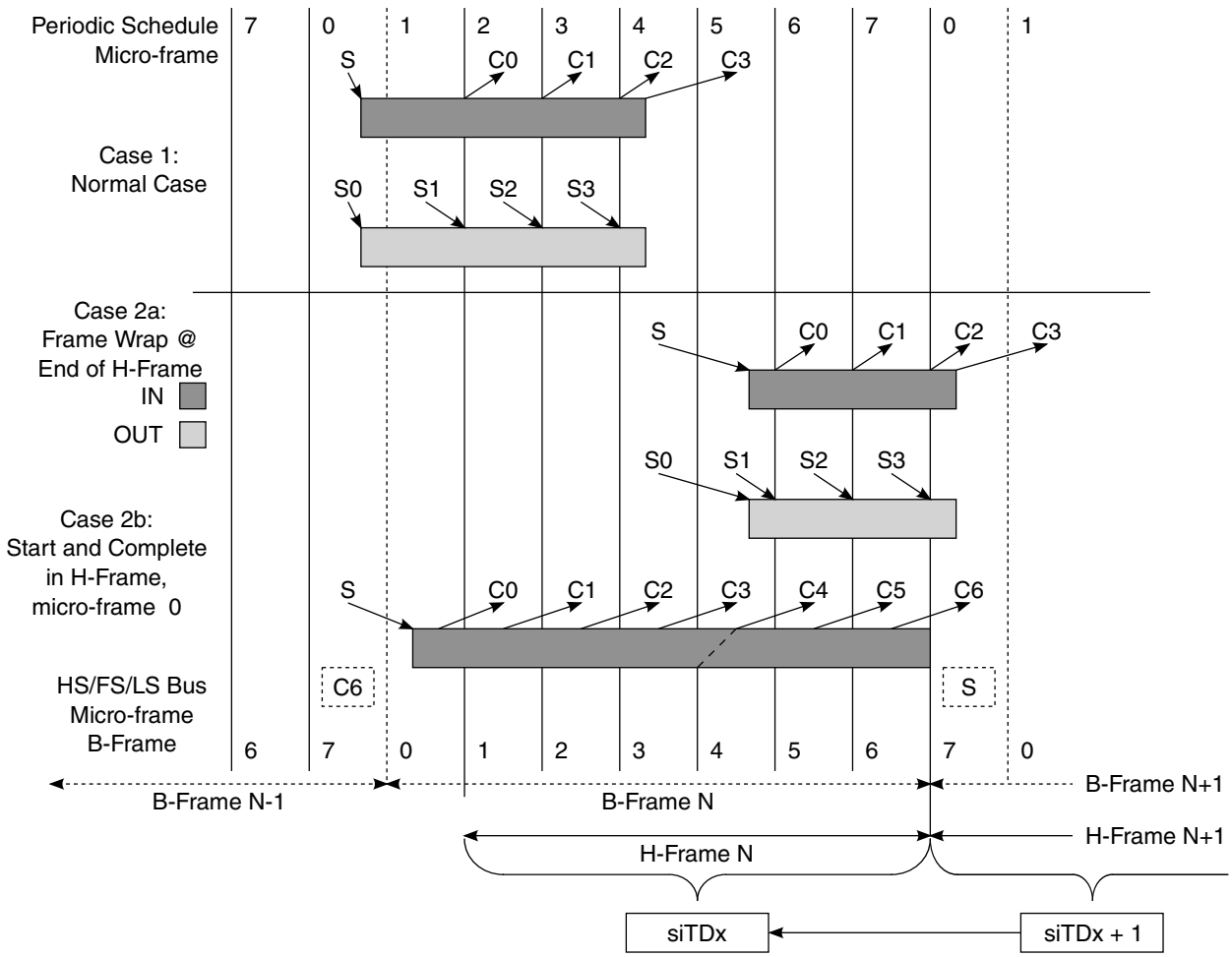


Figure 65-25. Split Transaction, Isochronous Scheduling Boundary Conditions

When the endpoint is an isochronous OUT, there are only start-splits, and no complete-splits. When the endpoint is an isochronous IN, there is at most one start-split and one to *N* complete-splits. The scheduling boundary cases are:

- *Case 1*: The entire split transaction is completely bounded by an *H-Frame*. For example: the start-splits and complete-splits are all scheduled to occur in the same *H-Frame*.

- *Case 2a*: This boundary case is where one or more (at most two) complete-splits of a split transaction IN are scheduled across an *H-Frame* boundary. This can only occur when the split transaction has the possibility of moving data in *B-Frame,* micro-frames 6 or 7 (*H-Frame* micro-frame 7 or 0). When an *H-Frame* boundary wrap condition occurs, the scheduling of the split transaction spans more than one location in the periodic list.(For example, it takes two siTDs in adjacent periodic frame list locations to fully describe the scheduling for the split transaction.)
- Although the scheduling of the split transaction may take two data structures, all of the complete-splits for each full-speed IN isochronous transaction must use only one data pointer. For this reason, siTDs contain a back pointer, the use of which is described below.
- Software must never schedule full-speed isochronous OUTs across an *H-Frame* boundary.
- *Case 2b*: This case can only occur for a very large isochronous IN. It is the only allowed scenario where a start-split and complete-split for the same endpoint can occur in the same micro-frame. Software must enforce this rule by scheduling the large transaction first. Large is defined to be anything larger than 579 byte maximum packet size.

A subset of the same mechanisms employed by full- and low-speed interrupt queue heads are employed in siTDs to schedule and track the portions of isochronous split transactions. The following fields are initialized by system software to instruct the host controller when to execute portions of the split transaction protocol.

- *SplitXState*. This is a single bit residing in the *Status* field of an siTD (see Figure 65-26). This bit is used to track the current state of the split transaction. The rules for managing this bit are described in Section Split Transaction Execution State Machine for Interrupt.
- *Frame S-mask*. This is a bit-field where-in system software sets a bit corresponding to the micro-frame (within an *H-Frame*) that the host controller should execute a start-split transaction. This is always qualified by the value of the *SplitXState* bit. For example, referring to the IN example in Figure 65-25, case one, the *S-mask* would have a value of 00000001b indicating that if the siTD is traversed by the host controller, and the *SplitXState* indicates Do Start Split, and the current micro-frame as indicated by USB_n_FRINDEX[2:0] is 0, then execute a start-split transaction.
- *Frame C-mask.* This is a bit-field where system software sets one or more bits corresponding to the micro-frames (within an *H-Frame*) that the host controller should execute complete-split transactions. The interpretation of this field is always qualified by the value of the *SplitXState* bit. For example, referring to the IN example in Figure 65-25, case one, the *C-mask* would have a value of 00111100b indicating that if the siTD is traversed by the host controller, and the *SplitXState* indicates Do

Complete Split, and the current micro-frame as indicated by *USB_n_FRINDEX*[2:0] is 2, 3, 4, or 5, then execute a complete-split transaction.

- *Back Pointer*. This field in a siTD is used to complete an IN split-transaction using the previous *H-Frame*'s siTD. This is only used when the scheduling of the complete-splits span an *H-Frame* boundary.

There exists a one-to-one relationship between a high-speed isochronous split transaction (including all start- and complete-splits) and one full-speed isochronous transaction. An siTD contains (amongst other things) buffer state and split transaction scheduling information. An siTD's buffer state always maps to one full-speed isochronous data payload. This means that for any full-speed transaction payload, a single siTD's data buffer must be used. This rule applies to both IN an OUTs. An siTD's scheduling information usually also maps to one high-speed isochronous split transaction. The exception to this rule is the *H-Frame* boundary wrap cases mentioned above.

The siTD data structure describes at most, one frame's worth of high-speed transactions and that description is strictly bounded within a frame boundary. The figure below illustrates some examples. On the top are examples of the full-speed transaction footprints for the boundary scheduling cases described above. In the middle are time-frame references for both the *B-Frames* (HS/FS/LS Bus) and the *H-Frames*. On the bottom is illustrated the relationship between the scope of an siTD description and the time references. Each *H-Frame* corresponds to a single location in the periodic frame list. The implication is that each siTD is reachable from a single periodic frame list location at a time.

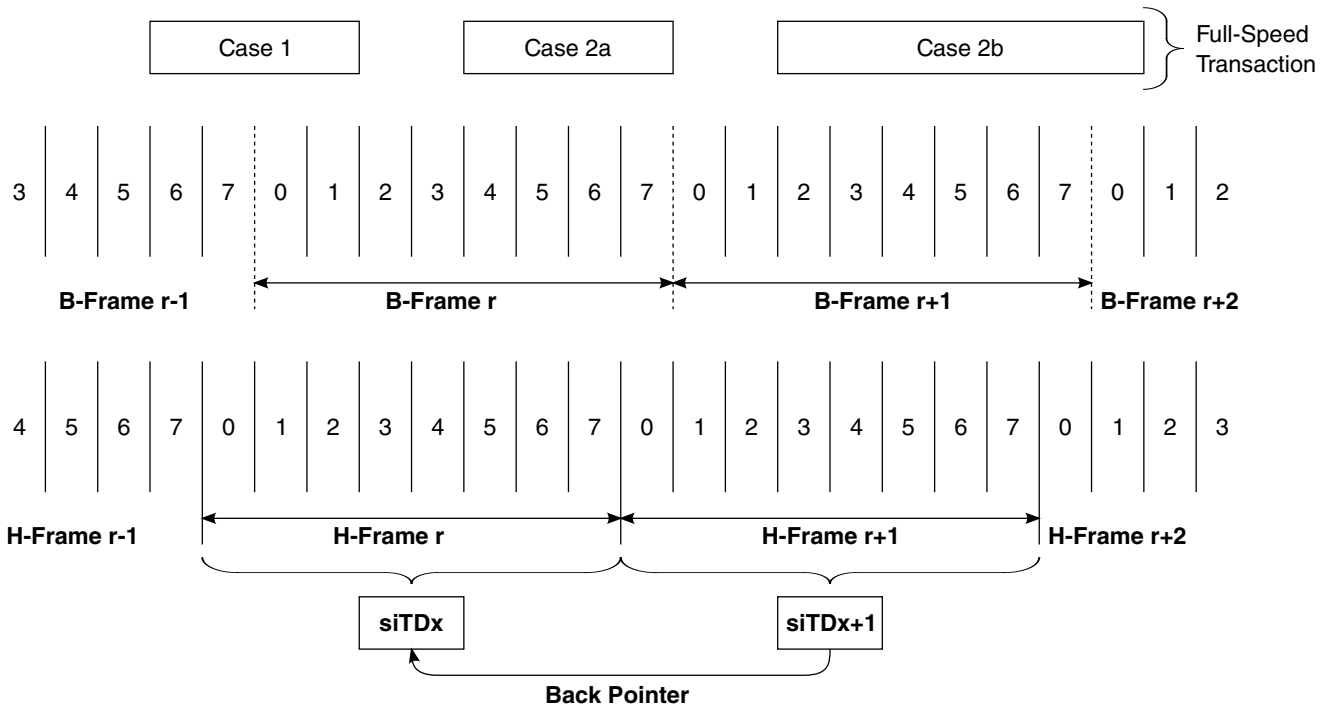


Figure 65-26. siTD Scheduling Boundary Examples

Each case is described below:

- *Case 1*: One siTD is sufficient to describe and complete the isochronous split transaction because the whole isochronous split transaction is tightly contained within a single *H-Frame*.
- *Case 2a, 2b*: Although both INs and OUTs can have these footprints, OUTs always take only one siTD to schedule. However, INs (for these boundary cases) require two siTDs to complete the scheduling of the isochronous split transaction $siTD_X$ is used to always issue the start-split and the first *N* complete-splits. The full-speed transaction (for these cases) can deliver data on the full-speed bus segment during micro-frame 7 of $H\text{-}Frame_{Y+1}$, or micro-frame 0 of $H\text{-}Frame_{Y+2}$. The complete splits are scheduled using $siTD_{X+2}$ (not shown). The complete-splits to extract this data must use the buffer pointer from $siTD_{X+1}$. The only way for the host controller to reach $siTD_{X+1}$ from $H\text{-}Frame_{Y+2}$is to use $siTD_{X+2}$'s back pointer. The host controller rules for when to use the back pointer are described is Section Periodic Isochronous - Do Complete Split .

Software must apply the following rules when calculating the schedule and linking the schedule data structures into the periodic schedule:

- Software must ensure that an isochronous split-transaction is started so that it will complete before the end of the *B-Frame*.
- Software must ensure that for a single full-speed isochronous endpoint, there is never a start-split and complete-split in *H-Frame, micro-frame 1*. This is mandated as a rule so that case 2a and case 2b can be discriminated. According to the core USB specification, the long isochronous transaction illustrated in Case 2b, could be scheduled so that the start-split was in micro-frame 1 of *H-Frame* N and the last complete-split would need to occur in micro-frame 1 of H-Frame N+1. However, it is impossible to discriminate between cases 2a and case 2b, which has significant impact on the complexity of the host controller.

65.4.3.12.3.2 Tracking Split Transaction Progress for Isochronous Transfers

To correctly maintain the data stream, the host controller must be able to detect and report errors where device to host data is lost. Isochronous endpoints do not employ the concept of a halt on error, however the host is required to identify and report per-packet errors observed in the data stream. This includes schedule traversal problems (skipped micro-frames), timeouts and corrupted data received.

In similar kind to interrupt split-transactions, the portions of the split transaction protocol must execute in the micro-frames they are scheduled. The queue head data structure used to manage full- and low-speed interrupt has several mechanisms for tracking when portions of a transaction have occurred. Isochronous transfers use siTDs, for their transfers, and the data structures are only reachable via the schedule in the exact micro-frame in which they are required (so all the mechanism employed for tracking in queue heads is not required for siTDs). Software has the option of reusing siTD several times in the complete periodic schedule. However, it must ensure that the results of split transaction N are consumed and the siTD reinitialized (activated) before the host controller gets back to the siTD (in a future micro-frame).

Split-transaction isochronous OUTs utilize a low-level protocol to indicate which portions of the split transaction data have arrived. Control over the low-level protocol is exposed in an siTD via the fields *Transaction Position* (*TP*) and *Transaction Count* (*T-count*). If the entire data payload for the OUT split transaction is larger than 188 bytes, there will be more than one start-split transaction, each of which require proper annotation. If host hold-offs occur, then the sequence of annotations received from the host will not be complete, which is detected and handled by the transaction translator. See Section Periodic Isochronous - Do Start Split for a description on how these fields are used during a sequence of start-split transactions.

The fields *siTD.T-Count* and *siTD.TP* are used by the host controller to drive and sequence the transaction position annotations. It is the responsibility of system software to properly initialize these fields in each siTD. Once the budget for a split-transaction

isochronous endpoint is established, *S-mask*, *T-Count*, and *TP* initialization values for all the siTD associated with the endpoint are constant. They remain constant until the budget for the endpoint is recalculated by software and the periodic schedule adjusted.

For IN-endpoints, the transaction translator simply annotates the response data packets with enough information to allow the host controller to identify the last data. As with split transaction Interrupt, it is the host controller's responsibility to detect when it has missed an opportunity to execute a complete-split. The following field in the siTD is used to track and detect errors in the execution of a split transaction for an IN isochronous endpoint.

- *C-prog-mask*. This is an eight-bit bit-vector where the host controller keeps track of which complete-splits have been executed. Due to the nature of the Transaction Translator periodic pipeline, the complete-splits need to be executed in-order. The host controller needs to detect when the complete-splits have not been executed in order. This can only occur due to system hold-offs where the host controller cannot get to the memory-based schedule. *C-prog-mask* is a simple bit-vector that the host controller sets a bit for each complete-split executed. The bit position is determined by the micro-frame (USB_n_FRINDEX[2:0]) number in which the complete-split was executed. The host controller always checks *C-prog-mask* before executing a complete-split transaction. If the previous complete-splits have not been executed, then it means one (or more) have been skipped and data has potentially been lost. System software is required to initialize this field to zero before setting an siTD's *Active* bit to a one.

If a transaction translator returns with the final data before all of the complete-splits have been executed, the state of the transfer is advanced so that the remaining complete-splits are not executed. Refer to Section Asynchronous - Do Complete Split for a description on how the state of the transfer is advanced. It is important to note that an IN siTD is retired based solely on the responses from the Transaction Translator to the complete-split transactions. This means, for example, that it is possible for a transaction translator to respond to a complete-split with an MDATA PID. The number of bytes in the MDATA's data payload could cause the siTD field *Total Bytes to Transfer* to decrement to zero. This response can occur, before all of the scheduled complete-splits have been executed. In other interface, data structures (for example, high-speed data streams through queue heads), the transition of *Total Bytes to Transfer* to zero signals the end of the transfer and results in setting of the *Active* bit to zero. However, in this case, the result has not been delivered by the Transaction Translator and the host must continue with the next complete-split transaction to extract the residual transaction state. This scenario occurs because of the pipeline rules for a Transaction Translator (see Chapter 11 of the Universal Serial Bus Revision 2.0). In summary the periodic pipeline rules require that on a micro-frame boundary, the Transaction Translator will hold the final two bytes received (if it has not seen an End Of Packet (EOP)) in the full-speed bus pipe stage and give the

remaining bytes to the high-speed pipeline stage. At the micro-frame boundary, the Transaction Translator could have received the entire packet (including both CRC bytes) but not received the packet EOP. In the next micro-frame, the Transaction Translator will respond with an MDATA and send all of the data bytes (with the two CRC bytes being held in the full-speed pipeline stage). This could cause the siTD to decrement its *Total Bytes to Transfer* field to zero, indicating it has received all expected data. The host must still execute one more (scheduled) complete-split transaction in order to extract the results of the full-speed transaction from the Transaction Translator (for example, the Transaction Translator may have detected a CRC failure, and this result must be forwarded to the host).

If the host experiences hold-offs that cause the host controller to skip one or more (but not all) scheduled split transactions for an isochronous OUT, then the protocol to the transaction translator will not be consistent and the transaction translator will detect and react to the problem. Likewise, for host hold-offs that cause the host controller to skip one or more (but not all) scheduled split transactions for an isochronous IN, the *C-prog-mask* is used by the host controller to detect errors. However, if the host experiences a hold-off that causes it to skip all of an siTD, or an siTD expires during a host hold off (for example, a hold-off occurs and the siTD is no longer reachable by the host controller in order for it to report the hold-off event), then system software must detect that the siTDs have not been processed by the host controller (that is, state not advanced) and report the appropriate error to the client driver.

65.4.3.12.3.3 Split Transaction Execution State Machine for Isochronous

In the following presentation, all references to micro-frame are in the context of a micro-frame within an *H-Frame.*

If the *Active* bit in the *Status* byte is a zero, the host controller will ignore the siTD and continue traversing the periodic schedule. Otherwise the host controller will process the siTD as specified below. A split transaction state machine is used to manage the split-transaction protocol sequence. The host controller uses the fields defined in Section Tracking Split Transaction Progress for Interrupt Transfers , plus the variable *cMicroFrameBit* defined in Section Split Transaction Execution State Machine for Interrupt to track the progress of an isochronous split transaction. The figure below illustrates the state machine for managing an siTD through an isochronous split transaction. Bold, dotted circles denote the state of the *Active* bit in the *Status* field of a siTD. The Bold, dotted arcs denote the transitions between these states. Solid circles denote the states of the split transaction state machine and the solid arcs denote the transitions between these states. Dotted arcs and boxes reference actions that take place either as a result of a transition or from being in a state.

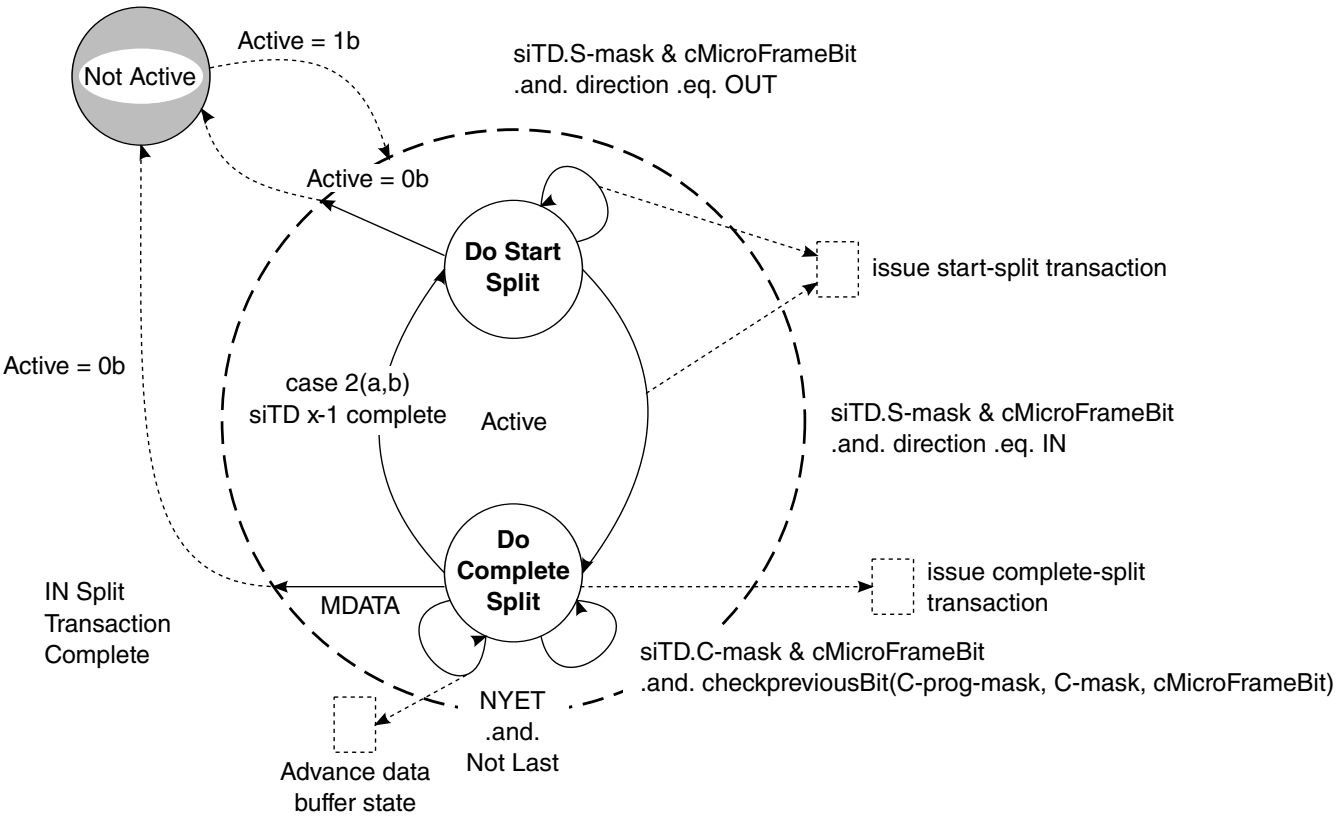


Figure 65-27. Split Transaction State Machine for Isochronous

65.4.3.12.3.4 Periodic Isochronous - Do Start Split

Isochronous split transaction OUTs use only this state.

An siTD for a split-transaction isochronous IN is either initialized to this state, or the siTD transitions to this state from Do Complete Split when a case 2a (IN) or 2b scheduling boundary isochronous split-transaction completes.

Each time the host controller reaches an active siTD in this state, it checks the *siTD.S-mask* against *cMicroFrameBit*. If there is a one in the appropriate position, the siTD will execute a start-split transaction. By definition, the host controller cannot *reach* an siTD at the wrong time. If the *I/O* field indicates an IN, then the start-split transaction includes only the extended token plus the full-speed token. Software must initialize the *siTD.Total Bytes To Transfer* field to the number of bytes expected. This is usually the maximum packet size for the full-speed endpoint. The host controller exits this state when the start-split transaction is complete.

The remainder of this section is specific to an isochronous OUT endpoint (that is, the *I/O* field indicates an OUT). When the host controller executes a start-split transaction for an isochronous OUT it includes a data payload in the start-split transaction. The memory

buffer address for the data payload is constructed by concatenating *siTD.Current Offset* with the page pointer indicated by the page selector field (*siTD.P*). A zero in this field selects Page 0 and a 1 selects Page 1. During the start-split for an OUT, if the data transfer crosses a page boundary during the transaction, the host controller must detect the page cross, update the *siTD.P*-bit from a zero to a one, and begin using the *siTD.Page 1* with *siTD.Current Offset* as the memory address pointer. The field *siTD.TP* is used to annotate each start-split transaction with the indication of which part of the split-transaction data the current payload represents (ALL, BEGIN, MID, END). In all cases the host controller simply uses the value in *siTD.TP* to mark the start-split with the correct transaction position code.

T-Count is always initialized to the number of start-splits for the current frame. *TP* is always initialized to the first required transaction position identifier. The scheduling boundary case (see Figure 65-26) is used to determine the initial value of *TP*. The initial cases are summarized in the following table.

Table 65-48. Initial Conditions for OUT siTD's TP and T-count Fields

Case	T-count	TP	Description
1, 2a	=1	ALL	When the OUT data payload is less than (or equal to) 188 bytes, only one start-split is required to move the data. The one start-split must be marked with an ALL.
1, 2a	!=1	BEGIN	When the OUT data payload is greater than 188 bytes more than one start-split must be used to move the data. The initial start-split must be marked with a BEGIN.

After each start-split transaction is complete, the host controller updates *T-Count* and *TP* appropriately so that the next start-split is correctly annotated.

The table below illustrates all of the *TP* and *T-count* transitions, which must be accomplished by the host controller.

.

Table 65-49. Transaction Position (TP)/Transaction Count (T-Count) Transition Table

TP	T-count next	TP next	Description
ALL	0	N/A	Transition from ALL, to done.
BEGIN	1	END	Transition from BEGIN to END. Occurs when *T-count* starts at 2.
BEGIN	!=1	MID	Transition from BEGIN to MID. Occurs when *T-count* starts at greater than 2.
MID	!=1	MID	*TP* stays at MID while *T-count* is not equal to 1 (that is, greater than 1). This case can occur for any of the scheduling boundary cases where the *T-count* starts greater than 3.
MID	1	END	Transition from MID to END. This case can occur for any of the scheduling boundary cases where the *T-count* starts greater than 2.

The start-split transactions do not receive a handshake from the transaction translator, so the host controller always advances the transfer state in the siTD after the bus transaction is complete. To advance the transfer state the following operations take place:

- The *siTD.Total Bytes To Transfer* and the *siTD.Current Offset* fields are adjusted to reflect the number of bytes transferred.
- The *siTD.P* (page selector) bit is updated appropriately.
- The *siTD.TP* and *siTD.T-count* fields are updated appropriately as defined in Table 65-49.

These fields are then written back to the memory based siTD. The *S-mask* is fixed for the life of the current budget. As mentioned above, *TP* and *T-count* are set specifically in each siTD to reflect the data to be sent from this siTD. Therefore, regardless of the value of *S-mask*, the actual number of start-split transactions depends on *T-count* (or equivalently, *Total Bytes to Transfer*). The host controller must set the *Active* bit to a zero when it detects that all of the schedule data has been sent to the bus. The preferred method is to detect when *T-Count* decrements to zero as a result of a start-split bus transaction. Equivalently, the host controller can detect when *Total Bytes to Transfer* decrements to zero. Either implementation must ensure that if the initial condition is *Total Bytes to Transfer* equal to zero and *T-count* is equal to a one, then the host controller will issue a single start-split, with a zero-length data payload. Software must ensure that *TP*, *T-count* and *Total Bytes to Transfer* are set to deliver the appropriate number of bus transactions from each siTD. An inconsistent combination will yield undefined behavior.

If the host experiences hold-offs that cause the host controller to skip start-split transactions for an OUT transfer, the state of the transfer will not progress appropriately. The transaction translator will observe protocol violations in the arrival of the start-splits for the OUT endpoint (that is, the transaction position annotation will be incorrect as received by the transaction translator).

Example scenarios are described in Section Split Transaction for Isochronous - Processing Examples.

A host controller implementation can optionally track the progress of an OUT split transaction by setting appropriate bits in the *siTD.C-prog-mask* as it executes each scheduled start-split. The *checkPreviousBit*() algorithm defined in Periodic Isochronous - Do Complete Split can be used prior to executing each start-split to determine whether start-splits were skipped. The host controller can use this mechanism to detect missed micro-frames. It can then set the siTD's *Active* bit to zero and stop execution of this siTD. This saves on both memory and high-speed bus bandwidth.

65.4.3.12.3.5 Periodic Isochronous - Do Complete Split

This state is only used by a split-transaction isochronous IN endpoint.

This state is entered unconditionally from the Do Start State after a start-split transaction is executed for an IN endpoint. Each time the host controller visits an siTD in this state, it conducts a number of tests to determine whether it should execute a complete-split transaction. The individual tests are listed below. The sequence they are applied depends on which micro-frame the host controller is currently executing which means that the tests might not be applied until after the siTD referenced from the back pointer has been fetched.

- Test A. *cMicroFrameBit* is bit-wise anded with *siTD.C-mask* field. A non-zero result indicates that software scheduled a complete-split for this endpoint, during this micro-frame. This test is always applied to a newly fetched siTD that is in this state.
- Test B. The *siTD.C-prog-mask* bit vector is checked to determine whether the previous complete splits have been executed. An example algorithm is below (this is slightly different than the algorithm used in Section Periodic Isochronous - Do Complete Split). The sequence in which this test is applied depends on the current value of USB_n_FRINDEX[2:0]. If USB_n_FRINDEX[2:0] is 0 or 1, it is not applied until the back pointer has been used. Otherwise it is applied immediately.

```
Algorithm Boolean CheckPreviousBit(siTD.C-prog-mask, siTD.C-mask, cMicroFrameBit)
Begin
          Boolean rvalue = TRUE;
          previousBit = cMicroFrameBit rotate-right(1)
          -- Bit-wise anding previousBit with C-mask indicates whether there was an intent
          -- to send a complete split in the previous micro-frame. So, if the
          -- 'previous bit' is set in C-mask, check C-prog-mask to make sure it
          -- happened.
          if previousBit bitAND siTD.C-mask then
                    if not (previousBit bitAND siTD.C-prog-mask) then
                              rvalue = FALSE
                    End if
          End if
          Return rvalue
End Algorithm
```

If Test A is true and USB_n_FRINDEX[2:0] is zero or one, then this is a case 2a or 2b scheduling boundary (see Figure 65-25). See Section Periodic Isochronous - Do Complete Split for details in handling this condition.

If Test A and Test B evaluate to true, then the host controller will execute a complete-split transaction using the transfer state of the current siTD. When the host controller commits to executing the complete-split transaction, it updates *QH.C-prog-mask* by bit-ORing with *cMicroFrameBit*. The transfer state is advanced based on the completion status of the complete-split transaction. To advance the transfer state of an IN siTD, the host controller must:

- Decrement the number of bytes received from *siTD.Total Bytes To Transfer,*
- Adjust *siTD.Current Offset* by the number of bytes received,

- Adjust *siTD.P* (page selector) field if the transfer caused the host controller to use the next page pointer, and
- Set any appropriate bits in the *siTD.Status* field, depending on the results of the transaction.

Note that if the host controller encounters a condition where *siTD.Total Bytes To Transfer* is zero, and it receives more data, the host controller must not write the additional data to memory. The *siTD.Status.Active* bit must be set to zero and the *siTD.Status.Babble Detected* bit must be set to a one. The fields *siTD.Total Bytes To Transfer*, *siTD.Current Offset*, and *siTD.P* (page selector) are not required to be updated as a result of this transaction attempt.

The host controller must accept (assuming good data packet CRC and sufficient room in the buffer as indicated by the value of *siTD.Total Bytes To Transfer*) MDATA and DATA0/1 data payloads up to and including 192 bytes. A host controller implementation may optionally set *siTD.Status Active* to a zero and *siTD.Status.Babble Detected* to a one when it receives and MDATA or DATA0/1 with a data payload of more than 192 bytes. The following responses have the noted effects:

- ERR. The full-speed transaction completed with a time-out or bad CRC and this is a reflection of that error to the host. The host controller sets the *ERR* bit in the *siTD.Status* field and sets the *Active* bit to a zero.
- Transaction Error (XactErr). The complete-split transaction encounters a Timeout, CRC16 failure, etc. The *siTD.Status* field *XactErr* field is set to a one and the complete-split transaction must be retried immediately. The host controller must use an internal error counter to count the number of retries as a counter field is not provided in the siTD data structure. The host controller will not retry more than two times. If the host controller exhausts the retries or the end of the micro-frame occurs, the *Active* bit is set to zero.
- DATAx (0 or 1). This response signals that the final data for the split transaction has arrived. The transfer state of the siTD is advanced and the *Active* bit is set to a zero. If the *Bytes To Transfer* field has not decremented to zero (including the reception of the data payload in the DATAx response), then less data than was expected, or allowed for was actually received. This *short packet* event does not set the USBINT status bit in the USBSTS register to a one. The host controller will not detect this condition.
- NYET (and Last). On each NYET response, the host controller also checks to determine whether this is the last complete-split for this split transaction. Last was defined in Section Periodic Isochronous - Do Complete Split . If it is the last complete-split (with a NYET response), then the transfer state of the siTD is not advanced (never received any data) and the *Active* bit is set to a zero. No bits are set in the *Status* field because this is essentially a skipped transaction. The transaction translator must have responded to all the scheduled clompete-splits with NYETs,

meaning that the start-split issued by the host controller was not received. This result should be interpreted by system software as if the transaction was completely skipped. The test for whether this is the last complete split can be performed by XORing C-mask with C-prog-mask. A zero result indicates that all complete-splits have been executed.

- MDATA (and Last). See above description for testing for Last. This can only occur when there is an error condition. Either there has been a babble condition on the full-speed link, which delayed the completion of the full-speed transaction, or software set up the *S-mask* and/or *C-mask*s incorrectly. The host controller must set *XactErr* bit to a one and the *Active* bit is set to a zero.
- NYET (and not Last). See above description for testing for Last. The complete-split transaction received a NYET response from the transaction translator. Do not update any transfer state (except for *C-prog-mask*) and stay in this state.
- MDATA (and not Last). The transaction translator responds with an MDATA when it has partial data for the split transaction. For example, the full-speed transaction data payload spans from micro-frame X to X+1 and during micro-frame X, the transaction translator will respond with an MDATA and the data accumulated up to the end of micro-frame X. The host controller advances the transfer state to reflect the number of bytes received.

If Test A succeeds, but Test B fails, it means that one or more of the complete-splits have been skipped. The host controller sets the *Missed Micro-Frame* status bit and sets the *Active* bit to a zero.

65.4.3.12.3.6 Complete-Split for Scheduling Boundary Cases 2a, 2b

Boundary cases 2a and 2b (INs only) (see Figure 65-25) require that the host controller use the transaction state context of the previous siTD to finish the split transaction. The table below enumerates the transaction state fields.

Table 65-50. Summary siTD Split Transaction State

Buffer State	Status	Execution Progress
Total Bytes To Transfer P (page select) Current Offset TP (transaction position) T-count (transaction count)	All bits in the status field	C-prog-mask

NOTE

TP and *T-count* are used only for Host to Device (OUT) endpoints.

If software has budgeted the schedule of this data stream with a frame wrap case, then it must initialize the *siTD.Back Pointer* field to reference a valid siTD and will have the *siTD.Back Pointer.T-bit* in the *siTD.Back Pointer*

field set to a zero. Otherwise, software must set the *siTD.Back Pointer.T-bit* in the *siTD.Back Pointer* field to a one. The host controller's rules for interpreting when to use the *siTD.Back Pointer* field are listed below. These rules apply only when the siTD's *Active* bit is a one and the *SplitXState* is Do Complete Split.

- When cMicroFrameBit is a 1h and the siTDX.Back Pointer.T-bit is a zero, or
- If cMicroFrameBit is a 2h and siTDX.S-mask[0] is a zero

When either of these conditions apply, then the host controller must use the transaction state from $siTD_{X-1}$.

In order to access $siTD_{X-1}$, the host controller reads on-chip the siTD referenced from $siTD_X$.*Back Pointer*.

The host controller must save the entire state from $siTD_X$ while processing $siTD_{X-1}$. This is to accommodate for case 2b processing. The host controller must not recursively walk the list of *siTD.Back Pointers*.

If $siTD_{X-1}$ is active (*Active* bit is a one and *SplitXStat* is Do Complete Split), then both Test A and Test B are applied as described above. If these criteria to execute a complete-split are met, the host controller executes the complete split and evaluates the results as described above. The transaction state (see Table 65-50) of $siTD_{X-1}$ is appropriately advanced based on the results and written back to memory. If the resultant state of $siTD_{X-1}$'s *Active* bit is a one, then the host controller returns to the context of $siTD_X$, and follows its next pointer to the next schedule item. No updates to $siTD_X$ are necessary.

If $siTD_{X-1}$ is active (*Active* bit is a one and *SplitXStat* is Do Start Split), then the host controller must set *Active* bit to a zero and *Missed Micro-Frame* status bit to a one and the resultant status written back to memory.

If $siTD_{X-1}$*'sActive* bit is a zero, (because it was zero when the host controller first visited $siTD_{X-1}$ via $siTD_X$*'s* back pointer, it transitioned to zero as a result of a detected error, or the results of $siTD_{X-1}$*'s* complete-split transaction transitioned it to zero), then the host controller returns to the context of $siTD_X$ and transitions its *SplitXState* to Do Start Split. The host controller then determines whether the case 2b start split boundary condition exists (that is, if *cMicroframeBit* is a 1b and $siTD_X$.*S-mask*[0] is a 1b). If this criterion is met the host controller immediately executes a start-split transaction and appropriately advances the transaction state of $siTD_X$, then follows $siTD_X$.*Next Pointer* to the next schedule item. If the criterion is not met, the host controller simply follows $siTD_X$.*Next Pointer* to the next schedule item. Note that in the case of a 2b boundary case, the split-transaction of $siTD_{X-1}$ will have its *Active* bit set to zero when the host controller returns

to the context of $siTD_X$. Also, note that software should not initialize an siTD with *C-mask* bits 0 and 1 set to a one and an *S-mask* with bit zero set to a one. This scheduling combination is not supported and the behavior of the host controller is undefined.

65.4.3.12.3.7 Split Transaction for Isochronous - Processing Examples

There is an important difference between how the hardware/software manages the isochronous split transaction state machine and how it manages the asynchronous and interrupt split transaction state machines.

The asynchronous and interrupt split transaction state machines are encapsulated within a single queue head. The progress of the data stream depends on the progress of each split transaction. In some respects, the split-transaction state machine is sequenced via the Execute Transaction queue head traversal state machine (see Figure 65-18).

Isochronous is a pure time-oriented transaction/data stream. The interface data structures are optimized to efficiently describe transactions that need to occur at specific times. The isochronous split-transaction state machine must be managed across these time-oriented data structures. This means that system software must correctly describe the scheduling of split-transactions across more than one data structure.

Then the host controller must make the appropriate state transitions at the appropriate times, in the correct data structures.

For example, the table below illustrates a couple of frames worth of scheduling required to schedule a case 2a full-speed isochronous data stream.

Table 65-51. Example Case 2a - Software Scheduling siTDs for an IN Endpoint

siTDX		Micro-Frames								Initial
#	Masks	0	1	2	3	4	5	6	7	SplitXState
X	S-Mask	-	-	-	-	1	-	-	-	Do Start Split
	C-Mask	1	1	-	-	-	-	1	1	
X+1	S-Mask	-	-	-	-	1	-	-	-	Do Complete Split
	C-Mask	1	1					1	1	
X+2	S-Mask	-	-	-	-	1	-	-	-	Do Complete Split
	C-Mask	1	1					1	1	
X+3	S-Mask	Repeats previous pattern								Do Complete Split
	C-Mask									

This example shows the first three siTDs for the transaction stream. Because this is the case-2a frame-wrap case, *S-masks* of all siTDs for this endpoint have a value of 10h (a one bit in micro-frame 4) and *C-mask* value of C3h (one-bits in micro-frames 0,1, 6 and 7). Additionally, sofware ensures that the *Back Pointer* field of each siTD references the appropriate siTD data structure (and the *Back PointerT-bits* are set to zero).

The initial *SplitXState* of the first siTD is Do Start Split. The host controller will visit the first siTD eight times during frame X. The C-mask bits in micro-frames 0 and 1 are ignored because the state is Do Start Split. During micro-frame 4, the host controller determines that it can run a start-split (and does) and changes *SplitXState* to Do Complete Split. During micro-frames 6 and 7, the host controller executes complete-splits. Notice the siTD for frame X+1 has its *SplitXState* initialized to Do Complete Split. As the host controller continues to traverse the schedule during *H-Frame* X+1, it will visit the second siTD eight times. During micro-frames 0 and 1 it will detect that it must execute complete-splits.

During *H-Frame* X+1, micro-frame 0, the host controller detects that $siTD_{X+1}$'s *Back Pointer.T-bit* is a zero, saves the state of $siTD_{X+1}$ and fetches $siTD_X$. It executes the complete split transaction using the transaction state of $siTD_X$. If the $siTD_X$ split transaction is complete, siTD's *Active* bit is set to zero and results written back to $siTD_X$. The host controller retains the fact that $siTD_X$ is retired and transitions the *SplitXState* in the $siTD_{X+1}$ to Do Start Split. At this point, the host controller is prepared to execute the start-split for $siTD_{X+1}$ when it reaches micro-frame 4. If the split-transaction completes early (transaction-complete is defined in Section Periodic Isochronous - Do Complete Split), that is, before all the scheduled complete-splits have been executed, the host controller will transition $siTD_X$.*SplitXState* to Do Start Split early and naturally skip the remaining scheduled complete-split transactions. For this example, $siTD_{X+1}$ does not receive a DATA0 response until *H-Frame* X+2, micro-frame 1.

During *H-Frame X+2,* micro-frame 0, the host controller detects that $siTD_{X+2}$'s *Back Pointer.T-bit* is a zero, saves the state of $siTD_{X+2}$ and fetches $siTD_{X+1}$. As described above, it executes another split transaction, receives an MDATA response, updates the transfer state, but does not modify the *Active* bit. The host controller returns to the context of $siTD_{X+2}$, and traverses its next pointer without any state change updates to $siTD_{X+2}$. S

During *H-Frame X+2*, micro-frame 1, the host controller detects $siTD_{X+2}$'s *S-mask[0]* is a zero, saves the state of $siTD_{X+2}$ and fetches $siTD_{X+1}$. It executes another complete-split transaction, receives a DATA0 response, updates the transfer state and sets the *Active* bit to a zero. It returns to the state of $siTD_{X+2}$ and changes its *SplitXState* to Do Start Split. At this point, the host controller is prepared to execute start-splits for $siTD_{X+2}$ when it reaches micro-frame 4.

65.4.3.13 Host Controller Pause

When the host controller's *HCHalted* bit in the USBSTS register is a zero, the host controller is sending SOF (Start OF Frame) packets down all enabled ports.

When the schedules are enabled, the EHCI host controller will access the schedules in main memory each micro-frame. This constant pinging of main memory is known to create Arm platform power management problems for mobile systems. Specifically, mobile systems aggressively manage the state of the Arm platform, based on recent history usage. In the more aggressive power saving modes, the Arm platform can disable its caches. Current PC architectures assume that bus-master accesses to main memory must be cache-coherent. So, when bus masters are busy touching memory, the Arm platform power management software can detect this activity over time and inhibit the transition of the Arm platform into its lowest power savings mode. USB controllers are bus-masters and the frequency at which they access their memory-based schedules keeps the Arm platform power management software from placing the Arm platform into its lowest power savings state.

USB Host controllers don't access main memory when they are suspended. However, there are a variety of reasons why placing the USB controllers into suspend won't work, but they are beyond the scope of this document. The base requirement is that the USB controller needs to be kept out of main memory, while at the same time, the USB bus is kept from going into suspend.

EHCI controllers provide a large-grained mechanism that can be manipulated by system software to change the memory access pattern of the host controller. System software can manipulate the schedule enable bits in the USBCMD register to turn on/off the scheduling traversal. A software heuristic can be applied to implement an on/off duty cycle that allows the USB to make reasonable progress and allow the Arm platform power management to get the Arm platform into its lowest power state. This method is not intended to be applied at all times to throttle USB, but should only be applied in very specific configurations and usage loads. For example, when only a keyboard or mouse is attached to the USB, the heuristic could detect times when the USB is attempting to move data only very infrequently and can adjust the duty cycle to allow the Arm platform to reach its low power state for longer periods of time. Similarly, it could detect increases in the USB load and adjust the duty cycle appropriately, even to the point where the schedules are never disabled. The assumption here is that the USB is moving data and the Arm platform will be required to process the data streams.

It is suggested that in order to provide a complete solution for the system, the companion host controllers should also provide a similar method to allow system software to inhibit the companion host controller from accessing its shared memory based data structures (schedule lists or otherwise).

65.4.3.14 Port Test Modes -Host Operational Model

EHCI host controllers must implement the port test modes Test J_State, Test K_State, Test_Packet, Test Force_Enable, and Test SE0_NAK as described in the USB Specification Revision 2.0.

The system is only allowed to test ports that are owned by the EHCI controller (for example, *CF-bit* is a one and *PortOwner* bit is a zero). System software is allowed to have at most one port in test mode at a time. Placing more than one port in test mode will yield undefined results. The required, per port test sequence is (assuming the *CF-bit* in the USB_n_CONFIGFLAG register is a one):

- Disable the periodic and asynchronous schedules by setting the *Asynchronous Schedule Enable* and *Periodic Schedule Enable* bits in the USBCMD register to a zero.
- Place all enabled root ports into the suspended state by setting the *Suspend* bit in each appropriate USB_n_PORTSC register to a one.
- Set the *Run/Stop* bit in the USBCMD register to a zero and wait for the *HCHalted* bit in the USBSTS register, to transition to a one. Note that an EHCI host controller implementation may optionally allow port testing with the *Run/Stop* bit set to a one. However, all host controllers must support port testing with *Run/Stop* set to a zero and *HCHalted* set to a one.
- Set the *Port Test Control* field in the port under test PORTSC register to the value corresponding to the desired test mode. If the selected test is Test_Force_Enable, then the *Run/Stop* bit in the USBCMD register must then be transitioned back to one, in order to enable transmission of SOFs out of the port under test.
- When the test is complete, system software must ensure the host controller is halted (*HCHalted* bit is a one) then it terminates and exits test mode by setting *HCReset* to a one.

65.4.3.15 Interrupts-Host Operational Model

The EHCI Host Controller hardware provides interrupt capability based on a number of sources.

There are several general groups of interrupt sources:

- Interrupts as a result of executing transactions from the schedule (success and error conditions),
- Host controller events (Port change events, etc.), and
- Host Controller error events

All transaction-based sources are maskable through the Host Controller's Interrupt Enable register (USBINTR, see Section Interrupt Enable Register (USB_*n*USBINTR)). Additionally, individual transfer descriptors can be marked to generate an interrupt on completion. This section describes each interrupt source and the processing that occurs in response to the interrupt.

During normal operation, interrupts may be immediate or deferred until the next interrupt threshold occurs. The interrupt threshold is a tunable parameter via the *Interrupt Threshold Control* field in the USBCMD register. The value of this register controls when the host controller will generate an interrupt on behalf of normal transaction execution. When a transaction completes during an interrupt interval period, the interrupt signaling the completion of the transfer will not occur until the interrupt threshold occurs. For example, the default value is eight micro-frames. This means that the host controller will not generate interrupts any more frequently than once every eight micro-frames.

Section Host System Error details effects of a host system error.

If an interrupt has been scheduled to be generated for the current interrupt threshold interval, the interrupt is not signaled until after the status for the last complete transaction in the interval has been written back to host memory. This may sometimes result in the interrupt not being signaled until the next interrupt threshold.

Initial interrupt processing is the same, regardless of the reason for the interrupt. When an interrupt is signaled by the hardware, Arm platform control is transferred to host controller's USB interrupt handler. The precise mechanism to accomplish the transfer is OS specific. For this discussion it is just assumed that control is received. When the interrupt handler receives control, its first action is to reads the USBSTS (USB Status Register). It then acknowledges the interrupt by clearing all of the interrupt status bits by writing ones to these bit positions. The handler then determines whether the interrupt is due to schedule processing or some other event. After acknowledging the interrupt, the handler (via an OS-specific mechanism), schedules a deferred procedure call (DPC) which will execute later. The DPC routine processes the results of the schedule execution. The precise mechanisms used are beyond the scope of this document.

Note: the host controller is not required to de-assert a currently active interrupt condition when software sets the interrupt enables (in the USBINR register, see Section Interrupt Enable Register (USB_*n*USBINTR)) to a zero. The only reliable method software should use for acknowledging an interrupt is by transitioning the appropriate status bits in the USBSTS register (Section USB Status Register (USB_*n*USBSTS)) from a one to a zero.

65.4.3.15.1 Transfer/Transaction Based Interrupts

These interrupt sources are associated with transfer and transaction progress. They are all dependent on the next interrupt threshold.

65.4.3.15.1.1 Transaction Error

A transaction error is any error that caused the host controller to think that the transfer did not complete successfully.

The table below lists the events/responses that the host can observe as a result of a transaction. The effects of the error counter and interrupt status are summarized in the following paragraphs. Most of these errors set the *XactErr* status bit in the appropriate interface data structure.

There is a small set of protocol errors that relate only when executing a queue head and fit under the umbrella of a WRONG PID error that are significant to explicitly identify. When these errors occur, the *XactErr* status bit in the queue head is set and the *CErr* field is decremented. When the *PIDCode* indicates a SETUP, the following responses are protocol errors and result in *XactErr* bit being set to a one and the *CErr* field being decremented.

- *EPS* field indicates a high-speed device and it returns a Nak handshake to a SETUP.
- *EPS* field indicates a high-speed device and it returns a Nyet handshake to a SETUP.
- *EPS* field indicates a low- or full-speed device and the complete-split receives a Nak handshake.

Table 65-52. Summary of Transaction Errors

Event / Result	Queue Head/qTD/iTD/siTD Side-effects		USB Status Register (USBSTS)
	Cerr	Status Field	USBERRINT
CRC	-1	XactErr set to a one.	1[1]
Timeout	-1	XactErr set to a one.	1[1]
Bad PID[2]	-1	XactErr set to a one.	1[1]
Babble	N/A	Section Serial Bus Babble	1
Buffer Error	N/A	Section Data Buffer Error	

1. If occurs in a queue head, then *USBERRINT* is asserted only when *CErr* counts down from a one to a zero. In addition the queue is halted, see Halting a Queue Head.
2. The host controller received a response from the device, but it could not recognize the PID as a valid PID.

65.4.3.15.1.2 Serial Bus Babble

When a device transmits more data on the USB than the host controller is expecting for this transaction, it is defined to be babbling. In general, this is called a *Packet Babble*.

When a device sends more data than the *Maximum Length* number of bytes, the host controller sets the *Babble Detected* bit to a one and halts the endpoint if it is using a queue head (see Halting a Queue Head). *Maximum Length* is defined as the minimum of *Total Bytes to Transfer* and *Maximum Packet Size*. The *CErr* field is not decremented for a packet babble condition (only applies to queue heads). A babble condition also exists if IN transaction is in progress at High-speed EOF2 point. This is called a frame babble. A frame babble condition is recorded into the appropriate schedule data structure. In addition, the host controller must disable the port to which the frame babble is detected.

The *USBERRINT* bit in the USB_n_USBSTS register is set to a one and if the *USB Error Interrupt Enable* bit in the USB_n_USBINTR register is a one, then a hardware interrupt is signaled to the system at the next interrupt threshold. The host controller must never start an OUT transaction that will babble across a micro-frame EOF.

NOTE

When a host controller detects a data PID mismatch, it must either: disable the packet babble checking for the duration of the bus transaction or do packet babble checking based solely on *Maximum Packet Size*. The USB core specification defines the requirements on a data receiver when it receives a data PID mismatch (for example, expects a DATA0 and gets a DATA1 or visa-versa). In summary, it must ignore the received data and respond with an ACK handshake, in order to advance the transmitter's data sequence.

The EHCI interface allows System software to provide buffers for a Control, Bulk or Interrupt IN endpoint that are not an even multiple of the maximum packet size specified by the device. Whenever a device misses an ACK for an IN endpoint, the host and device are out of synchronization with respect to the progress of the data transfer. The host controller may have advanced the transfer to a buffer that is less than maximum packet size. The device will re-send its maximum packet size data packet, with the original data PID, in response to the next IN token. In order to properly manage the bus protocol, the host controller must disable the packet babble check when it observes the data PID mismatch.

65.4.3.15.1.3 Data Buffer Error

This event indicates that an overrun of incoming data or a underrun of outgoing data has occurred for this transaction.

This would generally be caused by the host controller not being able to access required data buffers in memory within necessary latency requirements. These conditions are not considered transaction errors, and do not effect the error count in the queue head. When these errors do occur, the host controller records the fact the error occurred by setting the *Data Buffer Error* bit in the queue head, iTD or siTD.

If the data buffer error occurs on a non-isochronous IN, the host controller will not issue a handshake to the endpoint. This will force the endpoint to resend the same data (and data toggle) in response to the next IN to the endpoint.

If the data buffer error occurs on an OUT, the host controller must corrupt the end of the packet so that it cannot be interpreted by the device as a good data packet. Simply truncating the packet is not considered acceptable. An acceptable implementation option is to 1's complement the CRC bytes and send them. There are other options suggested in the Transaction Translator section of the USB Specification Revision 2.0.

65.4.3.15.1.4 USB Interrupt (Interrupt on Completion (IOC))

Transfer Descriptors (iTDs, siTDs, and queue heads (qTDs)) contain a bit that can be set to cause an interrupt on their completion. The completion of the transfer associated with that schedule item causes the USB Interrupt (USBINT) bit in the USB_n_USBSTS register to be set to a one.

In addition, if a short packet is encountered on an IN transaction associated with a queue head, then this event also causes USBINT to be set to a one. If the USB Interrupt Enable bit in the USB_n_USBINTR register is set to a one, a hardware interrupt is signaled to the system at the next interrupt threshold. If the completion is because of errors, the *USBERRINT* bit in the USB_n_USBSTS register is also set to a one.

65.4.3.15.1.5 Short Packet

Reception of a data packet that is less than the endpoint's Max Packet size during Control, Bulk or Interrupt transfers signals the completion of the transfer. Whenever a short packet completion occurs during a queue head execution, the *USBINT* bit in the USB_n_USBSTS register is set to a one.

If the *USB Interrupt Enable* bit is set in the USB_n_USBINTR register, a hardware interrupt is signaled to the system at the next interrupt threshold.

65.4.3.15.2 Host Controller Event Interrupts

These interrupt sources are independent of the interrupt threshold (with the one exception being the Interrupt on Async Advance, see Section Interrupt on Async Advance).

65.4.3.15.2.1 Port Change Events

Port registers contain status and status change bits. When the status change bits are set to a one, the host controller sets the *Port Change Detect* bit in the USBSTS register to a one.

If the *Port Change Interrupt Enable* bit in the USB_n_USBINTR register is a one, then the host controller will issue a hardware interrupt. The port status change bits include:

- Connect Status Change
- Port Enable/Disable Change
- Over-current Change
- Force Port Resume

65.4.3.15.2.2 Frame List Rollover

This event indicates that the host controller has wrapped the frame list. The current programmed size of the frame list effects how often this interrupt occurs.

If the frame list size is 1024, then the interrupt will occur every 1024 milliseconds, if it is 512, then it will occur every 512 milliseconds, etc. When a frame list rollover is detected, the host controller sets the *Frame List Rollover* bit in the USB.USBSTS register to a one. If the *Frame List Rollover Enable* bit in the USB.USBINTR register is set to a one, the host controller issues a hardware interrupt. This interrupt is not delayed to the next interrupt threshold.

65.4.3.15.2.3 Interrupt on Async Advance

This event is used for deterministic removal of queue heads from the asynchronous schedule. Whenever the host controller advances the on-chip context of the asynchronous schedule, it evaluates the value of the *Interrupt on Async Advance Doorbell* bit in the USB.USBCMD register.

If it is a one, it sets the *Interrupt on Async Advance* bit in the USB.USBSTS register to a one. If the *Interrupton Async Advance Enable* bit in the USB.USBINTR register is a one, the host controller issues a hardware interrupt at the next interrupt threshold. A detailed explanation of this feature is described in Section Removing Queue Heads from Asynchronous Schedule .

65.4.3.15.2.4 Host System Error

The host controller is a bus master and any interaction between the host controller and the system may experience errors.

The type of host error may be catastrophic to the host controller (such as a Master Abort) making it impossible for the host controller to continue in a coherent fashion.In the presence of non-catastrophic host errors, such as parity errors, the host controller could potentially continue operation. The recommended behavior for these types of errors is to escalate it to a catastrophic error and halt the host controller. Host-based error must result in the following actions:

- The *Run/Stop* bit in the USB.USBCMD register is set to a zero.
- The following bits in the USB.USBSTS register are set:
  - *Host System Error* bit is to a one.
  - *HCHalted* bit is set to a one.
- If the *Host System Error Enable* bit in the USB.USBINTR register is a one, then the host controller will issue a hardware interrupt. This interrupt is not delayed to the next interrupt threshold. The following table summarizes the required actions taken on the various host errors.

Table 65-53. Summary Behavior of EHCI Host Controller on Host System Errors

Cycle Type	Master Abort	Target Abort	Data Phase Parity
Frame list pointer fetch (read)	Fatal	Fatal	Fatal [o]
siTD fetch (read)	Fatal	Fatal	Fatal [o]
siTD status write-back (write)	Fatal [o]	Fatal [o]	Fatal [o]
iTD fetch (read)	Fatal	Fatal	Fatal [o]
iTD status write-back (write)	Fatal [o]	Fatal [o]	Fatal [o]
qTD fetch (read)	Fatal	Fatal	Fatal [o]
qHD status write-back (write)	Fatal [o]	Fatal [o]	Fatal [o]
Data write	Fatal [o]	Fatal [o]	Fatal [o]
Data read	Fatal	Fatal	Fatal [o]

Potentially, a host controller implementation could continue operation without a halt. However, the recommended behavior is to halt the host controller.

NOTE

After a *Host System Error*, Software must reset the host controller through *HCReset* in the USB.USBCMD register before re-initializing and restarting the host controller.

65.4.4 EHCI Deviation

For the purposes a dual-role Host/Device controller with support for On-The-Go applications, it is necessary to deviate from the EHCI specification. Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 0.95, November 2000, Intel Corporation. http://www.intel.com. Device operation & On-The-Go operation is not specified in the EHCI and thus the implementation supported in this core is proprietary.

The host mode operation of the core is near EHCI compatible with few minor differences documented in this section.

The particulars of the deviations occur in the areas summarized here:

- Embedded Transaction Translator - Allows direct attachment of FS and LS devices in host mode without the need for a companion controller.
- Device operation - In host mode the device operational registers are generally disabled and thus device mode is mostly transparent when in host mode. However, there are a couple exceptions documented in the following sections.
- Embedded design interface - This core does not have a PCI Interface and therefore the PCI configuration registers described in the EHCI specification are not applicable.
- On-The-Go Operation - This design includes an On-The-Go controller for Port #1.

65.4.4.1 Embedded Transaction Translator Function

The OTG controller supports directly connected full and low speed devices without requiring a companion controller by including the capabilities of a USB 2.0 high speed hub transaction translator.

Although there is no separate Transaction Translator block in the system, the transaction translator function normally associated with a high speed hub has been implemented within the DMA and Protocol engine blocks. The embedded transaction translator function is an extension to EHCI interface, but makes use of the standard data structures and operational models that exist in the EHCI specification to support full and low speed devices.

65.4.4.1.1 Capability Registers

The following additions have been added to the capability registers to support the embedded Transaction Translator Function:

- N_TT added to USB.HCSPARAMS - Host Control Structural Parameters
- N_PTT added to USB.HCSPARAMS - Host Control Structural Parameters

65.4.4.1.2 Operational Registers

The following additions have been added to the operational registers to support the embedded TT:

- Addition of two-bit Port Speed (PSPD) to the Port Status & Control (USB_*n*PORTSC1) register.

65.4.4.1.3 Discovery-EHCI Deviation

In a standard EHCI controller design, the EHCI host controller driver detects a Full speed (FS) or Low speed (LS) device by noting if the port enable bit is set after the port reset operation.

The port enable will only be set in a standard EHCI controller implementation after the port reset operation and when the host and device negotiate a High-Speed connection (that is, Chirp completes successfully).

Because this controller has an embedded Transaction Translator, the port enable will always be set after the port reset operation regardless of the result of the host device chirp result and the resulting port speed will be indicated by the PSPD field in USB.PORTSCx.

Therefore, the standard EHCI host controller driver requires an alteration to handle directly connected Full and Low speed devices or hubs.

The change is a fundamental one in that is summarized in the following table.

Table 65-54. Summary of EHCI

Standard EHCI	EHCI with embedded Transaction Translator
After port enable bit is set following a connection and reset sequence, the device/hub is assumed to be HS.	After port enable bit is set following a connection and reset sequence, the device/hub speed is noted from USB.PORTSCx.
FS and LS devices are assumed to be downstream from a HS hub thus, all port-level control is performed through the Hub Class to the nearest Hub.	FS and LS device can be either downstream from a HS hub or directly attached. When the FS/LS device is downstream from a HS hub, then port-level control is done using the Hub Class through the nearest Hub. When a FS/LS device is directly attached, then port-level control is accomplished using USB.PORTSCx.
FS and LS devices are assumed to be downstream from a HS hub with HubAddr=X. [where HubAddr > 0 and HubAddr is the address of the Hub where the bus transitions from HS to FS/LS (ie. Split target hub)]	FS and LS device can be either downstream from a HS hub with HubAddr = X [HubAddr > 0] or directly attached [where HubAddr = 0 and HubAddr is the address of the Root Hub where the bus transitions from HS to FS/LS (ie. Split target hub is the root hub)]

65.4.4.1.4 Data Structures

The same data structures used for FS/LS transactions though a HS hub are also used for transactions through the Root Hub with sm embedded Transaction Translator.

Here it is demonstrated how the Hub Address and Endpoint Speed fields should be set for directly attached FS/LS devices and hubs:

1. QH (for direct attach FS/LS) - Async. (Bulk/Control Endpoints) Periodic (Interrupt)
   - Hub Address = 0
   - Transactions to direct attached device/hub.
     - QH.EPS = Port Speed
   - Transactions to a device downstream from direct attached FS hub.
     - QH.EPS = Downstream Device Speed

NOTE

When QH.EPS = 01 (LS) and PORTSCx.PSPD = 00 (FS), a LS-pre-pid will be sent before the transmitting LS traffic.

Maximum Packet Size must be less than or equal 64 or undefined behaviour may result.

2. siTD (for direct attach FS) - Periodic (ISO Endpoint)
   - All FS ISO transactions:
     - Hub Address = 0
     - siTD.EPS = 00 (full speed)
       - Maximum Packet Size must less than or equal to 1023 or undefined behaviour may result.

65.4.4.1.5 Operational Model

The operational models are well defined for the behavior of the Transaction Translator (see USB 2.0 specification. Universal Serial Bus Specification, Revision 2.0, April 2000, Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips. http://www.usb.org) and for the EHCI controller moving packets between system memory and a USB-HS hub. Because the embedded Transaction Translator exists within the host controller there is no physical bus between EHCI host controller driver and the USB FS/LS bus. These sections will briefly discuss the operational model for how the EHCI and Transaction Translator operational models are combined without the physical bus between. The following sections assume the reader is familiar with both the EHCI and USB 2.0 Transaction Translator operational models.

65.4.4.1.5.1 Micro- frame Pipeline

The EHCI operational model uses the concept of H-frames and B-frames to describe the pipeline between the Host (H) and the Bus (B). The embedded Transaction Translator shall use the same pipeline algorithms specified in the USB 2.0 specification for a Hub-based Transaction Translator.

All periodic transfers always begin at B-frame 0 (after SOF) and continue until the stored periodic transfers are complete. As an example of the micro-frame pipeline implemented in the embedded Transaction Translator, all periodic transfers that are tagged in EHCI to execute in H-frame 0 will be ready to execute on the bus in B-frame 0.

It is important to note that when programming the S-mask and C-masks in the EHCI data structures to schedule periodic transfers for the embedded Transaction Translator, the EHCI host controller driver must follow the same rules specified in EHCI for programming the S-mask and C-mask for downstream Hub-based Transaction Translators.

Once periodic transfers are exhausted, any stored asynchronous transfer will be moved. Asynchronous transfers are opportunistic in that they shall execute whenever possible and their operation is not tied to H-frame and B-frame boundaries with the exception that an asynchronous transfer can not babble through the SOF (start of B-frame 0.)

65.4.4.1.5.2 Split State Machines

The start and complete split operational model differs from EHCI slightly because there is no bus medium between the EHCI controller and the embedded Transaction Translator.

Where a start or complete-split operation would occur by requesting the split to the HS hub, the start/complete split operation is simple an internal operation to the embedded Transaction Translator. The following table summarizes the conditions where handshakes are emulated from internal state instead of actual handshakes to HS split bus traffic.

Table 65-55. Summary of the Conditons of Handshakes[1]

Condition	Emulate TT Response
Start-Split: All asynchronous buffers full.	NAK
Start-Split: All periodic buffers full.	ERR
Start-Split: Success for start of Async. Transaction.	ACK
Start-Split: Start Periodic Transaction.	No Handshake (Ok)
Complete-Split: Failed to find transaction in queue.	Bus Time Out
Complete-Split: Transaction in Queue is Busy.	NYET
Complete-Split: Transaction in Queue is Complete.	[Actual Handshake from LS/FS device]

1. The un-shaded cells represent Start-Splits and the shaded cells represent Complete-Splits

65.4.4.1.5.3 Asynchronous Transaction Scheduling and Buffer Management

The following USB 2.0 specification items are implemented in the embedded Transaction Translator:

- Sequencing is provided & a packet length estimator ensures no full-speed/low-speed packet babbles into SOF time.
- Transaction tracking for 2 data pipes.

65.4.4.1.5.3.1 USB 2.0 - 11.17.3

- Sequencing is provided & a packet length estimator ensures no full-speed/low-speed packet babbles into SOF time.

65.4.4.1.5.3.2 USB 2.0 - 11.17.4

- Transaction tracking for 2 data pipes.

65.4.4.1.5.4 Periodic Transaction Scheduling and Buffer Management

The following USB 2.0 specification items are implemented in the embedded Transaction Translator:

- Abort of pending start-splits
  - EOF (and not started in micro-frames 6)
  - Idle for more than 4 micro-frames
- Abort of pending complete-splits
  - EOF
  - Idle for more than 4 micro-frames
- Transaction tracking for up to 16 data pipes.
- Complete-split transaction searching.

NOTE

There is no data schedule mechanism for these transactions other than the micro-frame pipeline. The embedded TT assumes the number of packets scheduled in a frame does not exceed the frame duration (1 ms) or else undefined behavior may result.

65.4.4.1.5.4.1 USB 2.0 - 11.18.6.[1-2]

- Abort of pending start-splits
  - EOF (and not started in micro-frames 6)
  - Idle for more than 4 micro-frames
- Abort of pending complete-splits

- EOF
- Idle for more than 4 micro-frames

65.4.4.1.5.4.2 USB 2.0 - 11.18.[7-8]

- Transaction tracking for up to 16 data pipes.
- Complete-split transaction searching.

NOTE

There is no data schedule mechanism for these transactions other than the micro-frame pipeline. The embedded TT assumes the number of packets scheduled in a frame does not exceed the frame duration (1 ms) or else undefined behavior may result.

65.4.4.1.5.5 Multiple Transaction Translators

The maximum number of embedded Transaction Translators that is currently supported is one as indicated by the N_TT field in the Host Controller Structural Parameters (USB_*n*HCSPARAMS) register.

65.4.4.2 Device Operation

The co-existence of a device operational controller within the host controller has little effect on EHCI compatibility for host operation except as noted in this section.

65.4.4.2.1 USB_USBMODE Register

Given that the dual-role controller is initialized in neither host nor device mode, the USB Device Mode (USB_*n*USBMODE) register must be programmed for host operation before the EHCI host controller driver can begin EHCI host operations.

65.4.4.2.2 Non-Zero Fields the Register File

Some of the reserved fields and reserved addresses in the capability registers and operational register have use in device mode, the following must be adhered to:

- Write operations to all EHCI reserved fields (some of which are device fields) with the operation registers should always be written to zero. This is an EHCI requirement of the device controller driver that must be adhered to.
- Read operations by the host controller must properly mask EHCI reserved fields (some of which are device fields) because fields that are used exclusive for device are undefined in host mode .

65.4.4.2.3 SOF Interrupt

This SOF Interrupt used for device mode is shared as a free running 125us interrupt for host mode.

EHCI does not specify this interrupt but it has been added for convenience and as a potential software time base. See USB Status Register (USB_*n*USBSTS) and Interrupt Enable Register (USB_*n*USBINTR) registers.

65.4.4.3 Embedded Design Interface

This is an Embedded USB Host Controller as defined by the EHCI specification and thus does not implement the PCI configuration registers.

65.4.4.3.1 Frame Adjust Register

Given that the optional PCI configuration registers are not included in this implementation, there is no corresponding bit level timing adjustments like is provided by the Frame Adjust register in the PCI configuration registers. Starts of micro-frames are timed precisely to 125 us using the transceiver clock as a reference clock. That is, a 60 Mhz transceiver clock for 8-bit physical interfaces & full-speed serial interfaces or 30 Mhz transceiver clock for 16-bit physical interfaces.

65.4.4.4 Miscellaneous variations from EHCI

65.4.4.4.1 Programmable Physical Interface Behaviour

This design supports multiple Physical interfaces which can operate in differing modes when the core is configured with software programmable Physical Interface Modes. Software programmability allows the selection of the Physical interface part during the board design phase instead of during the chip design phase. The control bits for selecting the Physical Interface operating mode have been added to the Port Status & Control (USB_*n*PORTSC1) register providing a capability that is not defined by EHCI.

65.4.4.4.2 Discovery

65.4.4.4.2.1 Port Reset

The port connect methods specified by EHCI require setting the port reset bit in the Port Status & Control (USB_*n*PORTSC1) register for a duration of 10ms. Due to the complexity required to support the attachment of devices that are not high speed there are counter already present in the design that can count the 10ms reset pulse to alleviate the requirement of the software to measure this duration. Therefore, the basic connection is then summarized as the following:

- [Port Change Interrupt] Port connect change occurs to notify the host controller driver that a device has attached.
- Software shall write a '1' to reset the device.
- Software shall write a '0' to reset the device after 10 ms.
  - This step, which is necessary in a standard EHCI design, may be omitted with this implementation. Should the EHCI host controller driver attempt to write a '0' to the reset bit while a reset is in progress, the write will simple be ignored and the reset will continue until completion.
- [Port Change Interrupt] Port enable change occurs to notify the host controller that the device in now operational and at this point the port speed has been determined.

65.4.4.4.2.2 Port Speed Detection

After the port change interrupt indicates that a port is enabled, the EHCI stack should determine the port speed. Unlike the EHCI implementation, which will re-assign the port owner for any device that does not connect at High-Speed, this host controller supports direct attach of non High-Speed devices.

Therefore, the following differences are important regarding port speed detection:

- Port Owner is read-only and always reads 0.
- A 2-bit Port Speed indicator has been added to PORTSC to provide the current operating speed of the port to the host controller driver.
- A 1-bit High Speed indicator has been added to PORTSC to signify that the port is in High-Speed vs. Full/Low Speed - *This information is redundant with the 2-bit Port Speed indicator above.*

65.4.4.4.3 Port Test Mode

Port Test Control mode behaves fully as described in EHCI. An alternate host controller driver procedure is not necessary or supported.

65.4.5 Device Data Structures

This section defines the interface data structures used to communicate control, status, and data between Device Controller Driver (DCD) Software and the Device Controller.

The data structure definitions in this chapter support a 32-bit memory buffer address space. The interface consists of device Queue Heads and Transfer Descriptors.

NOTE

Software must ensure that no interface data structure reachable by the Device Controller spans a 4K-page boundary.

The data structures defined in the chapter are (from the device controller's perspective) a mix of read-only and read/ writable fields. The device controller must preserve the read-only fields on all data structure writes.

The figure below shows the organization of the EndPoint Queue Head.

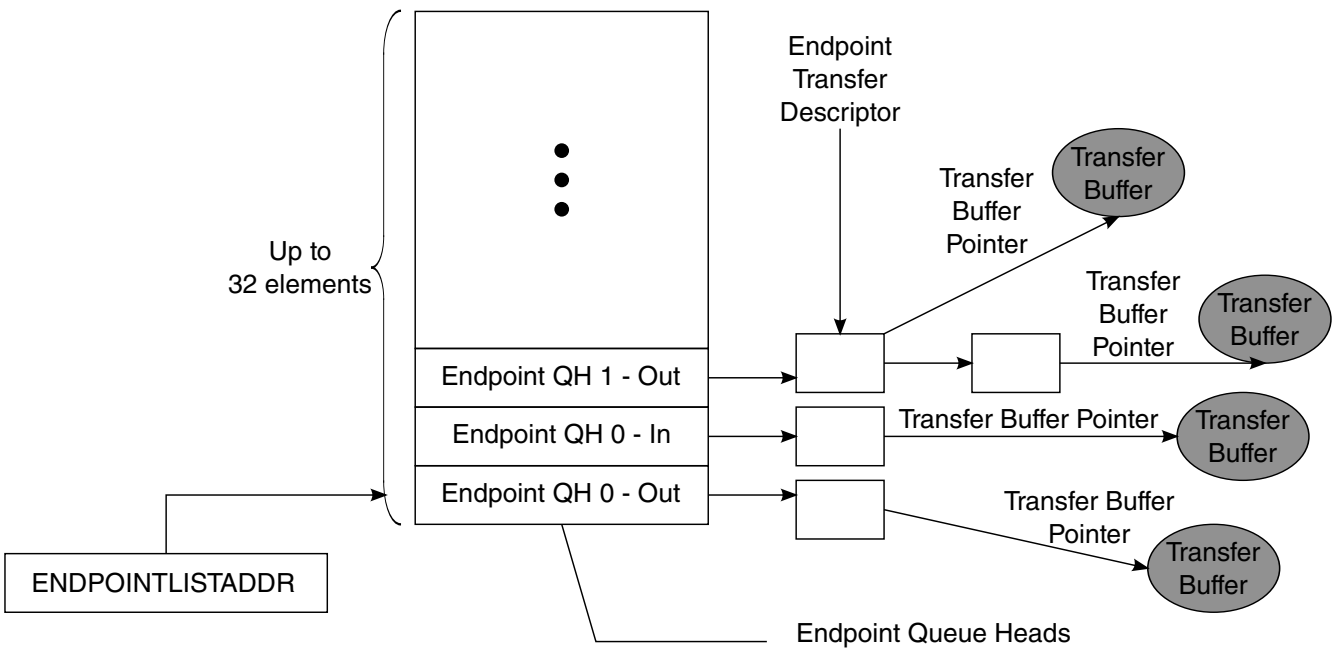


Figure 65-28. EndPoint Queue Head Organization

Endpoint queue heads are arranged in an array in a continuous area of memory pointed to by the USB.ENDPOINTLISTADDR pointer. The even -numbered device queue heads in the list support receive endpoints (OUT/SETUP) and the odd-numbered queue heads in the list are used for transmit endpoints (IN/INTERRUPT). The device controller will index into this array based upon the endpoint number received from the USB bus. All information necessary to respond to transactions for all primed transfers is contained in this list so the Device Controller can readily respond to incoming requests without having to traverse a linked list.

NOTE

The Endpoint Queue Head List must be aligned to a 2k boundary.

65.4.5.1 Endpoint Queue Head (dQH)

The device Endpoint Queue Head (dQH) is where all transfers for a given endpoint are managed. The dQH is a 48-byte data structure, but must be aligned on 64-byte boundaries.

During priming of an endpoint, the dTD (device transfer descriptor) is copied into the overlay area of the dQH, which starts at the nextTD pointer DWord and continues through the end of the buffer pointers DWords. After a transfer is complete, the dTD status DWord is updated in the dTD pointed to by the currentTD pointer. While a packet is in progress, the overlay area of the dQH is used as a staging area for the dTD so that the Device Controller can access needed information with little minimal latency.

Table 65-56. Endpoint Queue Head (dQH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mult		zlt	0		Maximum Packet Length											ios	0														
Current dTD Pointer																											0				
Next dTD Pointer																											0				T[1]
0	Total Bytes															ioc	0			MultO		0		Status							
Buffer Pointer (Page 0)																				Current Offset											
Buffer Pointer (Page 1)																				Reserved											
Buffer Pointer (Page 2)																				Reserved											
Buffer Pointer (Page 3)																				Reserved											
Buffer Pointer (Page 4)[1]																				Reserved											
Reserved																															
Set-up Buffer Bytes 3…0																															
Set-up Buffer Bytes 7…4																															

1. Transfer overlay starts at T and continues through Buffer Pointer (Page 4).

Host Controller Read/Write

Host Controller Read Only

65.4.5.1.1 Endpoint Capabilities/Characteristics

This DWord specifies static information about the endpoint, in other words, this information does not change over the lifetime of the endpoint. Device Controller software should not attempt to modify this information while the corresponding endpoint is enabled.

Table 65-57 describes the endpoint capabilities.

Table 65-57. Endpoint Capabilities/Characteristics

Bit	Description
31-30	Mult. This field is used to indicate the number of packets executed per transaction description as given by the following: 00 - Execute N Transactions as demonstrated by the USB variable length packet protocol where N is computed using the Maximum Packet Length (dQH) and the Total Bytes field (dTD) 01 Execute 1 Transaction. 10 Execute 2 Transactions. 11 Execute 3 Transactions. **NOTE:** Non-ISO endpoints must set Mult="00". ISO endpoints must set Mult="01", "10", or "11" as needed.
29	Zero Length Termination Select. This bit is used to indicate when a zero length packet is used to terminate transfers where to total transfer length is a multiple . This bit is not relevant for Isochronous 0 - Enable zero length packet to terminate transfers equal to a multiple of the Maximum Packet Length. (default). 1 - Disable the zero length packet on transfers that are equal in length to a multiple Maximum Packet Length.
28-27	Reserved. These bit reserved for future use and should be set to zero.
26-16	Maximum Packet Length. This directly corresponds to the maximum packet size of the associated endpoint (wMaxPacketSize). The maximum value this field may contain is 0x400 (1024).
15	Interrupt On Setup (IOS). This bit is used on control type endpoints to indicate if USBINT is set in response to a setup being received.
14-0	Reserved. Bits reserved for future use and should be set to zero.

65.4.5.1.2 Transfer Overlay-Endpoint Queue Head

The seven DWords in the overlay area represent a transaction working space for the device controller.

The general operational model is that the device controller can detect whether the overlay area contains a description of an active transfer. If it does not contain an active transfer, then it will not read the associated endpoint.

After an endpoint is readied, the dTD will be copied into this queue head overlay area by the device controller. Until a transfer is expired, software must not write the queue head overlay area or the associated transfer descriptor. When the transfer is complete, the device controller will write the results back to the original transfer descriptor and advance the queue.

See dTD for a description of the overlay fields.

65.4.5.1.3 Current dTD Pointer

The current dTD pointer is used by the device controller to locate the transfer in progress. This word is for Device Controller (hardware) use only and should not be modified by DCD software.

The following table describes the dTD Pointer.

Table 65-58. Next dTD Pointer

Bit	Description
31-5	Current dTD. This field is a pointer to the dTD that is represented in the transfer overlay area. This field will be modified by the Device Controller to next dTD pointer during endpoint priming or queue advance.
4-0	Reserved. Bit reserved for future use and should be set to zero.

65.4.5.1.4 Set-up Buffer

The set-up buffer is dedicated storage for the 8-byte data that follows a set-up PID.

NOTE

Each endpoint has a TX and an RX dQH associated with it, and only the RX queue head is used for receiving setup data packets.

The following table describes the Multiple Mode Control.

Table 65-59. Multiple Mode Control (HCCPARAMS)

DWord	Bits	Description
1	31-0	Setup Buffer 0. This buffer contains bytes 3 to 0 of an incoming setup buffer packet and is written by the device controller to be read by software.
2	31-0	Setup Buffer 1. This buffer contains bytes 7 to 4 of an incoming setup buffer packet and is written by the device controller to be read by software.

65.4.5.2 Endpoint Transfer Descriptor (dTD)

The dTD describes to the device controller the location and quantity of data to be sent/ received for a given transfer.

The DCD should not attempt to modify any field in an active dTD except the Next Like Pointer, which should only be modified as described in section Managing Transfers with Transfer Descriptors.

Table below shows the Endpoint Transfer Descriptor (dTD).

Table 65-60. Endpoint Transfer Descriptor (dTD)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Next Link Pointer																											0				T
0	Total Bytes															ioc	0			MultO		0		Status							
Buffer Pointer (Page 0)																				Current Offset											
Buffer Pointer (Page 1)																				0	Frame Number										
Buffer Pointer (Page 2)																				Reserved											
Buffer Pointer (Page 3)																				Reserved											
Buffer Pointer (Page 4)																				Reserved											

Host Controller Read/Write

Host Controller Read Only

The following table describes the dTD Pointer.

Table 65-61. Next dTD Pointer

Bit	Description
31-5	Next Transfer Element Pointer. This field contains the physical memory address of the next dTD to be processed. The field corresponds to memory address signals [31:5], respectively.
4-1	Reserved. Bits reserved for future use and should be set to zero.
0	Terminate (T). 1=pointer is invalid. 0=Pointer is valid (points to a valid Transfer Element Descriptor). This bit indicates to the Device Controller that there are no more valid entries in the queue.

The following table describes the dTD Token.

Table 65-62. dTD Token

Bit	Description
31	Reserved. Bit reserved for future use and should be set to zero.
30-16	Total Bytes. This field specifies the total number of bytes to be moved with this transfer descriptor. This field is decremented by the number of bytes actually moved during the transaction and only on the successful completion of the transaction. The maximum value software may store in the field is 5*4K (5000H). This is the maximum number of bytes 5 page pointers can access. Although it is possible to create a transfer up to 20K this assumes the 1st offset into the first page is 0. When the offset cannot be predetermined, crossing past the 5th page can be guaranteed by limiting the total bytes to 16K**. Therefore, the maximum recommended transfer is 16K (4000H). If the value of the field is zero when the host controller fetches this transfer descriptor (and the active bit is set), the device controller executes a zero-length transaction and retires the transfer descriptor.

Table continues on the next page...

Table 65-62. dTD Token (continued)

	It is not a requirement for IN transfers that Total Bytes To Transfer be an even multiple of *Maximum Packet Length*. If software builds such a transfer descriptor for an IN transfer, the last transaction will always be less that *Maximum Packet Length*.
15	Interrupt On Complete (IOC). This bit is used to indicate if USBINT is to be set in response to device controller being finished with this dTD.
14-12	Reserved. Bits reserved for future use and should be set to zero.
11-10	Multiplier Override (MultO). This field can be used for transmit ISO's (ie. ISO-IN) to override the multiplier in the QH. This field must be zero for all packet types that are not transmit-ISO. Example: if QH.multiplier = 3; Maximum packet size = 8; Total Bytes = 15; MultiO = 0 [default] Three packets are sent: {Data2(8); Data1(7); Data0(0)} if QH.multiplier = 3; Maximum packet size = 8; Total Bytes = 15; MultiO = 2 Two packets are sent: {Data1(8); Data0(7)} For maximal efficiency, software should compute MultO = greatest integer of (Total Bytes / Max. Packet Size) except for the case when Total Bytes = 0; then MultO should be 1. Note: Non-ISO and Non-TX endpoints must set MultO = "00".
9-8	Reserved. Bits reserved for future use and should be set to zero.
7-0	Status. This field is used by the Device Controller to communicate individual command execution states back to the Device Controller software. This field contains the status of the last transaction performed on this qTD. The bit encodings are: Bit Status Field Description 7 Active. 6 Halted. 5 Data Buffer Error. 3 Transaction Error. 4, 2, 0 Reserved.

The table below describes the dTD Buffer Page Pointer List.

Table 65-63. dTD Buffer Page Pointer List

Bit	Description
31-12	Buffer Pointer. Selects the page offset in memory for the packet buffer. Non virtual memory systems will typically set the buffer pointers to a series of incrementing integers.
0,11-0	Current Offset. Offset into the 4kb buffer where the packet is to begin.
1,10-0	Frame Number. Written by the device controller to indicate the frame number in which a packet finishes. This is typically be used to correlate relative completion times of packets on an ISO endpoint.

65.4.6 Device Operational Model

The function of the device operation is to transfer a request in the memory image to and from the Universal Serial Bus.

Using a set of linked list transfer descriptors, pointed to by a queue head, the device controller will perform the data transfers. The following sections explain the use of the device controller from the device controller driver (DCD) point-of-view and further describe how specific USB bus events relate to status changes in the device controller programmer's interface.

65.4.6.1 Device Controller Initialization

After hardware reset, the device is disabled until the Run/Stop bit is set to a '1'. In the disabled state, the pull-up on the USB D+ is not active which prevents an attach event from occurring. At a minimum, it is necessary to have the queue heads setup for endpoint zero before the device attach occurs.

Shortly after the device is enabled, a USB reset will occur followed by setup packet arriving at endpoint 0. A Queue head must be prepared so that the device controller can store the incoming setup packet.

In order to initialize a device, the software should perform the following steps:

- Set Controller Mode in the USB.USBMODE register to device mode.

NOTE

Transitioning from host mode to device mode requires a device controller reset before modifying USB.USBMODE.

- Allocate and Initialize device queue heads in system memory.
  - Minimum: Initialize device queue heads 0 Tx & 0 Rx.

NOTE

All device queue heads for control endpoints must be initialized before the endpoint is enabled. Non-Control device queue heads before the endpoint can be used.

  - For information on device queue heads, refer to section Device Data Structures.
- Configure USB.ENDPOINTLISTADDR Pointer.
  - For additional information on USB.ENDPOINTLISTADDR, refer to the register table.
- Enable the microprocessor interrupt associated with the USB core.
  - Recommended: enable all device interrupts including: USBINT, USBERRINT, Port Change Detect, USB Reset Received, DCSuspend.
  - For a list of available interrupts refer to the Interrupt Enable Register (USB_*n*USBINTR) and the USB Status Register (USB_*n*USBSTS) register tables.
- Set Run/Stop bit to Run Mode.

- After the Run bit is set and the device is connected to a host, a Bus Reset will be issued by host downstream port. The DCD must monitor the reset event and adjust the software state as described in the Bus Reset section of the Port State and Control section below.

NOTE

Endpoint 0 is designed as a control endpoint only and does not need to be configured using ENDPTCTRL0 register.

It is also not necessary to prime Endpoint 0 initially because the first packet received will always be a setup packet. The contents of the first setup packet will require a response in accordance with USB device framework (Chapter 9) command set.

65.4.6.2 Port State and Control

From a chip or system reset, the device controller enters the *powered* state. A transition from the *powered* state to the *attach* state occurs when the Run/Stop bit is set to a '1'.

After receiving a reset on the bus, the port will enter the *defaultFS or defaultHS* state in accordance with the reset protocol described in Appendix C.2 of the USB Specification Rev. 2.0.

The following state diagram depicts the state of a USB 2.0 device.

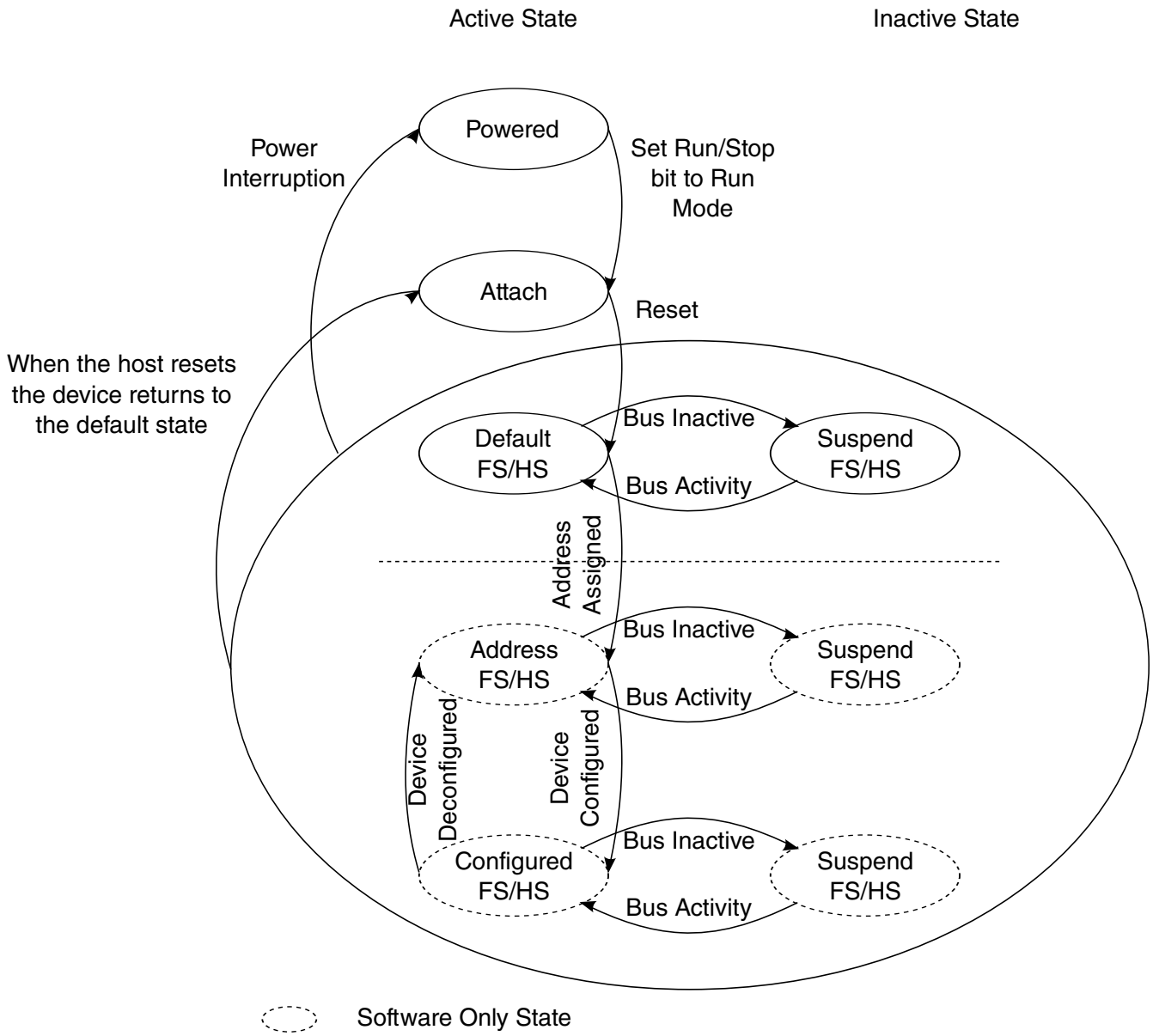


Figure 65-29. Device State Diagram

States *powered, attach, defaultFS/HS, suspendFS/HS* are implemented in the device controller and are communicated to the DCD using the following status bits:

The following table describes the Device Controller State Information Bits.

Table 65-64. Device Controller State Information Bits

Bit	Register
DCSuspend	USB Status Register (USB_*n*USBSTS)
USB Reset Received	USB Status Register (USB_*n*USBSTS)
Port Change Detect	USB Status Register (USB_*n*USBSTS)
High-Speed Port	Port Status & Control (USB_*n*PORTSC1)

It is the responsibility of the DCD to maintain a state variable to differentiate between the *DefaultFS/HS* state and the *Address/Configured* states. Change of state from *Default* to *Address* and the *Configured* states is part of the enumeration process described in the device framework section of the USB 2.0 Specification.

As a result of entering the *Address* state, the device address register (DEVICEADDR) must be programmed by the DCD.

Entry into the *Configured* indicates that all endpoints to be used in the operation of the device have been properly initialized by programming the USB_UOG_ENDPTCTRLx registers and initializing the associated queue heads.

65.4.6.2.1 Bus Reset

A bus reset is used by the host to initialize downstream devices.

When a bus reset is detected, the device controller will renegotiate its attachment speed, reset the device address to 0, and notify the DCD by interrupt (assuming the USB Reset Interrupt Enable is set). After a reset is received, all endpoints (except endpoint 0) are disabled and any primed transactions will be cancelled by the device controller. The concept of priming will be clarified below, but the DCD must perform the following tasks when a reset is received:

Clear all setup token semaphores by reading the Endpoint Status (USB_*n*ENDPTSTAT) register and writing the same value back to the Endpoint Status (USB_*n*ENDPTSTAT) register.

Clear all the endpoint complete status bits by reading the Endpoint Complete (USB_*n*ENDPTCOMPLETE) register and writing the same value back to the Endpoint Complete (USB_*n*ENDPTCOMPLETE) register.

Cancel all primed status by waiting until all bits in the Endpoint Prime (USB_*n*ENDPTPRIME) are 0 and then writing 0xFFFFFFFF to Endpoint Flush (USB_*n*ENDPTFLUSH).

Read the reset bit in the Port Status & Control (USB_*n*PORTSC1) register and make sure that it is still active. A USB reset will occur for a minimum of 3 ms and the DCD must reach this point in the reset cleanup before end of the reset occurs, otherwise a hardware reset of the device controller is recommended (rare.)

- A hardware reset can be performed by writing a one to the device controller reset bit in the USBCMD reset. Note: a hardware reset will cause the device to detach from the bus by clearing the Run/Stop bit. Thus, the DCD must completely re-initialize the device controller after a hardware reset.

Free all allocated dTDs because they will no longer be executed by the device controller. If this is the first time the DCD is processing a USB reset event, then it is likely that no dTDs have been allocated.

At this time, the DCD may release control back to the OS because no further changes to the device controller are permitted until a Port Change Detect is indicated.

After a Port Change Detect, the device has reached the default state and the DCD can read the Port Status & Control (USB_*n*PORTSC1) to determine if the device is operating in FS or HS mode. At this time, the device controller has reached normal operating mode and DCD can begin enumeration according to the USB Chapter 9 - Device Framework.

NOTE

The device DCD may use the FS/HS mode information to determine the bandwidth mode of the device

In some applications, it may not be possible to enable one or more pipes while in FS mode. *Beyond the data rate issue, there is no difference in DCD operation between FS and HS modes.*

65.4.6.2.2 Suspend/Resume

The detials of suspend and resume are explained in these sections.

65.4.6.2.2.1 Suspend

Suspend Description

In order to conserve power, USB devices automatically enter the suspended state when the device has observed no bus traffic for a specified period. When suspended, the USB device maintains any internal status, including its address and configuration. Attached devices must be prepared to suspend at any time they are powered, regardless of if they have been assigned a non-default address, are configured, or neither. Bus activity may cease due to the host entering a suspend mode of its own. In addition, a USB device shall also enter the suspended state when the hub port it is attached to is disabled.

A USB device exits suspend mode when there is bus activity. A USB device may also request the host to exit suspend mode or selective suspend by using electrical signaling to indicate remote wakeup. The ability of a device to signal remote wakeup is optional. If the USB device is capable of remote wakeup signaling, the device must support the ability of the host to enable and disable this capability. When the device is reset, remote wakeup signaling must be disabled.

Suspend Operational Model

The device controller moves into the suspend state when suspend signaling is detected or activity is missing on the upstream port for more than a specific period. After the device controller enters the suspend state, the DCD is notified by an interrupt (assuming *DC Suspend Interrupt* is enabled). When the *DCSuspend* bit in the Port Status & Control (USB_*n*PORTSC1) is set to a '1', the device controller is suspended.

DCD response when the device controller is suspended is application specific and may involve switching to low power operation.

Information on the bus power limits in suspend state can be found in USB 2.0 specification.

NOTE

Review system level clocking issues defined in section (Ref: Signals-Clocking) for the clocking requirements of a suspended device controller.

65.4.6.2.2.2 Resume

If the device controller is suspended, its operation is resumed when any non-idle signaling is received on its upstream facing port. In addition, the device can signal the system to resume operation by forcing resume signaling to the upstream port.

Resume signaling is sent upstream by writing a '1' to the Resume bit in the in the Port Status & Control (USB_*n*PORTSC1) while the device is in suspend state. Sending resume signal to an upstream port should cause the host to issue resume signaling and bring the suspended bus segment (one more devices) back to the active condition.

NOTE

Before resume signaling can be used, the host must enable it by using the Set Feature command defined in device framework (chapter 9) of the USB 2.0 Specification.

65.4.6.3 Managing Endpoints

The USB 2.0 specification defines an endpoint, also called a device endpoint or an address endpoint as a uniquely addressable portion of a USB device that can source or sink data in a communications channel between the host and the device.

The endpoint address is specified by the combination of the endpoint number and the endpoint direction.

The channel between the host and an endpoint at a specific device represents a data pipe. Endpoint 0 for a device is always a *control* type data channel used for device discovery and enumeration. Other types of endpoints support by USB include *bulk, interrupt*, and *isochronous*. Each endpoint type has specific behavior related to packet response and error handling. More detail on endpoint operation can be found in the USB 2.0 specification.

The USB OTG device controller hardware supports up to 8 endpoint numbers.

Each endpoint direction is essentially independent and can be configured with differing behavior in each direction. For example, the DCD can configure endpoint 1-IN to be a bulk endpoint and endpoint 1-OUT to be an isochronous endpoint. This helps to conserve the total number of endpoints required for device operation. The only exception is that control endpoints must use both directions on a single endpoint number to function as a control endpoint. Endpoint 0 is, for example, is always a control endpoint and uses the pair of directions.

Each endpoint direction requires a *queue head* allocated in memory. To support the 8 endpoint numbers, 16 *queue heads* are required. The operation of an endpoint and use of *queue heads* are described later in this document.

65.4.6.3.1 Endpoint Initialization

After hardware reset, all endpoints except endpoint zero are uninitialized and disabled. The DCD must configure and enable each endpoint by writing to configuration bit in the USB_UOG_ENDPTCTRLx register.

Each 32-bit USB_UOG_ENDPTCTRLx is split into an upper and lower half. The lower half of USB_UOG_ENDPTCTRLx is used to configure the receive or OUT endpoint and the upper half is likewise used to configure the corresponding transmit or IN endpoint. Control endpoints must be configured the same in both the upper and lower half of the USB_UOG_ENDPTCTRLx register otherwise the behavior is undefined. The following table shows how to construct a configuration word for endpoint initialization. The following table shows the fields and values for the Device Controller Endpoint initialization.

Table 65-65. Device Controller Endpoint Initialization

Field	Value
Data Toggle Reset	1
Data Toggle Inhibit	0
Endpoint Type	00 Control 01 Isochronous 10 Bulk

Table continues on the next page...

Table 65-65. Device Controller Endpoint Initialization (continued)

	11 Interrupt
Endpoint Stall	0

65.4.6.3.2 Stalling

There are two occasions where the device controller may need to return to the host a STALL.

The first occasion is the functional stall, which is a condition set by the DCD as described in the USB 2.0 device framework. A functional stall is only used on non-control endpoints and can be enabled in the device controller by setting the endpoint stall bit in the USB_UOG_ENDPTCTRLx register associated with the given endpoint and the given direction. In a functional stall condition, the device controller will continue to return STALL responses to all transactions occurring on the respective endpoint and direction until the endpoint stall bit is cleared by the DCD.

A protocol stall, unlike a function stall, is used on control endpoints is automatically cleared by the device controller at the start of a new control transaction (setup phase). When enabling a protocol stall, the DCD should enable the stall bits (both directions) as a pair. A single write to the USB_UOG_ENDPTCTRLx register can ensure that both stall bits are set at the same instant.

NOTE

Any write to the USB_UOG_ENDPTCTRLx register during operational mode must preserve the endpoint type field (that is, perform a read-modify-write).

The following table shows the response matrix for the Device Controller Stall.

Table 65-66. Device Controller Stall Response Matrix

USB Packet	Endpoint Stall Bit.	Effect on STALL bit.	USB Response
SETUP packet received by a non-control endpoint.	N/A	None.	STALL
IN/OUT/PING packet received by a non-control endpoint.	'1'	None.	STALL
IN/OUT/PING packet received by a non-control endpoint.	'0'	None.	ACK/ NAK/ NYET
SETUP packet received by a control endpoint.	N/A	Cleared	ACK
IN/OUT/PING packet received by a control endpoint	'1'	None	STALL

Table continues on the next page...

Table 65-66. Device Controller Stall Response Matrix (continued)

IN/OUT/PING packet received by a control endpoint.	'0'	None.	ACK/ NAK/ NYET

65.4.6.3.3 Data Toggle

Data toggle is a mechanism to maintain data coherency between host and device for any given data pipe.

For more information on data toggle, refer to the USB 2.0 specification.

65.4.6.3.3.1 Data Toggle Reset

The DCD may reset the data toggle state bit and cause the data toggle sequence to reset in the device controller by writing a '1' to the data toggle reset bit in the USB_UOG_ENDPTCTRLx register.

This should only be necessary when configuring/initializing an endpoint or returning from a STALL condition.

65.4.6.3.3.2 Data Toggle Inhibit

NOTE

This feature is for test purposes only and should never be used during normal device controller operation.

Setting the *data toggle Inhibit bit* active ('1') causes the device controller to ignore the data toggle pattern that is normally sent and accept all incoming data packets regardless of the data toggle state.

In normal operation, the device controller checks the DATA0/DATA1 bit against the data toggle to determine if the packet is valid. If Data PID does not match the data toggle state bit maintained by the device controller for that endpoint, the Data toggle is considered not valid. If the data toggle is not valid, the device controller assumes the packet was already received and discards the packet (not reporting it to the DCD). To prevent the host controller from re-sending the same packet, the device controller will respond to the error packet by acknowledging it with either an ACK or NYET response.

65.4.6.3.3.3 Priming Transmit Endpoints

Priming a transmit endpoint will cause the device controller to fetch the device transfer descriptor (dTD) for the transaction pointed to by the device queue head (dQH).

After the dTD is fetched, it will be stored in the dQH until the device controller completes the transfer described by the dTD. Storing the dTD in the dQH allows the device controller to fetch the operating context needed to handle a request from the host without the need to follow the linked list, starting at the dQH when the host request is received.

After the device has loaded the dTD, the leading data in the packet is stored in a FIFO in the device controller. This FIFO is split into virtual channels so that the leading data can be stored for any endpoint up to the maximum number of endpoints configured at device synthesis time.

After a priming request is complete, an endpoint state of primed is indicated in the USB_UOG_ENDPTSTATUS register. For a primed transmit endpoint, the device controller can respond to an IN request from the host and meet the stringent bus turnaround time of High Speed USB.

Because only the leading data is stored in the device controller FIFO, it is necessary for the device controller to begin filling in behind leading data after the transaction starts. The FIFO must be sized to account for the maximum latency that can be incurred by the system memory bus. More information about FIFO sizing is presented in section .

65.4.6.3.3.4 Priming Receive Endpoints

Priming receive endpoints is identical to priming of transmit endpoints from the point of view of the DCD. At the device controller the major difference in the operational model is that there is no data movement of the leading packet data simply because the data is to be received from the host.

Note as part of the architecture, the FIFO for the receive endpoints is not partitioned into multiple channels like the transmit FIFO. Thus, the size of the RX FIFO does not scale with the number of endpoints.

65.4.6.4 Operational Model For Packet Transfers

All transactions on the USB bus are initiated by the host and in turn, the device must respond to any request from the host within the turnaround time stated in the USB 2.0 Specification.

At USB 1.1 Full or Low Speed rates, this turnaround time was significant and the USB 1.1 device controllers were architected so that the device controller could access main memory or interrupt a host protocol processor in order to respond to the USB 1.1

transaction. The architecture of the USB 2.0 device controller must be different because same methods will not meet USB 2.0 High-speed turnaround time requirements by simply increasing clock rate.

A USB host will send requests to the device controller in an order that can not be precisely predicted as a single pipeline, so it is not possible to prepare a single packet for the device controller to execute. However, the order of packet requests is predictable when the endpoint number and direction is considered. For example, if endpoint 3 (transmit direction) is configured as a bulk pipe, then we can expect the host will send IN requests to that endpoint. This device controller is architected in such a way that it can prepare packets for each endpoint/direction in anticipation of the host request. The process of preparing the device controller to send or receive data in response to host initiated transaction on the bus is referred to as "priming" the endpoint. This term will be used throughout the following documentation to describe the device controller operation so the DCD can be architected properly use priming. Further, note that the term "flushing" is used to describe the action of clearing a packet that was queued for execution.

65.4.6.4.1 Interrupt/Bulk Endpoint Operational Model

The behaviors of the device controller for interrupt and bulk endpoints are identical.

All valid IN and OUT transactions to bulk pipes will handshake with a NAK unless the endpoint had been primed. Once the endpoint has been primed, data delivery will commence.

A dTD will be retired by the device controller when the packets described in the transfer descriptor have been completed. Each dTD describes N packets to be transferred according to the USB Variable Length transfer protocol. The formula and table on the following page describe how the device controller computes the number and length of the packets to be sent/received by the USB vary according to the total number of bytes and maximum packet length.

With Zero Length Termination (ZLT) = 0

N = INT(Number Of Bytes/Max. Packet Length) + 1

With Zero Length Termination (ZLT) = 1

N = MAXINT(Number Of Bytes/Max. Packet Length)

Table 65-67. Variable Length Transfer Protocol Example (ZLT = 0)

Bytes (dTD)	Max. Packet Length (dQH)	N	P1	P2	P3
511	256	2	256	255	

Table continues on the next page...

Table 65-67. Variable Length Transfer Protocol Example (ZLT = 0) (continued)

512	256	3	256	256	0
512	512	2	512	0	

Table 65-68. Variable Length Transfer Protocol Example (ZLT = 1)

Bytes (dTD)	Max. Packet Length (dQH)	N	P1	P2	P3
511	256	2	256	255	
512	256	2	256	256	
512	512	1	512		

NOTE

The MULT field in the dQH must be set to "00" for bulk, interrupt, and control endpoints.

TX-dTD is complete when:

- All packets described dTD were successfully transmitted. *** Total bytes in dTD will equal zero when this occurs.

RX-dTD is complete when:

- All packets described in dTD were successfully received. *** Total bytes in dTD will equal zero when this occurs.
- A short packet (number of bytes < maximum packet length) was received. *** This is a successful transfer completion; DCD must check Total Bytes in dTD to determine the number of bytes that are remaining. From the total bytes remaining in the dTD, the DCD can compute the actual bytes received.
- A long packet was received (number of bytes > maximum packet size) OR (total bytes received > total bytes specified). *** This is an error condition. The device controller will discard the remaining packet, and set the Buffer Error bit in the dTD. In addition, the endpoint will be flushed and the USBERR interrupt will become active.

On the successful completion of the packet(s) described by the dTD, the active bit in the dTD will be cleared and the next pointer will be followed when the Terminate bit is clear. When the Terminate bit is set, the device controller will flush the endpoint/direction and cease operations for that endpoint/direction.

On the unsuccessful completion of a packet (see long packet above), the dQH will be left pointing to the dTD that was in error. In order to recover from this error condition, the DCD must properly reinitialize the dQH by clearing the active bit and update the nextTD pointer before attempting to re-prime the endpoint.

NOTE

All packet level errors such as a missing handshake or CRC error will be retried automatically by the device controller.

There is no required interaction with the DCD for handling such errors.

65.4.6.4.1.1 Interrupt/Bulk Endpoint Bus Response Matrix

The table below shows the response matrix for Interrput/Bulk Endpoint Bus.

Table 65-69. Interrupt/Bulk Endpoint Bus Response Matrix

	Stall	Not Primed	Primed	Underflow	Overflow
Setup	Ignore	Ignore	Ignore	N/A	N/A
In	STALL	NAK	Transmit	BS Error	N/A
Out	STALL	NAK	Receive + NYET/ACK	N/A	NAK
Ping	STALL	NAK	ACK	N/A	N/A
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore

NOTE

BS Error = Force Bit Stuff Error

NYET/ACK - NYET unless the Transfer Descriptor has packets remaining according to the USB variable length protocol then ACK.

SYSERR - System error should never occur when the latency FIFOs are correctly sized and the DCD is responsive.

65.4.6.4.2 Control Endpoint Operation Model

This section details the setup phase, data phase, status phase, and the control endpoint bus response matrix.

65.4.6.4.2.1 Setup Phase

All requests to a control endpoint begin with a setup phase followed by an optional data phase and a required status phase. The device controller will always accept the setup phase unless the setup lockout is engaged.

The setup lockout will engage so that future setup packets are ignored. Lockout of setup packets ensures that while software is reading the setup packet stored in the queue head, that data is not written as it is being read potentially causing an invalid setup packet.

The setup lockout mechanism can be disabled and a new tripwire type semaphore will ensure that the setup packet payload is extracted from the queue head without being corrupted be an incoming setup packet. This is the preferred behavior because ignoring repeated setup packets due to long software interrupt latency would be a compliance issue.

- Disable Setup Lockout by writing 1 to Setup Lockout Mode (SLOM) in USB Device Mode (USB_*n*USBMODE). (once at initialization). Setup lockout is not necessary when using the tripwire as described below.

NOTE

Leaving the Setup Lockout Mode As 0 will result in pre-2.3 hardware behavior.

- After receiving an interrupt and inspecting Endpoint Setup Status (USB_*n*ENDPTSETUPSTAT) to determine that a setup packet was received on a particular pipe:
  a. Write 1 to clear corresponding bit Endpoint Setup Status (USB_*n*ENDPTSETUPSTAT).
  b. Write 1 to Setup Tripwire (SUTW) in USB Command Register (USB_*n*USBCMD) register.
  c. Duplicate contents of dQH.SetupBuffer into local software byte array.
  d. Read Setup TripWire (SUTW) in USB Command Register (USB_*n*USBCMD) register. (if set - continue; if cleared - goto 2)
  e. Write 0 to clear Setup Tripwire (SUTW) in USB Command Register (USB_*n*USBCMD) register.
  f. Process setup packet using local software byte array copy and execute status/ handshake phases.

NOTE

After receiving a new setup packet the status and/or handshake phases may still be pending from a previous control sequence. These should be flushed & deallocated before linking a new status and/or handshake dTD for the most recent setup packet.

65.4.6.4.2.2 Data Phase

Following the setup phase, the DCD must create a device transfer descriptor for the data phase and prime the transfer.

After priming the packet, the DCD must verify a new setup packet has not been received by reading the USB.ENDPTSETUPSTAT register immediately verifying that the prime had completed. A prime will complete when the associated bit in the Endpoint Prime (USB_*n*ENDPTPRIME) register is zero and the associated bit in the Endpoint Status

(USB_*n*ENDPTSTAT) register is a one. If a prime fails, ie. The Endpoint Prime (USB_*n*ENDPTPRIME) bit goes to zero and the Endpoint Status (USB_*n*ENDPTSTAT) bit is not set, then the prime has failed. This can only be due to improper setup of the dQH, dTD or a setup arriving during the prime operation. If a new setup packet is indicated after the ENDPTPRIME bit is cleared, then the transfer descriptor can be freed and the DCD must reinterpret the setup packet.

Should a setup arrive after the data stage is primed, the device controller will automatically clear the prime status (Endpoint Status (USB_*n*ENDPTSTAT)) to enforce data coherency with the setup packet.

NOTE

- The MULT field in the dQH must be set to "00" for bulk, interrupt, and control endpoints.
- Error handling of data phase packets is the same as bulk packets described previously.

65.4.6.4.2.3 Status Phase

Similar to the data phase, the DCD must create a transfer descriptor (with byte length equal zero) and prime the endpoint for the status phase.

The DCD must also perform the same checks of the USB.ENDPTSETUPSTAT as described above in the data phase.

NOTE

- The MULT field in the dQH must be set to 00 for bulk, interrupt, and control endpoints.
- Error handling of data phase packets is the same as bulk packets described previously.

65.4.6.4.2.4 Control Endpoint Bus Response Matrix

Shown in the following table is the device controller response to packets on a control endpoint according to the device controller state.

The table below shows the response matrix for the Control Endpoint Bus.

Table 65-70. Control Endpoint Bus Response Matrix

Token Type	Endpoint State					Setup Lockout
	Stall	Not Primed	Primed	Underflow	Overflow	
Setup	ACK	ACK	ACK	N/A	SYSERR	
In	STALL	NAK	Transmit	BS Error	N/A	N/A

Table continues on the next page...

Table 65-70. Control Endpoint Bus Response Matrix (continued)

Out	STALL	NAK	Receive + NYET/ACK	N/A	NAK	N/A
Ping	STALL	NAK	ACK	N/A	N/A	N/A
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore	Ignore

BS Error = Force Bit Stuff Error

NYET/ACK - NYET unless the Transfer Descriptor has packets remaining according to the USB variable length protocol then ACK.

SYSERR - System error should never occur when the latency FIFOs are correctly sized and the DCD is responsive.

65.4.6.4.3 Isochronous Endpoint Operational Model

Isochronous endpoints are used for real-time scheduled delivery of data and their operational model is significantly different than the host throttled Bulk, Interrupt, and Control data pipes.

Real time delivery by the device controller will be accomplished by the following:

- Exactly MULT Packets per (micro) Frame are transmitted/received. Note: MULT is a two-bit field in the device Queue Head. The variable length packet protocol is not used on isochronous endpoints.
- NAK responses are not used. Instead, zero length packets are sent in response to an IN request to an unprimed endpoints. For unprimed RX endpoints, the response to an OUT transaction is to ignore the packet within the device controller.
- Prime requests always schedule the transfer described in the dTD for the next (micro) frame. If the ISO-dTD is still active after that frame, then the ISO-dTD will be held ready until executed or canceled by the DCD.

An EHCI compatible host controller uses the periodic frame list to schedule data exchanges to Isochronous endpoints. The operational model for device mode does not use such a data structure. Instead, the same dTD used for Control/Bulk/Interrupt endpoints is also used for isochronous endpoints. The difference is in the handling of the dTD.

The first difference between bulk and ISO-endpoints is that priming an ISO-endpoint is a delayed operation such that an endpoint will become primed only after a SOF is received. After the DCD writes the prime bit, the prime bit will be cleared as usual to indicate to software that the device controller completed a priming the dTD for transfer. Internal to the design, the device controller hardware masks that prime start until the next frame boundary. This behavior is hidden from the DCD but occurs so that the device controller can match the dTD to a specific (micro)frame.

Another difference with isochronous endpoints is that the transaction must wholly complete in a (micro)frame. Once an ISO transaction is started in a (micro)frame it will retire the corresponding dTD when MULT transactions occur or the device controller finds a fulfillment condition.

The transaction error bit set in the status field indicates a fulfillment error condition. When a fulfillment error occurs, the frame after the transfer failed to complete wholly, the device controller will force retire the ISO-dTD and move to the next ISO-dTD.

It is important to note that fulfillment errors are only caused due to partially completed packets. If no activity occurs to a primed ISO-dTD, the transaction will stay primed indefinitely. This means it is up to software discard transmit ISO-dTDs that pile up from a failure of the host to move the data.

Finally, the last difference with ISO packets is in the data level error handling. When a CRC error occurs on a received packet, the packet is not retried similar to bulk and control endpoints. Instead, the CRC is noted by setting the *Transaction Error* bit and the data is stored as usual for the application software to sort out.

- TX Packet Retired
  - MULT counter reaches zero.
  - Fulfillment Error [*Transaction Error* bit is set]
    - # Packets Occurred > 0 AND # Packets Occurred < MULT

NOTE

For TX-ISO, MULT Counter can be loaded with a lesser value in the dTD Multiplier Override field in hardware versions 2.3 and later. If the Multiplier Override is zero, the MULT Counter is initialized to the Multiplier in the QH.

- RX Packet Retired:
  - MULT counter reaches zero.
  - Non-MDATA Data PID is received**
    - ** Exit criteria only valid in hardware version 2.3 or later. Previous to hardware version 2.3, any PID sequence that did not match the MULT field exactly would be flagged as a transaction error due to PID mismatch or fulfillment error.
  - Overflow Error:
    - Packet received is > maximum packet length. [*Buffer Error* bit is set]
    - Packet received exceeds total bytes allocated in dTD. [*Buffer Error* bit is set]
  - Fulfillment Error [*Transaction Error* bit is set]
    - # Packets Occurred > 0 AND # Packets Occurred < MULT
  - CRC Error [*Transaction Error* bit is set]

NOTE

For ISO, when a dTD is retired, the next dTD is primed for the next frame. For continuous (micro)frame to (micro)frame operation the DCD should ensure that the dTD linked-list is out ahead of the device controller by at least two (micro)frames.

65.4.6.4.3.1 Isochronous Pipe Synchronization

When it is necessary to synchronize an isochronous data pipe to the host, the (micro) frame number (USB_UOG_FRINDEX register) can be used as a marker.

To cause a packet transfer to occur at a specific (micro) frame number [N], the DCD should interrupt on SOF during frame N-1. When the USB_UOG_FRINDEX=N-1, the DCD must write the prime bit. The device controller will prime the isochronous endpoint in (micro) frame N-1 so that the device controller will execute delivery during (micro) frame N.

NOTE

Priming an endpoint towards the end of (micro) frame N-1 will not guarantee delivery in (micro) frame N. The delivery may actually occur in (micro) frame N+1 if device controller does not have enough time to complete the prime before the SOF for packet N is received.

65.4.6.4.3.2 Isochronous Endpoint Bus Response Matrix

The following table shows the response matrix for the Isochronous Endpoint Bus.

Table 65-71. Isochronous Endpoint Bus Response Matrix

	Stall	Not Primed	Primed	Underflow	Overflow
Setup	STALL	STALL	STALL	N/A	N/A
In	NULL Packet	NULL Packet	Transmit	BS Error	N/A
Out	Ignore	Ignore	Receive	N/A	Drop Packet
Ping	Ignore	Ignore	Ignore	Ignore	Ignore
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore

1. BS Error = Force Bit Stuff Error

   NULL Packet = Zero Length Packet

65.4.6.5 Managing Queue Heads

The following figure shows the End Point Queue Head.

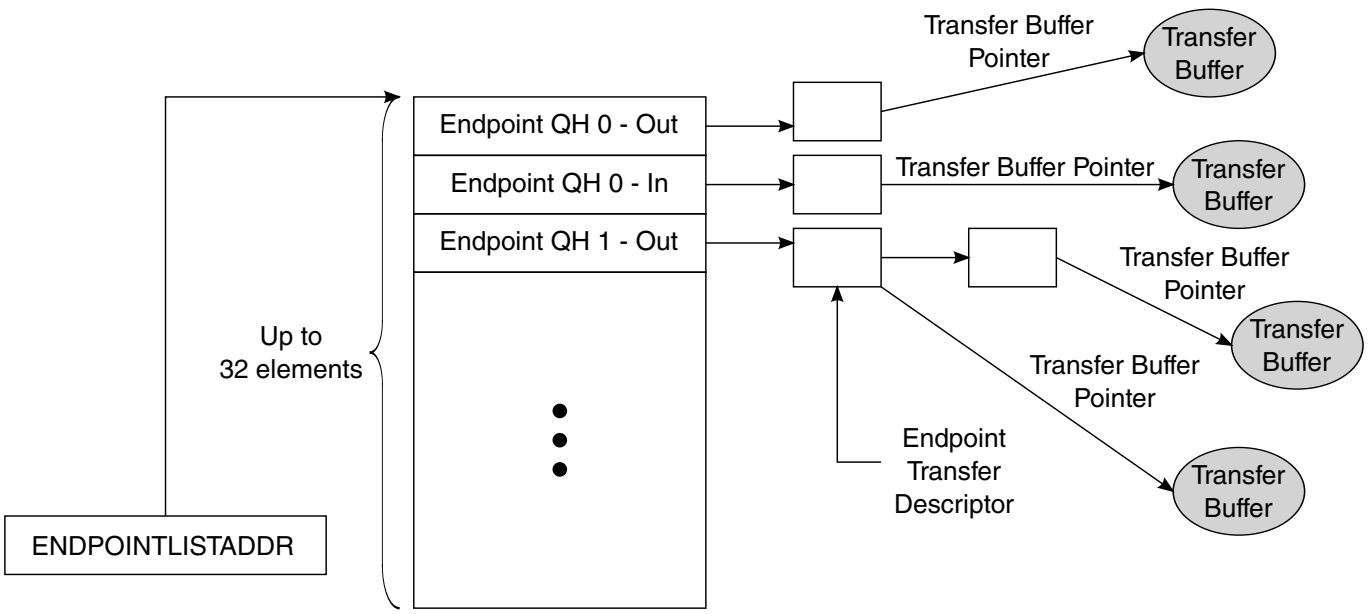


Figure 65-30. End Point Queue Head Diagram

The device queue head (dQH) points to the linked list of transfer tasks, each depicted by the device Transfer Descriptor (dTD). An area of memory pointed to by USB.ENDPOINTLISTADDR contains a group of all dQH's in a sequential list as shown in Figure 65-30. The even elements in the list of dQH's are used for receive endpoints (OUT/SETUP) and the odd elements are used for transmit endpoints (IN/INTERRUPT). Device transfer descriptors are linked head to tail starting at the queue head and ending at a terminate bit. Once the dTD has been retired, it will no longer be part of the linked list from the queue head. Therefore, software is required to track all transfer descriptors because pointers will no longer exist within the queue head once the dTD is retired (see section Software Link Pointers).

In addition to the current and next pointers and the dTD overlay examined in section Operational Model For Packet Transfers, the dQH also contains the following parameters for the associated endpoint: Multipler, Maximum Packet Length, Interrupt On Setup. The complete initialization of the dQH including these fields is demonstrated in the next section.

65.4.6.5.1 Queue Head Initialization

One device queue head must be initialized for each active endpoint.

To initialize a device queue head:

- Write the wMaxPacketSize field as required by the USB Chapter 9 or application specific protocol.
- Write the multiplier field to 0 for control, bulk, and interrupt endpoints. For ISO endpoints, set the multiplier to 1,2, or 3 as required bandwidth and in conjunction with the USB Chapter 9 protocol.

NOTE

In FS mode, the multiplier field can only be 1 for ISO endpoints.

- Write the next dTD Terminate bit field to 1.
- Write the Active bit in the status field to 0.
- Write the Halt bit in the status field to 0.

NOTE

The DCD must only modify dQH if the associated endpoint is not primed and there are no outstanding dTD's.

65.4.6.5.2 Operational Model For Setup Transfers

As discussed in section Control Endpoint Operation Model, setup transfer requires special treatment by the DCD. A setup transfer does not use a dTD but instead stores the incoming data from a setup packet in an 8-byte buffer within the dQH.

Upon receiving notification of the setup packet, the DCD should handle the setup transfer as demonstrated here:

1. Copy setup buffer contents from dQH - RX to software buffer.
2. Acknowledge setup backup by writing a "1" to the corresponding bit in ENDPTSETUPSTAT.

NOTE

- The acknowledge must occur before continuing to process the setup packet.
- After the acknowledge has occurred, the DCD must not attempt to access the setup buffer in the dQH - RX. Only the local software copy should be examined.

3. Check for pending data or status dTD's from previous control transfers and flush if any exist as discussed in section Flushing/De-priming an Endpoint.
4. Decode setup packet and prepare data phase [optional] and status phase transfer as required by the USB Chapter 9 or application specific protocol.

NOTE

It is possible for the device controller to receive setup packets before previous control transfers complete. Existing control packets in progress must be flushed and the new control packet completed.

65.4.6.6 Managing Transfers with Transfer Descriptors

65.4.6.6.1 Software Link Pointers

It is necessary for the DCD software to maintain head and tail pointers to the for the linked list of dTDs for each respective queue head.

This is necessary because the dQH only maintains pointers to the current working dTD and the next dTD to be executed. The operations described in next section for managing dTD will assume the DCD can use reference the head and tail of the dTD linked list. The following figure shows the Software Link Pointers.

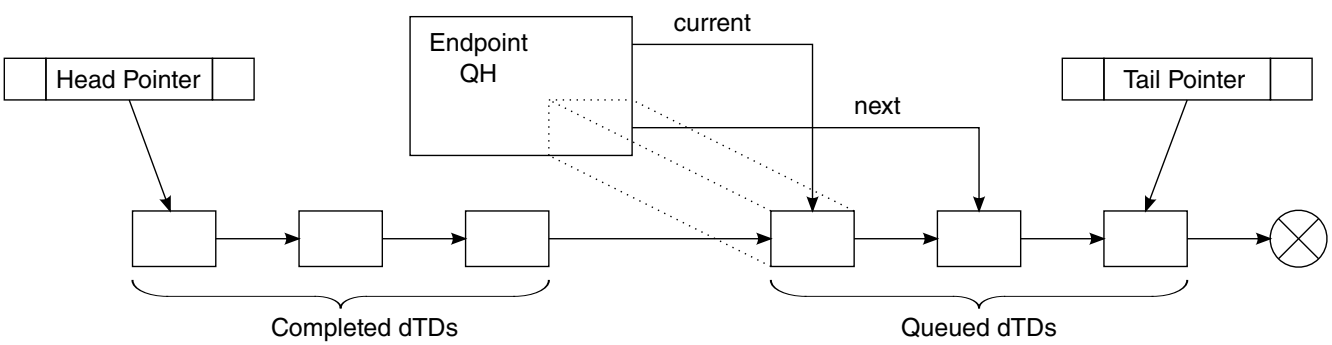


Figure 65-31. Software Link Pointers

NOTE

To conserve memory, the reserved fields at the end of the dQH can be used to store the Head & Tail pointers, but it still remains the responsibility of the DCD to maintain the pointers.

65.4.6.6.2 Building a Transfer Descriptor

Before a transfer can be executed from the linked list, a dTD must be built to describe the transfer.

Use the following procedure for building dTDs.

Allocate 8-DWord dTD block of memory aligned to 8-DWord boundaries. Example: bit address 4:0 would be equal to "00000"

Write the following fields:

1. Initialize first 7 DWords to 0.
2. Set the terminate bit to 1.
3. Fill in total bytes with transfer size.
4. Set the interrupt on complete if desired.
5. Initialize the status field with the active bit set to 1 and all remaining status bits set to 0.
6. Fill in buffer pointer page 0 and the current offset to point to the start of the data buffer.
7. Initialize buffer pointer page 1 through page 4 to be one greater than each of the previous buffer pointer.

65.4.6.6.3 Executing A Transfer Descriptor

To safely add a dTD, the DCD must be follow this procedure which will handle the event where the device controller reaches the end of the dTD list at the same time a new dTD is being added to the end of the list.

Determine whether the link list is empty: Check DCD driver to see if pipe is empty (internal representation of linked-list should indicate if any packets are outstanding).

- Case 1: Link list is empty
  a. Write dQH next pointer AND dQH terminate bit to 0 as a single DWord operation.
  b. Clear active & halt bit in dQH (in case set from a previous error).
  c. Prime endpoint by writing 1 to correct bit position in Endpoint Prime (USB_*n*ENDPTPRIME).
- Case 2: Link list is not empty
  a. Add dTD to end of linked list.
  b. Read correct prime bit in Endpoint Prime (USB_*n*ENDPTPRIME)- if 1 DONE.
  c. Set ATDTW bit in USBCMD register to 1.
  d. Read correct status bit in Endpoint Status (USB_*n*ENDPTSTAT). (store in tmp. variable for later)
  e. Read ATDTW bit in USBCMD register.
     - If 0 goto 3.
     - If 1 continue to 6.
  f. Write ATDTW bit in USBCMD register to 0.
  g. If status bit read in (3) is 1 DONE.

h. If status bit read in (3) is 0 then Goto Case 1: Step 1.

65.4.6.6.4 Transfer Completion

After a dTD has been initialized and the associated endpoint primed the device controller will execute the transfer upon the host-initiated request. The DCD will be notified with a USB interrupt if the Interrupt On Complete bit was set or alternately, the DCD can poll the endpoint complete register to find when the dTD had been executed. After a dTD has been executed, DCD can check the status bits to determine success or failure.

NOTE

Multiple dTD can be completed in a single endpoint complete notification. After clearing the notification, DCD must search the dTD linked list and retire all dTDs that have finished (Active bit cleared).

By reading the status fields of the completed dTDs, the DCD can determine if the transfers completed successfully. Success is determined with the following combination of status bits:

- Active = 0
- Halted = 0
- Transaction Error = 0
- Data Buffer Error = 0

Should any combination other than the one shown above exist, the DCD must take proper action. Transfer failure mechanisms are indicated in the Device Error Matrix.

In addition to checking the status bit the DCD must read the Transfer Bytes field to determine the actual bytes transferred. When a transfer is complete, the Total Bytes transferred is by decremented by the actual bytes transferred. For Transmit packets, a packet is only complete after the actual bytes reaches zero, but for receive packets, the host may send fewer bytes in the transfer according the USB variable length packet protocol.

65.4.6.6.5 Flushing/De-priming an Endpoint

It is necessary for the DCD to flush to de-prime one more endpoints on a USB device reset or during a broken control transfer.

There may also be application specific requirements to stop transfers in progress. The following procedure can be used by the DCD to stop a transfer in progress:

1. Write a '1' to the corresponding bit(s) in Endpoint Flush (USB_*n*ENDPTFLUSH).
2. Wait until all bits in Endpoint Flush (USB_*n*ENDPTFLUSH) are '0'.

- Software note: this operation may take a large amount of time depending on the USB bus activity. It is not desirable to have this wait loop within an interrupt service routine.

3. Read Endpoint Status (USB_*n*ENDPTSTAT) to ensure that for all endpoints commanded to be flushed, that the corresponding bits are now '0'. If the corresponding bits are '1' after step #2 has finished, then the flush failed as described in the following:
   - Explanation: In very rare cases, a packet is in progress to the particular endpoint when commanded flush using Endpoint Flush (USB_*n*ENDPTFLUSH). A safeguard is in place to refuse the flush to ensure that the packet in progress completes successfully. The DCD may need to repeatedly flush any endpoints that fail to flush be repeating steps 1-3 until each endpoint is successfully flushed.

65.4.6.6.6 Device Error Matrix

The following table summarizes packet errors that are not automatically handled by the Device Controller.

Table 65-72. Device Error Matrix

Error	Direction	Packet Type	Data Buffer Error Bit	Transaction Error Bit
Overflow **	RX	Any	1	0
ISO Packet Error	RX	ISO	0	1
ISO Fulfillment Error	Both	ISO	0	1

Notice that the device controller handles all errors on Bulk/Control/Interrupt Endpoints except for a data buffer overflow. However, for ISO endpoints, errors packets are not retried and errors are tagged as indicated. The table below describes the errors.

Table 65-73. Error Descriptions

Error	Description
Overflow	Number of bytes received exceeded max. packet size or total buffer length. ** This error will also set the Halt bit in the dQH and if there are dTDs remaining in the linked list for the endpoint, then those will not be executed.
ISO Packet Error	CRC Error on received ISO packet. Contents not guaranteed to be correct.
ISO Fulfillment Error	Host failed to complete the number of packets defined in the dQH mult field within the given (micro)frame. For scheduled data delivery the DCD may need to readjust the data queue because a fulfillment error will cause Device Controller to cease data transfers on the pipe for one (micro)frame. During the "dead" (micro)frame, the Device Controller reports error on the pipe and primes for the following frame.

65.4.6.7 Servicing Interrupts

The interrupt service routine must consider that there are high-frequency, low-frequency operations, and error operations and order accordingly.

65.4.6.7.1 High-Frequency Interrupts

High frequency interrupts in particular should be handled in the order below. The most important of these is listed first because the DCD must acknowledge a setup buffer in the timeliest manner possible.

The table below describes the High frequency interrupt events.

Table 65-74. High Frequency Interrupt Events

Execution Order	Interrupt	Action
1a	USB Interrupt - USB.ENDPTSETUPSTATUS	Copy contents of setup buffer and acknowledge setup packet (as indicated in Figure 65-30 shows the End Point Queue Head). Process setup packet according to USB 2.0 Chapter 9 or application specific protocol.
1b	USB Interrupt[1] - USB.ENDPTCOMPLETE	Handle completion of dTD as indicated in Figure 65-30 shows the End Point Queue Head.
2	SOF Interrupt	Action as deemed necessary by application. This interrupt may not have a use in all applications.

1. It is likely that multiple interrupts to stack up on any call to the Interrupt Service Routine AND during the Interrupt Service Routine.

65.4.6.7.2 Low-Frequency Interrupts

The low frequency interrupts can be handled in any order because they do not occur often in comparison to the high-frequency interrupts.

The table below shows the Low frequency interrupt events.

Table 65-75. Low Frequency Interrupt Events

Interrupt	Action
Port Change	Change software state information.
Sleep Enable (Suspend)	Change software state information. Low power handling as necessary.
Reset Received	Change software state information. Abort pending transfers.

65.4.6.7.3 Error Interrupts

Error interrupts will be least frequent and should be placed last in the interrupt service routine.

The following table shows the error interrput events.

Table 65-76. Error Interrupt Events

Interrupt	Action
USB Error Interrupt	This error is redundant because it combines USB Interrupt and an error status in the dTD. The DCD will more aptly handle packet-level errors by checking dTD status field upon receipt of USB Interrupt (w/ USB.ENDPTCOMPLETE).
System Error	Unrecoverable error. Immediate Reset of core; free transfers buffers in progress and restart the DCD.

65.5 USB Non-Core Memory Map/Register Definition

There are two kinds of registers in the USB module: USB core registers and USB non-core registers. USB core registers are used to control USB core functions, and more independent of USB features. Each USB controller core has its own core registers. USB non-core registers are additional to USB core registers, and more dependent on USB features. i.MX series products vary in non-core registers.

This section describes only the USB non-core registers. For detailed descriptions of USB core registers, please refer to Register Interface.

NOTE

- For reserved bits, please preserve the value when writing (read its reset value, then write this value back)
- USBNC_USB_" prefix in register name indicates it is a USB non-core register.

USBNC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_4800	USB OTG Control Register (USBNC_USB_OTG_CTRL)	32	R/W	0000_3000h	65.5.1/5391
218_4804	USB Host1 Control Register (USBNC_USB_UH1_CTRL)	32	R/W	0000_3000h	65.5.2/5394
218_4808	USB Host2 Control Register (USBNC_USB_UH2_CTRL)	32	R/W	0000_1000h	65.5.3/5397
218_480C	USB Host3 Control Register (USBNC_USB_UH3_CTRL)	32	R/W	0000_1000h	65.5.4/5399

Table continues on the next page...

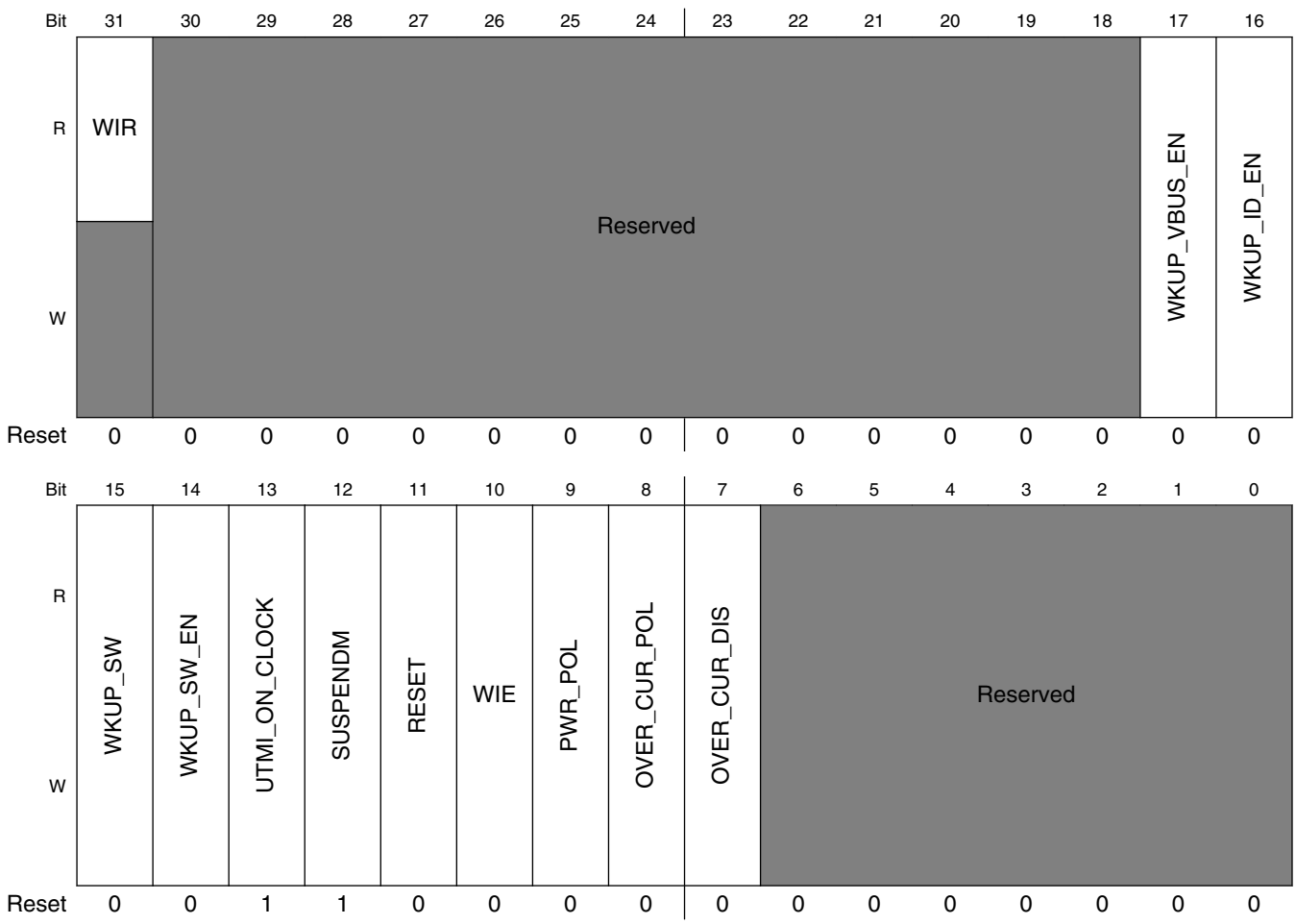
USBNC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_4810	USB Host2 HSIC Control Register (USBNC_USB_UH2_HSIC_CTRL)	32	R/W	0000_0042h	65.5.5/5401
218_4814	USB Host3 HSIC Control Register (USBNC_USB_UH3_HSIC_CTRL)	32	R/W	0000_0042h	65.5.6/5403
218_4818	OTG UTMI PHY Control 0 Register (USBNC_USB_OTG_PHY_CTRL_0)	32	R/W	0000_0000h	65.5.7/5404
218_481C	Host1 UTMI PHY Control 0 Register (USBNC_USB_UH1_PHY_CTRL_0)	32	R/W	0000_0098h	65.5.8/5405

65.5.1 USB OTG Control Register (USBNC_USB_OTG_CTRL)

The USB OTG control register controls the integration specific features of the USB OTG module. These features are not directly related to the USB functionality, but control special features, interfacing on the USB ports, as well as power control and wake-up functionality.

Address: 218_4000h base + 800h offset = 218_4800h



USBNC_USB_OTG_CTRL field descriptions

Field	Description
31 WIR	OTG Wake-up Interrupt Request This bit indicates that a wake-up interrupt request is received on the OTG port. This bit is cleared by disabling the wake-up interrupt (clearing bit "OWIE"). 1 Wake-up Interrupt Request received 0 No wake-up interrupt request received

Table continues on the next page...

USBNC_USB_OTG_CTRL field descriptions (continued)

Field	Description
30–18 -	This field is reserved. Reserved
17 WKUP_VBUS_EN	OTG wake-up on VBUS change enable 1 Enable 0 Disable
16 WKUP_ID_EN	OTG Wake-up on ID change enable 1 Enable 0 Disable
15 WKUP_SW	OTG Software Wake-up 1 Force wake-up 0 Inactive
14 WKUP_SW_EN	OTG Software Wake-up Enable 1 Enable 0 Disable
13 UTMI_ON_CLOCK	Force OTG UTMI PHY clock output on even if suspend mode. 1 Force clock output on 0 Inactive
12 SUSPENDM	Force OTG UTMI PHY Suspend. For NXP test only. This bit is used to put PHY into low-power suspend mode. During normal operation, S/W should set bits SUSP and PHCD in USB core register PORTSC1 to put PHY into suspend mode. 1 Inactive 0 Force OTG UTMI PHY Suspend
11 RESET	Force OTG UTMI PHY Reset This bit is used to force a reset to the UTMI PHY. For NXP test only. During normal operation, S/W should use USBCMD.RST bit to reset the UTMI PHY 1 Reset the PHY 0 Inactive
10 WIE	OTG Wake-up Interrupt Enable This bit enables or disables the OTG wake-up interrupt. Disabling the interrupt also clears the Interrupt request bit. Wake-up interrupt enable should be turned off after receiving a wake-up interrupt and turned on again prior to going in suspend mode 1 Interrupt Enabled 0 Interrupt Disabled
9 PWR_POL	OTG Power Polarity This bit should be set according to power switch's enable polarity. 1 Power switch has an active-high enable input 0 Power switch has an active-low enable input

Table continues on the next page...

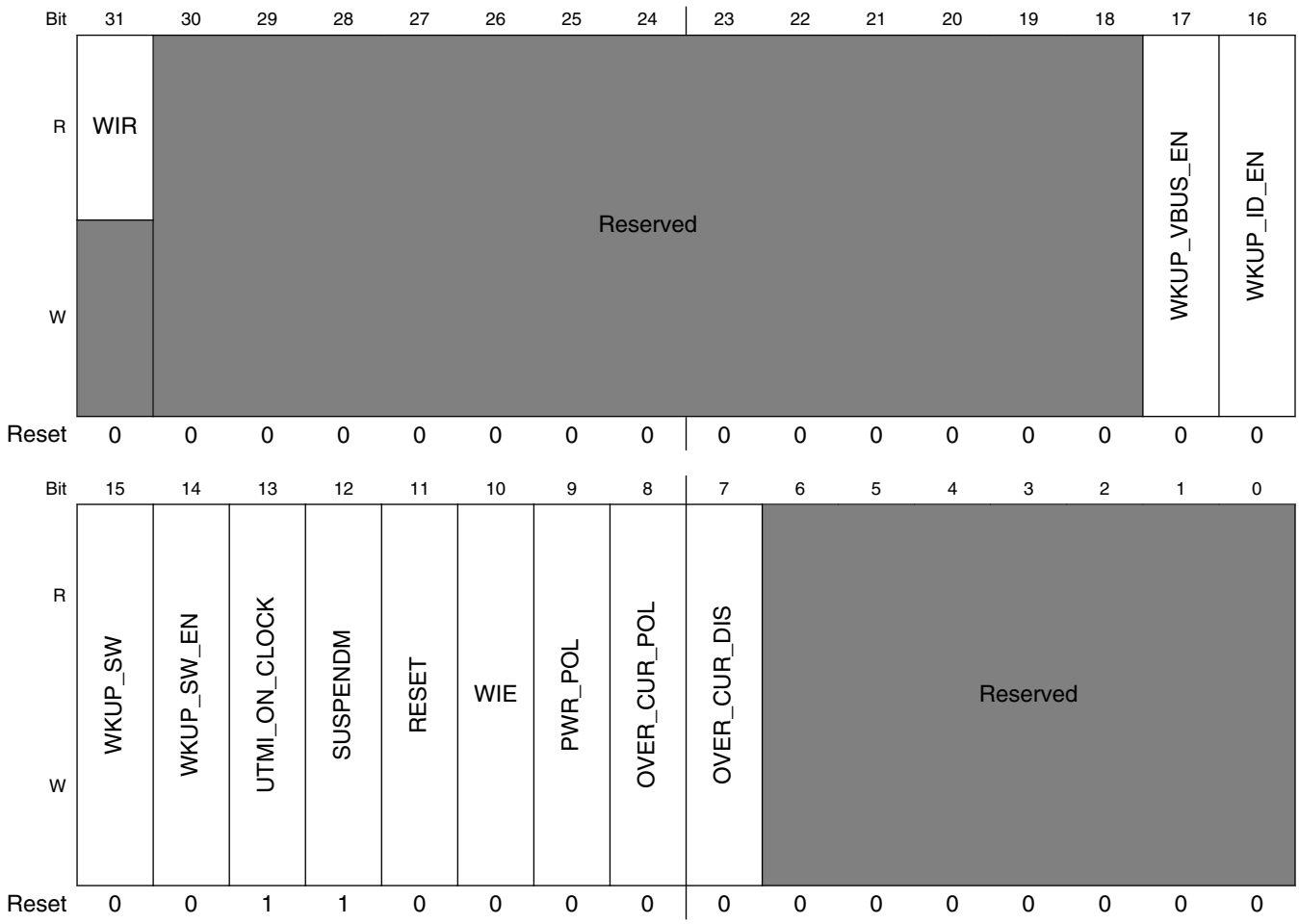
USBNC_USB_OTG_CTRL field descriptions (continued)

Field	Description
8 OVER_CUR_POL	OTG Polarity of Overcurrent The polarity of OTG port overcurrent event 1 Low active (low on this signal represents an overcurrent condition) 0 High active (high on this signal represents an overcurrent condition)
7 OVER_CUR_DIS	Disable OTG Overcurrent Detection 1 Disables overcurrent detection 0 Enables overcurrent detection
-	This field is reserved. Reserved

65.5.2 USB Host1 Control Register (USBNC_USB_UH1_CTRL)

The USB Host1 control register controls the integration specific features of the USB Host1 module. These features are not directly related to the USB functionality, but control special features, interfacing on the USB ports, as well as power control and wake-up functionality.

Address: 218_4000h base + 804h offset = 218_4804h



USBNC_USB_UH1_CTRL field descriptions

Field	Description
31 WIR	Host 1 Wake-up Interrupt Request This bit indicates that a wake-up interrupt request is received on the OTG port. This bit is cleared by disabling the wake-up interrupt (clearing bit "OWIE"). 1 Wake-up Interrupt Request received 0 No wake-up interrupt request received

Table continues on the next page...

USBNC_USB_UH1_CTRL field descriptions (continued)

Field	Description
30–18 -	This field is reserved. Reserved
17 WKUP_VBUS_EN	Host 1 wake-up on VBUS change enable 1 Enable 0 Disable
16 WKUP_ID_EN	Host 1 Wake-up on ID change enable 1 Enable 0 Disable
15 WKUP_SW	Host 1 Software Wake-up 1 Force wake-up 0 Inactive
14 WKUP_SW_EN	Host 1 Software Wake-up Enable 1 Enable 0 Disable
13 UTMI_ON_CLOCK	Force Host 1 UTMI PHY clock output on even if in low-power suspend mode. 1 Enable 0 Disable
12 SUSPENDM	Force Host 1 UTMI PHY Suspend. For NXP test only. This bit is used to put PHY into low-power suspend mode. During normal operation, S/W should set bits SUSP and PHCD in USB core register PORTSC1 to put PHY into suspend mode. 1 Disable 0 Enable
11 RESET	Force Host 1 UTMI PHY Reset. For NXP test only. This bit is used to force a reset to the UTMI PHY. During normal operation, S/W should use USBCMD.RST bit to reset the UTMI PHY 1 Reset the PHY 0 Inactive
10 WIE	Host 1 Wake-up Interrupt Enable This bit enables or disables the Host 1 wake-up interrupt. Disabling the interrupt also clears the Interrupt request bit. Wake-up interrupt enable should be turned off after receiving a wake-up interrupt and turned on again prior to going in suspend mode 1 Interrupt Enabled 0 Interrupt Disabled
9 PWR_POL	Host1 Power Polarity This bit should be set according to the power switch's enable polarity. 1 Power switch has an active-high enable input 0 Power switch has an active-low enable input

Table continues on the next page...

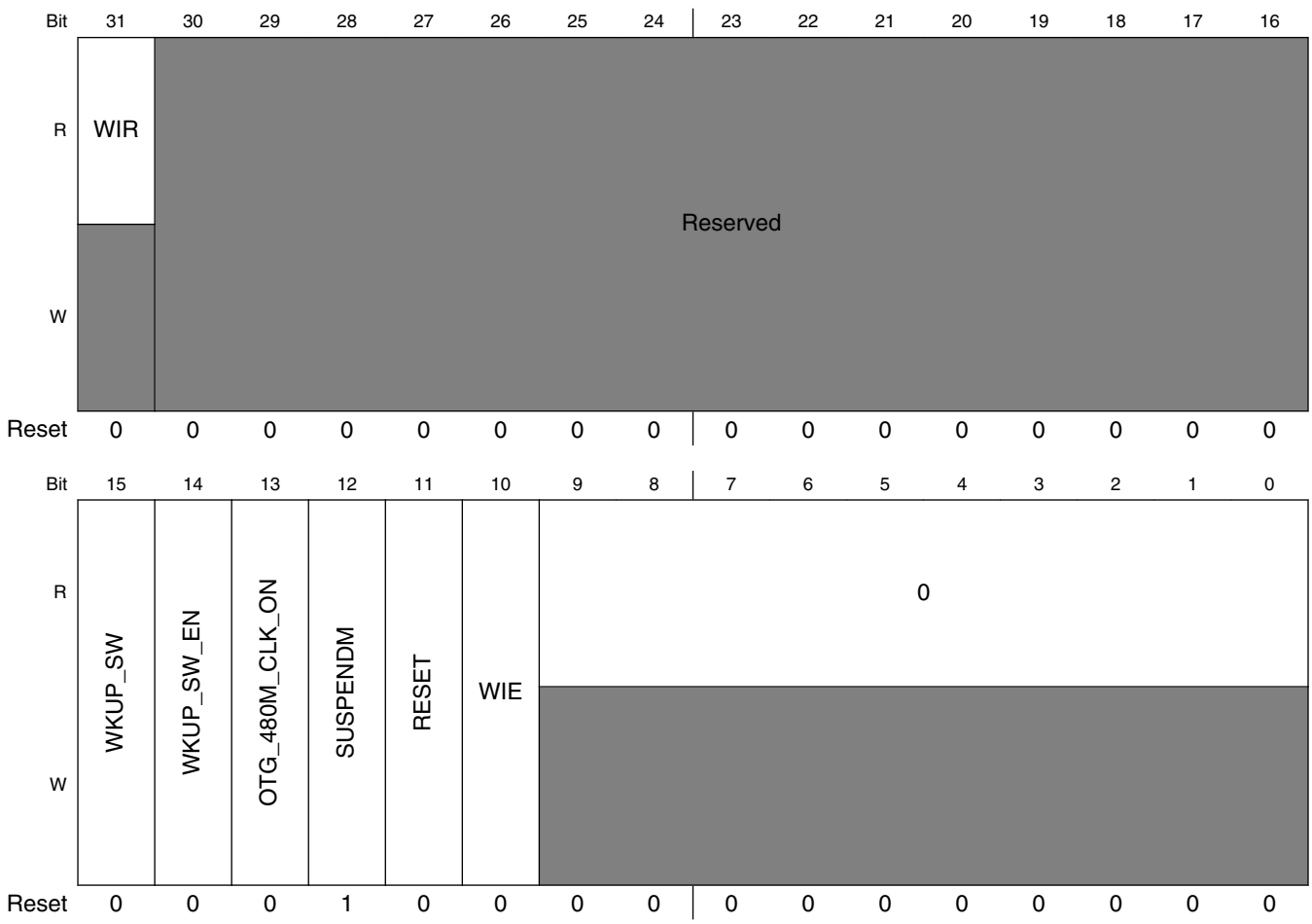
USBNC_USB_UH1_CTRL field descriptions (continued)

Field	Description
8 OVER_CUR_POL	Host 1 Polarity of Overcurrent The polarity of Host 1 port overcurrent event 1 Low active (low on this signal represents an overcurrent condition) 0 High active (high on this signal represents an overcurrent condition)
7 OVER_CUR_DIS	Disable Host 1 Overcurrent Detection 1 Disables overcurrent detection 0 Enables overcurrent detection
-	This field is reserved. Reserved

65.5.3 USB Host2 Control Register (USBNC_USB_UH2_CTRL)

The USB Host2 control register controls the integration specific features of the USB host2 module. These features are not directly related to the USB functionality, but control special features, interfacing on the USB ports, as well as power control and wake-up functionality.

Address: 218_4000h base + 808h offset = 218_4808h



USBNC_USB_UH2_CTRL field descriptions

Field	Description
31 WIR	Host 2 Wake-up Interrupt Request This bit indicates that a wake-up interrupt request is received on the Host 2 port. This bit is cleared by disabling the wake-up interrupt (clearing bit "OWIE"). 1 Wake-up Interrupt Request received 0 No wake-up interrupt request received

Table continues on the next page...

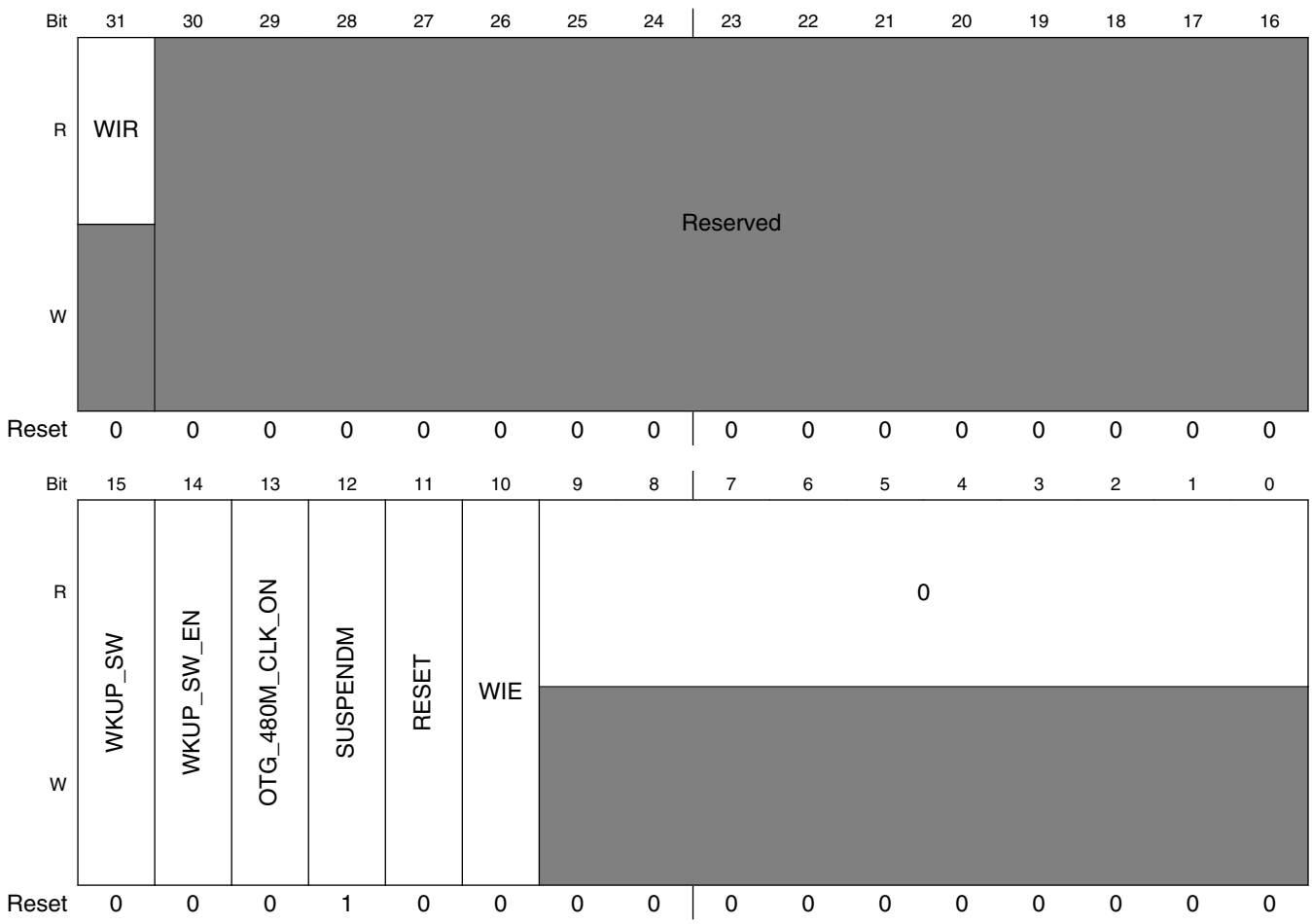
USBNC_USB_UH2_CTRL field descriptions (continued)

Field	Description
30–16 -	This field is reserved. Reserved
15 WKUP_SW	Host 2 Software Wake-up 1 Force wake-up 0 Inactive
14 WKUP_SW_EN	Host 2 Software Wake-up Enable 1 Enable 0 Disable
13 OTG_480M_ CLK_ON	Force OTG UTMI PHY 480M clock output on when Host 2 is not in suspend mode. 1 Force OTG UTMI PHY 480M clock output on 0 Inactive
12 SUSPENDM	Force Host 2 UTMI PHY Suspend This bit is used to put PHY into suspend mode. During normal operation, S/W should set bits SUSP and PHCD in USB core register PORTSC1 to put PHY into suspend mode. For NXP test only. 1 Disable 0 Enable
11 RESET	Force Host 2 UTMI PHY Reset This bit is used to force a reset to the UTMI PHY. During normal operation, S/W should set USBCMD.RST bit to reset the UTMI PHY For NXP test only. 1 Reset the PHY 0 Inactive
10 WIE	Host 2 Wake-up Interrupt Enable This bit enables or disables the Host 2 wake-up interrupt. Disabling the interrupt also clears the Interrupt request bit. Wake-up interrupt enable should be turned off after receiving a wake-up interrupt and turned on again prior to going in suspend mode 1 Interrupt Enabled 0 Interrupt Disabled
Reserved	This read-only field is reserved and always has the value 0.

65.5.4 USB Host3 Control Register (USBNC_USB_UH3_CTRL)

The USB Host3 control register controls the integration specific features of the USB Host3 module. These features are not directly related to the USB functionality, but control special features, interfacing on the USB ports, as well as power control and wake-up functionality.

Address: 218_4000h base + 80Ch offset = 218_480Ch



USBNC_USB_UH3_CTRL field descriptions

Field	Description
31 WIR	Host 3 Wake-up Interrupt Request This bit indicates that a wake-up interrupt request is received on the OTG port. This bit is cleared by disabling the wake-up interrupt (clearing bit "OWIE"). 1 Wake-up interrupt received 0 No Wake-up interrupt received

Table continues on the next page...

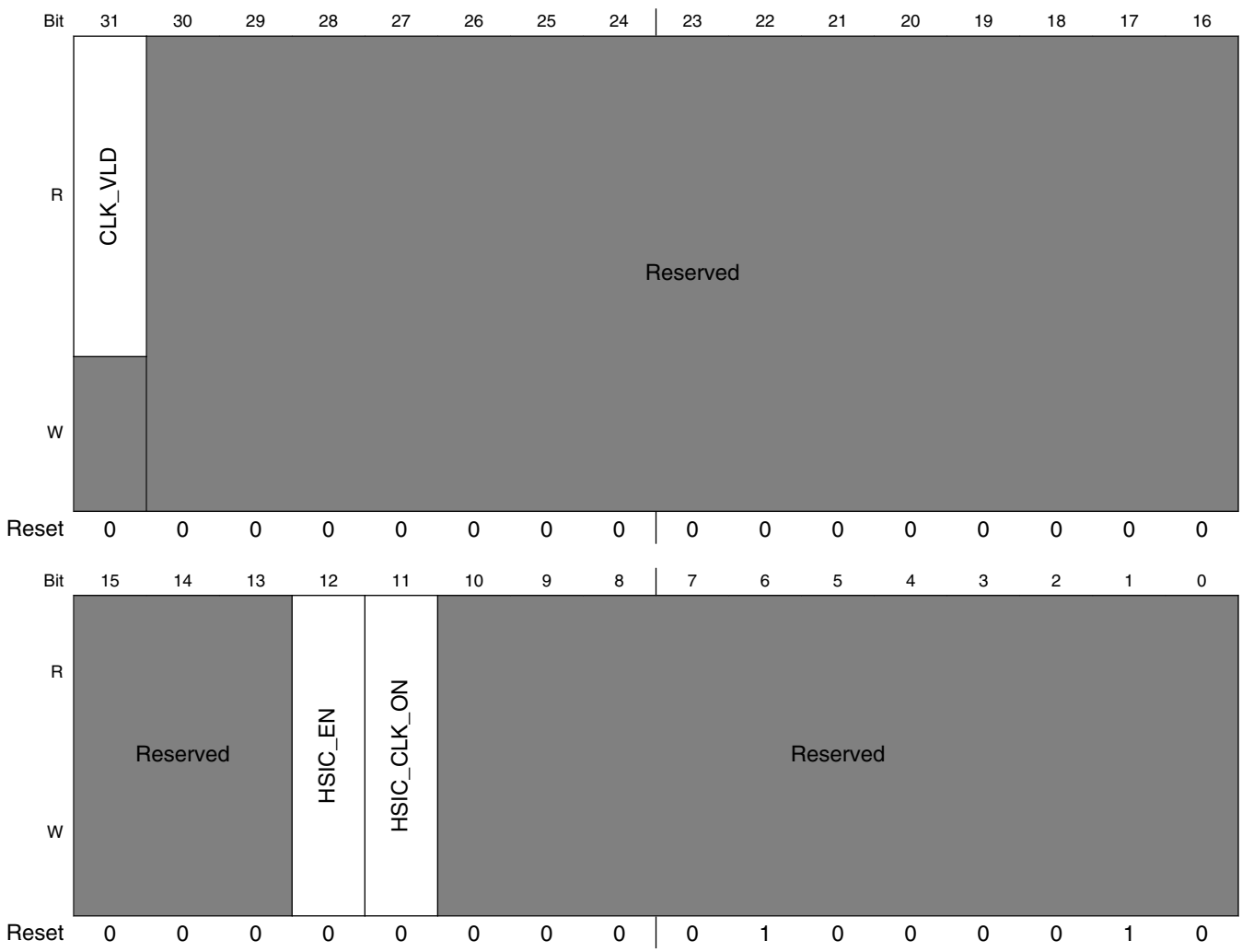
USBNC_USB_UH3_CTRL field descriptions (continued)

Field	Description
30–16 -	This field is reserved. Reserved
15 WKUP_SW	Host 3 Software Wake-up 1 Force wake-up 0 Inactive
14 WKUP_SW_EN	Host 3 Software Wake-up Enable 1 Enable 0 Disable
13 OTG_480M_ CLK_ON	Force OTG UTMI PHY 480M clock output on when Host 3 is not in suspend mode. 1 Force OTG UTMI PHY 480M clock output on 0 Inactive
12 SUSPENDM	Force Host 3 UTMI PHY Suspend This bit is used to put PHY into suspend mode. During normal operation, S/W should set bits SUSP and PHCD in USB core register PORTSC1 to put PHY into suspend mode. For NXP test only. 1 Inactive 0 Force OTG UTMI PHY Suspend
11 RESET	Force Host 3 UTMI PHY Reset This bit is used to force a reset to the UTMI PHY. During normal operation, S/W should set USBCMD.RST bit to reset the UTMI PHY For NXP test only. 1 Reset the PHY 0 Inactive
10 WIE	Host 3 Wake-up Interrupt Enable This bit enables or disables the Host 3 wake-up interrupt. Disabling the interrupt also clears the Interrupt request bit. Wake-up interrupt enable should be turned off after receiving a wake-up interrupt and turned on again prior to going in suspend mode 1 Interrupt Enabled 0 Interrupt Disabled
Reserved	This read-only field is reserved and always has the value 0.

65.5.5 USB Host2 HSIC Control Register (USBNC_USB_UH2_HSIC_CTRL)

The USB Host2 HSIC control register controls Host2 high speed IC configuration. These features are not directly related to the USB functionality, but control special features, interfacing on the USB ports, as well as power control.

Address: 218_4000h base + 810h offset = 218_4810h



USBNC_USB_UH2_HSIC_CTRL field descriptions

Field	Description
31 CLK_VLD	Indicating whether Host2 HSIC clock is valid.

Table continues on the next page...

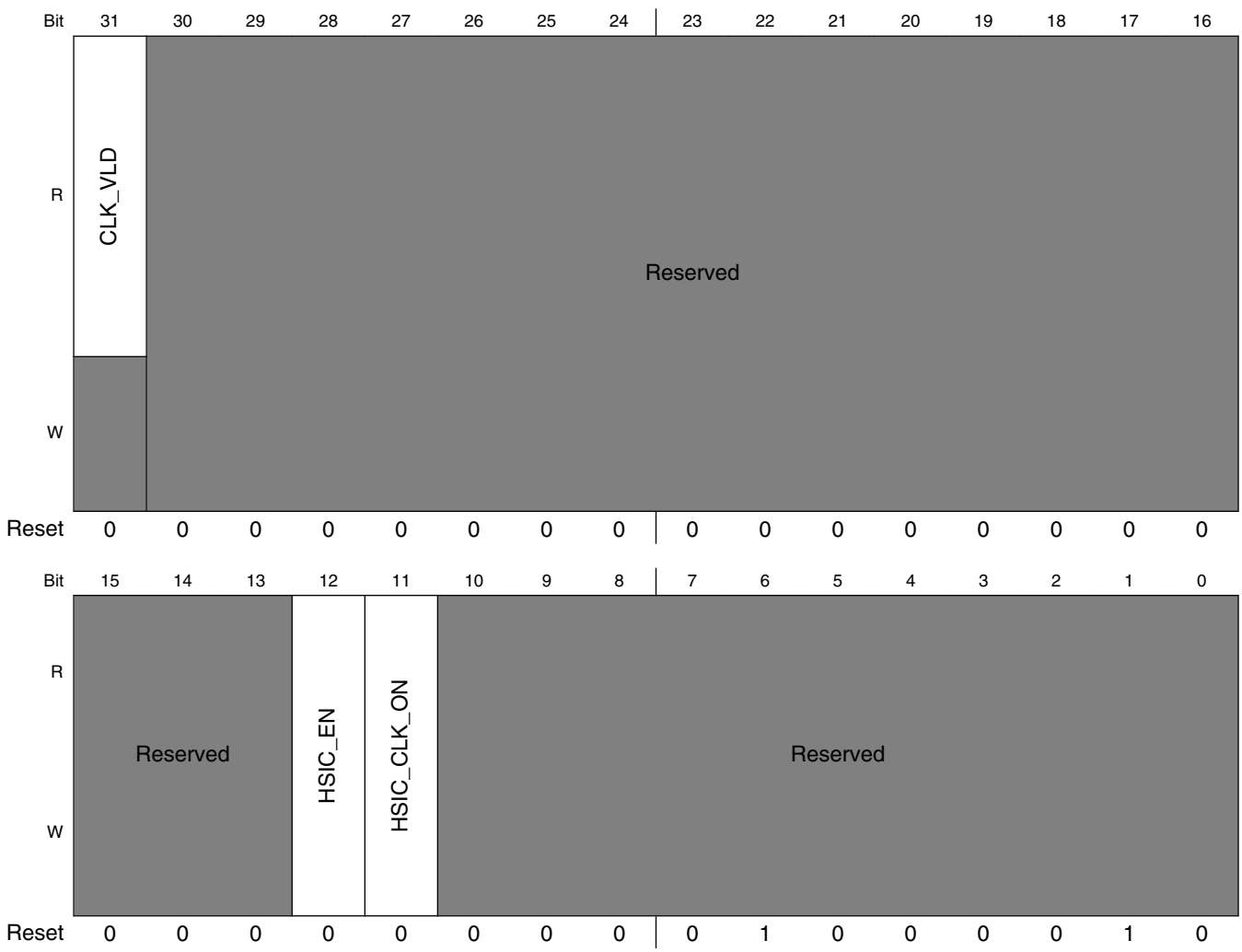
USBNC_USB_UH2_HSIC_CTRL field descriptions (continued)

Field	Description
	1 Valid 2 Invalid
30–13 -	This field is reserved. Reserved
12 HSIC_EN	Host2 HSIC enable 1 Enabled 0 Disabled
11 HSIC_CLK_ON	Force Host2 HSIC module 480M clock on, even when in Host 2 is in suspend mode. 1 Active 0 Inactive
-	This field is reserved. Reserved

65.5.6 USB Host3 HSIC Control Register (USBNC_USB_UH3_HSIC_CTRL)

The USB Host3 HSIC control register controls Host3 high speed IC configuration. These features are not directly related to the USB functionality, but control special features, interfacing on the USB ports, as well as power control.

Address: 218_4000h base + 814h offset = 218_4814h



USBNC_USB_UH3_HSIC_CTRL field descriptions

Field	Description
31 CLK_VLD	Indicating whether Host3 HSIC clock is valid.

Table continues on the next page...

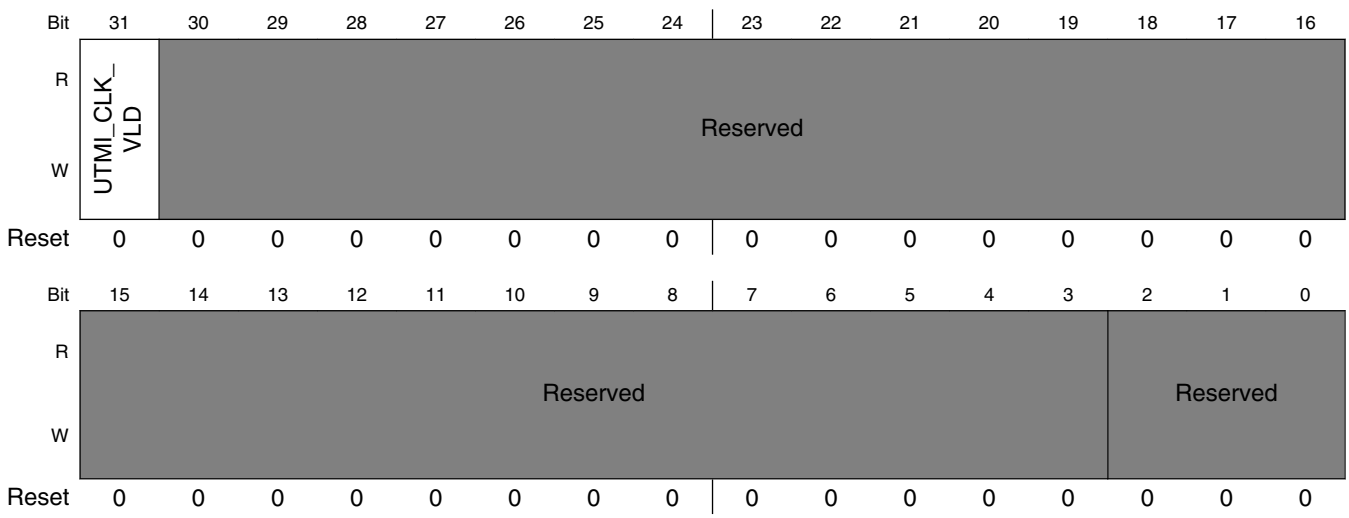
USBNC_USB_UH3_HSIC_CTRL field descriptions (continued)

Field	Description
	1 Valid 2 Invalid
30–13 -	This field is reserved. Reserved
12 HSIC_EN	Host3 HSIC enable 1 Enabled 0 Disabled
11 HSIC_CLK_ON	Force Host3 HSIC module 480M clock on, even when in Host 2 is in suspend mode. 1 Active 0 Inactive
-	This field is reserved. Reserved

65.5.7 OTG UTMI PHY Control 0 Register (USBNC_USB_OTG_PHY_CTRL_0)

USB OTG UTMI PHY control register 0 is used to control the on-chip OTG UTMI PHY.

Address: 218_4000h base + 818h offset = 218_4818h



USBNC_USB_OTG_PHY_CTRL_0 field descriptions

Field	Description
31 UTMI_CLK_VLD	Indicating whether OTG UTMI PHY clock is valid

Table continues on the next page...

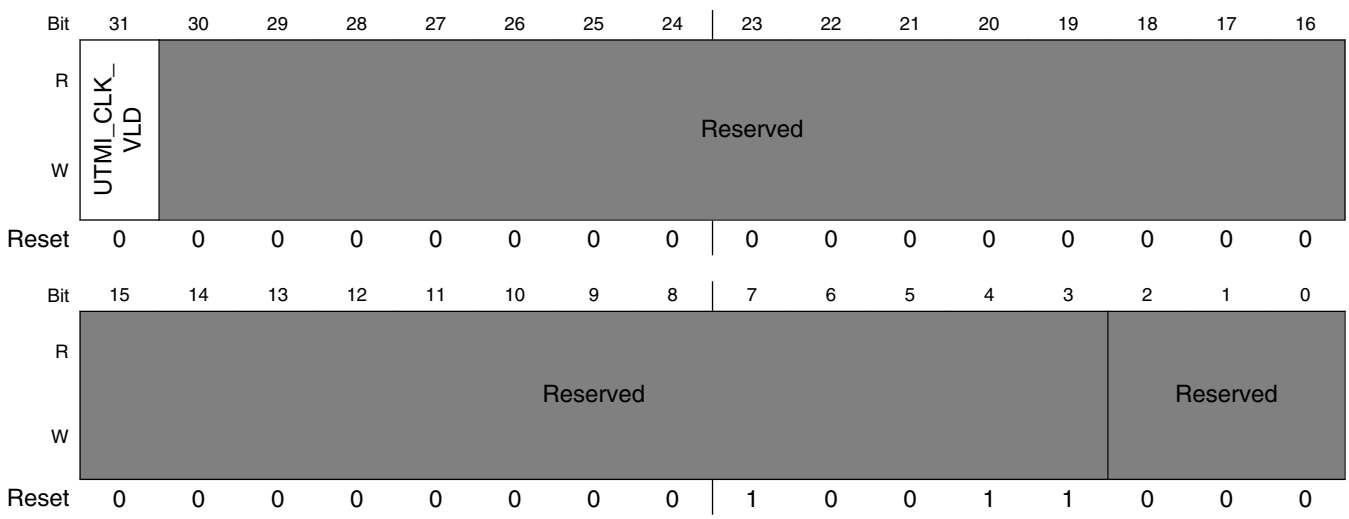
USBNC_USB_OTG_PHY_CTRL_0 field descriptions (continued)

Field	Description
	1 Valid 0 Invalid
30–3 -	This field is reserved. Reserved
-	This field is reserved. Reserved

65.5.8 Host1 UTMI PHY Control 0 Register (USBNC_USB_UH1_PHY_CTRL_0)

USB Host1 UTMI PHY Control Register 0 are used to control the on-chip Host1 UTMI PHY.

Address: 218_4000h base + 81Ch offset = 218_481Ch



USBNC_USB_UH1_PHY_CTRL_0 field descriptions

Field	Description
31 UTMI_CLK_VLD	Indicating whether Host 1 UTMI PHY clock is valid 1 Valid 0 Invalid
30–3 -	This field is reserved. Reserved
-	This field is reserved. Reserved

65.6 USB Core Memory Map/Register Definition

USB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_4000	Identification register (USB_UOG_ID)	32	R	E401_FA05h	65.6.1/5411
218_4004	Hardware General (USB_UOG_HWGENERAL)	32	R	0000_0015h	65.6.2/5412
218_4008	Host Hardware Parameters (USB_UOG_HWHOST)	32	R	1002_0001h	65.6.3/5413
218_400C	Device Hardware Parameters (USB_UOG_HWDEVICE)	32	R	0000_0011h	65.6.4/5414
218_4010	TX Buffer Hardware Parameters (USB_UOG_HWTXBUF)	32	R	8008_0B08h	65.6.5/5414
218_4014	RX Buffer Hardware Parameters (USB_UOG_HWRXBUF)	32	R	0000_0808h	65.6.6/5415
218_4080	General Purpose Timer #0 Load (USB_UOG_GPTIMER0LD)	32	R/W	0000_0000h	65.6.7/5416
218_4084	General Purpose Timer #0 Controller (USB_UOG_GPTIMER0CTRL)	32	R/W	0000_0000h	65.6.8/5416
218_4088	General Purpose Timer #1 Load (USB_UOG_GPTIMER1LD)	32	R/W	0000_0000h	65.6.9/5418
218_408C	General Purpose Timer #1 Controller (USB_UOG_GPTIMER1CTRL)	32	R/W	0000_0000h	65.6.10/ 5418
218_4090	System Bus Config (USB_UOG_SBUSCFG)	32	R/W	0000_0002h	65.6.11/ 5419
218_4100	Capability Registers Length (USB_UOG_CAPLENGTH)	8	R	40h	65.6.12/ 5420
218_4102	Host Controller Interface Version (USB_UOG_HCIVERSION)	16	R	0100h	65.6.13/ 5421
218_4104	Host Controller Structural Parameters (USB_UOG_HCSPARAMS)	32	R	0001_0011h	65.6.14/ 5421
218_4108	Host Controller Capability Parameters (USB_UOG_HCCPARAMS)	32	R	0000_0006h	65.6.15/ 5423
218_4120	Device Controller Interface Version (USB_UOG_DCIVERSION)	16	R	0001h	65.6.16/ 5425
218_4124	Device Controller Capability Parameters (USB_UOG_DCCPARAMS)	32	R	0000_0188h	65.6.17/ 5425
218_4140	USB Command Register (USB_UOG_USBCMD)	32	R/W	0008_0000h	65.6.18/ 5427
218_4144	USB Status Register (USB_UOG_USBSTS)	32	R/W	0000_0000h	65.6.19/ 5431
218_4148	Interrupt Enable Register (USB_UOG_USBINTR)	32	R/W	0000_0000h	65.6.20/ 5435
218_414C	USB Frame Index (USB_UOG_FRINDEX)	32	R/W	0000_0000h	65.6.21/ 5437
218_4154	Frame List Base Address (USB_UOG_PERIODICLISTBASE)	32	R/W	0000_0000h	65.6.22/ 5438

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_4154	Device Address (USB_UOG_DEVICEADDR)	32	R/W	0000_0000h	65.6.23/ 5438
218_4158	Next Asynch. Address (USB_UOG_ASYNCLISTADDR)	32	R/W	0000_0000h	65.6.24/ 5439
218_4158	Endpoint List Address (USB_UOG_ENDPTLISTADDR)	32	R/W	0000_0000h	65.6.25/ 5440
218_4160	Programmable Burst Size (USB_UOG_BURSTSIZE)	32	R/W	0000_0000h	65.6.26/ 5440
218_4164	TX FIFO Fill Tuning (USB_UOG_TXFILLTUNING)	32	R/W	0000_0808h	65.6.27/ 5441
218_4178	Endpoint NAK (USB_UOG_ENDPTNAK)	32	R/W	0000_0000h	65.6.28/ 5443
218_417C	Endpoint NAK Enable (USB_UOG_ENDPTNAKEN)	32	R/W	0000_0000h	65.6.29/ 5443
218_4180	Configure Flag Register (USB_UOG_CONFIGFLAG)	32	R	0000_0001h	65.6.30/ 5444
218_4184	Port Status & Control (USB_UOG_PORTSC1)	32	R/W	1000_0000h	65.6.31/ 5444
218_41A4	On-The-Go Status & control (USB_UOG_OTGSC)	32	R/W	0000_0120h	65.6.32/ 5451
218_41A8	USB Device Mode (USB_UOG_USBMODE)	32	R/W	0000_0000h	65.6.33/ 5455
218_41AC	Endpoint Setup Status (USB_UOG_ENDPTSETUPSTAT)	32	R/W	0000_0000h	65.6.34/ 5456
218_41B0	Endpoint Prime (USB_UOG_ENDPTPRIME)	32	R/W	0000_0000h	65.6.35/ 5457
218_41B4	Endpoint Flush (USB_UOG_ENDPTFLUSH)	32	R/W	0000_0000h	65.6.36/ 5458
218_41B8	Endpoint Status (USB_UOG_ENDPTSTAT)	32	R	0000_0000h	65.6.37/ 5458
218_41BC	Endpoint Complete (USB_UOG_ENDPTCOMPLETE)	32	R/W	0000_0000h	65.6.38/ 5459
218_41C0	Endpoint Control0 (USB_UOG_ENDPTCTRL0)	32	R/W	0080_0080h	65.6.39/ 5460
218_41C4	Endpoint Control 1 (USB_UOG_ENDPTCTRL1)	32	R/W	0000_0000h	65.6.40/ 5462
218_41C8	Endpoint Control 2 (USB_UOG_ENDPTCTRL2)	32	R/W	0000_0000h	65.6.41/ 5465
218_41CC	Endpoint Control 3 (USB_UOG_ENDPTCTRL3)	32	R/W	0000_0000h	65.6.42/ 5467
218_41D0	Endpoint Control 4 (USB_UOG_ENDPTCTRL4)	32	R/W	0000_0000h	65.6.43/ 5470
218_41D4	Endpoint Control 5 (USB_UOG_ENDPTCTRL5)	32	R/W	0000_0000h	65.6.44/ 5473

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_41D8	Endpoint Control 6 (USB_UOG_ENDPTCTRL6)	32	R/W	0000_0000h	65.6.45/ 5476
218_41DC	Endpoint Control 7 (USB_UOG_ENDPTCTRL7)	32	R/W	0000_0000h	65.6.46/ 5479
218_4200	Identification register (USB_UH1_ID)	32	R	E401_FA05h	65.6.1/5411
218_4204	Hardware General (USB_UH1_HWGENERAL)	32	R	0000_0015h	65.6.2/5412
218_4208	Host Hardware Parameters (USB_UH1_HWHOST)	32	R	1002_0001h	65.6.3/5413
218_4210	TX Buffer Hardware Parameters (USB_UH1_HWTXBUF)	32	R	8008_0B08h	65.6.5/5414
218_4214	RX Buffer Hardware Parameters (USB_UH1_HWRXBUF)	32	R	0000_0808h	65.6.6/5415
218_4280	General Purpose Timer #0 Load (USB_UH1_GPTIMER0LD)	32	R/W	0000_0000h	65.6.7/5416
218_4284	General Purpose Timer #0 Controller (USB_UH1_GPTIMER0CTRL)	32	R/W	0000_0000h	65.6.8/5416
218_4288	General Purpose Timer #1 Load (USB_UH1_GPTIMER1LD)	32	R/W	0000_0000h	65.6.9/5418
218_428C	General Purpose Timer #1 Controller (USB_UH1_GPTIMER1CTRL)	32	R/W	0000_0000h	65.6.10/ 5418
218_4290	System Bus Config (USB_UH1_SBUSCFG)	32	R/W	0000_0002h	65.6.11/ 5419
218_4300	Capability Registers Length (USB_UH1_CAPLENGTH)	8	R	40h	65.6.12/ 5420
218_4302	Host Controller Interface Version (USB_UH1_HCIVERSION)	16	R	0100h	65.6.13/ 5421
218_4304	Host Controller Structural Parameters (USB_UH1_HCSPARAMS)	32	R	0001_0011h	65.6.14/ 5421
218_4308	Host Controller Capability Parameters (USB_UH1_HCCPARAMS)	32	R	0000_0006h	65.6.15/ 5423
218_4340	USB Command Register (USB_UH1_USBCMD)	32	R/W	0008_0000h	65.6.18/ 5427
218_4344	USB Status Register (USB_UH1_USBSTS)	32	R/W	0000_0000h	65.6.19/ 5431
218_4348	Interrupt Enable Register (USB_UH1_USBINTR)	32	R/W	0000_0000h	65.6.20/ 5435
218_434C	USB Frame Index (USB_UH1_FRINDEX)	32	R/W	0000_0000h	65.6.21/ 5437
218_4354	Frame List Base Address (USB_UH1_PERIODICLISTBASE)	32	R/W	0000_0000h	65.6.22/ 5438
218_4358	Next Asynch. Address (USB_UH1_ASYNCLISTADDR)	32	R/W	0000_0000h	65.6.24/ 5439
218_4360	Programmable Burst Size (USB_UH1_BURSTSIZE)	32	R/W	0000_0000h	65.6.26/ 5440
218_4364	TX FIFO Fill Tuning (USB_UH1_TXFILLTUNING)	32	R/W	0000_0808h	65.6.27/ 5441
218_4380	Configure Flag Register (USB_UH1_CONFIGFLAG)	32	R	0000_0001h	65.6.30/ 5444

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_4384	Port Status & Control (USB_UH1_PORTSC1)	32	R/W	1000_0000h	65.6.31/ 5444
218_43A8	USB Device Mode (USB_UH1_USBMODE)	32	R/W	0000_0000h	65.6.33/ 5455
218_4400	Identification register (USB_UH2_ID)	32	R	E401_FA05h	65.6.1/5411
218_4404	Hardware General (USB_UH2_HWGENERAL)	32	R	0000_0015h	65.6.2/5412
218_4408	Host Hardware Parameters (USB_UH2_HWHOST)	32	R	1002_0001h	65.6.3/5413
218_4410	TX Buffer Hardware Parameters (USB_UH2_HWTXBUF)	32	R	8008_0B08h	65.6.5/5414
218_4414	RX Buffer Hardware Parameters (USB_UH2_HWRXBUF)	32	R	0000_0808h	65.6.6/5415
218_4480	General Purpose Timer #0 Load (USB_UH2_GPTIMER0LD)	32	R/W	0000_0000h	65.6.7/5416
218_4484	General Purpose Timer #0 Controller (USB_UH2_GPTIMER0CTRL)	32	R/W	0000_0000h	65.6.8/5416
218_4488	General Purpose Timer #1 Load (USB_UH2_GPTIMER1LD)	32	R/W	0000_0000h	65.6.9/5418
218_448C	General Purpose Timer #1 Controller (USB_UH2_GPTIMER1CTRL)	32	R/W	0000_0000h	65.6.10/ 5418
218_4490	System Bus Config (USB_UH2_SBUSCFG)	32	R/W	0000_0002h	65.6.11/ 5419
218_4500	Capability Registers Length (USB_UH2_CAPLENGTH)	8	R	40h	65.6.12/ 5420
218_4502	Host Controller Interface Version (USB_UH2_HCIVERSION)	16	R	0100h	65.6.13/ 5421
218_4504	Host Controller Structural Parameters (USB_UH2_HCSPARAMS)	32	R	0001_0011h	65.6.14/ 5421
218_4508	Host Controller Capability Parameters (USB_UH2_HCCPARAMS)	32	R	0000_0006h	65.6.15/ 5423
218_4540	USB Command Register (USB_UH2_USBCMD)	32	R/W	0008_0000h	65.6.18/ 5427
218_4544	USB Status Register (USB_UH2_USBSTS)	32	R/W	0000_0000h	65.6.19/ 5431
218_4548	Interrupt Enable Register (USB_UH2_USBINTR)	32	R/W	0000_0000h	65.6.20/ 5435
218_454C	USB Frame Index (USB_UH2_FRINDEX)	32	R/W	0000_0000h	65.6.21/ 5437
218_4554	Frame List Base Address (USB_UH2_PERIODICLISTBASE)	32	R/W	0000_0000h	65.6.22/ 5438
218_4558	Next Asynch. Address (USB_UH2_ASYNCLISTADDR)	32	R/W	0000_0000h	65.6.24/ 5439
218_4560	Programmable Burst Size (USB_UH2_BURSTSIZE)	32	R/W	0000_0000h	65.6.26/ 5440
218_4564	TX FIFO Fill Tuning (USB_UH2_TXFILLTUNING)	32	R/W	0000_0808h	65.6.27/ 5441
218_4580	Configure Flag Register (USB_UH2_CONFIGFLAG)	32	R	0000_0001h	65.6.30/ 5444

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_4584	Port Status & Control (USB_UH2_PORTSC1)	32	R/W	1000_0000h	65.6.31/ 5444
218_45A8	USB Device Mode (USB_UH2_USBMODE)	32	R/W	0000_0000h	65.6.33/ 5455
218_4600	Identification register (USB_UH3_ID)	32	R	E401_FA05h	65.6.1/5411
218_4604	Hardware General (USB_UH3_HWGENERAL)	32	R	0000_0015h	65.6.2/5412
218_4608	Host Hardware Parameters (USB_UH3_HWHOST)	32	R	1002_0001h	65.6.3/5413
218_4610	TX Buffer Hardware Parameters (USB_UH3_HWTXBUF)	32	R	8008_0B08h	65.6.5/5414
218_4614	RX Buffer Hardware Parameters (USB_UH3_HWRXBUF)	32	R	0000_0808h	65.6.6/5415
218_4680	General Purpose Timer #0 Load (USB_UH3_GPTIMER0LD)	32	R/W	0000_0000h	65.6.7/5416
218_4684	General Purpose Timer #0 Controller (USB_UH3_GPTIMER0CTRL)	32	R/W	0000_0000h	65.6.8/5416
218_4688	General Purpose Timer #1 Load (USB_UH3_GPTIMER1LD)	32	R/W	0000_0000h	65.6.9/5418
218_468C	General Purpose Timer #1 Controller (USB_UH3_GPTIMER1CTRL)	32	R/W	0000_0000h	65.6.10/ 5418
218_4690	System Bus Config (USB_UH3_SBUSCFG)	32	R/W	0000_0002h	65.6.11/ 5419
218_4700	Capability Registers Length (USB_UH3_CAPLENGTH)	8	R	40h	65.6.12/ 5420
218_4702	Host Controller Interface Version (USB_UH3_HCIVERSION)	16	R	0100h	65.6.13/ 5421
218_4704	Host Controller Structural Parameters (USB_UH3_HCSPARAMS)	32	R	0001_0011h	65.6.14/ 5421
218_4708	Host Controller Capability Parameters (USB_UH3_HCCPARAMS)	32	R	0000_0006h	65.6.15/ 5423
218_4740	USB Command Register (USB_UH3_USBCMD)	32	R/W	0008_0000h	65.6.18/ 5427
218_4744	USB Status Register (USB_UH3_USBSTS)	32	R/W	0000_0000h	65.6.19/ 5431
218_4748	Interrupt Enable Register (USB_UH3_USBINTR)	32	R/W	0000_0000h	65.6.20/ 5435
218_474C	USB Frame Index (USB_UH3_FRINDEX)	32	R/W	0000_0000h	65.6.21/ 5437
218_4754	Frame List Base Address (USB_UH3_PERIODICLISTBASE)	32	R/W	0000_0000h	65.6.22/ 5438
218_4758	Next Asynch. Address (USB_UH3_ASYNCLISTADDR)	32	R/W	0000_0000h	65.6.24/ 5439
218_4760	Programmable Burst Size (USB_UH3_BURSTSIZE)	32	R/W	0000_0000h	65.6.26/ 5440
218_4764	TX FIFO Fill Tuning (USB_UH3_TXFILLTUNING)	32	R/W	0000_0808h	65.6.27/ 5441
218_4780	Configure Flag Register (USB_UH3_CONFIGFLAG)	32	R	0000_0001h	65.6.30/ 5444

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
218_4784	Port Status & Control (USB_UH3_PORTSC1)	32	R/W	1000_0000h	65.6.31/ 5444
218_47A8	USB Device Mode (USB_UH3_USBMODE)	32	R/W	0000_0000h	65.6.33/ 5455

65.6.1 Identification register (USB_*n*ID)

The ID register identifies the USB 2.0 High-Speed core and its revision.

Address: 218_4000h base + 0h offset + (512d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								REVISION							
W																
Reset	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		NID						Reserved		ID					
W																
Reset	1	1	1	1	1	0	1	0	0	0	0	0	0	1	0	1

USB_*n*ID field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 REVISION	Revision number of the controller core.
15–14 -	This field is reserved. Reserved
13–8 NID	Complement version of ID
7–6 -	This field is reserved. Reserved
ID	Configuration number. This number is set to 0x05 and indicates that the peripheral is USB 2.0 High-Speed core.

65.6.2 Hardware General (USB_*n*HWGENERAL)

General hardware parameters as defined in System Level Issues and Core Configuration.

NOTE

The reset value could vary from instance to instance. Please see the detail in bit field description and ignore reset value in summary table in this case!

Address: 218_4000h base + 4h offset + (512d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					SM			PHYM			PHYW	Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1

USB_*n*HWGENERAL field descriptions

Field	Description
31–11 -	This field is reserved. Reserved
10–9 SM	Serial interface mode capability SM bit reset value is '00b' 00 No Serial Engine, always use parallel signalling. 01 Serial Engine present, always use serial signalling for FS/LS. 10 Software programmable - Reset to use parallel signalling for FS/LS 11 Software programmable - Reset to use serial signalling for FS/LS
8–6 PHYM	Transciever type PHYM bit reset value: '0000b' for OTG controller core, '0100b' for Host-only controller core. 000 UTMI/UMTI+ 001 ULPI DDR 010 ULPI 011 Serial Only 100 Software programmable - reset to UTMI/UTMI+ 101 Software programmable - reset to ULPI DDR 110 Software programmable - reset to ULPI 111 Software programmable - reset to Serial 1000 IC-USB 1001 Software programmable - reset to IC-USB

Table continues on the next page...

USB_*n*HWGENERAL field descriptions (continued)

Field	Description
	1010 HSIC 1011 Software programmable - reset to HSIC
5–4 PHYW	Data width of the transciever connected to the controller core. PHYW bit reset value is PHYW bit reset value is '01b'. 00 8 bit wide data bus Software non-programmable 01 16 bit wide data bus Software non-programmable 10 Reset to 8 bit wide data bus Software programmable 11 Reset to 16 bit wide data bus Software programmable
-	This field is reserved. Reserved

65.6.3 Host Hardware Parameters (USB_*n*HWHOST)

Address: 218_4000h base + 8h offset + (512d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												NPORT			HC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

USB_*n*HWHOST field descriptions

Field	Description
31–4 -	This field is reserved. Reserved
3–1 NPORT	The Nmber of downstream ports supported by the host controller is (NPORT+1). **NOTE:** When these bits value is '000', it indicates a single-port host controller.
0 HC	Host Capable. Indicating whether host operation mode is supported or not. 1 Supported 0 Not supported

65.6.4 Device Hardware Parameters (USB_*n*HWDEVICE)

NOTE

This register is only available in OTG core.

Address: 218_4000h base + Ch offset + (512d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										DEVEP					DC
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

USB_*n*HWDEVICE field descriptions

Field	Description
31–6 -	This field is reserved. Reserved
5–1 DEVEP	Device Endpoint Number
0 DC	Device Capable. Indicating whether device operation mode is supported or not. 1 Supported 0 Not supported

65.6.5 TX Buffer Hardware Parameters (USB_*n*HWTXBUF)

Address: 218_4000h base + 10h offset + (512d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TXCHANADD								Reserved								TXBURST							
Reset	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1	0	0	0

USB_*n*HWTXBUF field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 TXCHANADD	TX FIFO Buffer size is: (2^TXCHANADD) * 4 Bytes. These bits are set to '08h', so buffer size is 256*4 Bytes. For the OTG controller operating in device mode, this is the FIFO buffer size per endpoint. As the OTG controller has 8 TX endpoint, there are 8 of these buffers. For the OTG controller operating in host mode, or for Host-only controller, the entire buffer memory is used as a single TX buffer. Therefore, there is only 1 of this buffer
15–8 -	This field is reserved. Reserved
TXBURST	Default burst size for memory to TX buffer transfer. This is reset value of TXPBURST bits in USB core regsiter USB_n_BURSTSIZE. Please see Programmable Burst Size (USB_*n*BURSTSIZE) .

65.6.6 RX Buffer Hardware Parameters (USB_*n*HWRXBUF)

Address: 218_4000h base + 14h offset + (512d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																RXADD								RXBURST							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

USB_*n*HWRXBUF field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15–8 RXADD	Buffer total size for all receive endpoints is (2^RXADD). RX Buffer size is: (2^RXADD) * 4 Bytes. These bits are set to '08h', so buffer size is 256*4 Bytes. There is a single Receive FIFO buffer in the USB controller. The buffer is shared for all endpoints for the OTG controller in device mode.
RXBURST	Default burst size for memory to RX buffer transfer. This is reset value of RXPBURST bits in USB core regsiter USB_n_BURSTSIZE. Please see Programmable Burst Size (USB_*n*BURSTSIZE) .

65.6.7 General Purpose Timer #0 Load (USB_*n*GPTIMER0LD)

This register controls load value of the count timer in register n_GPTIMER0CTRL. Please see General Purpose Timer #0 Controller (USB_*n*GPTIMER0CTRL) .

Address: 218_4000h base + 80h offset + (512d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								GPTLD																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*GPTIMER0LD field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
GPTLD	General Purpose Timer Load Value These bit fields are loaded to GPTCNT bits when GPTRST bit is set '1b'. This value represents the time in microseconds minus 1 for the timer duration. Example: for a one millisecond timer, load 1000-1=999 or 0x0003E7. **NOTE:** Max value is 0xFFFFFF or 16.777215 seconds.

65.6.8 General Purpose Timer #0 Controller (USB_*n*GPTIMER0CTRL)

This register contains the control for this countdown timer and a data field can be queried to determine the running count value. This timer has granularity on 1 us and can be programmed to a little over 16 seconds. There are two counter modes which are described in the register table below. When the timer counter value transitions to zero, an interrupt could be generated if enable.

Interrupt status bit is TI0 bit in n_USBSTS register (See USB Status Register (USB_*n*USBSTS)), interrupt enable bit is TIE0 bit in n_USBINTR register. (See Interrupt Enable Register (USB_*n*USBINTR) .)

Address: 218_4000h base + 84h offset + (512d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPTRUN	GPTRST	Reserved					GPTMODE	GPTCNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPTCNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*GPTIMER0CTRL field descriptions

Field	Description
31 GPTRUN	General Purpose Timer Run GPTCNT bits are not effected when setting or clearing this bit. 0 Stop counting 1 Run
30 GPTRST	General Purpose Timer Reset 0 No action 1 Load counter value from GPTLD bits in n_GPTIMER0LD
29–25 -	This field is reserved. Reserved
24 GPTMODE	General Purpose Timer Mode In one shot mode, the timer will count down to zero, generate an interrupt, and stop until the counter is reset by software; In repeat mode, the timer will count down to zero, generate an interrupt and automatically reload the counter value from GPTLD bits to start again. 0 One Shot Mode 1 Repeat Mode
GPTCNT	General Purpose Timer Counter. This field is the count value of the countdown timer.

65.6.9 General Purpose Timer #1 Load (USB_*n*GPTIMER1LD)

This register controls load value of the count timer in register n_GPTIMER1CTRL. Please see General Purpose Timer #1 Controller (USB_*n*GPTIMER1CTRL) .

Address: 218_4000h base + 88h offset + (512d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								GPTLD																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*GPTIMER1LD field descriptions

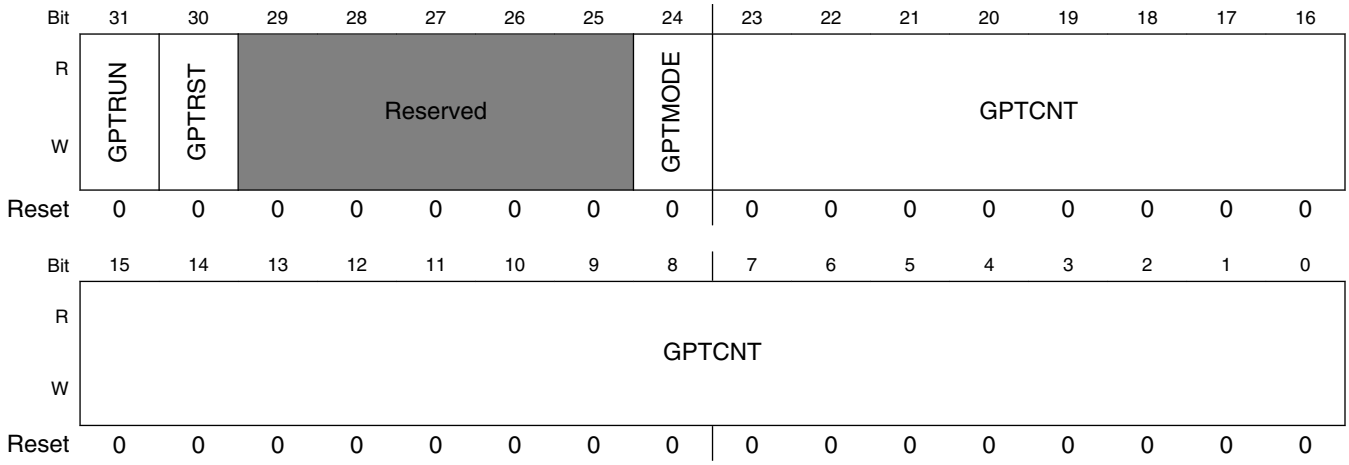
Field	Description
31–24 -	This field is reserved. Reserved
GPTLD	General Purpose Timer Load Value These bit fields are loaded to GPTCNT bits when GPTRST bit is set '1b'. This value represents the time in microseconds minus 1 for the timer duration. Example: for a one millisecond timer, load 1000-1=999 or 0x0003E7. **NOTE:** Max value is 0xFFFFFF or 16.777215 seconds.

65.6.10 General Purpose Timer #1 Controller (USB_*n*GPTIMER1CTRL)

This register contains the control for this countdown timer and a data field can be queried to determine the running count value. This timer has granularity on 1 us and can be programmed to a little over 16 seconds. There are two counter modes which are described in the register table below. When the timer counter value transitions to zero, an interrupt could be generated if enable.

Interrupt status bit is TI1 bit in USB_n_USBSTS register (See USB Status Register (USB_*n*USBSTS)), interrupt enable bit is TIE1 bit in n_USBINTR register (See Interrupt Enable Register (USB_*n*USBINTR)).

Address: 218_4000h base + 8Ch offset + (512d × i), where i=0d to 3d

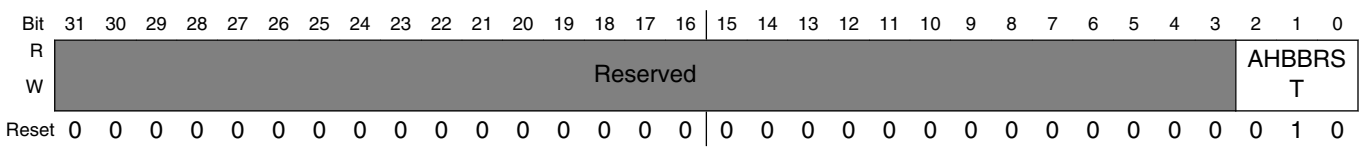


USB_*n*GPTIMER1CTRL field descriptions

Field	Description
31 GPTRUN	General Purpose Timer Run GPTCNT bits are not effected when setting or clearing this bit. 0 Stop counting 1 Run
30 GPTRST	General Purpose Timer Reset 0 No action 1 Load counter value from GPTLD bits in USB_n_GPTIMER0LD
29–25 -	This field is reserved. Reserved
24 GPTMODE	General Purpose Timer Mode In one shot mode, the timer will count down to zero, generate an interrupt, and stop until the counter is reset by software. In repeat mode, the timer will count down to zero, generate an interrupt and automatically reload the counter value from GPTLD bits to start again. 0 One Shot Mode 1 Repeat Mode
GPTCNT	General Purpose Timer Counter. This field is the count value of the countdown timer.

65.6.11 System Bus Config (USB_*n*SBUSCFG)

Address: 218_4000h base + 90h offset + (512d × i), where i=0d to 3d



USB_*n*SBUSCFG field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
AHBBRST	AHB master interface Burst configuration These bits control AHB master transfer type sequence (or priority). **NOTE:** This register overrides n_BURSTSIZE register when its value is not zero. 000 Incremental burst of unspecified length only 001 INCR4 burst, then single transfer 010 INCR8 burst, INCR4 burst, then single transfer 011 INCR16 burst, INCR8 burst, INCR4 burst, then single transfer 100 Reserved, don't use 101 INCR4 burst, then incremental burst of unspecified length 110 INCR8 burst, INCR4 burst, then incremental burst of unspecified length 111 INCR16 burst, INCR8 burst, INCR4 burst, then incremental burst of unspecified length

65.6.12 Capability Registers Length (USB_*n*CAPLENGTH)

The Capability Registers Length register contains the address offset to the Operational registers relative to the CAPLENGTH register.

Address: 218_4000h base + 100h offset + (512d × i), where i=0d to 3d

Bit	7	6	5	4	3	2	1	0
Read	CAPLENGTH							
Write								
Reset	0	1	0	0	0	0	0	0

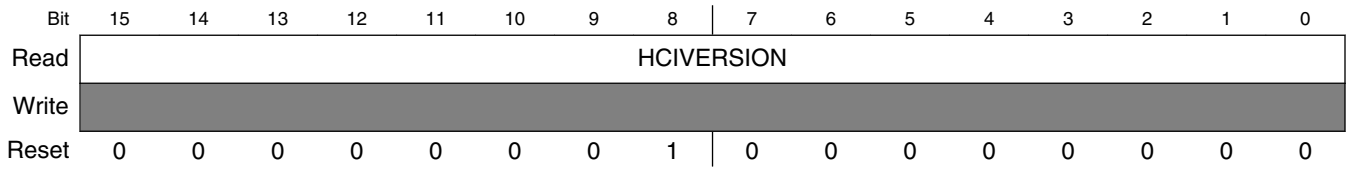
USB_*n*CAPLENGTH field descriptions

Field	Description
CAPLENGTH	These bits are used as an offset to add to register base to find the beginning of the Operational Register. Default value is '40h'.

65.6.13 Host Controller Interface Version (USB_*n*HCIVERSION)

This is a 2-byte register containing a BCD encoding of the EHCI revision number supported by this host controller. The most significant byte of this register represents a major revision and the least significant byte is the minor revision.

Address: 218_4000h base + 102h offset + (512d × i), where i=0d to 3d



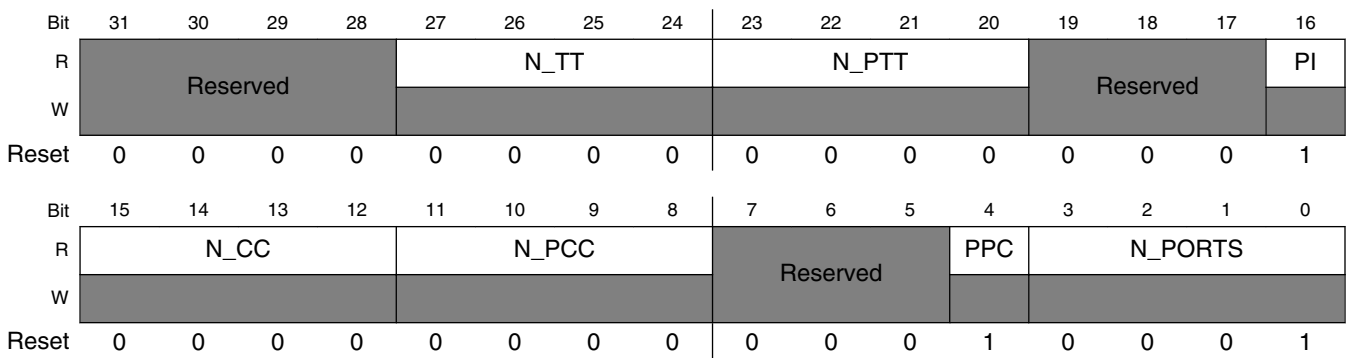
USB_*n*HCIVERSION field descriptions

Field	Description
HCIVERSION	Host Controller Interface Version Number Default value is '10h', which means EHCI rev1.0.

65.6.14 Host Controller Structural Parameters (USB_*n*HCSPARAMS)

The following figure shows the port steering logic capabilities of Host Control Structural Parameters (n_HCSPARAMS).

Address: 218_4000h base + 104h offset + (512d × i), where i=0d to 3d



USB_*n*HCSPARAMS field descriptions

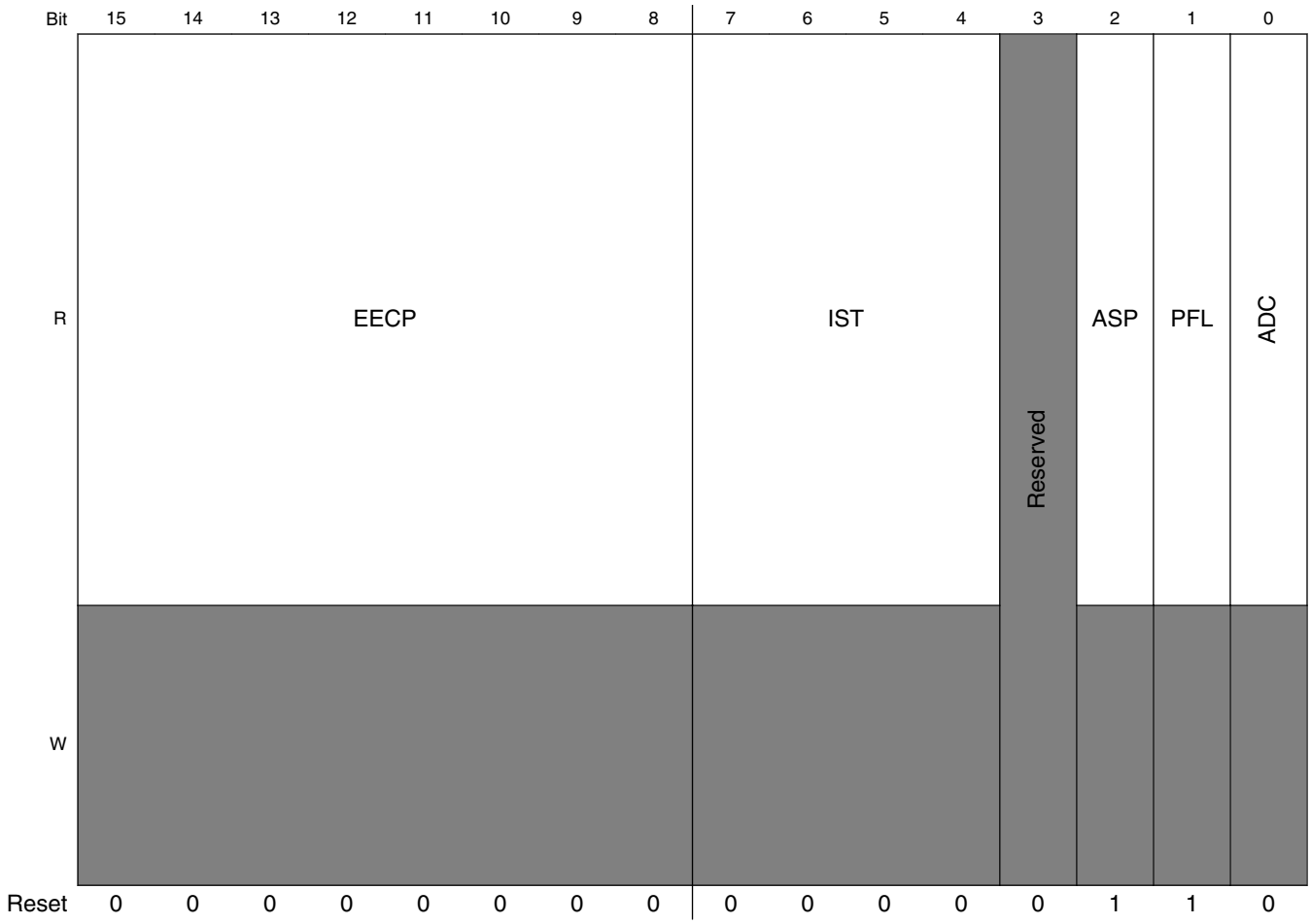
Field	Description
31–28 -	This field is reserved. Reserved
27–24 N_TT	Number of Transaction Translators (N_TT). Default value '0000b' This field indicates the number of embedded transaction translators associated with the USB2.0 host controller. These bits would be set to '0001b' for Multi-Port Host, and '0000b' for Single-Port Host.
23–20 N_PTT	Number of Ports per Transaction Translator (N_PTT). Default value '0000b' This field indicates the number of ports assigned to each transaction translator within the USB2.0 host controller. These bits would be set to equal N_PORTS for Multi-Port Host, and '0000b' for Single-Port Host.
19–17 -	This field is reserved. Reserved
16 PI	Port Indicators (P INDICATOR) This bit indicates whether the ports support port indicator control. When set to one, the port status and control registers include a read/writeable field for controlling the state of the port indicator This bit is "1b" in all controller core.
15–12 N_CC	Number of Companion Controller (N_CC). This field indicates the number of companion controllers associated with this USB2.0 host controller. These bits are '0000b' in all controller core. 0 There is no internal Companion Controller and port-ownership hand-off is not supported. 1 There are internal companion controller(s) and port-ownership hand-offs is supported.
11–8 N_PCC	Number of Ports per Companion Controller This field indicates the number of ports supported per internal Companion Controller. It is used to indicate the port routing configuration to the system software. For example, if N_PORTS has a value of 6 and N_CC has a value of 2 then N_PCC could have a value of 3. The convention is that the first N_PCC ports are assumed to be routed to companion controller 1, the next N_PCC ports to companion controller 2, etc. In the previous example, the N_PCC could have been 4, where the first 4 are routed to companion controller 1 and the last two are routed to companion controller 2. The number in this field must be consistent with N_PORTS and N_CC. These bits are '0000b' in all controller core.
7–5 -	This field is reserved. Reserved
4 PPC	Port Power Control This field indicates whether the host controller implementation includes port power control. A one indicates the ports have port power switches. A zero indicates the ports do not have port power switches. The value of this field affects the functionality of the Port Power field in each port status and control register
N_PORTS	Number of downstream ports. This field specifies the number of physical downstream ports implemented on this host controller. The value of this field determines how many port registers are addressable in the Operational Register. Valid values are in the range of 1h to Fh. A zero in this field is undefined. These bits are always set to '0001b' because all controller cores are Single-Port Host.

65.6.15 Host Controller Capability Parameters (USB_*n*HCCPARAMS)

This register identifies multiple mode control (time-base bit functionality), addressing capability.

Address: 218_4000h base + 108h offset + (512d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



USB_*n*HCCPARAMS field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15–8 EECP	EHCI Extended Capabilities Pointer. This field indicates the existence of a capabilities list. A value of 00h indicates no extended capabilities are implemented. A non-zero value in this register indicates the offset in PCI configuration space of the first EHCI extended capability. The pointer value must be 40h or greater if implemented to maintain the consistency of the PCI header defined for this class of device. **NOTE:** These bits are set '00h' in all controller core.
7–4 IST	Isochronous Scheduling Threshold. This field indicates, relative to the current position of the executing host controller, where software can reliably update the isochronous schedule. When bit [7] is zero, the value of the least significant 3 bits indicates the number of micro-frames a host controller can hold a set of isochronous data structures (one or more) before flushing the state. When bit [7] is a one, then host software assumes the host controller may cache an isochronous data structure for an entire frame. These bits are set '00h' in all controller core.
3 -	This field is reserved. Reserved

Table continues on the next page...

USB_*n*HCCPARAMS field descriptions (continued)

Field	Description
2 ASP	Asynchronous Schedule Park Capability If this bit is set to a one, then the host controller supports the park feature for high-speed queue heads in the Asynchronous Schedule. The feature can be disabled or enabled and set to a specific level by using the *Asynchronous Schedule Park Mode Enable* and *Asynchronous Schedule Park Mode Count* fields in the USBCMD register. **NOTE:** ASP bit reset value: '00b' for OTG controller core, '11b' for Host-only controller core.
1 PFL	Programmable Frame List Flag If this bit is set to zero, then the system software must use a frame list length of 1024 elements with this host controller. The USBCMD register Frame List Size field is a read-only register and must be set to zero. If set to a one, then the system software can specify and use a smaller frame list and configure the host controller via the USBCMD register Frame List Size field. The frame list must always be aligned on a 4K-page boundary. This requirement ensures that the frame list is always physically contiguous. This bit is set '1b' in all controller core.
0 ADC	64-bit Addressing Capability This bit is set '0b' in all controller core, no 64-bit addressing capability is supported.

65.6.16 Device Controller Interface Version (USB_*n*DCIVERSION)

This register indicates the two-byte BCD encoding of the device controller interface version number.

Address: 218_4000h base + 120h offset + (512d × i), where i=0d to 0d

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	DCIVERSION															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

USB_*n*DCIVERSION field descriptions

Field	Description
DCIVERSION	Device Controller Interface Version Number Default value is '01h', which means rev0.1.

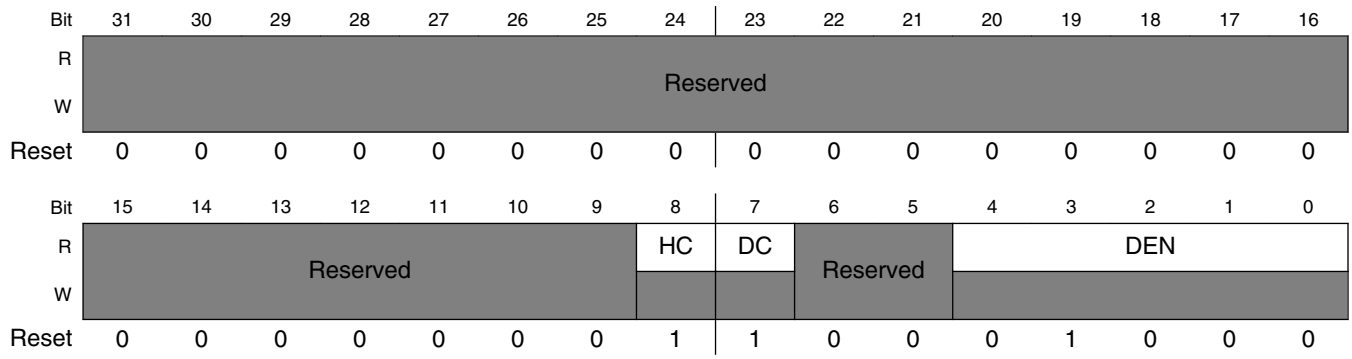
65.6.17 Device Controller Capability Parameters (USB_*n*DCCPARAMS)

These fields describe the overall device capability of the controller.

NOTE

This register is only available in OTG controller core.

Address: 218_4000h base + 124h offset + (512d × i), where i=0d to 0d



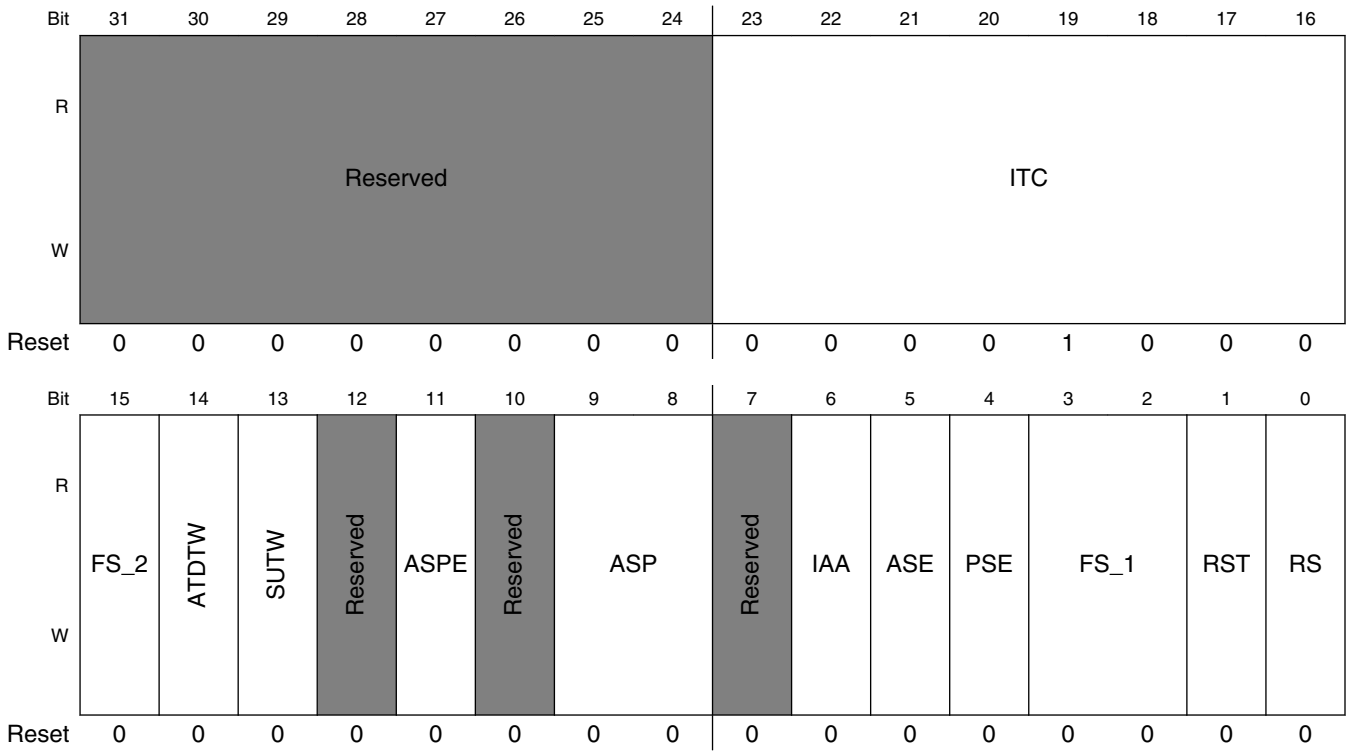
USB_*n*DCCPARAMS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 HC	Host Capable When this bit is 1, this controller is capable of operating as an EHCI compatible USB 2.0 host controller.
7 DC	Device Capable When this bit is 1, this controller is capable of operating as a USB 2.0 device.
6–5 -	This field is reserved. Reserved
DEN	Device Endpoint Number This field indicates the number of endpoints built into the device controller. If this controller is not device capable, then this field will be zero. Valid values are 0 - 15.

65.6.18 USB Command Register (USB_*n*USBCMD)

The Command Register indicates the command to be executed by the serial bus host/device controller. Writing to the register causes a command to be executed.

Address: 218_4000h base + 140h offset + (512d × i), where i=0d to 3d



USB_*n*USBCMD field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 ITC	Interrupt Threshold Control -Read/Write. The system software uses this field to set the maximum rate at which the host/device controller will issue interrupts. ITC contains the maximum interrupt interval measured in micro-frames. Valid values are shown below. Value Maximum Interrupt Interval 0x00 Immediate (no threshold) 0x01 1 micro-frame 0x02 2 micro-frames 0x04 4 micro-frames 0x08 8 micro-frames 0x10 16 micro-frames

Table continues on the next page...

USB_*n*USBCMD field descriptions (continued)

Field	Description
	0x20 32 micro-frames 0x40 64 micro-frames
15 FS_2	Frame List Size - (Read/Write or Read Only). [host mode only] This field is Read/Write only if Programmable Frame List Flag in the HCCPARAMS registers is set to one. This field specifies the size of the frame list that controls which bits in the Frame Index Register should be used for the Frame List Current index. **NOTE:** This field is made up from USBCMD bits 15, 3 and 2. Value Meaning 000 1024 elements (4096 bytes) Default value 001 512 elements (2048 bytes) 010 256 elements (1024 bytes) 011 128 elements (512 bytes) 100 64 elements (256 bytes) 101 32 elements (128 bytes) 110 16 elements (64 bytes) 111 8 elements (32 bytes)
14 ATDTW	Add dTD TripWire - Read/Write. [device mode only] This bit is used as a semaphore to ensure proper addition of a new dTD to an active (primed) endpoint's linked list. This bit is set and cleared by software. This bit would also be cleared by hardware when state machine is hazard region for which adding a dTD to a primed endpoint may go unrecognized.
13 SUTW	Setup TripWire - Read/Write. [device mode only] This bit is used as a semaphore to ensure that the setup data payload of 8 bytes is extracted from a QH by the DCD without being corrupted. If the setup lockout mode is off (SLOM bit in USB core register n_USBMODE, see USB Device Mode (USB_*n*USBMODE)) then there is a hazard when new setup data arrives while the DCD is copying the setup data payload from the QH for a previous setup packet. This bit is set and cleared by software. This bit would also be cleared by hardware when a hazard detected.
12 -	This field is reserved. Reserved
11 ASPE	Asynchronous Schedule Park Mode Enable - Read/Write. If the *Asynchronous Park Capability* bit in the HCCPARAMS register is a one, then this bit defaults to a 1h and is R/W. Otherwise the bit must be a zero and is RO. Software uses this bit to enable or disable Park mode. When this bit is one, Park mode is enabled. When this bit is a zero, Park mode is disabled. **NOTE:** ASPE bit reset value: '0b' for OTG controller core, '1b' for Host-only controller core.
10 -	This field is reserved. Reserved
9–8 ASP	Asynchronous Schedule Park Mode Count - Read/Write. If the *Asynchronous Park Capability* bit in the HCCPARAMS register is a one, then this field defaults to 3h and is R/W. Otherwise it defaults to zero and is Read-Only. It contains a count of the number of successive transactions the host controller is allowed to execute from a high-speed queue head on the

Table continues on the next page...

USB_*n*USBCMD field descriptions (continued)

Field	Description
	Asynchronous schedule before continuing traversal of the Asynchronous schedule. Valid values are 1h to 3h. Software must not write a zero to this bit when *Park Mode Enable* is a one as this will result in undefined behavior. This field is set to 3h in all controller core.
7 -	This field is reserved. Reserved
6 IAA	Interrupt on Async Advance Doorbell - Read/Write. This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to ring the doorbell. When the host controller has evicted all appropriate cached schedule states, it sets the Interrupt on Async Advance status bit in the USBSTS register. If the Interrupt on Sync Advance Enable bit in the USBINTR register is one, then the host controller will assert an interrupt at the next interrupt threshold. The host controller sets this bit to zero after it has set the Interrupt on Sync Advance status bit in the USBSTS register to one. Software should not write a one to this bit when the asynchronous schedule is inactive. Doing so will yield undefined results. This bit is only used in host mode. Writing a one to this bit when device mode is selected will have undefined results.
5 ASE	Asynchronous Schedule Enable - Read/Write. Default 0b. This bit controls whether the host controller skips processing the Asynchronous Schedule. Only the host controller uses this bit. Values Meaning 0 Do not process the Asynchronous Schedule. 1 Use the ASYNCLISTADDR register to access the Asynchronous Schedule.
4 PSE	Periodic Schedule Enable- Read/Write. Default 0b. This bit controls whether the host controller skips processing the Periodic Schedule. Only the host controller uses this bit. Values Meaning 0 Do not process the Periodic Schedule 1 Use the PERIODICLISTBASE register to access the Periodic Schedule.
3–2 FS_1	See description at bit 15
1 RST	Controller Reset (RESET) - Read/Write. Software uses this bit to reset the controller. This bit is set to zero by the Host/Device Controller when the reset process is complete. Software cannot terminate the reset process early by writing a zero to this register. Host operation mode: When software writes a one to this bit, the Controller resets its internal pipelines, timers, counters, state machines etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports. Software should not set this bit to a one when the HCHalted bit in the USBSTS register is a zero. Attempting to reset an actively running host controller will result in undefined behavior. Device operation mode:

Table continues on the next page...

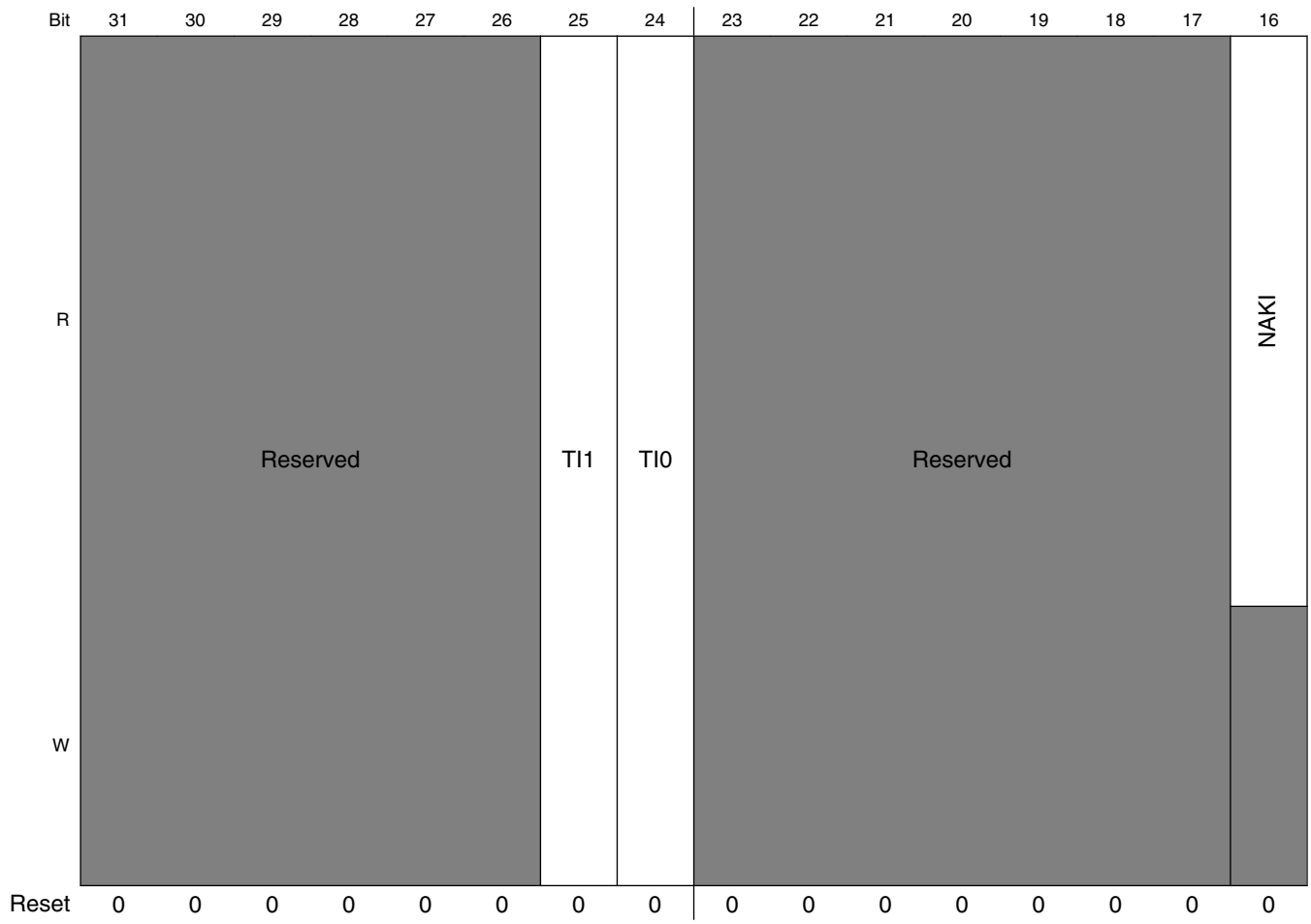
USB_*n*USBCMD field descriptions (continued)

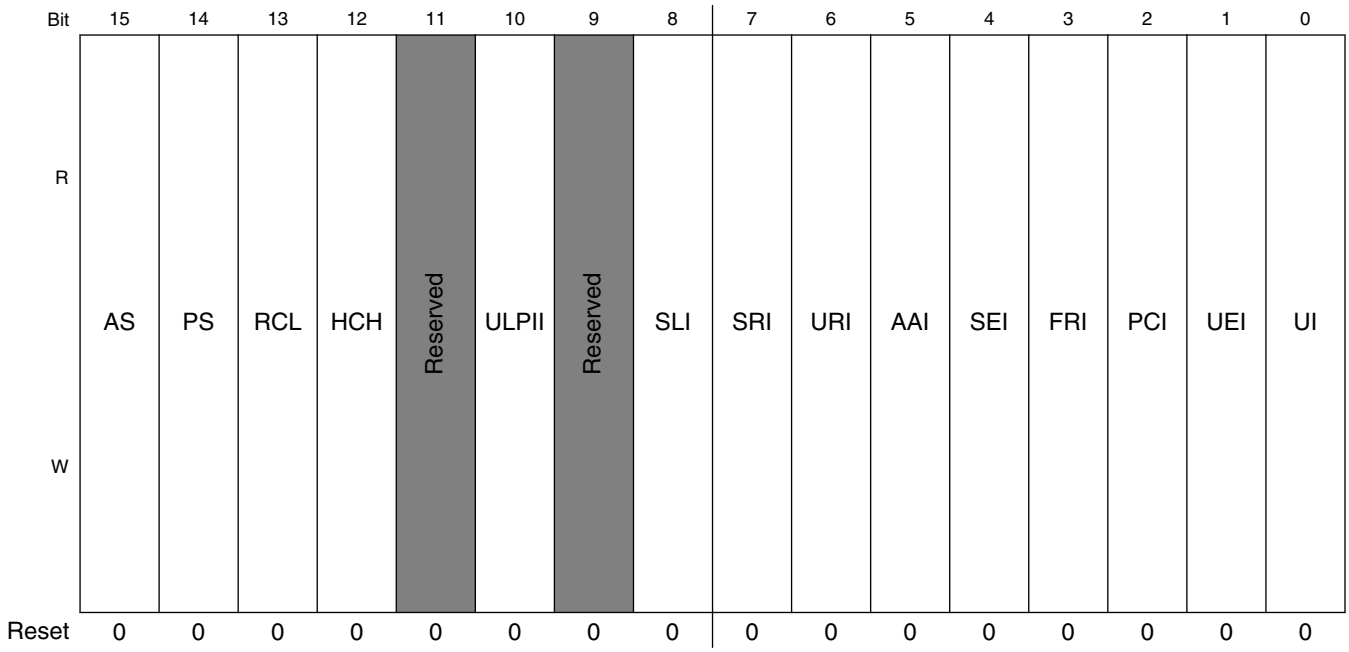
Field	Description
	When software writes a one to this bit, the Controller resets its internal pipelines, timers, counters, state machines etc. to their initial value. Writing a one to this bit when the device is in the attached state is not recommended, because the effect on an attached host is undefined. In order to ensure that the device is not in an attached state before initiating a device controller reset, all primed endpoints should be flushed and the USBCMD Run/Stop bit should be set to 0.
0 RS	Run/Stop (RS) - Read/Write. Default 0b. 1=Run. 0=Stop. Host operation mode: When set to '1b', the Controller proceeds with the execution of the schedule. The Controller continues execution as long as this bit is set to a one. When this bit is set to 0, the Host Controller completes the current transaction on the USB and then halts. The HC Halted bit in the status register indicates when the Controller has finished the transaction and has entered the stopped state. Software should not write a one to this field unless the controller is in the Halted state (that is, HCHalted in the USBSTS register is a one). Device operation mode: Writing a one to this bit will cause the controller to enable a pull-up on D+ and initiate an attach event. This control bit is not directly connected to the pull-up enable, as the pull-up will become disabled upon transitioning into high-speed mode. Software should use this bit to prevent an attach event before the controller has been properly initialized. Writing a 0 to this will cause a detach event.

65.6.19 USB Status Register (USB_*n*USBSTS)

This register indicates various states of the Host/Device Controller and any pending interrupts. This register does not indicate status resulting from a transaction on the serial bus.

Address: 218_4000h base + 144h offset + (512d × i), where i=0d to 3d





USB_*n*USBSTS field descriptions

Field	Description
31–26 -	This field is reserved. Reserved
25 TI1	General Purpose Timer Interrupt 1(GPTINT1)--R/WC. This bit is set when the counter in the GPTIMER1CTRL register transitions to zero, writing a one to this bit will clear it.
24 TI0	General Purpose Timer Interrupt 0(GPTINT0)--R/WC. This bit is set when the counter in the GPTIMER0CTRL register transitions to zero, writing a one to this bit clears it.
23–17 -	This field is reserved. Reserved
16 NAKI	NAK Interrupt Bit--RO. This bit is set by hardware when for a particular endpoint both the TX/RX Endpoint NAK bit and corresponding TX/RX Endpoint NAK Enable bit are set. This bit is automatically cleared by hardware when all Enabled TX/RX Endpoint NAK bits are cleared.
15 AS	Asynchronous Schedule Status - Read Only. This bit reports the current real status of the Asynchronous Schedule. When set to zero the asynchronous schedule status is disabled and if set to one the status is enabled. The Host Controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions the Asynchronous Schedule Enable bit in the USBCMD register. When this bit and the Asynchronous Schedule Enable bit are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0). Only used in the host operation mode.
14 PS	Periodic Schedule Status - Read Only.

Table continues on the next page...

USB_*n*USBSTS field descriptions (continued)

Field	Description
	This bit reports the current real status of the Periodic Schedule. When set to zero the periodic schedule is disabled, and if set to one the status is enabled. The Host Controller is not required to immediately disable or enable the Periodic Schedule when software transitions the Periodic Schedule Enable bit in the USBCMD register. When this bit and the Periodic Schedule Enable bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0). Only used in the host operation mode.
13 RCL	Reclamation - Read Only. This is a read-only status bit used to detect an empty asynchronous schedule. Only used in the host operation mode.
12 HCH	HCHalted - Read Only. This bit is a zero whenever the Run/Stop bit is a one. The Controller sets this bit to one after it has stopped executing because of the Run/Stop bit being set to 0, either by software or by the Controller hardware (for example, an internal error). Only used in the host operation mode. Default value is '0b' for OTG core, and '1b' for Host1/Host2/Host3 core. This is because OTG core is not operating as host in default. Please see CM bit in USB_n_USBMODE register. **NOTE:** HCH bit reset value: '0b' for OTG controller core, '1b' for Host-only controller core.
11 -	This field is reserved. Reserved
10 ULPII	ULPI Interrupt - R/WC. This bit will be set '1b' by hardware when there is an event completion in ULPI viewport. This bit is usable only if the controller support UPLI interface mode.
9 -	This field is reserved. Reserved
8 SLI	DCSuspend - R/WC. When a controller enters a suspend state from an active state, this bit will be set to a one. The device controller clears the bit upon exiting from a suspend state. Only used in device operation mode.
7 SRI	SOF Received - R/WC. When the device controller detects a Start Of (micro) Frame, this bit will be set to a one. When a SOF is extremely late, the device controller will automatically set this bit to indicate that an SOF was expected. Therefore, this bit will be set roughly every 1ms in device FS mode and every 125ms in HS mode and will be synchronized to the actual SOF that is received. Because the device controller is initialized to FS before connect, this bit will be set at an interval of 1ms during the prelude to connect and chirp. In host mode, this bit will be set every 125us and can be used by host controller driver as a time base. Software writes a 1 to this bit to clear it.
6 URI	USB Reset Received - R/WC. When the device controller detects a USB Reset and enters the default state, this bit will be set to a one. Software can write a 1 to this bit to clear the USB Reset Received status bit. Only used in device operation mode.

Table continues on the next page...

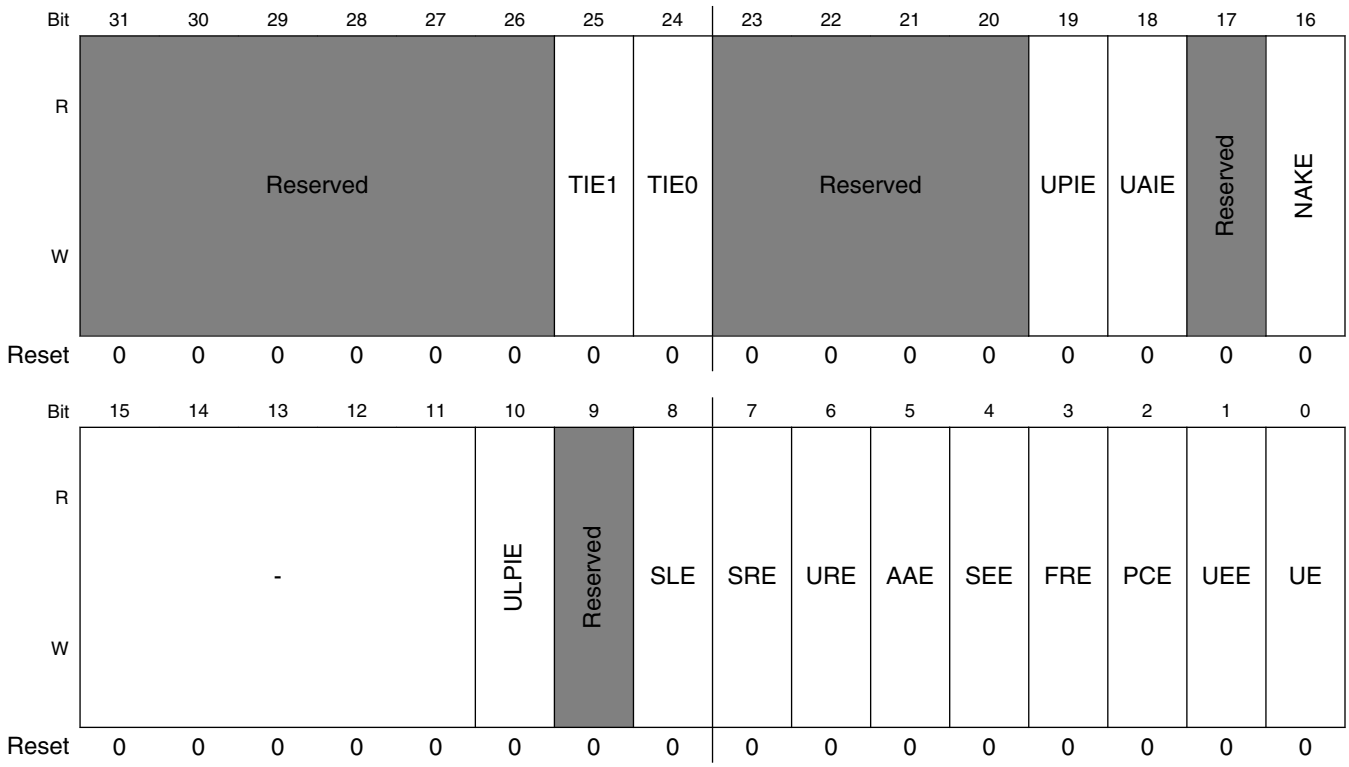
USB_*n*USBSTS field descriptions (continued)

Field	Description
5 AAI	Interrupt on Async Advance - R/WC. System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a one to the Interrupt on Async Advance Doorbell bit in the n_USBCMD register. This status bit indicates the assertion of that interrupt source. Only used in host operation mode.
4 SEI	System Error- R/WC. This bit is will be set to '1b' when an Error response is seen to a read on the system interface.
3 FRI	Frame List Rollover - R/WC. The Host Controller sets this bit to a one when the Frame List Index rolls over from its maximum value to zero. The exact value at which the rollover occurs depends on the frame list size. For example. If the frame list size (as programmed in the Frame List Size field of the USB_n_USBCMD register) is 1024, the Frame Index Register rolls over every time FRINDEX [13] toggles. Similarly, if the size is 512, the Host Controller sets this bit to a one every time FHINDEX [12] toggles. Only used in host operation mode.
2 PCI	Port Change Detect - R/WC. The Host Controller sets this bit to a one when on any port a Connect Status occurs, a Port Enable/ Disable Change occurs, or the Force Port Resume bit is set as the result of a J-K transition on the suspended port. The Device Controller sets this bit to a one when the port controller enters the full or high-speed operational state. When the port controller exits the full or high-speed operation states due to Reset or Suspend events, the notification mechanisms are the USB Reset Received bit and the DCSuspend bits respectively.
1 UEI	USB Error Interrupt (USBERRINT) - R/WC. When completion of a USB transaction results in an error condition, this bit is set by the Host/Device Controller. This bit is set along with the USBINT bit, if the TD on which the error interrupt occurred also had its interrupt on complete (IOC) bit set. The device controller detects resume signaling only.
0 UI	USB Interrupt (USBINT) - R/WC. This bit is set by the Host/Device Controller when the cause of an interrupt is a completion of a USB transaction where the Transfer Descriptor (TD) has an interrupt on complete (IOC) bit set. This bit is also set by the Host/Device Controller when a short packet is detected. A short packet is when the actual number of bytes received was less than the expected number of bytes.

65.6.20 Interrupt Enable Register (USB_*n*USBINTR)

The interrupts to software are enabled with this register. An interrupt is generated when a bit is set and the corresponding interrupt source is active. The USB Status register (n_USBSTS) still shows interrupt sources even if they are disabled by the n_USBINTR register, allowing polling of interrupt events by the software.

Address: 218_4000h base + 148h offset + (512d × i), where i=0d to 3d



USB_*n*USBINTR field descriptions

Field	Description
31–26 -	This field is reserved. Reserved
25 TIE1	General Purpose Timer #1 Interrupt Enable When this bit is one and the TI1 bit in n_USBSTS register is a one the controller will issue an interrupt.
24 TIE0	General Purpose Timer #0 Interrupt Enable When this bit is one and the TI0 bit in n_USBSTS register is a one the controller will issue an interrupt.
23–20 -	This field is reserved. Reserved
19 UPIE	USB Host Periodic Interrupt Enable

Table continues on the next page...

USB_*n*USBINTR field descriptions (continued)

Field	Description
	When this bit is one, and the UPI bit in the n_USBSTS register is one, host controller will issue an interrupt at the next interrupt threshold.
18 UAIE	USB Host Asynchronous Interrupt Enable When this bit is one, and the UAI bit in the n_USBSTS register is one, host controller will issue an interrupt at the next interrupt threshold.
17 -	This field is reserved. Reserved
16 NAKE	NAK Interrupt Enable When this bit is one and the NAKI bit in n_USBSTS register is a one the controller will issue an interrupt.
15–11 -	These bits are reserved and should be set to zero.
10 ULPIE	ULPI Interrupt Enable When this bit is one and the UPLII bit in n_USBSTS register is a one the controller will issue an interrupt. This bit is usable only if the controller support UPLI interface mode.
9 -	This field is reserved. Reserved
8 SLE	Sleep Interrupt Enable When this bit is one and the SLI bit in n_n_USBSTS register is a one the controller will issue an interrupt. Only used in device operation mode.
7 SRE	SOF Received Interrupt Enable When this bit is one and the SRI bit in n_USBSTS register is a one the controller will issue an interrupt.
6 URE	USB Reset Interrupt Enable When this bit is one and the URI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in device operation mode.
5 AAE	Async Advance Interrupt Enable When this bit is one and the AAI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in host operation mode.
4 SEE	System Error Interrupt Enable When this bit is one and the SEI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in host operation mode.
3 FRE	Frame List Rollover Interrupt Enable When this bit is one and the FRI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in host operation mode.
2 PCE	Port Change Detect Interrupt Enable When this bit is one and the PCI bit in n_USBSTS register is a one the controller will issue an interrupt.
1 UEE	USB Error Interrupt Enable When this bit is one and the UEI bit in n_USBSTS register is a one the controller will issue an interrupt.
0 UE	USB Interrupt Enable When this bit is one and the UI bit in n_USBSTS register is a one the controller will issue an interrupt.

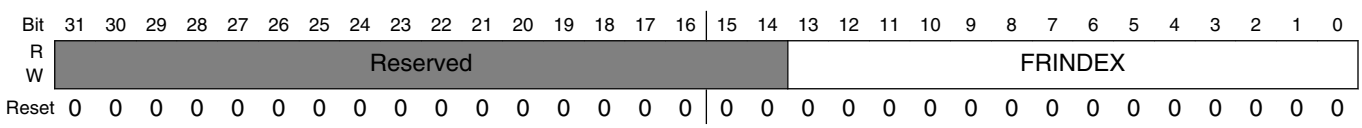
65.6.21 USB Frame Index (USB_*n*FRINDEX)

This register is used by the host controller to index the periodic frame list. The register updates every 125 microseconds (once each micro-frame). Bits [N: 3] are used to select a particular entry in the Periodic Frame List during periodic schedule execution. The number of bits used for the index depends on the size of the frame list as set by system software in the Frame List Size field in the n_USBCMD register.

This register must be written as a DWord. Byte writes produce undefined results. This register cannot be written unless the Host Controller is in the 'Halted' state as indicated by the HCHalted bit. A write to this register while the Run/Stop hit is set to a one produces undefined results. Writes to this register also affect the SOF value.

In device mode this register is read only and, the device controller updates the FRINDEX [13:3] register from the frame number indicated by the SOF marker. Whenever a SOF is received by the USB bus, FRINDEX [13:3] will be checked against the SOF marker. If FRINDEX [13:3] is different from the SOF marker, FRINDEX [13:3] will be set to the SOF value and FRINDEX [2:0] will be set to zero (that is, SOF for 1 ms frame). If FRINDEX [13:3] is equal to the SOF value, FRINDEX [2:0] will be increment (that is, SOF for 125 us micro-frame.).

Address: 218_4000h base + 14Ch offset + (512d × i), where i=0d to 3d



USB_*n*FRINDEX field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
FRINDEX	Frame Index. The value, in this register, increments at the end of each time frame (micro-frame). Bits [N: 3] are used for the Frame List current index. This means that each location of the frame list is accessed 8 times (frames or micro-frames) before moving to the next index. The following illustrates values of N based on the value of the Frame List Size field in the USBCMD register, when used in host mode. USBCMD [Frame List Size] Number Elements N In device mode the value is the current frame number of the last frame transmitted. It is not used as an index. In either mode bits 2:0 indicate the current microframe. 000 (1024) 12 001 (512) 11

Table continues on the next page...

USB_*n*FRINDEX field descriptions (continued)

Field	Description
	010 (256) 10 011 (128) 9 100 (64) 8 101 (32) 7 110 (16) 6 111 (8) 5

65.6.22 Frame List Base Address (USB_*n*PERIODICLISTBASE)

Host Controller only

This 32-bit register contains the beginning address of the Periodic Frame List in the system memory. HCD loads this register prior to starting the schedule execution by the Host Controller. The memory structure referenced by this physical memory pointer is assumed to be 4-Kbyte aligned. The contents of this register are combined with the Frame Index Register (USB_n_FRINDEX) to enable the Host Controller to step through the Periodic Frame List in sequence.

Address: 218_4000h base + 154h offset + (512d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASEADR																Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*PERIODICLISTBASE field descriptions

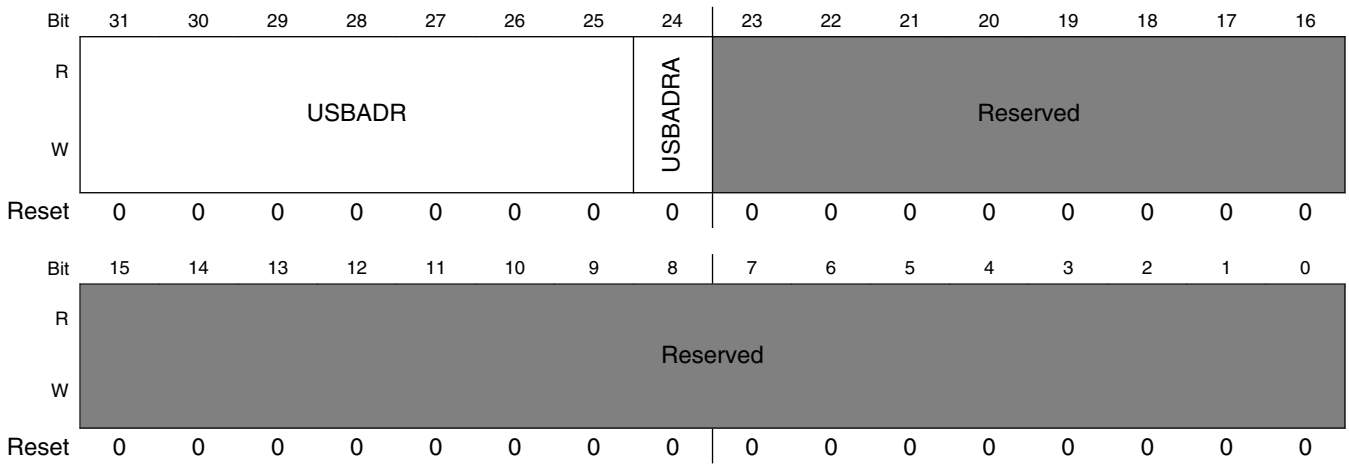
Field	Description
31–12 BASEADR	Base Address (Low). These bits correspond to memory address signals [31:12], respectively. Only used by the host controller.
-	This field is reserved. Reserved

65.6.23 Device Address (USB_*n*DEVICEADDR)

Device Controller only

The upper seven bits of this register represent the device address. After any controller reset or a USB reset, the device address is set to the default address (0). The default address will match all incoming addresses. Software shall reprogram the address after receiving a SET_ADDRESS descriptor.

Address: 218_4000h base + 154h offset + (512d × i), where i=0d to 0d



USB_*n*DEVICEADDR field descriptions

Field	Description
31–25 USBADR	Device Address. These bits correspond to the USB device address
24 USBADRA	Device Address Advance. Default=0. When this bit is '0', any writes to USBADR are instantaneous. When this bit is written to a '1' at the same time or before USBADR is written, the write to the USBADR field is staged and held in a hidden register. After an IN occurs on endpoint 0 and is ACKed, USBADR will be loaded from the holding register. Hardware will automatically clear this bit on the following conditions: 1) IN is ACKed to endpoint 0. (USBADR is updated from staging register). 2) OUT/SETUP occur to endpoint 0. (USBADR is not updated). 3) Device Reset occurs (USBADR is reset to 0). **NOTE:** After the status phase of the SET_ADDRESS descriptor, the DCD has 2 ms to program the USBADR field. This mechanism will ensure this specification is met when the DCD can not write of the device address within 2ms from the SET_ADDRESS status phase. If the DCD writes the USBADR with USBADRA=1 after the SET_ADDRESS data phase (before the prime of the status phase), the USBADR will be programmed instantly at the correct time and meet the 2ms USB requirement.
-	This field is reserved. Reserved

65.6.24 Next Asynch. Address (USB_*n*ASYNCLISTADDR)

Host Controller only

This 32-bit register contains the address of the next asynchronous queue head to be executed by the host. Bits [4:0] of this register cannot be modified by the system software and will always return a zero when read.

Address: 218_4000h base + 158h offset + (512d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ASYBASE																Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*ASYNCLISTADDR field descriptions

Field	Description
31–5 ASYBASE	Link Pointer Low (LPL). These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (QH). Only used by the host controller.
-	This field is reserved. Reserved

65.6.25 Endpoint List Address (USB_*n*ENDPTLISTADDR)

Device Controller only

In device mode, this register contains the address of the top of the endpoint list in system memory. Bits [10:0] of this register cannot be modified by the system software and will always return a zero when read.

The memory structure referenced by this physical memory pointer is assumed 64-byte.

Address: 218_4000h base + 158h offset + (512d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EPBASE																					Reserved										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

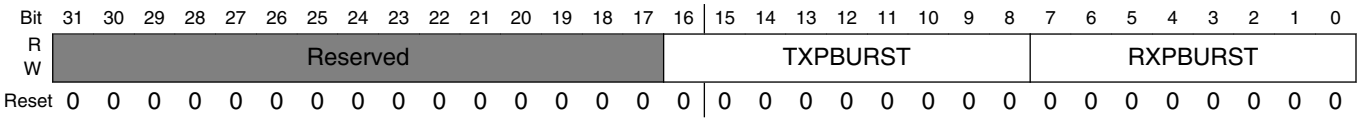
USB_*n*ENDPTLISTADDR field descriptions

Field	Description
31–11 EPBASE	Endpoint List Pointer(Low). These bits correspond to memory address signals [31:11], respectively. This field will reference a list of up to 32 Queue Head (QH) (that is, one queue head per endpoint & direction).
-	This field is reserved. Reserved

65.6.26 Programmable Burst Size (USB_*n*BURSTSIZE)

This register is used to control the burst size used during data movement on the AHB master interface. This register is ignored if AHBBRST bits in SBUSCFG register is non-zero value.

Address: 218_4000h base + 160h offset + (512d × i), where i=0d to 3d



USB_*n*BURSTSIZE field descriptions

Field	Description
31–17 -	This field is reserved. Reserved
16–8 TXPBURST	Programmable TX Burst Size. Default value is determined by TXBURST bits in n_HWTXBUF. This register represents the maximum length of a the burst in 32-bit words while moving data from system memory to the USB bus.
RXPBURST	Programmable RX Burst Size. Default value is determined by TXBURST bits in n_HWRXBUF. This register represents the maximum length of a the burst in 32-bit words while moving data from the USB bus to system memory.

65.6.27 TX FIFO Fill Tuning (USB_*n*TXFILLTUNING)

The fields in this register control performance tuning associated with how the host controller posts data to the TX latency FIFO before moving the data onto the USB bus. The specific areas of performance include the how much data to post into the FIFO and an estimate for how long that operation should take in the target system.

Definitions:

T_0 = Standard packet overhead

T_1 = Time to send data payload

T_{ff} = Time to fetch packet into TX FIFO up to specified level.

T_s = Total Packet Flight Time (send-only) packet

$T_s = T_0 + T_1$

T_p = Total Packet Time (fetch and send) packet

$T_p = T_{ff} + T_0 + T_1$

Upon discovery of a transmit (OUT/SETUP) packet in the data structures, host controller checks to ensure T_p remains before the end of the [micro]frame. If so it proceeds to pre-fill the TX FIFO. If at anytime during the pre-fill operation the time remaining the [micro]frame is < T_s then the packet attempt ceases and the packet is tried at a later time. Although this is not an error condition and the host controller will eventually recover, a

mark will be made the scheduler health counter to note the occurrence of a "back-off" event. When a back-off event is detected, the partial packet fetched may need to be discarded from the latency buffer to make room for periodic traffic that will begin after the next SOF. Too many back-off events can waste bandwidth and power on the system bus and thus should be minimized (not necessarily eliminated). Back-offs can be minimized with use of the n_TSCHHEALTH (T_{ff}) described below.

NOTE

The reset value could vary from instance to instance. Please see the detail in bit field description and ignore reset value in summary table in this case!

Address: 218_4000h base + 164h offset + (512d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										TXFIFOTHRES						Reserved			TXSCHHEALTH					TXSCHOH							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

USB_*n*TXFILLTUNING field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–16 TXFIFOTHRES	FIFO Burst Threshold. (Read/Write) This register controls the number of data bursts that are posted to the TX latency FIFO in host mode before the packet begins on to the bus. The minimum value is 2 and this value should be a low as possible to maximize USB performance. A higher value can be used in systems with unpredictable latency and/or insufficient bandwidth where the FIFO may underrun because the data transferred from the latency FIFO to USB occurs before it can be replenished from system memory. This value is ignored if the Stream Disable bit in USB_n_USBMODE register is set. Default value is '00h' for OTG controller core, and '02h' for Host-only controller core.
15–13 -	This field is reserved. Reserved
12–8 TXSCHHEALTH	Scheduler Health Counter. (Read/Write To Clear) This register increments when the host controller fails to fill the TX latency FIFO to the level programmed by TXFIFOTHRES before running out of time to send the packet before the next Start-Of-Frame. This health counter measures the number of times this occurs to provide feedback to selecting a proper TXSCHOH. Writing to this register will clear the counter and this counter will max. at 31. Default value is '08h' for OTG controller core, and '00h' for Host-only controller core.
TXSCHOH	Scheduler Overhead. (Read/Write) [Default = 0] This register adds an additional fixed offset to the schedule time estimator described above as Tff. As an approximation, the value chosen for this register should limit the number of back-off events captured in the TXSCHHEALTH to less than 10 per second in a highly utilized bus. Choosing a value that is too high for this register is not desired as it can needlessly reduce USB utilization. The time unit represented in this register is 1.267us when a device is connected in High-Speed Mode. The time unit represented in this register is 6.333us when a device is connected in Low/Full Speed Mode. Default value is '08h' for OTG controller core, and '00h' for Host-only controller core.

65.6.28 Endpoint NAK (USB_*n*ENDPTNAK)

Address: 218_4000h base + 178h offset + (512d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								EPTN								Reserved								EPRN							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*ENDPTNAK field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 EPTN	TX Endpoint NAK - R/WC. Each TX endpoint has 1 bit in this field. The bit is set when the device sends a NAK handshake on a received IN token for the corresponding endpoint. Bit [N] - Endpoint #[N], N is 0-7
15–8 -	This field is reserved. Reserved
EPRN	RX Endpoint NAK - R/WC. Each RX endpoint has 1 bit in this field. The bit is set when the device sends a NAK handshake on a received OUT or PING token for the corresponding endpoint. Bit [N] - Endpoint #[N], N is 0-7

65.6.29 Endpoint NAK Enable (USB_*n*ENDPTNAKEN)

Address: 218_4000h base + 17Ch offset + (512d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								EPTNE								Reserved								EPRNE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*ENDPTNAKEN field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 EPTNE	TX Endpoint NAK Enable - R/W. Each bit is an enable bit for the corresponding TX Endpoint NAK bit. If this bit is set and the corresponding TX Endpoint NAK bit is set, the NAK Interrupt bit is set. Bit [N] - Endpoint #[N], N is 0-7
15–8 -	This field is reserved. Reserved

Table continues on the next page...

USB_*n*ENDPTNAKEN field descriptions (continued)

Field	Description
EPRNE	RX Endpoint NAK Enable - R/W. Each bit is an enable bit for the corresponding RX Endpoint NAK bit. If this bit is set and the corresponding RX Endpoint NAK bit is set, the NAK Interrupt bit is set. Bit [N] - Endpoint #[N], N is 0-7

65.6.30 Configure Flag Register (USB_*n*CONFIGFLAG)

Address: 218_4000h base + 180h offset + (512d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															CF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

USB_*n*CONFIGFLAG field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 CF	Configure Flag Host software sets this bit as the last action in its process of configuring the Host Controller. This bit controls the default port-routing control logic. 0 Port routing control logic default-routes each port to an implementation dependent classic host controller. 1 Port routing control logic default-routes all ports to this host controller.

65.6.31 Port Status & Control (USB_*n*PORTSC1)

Host Controller

A host controller could implement one to eight port status and control registers. The number is determined by N_PORTs bits in HWSPARAMs register (please see Host Controller Structural Parameters (USB_*n*HCSPARAMS)). Software could read this parameter register to determine how many ports need service.

All controller cores on this product are Single-Port, so there is only one port status and control register for each controller core.

This register is only reset by power on reset or controller core reset. The initial conditions of a port are:

- No device connected
- Port disabled

If the port supports power control, this state remains until port power is supplied (by software).

Device Controller

A controller operating in device mode has only port register one (PORTSC1) and it does not support power control in that mode. Port control in device mode is only used for status port reset, suspend, and current connect status. It is also used to initiate test mode or force signaling and allows software to put the PHY into low power suspend mode and disable the PHY clock.

Address: 218_4000h base + 184h offset + (512d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PTS_1		STS	PTW	PSPD		PTS_2	PFSC	PHCD	WKOC	WKDC	WKCN	PTC			
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PIC		PO	PP	LS		HSP	PR	SUSP	FPR	OCC	OCA	PEC	PE	CSC	CCS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*PORTSC1 field descriptions

Field	Description
31–30 PTS_1	Bit field {bit25, bit31, bit30}: "000b" UTMI/UTMI+ "001b" Reserved "010b" ULPI "011b" Serial/USB 1.1 PHY/IC-USB (FS Only) "100b" HSIC Parallel Transceiver Select (bit25, bit31, bi30). For OTG core, it is Read-Only. Reset value is 000b. For Host1/Host2/Host3 core, it is Read/Write. Reset value is 000b. **NOTE:** All USB port interface modes are listed in this field description, but not all are supported. For detail feature of each controller core, please see Features . The behaviour is unknown when unsupported interface mode is selected.
29 STS	Serial Transceiver Select - Read Only Serial Transceiver Select 1 Serial Interface Engine is selected 0 Parallel Interface signals is selected Serial Interface Engine can be used in combination with UTMI+/ULPI physical interface to provide FS/LS signaling instead of the parallel interface signals. When this bit is set '1b', serial interface engine will be used instead of parallel interface signals. This bit has no effect unless PTS bits is set to select UTMI+/ULPI interface. The Serial/USB1.1 PHY/IC-USB will use the serial interface engine for FS/LS signaling regardless of this bit value.
28 PTW	Parallel Transceiver Width This bit has no effect if serial interface engine is used. For OTG/Host1/Host2/Host3 core, it is Read-Only. Reset value is '1b'. 0 Select the 8-bit UTMI interface [60MHz] 1 Select the 16-bit UTMI interface [30MHz]
27–26 PSPD	Port Speed - Read Only. This register field indicates the speed at which the port is operating. 00 Full Speed 01 Low Speed 10 High Speed 11 Undefined
25 PTS_2	See description at bits 31-30
24 PFSC	Port Force Full Speed Connect - Read/Write. Default = 0b. When this bit is set to '1b', the port will be forced to only connect at Full Speed, It disables the chirp sequence that allows the port to identify itself as High Speed.

Table continues on the next page...

USB_*n*PORTSC1 field descriptions (continued)

Field	Description
	1 Forced to full speed 0 Normal operation
23 PHCD	PHY Low Power Suspend - Clock Disable (PLPSCD) - Read/Write. Default = 0b. When this bit is set to '1b', the PHY clock is disabled. Reading this bit will indicate the status of the PHY clock. **NOTE:** The PHY clock cannot be disabled if it is being used as the system clock. In device mode, The PHY can be put into Low Power Suspend when the device is not running (USBCMD Run/Stop=0b) or the host has signalled suspend (PORTSC1 SUSPEND=1b). PHY Low power suspend will be cleared automatically when the host initials resume. Before forcing a resume from the device, the device controller driver must clear this bit. In host mode, the PHY can be put into Low Power Suspend when the downstream device has been put into suspend mode or when no downstream device is connected. Low power suspend is completely under the control of software. 1 Disable PHY clock 0 Enable PHY clock
22 WKOC	Wake on Over-current Enable (WKOC_E) - Read/Write. Default = 0b. Writing this bit to a one enables the port to be sensitive to over-current conditions as wake-up events. This field is zero if *Port Power(*Port Status & Control (USB_*n*PORTSC1)*)* is zero.
21 WKDC	Wake on Disconnect Enable (WKDSCNNT_E) - Read/Write. Default=0b. Writing this bit to a one enables the port to be sensitive to device disconnects as wake-up events. This field is zero if *Port Power(*Port Status & Control (USB_*n*PORTSC1)*)* is zero or in device mode.
20 WKCN	Wake on Connect Enable (WKCNNT_E) - Read/Write. Default=0b. Writing this bit to a one enables the port to be sensitive to device connects as wake-up events. This field is zero if *Port Power(*Port Status & Control (USB_*n*PORTSC1)*)* is zero or in device mode.
19–16 PTC	Port Test Control - Read/Write. Default = 0000b. Refer to Port Test Mode for the operational model for using these test modes and the USB Specification Revision 2.0, Chapter 7 for details on each test mode. The FORCE_ENABLE_FS and FORCE ENABLE_LS are extensions to the test mode support specified in the EHCI specification. Writing the PTC field to any of the FORCE_ENABLE_{HS/FS/LS} values will force the port into the connected and enabled state at the selected speed. Writing the PTC field back to TEST_MODE_DISABLE will allow the port state machines to progress normally from that point. **NOTE:** *Low speed operations are not supported as a peripheral device.* Any other value than zero indicates that the port is operating in test mode. Value Specific Test 0000 TEST_MODE_DISABLE 0001 J_STATE 0010 K_STATE 0011 SE0 (host) / NAK (device) 0100 Packet 0101 FORCE_ENABLE_HS 0110 FORCE_ENABLE_FS

Table continues on the next page...

USB_*n*PORTSC1 field descriptions (continued)

Field	Description
	0111 FORCE_ENABLE_LS 1000-1111 Reserved
15–14 PIC	Port Indicator Control - Read/Write. Default = Ob. Writing to this field has no effect if the P_INDICATOR bit in the HCSPARAMS register is a zero. Refer to the USB Specification Revision 2.0 for a description on how these bits are to be used. This field is zero if *Port Power* is zero. Bit Value Meaning 00 Port indicators are off 01 Amber 10 Green 11 Undefined
13 PO	Port Owner-Read/Write. Default = 0. This bit unconditionally goes to a 0 when the configured bit in the CONFIGFLAG register makes a 0 to 1 transition. This bit unconditionally goes to 1 whenever the Configured bit is zero System software uses this field to release ownership of the port to a selected host controller (in the event that the attached device is not a high-speed device). Software writes a one to this bit when the attached device is not a high-speed device. A one in this bit means that an internal companion controller owns and controls the port. Port owner handoff is not supported in all controller cores, therefore this bit will always be 0.
12 PP	Port Power (PP)-Read/Write or Read Only. The function of this bit depends on the value of the Port Power Switching (PPC) field in the HCSPARAMS register. The behavior is as follows: PPC PP Operation 0 1b *Read Only - Host controller does not have port power control switches. Each port is hard-wired to power.* 1 1b/0b - *Read/Write. Host/OTG controller requires port power control switches. This bit represents the current setting of the switch (0=off, 1=on). When power is not available on a port (that is, PP equals a 0), the port is non-functional and will not report attaches, detaches, etc.* When an over-current condition is detected on a powered port and PPC is a one, the PP bit in each affected port may be transitional by the host controller driver from a one to a zero (removing power from the port). This feature is implemented in all controller cores (PPC = 1).
11–10 LS	Line Status-Read Only. These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. In host mode, the use of linestate by the host controller driver is not necessary (unlike EHCI), because the port controller state machine and the port routing manage the connection of LS and FS. In device mode, the use of linestate by the device controller driver is not necessary. The encoding of the bits are: Bits [11:10] Meaning 00 SE0

Table continues on the next page...

USB_*n*PORTSC1 field descriptions (continued)

Field	Description
	10 J-state 01 K-state 11 Undefined
9 HSP	High-Speed Port - Read Only. Default = 0b. When the bit is one, the host/device connected to the port is in high-speed mode and if set to zero, the host/device connected to the port is not in a high-speed mode. **NOTE:** HSP is redundant with PSPD(bit 27, 26) but remained for compatibility.
8 PR	Port Reset - Read/Write or Read Only. Default = 0b. In Host Mode: Read/Write. 1=Port is in Reset. 0=Port is not in Reset. Default 0. When software writes a one to this bit the bus-reset sequence as defined in the USB Specification Revision 2.0 is started. *This bit will automatically change to zero after the reset sequence is complete. This behavior is different from EHCI where the host controller driver is required to set this bit to a zero after the reset duration is timed in the driver.* In Device Mode: This bit is a read only status bit. Device reset from the USB bus is also indicated in the USBSTS register. This field is zero if *Port Power(*Port Status & Control (USB_*n*PORTSC1)*)* is zero.
7 SUSP	Suspend - Read/Write or Read Only. Default = 0b. 1=Port in suspend state. 0=Port not in suspend state. In Host Mode: Read/Write. Port Enabled Bit and Suspend bit of this register define the port states as follows: Bits [Port Enabled, Suspend] Port State 0x Disable 10 Enable 11 Suspend When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB. The host controller will unconditionally set this bit to zero when software sets the *Force Port Resume* bit to zero. The host controller ignores a write of zero to this bit. If host software sets this bit to a one when the port is not enabled (that is, *Port enabled* bit is a zero) the results are undefined. This field is zero if *Port Power(*Port Status & Control (USB_*n*PORTSC1)*)* is zero in host mode. In Device Mode: Read Only. In device mode this bit is a read only status bit.
6 FPR	Force Port Resume -Read/Write. 1= Resume detected/driven on port. 0=No resume (K-state) detected/driven on port. Default = 0. In Host Mode: Software sets this bit to one to drive resume signaling. The Host Controller sets this bit to one if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to a one because a J-to-K transition is detected, the *Port Change Detect* bit in the USBSTS register is also set to one. *This bit*

Table continues on the next page...

USB_*n*PORTSC1 field descriptions (continued)

Field	Description
	will automatically change to zero after the resume sequence is complete. This behavior is different from EHCI where the host controller driver is required to set this bit to a zero after the resume duration is timed in the driver. Note that when the Host controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains a one. This bit will remain a one until the port has switched to the high-speed idle. Writing a zero has no effect because the port controller will time the resume operation, clear the bit the port control state switches to HS or FS idle. This field is zero if *Port Power(*Port Status & Control (USB_*n*PORTSC1)*)* is zero in host mode. This bit is not-EHCI compatible. In Device mode: After the device has been in Suspend State for 5ms or more, software must set this bit to one to drive resume signaling before clearing. The Device Controller will set this bit to one if a J-to-K transition is detected while the port is in the Suspend state. The bit will be cleared when the device returns to normal operation. Also, when this bit wil be cleared because a K-to-J transition detected, the *Port Change Detect* bit in the USBSTS register is also set to one.
5 OCC	Over-current Change-R/WC. Default=0. This bit is set '1b' by hardware when there is a change to Over-current Active. Software can clear this bit by writing a one to this bit position.
4 OCA	Over-current Active-Read Only. Default 0. This bit will automatically transition from one to zero when the over current condition is removed. 1 This port currently has an over-current condition 0 This port does not have an over-current condition.
3 PEC	Port Enable/Disable Change-R/WC. 1=Port enabled/disabled status has changed. 0=No change. Default = 0. In Host Mode: For the root hub, this bit is set to a one only when a port is disabled due to disconnect on the port or due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification). Software clears this by writing a one to it. This field is zero if *Port Power(*Port Status & Control (USB_*n*PORTSC1)*)* is zero. In Device mode: The device port is always enabled, so this bit is always '0b'.
2 PE	Port Enabled/Disabled-Read/Write. 1=Enable. 0=Disable. Default 0. In Host Mode: Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a one to this field. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by the host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events. When the port is disabled, (0b) downstream propagation of data is blocked except for reset. This field is zero if *Port Power(*Port Status & Control (USB_*n*PORTSC1)*)* is zero in host mode. In Device Mode: The device port is always enabled, so this bit is always '1b'.

Table continues on the next page...

USB_*n*PORTSC1 field descriptions (continued)

Field	Description
1 CSC	Connect Status Change-R/WC. 1 =Change in Current Connect Status. 0=No change. Default 0. In Host Mode: Indicates a change has occurred in the port's Current Connect Status. The host/device controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be 'setting' an already-set bit (that is, the bit will remain set). Software clears this bit by writing a one to it. This field is zero if *Port Power(*Port Status & Control (USB_*n*PORTSC1)*)* is zero in host mode. In Device Mode: This bit is undefined in device controller mode.
0 CCS	Current Connect Status-Read Only. In Host Mode: 1=Device is present on port. 0=No device is present. Default = 0. This value reflects the current state of the port, and may not correspond directly to the event that caused the *Connect Status Change* bit (Bit 1) to be set. This field is zero if *Port Power(*Port Status & Control (USB_*n*PORTSC1)*)* is zero in host mode. In Device Mode: 1=Attached. 0=Not Attached. Default=0. A one indicates that the device successfully attached and is operating in either high speed or full speed as indicated by the High Speed Port bit in this register. A zero indicates that the device did not attach successfully or was forcibly disconnected by the software writing a zero to the Run bit in the USBCMD register. It does not state the device being disconnected or suspended.

65.6.32 On-The-Go Status & control (USB_*n*OTGSC)

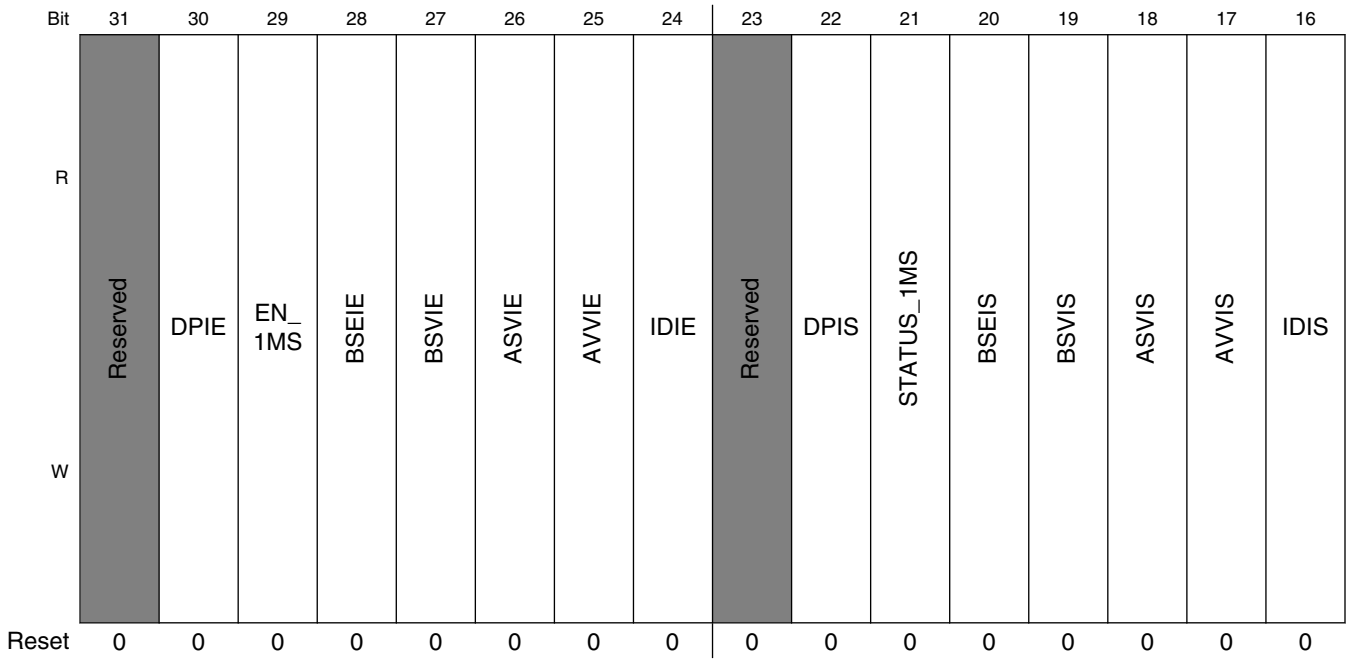
This register is availabe only in OTG controller core. It has four sections:

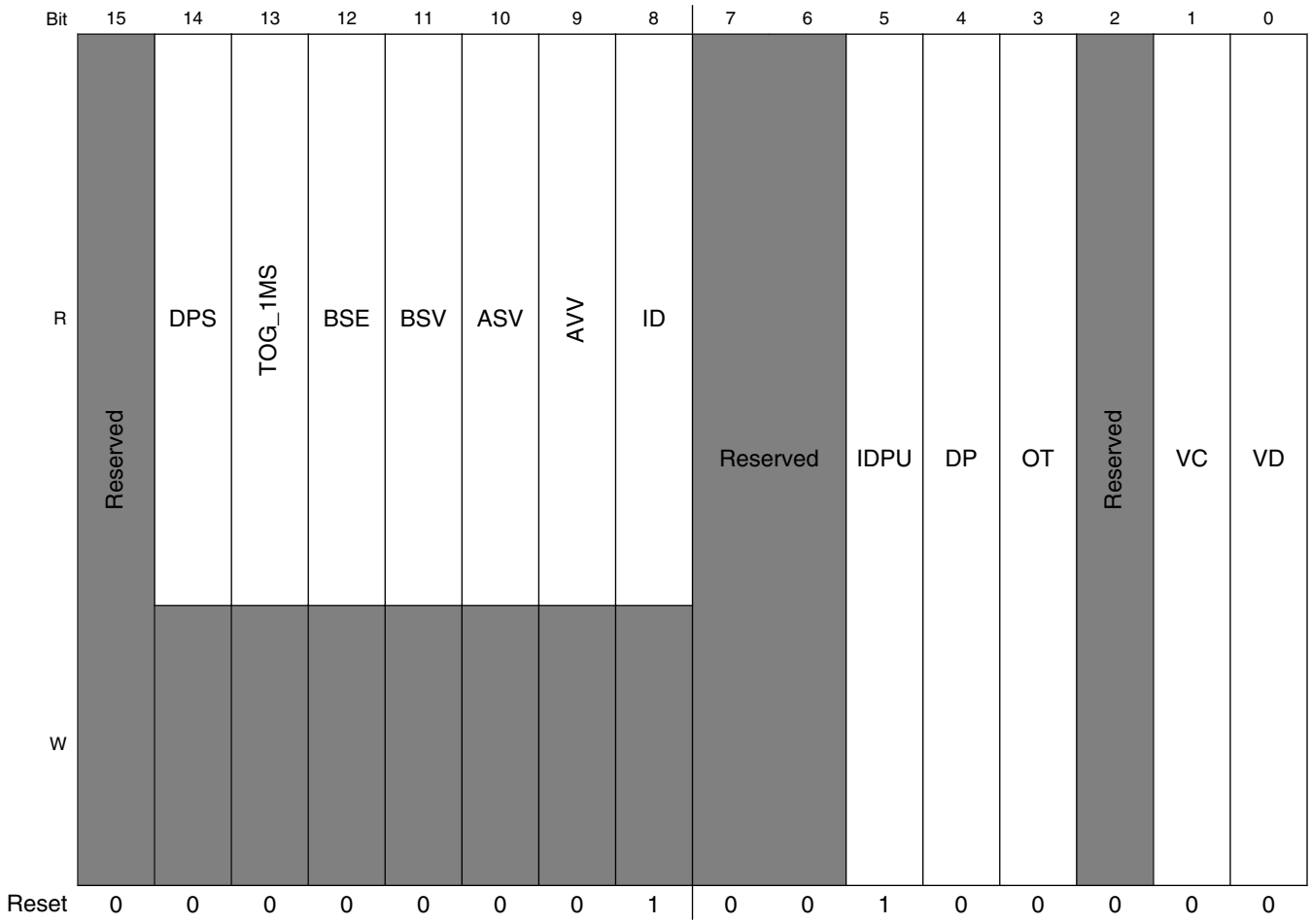
- OTG Interrupt enables (Read/Write)
- OTG Interrupt status (Read/Write to Clear)
- OTG Status inputs (Read Only)
- OTG Controls (Read/Write)

The status inputs are debounced using a 1 ms time constant. Values on the status inputs that do not persist for more than 1 ms does not cause an update of the status input register, or cause an OTG interrupt.

See also USB Device Mode (USB_*n*USBMODE) register.

Address: 218_4000h base + 1A4h offset + (512d × i), where i=0d to 0d





USB_*n*OTGSC field descriptions

Field	Description
31 -	This field is reserved. Reserved
30 DPIE	Data Pulse Interrupt Enable
29 EN_1MS	1 millisecond timer Interrupt Enable - Read/Write
28 BSEIE	B Session End Interrupt Enable - Read/Write. Setting this bit enables the B session end interrupt.
27 BSVIE	B Session Valid Interrupt Enable - Read/Write. Setting this bit enables the B session valid interrupt.
26 ASVIE	A Session Valid Interrupt Enable - Read/Write. Setting this bit enables the A session valid interrupt.
25 AVVIE	A VBus Valid Interrupt Enable - Read/Write. Setting this bit enables the A VBus valid interrupt.
24 IDIE	USB ID Interrupt Enable - Read/Write. Setting this bit enables the USB ID interrupt.
23 -	This field is reserved. Reserved
22 DPIS	Data Pulse Interrupt Status - Read/Write to Clear. This bit is set when data bus pulsing occurs on DP or DM. Data bus pulsing is only detected when USBMODE.CM = Host (11) and PORTSC1(0)[PP] = 0. Software must write a one to clear this bit.
21 STATUS_1MS	1 millisecond timer Interrupt Status - Read/Write to Clear. This bit is set once every millisecond. Software must write a one to clear this bit.
20 BSEIS	B Session End Interrupt Status - Read/Write to Clear. This bit is set when VBus has fallen below the B session end threshold. Software must write a one to clear this bit.
19 BSVIS	B Session Valid Interrupt Status - Read/Write to Clear. This bit is set when VBus has either risen above or fallen below the B session valid threshold. Software must write a one to clear this bit.
18 ASVIS	A Session Valid Interrupt Status - Read/Write to Clear. This bit is set when VBus has either risen above or fallen below the A session valid threshold. Software must write a one to clear this bit.
17 AVVIS	A VBus Valid Interrupt Status - Read/Write to Clear. This bit is set when VBus has either risen above or fallen below the VBus valid threshold on an A device. Software must write a one to clear this bit.
16 IDIS	USB ID Interrupt Status - Read/Write. This bit is set when a change on the ID input has been detected. Software must write a one to clear this bit.

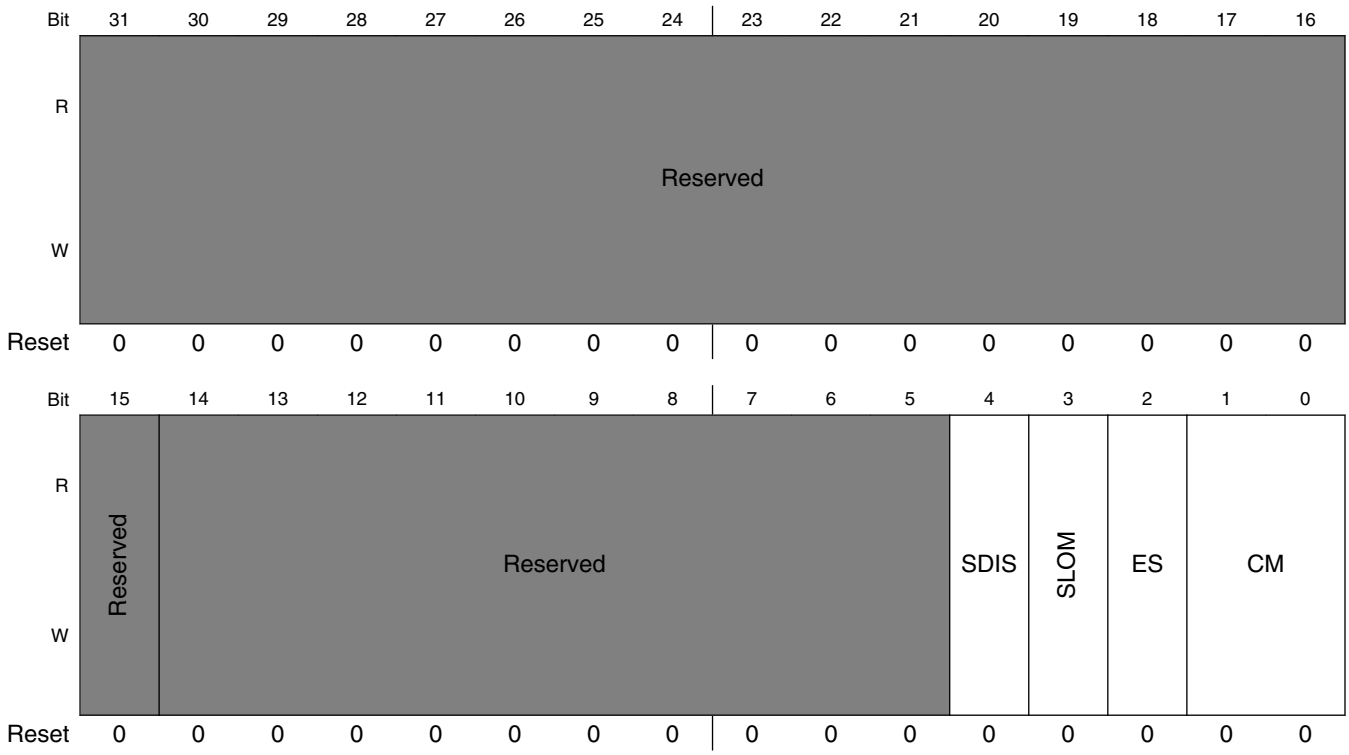
Table continues on the next page...

USB_*n*OTGSC field descriptions (continued)

Field	Description
15 -	This field is reserved. Reserved
14 DPS	Data Bus Pulsing Status - Read Only. A '1' indicates data bus pulsing is being detected on the port.
13 TOG_1MS	1 millisecond timer toggle - Read Only. This bit toggles once per millisecond.
12 BSE	B Session End - Read Only. Indicates VBus is below the B session end threshold.
11 BSV	B Session Valid - Read Only. Indicates VBus is above the B session valid threshold.
10 ASV	A Session Valid - Read Only. Indicates VBus is above the A session valid threshold.
9 AVV	A VBus Valid - Read Only. Indicates VBus is above the A VBus valid threshold.
8 ID	USB ID - Read Only. 0 = A device, 1 = B device
7–6 -	This field is reserved. Reserved
5 IDPU	ID Pullup - Read/Write This bit provide control over the ID pull-up resistor; 0 = off, 1 = on [default]. When this bit is 0, the ID input will not be sampled.
4 DP	Data Pulsing - Read/Write. Setting this bit causes the pullup on DP to be asserted for data pulsing during SRP.
3 OT	OTG Termination - Read/Write. This bit must be set when the OTG device is in device mode, this controls the pulldown on DM.
2 -	This field is reserved. Reserved
1 VC	VBUS Charge - Read/Write. Setting this bit causes the VBus line to be charged. This is used for VBus pulsing during SRP.
0 VD	VBUS_Discharge - Read/Write. Setting this bit causes VBus to discharge through a resistor.

65.6.33 USB Device Mode (USB_*n*USBMODE)

Address: 218_4000h base + 1A8h offset + (512d × i), where i=0d to 3d



USB_*n*USBMODE field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14–5 -	This field is reserved. Reserved
4 SDIS	Stream Disable Mode. (0 - Inactive [default]; 1 - Active) Device Mode: Setting to a '1' disables double priming on both RX and TX for low bandwidth systems. This mode ensures that when the RX and TX buffers are sufficient to contain an entire packet that the standard double buffering scheme is disabled to prevent overruns/underruns in bandwidth limited systems. Note: In High Speed Mode, all packets received are responded to with a NYET handshake when stream disable is active. Host Mode: Setting to a '1' ensures that overruns/underruns of the latency FIFO are eliminated for low bandwidth systems where the RX and TX buffers are sufficient to contain the entire packet. Enabling stream disable also has the effect of ensuring the TX latency is filled to capacity before the packet is launched onto the USB. **NOTE:** Time duration to pre-fill the FIFO becomes significant when stream disable is active. See TX FIFO Fill Tuning (USB_*n*TXFILLTUNING) and TXTTFILLTUNING [MPH Only] to characterize the adjustments needed for the scheduler when using this feature.

Table continues on the next page...

USB_*n*USBMODE field descriptions (continued)

Field	Description
	NOTE: The use of this feature substantially limits of the overall USB performance that can be achieved.
3 SLOM	Setup Lockout Mode. In device mode, this bit controls behavior of the setup lock mechanism. See Control Endpoint Operation Model . 0 Setup Lockouts On (default); 1 Setup Lockouts Off (DCD requires use of Setup Data Buffer Tripwire in USB Command Register (USB_*n*USBCMD) .
2 ES	Endian Select - Read/Write. This bit can change the byte alignment of the transfer buffers to match the host microprocessor. The bit fields in the microprocessor interface and the data structures are unaffected by the value of this bit because they are based upon the 32-bit word. Bit Meaning 0 Little Endian [Default] 1 Big Endian
CM	Controller Mode - R/WO. Controller mode is defaulted to the proper mode for host only and device only implementations. For those designs that contain both host & device capability, the controller defaults to an idle state and needs to be initialized to the desired operating mode after reset. For combination host/device controllers, this register can only be written once after reset. If it is necessary to switch modes, software must reset the controller by writing to the *RESET* bit in the USBCMD register before reprogramming this register. For OTG controller core, reset value is '00b'. For Host-only controller core, reset value is '11b'. 00 Idle [Default for combination host/device] 01 Reserved 10 Device Controller [Default for device only controller] 11 Host Controller [Default for host only controller]

65.6.34 Endpoint Setup Status (USB_*n*ENDPTSETUPSTAT)

Address: 218_4000h base + 1ACh offset + (512d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																ENDPTSETUPSTAT																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*ENDPTSETUPSTAT field descriptions

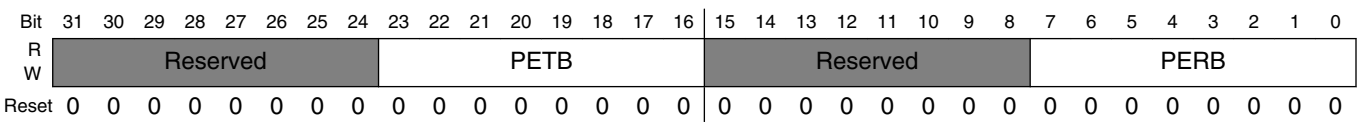
Field	Description
31–16 -	This field is reserved. Reserved
ENDPTSETUPSTAT	Setup Endpoint Status. For every setup transaction that is received, a corresponding bit in this register is set to one. Software must clear or acknowledge the setup transfer by writing a one to a respective bit after it has read the setup data from Queue head. The response to a setup packet as in the order of operations and total response time is crucial to limit bus time outs while the setup lock our mechanism is engaged. See Managing Endpoints in the Device Operational Model. This register is only used in device mode.

65.6.35 Endpoint Prime (USB_*n*ENDPTPRIME)

This register is only used in device mode.

When software sets the prime bit for a given endpoint, the device controller loads the transfer descriptor, pointed to by the queue head, such that the endpoint is ready to transmit or receive when the host sends a request (IN/OUT token). The endpoint will NAK all requests from the host until the endpoint is primed. The controller will automatically re-prime the endpoint with a new transfer descriptor when one is found via the next_dtd pointer of the current transfer descriptor. Hence, the prime bit must only be set by software when a descriptor is added to the queue head.

Address: 218_4000h base + 1B0h offset + (512d × i), where i=0d to 0d



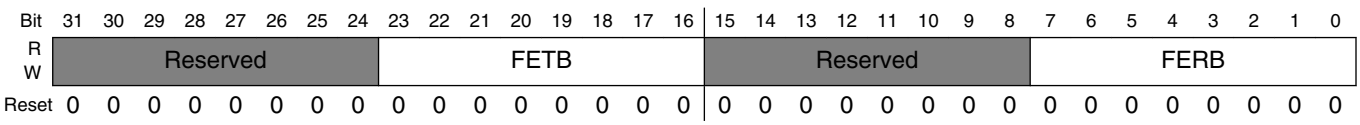
USB_*n*ENDPTPRIME field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 PETB	Prime Endpoint Transmit Buffer - R/WS. For each endpoint a corresponding bit is used to request that a buffer is prepared for a transmit operation in order to respond to a USB IN/INTERRUPT transaction. Software should write a one to the corresponding bit when posting a new transfer descriptor to an endpoint queue head. Hardware automatically uses this bit to begin parsing for a new transfer descriptor from the queue head and prepare a transmit buffer. Hardware clears this bit when the associated endpoint(s) is (are) successfully primed. **NOTE:** These bits are momentarily set by hardware during hardware re-priming operations when a dTD is retired, and the dQH is updated. PETB[N] - Endpoint #N, N is in 0..7
15–8 -	This field is reserved. Reserved
PERB	Prime Endpoint Receive Buffer - R/WS. For each endpoint, a corresponding bit is used to request a buffer prepare for a receive operation for when a USB host initiates a USB OUT transaction. Software should write a one to the corresponding bit whenever posting a new transfer descriptor to an endpoint queue head. Hardware automatically uses this bit to begin parsing for a new transfer descriptor from the queue head and prepare a receive buffer. Hardware clears this bit when the associated endpoint(s) is (are) successfully primed. **NOTE:** These bits are momentarily set by hardware during hardware re-priming operations when a dTD is retired, and the dQH is updated. PERB[N] - Endpoint #N, N is in 0..7

65.6.36 Endpoint Flush (USB_*n*ENDPTFLUSH)

This register is only used in device mode.

Address: 218_4000h base + 1B4h offset + (512d × i), where i=0d to 0d



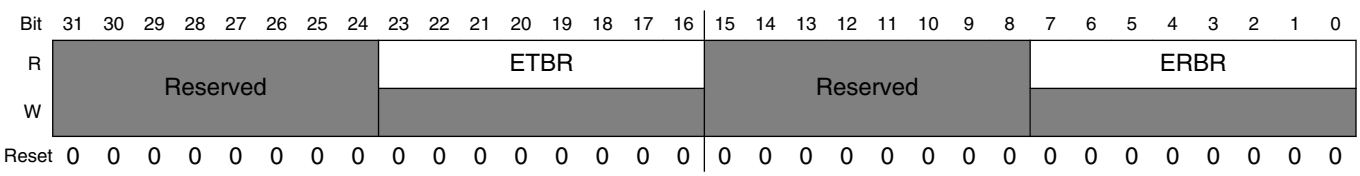
USB_*n*ENDPTFLUSH field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 FETB	Flush Endpoint Transmit Buffer - R/WS. Writing one to a bit(s) in this register causes the associated endpoint(s) to clear any primed buffers. If a packet is in progress for one of the associated endpoints, then that transfer continues until completion. Hardware clears this register after the endpoint flush operation is successful. FETB[N] - Endpoint #N, N is in 0..7
15–8 -	This field is reserved. Reserved
FERB	Flush Endpoint Receive Buffer - R/WS. Writing one to a bit(s) causes the assocUOGiated endpoint(s) to clear any primed buffers. If a packet is in progress for one of the associated endpoints, then that transfer continues until completion. Hardware clears this register after the endpoint flush operation is successful. FERB[N] - Endpoint #N, N is in 0..7\

65.6.37 Endpoint Status (USB_*n*ENDPTSTAT)

This register is only used in device mode.

Address: 218_4000h base + 1B8h offset + (512d × i), where i=0d to 0d



USB_*n*ENDPTSTAT field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 ETBR	Endpoint Transmit Buffer Ready -- Read Only. One bit for each endpoint indicates status of the respective endpoint buffer. This bit is set to one by the hardware as a response to receiving a command from a corresponding bit in the ENDPTPRIME register. There is always a delay between setting a bit in the ENDPTPRIME register and endpoint indicating ready. This delay time varies based upon the current USB traffic and the number of bits set in the ENDPRIME register. Buffer ready is cleared by USB reset, by the USB DMA system, or through the ENDPTFLUSH register. **NOTE:** These bits are momentarily cleared by hardware during hardware endpoint re-priming operations when a dTD is retired, and the dQH is updated. ETBR[N] - Endpoint #N, N is in 0..7
15–8 -	This field is reserved. Reserved
ERBR	Endpoint Receive Buffer Ready -- Read Only. One bit for each endpoint indicates status of the respective endpoint buffer. This bit is set to a one by the hardware as a response to receiving a command from a corresponding bit in the ENDPRIME register. There is always a delay between setting a bit in the ENDPRIME register and endpoint indicating ready. This delay time varies based upon the current USB traffic and the number of bits set in the ENDPRIME register. Buffer ready is cleared by USB reset, by the USB DMA system, or through the ENDPTFLUSH register. **NOTE:** These bits are momentarily cleared by hardware during hardware endpoint re-priming operations when a dTD is retired, and the dQH is updated. ERBR[N] - Endpoint #N, N is in 0..7

65.6.38 Endpoint Complete (USB_*n*ENDPTCOMPLETE)

This register is only used in device mode.

Address: 218_4000h base + 1BCh offset + (512d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								ETCE								Reserved								ERCE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*ENDPTCOMPLETE field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 ETCE	Endpoint Transmit Complete Event - R/WC. Each bit indicates a transmit event (IN/INTERRUPT) occurred and software should read the corresponding endpoint queue to determine the endpoint status. If the corresponding IOC bit is set in the Transfer Descriptor, then this bit is set simultaneously with the *USBINT* . Writing one clears the corresponding bit in this register. ETCE[N] - Endpoint #N, N is in 0..7

Table continues on the next page...

USB_*n*ENDPTCOMPLETE field descriptions (continued)

Field	Description
15–8 -	This field is reserved. Reserved
ERCE	Endpoint Receive Complete Event - RW/C. Each bit indicates a received event (OUT/SETUP) occurred and software should read the corresponding endpoint queue to determine the transfer status. If the corresponding IOC bit is set in the Transfer Descriptor, then this bit is set simultaneously with the *USBINT* . Writing one clears the corresponding bit in this register. ERCE[N] - Endpoint #N, N is in 0..7

65.6.39 Endpoint Control0 (USB_*n*ENDPTCTRL0)

Every Device implements Endpoint 0 as a control endpoint.

Address: 218_4000h base + 1C0h offset + (512d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								TXE	Reserved			TXT		Reserved	TXS
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RXE	Reserved			RXT		Reserved	RXS
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

USB_*n*ENDPTCTRL0 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 TXE	TX Endpoint Enable 1 Enabled

Table continues on the next page...

USB_*n*ENDPTCTRL0 field descriptions (continued)

Field	Description
	Endpoint0 is always enabled.
22–20 -	This field is reserved. Reserved
19–18 TXT	TX Endpoint Type - Read/Write 00 - Control Endpoint0 is fixed as a Control End Point.
17 -	This field is reserved. Reserved
16 TXS	TX Endpoint Stall - Read/Write 0 End Point OK [Default] 1 End Point Stalled Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. It continues returning STALL until the bit is cleared by software or it is automatically cleared upon receipt of a new SETUP request. After receiving a SETUP request, this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. **NOTE:** There is a slight delay (50 clocks max.) between the endptsetupstat being cleared and hardware continuing to clear this bit. In most systems it is unlikely the DCD software will observe this delay. However, should the dcd observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a newsetup has been received by checking the associated endptsetupstat bit.
15–8 -	This field is reserved. Reserved
7 RXE	RX Endpoint Enable 1 Enabled Endpoint0 is always enabled.
6–4 -	This field is reserved. Reserved
3–2 RXT	RX Endpoint Type - Read/Write 00 Control Endpoint0 is fixed as a Control End Point.
1 -	This field is reserved. Reserved
0 RXS	RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. It continues returning STALL until the bit is cleared by software or it is automatically cleared upon receipt of a new SETUP request. After receiving a SETUP request, this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared.

Table continues on the next page...

USB_*n*ENDPTCTRL0 field descriptions (continued)

Field	Description
	NOTE: There is a slight delay (50 clocks max.) between the endptsetupstat being cleared and hardware continuing to clear this bit. In most systems it is unlikely the dcd software will observe this delay. However, should the dcd observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a newsetup has been received by checking the associated endptsetupstat bit.

65.6.40 Endpoint Control 1 (USB_*n*ENDPTCTRL1)

This is endpoint control register for endpoint 1 in device operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is Bulk-type). leaving an unconfigured endpoint control causes undefined behavior for the data pid tracking on the active endpoint/direction.

Address: 218_4000h base + 1C4h offset + (512d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								TXE	TXR	TXI	Reserved	TXT		TXD	TXS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*ENDPTCTRL1 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 TXE	TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22 TXR	TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled. This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
20 -	This field is reserved. Reserved
19–18 TXT	TX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
17 TXD	TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled This bit will be cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
15–8 -	This field is reserved. Reserved

Table continues on the next page...

USB_*n*ENDPTCTRL1 field descriptions (continued)

Field	Description
7 RXE	RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5 RXI	RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 -	This field is reserved. Reserved.
3–2 RXT	RX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Reserved
1 RXD	RX Endpoint Data Sink - Read/Write 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpointand this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.

65.6.41 Endpoint Control 2 (USB_*n*ENDPTCTRL2)

This is endpoint control register for endpoint 2 in device operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is Bulk-type). leaving an unconfigured endpoint control causes undefined behavior for the data pid tracking on the active endpoint/direction.

Address: 218_4000h base + 1C8h offset + (512d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								TXE	TXR	TXI	Reserved	TXT		TXD	TXS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*ENDPTCTRL2 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 TXE	TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.

Table continues on the next page...

USB_*n*ENDPTCTRL2 field descriptions (continued)

Field	Description
22 TXR	TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled. This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
20 -	This field is reserved. Reserved
19–18 TXT	TX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
17 TXD	TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled This bit will be cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
15–8 -	This field is reserved. Reserved
7 RXE	RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence

Table continues on the next page...

USB_*n*ENDPTCTRL2 field descriptions (continued)

Field	Description
	Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5 RXI	RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 -	This field is reserved. Reserved.
3–2 RXT	RX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Reserved
1 RXD	RX Endpoint Data Sink - Read/Write 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpointand this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.

65.6.42 Endpoint Control 3 (USB_*n*ENDPTCTRL3)

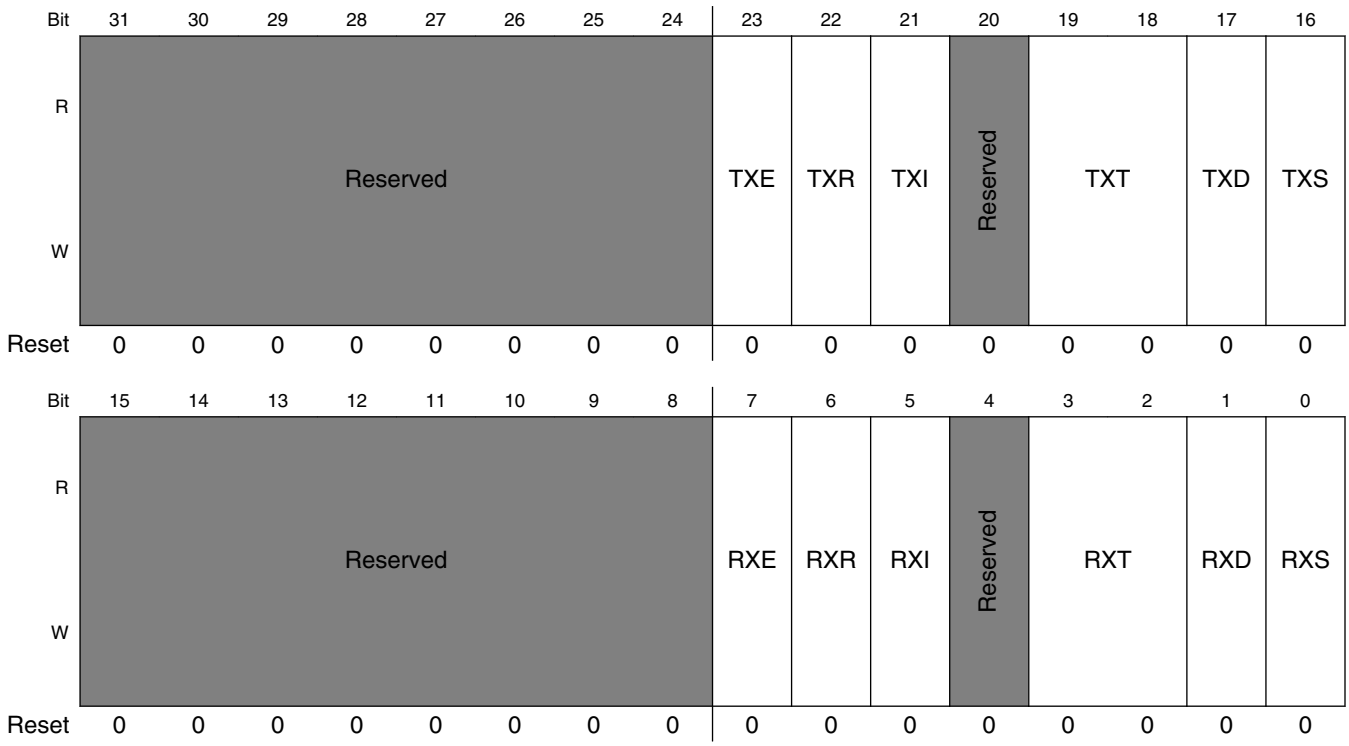
This is endpoint control register for endpoint 3 in device operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is Bulk-type). leaving an unconfigured endpoint control

causes undefined behavior for the data pid tracking on the active endpoint/direction.

Address: 218_4000h base + 1CCh offset + (512d × i), where i=0d to 0d



USB_*n*ENDPTCTRL3 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 TXE	TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22 TXR	TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled. This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
20 -	This field is reserved. Reserved

Table continues on the next page...

USB_*n*ENDPTCTRL3 field descriptions (continued)

Field	Description
19–18 TXT	TX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
17 TXD	TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled This bit will be cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
15–8 -	This field is reserved. Reserved
7 RXE	RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5 RXI	RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 -	This field is reserved. Reserved.
3–2 RXT	RX Endpoint Type - Read/Write 00 Control

Table continues on the next page...

USB_*n*ENDPTCTRL3 field descriptions (continued)

Field	Description
	01 Isochronous 10 Bulk 11 Reserved
1 RXD	RX Endpoint Data Sink - Read/Write 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpointand this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.

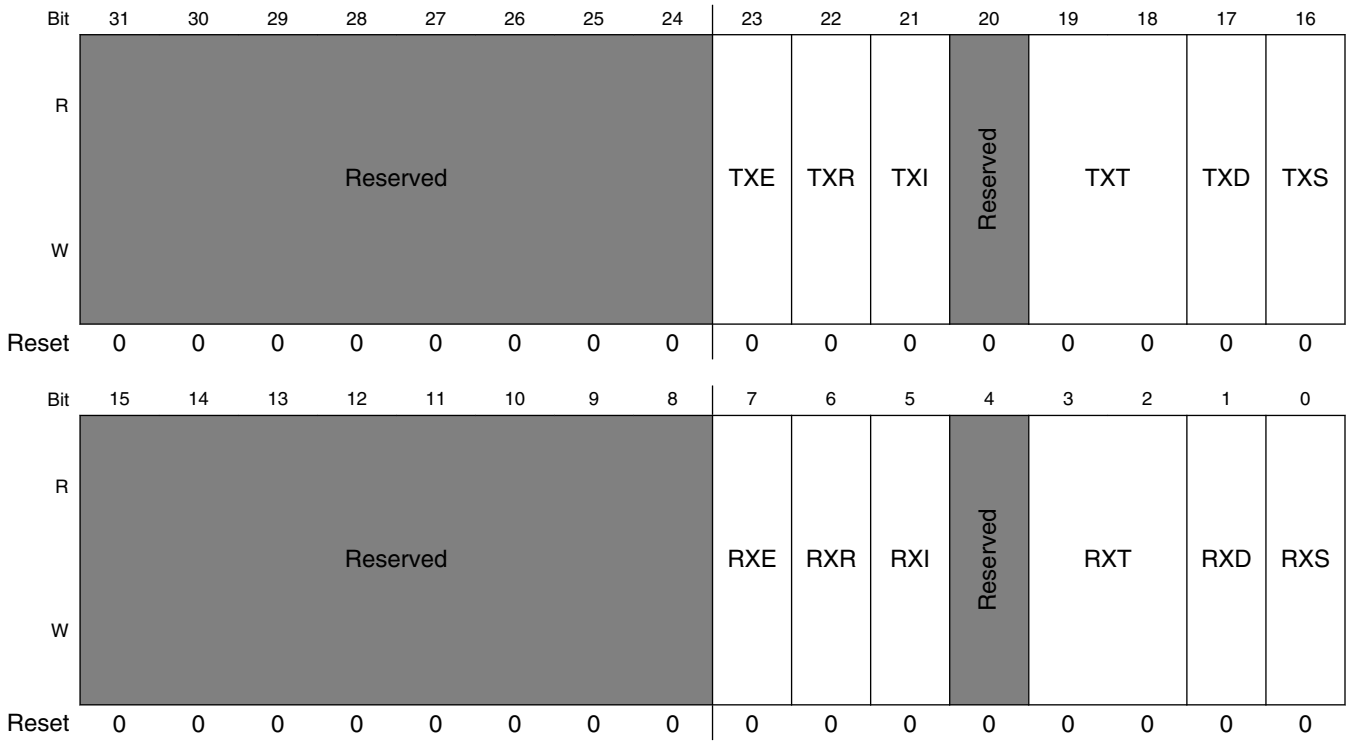
65.6.43 Endpoint Control 4 (USB_*n*ENDPTCTRL4)

This is endpoint control register for endpoint 4 in device operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is Bulk-type). leaving an unconfigured endpoint control causes undefined behavior for the data pid tracking on the active endpoint/direction.

Address: 218_4000h base + 1D0h offset + (512d × i), where i=0d to 0d



USB_*n*ENDPTCTRL4 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 TXE	TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22 TXR	TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled. This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
20 -	This field is reserved. Reserved
19–18 TXT	TX Endpoint Type - Read/Write 00 Control 01 Isochronous

Table continues on the next page...

USB_*n*ENDPTCTRL4 field descriptions (continued)

Field	Description
	10 Bulk 11 Interrupt
17 TXD	TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled This bit will be cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
15–8 -	This field is reserved. Reserved
7 RXE	RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5 RXI	RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 -	This field is reserved. Reserved.
3–2 RXT	RX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Reserved

Table continues on the next page...

USB_*n*ENDPTCTRL4 field descriptions (continued)

Field	Description
1 RXD	RX Endpoint Data Sink - Read/Write 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpointand this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.

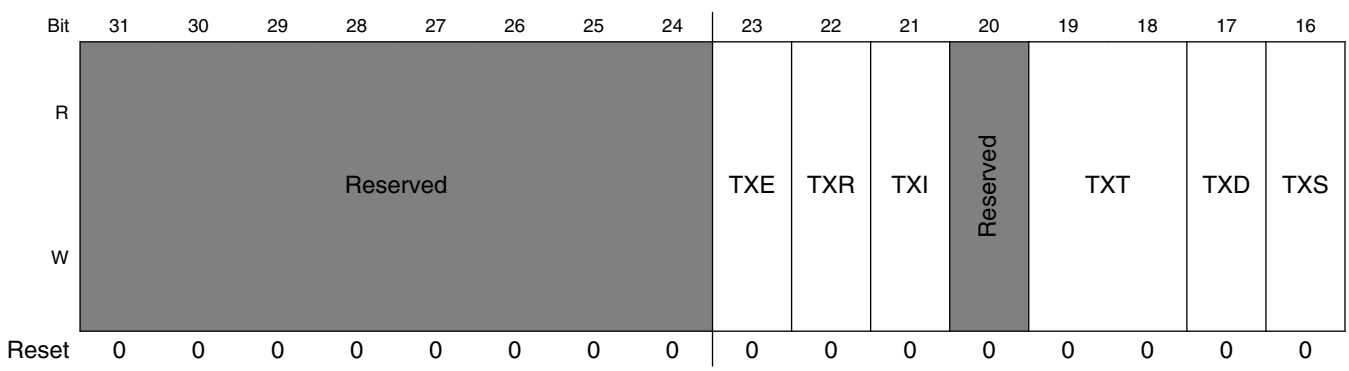
65.6.44 Endpoint Control 5 (USB_*n*ENDPTCTRL5)

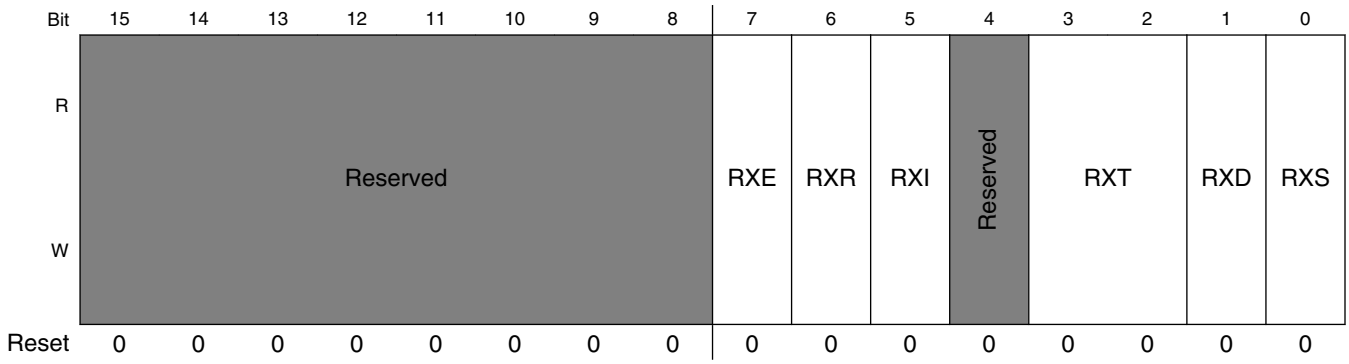
This is endpoint control register for endpoint 5 in device operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is Bulk-type). leaving an unconfigured endpoint control causes undefined behavior for the data pid tracking on the active endpoint/direction.

Address: 218_4000h base + 1D4h offset + (512d × i), where i=0d to 0d





USB_*n*ENDPTCTRL5 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 TXE	TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22 TXR	TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled. This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
20 -	This field is reserved. Reserved
19–18 TXT	TX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
17 TXD	TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled

Table continues on the next page...

USB_*n*ENDPTCTRL5 field descriptions (continued)

Field	Description
	This bit will be cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
15–8 -	This field is reserved. Reserved
7 RXE	RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5 RXI	RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 -	This field is reserved. Reserved.
3–2 RXT	RX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Reserved
1 RXD	RX Endpoint Data Sink - Read/Write 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled

Table continues on the next page...

USB_*n*ENDPTCTRL5 field descriptions (continued)

Field	Description
	This bit is set automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpointand this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.

65.6.45 Endpoint Control 6 (USB_*n*ENDPTCTRL6)

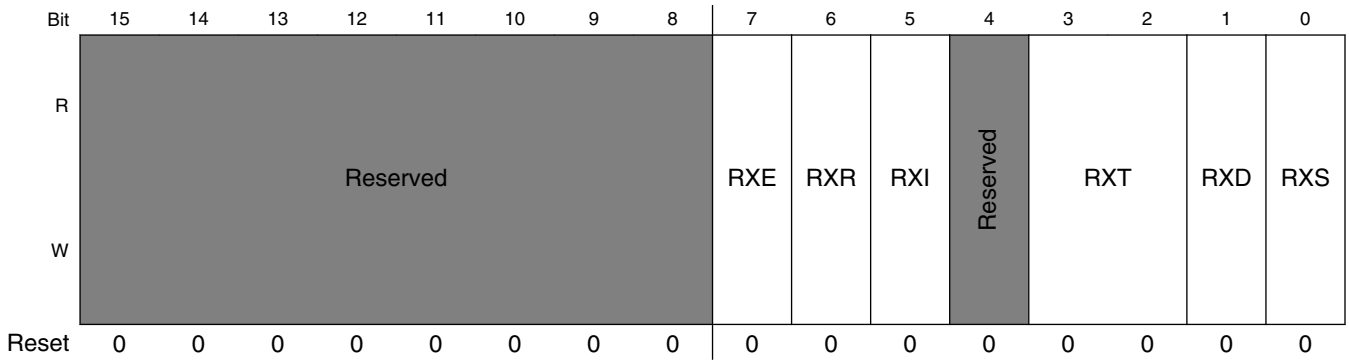
This is endpoint control register for endpoint 6 in device operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is Bulk-type). leaving an unconfigured endpoint control causes undefined behavior for the data pid tracking on the active endpoint/direction.

Address: 218_4000h base + 1D8h offset + (512d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								TXE	TXR	TXI	Reserved	TXT		TXD	TXS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



USB_*n*ENDPTCTRL6 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 TXE	TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22 TXR	TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled. This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
20 -	This field is reserved. Reserved
19–18 TXT	TX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
17 TXD	TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled

Table continues on the next page...

USB_*n*ENDPTCTRL6 field descriptions (continued)

Field	Description
	This bit will be cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
15–8 -	This field is reserved. Reserved
7 RXE	RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5 RXI	RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 -	This field is reserved. Reserved.
3–2 RXT	RX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Reserved
1 RXD	RX Endpoint Data Sink - Read/Write 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled

Table continues on the next page...

USB_*n*ENDPTCTRL6 field descriptions (continued)

Field	Description
	This bit is set automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpointand this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.

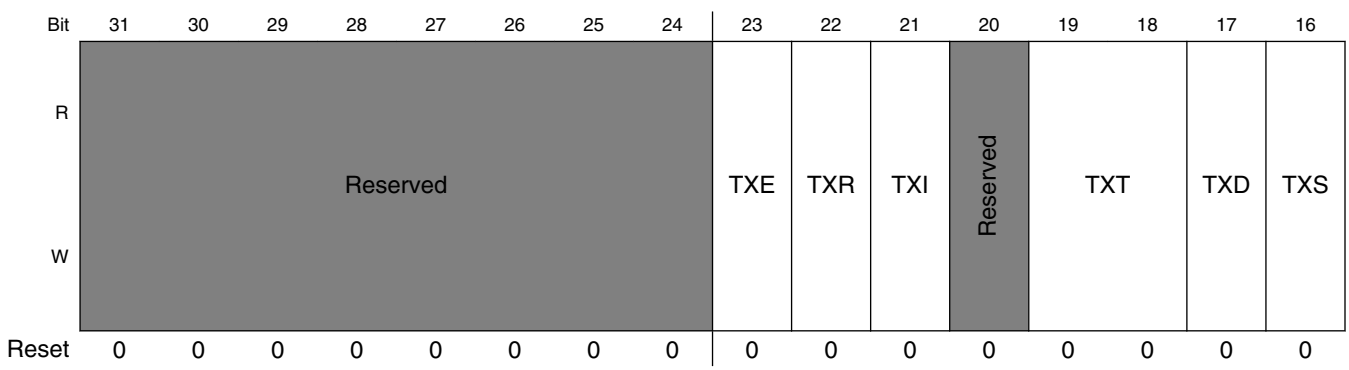
65.6.46 Endpoint Control 7 (USB_*n*ENDPTCTRL7)

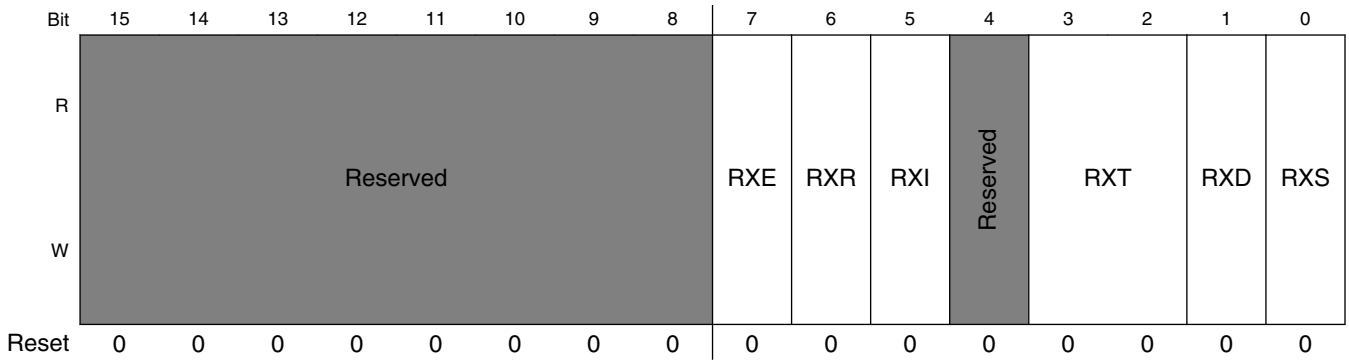
This is endpoint control register for endpoint 7 in device operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is Bulk-type). leaving an unconfigured endpoint control causes undefined behavior for the data pid tracking on the active endpoint/direction.

Address: 218_4000h base + 1DCh offset + (512d × i), where i=0d to 0d





USB_*n*ENDPTCTRL7 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 TXE	TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22 TXR	TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled. This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
20 -	This field is reserved. Reserved
19–18 TXT	TX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
17 TXD	TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled

Table continues on the next page...

USB_*n*ENDPTCTRL7 field descriptions (continued)

Field	Description
	This bit will be cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
15–8 -	This field is reserved. Reserved
7 RXE	RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5 RXI	RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 -	This field is reserved. Reserved.
3–2 RXT	RX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Reserved
1 RXD	RX Endpoint Data Sink - Read/Write 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled

Table continues on the next page...

USB_*n*ENDPTCTRL7 field descriptions (continued)

Field	Description
	This bit is set automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpointand this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.

Chapter 66
Universal Serial Bus 2.0 Integrated PHY (USB-PHY)

66.1 USB PHY Overview

The chip contains 2 integrated USB 2.0 PHY macrocells capable of connecting to USB host/device systems at the USB low-speed (LS) rate of 1.5 Mbits/s, full-speed (FS) rate of 12 Mbits/s or at the USB 2.0 high-speed (HS) rate of 480 Mbits/s.

See Figure 66-1 for a block diagram of the PHY. The integrated PHY provides a standard UTM interface. The USB_*n*_DN and USB_*n*_DP pins connect directly to a USB connector.

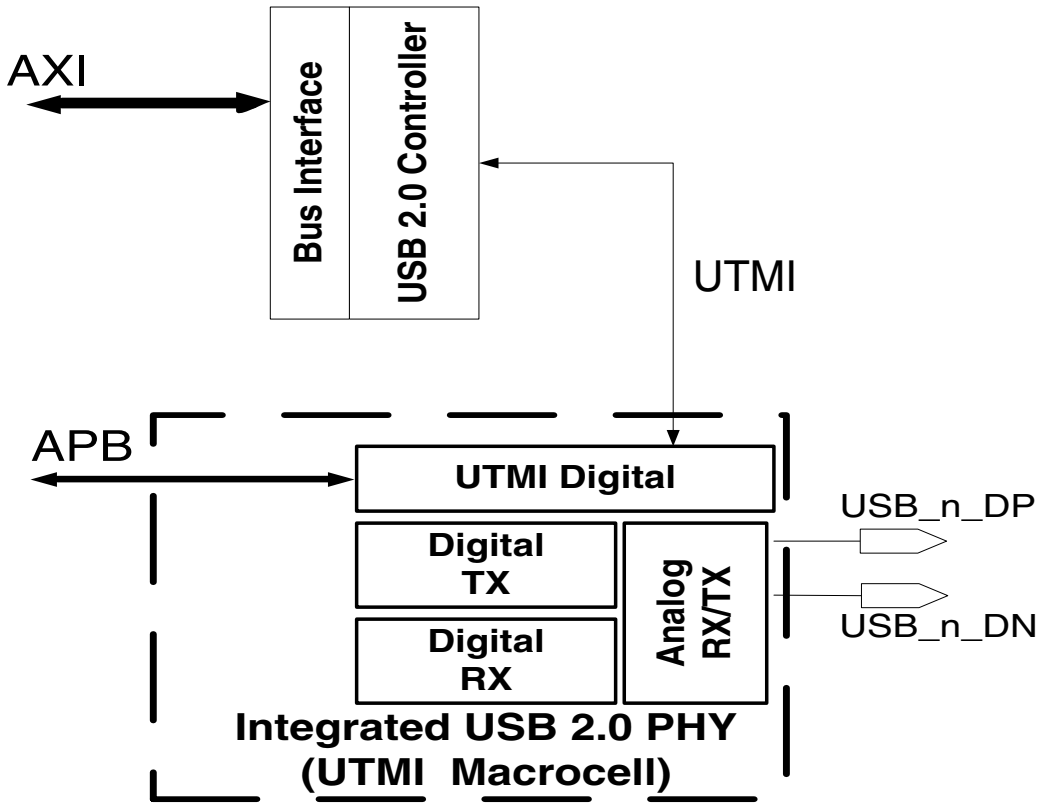


Figure 66-1. USB 2.0 PHY Block Diagram

USBPHY1 is the PHY interface for USB OTG controller; USBPHY2 is the PHY interface for USB Host1 controller.

The following subsections describe the external interfaces, internal interfaces, major blocks, and programmable registers that comprise the integrated USB 2.0 PHY.

66.2 Operation

The UTM provides a 16-bit interface to the USB controller. This interface is clocked at 30 MHz.

- The digital portions of the USBPHY block include the UTMI, digital transmitter, digital receiver, and the programmable registers.
- The analog transceiver section comprises an analog receiver and an analog transmitter, as shown in Figure 66-2.

66.2.1 UTMI

The UTMI block handles the line_state bits, reset buffering, suspend distribution, transceiver speed selection, and transceiver termination selection.

The PLL supplies a 120 MHz signal to all of the digital logic. The UTMI block does a final divide-by-four to develop the 30 MHz clock used in the interface.

66.2.2 Digital Transmitter

The digital transmitter receives the 16-bit transmit data from the USB controller and handles the tx_valid, tx_validh and tx_ready handshake.

In addition, it contains the transmit serializer that converts the 16-bit parallel words at 30 MHz to a single bitstream at 480 Mbit for high-speed or 12 Mbit for full-speed or 1.5 Mbit for low-speed. It does this while implementing the bit-stuffing algorithm and the NRZI encoder that are used to remove the DC component from the serial bitstream. The output of this encoder is sent to the low-speed (LS), full-speed (FS) or high-speed (HS) drivers in the analog transceiver section's transmitter block.

66.2.3 Digital Receiver

The digital receiver receives the raw serial bitstream from the low speed (LS) differential transceiver, full speed (FS) differential transceiver, and a 9X, 480 MHz sampled data from the high speed (HS) differential transceiver.

As the phase of the USB host transmitter shifts relative to the local PLL, the receiver section's HS DLL tracks these changes to give a reliable sample of the incoming 480 Mbit/s bitstream. Since this sample point shifts relative to the PLL phase used by the digital logic, a rate-matching elastic buffer is provided to cross this clock domain boundary. Once the bitstream is in the local clock domain, an NRZI decoder and bit unstuffer restore the original payload data bitstream and pass it to a deserializer and holding register. The receive state machine handles the rx_valid, rx_validh, and handshake with the USB controller. The handshake is not interlocked, in that there is no rx_ready signal coming from the controller. The controller must take each 16-bit value as presented by the PHY. The receive state machine provides an rx_active signal to the controller that indicates when it is inside a valid packet (SYNC detected, and so on).

66.2.4 Analog Receiver

The analog receiver comprises five differential receivers, two single-ended receivers, and a 9X, 480 MHz HS data sampling module

, as shown in the figure below and described further in this section.

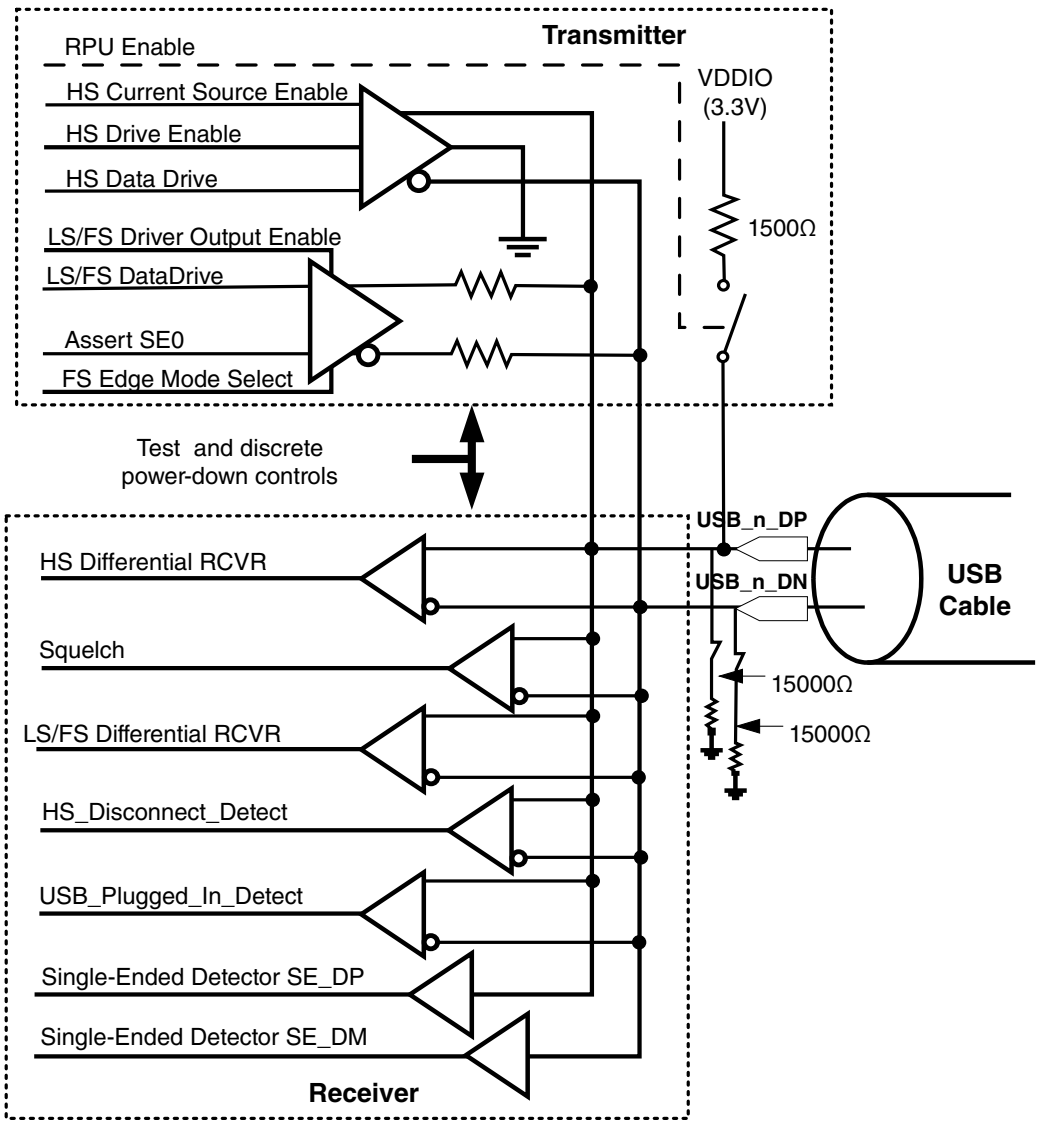


Figure 66-2. USB 2.0 PHY Analog Transceiver Block Diagram

66.2.4.1 HS Differential Receiver

The high-speed differential receiver is both a differential analog receiver and threshold comparator. Its output is a one if the differential signal is greater than a 0-V threshold.

Otherwise, its output is 0. Its purpose is to discriminate the ± 400-mV differential voltage resulting from the high-speed drivers current flow into the dual 45Ω terminations found on each pin of the differential pair. The envelope or squelch detector, described below, ensures that the differential signal has sufficient magnitude to be valid. The HS differential receiver tolerates up to 500 mV of common mode offset.

66.2.4.2 Squelch Detector

The squelch detector is a differential analog receiver and threshold comparator.

Its output is 1, if the differential magnitude is less than a nominal 100 mV threshold. Otherwise, its output is 0.

Its purpose is to invalidate the HS differential receiver when the incoming signal is simply too low to receive reliably.

66.2.4.3 LS/FS Differential Receiver

The low-speed/full-speed differential receiver is both a differential analog receiver and threshold comparator.

The crossover voltage falls between 1.3 V and 2.0 V. Its output is 1, when the USB_*n*_DP line is above the crossover point and the USB_*n*_DN line is below the crossover point. The digital receiver section decodes the receiver data into J or K state according to the speed.

66.2.4.4 HS Disconnect Detector

It is a differential analog receiver and threshold comparator. It outputs high when differential magnitude is greater than a nominal 575-mV threshold. Otherwise, it outputs low.

66.2.4.5 USB Plugged-In Detector

The USB plugged-in detector looks for both USB_*n*_DP and USB_*n*_DN to be high. There is a pair of large on-chip pullup resistors (200 KΩ) that hold both USB_*n*_DP and USB_*n*_DN high when the USB cable is not attached. The USB plugged-in detector signals a 0 in this case.

When operating in device mode, the upstream port in host/hub interface contains a 15 KΩ pulldown resistor which could easily override the 200 KΩ pullup resistor. When plugged in, at least one signal in the pair will be low, which will force the plugged-in detector's output high.

66.2.4.6 Single-Ended USB_DP Receiver

The single-ended USB_*n*_DP receiver output is high whenever the USB_*n*_DP input is above its nominal 1.8 V threshold.

66.2.4.7 Single-Ended USB_DN Receiver

The single-ended USB_*n*_DN receiver output is high whenever the USB_*n*_DN input is above its nominal 1.8 V threshold.

66.2.4.8 9X Oversample Module

The 9X oversample module uses nine identically spaced phases of the 480 MHz clock to sample a high speed bit data. The squelch signal is sampled only 1X.

66.2.5 Analog Transmitter

The analog transmitter comprises two differential drivers: one for high-speed signaling and one for full-speed signaling. It also contains the switchable 1.5 KΩ pullup resistor.

See Figure 66-2.

66.2.5.1 Switchable High-Speed 45Ω Termination Resistors

High-speed current mode differential signaling requires good 90 Ω differential termination at each end of the USB cable. This results from switching in 45 Ω terminating resistors from each signal line to ground at each end of the cable.

Because each signal is parallel terminated with 45 Ω at each end, each driver sees a 22.5 Ω load. This load impedance is much too low for full-speed signaling levels—hence the need for switchable high-speed terminating resistors. Switchable trimming resistors are provided to tune the actual termination resistance of each device, as shown in Figure

66-3. The HW_USBPHY_TX_TXCAL45DP bit field, for example, allows one of 16 trimming resistor values to be placed in parallel with the 45Ω terminator on the USB_*n*_DP signal.

66.2.5.2 Low-Speed/Full-Speed Differential Driver

The low-speed/full-speed differential drivers are essentially low-impedance pulldown devices that are switched in a differential mode for low-speed or full-speed signaling, that is, either one or the other device is turned on to signal the "J" state or the "K" state.

66.2.5.3 High-Speed Differential Driver

The high-speed differential driver receives a 17.78 mA current from the constant current source (Iref) and essentially steers it down either the USB_DP signal or the USB_DN signal or alternatively to ground.

This current will produce approximately a 400 mV drop across the 22.5 Ω termination seen by the driver when it is steered onto one of the signal lines. The approximately 17.78 mA current source is referenced back to the integrated voltage-band-gap (Vbg) circuit. The Iref, Ibias, and V to I circuits are shared with the integrated battery charger.

66.2.5.4 Switchable 1.5KΩ USB_DP Pullup Resistor

This product contains a switchable 1.5 KΩ pullup resistor on the USB_*n*_DP signal.

This resistor is switched on to indicate to the host/hub controller that a full-speed-capable device is on the USB cable, powered on, and ready. This resistor is switched off at power-on reset so the host does not recognize a USB device until the processor software enables the announcement of a full-speed device.

66.2.5.5 Switchable 15KΩ USB_DP Pulldown Resistor

This product contains a switchable 15 KΩ pulldown resistor on both USB_*n*_DP and USB_*n*_DN signals. This is used in host mode to indicate to the device controller that a host is present.

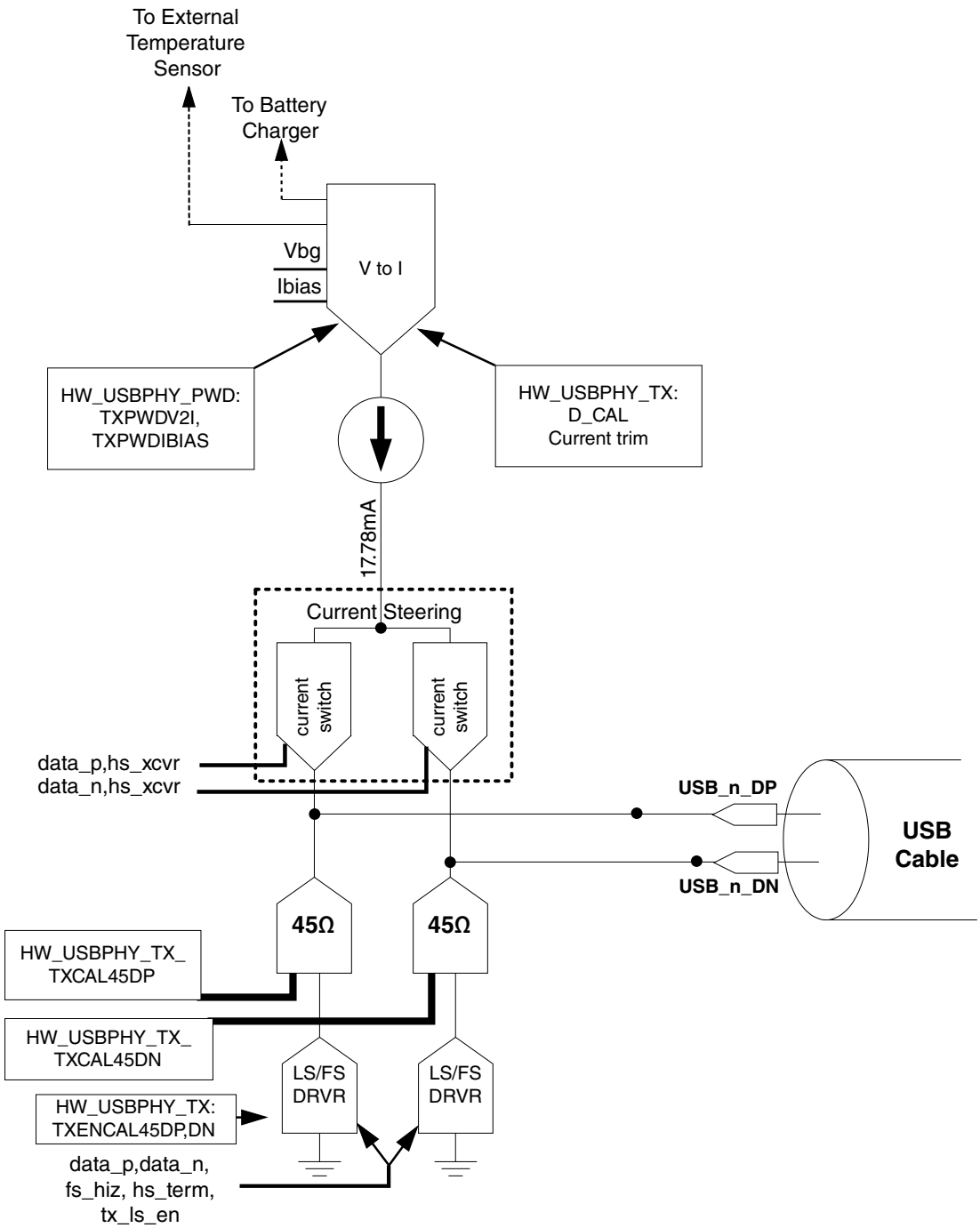


Figure 66-3. USB 2.0 PHY Transmitter Block Diagram

66.2.6 Recommended Register Configuration for USB Certification

The register settings in this section are recommended for passing USB certification.

The following settings lower the J/K levels to certifiable limits:

HW_USBPHY_TX_TXCAL45DP = 0x0
HW_USBPHY_TX_TXCAL45DN = 0x0
HW_USBPHY_TX_D_CAL = 0x7

66.2.7 Charger detection

The USB charger detector is a block that detects whether the upstream-facing device is connected to a down-stream facing charger, either a dedicated USB charger or a host charger.

The USB charger detector is comprised of two sub-blocks, namely the USB data-pin contact detector and the charger detector.

This section details those two sub-blocks and gives the software flow of USB charger detection. Finally, this chapter discusses the detection of a USB charger in case of a dead battery.

66.2.7.1 Charger detect control table

Before we dive into the details of the detectors, we show the logic table of the control signals to give the user an overall picture of the charger detector.

Table 66-1. Charger detection control table

EN_B	CHK_CHRG_B	CHK_CONTACT	Data pin contact detector	Charger detector
0	1	1	Enabled	Disabled
0	0	x(don't care)	Disabled	Enabled
1	x	x(don't care)	Disabled	Disabled

66.2.7.2 Data pin contact detector

According to Battery Charging Specification (rev 1.2), USB plugs and receptacles are designed such that when the plug is inserted into the receptacle, the power pins make contact before the data pins make contact. Therefore, there is inevitably a time interval during which USB_*n*_VBUS has been observed by the device while the USB_*n*_DP and USB_*n*_DN pins are not still pending for contact. The USB data pin contact detector is designed to give the software an indication of the contact of the data pins.

To enable the USB data pin contact detector, the user should set the CHK_CONTACT bit of the USB1_CHRG_DETECT register to 1 and monitor the PLUG_CONTACT bit status of the USB1_CHRG_DETECT_STAT register. If PLUG_CONTACT is 1, then it indicates that the data pins have make good contacts, otherwise the user should continue to wait until this bit is set.

According to Table 1, it should be noted that the data pin contact detector only works when EN_B=0 and CHK_CHRG_B=1, both bit being of the USB1_CHRG_DETECT register.

66.2.7.3 Charger detector

Once the data pins make contact, the user should enable the charger detector by clearing the CHK_CHRG_B bit that is low-active. Then the user should wait for 40ms and then check the status bit of CHRG_DETECTED in register hw_anadig_usb1_chrg_det_stat. CHRG_DETECTED=1 means that the device is connected to a charger, either a dedicated charger or a charging downstream port (or equivalently called a host charger, or charging host). To further differentiate between a host charger and a dedicated charger, the user is suggested to pull up USB_*n*_DP signaling a connect event to the host. Then the user should monitor the USB_*n*_DN line status. If USB_*n*_DN=1, then the charger is a dedicated charger; if USB_*n*_DN=0, then it is a host charger.

66.2.7.4 Charger detection software flow

Upon seeing VBUS, the software should follow the software flow for the charger detection process. The flow chart mentions the "enable the vdd3p0 current limiter". Please refer to the power chapter for details.

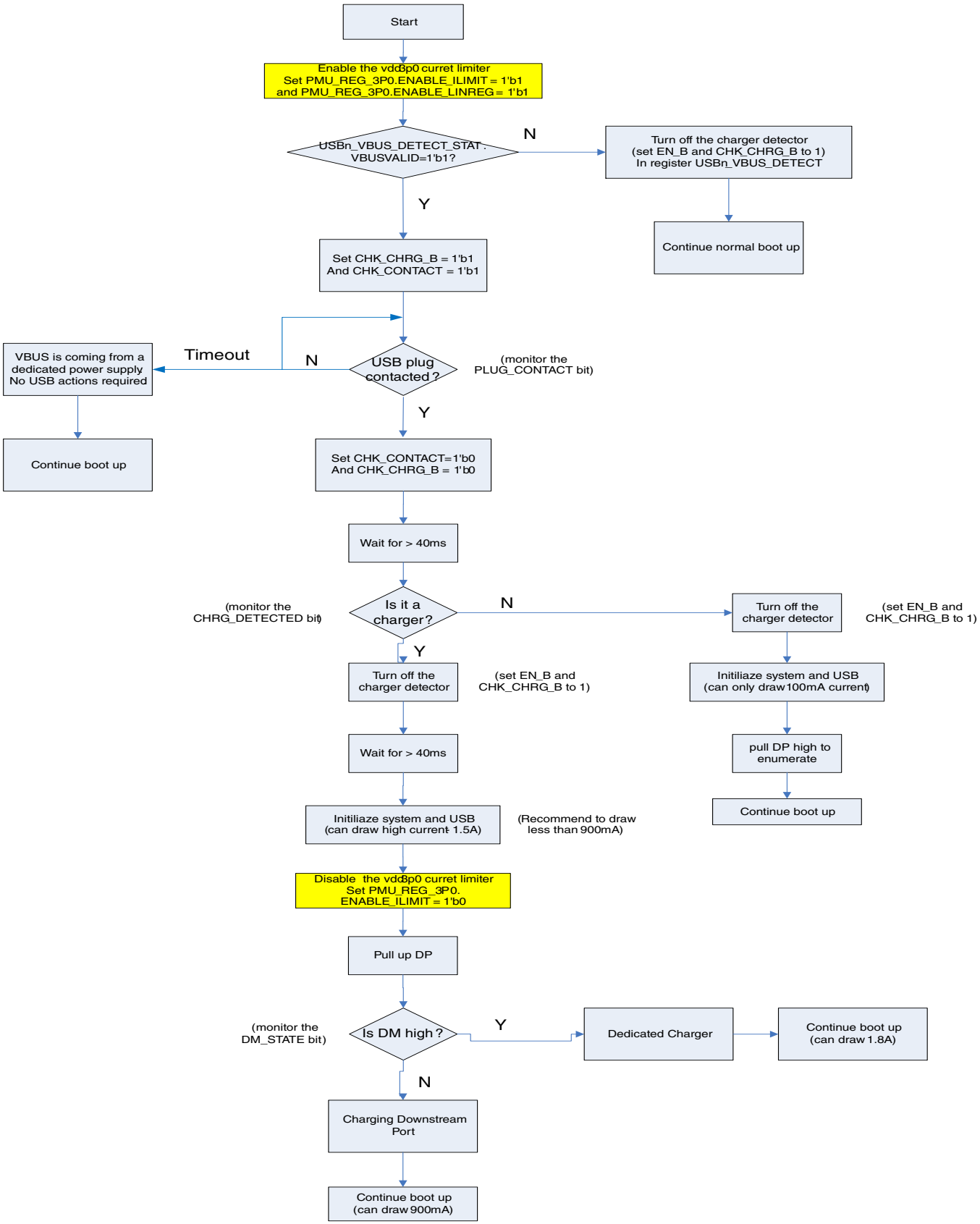


Figure 66-4. USBPHY Charger Detection Software Flow

66.2.7.5 Dead Battery Protect

All the descriptions above are based on the assumption that all the power supplies have been on when the device is plugged into a remote host (or charger). However, there are cases when the local battery of the portable device has been so depleted that the system could not be turned on. In such scenarios the user may prefer a method of signaling the external power management unit (PMIC) the existence of the USB charger to draw a current larger than 100mA from the remote host to speed up system boot up or battery charging. The charger detector indeed supports this function.

When we have a fully depleted battery, all the power supplies might be off. Upon insertion of the 5V, the supplies are brought up by the external PMIC and the internal regulators. Due to the 100mA inrush current limit of the USB spec, we cannot draw larger than 100mA current which might be a limit for system boot-up. Since by default, EN_B=0, CHK_CHRG_B=0 and CHK_CONTACT=1, the usb charger detector is automatically enabled without any software operation needed and it can signal the external PMIC the existence of a USB charger through the open-drain output pin USB_OTG_CHD_B. This pin should be pulled up to an external voltage that is acceptable to the PMIC. If this signal is low, then the PMIC can get that the device is connected to a charger. In this case, the PMIC can draw more than 100mA current from the USB.

It should be noted that this function requires cooperation between the chip and the external PMIC. It is suggested that the user consult NXP for such use cases.

66.3 USB PHY Memory Map/Register Definition

USBPHY Hardware Register Format Summary

USBPHY memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_9000	USB PHY Power-Down Register (USBPHY1_PWD)	32	R/W	001E_1C00h	66.3.1/5497
20C_9004	USB PHY Power-Down Register (USBPHY1_PWD_SET)	32	R/W	001E_1C00h	66.3.1/5497
20C_9008	USB PHY Power-Down Register (USBPHY1_PWD_CLR)	32	R/W	001E_1C00h	66.3.1/5497
20C_900C	USB PHY Power-Down Register (USBPHY1_PWD_TOG)	32	R/W	001E_1C00h	66.3.1/5497
20C_9010	USB PHY Transmitter Control Register (USBPHY1_TX)	32	R/W	1006_0607h	66.3.2/5499

Table continues on the next page...

USBPHY memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_9014	USB PHY Transmitter Control Register (USBPHY1_TX_SET)	32	R/W	1006_0607h	66.3.2/5499
20C_9018	USB PHY Transmitter Control Register (USBPHY1_TX_CLR)	32	R/W	1006_0607h	66.3.2/5499
20C_901C	USB PHY Transmitter Control Register (USBPHY1_TX_TOG)	32	R/W	1006_0607h	66.3.2/5499
20C_9020	USB PHY Receiver Control Register (USBPHY1_RX)	32	R/W	0000_0000h	66.3.3/5500
20C_9024	USB PHY Receiver Control Register (USBPHY1_RX_SET)	32	R/W	0000_0000h	66.3.3/5500
20C_9028	USB PHY Receiver Control Register (USBPHY1_RX_CLR)	32	R/W	0000_0000h	66.3.3/5500
20C_902C	USB PHY Receiver Control Register (USBPHY1_RX_TOG)	32	R/W	0000_0000h	66.3.3/5500
20C_9030	USB PHY General Control Register (USBPHY1_CTRL)	32	R/W	C020_0000h	66.3.4/5502
20C_9034	USB PHY General Control Register (USBPHY1_CTRL_SET)	32	R/W	C020_0000h	66.3.4/5502
20C_9038	USB PHY General Control Register (USBPHY1_CTRL_CLR)	32	R/W	C020_0000h	66.3.4/5502
20C_903C	USB PHY General Control Register (USBPHY1_CTRL_TOG)	32	R/W	C020_0000h	66.3.4/5502
20C_9040	USB PHY Status Register (USBPHY1_STATUS)	32	R/W	0000_0000h	66.3.5/5505
20C_9050	USB PHY Debug Register (USBPHY1_DEBUG)	32	R/W	7F18_0000h	66.3.6/5507
20C_9054	USB PHY Debug Register (USBPHY1_DEBUG_SET)	32	R/W	7F18_0000h	66.3.6/5507
20C_9058	USB PHY Debug Register (USBPHY1_DEBUG_CLR)	32	R/W	7F18_0000h	66.3.6/5507
20C_905C	USB PHY Debug Register (USBPHY1_DEBUG_TOG)	32	R/W	7F18_0000h	66.3.6/5507
20C_9060	UTMI Debug Status Register 0 (USBPHY1_DEBUG0_STATUS)	32	R	0000_0000h	66.3.7/5509
20C_9070	UTMI Debug Status Register 1 (USBPHY1_DEBUG1)	32	R/W	0000_1000h	66.3.8/5510
20C_9074	UTMI Debug Status Register 1 (USBPHY1_DEBUG1_SET)	32	R/W	0000_1000h	66.3.8/5510
20C_9078	UTMI Debug Status Register 1 (USBPHY1_DEBUG1_CLR)	32	R/W	0000_1000h	66.3.8/5510
20C_907C	UTMI Debug Status Register 1 (USBPHY1_DEBUG1_TOG)	32	R/W	0000_1000h	66.3.8/5510
20C_9080	UTMI RTL Version (USBPHY1_VERSION)	32	R	0402_0000h	66.3.9/5511
20C_A000	USB PHY Power-Down Register (USBPHY2_PWD)	32	R/W	001E_1C00h	66.3.1/5497
20C_A004	USB PHY Power-Down Register (USBPHY2_PWD_SET)	32	R/W	001E_1C00h	66.3.1/5497
20C_A008	USB PHY Power-Down Register (USBPHY2_PWD_CLR)	32	R/W	001E_1C00h	66.3.1/5497
20C_A00C	USB PHY Power-Down Register (USBPHY2_PWD_TOG)	32	R/W	001E_1C00h	66.3.1/5497
20C_A010	USB PHY Transmitter Control Register (USBPHY2_TX)	32	R/W	1006_0607h	66.3.2/5499
20C_A014	USB PHY Transmitter Control Register (USBPHY2_TX_SET)	32	R/W	1006_0607h	66.3.2/5499
20C_A018	USB PHY Transmitter Control Register (USBPHY2_TX_CLR)	32	R/W	1006_0607h	66.3.2/5499
20C_A01C	USB PHY Transmitter Control Register (USBPHY2_TX_TOG)	32	R/W	1006_0607h	66.3.2/5499
20C_A020	USB PHY Receiver Control Register (USBPHY2_RX)	32	R/W	0000_0000h	66.3.3/5500

Table continues on the next page...

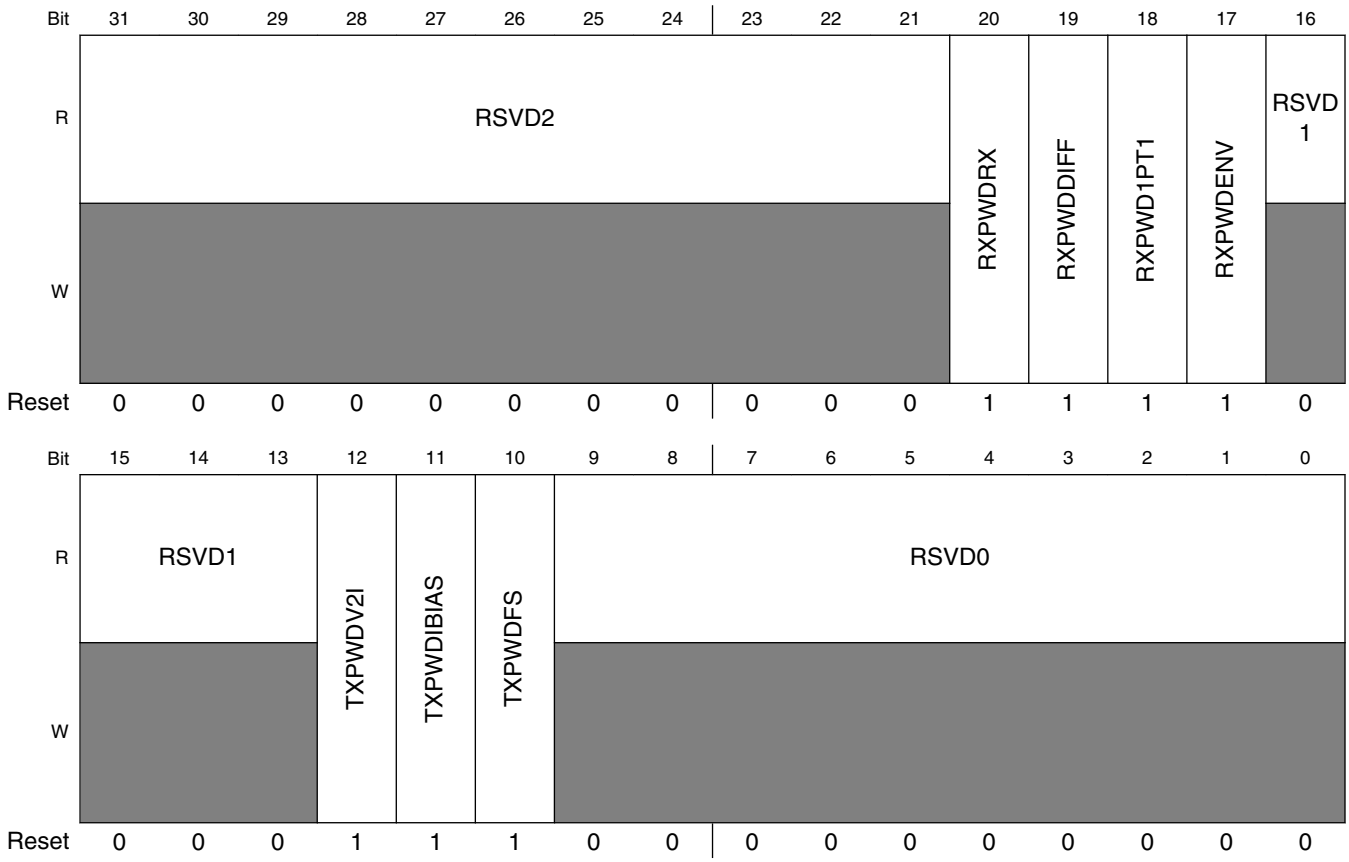
USBPHY memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_A024	USB PHY Receiver Control Register (USBPHY2_RX_SET)	32	R/W	0000_0000h	66.3.3/5500
20C_A028	USB PHY Receiver Control Register (USBPHY2_RX_CLR)	32	R/W	0000_0000h	66.3.3/5500
20C_A02C	USB PHY Receiver Control Register (USBPHY2_RX_TOG)	32	R/W	0000_0000h	66.3.3/5500
20C_A030	USB PHY General Control Register (USBPHY2_CTRL)	32	R/W	C020_0000h	66.3.4/5502
20C_A034	USB PHY General Control Register (USBPHY2_CTRL_SET)	32	R/W	C020_0000h	66.3.4/5502
20C_A038	USB PHY General Control Register (USBPHY2_CTRL_CLR)	32	R/W	C020_0000h	66.3.4/5502
20C_A03C	USB PHY General Control Register (USBPHY2_CTRL_TOG)	32	R/W	C020_0000h	66.3.4/5502
20C_A040	USB PHY Status Register (USBPHY2_STATUS)	32	R/W	0000_0000h	66.3.5/5505
20C_A050	USB PHY Debug Register (USBPHY2_DEBUG)	32	R/W	7F18_0000h	66.3.6/5507
20C_A054	USB PHY Debug Register (USBPHY2_DEBUG_SET)	32	R/W	7F18_0000h	66.3.6/5507
20C_A058	USB PHY Debug Register (USBPHY2_DEBUG_CLR)	32	R/W	7F18_0000h	66.3.6/5507
20C_A05C	USB PHY Debug Register (USBPHY2_DEBUG_TOG)	32	R/W	7F18_0000h	66.3.6/5507
20C_A060	UTMI Debug Status Register 0 (USBPHY2_DEBUG0_STATUS)	32	R	0000_0000h	66.3.7/5509
20C_A070	UTMI Debug Status Register 1 (USBPHY2_DEBUG1)	32	R/W	0000_1000h	66.3.8/5510
20C_A074	UTMI Debug Status Register 1 (USBPHY2_DEBUG1_SET)	32	R/W	0000_1000h	66.3.8/5510
20C_A078	UTMI Debug Status Register 1 (USBPHY2_DEBUG1_CLR)	32	R/W	0000_1000h	66.3.8/5510
20C_A07C	UTMI Debug Status Register 1 (USBPHY2_DEBUG1_TOG)	32	R/W	0000_1000h	66.3.8/5510
20C_A080	UTMI RTL Version (USBPHY2_VERSION)	32	R	0402_0000h	66.3.9/5511

66.3.1 USB PHY Power-Down Register (USBPHYx_PWD*n*)

The USB PHY Power-Down Register provides overall control of the PHY power state. Before programming this register, the PHY clocks must be enabled in registers USBPHYx_CTRLn and CCM_ANALOG_USBPHYx_PLL_480_CTRLn.

Address: Base address + 0h offset + (4d × i), where i=0d to 3d



USBPHYx_PWD*n* field descriptions

Field	Description
31–21 RSVD2	Reserved.
20 RXPWDRX	0 = Normal operation. 1 = Power-down the entire USB PHY receiver block except for the full-speed differential receiver. Note that this bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of USBPHYx_CTRL is enabled.
19 RXPWDDIFF	0 = Normal operation. 1 = Power-down the USB high-speed differential receiver.

Table continues on the next page...

USBPHYx_PWD*n* field descriptions (continued)

Field	Description
	Note that this bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of USBPHYx_CTRL is enabled.
18 RXPWD1PT1	0 = Normal operation. 1 = Power-down the USB full-speed differential receiver. Note that this bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of USBPHYx_CTRL is enabled.
17 RXPWDENV	0 = Normal operation. 1 = Power-down the USB high-speed receiver envelope detector (squelch signal). Note that this bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of USBPHYx_CTRL is enabled.
16–13 RSVD1	Reserved.
12 TXPWDV2I	0 = Normal operation. 1 = Power-down the USB PHY transmit V-to-I converter and the current mirror. Note that this bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of USBPHYx_CTRL is enabled. Note that these circuits are shared with the battery charge circuit. Setting this to 1 does not power-down these circuits, unless the corresponding bit in the battery charger is also set for power-down.
11 TXPWDIBIAS	0 = Normal operation. 1 = Power-down the USB PHY current bias block for the transmitter. This bit should be set only when the USB is in suspend mode. This effectively powers down the entire USB transmit path. Note that this bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of USBPHYx_CTRL is enabled. Note that these circuits are shared with the battery charge circuit. Setting this bit to 1 does not power-down these circuits, unless the corresponding bit in the battery charger is also set for power-down.
10 TXPWDFS	0 = Normal operation. 1 = Power-down the USB full-speed drivers. This turns off the current starvation sources and puts the drivers into high-impedance output. Note that this bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of USBPHYx_CTRL is enabled.
RSVD0	Reserved.

66.3.2 USB PHY Transmitter Control Register (USBPHY*x*_TX*n*)

The USB PHY Transmitter Control Register handles the transmit controls.

Address: Base address + 10h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVD5			USBPHY_TX_ EDGECTRL			RSVD2						TXCAL45DP			
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD1				TXCAL45DN				RSVD0				D_CAL			
W																
Reset	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1

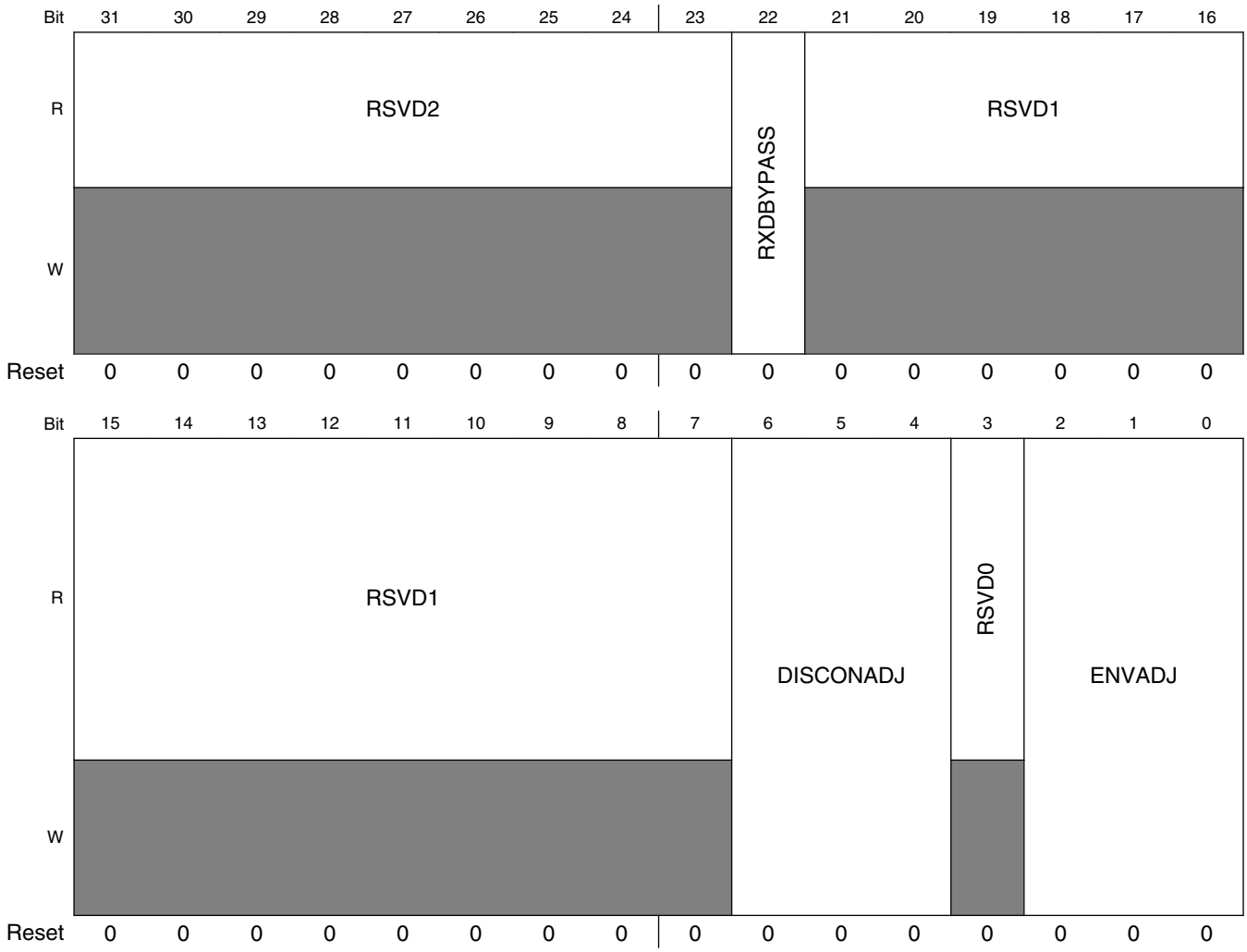
USBPHY*x*_TX*n* field descriptions

Field	Description
31–29 RSVD5	Reserved.
28–26 USBPHY_TX_ EDGECTRL	Controls the edge-rate of the current sensing transistors used in HS transmit. NOT FOR CUSTOMER USE.
25–20 RSVD2	Reserved.
19–16 TXCAL45DP	Decode to select a 45-Ohm resistance to the USB_DP output pin. Maximum resistance = 0000. Resistance is centered by design at 0110.
15–12 RSVD1	Reserved. **Note:** This bit should remain clear.
11–8 TXCAL45DN	Decode to select a 45-Ohm resistance to the USB_DN output pin. Maximum resistance = 0000. Resistance is centered by design at 0110.
7–4 RSVD0	Reserved. **Note:** This bit should remain clear.
D_CAL	Resistor Trimming Code: 0000 = 0.16% 0111 = Nominal 1111 = +25%

66.3.3 USB PHY Receiver Control Register (USBPHY*x*_RX*n*)

The USB PHY Receiver Control Register handles receive path controls.

Address: Base address + 20h offset + (4d × i), where i=0d to 3d



USBPHY*x*_RX*n* field descriptions

Field	Description
31–23 RSVD2	Reserved.
22 RXDBYPASS	0 = Normal operation. 1 = Use the output of the USB_DP single-ended receiver in place of the full-speed differential receiver. This test mode is intended for lab use only.
21–7 RSVD1	Reserved.

Table continues on the next page...

USBPHY*x*_RX*n* field descriptions (continued)

Field	Description
6–4 DISCONADJ	The DISCONADJ field adjusts the trip point for the disconnect detector: 000 = Trip-Level Voltage is 0.57500 V 001 = Trip-Level Voltage is 0.56875 V 010 = Trip-Level Voltage is 0.58125 V 011 = Trip-Level Voltage is 0.58750 V 1XX = Reserved
3 RSVD0	Reserved.
ENVADJ	The ENVADJ field adjusts the trip point for the envelope detector. 000 = Trip-Level Voltage is 0.12500 V 001 = Trip-Level Voltage is 0.10000 V 010 = Trip-Level Voltage is 0.13750 V 011 = Trip-Level Voltage is 0.15000 V 1XX = Reserved

66.3.4 USB PHY General Control Register (USBPHYx_CTRLn)

The USB PHY General Control Register handles OTG and Host controls. This register also includes interrupt enables and connectivity detect enables and results.

Address: Base address + 30h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SFTRST	CLKGATE	UTMI_SUSPENDM	HOST_FORCE_LS_SE0	OTG_ID_VALUE	RSVD1		FSDLL_RST_EN	ENVBUSCHG_WKUP	ENIDCHG_WKUP	ENDPDMCHG_WKUP	ENAUTOCLR_PHY_PWD	ENAUTOCLR_CLKGATE	RSVD0	WAKEUP_IRQ	ENIRQWAKEUP
W																
Reset	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ENUTMILEVEL3	ENUTMILEVEL2	DATA_ON_LRADC	DEVPLUGIN_IRQ	ENIRQDEVPLUGIN	RESUME_IRQ	ENIRQRESUMEDETECT	RESUMEIRQSTICKY	ENOTGIDDETECT	OTG_ID_CHG_IRQ	DEVPLUGIN_POLARITY	ENDEVPLUGINDETECT	HOSTDISCONDETECT_IRQ	ENIRQHOSTDISCON	ENHOSTDISCONDETECT	ENOTG_ID_CHG_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBPHYx_CTRL*n* field descriptions

Field	Description
31 SFTRST	Writing a 1 to this bit will soft-reset the USBPHYx_PWD, USBPHYx_TX, USBPHYx_RX, and USBPHYx_CTRL registers. Set to 0 to release the PHY from reset.
30 CLKGATE	Gate UTMI Clocks. Clear to 0 to run clocks. Set to 1 to gate clocks. Set this to save power while the USB is not actively being used. Configuration state is kept while the clock is gated. Note this bit can be auto-cleared if there is any wakeup event when USB is suspended while ENAUTOCLR_CLKGATE bit of USBPHYx_CTRL is enabled.
29 UTMI_SUSPENDM	Used by the PHY to indicate a powered-down state. If all the power-down bits in the USBPHYx_PWD are enabled, UTMI_SUSPENDM will be 0, otherwise 1. UTMI_SUSPENDM is negative logic, as required by the UTMI specification.
28 HOST_FORCE_LS_SE0	Forces the next FS packet that is transmitted to have a EOP with LS timing. This bit is used in host mode for the resume sequence. After the packet is transferred, this bit is cleared. The design can use this function to force the LS SE0 or use the USBPHYx_CTRL_UTMI_SUSPENDM to trigger this event when leaving suspend. This bit is used in conjunction with USBPHYx_DEBUG_HOST_RESUME_DEBUG.
27 OTG_ID_VALUE	Almost same as OTGID_STATUS in USBPHYx_STATUS Register. The only difference is that OTG_ID_VALUE has debounce logic to filter the glitches on ID Pad.
26–25 RSVD1	Reserved.
24 FSDLL_RST_EN	Enables the feature to reset the FSDLL lock detection logic at the end of each TX packet.
23 ENVBUSCHG_WKUP	Enables the feature to wakeup USB if VBUS is toggled when USB is suspended.
22 ENIDCHG_WKUP	Enables the feature to wakeup USB if ID is toggled when USB is suspended.
21 ENDPDMCHG_WKUP	Enables the feature to wakeup USB if DP/DM is toggled when USB is suspended. This bit is enabled by default.
20 ENAUTOCLR_PHY_PWD	Enables the feature to auto-clear the PWD register bits in USBPHYx_PWD if there is wakeup event while USB is suspended. This should be enabled if needs to support auto wakeup without S/W's interaction.
19 ENAUTOCLR_CLKGATE	Enables the feature to auto-clear the CLKGATE bit if there is wakeup event while USB is suspended. This should be enabled if needs to support auto wakeup without S/W's interaction.
18 RSVD0	Reserved.
17 WAKEUP_IRQ	Indicates that there is a wakeup event. Reset this bit by writing a 1 to the clear address space and not by a general write.
16 ENIRQWAKEUP	Enables interrupt for the wakeup events.
15 ENUTMILEVEL3	Enables UTMI+ Level3. This should be enabled if needs to support external FS Hub with LS device connected
14 ENUTMILEVEL2	Enables UTMI+ Level2. This should be enabled if needs to support LS device
13 DATA_ON_LRADC	Enables the LRADC to monitor USB_DP and USB_DM. This is for use in non-USB modes only.
12 DEVPLUGIN_IRQ	Indicates that the device is connected. Reset this bit by writing a 1 to the clear address space and not by a general write.
11 ENIRQDEVPLUGIN	Enables interrupt for the detection of connectivity to the USB line.

Table continues on the next page...

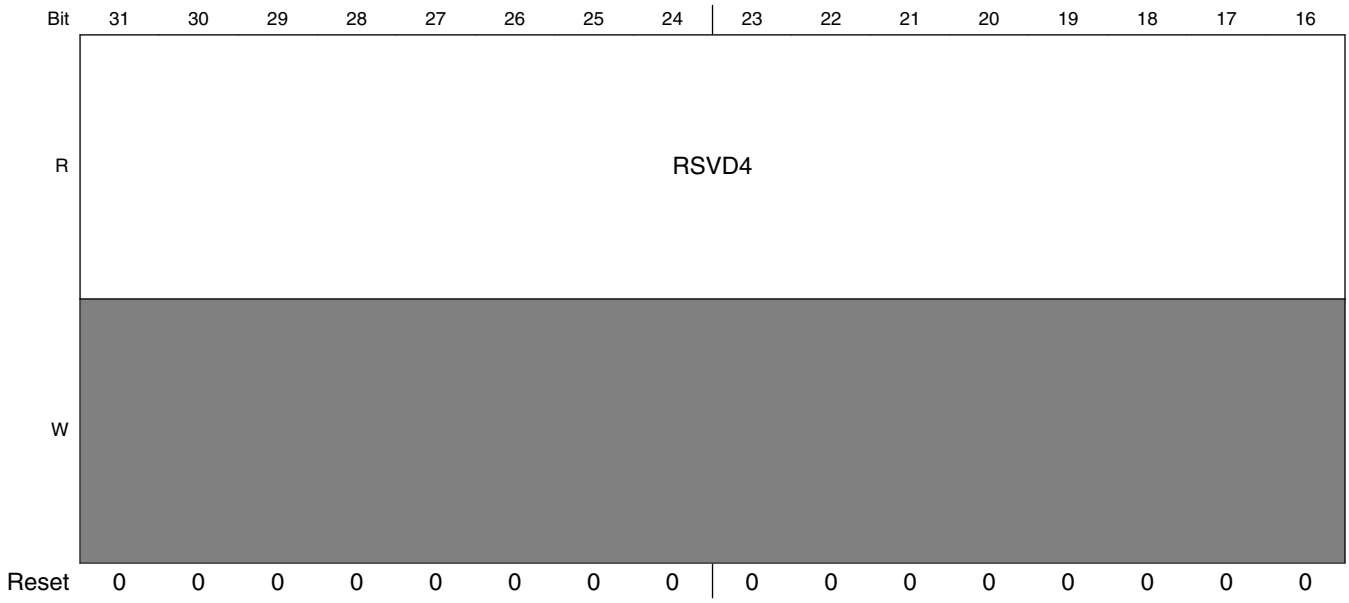
USBPHY*x*_CTRL*n* field descriptions (continued)

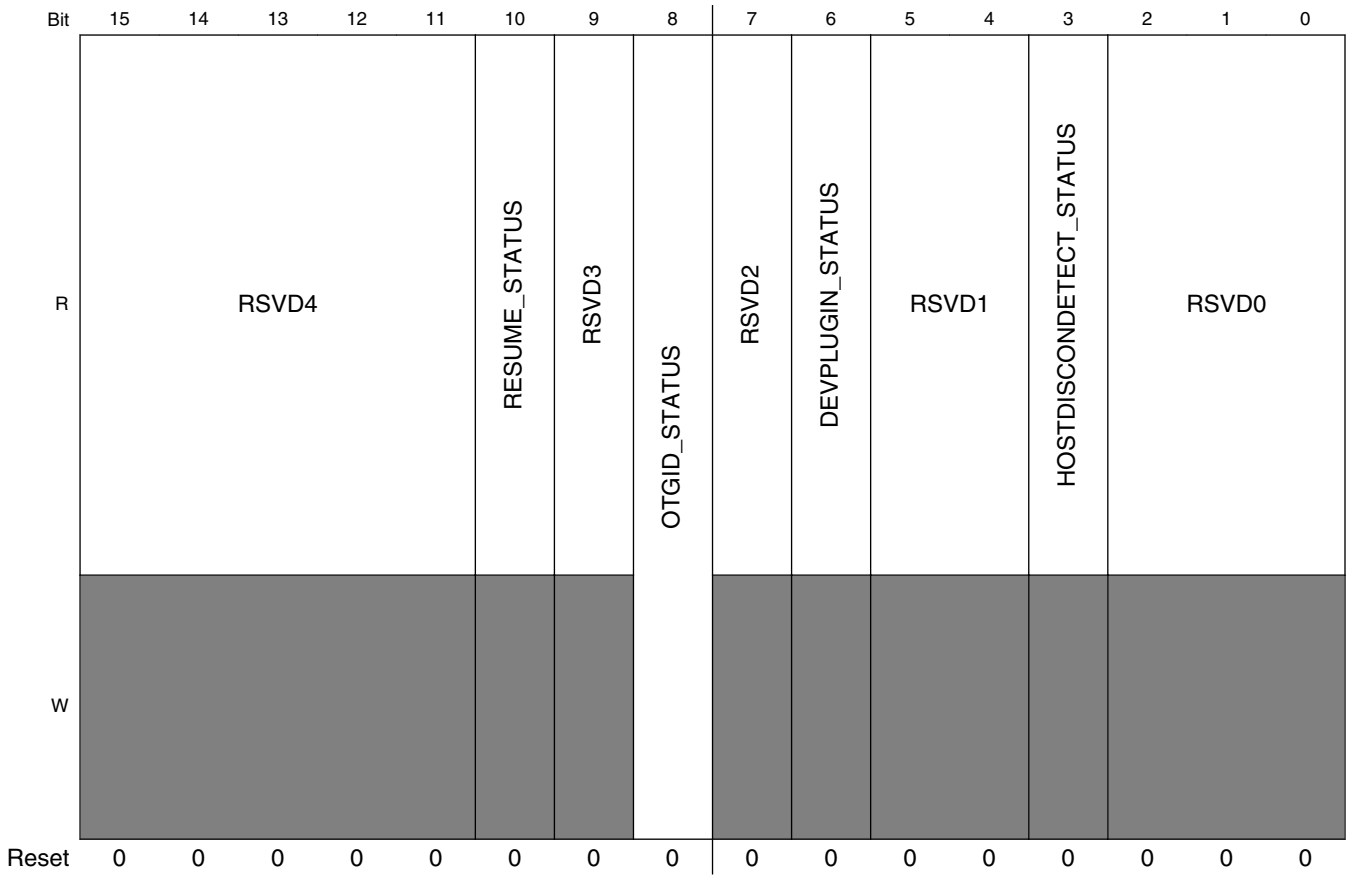
Field	Description
10 RESUME_IRQ	Indicates that the host is sending a wake-up after suspend. This bit is also set on a reset during suspend. Use this bit to wake up from suspend for either the resume or the reset case. Reset this bit by writing a 1 to the clear address space and not by a general write.
9 ENIRQRESUMEDETECT	Enables interrupt for detection of a non-J state on the USB line. This should only be enabled after the device has entered suspend mode.
8 RESUMEIRQSTICKY	Set to 1 will make RESUME_IRQ bit a sticky bit until software clear it. Set to 0, RESUME_IRQ only set during the wake-up period.
7 ENOTGIDDETECT	Enables circuit to detect resistance of MiniAB ID pin.
6 OTG_ID_CHG_IRQ	OTG ID change interrupt. Indicates the value of ID pin changed.
5 DEVPLUGIN_POLARITY	For device mode, if this bit is cleared to 0, then it trips the interrupt if the device is plugged in. If set to 1, then it trips the interrupt if the device is unplugged.
4 ENDEVPLUGINDETECT	For device mode, enables 200-KOhm pullups for detecting connectivity to the host.
3 HOSTDISCONDETECT_ IRQ	Indicates that the device has disconnected in high-speed mode. Reset this bit by writing a 1 to the clear address space and not by a general write.
2 ENIRQHOSTDISCON	Enables interrupt for detection of disconnection to Device when in high-speed host mode. This should be enabled after ENDEVPLUGINDETECT is enabled.
1 ENHOSTDISCONDETECT	For host mode, enables high-speed disconnect detector. This signal allows the override of enabling the detection that is normally done in the UTMI controller. The UTMI controller enables this circuit whenever the host sends a start-of-frame packet. SW shall set this bit when it found the high-speed device is connected, suggested during bus reset, after found high-speed device in USB_PORTSC1.PSPD). SW shall make sure this bit is not set at the end of resume, otherwise a wrong disconnect status may be detected. Suggest clear it after set USB_PORTSC1.SUSP, set it again after resume is ended(USB_PORTSC1.FPR==0).
0 ENOTG_ID_CHG_IRQ	Enable OTG_ID_CHG_IRQ.

66.3.5 USB PHY Status Register (USBPHYx_STATUS)

The USB PHY Status Register holds results of IRQ and other detects.

Address: Base address + 40h offset





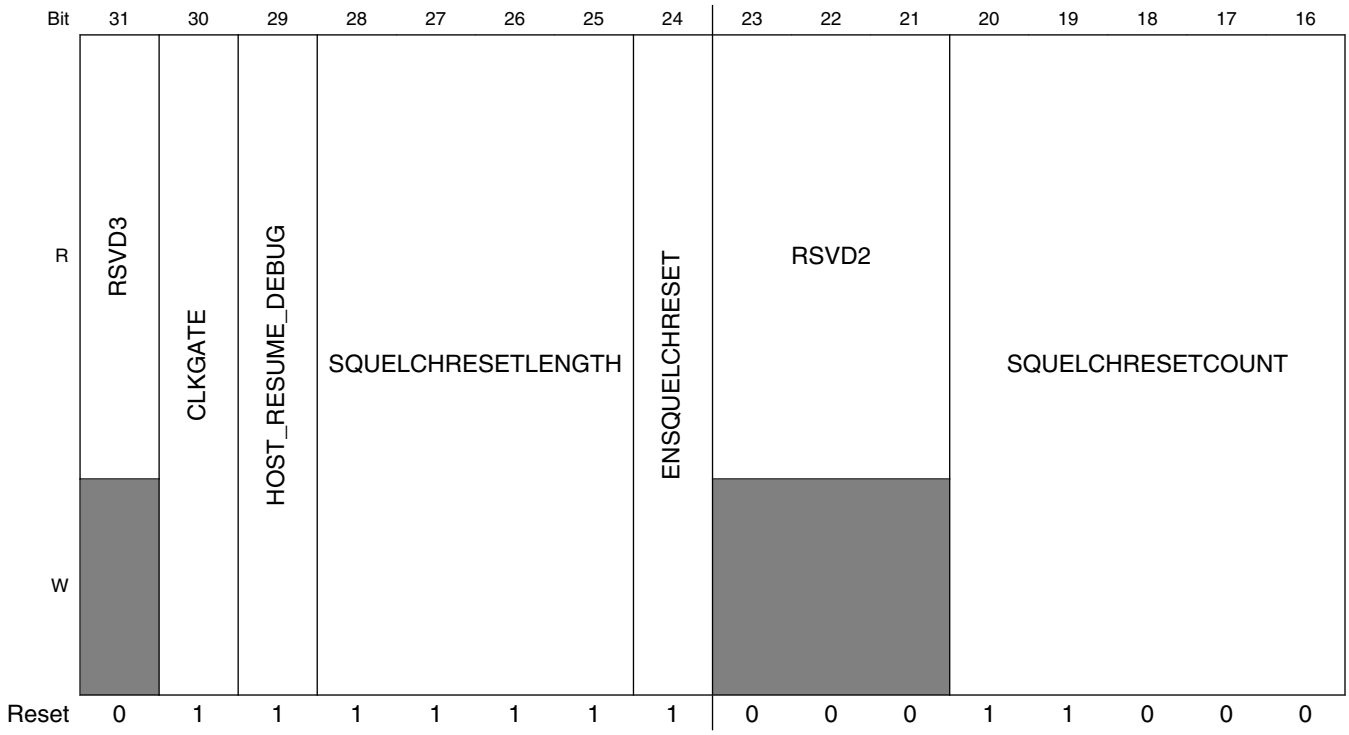
USBPHYx_STATUS field descriptions

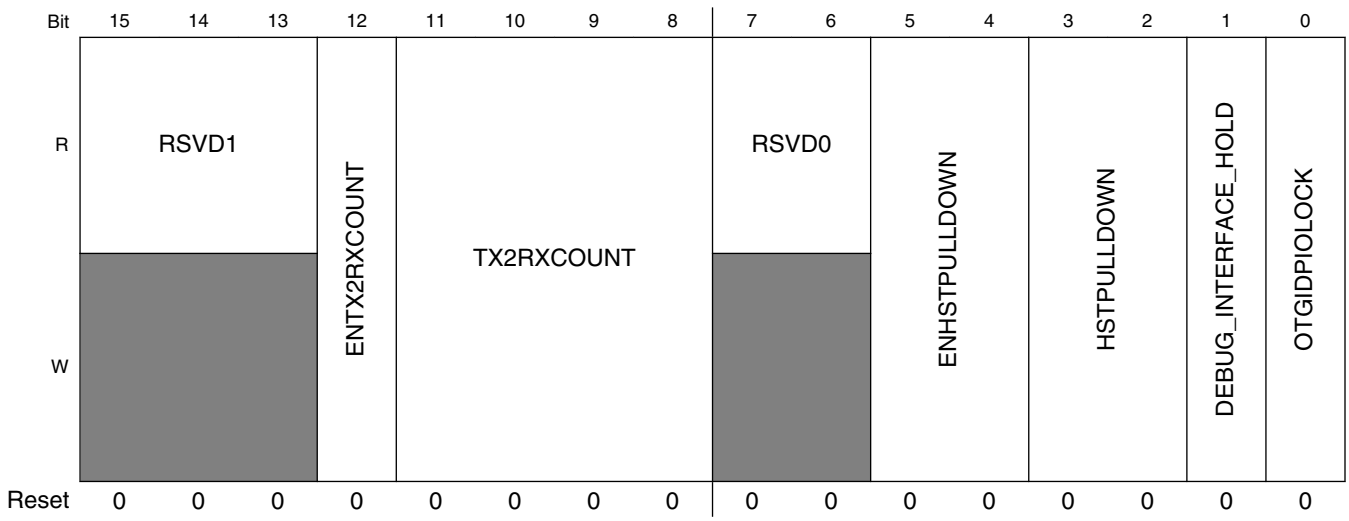
Field	Description
31–11 RSVD4	Reserved.
10 RESUME_STATUS	Indicates that the host is sending a wake-up after suspend and has triggered an interrupt.
9 RSVD3	Reserved.
8 OTGID_STATUS	Indicates the results of ID pin on MiniAB plug. False (0) is when ID resistance is less than Ra_Plug_ID, indicating host (A) side. True (1) is when ID resistance is greater than Rb_Plug_ID, indicating device (B) side.
7 RSVD2	Reserved.
6 DEVPLUGIN_STATUS	Indicates that the device has been connected on the USB_DP and USB_DM lines.
5–4 RSVD1	Reserved.
3 HOSTDISCONDETECT_ STATUS	Indicates that the device has disconnected while in high-speed host mode.
RSVD0	Reserved.

66.3.6 USB PHY Debug Register (USBPHY*x*_DEBUG*n*)

This register is used to debug the USB PHY.

Address: Base address + 50h offset + (4d × i), where i=0d to 3d





USBPHYx_DEBUG*n* field descriptions

Field	Description
31 RSVD3	Reserved.
30 CLKGATE	Gate Test Clocks. Clear to 0 for running clocks. Set to 1 to gate clocks. Set this to save power while the USB is not actively being used. Configuration state is kept while the clock is gated.
29 HOST_RESUME_DEBUG	Choose to trigger the host resume SE0 with HOST_FORCE_LS_SE0 = 0 or UTMI_SUSPEND = 1.
28–25 SQUELCHRESETLENGTH	Duration of RESET in terms of the number of 480-MHz cycles.
24 ENSQUELCHRESET	Set bit to allow squelch to reset high-speed receive.
23–21 RSVD2	Reserved.
20–16 SQUELCHRESETCOUNT	Delay in between the detection of squelch to the reset of high-speed RX.
15–13 RSVD1	Reserved.
12 ENTX2RXCOUNT	Set this bit to allow a countdown to transition in between TX and RX.
11–8 TX2RXCOUNT	Delay in between the end of transmit to the beginning of receive. This is a Johnson count value and thus will count to 8.
7–6 RSVD0	Reserved.
5–4 ENHSTPULLDOWN	Set bit 5 to 1 to override the control of the USB_DP 15-KOhm pulldown. Set bit 4 to 1 to override the control of the USB_DM 15-KOhm pulldown. Clear to 0 to disable.
3–2 HSTPULLDOWN	Set bit 3 to 1 to pull down 15-KOhm on USB_DP line. Set bit 2 to 1 to pull down 15-KOhm on USB_DM line. Clear to 0 to disable.
1 DEBUG_INTERFACE_HOLD	Use holding registers to assist in timing for external UTMI interface.
0 OTGIDPIOLOCK	Once OTG ID from USBPHYx_STATUS_OTGID_STATUS, use this to hold the value. This is to save power for the comparators that are used to determine the ID status.

66.3.7 UTMI Debug Status Register 0 (USBPHYx_DEBUG0_STATUS)

The UTMI Debug Status Register 0 holds multiple views for counters and status of state machines. This is used in conjunction with the USBPHYx_DEBUG1_DBG_ADDRESS field to choose which function to view. The default is described in the bit fields below and is used to count errors.

Address: Base address + 60h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SQUELCH_COUNT						UTMI_RXERROR_FAIL_COUNT										LOOP_BACK_FAIL_COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

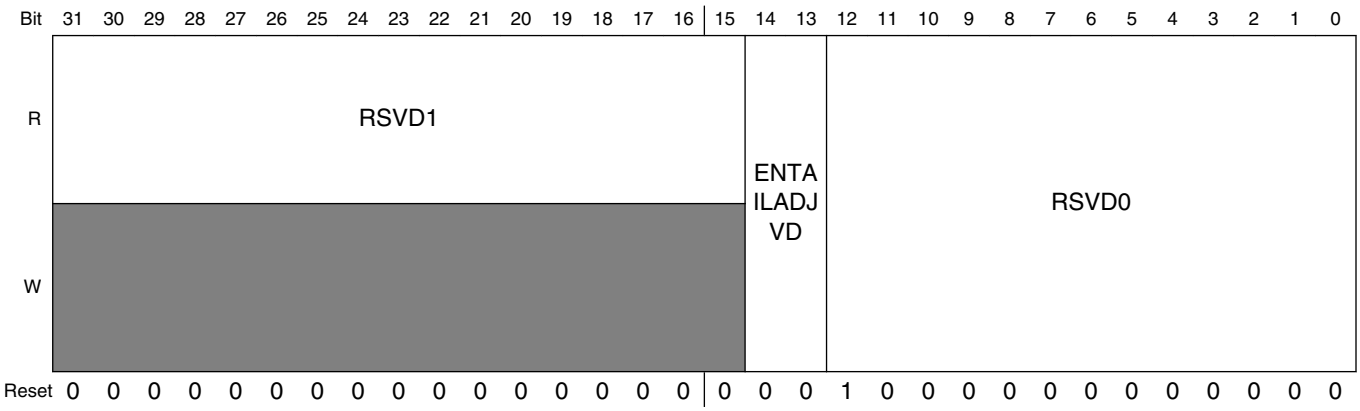
USBPHYx_DEBUG0_STATUS field descriptions

Field	Description
31–26 SQUELCH_ COUNT	Running count of the squelch reset instead of normal end for HS RX.
25–16 UTMI_ RXERROR_ FAIL_COUNT	Running count of the UTMI_RXERROR.
LOOP_BACK_ FAIL_COUNT	Running count of the failed pseudo-random generator loopback. Each time entering testmode, counter goes to 900D and will count up for every detected packet failure in digital/analog loopback tests.

66.3.8 UTMI Debug Status Register 1 (USBPHYx_DEBUG1*n*)

Chooses the muxing of the debug register to be shown in USBPHYx_DEBUG0_STATUS.

Address: Base address + 70h offset + (4d × i), where i=0d to 3d



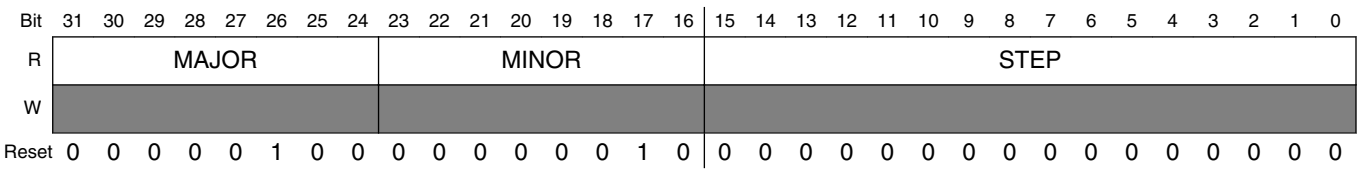
USBPHYx_DEBUG1*n* field descriptions

Field	Description
31–15 RSVD1	Reserved.
14–13 ENTAILADJVD	Delay increment of the rise of squelch: 00 = Delay is nominal 01 = Delay is +20% 10 = Delay is -20% 11 = Delay is -40%
RSVD0	Reserved. **Note:** This bit should remain clear.

66.3.9 UTMI RTL Version (USBPHYx_VERSION)

Fields for RTL Version.

Address: Base address + 80h offset



USBPHYx_VERSION field descriptions

Field	Description
31–24 MAJOR	Fixed read-only value reflecting the MAJOR field of the RTL version.
23–16 MINOR	Fixed read-only value reflecting the MINOR field of the RTL version.
STEP	Fixed read-only value reflecting the stepping of the RTL version.

66.4 USB Analog Memory Map/Register Definition

USB_ANALOG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_81A0	USB VBUS Detect Register (USB_ANALOG_USB1_VBUS_DETECT)	32	R/W	0010_0004h	66.4.1/5513
20C_81A4	USB VBUS Detect Register (USB_ANALOG_USB1_VBUS_DETECT_SET)	32	R/W	0010_0004h	66.4.1/5513
20C_81A8	USB VBUS Detect Register (USB_ANALOG_USB1_VBUS_DETECT_CLR)	32	R/W	0010_0004h	66.4.1/5513
20C_81AC	USB VBUS Detect Register (USB_ANALOG_USB1_VBUS_DETECT_TOG)	32	R/W	0010_0004h	66.4.1/5513
20C_81B0	USB Charger Detect Register (USB_ANALOG_USB1_CHRG_DETECT)	32	R/W	0000_0000h	66.4.2/5514
20C_81B4	USB Charger Detect Register (USB_ANALOG_USB1_CHRG_DETECT_SET)	32	R/W	0000_0000h	66.4.2/5514
20C_81B8	USB Charger Detect Register (USB_ANALOG_USB1_CHRG_DETECT_CLR)	32	R/W	0000_0000h	66.4.2/5514
20C_81BC	USB Charger Detect Register (USB_ANALOG_USB1_CHRG_DETECT_TOG)	32	R/W	0000_0000h	66.4.2/5514

Table continues on the next page...

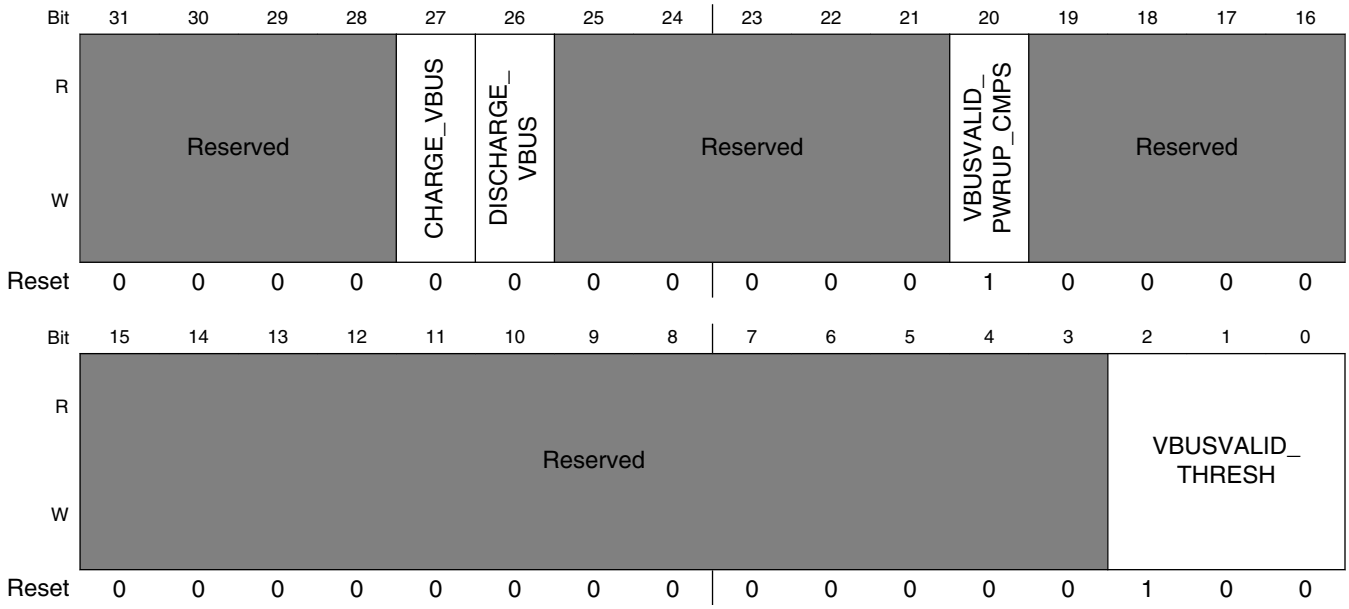
USB_ANALOG memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_81C0	USB VBUS Detect Status Register (USB_ANALOG_USB1_VBUS_DETECT_STAT)	32	R	0000_0000h	66.4.3/5516
20C_81D0	USB Charger Detect Status Register (USB_ANALOG_USB1_CHRG_DETECT_STAT)	32	R	0000_0000h	66.4.4/5518
20C_81F0	USB Misc Register (USB_ANALOG_USB1_MISC)	32	R/W	0000_0002h	66.4.5/5519
20C_81F4	USB Misc Register (USB_ANALOG_USB1_MISC_SET)	32	R/W	0000_0002h	66.4.5/5519
20C_81F8	USB Misc Register (USB_ANALOG_USB1_MISC_CLR)	32	R/W	0000_0002h	66.4.5/5519
20C_81FC	USB Misc Register (USB_ANALOG_USB1_MISC_TOG)	32	R/W	0000_0002h	66.4.5/5519
20C_8200	USB VBUS Detect Register (USB_ANALOG_USB2_VBUS_DETECT)	32	R/W	0010_0004h	66.4.6/5520
20C_8204	USB VBUS Detect Register (USB_ANALOG_USB2_VBUS_DETECT_SET)	32	R/W	0010_0004h	66.4.6/5520
20C_8208	USB VBUS Detect Register (USB_ANALOG_USB2_VBUS_DETECT_CLR)	32	R/W	0010_0004h	66.4.6/5520
20C_820C	USB VBUS Detect Register (USB_ANALOG_USB2_VBUS_DETECT_TOG)	32	R/W	0010_0004h	66.4.6/5520
20C_8210	USB Charger Detect Register (USB_ANALOG_USB2_CHRG_DETECT)	32	R/W	0000_0000h	66.4.7/5522
20C_8214	USB Charger Detect Register (USB_ANALOG_USB2_CHRG_DETECT_SET)	32	R/W	0000_0000h	66.4.7/5522
20C_8218	USB Charger Detect Register (USB_ANALOG_USB2_CHRG_DETECT_CLR)	32	R/W	0000_0000h	66.4.7/5522
20C_821C	USB Charger Detect Register (USB_ANALOG_USB2_CHRG_DETECT_TOG)	32	R/W	0000_0000h	66.4.7/5522
20C_8220	USB VBUS Detect Status Register (USB_ANALOG_USB2_VBUS_DETECT_STAT)	32	R	0000_0000h	66.4.8/5524
20C_8230	USB Charger Detect Status Register (USB_ANALOG_USB2_CHRG_DETECT_STAT)	32	R	0000_0000h	66.4.9/5526
20C_8250	USB Misc Register (USB_ANALOG_USB2_MISC)	32	R/W	0000_0002h	66.4.10/ 5527
20C_8254	USB Misc Register (USB_ANALOG_USB2_MISC_SET)	32	R/W	0000_0002h	66.4.10/ 5527
20C_8258	USB Misc Register (USB_ANALOG_USB2_MISC_CLR)	32	R/W	0000_0002h	66.4.10/ 5527
20C_825C	USB Misc Register (USB_ANALOG_USB2_MISC_TOG)	32	R/W	0000_0002h	66.4.10/ 5527
20C_8260	Chip Silicon Version (USB_ANALOG_DIGPROG)	32	R	0000_0000h	66.4.11/ 5528

66.4.1 USB VBUS Detect Register (USB_ANALOG_USB1_VBUS_DETECT*n*)

This register defines controls for USB VBUS detect.

Address: 20C_8000h base + 1A0h offset + (4d × i), where i=0d to 3d



USB_ANALOG_USB1_VBUS_DETECT*n* field descriptions

Field	Description
31–28 -	This field is reserved. Reserved.
27 CHARGE_VBUS	USB OTG charge VBUS.
26 DISCHARGE_VBUS	USB OTG discharge VBUS.
25–21 -	This field is reserved. Reserved.
20 VBUSVALID_PWRUP_CMPS	Powers up comparators for vbus_valid detector.
19–3 -	This field is reserved. Reserved.
VBUSVALID_THRESH	Set the threshold for the VBUSVALID comparator. This comparator is the most accurate method to determine the presence of 5v, and includes hystersis to minimize the need for software debounce of the detection. This comparator has ~50mV of hystersis to prevent chattering at the comparator trip point. 000 **4V0** — 4.0V

Table continues on the next page...

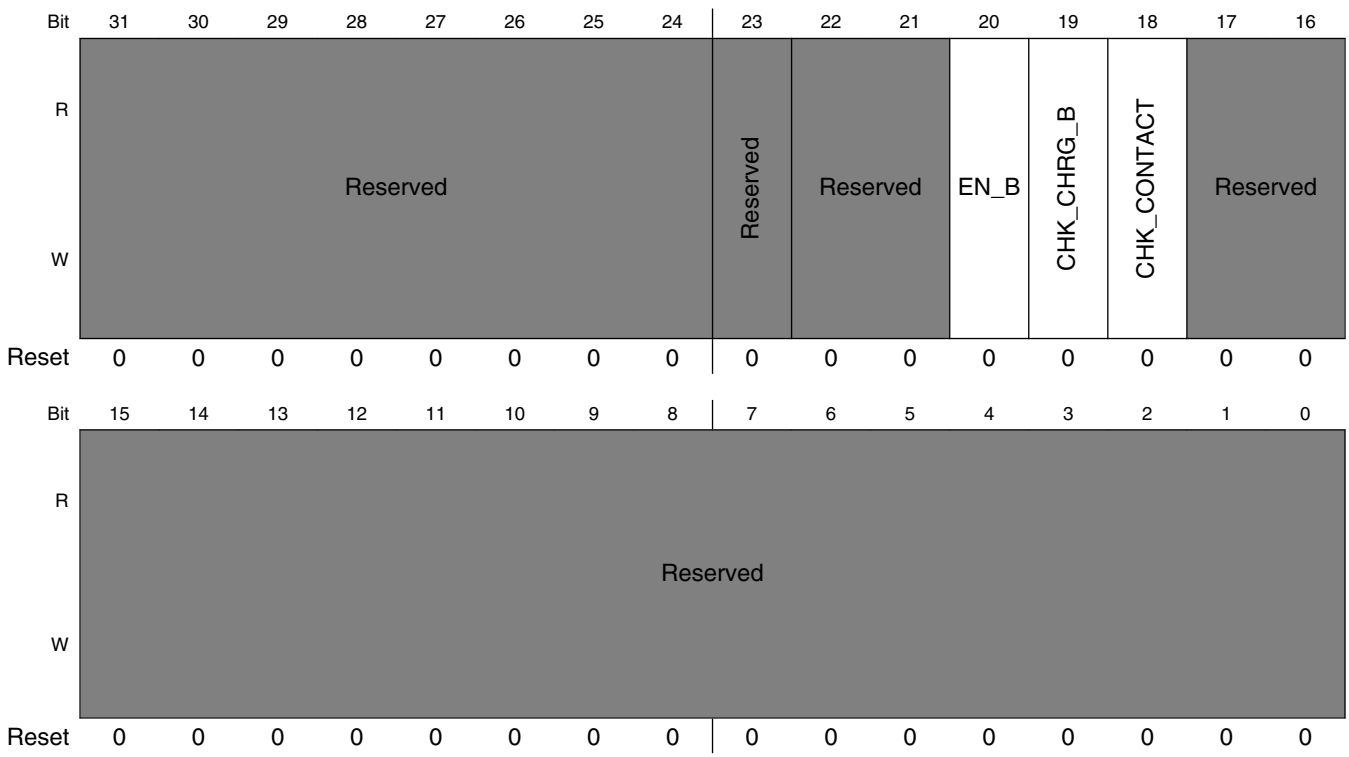
USB_ANALOG_USB1_VBUS_DETECT*n* field descriptions (continued)

Field	Description
	001 **4V1** — 4.1V 010 **4V2** — 4.2V 011 **4V3** — 4.3V 100 **4V4** — 4.4V (default) 101 **4V5** — 4.5V 110 **4V6** — 4.6V 111 **4V7** — 4.7V

66.4.2 USB Charger Detect Register (USB_ANALOG_USB1_CHRG_DETECT*n*)

This register defines controls for USB charger detect.

Address: 20C_8000h base + 1B0h offset + (4d × i), where i=0d to 3d



USB_ANALOG_USB1_CHRG_DETECT*n* field descriptions

Field	Description
31–24 -	This field is reserved. Reserved.

Table continues on the next page...

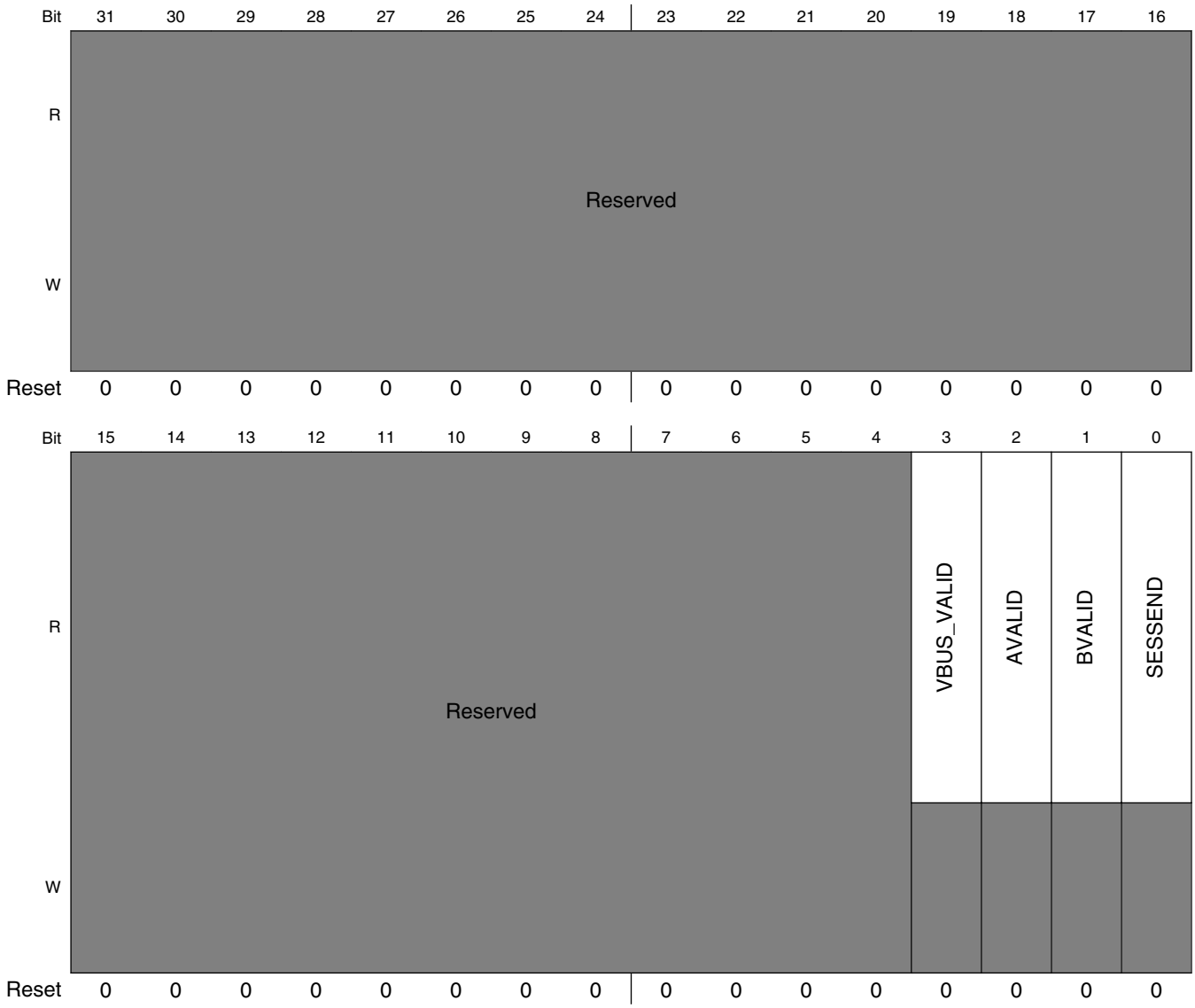
USB_ANALOG_USB1_CHRG_DETECT*n* field descriptions (continued)

Field	Description
23 -	This field is reserved. Reserved.
22–21 -	This field is reserved. Reserved.
20 EN_B	Control the charger detector. 0 **ENABLE** — Enable the charger detector. 1 **DISABLE** — Disable the charger detector.
19 CHK_CHRG_B	Check the charger connection 0 **CHECK** — Check whether a charger (either a dedicated charger or a host charger) is connected to USB port. 1 **NO_CHECK** — Do not check whether a charger is connected to the USB port.
18 CHK_CONTACT	Check the contact of USB plug 0 **NO_CHECK** — Do not check the contact of USB plug. 1 **CHECK** — Check whether the USB plug has been in contact with each other
-	This field is reserved. Reserved.

66.4.3 USB VBUS Detect Status Register (USB_ANALOG_USB1_VBUS_DETECT_STAT)

This register defines fields for USB VBUS Detect status.

Address: 20C_8000h base + 1C0h offset = 20C_81C0h



USB_ANALOG_USB1_VBUS_DETECT_STAT field descriptions

Field	Description
31–4 -	This field is reserved. Reserved.

Table continues on the next page...

USB_ANALOG_USB1_VBUS_DETECT_STAT field descriptions (continued)

Field	Description
3 VBUS_VALID	VBus valid for USB OTG. This bit is a read only version of the state of the analog signal. It can not be overwritten by software.
2 AVALID	Indicates VBus is valid for a A-peripheral. This bit is a read only version of the state of the analog signal. It can not be overritten by software.
1 BVALID	Indicates VBus is valid for a B-peripheral. This bit is a read only version of the state of the analog signal. It can not be overritten by software.
0 SESSEND	Session End for USB OTG. This bit is a read only version of the state of the analog signal. It can not be overwritten by software like the SESSEND bit below. NOTE: This bit's default value depends on whether VDD5V is present, 0 if VDD5V is present, 1 if VDD5V is not present.

66.4.4 USB Charger Detect Status Register (USB_ANALOG_USB1_CHRG_DETECT_STAT)

This register defines fields for USB charger detect status.

Address: 20C_8000h base + 1D0h offset = 20C_81D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												DP_STATE	DM_STATE	CHRG_DETECTED	PLUG_CONTACT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_ANALOG_USB1_CHRG_DETECT_STAT field descriptions

Field	Description
31–4 -	This field is reserved. Reserved.
3 DP_STATE	DP line state output of the charger detector.
2 DM_STATE	DM line state output of the charger detector.
1 CHRG_ DETECTED	State of charger detection. This bit is a read only version of the state of the analog signal. 0 **CHARGER_NOT_PRESENT** — The USB port is not connected to a charger. 1 **CHARGER_PRESENT** — A charger (either a dedicated charger or a host charger) is connected to the USB port.
0 PLUG_ CONTACT	State of the USB plug contact detector. 0 **NO_CONTACT** — The USB plug has not made contact. 1 **GOOD_CONTACT** — The USB plug has made good contact.

66.4.5 USB Misc Register (USB_ANALOG_USB1_MISC*n*)

This register defines controls for USB.

Address: 20C_8000h base + 1F0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	EN_CLK_UTMI	Reserved													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														EN_DEGLITCH	HS_USE_EXTERNAL_R
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

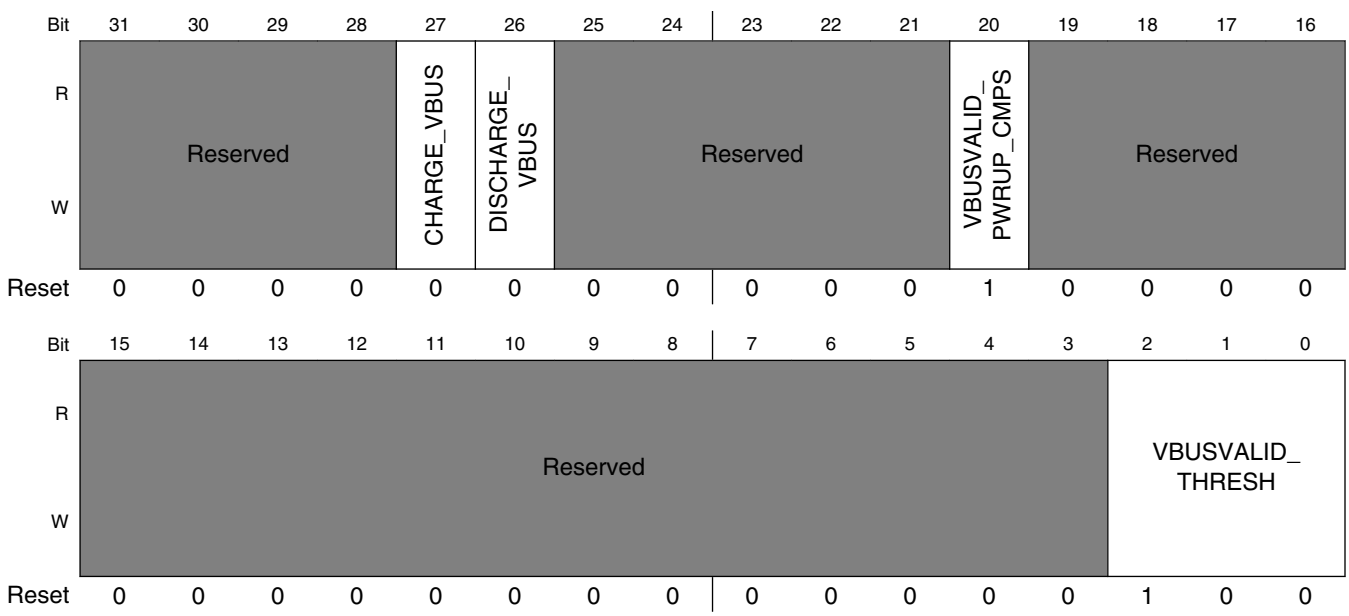
USB_ANALOG_USB1_MISC*n* field descriptions

Field	Description
31 -	This field is reserved. Reserved.
30 EN_CLK_UTMI	Enables the clk to the UTMI block.
29–2 -	This field is reserved. Reserved.
1 EN_DEGLITCH	Enable the deglitching circuit of the USB PLL output.
0 HS_USE_ EXTERNAL_R	Use external resistor to generate the current bias for the high speed transmitter. This bit should not be changed unless recommended by NXP.

66.4.6 USB VBUS Detect Register (USB_ANALOG_USB2_VBUS_DETECT*n*)

This register defines controls for USB VBUS detect.

Address: 20C_8000h base + 200h offset + (4d × i), where i=0d to 3d



USB_ANALOG_USB2_VBUS_DETECT*n* field descriptions

Field	Description
31–28 -	This field is reserved. Reserved.

Table continues on the next page...

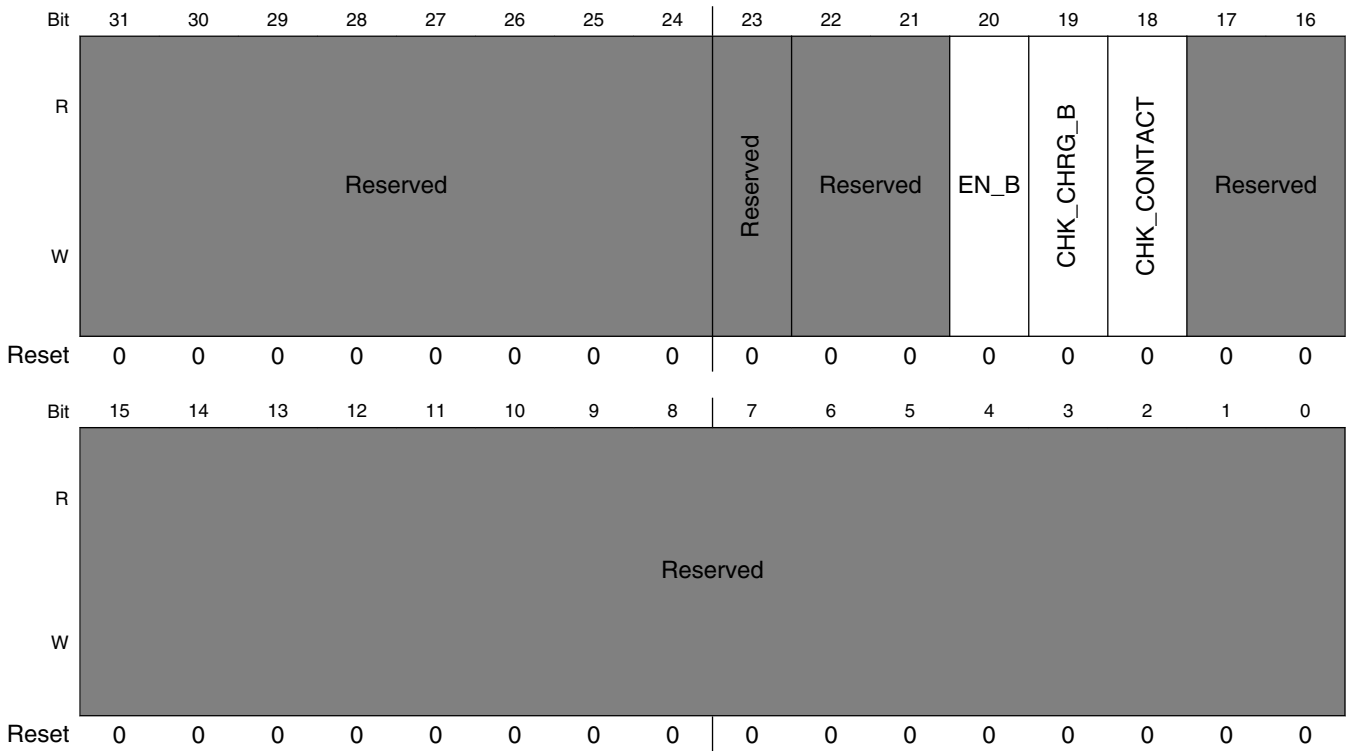
USB_ANALOG_USB2_VBUS_DETECT*n* field descriptions (continued)

Field	Description
27 CHARGE_VBUS	USB OTG charge VBUS.
26 DISCHARGE_ VBUS	USB OTG discharge VBUS.
25–21 -	This field is reserved. Reserved.
20 VBUSVALID_ PWRUP_CMPS	Powers up comparators for vbus_valid detector.
19–3 -	This field is reserved. Reserved.
VBUSVALID_ THRESH	Set the threshold for the VBUSVALID comparator. This comparator is the most accurate method to determine the presence of 5v, and includes hystersis to minimize the need for software debounce of the detection. This comparator has ~50mV of hystersis to prevent chattering at the comparator trip point. 000 **4V0** — 4.0V 001 **4V1** — 4.1V 010 **4V2** — 4.2V 011 **4V3** — 4.3V 100 **4V4** — 4.4V (default) 101 **4V5** — 4.5V 110 **4V6** — 4.6V 111 **4V7** — 4.7V

66.4.7 USB Charger Detect Register (USB_ANALOG_USB2_CHRG_DETECT*n*)

This register defines controls for USB charger detect.

Address: 20C_8000h base + 210h offset + (4d × i), where i=0d to 3d



USB_ANALOG_USB2_CHRG_DETECT*n* field descriptions

Field	Description
31–24 -	This field is reserved. Reserved.
23 -	This field is reserved. Reserved.
22–21 -	This field is reserved. Reserved.
20 EN_B	Control the charger detector. 0 **ENABLE** — Enable the charger detector. 1 **DISABLE** — Disable the charger detector.
19 CHK_CHRG_B	Check the charger connection

Table continues on the next page...

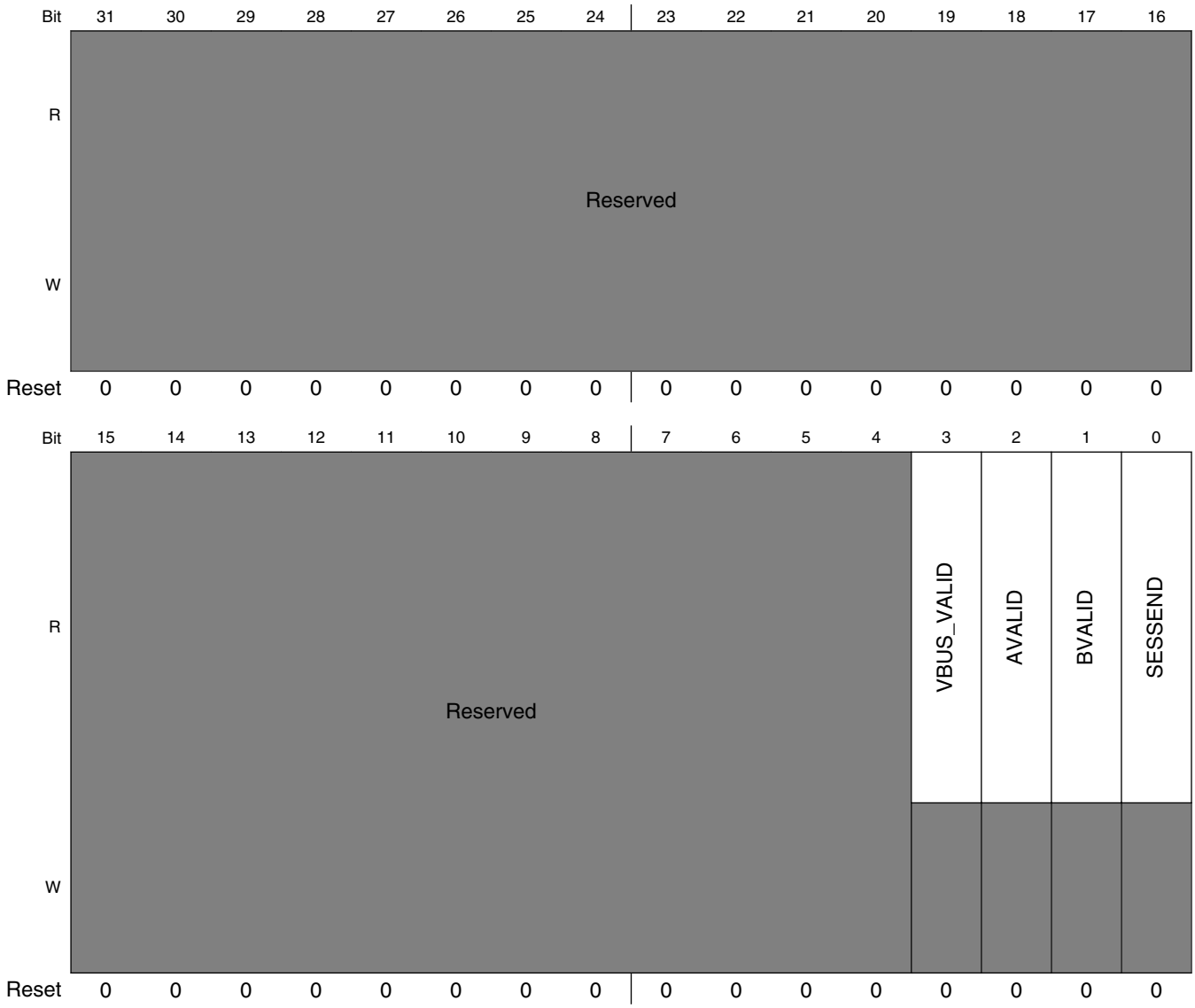
USB_ANALOG_USB2_CHRG_DETECT*n* field descriptions (continued)

Field	Description
	0 **CHECK** — Check whether a charger (either a dedicated charger or a host charger) is connected to USB port. 1 **NO_CHECK** — Do not check whether a charger is connected to the USB port.
18 CHK_CONTACT	Check the contact of USB plug 0 **NO_CHECK** — Do not check the contact of USB plug. 1 **CHECK** — Check whether the USB plug has been in contact with each other
-	This field is reserved. Reserved.

66.4.8 USB VBUS Detect Status Register (USB_ANALOG_USB2_VBUS_DETECT_STAT)

This register defines fields for USB VBUS Detect status.

Address: 20C_8000h base + 220h offset = 20C_8220h



USB_ANALOG_USB2_VBUS_DETECT_STAT field descriptions

Field	Description
31–4 -	This field is reserved. Reserved.

Table continues on the next page...

USB_ANALOG_USB2_VBUS_DETECT_STAT field descriptions (continued)

Field	Description
3 VBUS_VALID	VBus valid for USB OTG. This bit is a read only version of the state of the analog signal. It can not be overwritten by software.
2 AVALID	Indicates VBus is valid for a A-peripheral. This bit is a read only version of the state of the analog signal. It can not be overritten by software.
1 BVALID	Indicates VBus is valid for a B-peripheral. This bit is a read only version of the state of the analog signal. It can not be overritten by software.
0 SESSEND	Session End for USB OTG. This bit is a read only version of the state of the analog signal. It can not be overwritten by software like the SESSEND bit below. NOTE: This bit's default value depends on whether VDD5V is present, 0 if VDD5V is present, 1 if VDD5V is not present.

66.4.9 USB Charger Detect Status Register (USB_ANALOG_USB2_CHRG_DETECT_STAT)

This register defines fields for USB charger detect status.

Address: 20C_8000h base + 230h offset = 20C_8230h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												DP_STATE	DM_STATE	CHRG_DETECTED	PLUG_CONTACT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_ANALOG_USB2_CHRG_DETECT_STAT field descriptions

Field	Description
31–4 -	This field is reserved. Reserved.
3 DP_STATE	DP line state output of the charger detector.
2 DM_STATE	DM line state output of the charger detector.
1 CHRG_ DETECTED	State of charger detection. This bit is a read only version of the state of the analog signal. 0 **CHARGER_NOT_PRESENT** — The USB port is not connected to a charger. 1 **CHARGER_PRESENT** — A charger (either a dedicated charger or a host charger) is connected to the USB port.
0 PLUG_ CONTACT	State of the USB plug contact detector. 0 **NO_CONTACT** — The USB plug has not made contact. 1 **GOOD_CONTACT** — The USB plug has made good contact.

66.4.10 USB Misc Register (USB_ANALOG_USB2_MISC*n*)

This register defines controls for USB.

Address: 20C_8000h base + 250h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	EN_CLK_UTMI	Reserved													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														EN_DEGLITCH	HS_USE_EXTERNAL_R
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

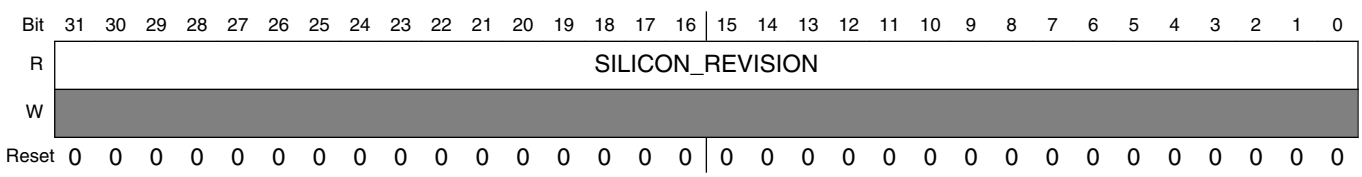
USB_ANALOG_USB2_MISC*n* field descriptions

Field	Description
31 -	This field is reserved. Reserved.
30 EN_CLK_UTMI	Enables the clk to the UTMI block.
29–2 -	This field is reserved. Reserved.
1 EN_DEGLITCH	Enable the deglitching circuit of the USB PLL output.
0 HS_USE_ EXTERNAL_R	Use external resistor to generate the current bias for the high speed transmitter. This bit should not be changed unless recommended by NXP.

66.4.11 Chip Silicon Version (USB_ANALOG_DIGPROG)

The DIGPROG register returns the silicon revision ID.

Address: 20C_8000h base + 260h offset = 20C_8260h



USB_ANALOG_DIGPROG field descriptions

Field	Description
SILICON_ REVISION	Chip silicon revision 0x00610000 Silicon revision 1.0 0x00610001 Silicon revision 1.1 0x00610002 Silicon revision 1.2 0x00610003 Silicon revision 1.3 0x00610004 Silicon revision 1.4

Chapter 67
Ultra Secured Digital Host Controller (uSDHC)

67.1 Overview

The Ultra Secured Digital Host Controller (uSDHC) provides the interface between the host system and the SD/SDIO/MMC cards, as depicted in Figure 67-1.

The uSDHC acts as a bridge, passing host bus transactions to the SD/SDIO/MMC cards by sending commands and performing data accesses to/from the cards.

It handles the SD/SDIO/MMC protocols at the transmission level.

The following are brief descriptions of the cards supported by the uSDHC:

The Multi Media Card (MMC) is a universal low cost data storage and communication media designed to cover a wide array of applications including mobile video and gaming. Previous MMC cards were based on a 7-pin serial bus with a single data pin, while the new high speed MMC communication is based on an advanced 11-pin serial bus designed to operate in the low voltage range.

The Secure Digital Card (SD) is an evolution of the old MMC technology. It is specifically designed to meet the security, capacity, performance, and environment requirements inherent in newly-emerging audio and video consumer electronic devices. The physical form factor, pin assignment and data transfer protocol are forward-compatible with the old MMC (with some additions).

Under the SD protocol, it can be categorized into Memory card, I/O card and Combo card, which has both memory and I/O functions. The memory card invokes a copyright protection mechanism that complies with the security of the SDMI standard. The I/O card, which is also known as SDIO card, provides high-speed data I/O with low power consumption for mobile electronic devices. For the sake of simplicity, the following figure does not show cards with reduced size or mini cards.

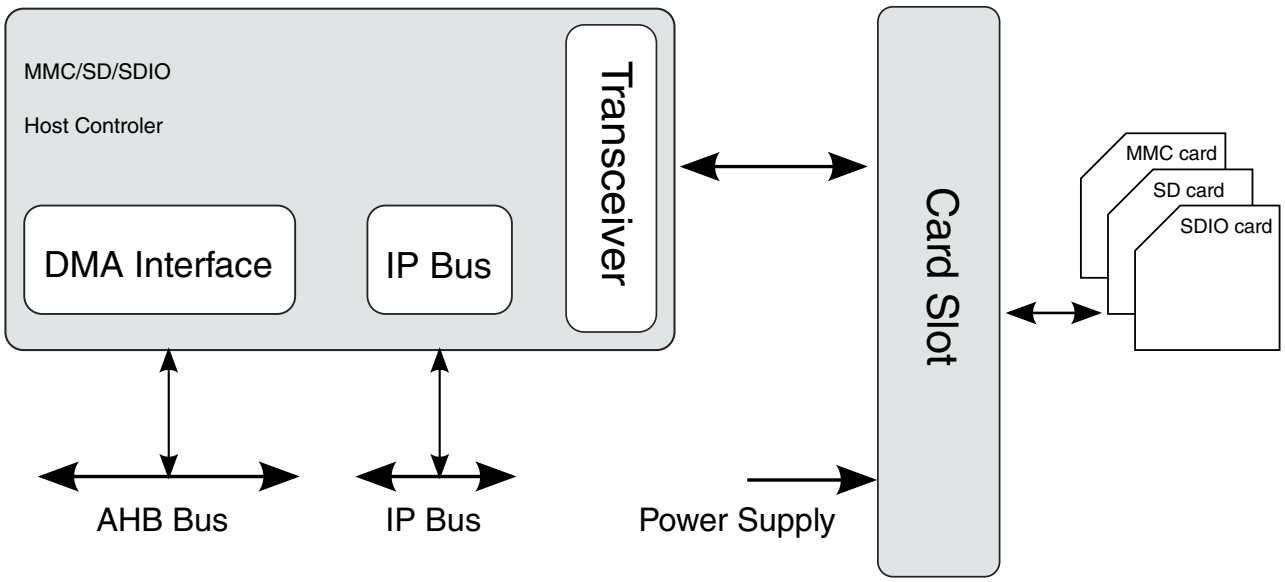


Figure 67-1. System Connection of the uSDHC

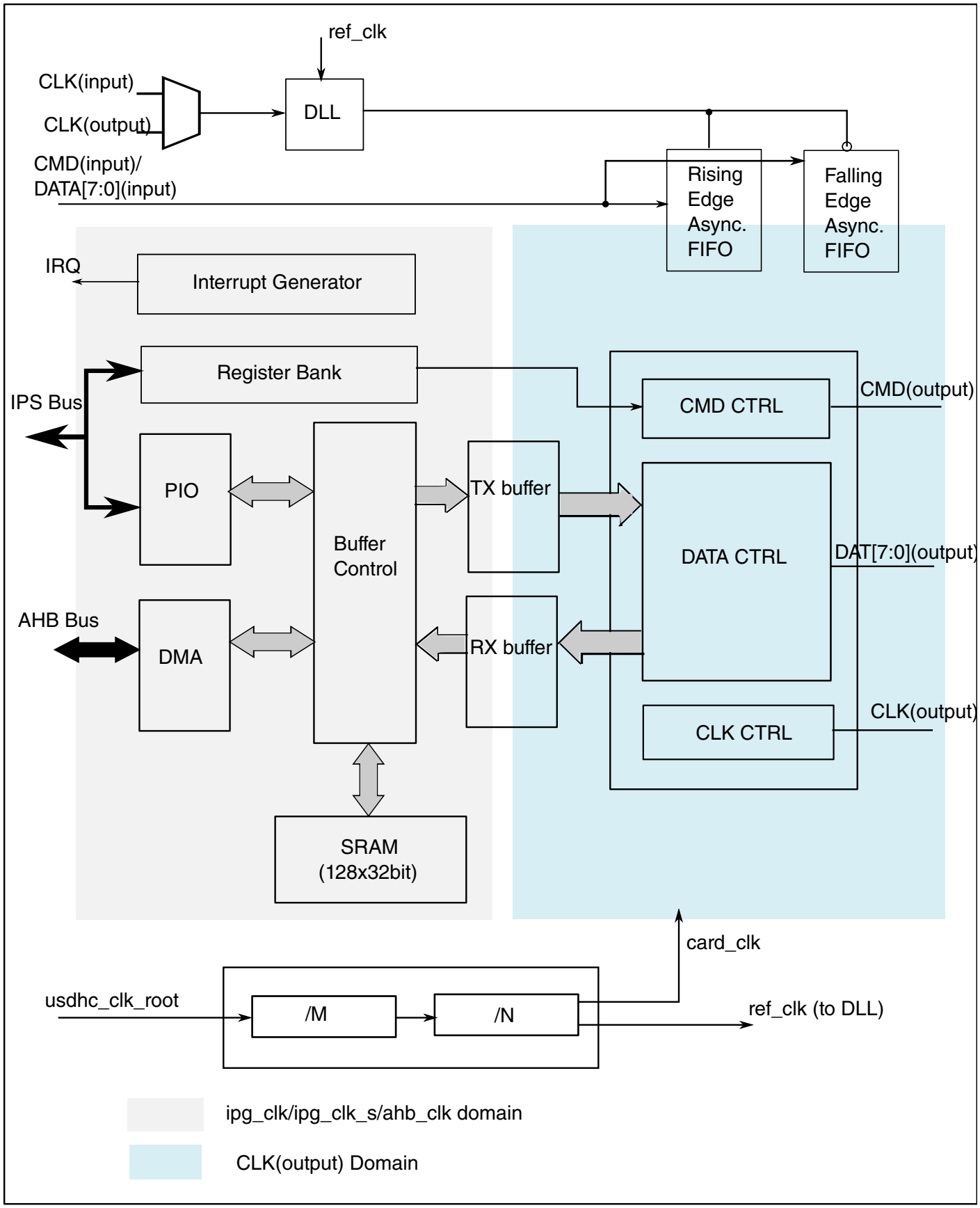


Figure 67-2. ultra Secure Digital Host Controller Block Diagram

67.1.1 Features

The features of the uSDHC module include the following:

- Conforms to the SD Host Controller Standard Specification version 3.0
- Compatible with the MMC System Specification version 4.2/4.3/4.4/4.41/4.5
- Compatible with the SD Memory Card Specification version 3.0 and supports the Extended Capacity SD Memory Card
- Compatible with the SDIO Card Specification version 3.0
- Designed to work with SD Memory, miniSD Memory, SDIO, miniSDIO, SD Combo, MMC, MMC plus, and MMC RS cards
- Card bus clock frequency up to 208 MHz
- Supports 1-bit / 4-bit SD and SDIO modes, 1-bit / 4-bit / 8-bit MMC modes
  - Up to 832 Mbps of data transfer for SDIO cards using 4 parallel data lines in SDR(Single Data Rate) mode
  - Up to 400 Mbps of data transfer for SDIO card using 4 parallel data lines in DDR(Dual Data Rate) mode
  - Up to 832 Mbps of data transfer for SDXC cards using 4 parallel data lines in SDR(Single Data Rate) mode
  - Up to 400 Mbps of data transfer for SDXC card using 4 parallel data lines in DDR(Dual Data Rate) mode
  - Up to 416 Mbps of data transfer for MMC cards using 8 parallel data lines in SDR(Single Data Rate) mode
  - Up to 832 Mbps of data transfer for MMC cards using 8 parallel data lines in DDR(Dual Data Rate) mode
- Supports single block/multi-block read and write
- Supports block sizes of 1 ~ 4096 bytes
- Supports the write protection switch for write operations
- Supports both synchronous and asynchronous abort
- Supports pause during the data transfer at block gap
- Supports SDIO Read Wait and Suspend Resume operations
- Supports Auto CMD12 for multi-block transfer
- Host can initiate non-data transfer command while data transfer is in progress
- Allows cards to interrupt the host in 1-bit and 4-bit SDIO modes, also supports interrupt period
- Embodies a fully configurable 128x32-bit FIFO for read/write data
- Supports internal and external DMA capabilities
- Support voltage selection by configuring vendor specific register bit
- Supports Advanced DMA to perform linked memory access

67.1.2 Modes and Operations

67.1.2.1 Data Transfer Modes

The uSDHC can select the following modes for data transfer:

- SD 1-bit
- SD 4-bit
- MMC 1-bit
- MMC 4-bit
- MMC 8-bit
- Identification Mode (up to 400 kHz)
- MMC full speed mode (up to 26 MHz)
- MMC high speed mode (up to 52 MHz)
- MMC HS200 mode (up to 200 MHz)
- MMC DDR mode (52 MHz both edges)
- SD/SDIO full speed mode (up to 25 MHz)
- SD/SDIO high speed mode (up to 50 MHz)
- SD/SDIO UHS-I mode (up to 208 MHz in SDR mode, up to 50 MHz in DDR mode)

67.2 External Signals

The following table describes the external signals of USDHC:

Table 67-1. USDHC1 External Signals

Signal	Description	Pad	Mode	Direction
SD1_CD_B (CD_B)	Card detection pin If not used(for the embedded memory),tie low to indicate there is a card attached.	GPIO_1	ALT6	I
SD1_CLK (CLK)	Clock for MMC/SD/SDIO card	SD1_CLK	ALT0	O
SD1_CMD (CMD)	CMD line connect to card	SD1_CMD	ALT0	IO
SD1_DATA0 (DATA0)	DATA0 line in all modes Also used to detect busy state	SD1_DAT0	ALT0	IO
SD1_DATA1 (DATA1)	DATA1 line in 4/8-bit mode Also used to detect interrupt in 1/4-bit mode	SD1_DAT1	ALT0	IO
SD1_DATA2 (DATA2)	DATA2 line or Read Wait in 4-bit mode	SD1_DAT2	ALT0	IO

Table continues on the next page...

Table 67-1. USDHC1 External Signals (continued)

Signal	Description	Pad	Mode	Direction
	Read Wait in 1-bit mode			
SD1_DATA3 (DATA3)	DATA3 line in 4/8-bit mode or configured as card detection pin May be configured as card detection pin in 1-bit mode	SD1_DAT3	ALT0	IO
SD1_DATA4 (DATA4)	DATA4 line in 8-bit mode, not used in other modes	NANDF_D0	ALT1	IO
SD1_DATA5 (DATA5)	DATA5 line in 8-bit mode, not used in other modes	NANDF_D1	ALT1	IO
SD1_DATA6 (DATA6)	DATA6 line in 8-bit mode, not used in other modes	NANDF_D2	ALT1	IO
SD1_DATA7 (DATA7)	DATA7 line in 8-bit mode, not used in other modes	NANDF_D3	ALT1	IO
SD1_LCTL (LCTL)	LED control used to drive an external LED Active high Fully controlled by the driver Optional output	GPIO_16	ALT3	O
SD1_VSELECT (VSELECT)	IO power voltage selection signal	KEY_COL1	ALT6	O
		KEY_ROW3	ALT6	
SD1_WP (WP)	Card write protect detect If not used(for the embedded memory), tie low to indicate it's not write protected.	DI0_PIN4	ALT3	I
		GPIO_9	ALT6	

Table 67-2. USDHC2 External Signals

Signal	Description	Pad	Mode	Direction
SD2_CD_B (CD_B)	Card detection pin If not used(for the embedded memory),tie low to indicate there is a card attached.	GPIO_4	ALT6	I
SD2_CLK (CLK)	Clock for MMC/SD/SDIO card	SD2_CLK	ALT0	O
SD2_CMD (CMD)	CMD line connect to card	SD2_CMD	ALT0	IO
SD2_DATA0 (DATA0)	DATA0 line in all modes Also used to detect busy state	SD2_DAT0	ALT0	IO
SD2_DATA1 (DATA1)	DATA1 line in 4/8-bit mode Also used to detect interrupt in 1/4-bit mode	SD2_DAT1	ALT0	IO
SD2_DATA2 (DATA2)	DATA2 line or Read Wait in 4-bit mode Read Wait in 1-bit mode	SD2_DAT2	ALT0	IO
SD2_DATA3 (DATA3)	DATA3 line in 4/8-bit mode or configured as card detection pin	SD2_DAT3	ALT0	IO

Table continues on the next page...

Table 67-2. USDHC2 External Signals (continued)

Signal	Description	Pad	Mode	Direction
	May be configured as card detection pin in 1-bit mode			
SD2_DATA4 (DATA4)	DATA4 line in 8-bit mode, not used in other modes	NANDF_D4	ALT1	IO
SD2_DATA5 (DATA5)	DATA5 line in 8-bit mode, not used in other modes	NANDF_D5	ALT1	IO
SD2_DATA6 (DATA6)	DATA6 line in 8-bit mode, not used in other modes	NANDF_D6	ALT1	IO
SD2_DATA7 (DATA7)	DATA7 line in 8-bit mode, not used in other modes	NANDF_D7	ALT1	IO
SD2_LCTL (LCTL)	LED control used to drive an external LED Active high Fully controlled by the driver Optional output	GPIO_6	ALT6	O
SD2_VSELECT (VSELECT)	IO power voltage selection signal	KEY_ROW1	ALT6	O
		KEY_ROW2	ALT4	
SD2_WP (WP)	Card write protect detect If not used(for the embedded memory), tie low to indicate it's not write protected.	GPIO_2	ALT6	I

Table 67-3. USDHC3 External Signals

Signal	Description	Pad	Mode	Direction
SD3_CLK (CLK)	Clock for MMC/SD/SDIO card	SD3_CLK	ALT0	O
SD3_CMD (CMD)	CMD line connect to card	SD3_CMD	ALT0	IO
SD3_DATA0 (DATA0)	DATA0 line in all modes Also used to detect busy state	SD3_DAT0	ALT0	IO
SD3_DATA1 (DATA1)	DATA1 line in 4/8-bit mode Also used to detect interrupt in 1/4-bit mode	SD3_DAT1	ALT0	IO
SD3_DATA2 (DATA2)	DATA2 line or Read Wait in 4-bit mode Read Wait in 1-bit mode	SD3_DAT2	ALT0	IO
SD3_DATA3 (DATA3)	DATA3 line in 4/8-bit mode or configured as card detection pin May be configured as card detection pin in 1-bit mode	SD3_DAT3	ALT0	IO
SD3_DATA4 (DATA4)	DATA4 line in 8-bit mode, not used in other modes	SD3_DAT4	ALT0	IO
SD3_DATA5 (DATA5)	DATA5 line in 8-bit mode, not used in other modes	SD3_DAT5	ALT0	IO
SD3_DATA6 (DATA6)	DATA6 line in 8-bit mode, not used in other modes	SD3_DAT6	ALT0	IO

Table continues on the next page...

Table 67-3. USDHC3 External Signals (continued)

Signal	Description	Pad	Mode	Direction
SD3_DATA7 (DATA7)	DATA7 line in 8-bit mode, not used in other modes	SD3_DAT7	ALT0	IO
SD3_RESET (RESET)	Card hardware reset signal, active LOW	SD3_RST	ALT0	O
SD3_VSELECT (VSELECT)	IO power voltage selection signal	GPIO_18	ALT2	O
		NANDF_CS1	ALT2	

Table 67-4. USDHC4 External Signals

Signal	Description	Pad	Mode	Direction
SD4_CLK (CLK)	Clock for MMC/SD/SDIO card	SD4_CLK	ALT0	O
SD4_CMD (CMD)	CMD line connect to card	SD4_CMD	ALT0	IO
SD4_DATA0 (DATA0)	DATA0 line in all modes Also used to detect busy state	SD4_DAT0	ALT1	IO
SD4_DATA1 (DATA1)	DATA1 line in 4/8-bit mode Also used to detect interrupt in 1/4-bit mode	SD4_DAT1	ALT1	IO
SD4_DATA2 (DATA2)	DATA2 line or Read Wait in 4-bit mode Read Wait in 1-bit mode	SD4_DAT2	ALT1	IO
SD4_DATA3 (DATA3)	DATA3 line in 4/8-bit mode or configured as card detection pin May be configured as card detection pin in 1-bit mode	SD4_DAT3	ALT1	IO
SD4_DATA4 (DATA4)	DATA4 line in 8-bit mode, not used in other modes	SD4_DAT4	ALT1	IO
SD4_DATA5 (DATA5)	DATA5 line in 8-bit mode, not used in other modes	SD4_DAT5	ALT1	IO
SD4_DATA6 (DATA6)	DATA6 line in 8-bit mode, not used in other modes	SD4_DAT6	ALT1	IO
SD4_DATA7 (DATA7)	DATA7 line in 8-bit mode, not used in other modes	SD4_DAT7	ALT1	IO
SD4_RESET (RESET)	Card hardware reset signal, active LOW	NANDF_ALE	ALT1	O
SD4_VSELECT (VSELECT)	IO power voltage selection signal	NANDF_CS1	ALT1	O

67.2.1 Signals Overview

The uSDHC has 14 associated I/O signals.

- The CLK is an internally generated clock used to drive the MMC, SD, SDIO cards.

- The CMD I/O is used to send commands and receive responses to and from the card. Eight data lines (DAT7~DAT0) are used to perform data transfers between the uSDHC and the card.
- The CD and WP are card detection and write protection signals directly routed from the socket. These two signals are active low (0). A low on CD# means that a card is inserted, and a high on WP means that the write protect switch is active.
- LCTL is an output signal used to drive an external LED to indicate that the SD interface is busy.
- RST is an output signal used to reset the MMC card.
- VSELECT is an output signal used to change the voltage of the external power supplier.

CD, WP, LCTL, RST and VSELECT are all optional for system implementation. If the uSDHC needs to support a 4-bit data transfer, DAT7~DAT4 can also be optional and tied to high.

67.3 Clocks

The table found here describes the clock sources for uSDHC.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 67-5. uSDHC Clocks

Clock name	Clock Root	Description
hclk	ahb_clk_root	AHB bus clock
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_perclk	usdhc_clk_root	Base clock
ipg_clk_s	ipg_clk_root	Peripheral access clock for register accesses

67.4 Functional Description

The following sections provide a brief functional description of the major system blocks, including the Data Buffer, DMA AHB interface, register bank as well as IP Bus interface, dual-port memory wrapper, data/command controller, clock and reset manager, and clock generator.

67.4.1 Data Buffer

The uSDHC uses one configurable data buffer to transfer data between the system bus (IP Bus or AHB Bus) and the SD card in an optimized manner, maximizing throughput between the two clock domains (IP peripheral clock and the master clock).

The buffer is used as temporary storage for data being transferred between the host system and the card. The watermark levels for read and write are both configurable and can be from 1 to 128 words. The burst lengths for read and write are also configurable and can be from 1 to 31 words.

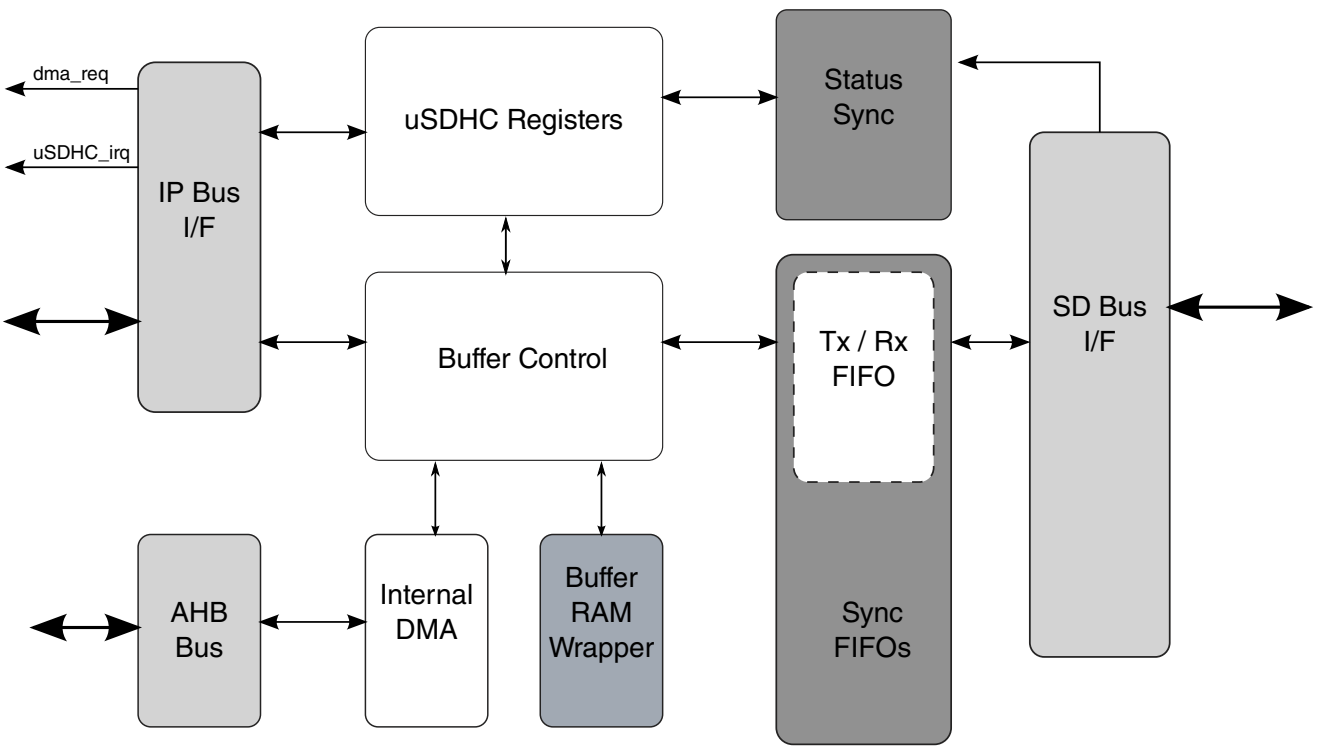


Figure 67-3. uSDHC Buffer Scheme

There are 3 transfer modes to access the data buffer:

- CPU polling mode:
  - For a host read operation, when the number of words received in the buffer meets or exceeds the RD_WML watermark value, by polling the BRR bit, the Host Driver can read the Buffer Data Port register to fetch the amount of words set in the RD_WML register from the buffer. The write operation is similar.
- External DMA mode:
  - For a read operation, when there are more words received in the buffer than the amount set in the RD_WML register, a DMA request is sent out to inform the external DMA to fetch the data. The request will be immediately de-asserted when there is an access on the Buffer Data Port register. If the number of words

in the buffer after the current burst meets or exceeds RD_WML value, the DMA request is asserted again. For instance, if there are twice as many words in the buffer as there are in the RD_WML value, there are two successive DMA requests with only one cycle of de-assertion between. The write operation is similar. Note the accesses CPU polling mode and external DMA mode both use the IP bus, and if the external DMA is enabled, in both modes an external DMA request is sent when the buffer is ready.

- Internal DMA mode (includes simple and advanced DMA accesses):
  - The internal DMA access, either by simple or advanced DMA, is over the AHB bus. For internal DMA access mode, the external DMA request will never be sent out.

For a read operation, when there are more words in the buffer than the amount set in the RD_WML register, the internal DMA starts fetching data over the AHB bus. Except for INCR4 and INCR8, the burst type is always INCR mode and the burst length depends on the shortest of following factors:

- Burst length configured in the burst length field of the Watermark Level register
- Watermark Level boundary
- Block size boundary
- Data boundary configured in the current descriptor (if the ADMA is active)
- 1 Kbyte address boundary defined in the AHB protocol

Write operation is similar.

Sequential and contiguous access is necessary to ensure the pointer address value is correct. Random or skipped access is not possible. The byte order, by reset, is little endian mode. The actual byte order is swapped inside the buffer, according to the endian mode configured by software (see the following figures). For a host write operation, byte order is swapped after data is fetched from the buffer and ready to send to the SD Bus. For a host read operation, byte order is swapped before the data is stored in the buffer.

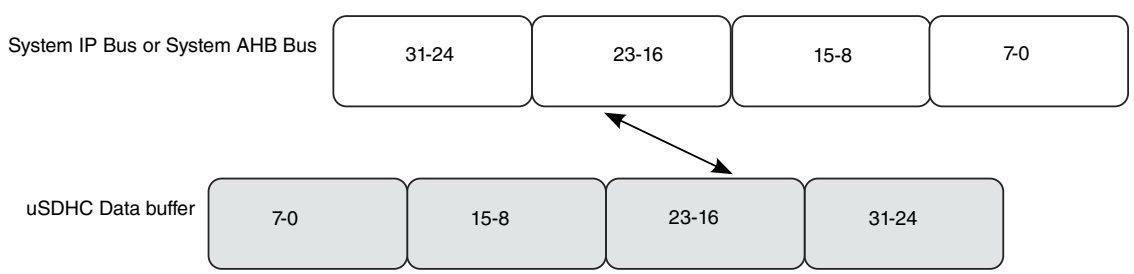


Figure 67-4. Data Swap between System Bus and uSDHC Data Buffer in Byte Little Endian Mode

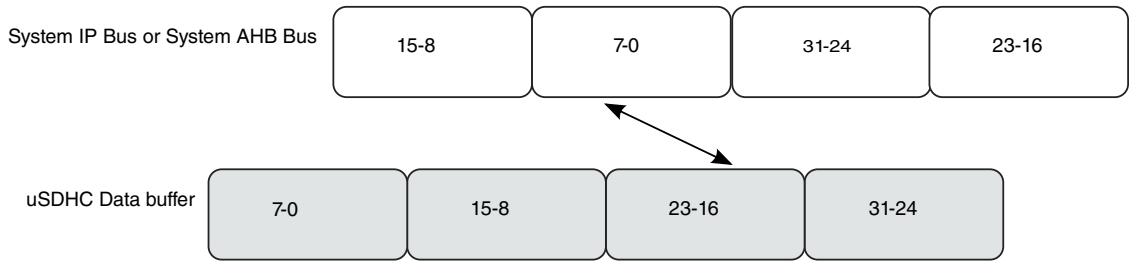


Figure 67-5. Data Swap between System Bus and uSDHC Data Buffer in Half Word Big Endian Mode

67.4.1.1 Write Operation Sequence

There are three ways to write data into the buffer when the user transfers data to the card:

- External DMA through the uSDHC DMA request signal
- Processor core polling through the BWR bit in Interrupt Status register (interrupt or polling)
- Internal DMA

When the internal DMA is not used, (the DMAEN bit in the Transfer Type register is not set when the command is sent), the uSDHC asserts a DMA request when the amount of buffer space exceeds the value set in the WR_WML register, and is ready for receiving new data. At the same time, the uSDHC sets the BWR bit. The buffer write ready interrupt will be generated if it is enabled by software.

When internal DMA is used, the uSDHC will not inform the system before all the required number of bytes are transferred (if no error was encountered). When an error occurs during the data transfer, the uSDHC will abort the data transfer and abandon the current block. The Host Driver should read the contents of the DMA System Address register to obtain the starting address of the abandoned data block. If the current data transfer is in multi-block mode, the uSDHC will not automatically send CMD12, even though the AC12EN bit in the Transfer Type register is set. The Host Driver sends CMD12 in this scenario and re-starts the write operation from that address. It is recommended that a Software Reset for Data be applied before the transfer is re-started.

The uSDHC will not start data transmission until the number of words set in the WR_WML register can be held in the buffer. If the buffer is empty and the Host System does not write data in time, the uSDHC will stop the CLK to avoid the data buffer under-run situation.

67.4.1.2 Read Operation Sequence

There are three ways to read data from the buffer when the user transfers data to the card:

- External DMA through the uSDHC DMA request signal
- Processor core polling through the BRR bit in Interrupt Status register (interrupt or polling)
- Internal DMA

When internal DMA is not used (DMAEN bit in Transfer Type register is not set when the command is sent), the uSDHC asserts a DMA request when the amount of data exceeds the value set in the RD_WML register, that is available and ready for system fetching data. At the same time, the uSDHC sets the BRR bit. The buffer read ready interrupt will be generated if it is enabled by software.

When internal DMA is used, the uSDHC will not inform the system before all the required number of bytes are transferred (if no error was encountered). When an error occurs during the data transfer, the uSDHC will abort the data transfer and abandon the current block. The Host Driver should read the content of the DMA System Address register to get the starting address of the abandoned data block. If the current data transfer is in multi-block mode, the uSDHC will not automatically send CMD12, even though the AC12EN bit in the Transfer Type register is set. The Host Driver sends CMD12 in this scenario and re-starts the read operation from that address. It is recommended that a Software Reset for Data be applied before the transfer is re-started.

For any write transfer mode, the uSDHC will not start data transmission until the number of words set in the RD_WML register are in the buffer. If the buffer is full and the Host System does not read data in time, the uSDHC will stop the CLK to avoid the data buffer over-run situation.

67.4.1.3 Data Buffer and Block Size

The user needs to know the buffer size for the buffer operation during a data transfer to utilize it in the most optimized way. In the uSDHC, the only data buffer can hold up to 128 words (32-bit) and the watermark levels for write and read can be configured accordingly.

For both read and write, the watermark level can be from 1 to 128 words. For both read and write the burst length can be from 1 to 31 words. The Host Driver may configure the value according to the system situation and requirement.

During a multi-block data transfer, the block length can be set to any value between 1 and 4096 bytes, satisfying the requirements of the external card. The only restriction is from the external card, which can be limited in size or support of a partial block access (which is not the integer times of 512 bytes).

As uSDHC treats each block individually, for block sizes which are not multiples of four (not word-aligned) stuffed bytes are required at the end of each block. For example, if the block size is 7 bytes and there are 12 blocks to write, the system side must write two times for each block. For each block the ending byte will be abandoned by uSDHC because it only sends 7 bytes to the card and picks data from the following system write, resulting in 24 beats of write access in total.

67.4.1.4 Dividing Large Data Transfer

This SDIO command CMD53 definition limits the maximum data size of data transfers according to the following formula:

Max data size = Block size x Block count

The length of a multiple block transfer needs to be in block size units. If the total data length can't be divided evenly into a multiple of the block size, then there are two ways to transfer the data which depend on the function and the card design. Option 1 is for the Host Driver to split the transaction. The remainder of the block size data is then transferred by using a single block command at the end. Option 2 is to add dummy data in the last block to fill the block size. For option 2, the card must manage the removal of the dummy data.

See the figure below for an example showing the dividing of large data transfers, assuming a kind of WLAN SDIO card that only supports a block size up to 64 bytes. Although the uSDHC supports a block size of up to 4096 bytes, the SDIO can only accept a block size less than 64 bytes, so the data must be divided (see example below).

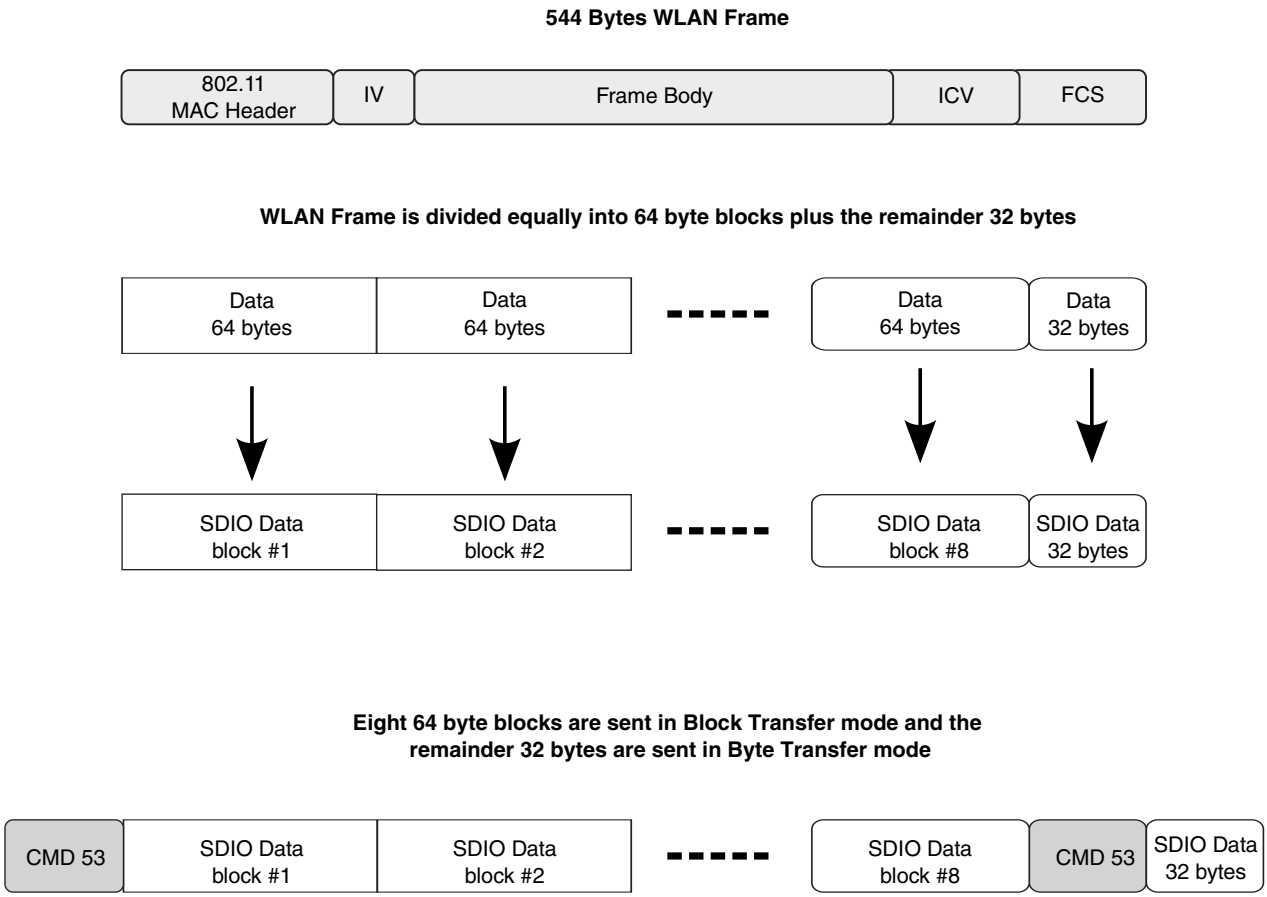


Figure 67-6. Example for Dividing Large Data Transfers

67.4.1.5 External DMA Request

When the internal DMA is not in use and external DMA is enabled, the Data Buffer will generate a DMA request to the system. During a write operation, when the number of WR_WML words can be held in the buffer free space, the signal uSDHC_dreq_b is asserted to 0, informing the Host System of a DMA write.

The BWR bit in the Interrupt Status register is also set, as long as the BWRSEN bit in the Interrupt Status Enable register is set. The DMA request is de-asserted after several accesses to the Data Port register are made while the buffer's free space can't meet the watermark condition (free space> write watermark level).

On read operation, when the number of RD_WML words are already in the buffer, the signal uSDHC_dreq_b is asserted to 0, informing the Host System for a DMA read. The BRR bit in the Interrupt Status register is also set, as long as the BRRSEN bit in the

Interrupt Status Enable register is set. The DMA request is de-asserted after several accesses to the Data Port register are made while the buffer's data can't meet the watermark condition (the number of data in buffer > read watermark level).

If the DMA burst length can't change during a data transfer for an external DMA transfer, the watermark level (read or write) must be a divisor of the block size. If it is not, transferring the block may cause buffer under-run (read operation) or over-run (write operation). For example, if the block size is 512 bytes, the watermark level of read (or write) must be a power of two between 1 and 128. For processor core polling access there is no such issue, as the last access in the block transfer can be controlled by software. The watermark level can be any value, even larger than the block size (but no greater than 128 words) because the actual number of bytes transferred by the software can be controlled and does not exceed the block size in each transfer.

The uSDHC also supports non-word aligned block size, as long as the card supports that block size. In this case, the watermark level should be set as the number of words. For example, if the block size is 31 bytes, the watermark level can be set to any number of words. For this case, the BLKSIZE bits of the Block Attribute register will be set as 1fh. For the CPU polling access, the burst length can be 1 to 128 words, without restriction. This is because the software will transfer 8 words, and the uSDHC will also set the BWR or BRR bits when the remaining data does not violate data buffer. See DMA Burst Length for more details about the dynamic watermark level of the data buffer.

For the above example, even though 8 words are transferred via the Data Port register, the uSDHC will transfer only 31 bytes over the SD Bus, as required by the BLKSIZE bits. In this data transfer, with non-word aligned block size, the endian mode should be set cautiously or invalid data will be transferred to and from the card.

67.4.2 DMA AHB Interface

The internal DMA implements a DMA engine and the AHB master. When the internal DMA is enabled, the uSDHC_dreq_b will not be asserted during the transfer, but the BWR and BRR bits will be set if the BWRSEN and BRRSEN bits have been set in the Interrupt Status Enable register.

See the figure below for an illustration of the DMA AHB interface block.

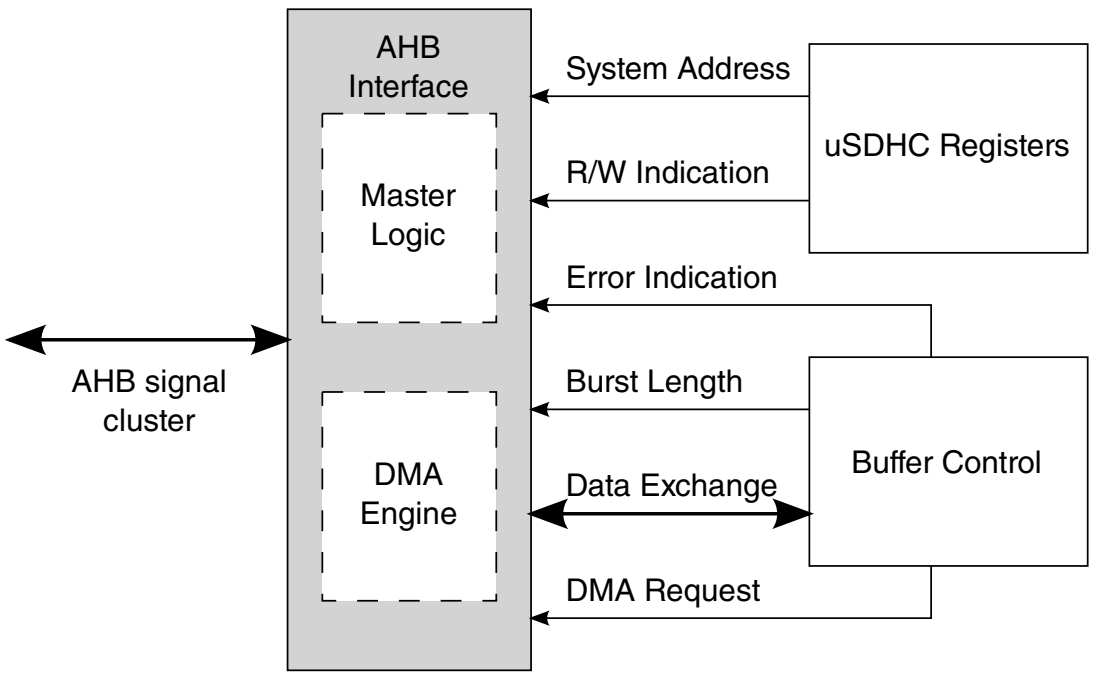


Figure 67-7. DMA AHB Interface Block

67.4.2.1 Internal DMA Request

If the watermark level requirement is met in data transfer or if the last data of current block is ready in the data buffer, and the Internal DMA is enabled, the Data Buffer block will send a DMA request to AHB interface. Meanwhile, the external DMA request signal (uSDHC_dreq_b) is disabled.

The delay in response from the internal DMA engine depends on the system AHB bus loading and the priority assigned to the uSDHC. The DMA engine does not respond to the request during its burst transfer, but is ready to serve as soon as the burst is over. The Data Buffer de-asserts the request if the data buffer space(for write) or bytes in data buffer is smaller than the watermark level. Upon access to the buffer by internal DMA,

the Data Buffer updates its internal buffer pointer, and when the watermark level is satisfied or the last data of current block is ready in the data buffer, another DMA request is sent.

The data transfer is in the block unit, and the subsequent watermark level is always set as the remaining number of words. For instance, for a multi block data read with each block size of 31 bytes, and the burst length set to 6 words. After the first burst transfer, if there are more than 2 words in the buffer (which might contain some data of the next block), another DMA request is sent. This is because the remaining number of words to send for the current block is (31 - 6 * 4) / 4 = 2. The uSDHC will read 2 words out of the buffer, with 7 valid bytes and 1 stuffed byte.

67.4.2.2 DMA Burst Length

Just like a CPU polling access, the DMA burst length for the internal DMA engine can be from 1 to 16 words. The actual burst length for the DMA depends on the lesser of the configured burst length or the remaining words of the current block.

See the example in Internal DMA Request. After 6 words are read, the burst length will be 2 words, then the next burst length will be 6 words. This is because the next block starts, which is 31 bytes, more than 6 words. The Host Driver may take this variable burst length into account. It is also acceptable to configure the burst length as the divisor of the block size, so that each time the burst length will be the same.

67.4.2.3 AHB Master Interface

It is possible that the internal AHB DMA engine could fail during the data transfer. Upon detection of an AHB bus error during DMA transfer, the DMA engine stops the transfer and goes to the idle state.At that point, the internal data buffer stops receiving incoming data and sending out data. The DMAE bit in the Interrupt Status register will be generated to host CPU to report a bus error condition.

Once the DMAE interrupt is received, the software shall send a CMD12 to abort the current transfer and read the DS_ADDR bits of the DMA System Address register to get the starting address of the corrupted block. After the DMA error is fixed, the software should apply a data reset and re-start the transfer from this address to recover the corrupted block. DMA operation will resume when the interrupt is serviced by software.

67.4.2.4 ADMA Engine

In the SD Host Controller Standard, a new DMA transfer algorithm called the ADMA (Advanced DMA) is defined. For Simple DMA, once the page boundary is reached, a DMA interrupt will be generated and the new system address shall be programmed by the Host Driver.

The ADMA defines the programmable descriptor table in the system memory. The Host Driver can calculate the system address at the page boundary and program the descriptor table before executing ADMA. It reduces the frequency of interrupts to the host system. Therefore, higher speed DMA transfers could be realized since the Host MCU intervention would not be needed during long DMA based data transfers.

There are two types of ADMA: ADMA1 and ADMA2 in Host Controller. ADMA1 can support data transfer of 4KB aligned data in system memory. ADMA2 improves the restriction so that data of any location and any size can be transferred in system memory. Their formats of Descriptor Table are different.

ADMA can recognize all kinds of descriptors define in SD Host Controller Standard, and if 'End' flag is detected in the descriptor, ADMA will stop after this descriptor is processed.

67.4.2.4.1 ADMA Concept and Descriptor Format

For ADMA1, including the following descriptors:

- Valid/Invalid descriptor.
- Nop descriptor.
- Set data length descriptor.
- Set data address descriptor.
- Link descriptor.
- Interrupt flag and End flag in descriptor.

For ADMA2, including the following descriptors:

- Valid/Invalid descriptor.
- Nop descriptor.
- Rsv descriptor.
- Set data length & address descriptor.
- Link descriptor.
- Interrupt flag and End flag in descriptor.

See Figure 67-8 for the format of the descriptor table for ADMA1.

Figure 67-11 explains the ADMA2 format. ADMA2 deals with the lower 32-bit first, and then the higher 32-bit. If the 'Valid' flag of descriptor is 0, it will ignore the high 32-bit. Address field shall be set on word aligned(lower 2-bit is always set to 0). Data length is in byte unit.

ADMA will start read/write operation after it reaches the Tran state, using the data length and data address analyzed from most recent descriptor(s).

For ADMA1, the valid data length descriptor is the last Set type descriptor before Tran type descriptor. Every Tran type will trigger a transfer, and the transfer data length is extracted from the most recent Set type descriptor. If there is no Set type descriptor after the previous Trans descriptor, the data length will be the value for previous transfer, or 0 if no Set descriptor is ever met.

For ADMA2, Tran type descriptor contains both data length and transfer data address, so only a Tran type descriptor can start a data transfer

Address/ Page Field		Address/ Page Field		Attribute Field					
31	12	11	6	5	4	3	2	1	0
Address or Data Length		000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	31- 28	27- 12
0	0	Nop	No Operation	Don't Care	
0	1	Set	Set Data Length	0000	Data Length
1	0	Tran	Transfer Data	Data Address	
1	1	Link	Link Descriptor	Descriptor Address	

Valid	Valid = 1 indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is processed.

Figure 67-8. Format of the ADMA1 Descriptor Table

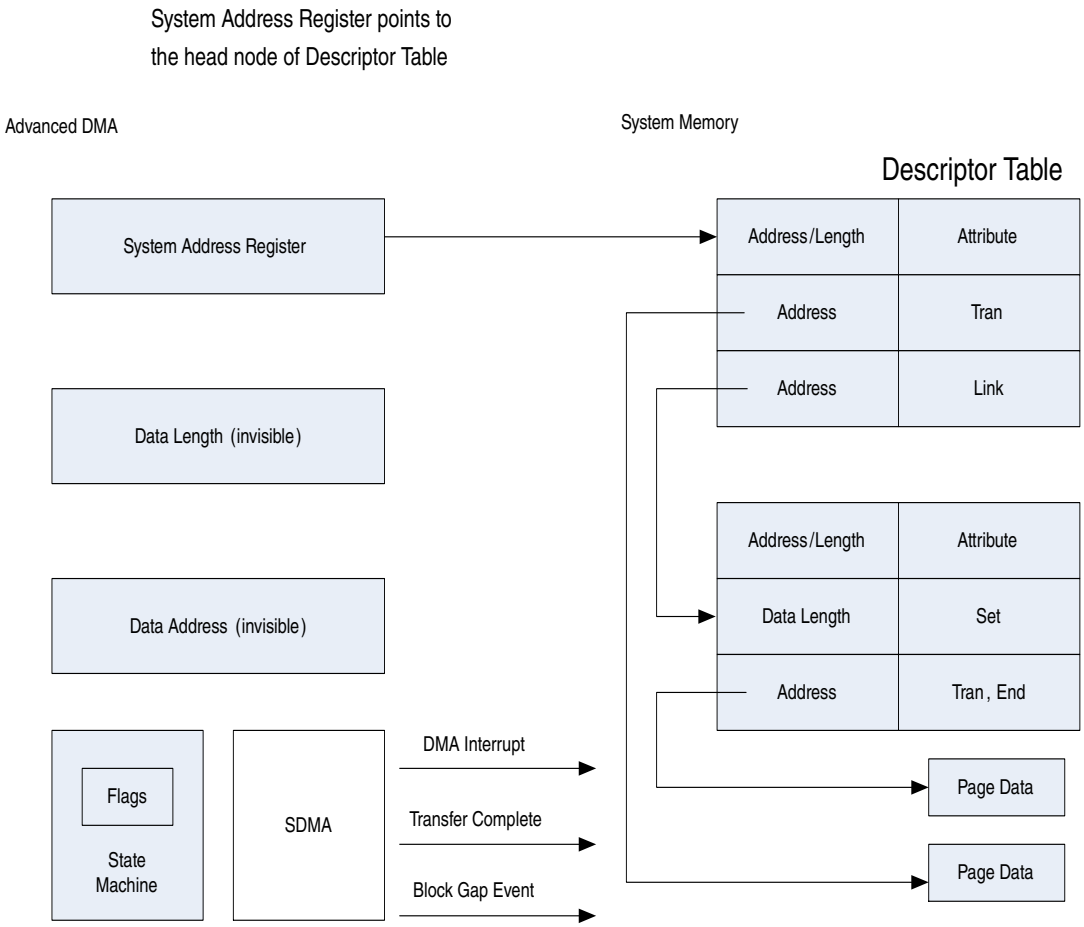


Figure 67-9. Concept and Access Method of ADMA1 Descriptor Table

Address Field		Length		Reserved		Attribute Field					
63	32	31	16	15	06	05	04	03	02	01	00
32-bit Address		16-bit length		0000000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	Operation
0	0	Nop	No Operation	Don't Care
0	1	Rsv	Reserved	Same as Nop. Read this line and go to next one
1	0	Tran	Transfer Data	Transfer data with address and length set in this descriptor line
1	1	Link	Link Descriptor	Link to another descriptor

Valid	Valid = 1 indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is processed.

Figure 67-10. Format of the ADMA2 Descriptor Table

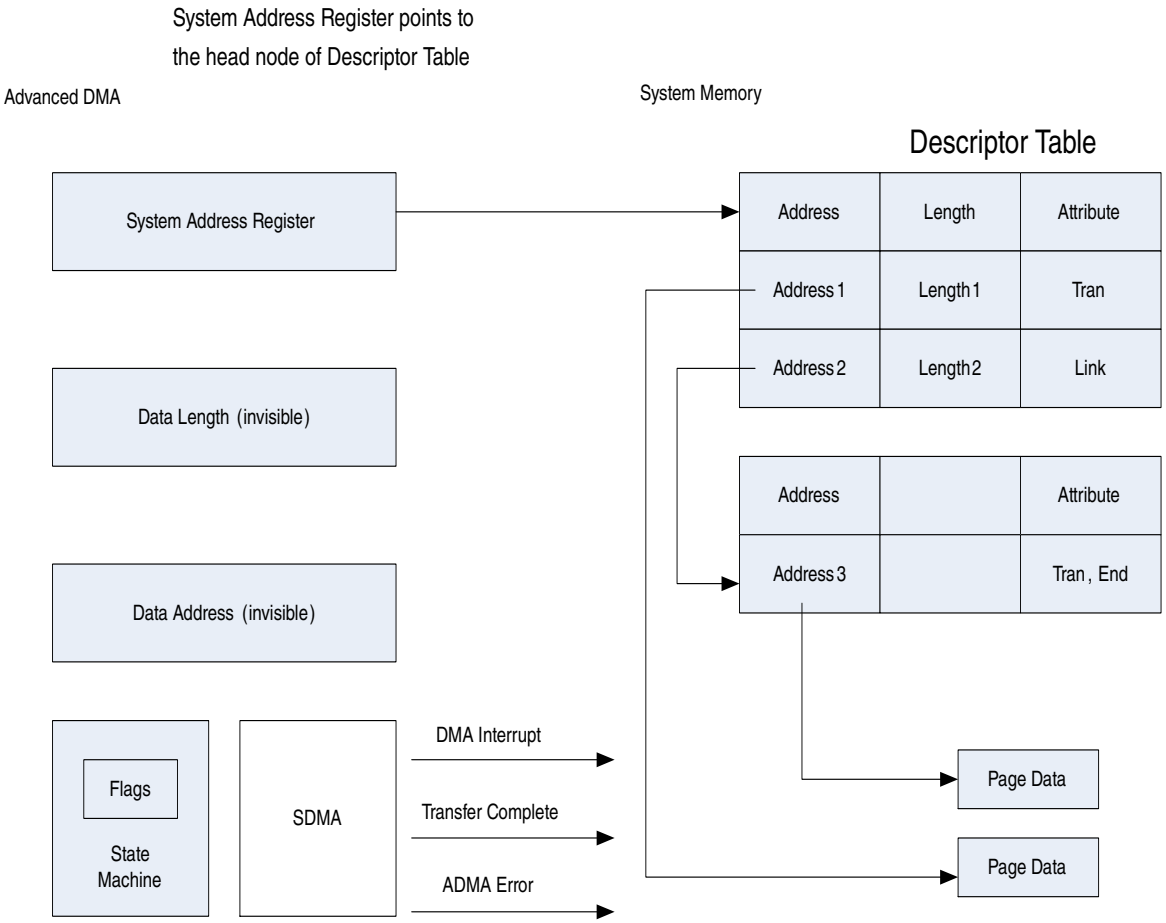


Figure 67-11. Concept and Access Method of ADMA2 Descriptor Table

67.4.2.4.2 ADMA Interrupt

If the interrupt flag descriptor is set, ADMA will generate an interrupt according to various types of descriptors:

For ADMA1:

- Set type of descriptor: interrupt is generated when data length is set.
- Tran type descriptor: interrupt is generated when this transfer is complete.
- Link type of descriptor: interrupt is generated when new descriptor address is set.
- Nop type of descriptor: interrupt is generated just after this descriptor is fetched.

For ADMA2:

- Tran type of descriptor: interrupt is generated when this transfer is complete.
- Link type of descriptor: interrupt is generated when new descriptor address is set.
- Nop/Rsv type of descriptor: interrupt is generated just after this descriptor is fetched.

67.4.2.4.3 ADMA Error

The ADMA will stop whenever any error is encountered. These errors include:

- Fetching descriptor error
- AHB response error
- Data length mismatch error

An ADMA descriptor error will be generated when it fails to detect a 'Valid' flag in the descriptor. If an ADMA descriptor error occurs, the interrupt is not generated even if the 'Interrupt' flag of this descriptor is set.

When BLKCNTEN bit is set, data length set in buffer must be equal to the whole data length set in descriptor nodes, otherwise data length mismatch error will be generated.

When BLKCNTEN bit is not set, then whole data length set in descriptor should be a multiple of block lengths; otherwise, when data set in the descriptor nodes are not performed at block boundaries, then data mismatch errors will occur.

67.4.3 Register Bank with IP Bus Interface

Register accesses via the IP Bus interface are actually on the Register Bank.

See Figure 67-12 below for the block diagram.

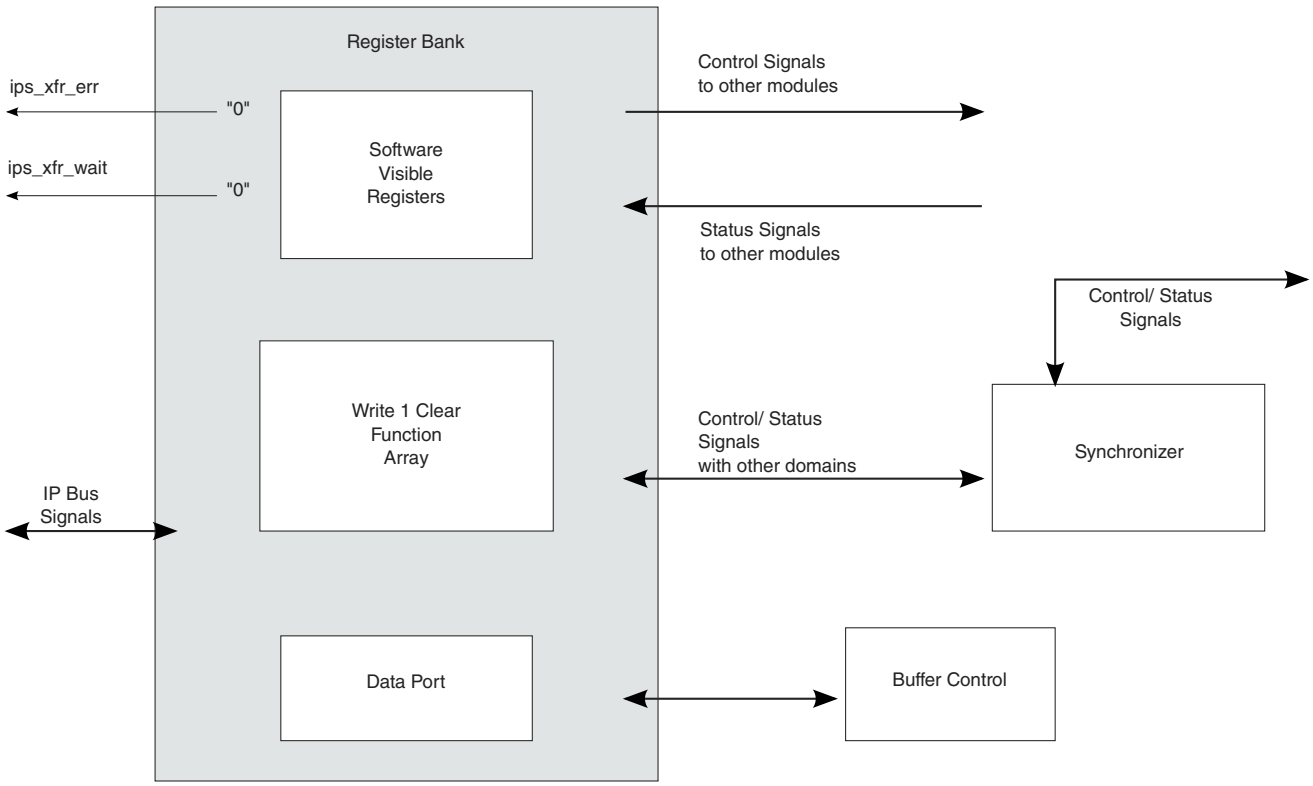


Figure 67-12. Register Bank Diagram

Only 32-bit access is allowed, and no partial read / write is supported, thus all accesses are word aligned.

67.4.3.1 SD Protocol Unit

The SD protocol unit deals with all SD protocol affairs.

The SD Protocol Unit performs the following functions:

- Acts as the bridge between the internal buffer and the SD bus
- Sends the command data as well as its argument serially
- Stores the serial response bit stream into corresponding registers
- Detects the bus state on the CMD/DAT lines
- Monitors the interrupt from the SDIO card
- Asserts the read wait signal
- Gates off the SD clock when buffer is announcing danger status
- Detects the write protect state

The SD Protocol Unit consists of four sub modules:

1. SD control misc.

2. Command control.
3. Data control.
4. Clock control

67.4.3.2 SD control misc

In the SD control misc unit, the card detect(include the CD_B and DATA3 used as Card Detection), write protection and card interrupt are implemented.

This module monitors the signal level on all 8 data lines, the command lines, and directly routes the level values into the Register Bank. The driver can use this for debug purposes.

The module also detects the WP (Write Protect) line. If WP is active, writes to the register bank will be ignored.

This module also drives the LCTL output signal when the LCTL bit is set by the driver.

67.4.3.3 SD Clock control

If the internal data buffer is near full(for read) or near empty(for write), the SD clock must be gated off to avoid buffer over/under-run, this module will assert the gate of the output SD clock to shut the clock off. After the buffer has space(for read) or has data(for write), the clock gate of this module will open and the SD clock will be active again.

67.4.3.4 Command control

The Command Control module deals with the transactions on the CMD line.

See the figure below for an illustration of the structure for the Command CRC Shift Register.

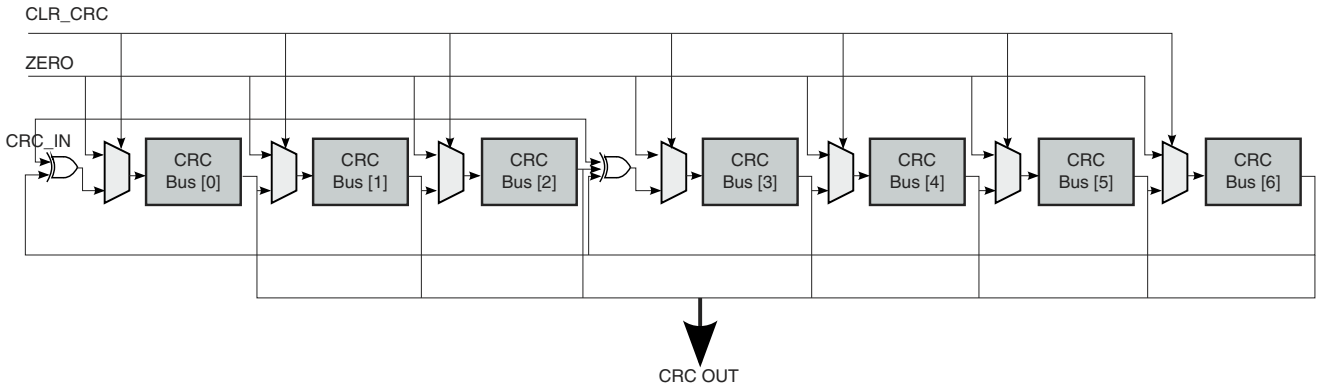


Figure 67-13. Command CRC Shift Register

The CRC polynomials for the CMD are as follows:

```
Generator polynomial: G(x) = x^7 + x^3 + 1
M(x) = (first bit) * x^n + (second bit) * x^n-1 +...+ (last bit) * x^0
CRC[6:0] = Remainder [(M(x) * x^7) / G(x)]
```

67.4.3.5 Data control

The Data Agent deals with the transactions on the eight data lines. Moreover, this module also detects the busy state on the DATA0 line, and generates the Read Wait state by the request from the Transceiver.

The CRC polynomials for the DATA are as follows:

```
Generator polynomial: G(x) = x^16 + x^12 + x^5 +1
M(x) = (first bit) * x^n + (second bit) * x^n-1 +...+ (last bit) * x^0
CRC[15:0] = Remainder [(M(x) * x^16) / G(x)]
```

67.4.4 Clock & Reset Manager

This module controls all the reset signals within the uSDHC.

There are four kinds of reset signals within uSDHC:

1. Hardware reset.
2. Software reset for all logic.
3. Software reset for the data logic.
4. Software reset for the command logic.

All these signals are fed into this module and stable signals are generated inside the module to reset all other modules. The module also gates off all the inside signals.

67.4.5 Clock Generator

The Clock Generator generates the card CLK by peripheral source clock in two stages.

Refer to the figure below for the structure of the divider. The term "Base" represents the frequency of peripheral source clock.

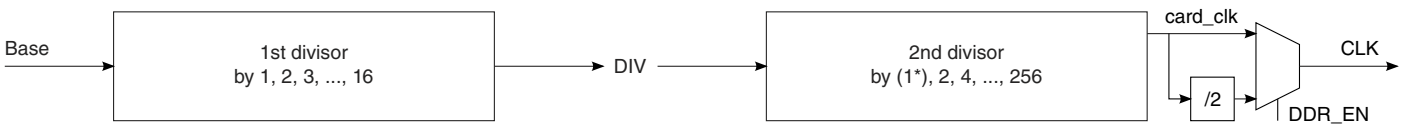


Figure 67-14. Two Stages of the Clock Divider

The first divisor stage (controlled by uSDHCx_SYS_CTRL[DVS]) outputs an intermediate clock (DIV), which can be Base, Base/2, Base/3, ..., or Base/16.

The second divisor stage (controlled by uSDHCx_SYS_CTRL[SDCLKFS]) outputs the actual internal working clock (card_clk). This clock is the driving clock for all sub modules of the SD Protocol Unit, and the sync FIFOs (see Figure 67-3) to synchronize with the data rate from the internal data buffer.

Please note, in SDR mode and DDR mode, the CLK are different.

- In SDR mode, CLK is equal to the internal working clock(card_clk).

- In DDR mode, CLK is equal to the card_clk/2.

67.4.6 SDIO Card Interrupt

Information on Interrupts in 1-bit Mode, Interrupts in 4-bit Mode, and Card Interrupt Handling are detailed in the sections below.

67.4.6.1 Interrupts in 1-bit Mode

In this case the DATA1 pin is dedicated to providing the interrupt function. An interrupt is asserted by pulling the DATA1 low from the SDIO card, until the interrupt service is finished to clear the interrupt.

67.4.6.2 Interrupt in 4-bit Mode

Since the interrupt and data line 1 share Pin 8 in 4-bit mode, an interrupt will only be sent by the card and recognized by the host during a specific time. This is known as the Interrupt Period. The uSDHC will only sample the level on Pin 8 during the Interrupt Period. At all other times, the host will ignore the level on Pin 8, and treat it as the data signal. The definition of the Interrupt Period is different for operations with single block and multiple block data transfers.

In the case of normal single data block transmissions, the Interrupt Period becomes active two clock cycles after the completion of a data packet. This Interrupt Period lasts until after the card receives the end bit of the next command that has a data block transfer associated with it.

For multiple block data transfers in 4-bit mode, there is only a limited period of time that the Interrupt Period can be active due to the limited period of data line availability between the multiple blocks of data. This requires a more strict definition of the Interrupt Period. For this case, the Interrupt Period is limited to two clock cycles. This begins two clocks after the end bit of the previous data block. During this 2-clock cycle interrupt period, if an interrupt is pending, the DATA1 line will be held low for one clock cycle with the last clock cycle pulling DATA1 high. On completion of the Interrupt Period, the card releases the DATA1 line into the high Z state. The uSDHC samples the DATA1 during the Interrupt Period when the IABG bit in the Protocol Control register is set.

Refer to SDIO Card Specification v1.10f for further information about the SDIO card interrupt.

67.4.6.3 Card Interrupt Handling

When the CINTIEN bit in the Interrupt Signal Enable Register is set to 0, the uSDHC clears the interrupt request to the Host System. The Host Driver should clear this bit before servicing the SDIO Interrupt and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.

The SDIO Card Interrupt Status can be cleared by writing 1 to this bit. But as the interrupt source from the SDIO card does not clear, this bit is set again. In order to clear this bit, it is required to reset the interrupt source from the external card followed by a writing 1 to this bit. In 1-bit mode, the uSDHC will detect the SDIO Interrupt with or without the SD clock (to support wakeup). In 4-bit mode, the interrupt signal is sampled during the Interrupt Period, so there are some sample delays between the interrupt signal from the SDIO card and the interrupt to the Host System Interrupt Controller. When the SDIO status has been set, and the Host Driver needs to service this interrupt, so the SDIO bit in the Interrupt Control Register of SDIO card will be cleared. This is required to clear

the SDIO interrupt status latched in the uSDHC and to stop driving the interrupt signal to the System Interrupt Controller. The Host Driver must issue a CMD52 to clear the card interrupt. After completion of the card interrupt service, the SDIO Interrupt Status Enable bit is set to 1, and the uSDHC starts sampling the interrupt signal again.

See the figure below for an illustration of the SDIO card interrupt scheme and for the sequences of software and hardware events that take place during a card interrupt handling procedure.

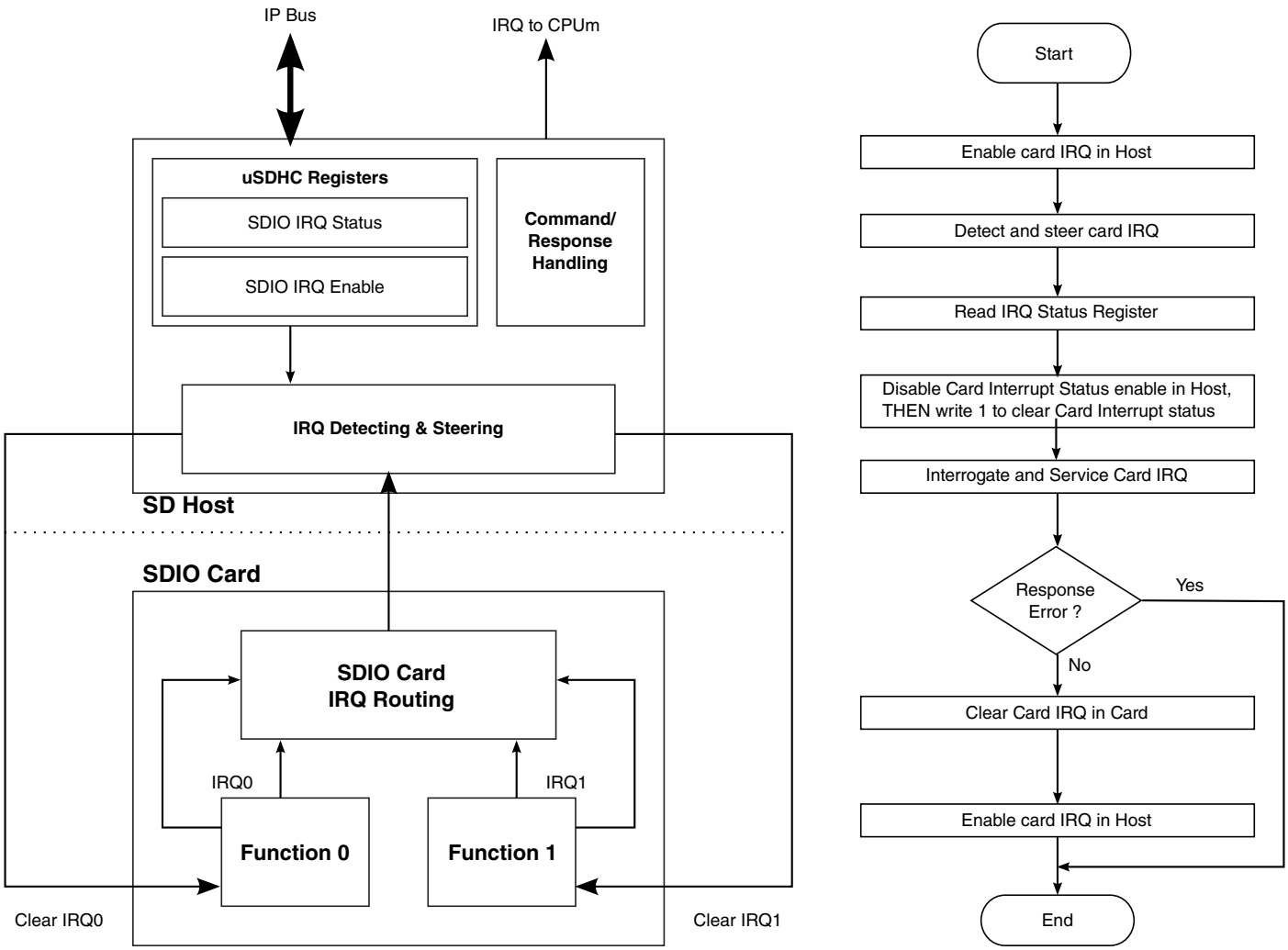


Figure 67-15. Card Interrupt Scheme and Card Interrupt Detection and Handling Procedure

67.4.7 Card Insertion and Removal Detection

The uSDHC uses either the DATA3 pin or the CD_B pin to detect card insertion or removal. When there is no card on the MMC/SD bus, the DATA3 will be pulled to a low voltage level by default.

When any card is inserted to or removed from the socket, the uSDHC detects the logic value changes on the DATA3 pin and generates an interrupt. When the DATA3 pin is not used for card detection (for example, it is implemented in GPIO), the CD_B pin must be connected for card detection. Whether DATA3 is configured for card detection or not, the CD_B pin is always a reference for card detection. Whether the DATA3 pin or the CD_B pin is used to detect card insertion, the uSDHC will send an interrupt (if enabled) to inform the Host system that a card is inserted.

67.4.8 Power Management and Wake Up Events

When there is no operation between the uSDHC and the card through the SD bus, the user can completely disable the ipg_clk and ipg_perclk in the chip level clock control module to save power. When the user needs to use the uSDHC to communicate with the card, it can enable the clock and start the operation.

In some circumstances, when the clocks to the uSDHC are disabled, for instance, when the system is in low power mode, there are some events for which the user needs to enable the clock and handle the event. These events are called wakeup interrupts. The uSDHC can generate these interrupt even when there are no clocks enabled. The three interrupts which can be used as wake up events are:

1. Card Removal Interrupt
2. Card Insertion Interrupt
3. Interrupt from SDIO card

The uSDHC offers a power management feature. By clearing the clock enabled bits in the System Control Register, the clocks are gated in the low position to the uSDHC. For maximum power saving, the user can disable all the clocks to the uSDHC when there is no operation in progress.

These three wake up events (or wakeup interrupts) can also be used to wake up the system from low-power modes.

NOTE

To make the interrupt a wakeup event, when all the clocks to the uSDHC are disabled or when the whole system is in low power mode, the corresponding wakeup enabled bit needs to be

set. Refer to Protocol Control (uSDHC_PROT_CTRL) for more information on the uSDHC Protocol Control register.

67.4.8.1 Setting Wake Up Events

For the uSDHC to respond to a wakeup event, the software must set the respective wakeup enable bit before the CPU enters sleep mode.

Before the software disables the host clock, it should ensure that all of the following conditions have been met:

- No Read or Write Transfer is active
- Data and Command lines are not active
- No interrupts are pending
- Internal data buffer is empty

67.4.9 MMC fast boot

The Embedded MultiMediaCard (eMMC4.3) specification adds a fast boot feature which requires hardware support. There are two types of fast boot mode, boot operation, and alternative boot operation in the eMMC4.3 specification. Each type also has with-acknowledge and without-acknowledge modes.

In boot operation mode, the master (MultiMediaCard host) can read boot data from the slave (MMC device) by keeping CMD line low after power-on, or sending CMD0 with argument + 0xFFFFFFFA (optional for slave), before issuing CMD1.

NOTE

For the eMMC4.3 card setting, please see the eMMC4.3 specification.

67.4.9.1 Boot operation

NOTE

For the purposes of this documentation, fast boot is called "normal fast boot mode".

If the CMD line is held LOW for 74 clock cycles and more after power-up before the first command is issued, the slave recognizes that boot mode is being initiated and starts preparing boot data internally.

Within 1 second after the CMD line goes LOW, the slave starts to send the first boot data to the master on the DATA line(s). The master must keep the CMD line LOW to read all of the boot data.

If boot acknowledge is enabled, the slave has to send acknowledge pattern '010' to the master within 50ms after the CMD line goes LOW. If boot acknowledge is disabled, the slave will not send out acknowledge pattern '010'.

The master can terminate boot mode with the CMD line HIGH.

Boot operation will be terminated when all contents of the enabled boot data are sent to the master. After boot operation is executed, the slave shall be ready for CMD1 operation and the master needs to start a normal MMC initialization sequence by sending CMD1.

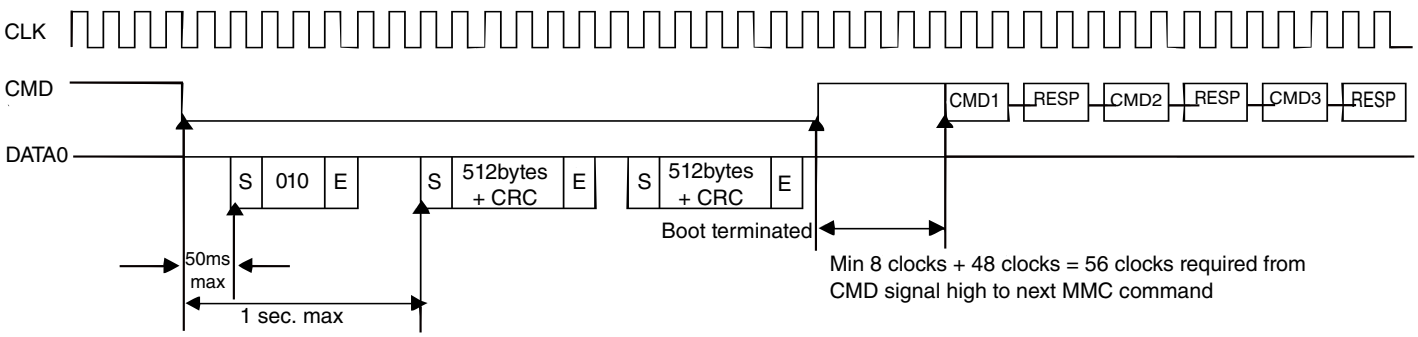


Figure 67-16. MultiMediaCard state diagram (normal boot mode)

67.4.9.2 Alternative boot operation

This boot function is optional for the device. If bit 0 in the extended CSD byte[228] is set to '1', the device supports the alternative boot operation.

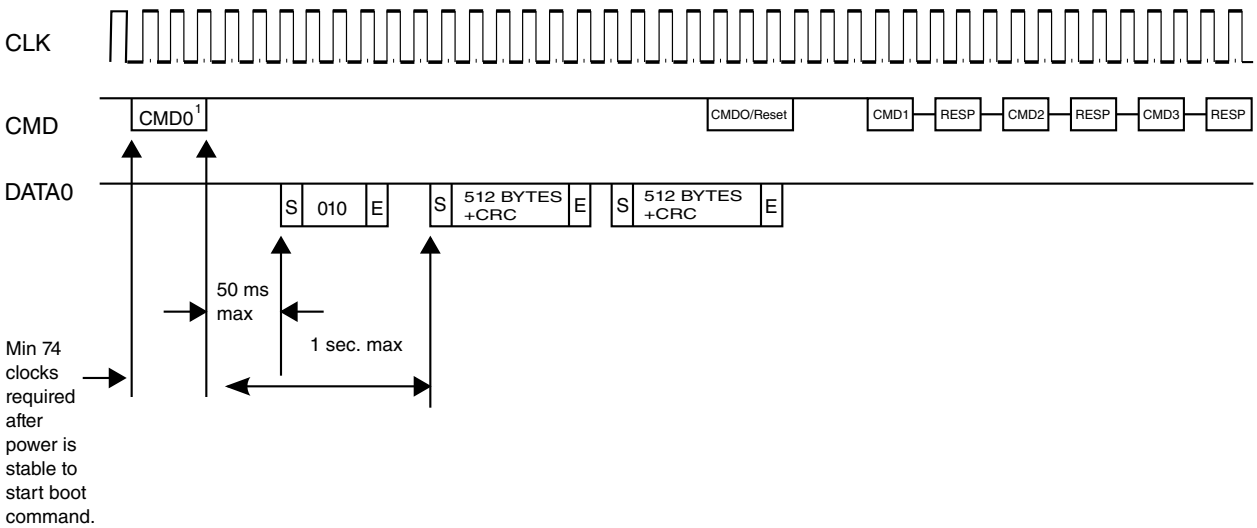
After power-up, if the host issues CMD0 with the argument of 0xFFFFFFFA after 74 clock cycles, before CMD1 is issued or the CMD line goes low, the slave recognizes that boot mode is being initiated and starts preparing boot data internally.

Within 1 second after CMD0 with the argument of 0xFFFFFFFA is issued, the slave starts to send the first boot data to the master on the DATA line(s).

If boot acknowledge is enabled, the slave has to send the acknowledge pattern '010' to the master within 50ms after the CMD0 with the argument of 0xFFFFFFFA is received. If boot acknowledge is disabled, theslave will not send out acknowledge pattern '010'.

The master can terminate boot mode by issuing CMD0 (Reset).

Boot operation will be terminated when all contents of the enabled boot data are sent to the master. After boot operation is executed, the slave shall be ready for CMD1 operation and the master needs to start a normal MMC initialization sequence by sending CMD1.



NOTE 1. CMD0 with argument 0xFFFFFFFA

Figure 67-17. MultiMediaCard state diagram (alternative boot mode)

67.5 Initialization/Application of uSDHC

All communication between the system and cards are controlled by the host. The host sends commands of two types: broadcast and addressed (point-to-point).

Broadcast commands are intended for all cards, such as GO_IDLE_STATE, SEND_OP_COND, ALL_SEND_CID. In Broadcast mode, all cards are in the open-drain mode to avoid bus contention. Refer to Commands for MMC/SD/SDIO for the commands of bc and bcr categories.

After the Broadcast command CMD3 is issued, the cards enter standby mode. Addressed type commands are used from this point. In this mode, the CMD/DATA I/O pads will turn to push-pull mode, to have the driving capability for maximum frequency operation. Refer to Commands for MMC/SD/SDIO for the commands of ac and adtc categories.

67.5.1 Command Send & Response Receive Basic Operation

Assuming the data type WORD is an unsigned 32-bit integer, the below flow is a guideline for sending a command to the card(s):

```
send_command(cmd_index, cmd_arg, other requirements)
{
WORD wCmd; // 32-bit integer to make up the data to write into Transfer Type register, it is
recommended to implement in a bit-field manner
wCmd = (<cmd_index> & 0x3f) >> 24; // set the first 8 bits as '00'+<cmd_index>
set CMDTYP, DPSEL, CICEN, CCCEN, RSTTYP, DTDSEL accorind to the command index;
if (internal DMA is used) wCmd |= 0x1;
if (multi-block transfer) {
          set MSBSEL bit;
          if (finite block number) {
                    set BCEN bit;
                    if (auto12 command is to use) set AC12EN bit;
          }
}
write_reg(CMDARG, <cmd_arg>); // configure the command argument
write_reg(XFERTYP, wCmd); // set Transfer Type register as wCmd value to issue the command
}
wait_for_response(cmd_index)
{
while (CC bit in IRQ Status register is not set); // wait until Command Complete bit is set
read IRQ Status register and check if any error bits about Command are set
if (any error bits are set) report error;
write 1 to clear CC bit and all Command Error bits;
}
```

For the sake of simplicity, the function wait_for_response is implemented here by means of polling. For an effective and formal way, the response is usually checked after the Command Complete Interrupt is received. When doing this, make sure the corresponding interrupt status bits are enabled.

For some scenarios, the response time-out is expected. For instance, after all cards respond to CMD3 and go to the Standby State, no response to the Host when CMD2 is sent. The Host Driver will deal with "fake" errors like this with caution.

67.5.2 Card Identification Mode

When a card is inserted to the socket or the card was reset by the host, the host needs to validate the operation voltage range, identify the cards, request the cards to publish the Relative Card Address (RCA) or to set the RCA for the MMC cards.

67.5.2.1 Card Detect

See the figure below for a flow diagram showing the detection of MMC, SD and SDIO cards using the uSDHC.

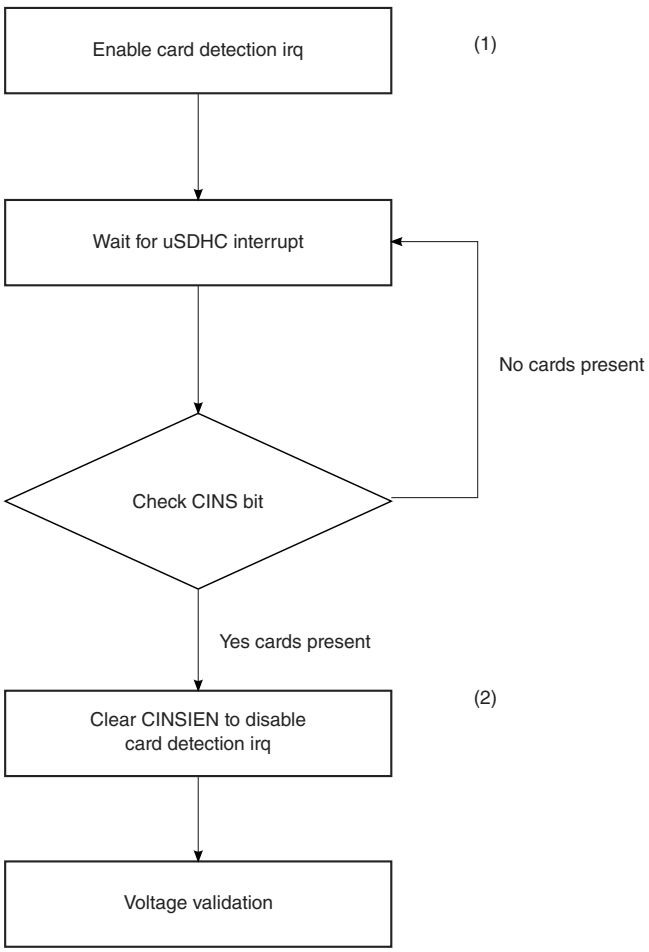


Figure 67-18. Flow Diagram for Card Detection

Here is the card detect sequence:

- Set the CINSIEN bit to enable card detection interrupt
- When an interrupt from the uSDHC is received, check the CINS bit in the Interrupt Status register to see if it was caused by card insertion
- Clear the CINSIEN bit to disable the card detection interrupt and ignore all card insertion interrupts afterwards

67.5.2.2 Reset

The host consists of three types of resets:

- Hardware reset (Card and Host) which is driven by POR (Power On Reset)

- Software reset (Host Only) is initiated by the write operation on the RSTD, RSTC, or RSTA bits of the System Control register to reset the data part, command part, or all parts of the Host Controller, respectively
- Card reset (Card Only). The command, "Go_Idle_State" (CMD0), is the software reset command for all types of MMC cards, SD Memory cards. This command sets each card into the Idle State regardless of the current card state. For an SD I/O Card, CMD52 is used to write an I/O reset in the CCCR. The cards are initialized with a default relative card address (RCA=0x0000) and with a default driver stage register setting (lowest speed, highest driving current capability).

After the card is reset, the host needs to validate the voltage range of the card. See the figure below for the software flow to reset both the uSDHC and the card.

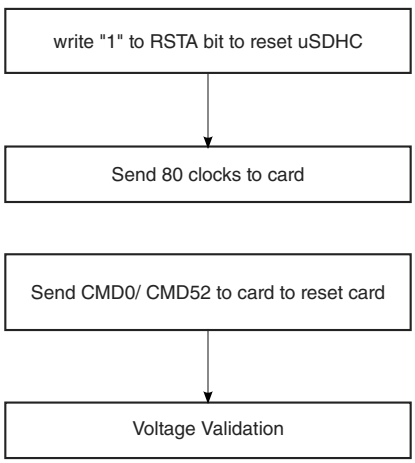


Figure 67-19. Flow Chart for Reset of the uSDHC and SD I/O Card

```
software_reset()
{
set_bit(SYSCTRL, RSTA); // software reset the Host
set DTOCV and SDCLKFS bit fields to get the CLK of frequency around 400kHz
configure IO pad to set the power voltage of external card to around 3.0V
poll bits CIHB and CDIHB bits of PRSSTAT to wait both bits are cleared
set_bit(SYSCTRL, INTIA); // send 80 clock ticks for card to power up
send_command(CMD_GO_IDLE_STATE, <other parameters>); // reset the card with CMD0
or send_command(CMD_IO_RW_DIRECT, <other parameters>);
}
```

67.5.2.3 Voltage Validation

All cards should be able to establish communication with the host using any operation voltage in the maximum allowed voltage range specified in the card specification. However, the supported minimum and maximum values for Vdd are defined in the Operation Conditions Register (OCR) and may not cover the whole range.

Cards that store the CID and CSD data in the preload memory are only able to communicate this information under data transfer Vdd conditions. This means if the host and card have non-common Vdd ranges, the card will not be able to complete the identification cycle, nor will it be able to send CSD data.

Therefore, a special command Send_Op_Cont (CMD1 for MMC), SD_Send_Op_Cont (ACMD41 for SD Memory) and IO_Send_Op_Cont (CMD5 for SD I/O) is used. The voltage validation procedure is designed to provide a mechanism to identify and reject cards which do not match the Vdd range(s) desired by the host. This is accomplished by the host sending the desired Vdd voltage window as the operand of this command. Cards that can't perform the data transfer in the specified range must discard themselves from further bus operations and go into the Inactive State. By omitting the voltage range in the command, the host can query each card and determine the common voltage range before sending out-of-range cards into the Inactive State. This query should be used if the host is able to select a common voltage range or if a notification shall be sent to the system when a non-usable card in the stack is detected.

The following steps show how to perform voltage validation when a card is inserted:

```
voltage_validation(voltage_range_arguement)
{
label the card as UNKNOWN;
send_command(IO_SEND_OP_COND, 0x0, <other parameters are omitted>); // CMD5, check SDIO
operation voltage, command argument is zero
if (RESP_TIMEOUT != wait_for_response(IO_SEND_OP_COND)) { // SDIO command is accepted
          if (0 < number of IO functions) {
                    label the card as SDIO;
                    IORDY = 0;
                    while (!(IORDY in IO OCR response)) { // set voltage range for each IO
function
                              send_command(IO_SEND_OP_COND, <voltage range>, <other
parameter>);
                              wait_for_response(IO_SEND_OP_COND);
                    } // end of while ...
          } // end of if (0 < ...
          if (memory part is present inside SDIO card) Label the card as SDCombo; // this is
an
SD-Combo card
} // end of if (RESP_TIMEOUT ...
if (the card is labelled as SDIO card) return; // card type is identified and voltage range
is
set, so exit the function;
send_command(APP_CMD, 0x0, <other parameters are omitted>); // CMD55, Application specific
CMD
prefix
if (no error calling wait_for_response(APP_CMD, <...>) { // CMD55 is accepted
          send_command(SD_APP_OP_COND, <voltage range>, <...>); // ACMD41, to set voltage
range
for memory part or SD card
          wait_for_response(SD_APP_OP_COND); // voltage range is set
          if (card type is UNKNOWN) label the card as SD;
          return; //
} // end of if (no error ...
else if (errors other than time-out occur) { // command/response pair is corrupted
          deal with it by program specific manner;
} // of else if (response time-out
else { // CMD55 is refuse, it must be MMC card  if (card is already labelled as SDCombo)
{ //
change label
```

```
                    re-label the card as SDIO;
                    ignore the error or report it;
                    return; // card is identified as SDIO card
          } // of if (card is ...
          send_command(SEND_OP_COND, <voltage range>, <...>);
          if (RESP_TIMEOUT == wait_for_response(SEND_OP_COND)) { // CMD1 is not accepted,
either
                    label the card as UNKNOWN;
                    return;
          } // of if (RESP_TIMEOUT ...
} // of else
}
```

67.5.2.4 Card Registry

Card registry for the MMC and SD/SDIO/SD Combo cards are different. For the SD Card, the Identification process starts at a clock rate lower than 400 kHz and the power voltage higher than 2.7 V (as defined by the Card spec). At this time, the CMD line output drives are push-pull drivers instead of open-drain. After the bus is activated, the host will request the card to send their valid operation conditions.

The response to ACMD41 is the operation condition register of the card. The same command shall be send to all of the new cards in the system. Incompatible cards are put into the Inactive State. The host then issues the command, All_Send_CID (CMD2), to each card to get its unique card identification (CID) number. Cards that are currently unidentified (in the Ready State), send their CID number as the response. After the CID is sent by the card, the card goes into the Identification State.

The host then issues Send_Relative_Addr (CMD3), requesting the card to publish a new relative card address (RCA) that is shorter than the CID. This RCA will be used to address the card for future data transfer operations. Once the RCA is received, the card changes its state to the Standby State. At this point, if the host wants the card to have an alternative RCA number, it may ask the card to publish a new number by sending another Send_Relative_Addr command to the card. The last published RCA is the actual RCA of the card.

The host repeats the identification process with CMD2 and CMD3 for each card in the system until the last CMD2 gets no response from any of the cards in system.

For MMC operation, the host starts the card identification process in open-drain mode with the identification clock rate lower than 400 kHz and the power voltage higher than 2.7 V. The open drain driver stages on the CMD line allow parallel card operation during card identification. After the bus is activated the host will request the cards to send their valid operation conditions (CMD1). The response to CMD1 is the "wired OR" operation on the condition restrictions of all cards in the system. Incompatible cards are sent into the Inactive State. The host then issues the broadcast command All_Send_CID (CMD2), asking all cards for their unique card identification (CID) number. All unidentified cards

(the cards in Ready State) simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bit stream. Those cards, whose outgoing CID bits do not match the corresponding bits on the command line in any one of the bit periods, stop sending their CID immediately and must wait for the next identification cycle. Since the CID is unique for each card, only one card can be successfully send its full CID to the host. This card then goes into the Identification State. Thereafter, the host issues Set_Relative_Addr (CMD3) to assign to the card a relative card address (RCA). Once the RCA is received the card state changes to the Stand-by State, and the card does not react in further identification cycles, and its output driver switches from open-drain to push-pull. The host repeats the process, mainly CMD2 and CMD3, until the host receives a time-out condition to recognize the completion of the identification process.

For operation as MMC cards:

```
card_registry()
{
do { // decide RCA for each card until response time-out
          if(card is labelled as SDCombo or SDIO) { // for SDIO card like device
                    send_command(SET_RELATIVE_ADDR, 0x00, <...>); // ask SDIO card to
publish its
RCA
                    retrieve RCA from response;
          } // end if (card is labelled as SDCombo ...
          else if (card is labelled as SD) { // for SD card
                    send_command(ALL_SEND_CID, <...>);
                    if (RESP_TIMEOUT == wait_for_response(ALL_SEND_CID)) break;
                    send_command(SET_RELATIVE_ADDR, <...>);
                    retrieve RCA from response;
          } // else if (card is labelled as SD ...
          else if (card is labelled as MMC) {
                    send_command(ALL_SEND_CID, <...>);
                    rca = 0x1; // arbitrarily set RCA, 1 here for example
                    send_command(SET_RELATIVE_ADDR, 0x1 << 16, <...>); // send RCA at upper
16
bits
          } // end of else if (card is labelled as MMC ...
} while (response is not time-out);
}
```

67.5.3 Card Access

Information about Block Write, Block Read, Suspense Resume, ADMA Usage, Transfer Error, and Card Interrupt are detailed in the sections below.

67.5.3.1 Block Write

Information on Normal Write, DDR Write, and Write with Pause are detailed in the sections below.

67.5.3.1.1 Normal Write

During a block write (CMD24 - 27, CMD60, CMD61), one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block by the host. If the CRC fails, the card shall indicate the failure on the DATA line. The transferred data will be discarded and not written, and all further transmitted blocks (in multiple block write mode) will be ignored.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card detects the block misalignment error and aborts the programming before the beginning of the first misaligned block. The card sets the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer, waits in the Receive-data-State for a stop command. The write operation is also aborted if the host tries to write over a write protected area.

For MMC and SD cards, programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card will report an error and not change any register contents.

For all types of cards, some may require long and unpredictable periods of time to write a block of data. After receiving a block of data and completing the CRC check, the card will begin writing and hold the DATA line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) or other means for SDIO cards at any time, and the card will respond with its status. The responded status indicates whether the card can accept new data or whether the write process is still in progress. The host may deselect the card by issuing a CMD7 (to select a different card) to place the card into the Standby State and release the DATA line without interrupting the write operation. When re-selecting the card, it will reactivate the busy indication by pulling DATA to low if the programming is still in progress and the write buffer is unavailable.

The software flow to write to a card incorporates the internal DMA and the write operation is a multi-block write with the Auto CMD12 enabled. For the other two methods (by means of external DMA or CPU polling status) with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to write to a card is described below:

1. Check the card status, wait until the card is ready for data.
2. Set the card block length/size:

- For SD/MMC cards, use SET_BLOCKLEN (CMD16)
- For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT (CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)

3. Set the uSDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the uSDHC number block register (NOB), nob is 5 (for instance).
5. Disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a write CRC error occurred, or some another error, that occurred during the auto12 command sending and response receiving.

67.5.3.1.2 DDR Write

uSDHC supports dual data rate mode.

The software flow to write to a card in ddr mode is described as below:

1. Check the card status, wait until the card is ready for data.
2. For eMMC4.4 card, block length only can be set to 512byte.
3. Set the uSDHC number block register (NOB), nob is 5 (for instance).
4. Set eMMC4.4 card to high speed mode, use SWITCH(CMD6).
5. Set eMMC4.4 card bus with (4-bit /8-bit ddr mode), use SWITCH(CMD6).
6. Disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer.The DDR_EN bit should be set. The AC12EN bit should also be set.
7. Wait for the Transfer Complete interrupt.
8. Check the status bit to see if a write CRC error occurred, or some another error, that occurred during the auto12 command sending and response receiving.

67.5.3.1.3 Write with Pause

The write operation can be paused during the transfer. Instead of stopping the CLK at any time to pause all the operations, which is also inaccessible to the Host Driver, the Driver can set the Stop At Block Gap Request(SABGREQ) bit in the Protocol Control register to pause the transfer between the data blocks. As there is no time-out condition in a write operation during the data blocks, a write to all types of cards can be paused in this way, and if the DATA0 line is not required to de-assert to release the busy state, no suspend command is needed.

Like in the flow described in Normal Write, the write with pause is shown with the same kind of write operation:

1. Check the card status, wait until card is ready for data.
2. Set the card block length/size:
   - For SD/MMC, use SET_BLOCKLEN (CMD16)
   - For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)
3. Set the uSDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the uSDHC number block register (NOB), nob is 5 (for instance).
5. Disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Set the SABGREQ bit.
7. Wait for the Transfer Complete interrupt.
8. Clear the SABGREQ bit.
9. Check the status bit to see if a write CRC error occurred.
10. Set the CREQ bit to continue the write operation.
11. Wait for the Transfer Complete interrupt.
12. Check the status bit to see if a write CRC error occurred, or some another error, that occurred during the auto12 command sending and response receiving.

The number of blocks left during the data transfer is accessible by reading the contents of the BLKCNT field in the Block Attribute register. As the data transfer and the setting of the SABGREQ bit are concurrent, and the delay of register read and the register setting, the actual number of blocks left may not be exactly the value read earlier. The Driver shall read the value of BLKCNT after the transfer is paused and the Transfer Complete interrupt is received.

It is also possible the last block has begun when the Stop At Block Gap Request is sent to the buffer. In this case, the next block gap is actually the end of the transfer. These types of requests are ignored and the Driver should treat this as a non-pause transfer and deal with it as a common write operation.

When the write operation is paused, the data transfer inside the Host System is not stopped, and the transfer is active until the data buffer is full. Because of this (if not needed), it is recommended to avoid using the Suspend Command for the SDIO card. This is because when such a command is sent, the uSDHC thinks the System will switch to another function on the SDIO card, and flush the data buffer. The uSDHC takes the Resume Command as a normal command with data transfer, and it is left for the Driver to set all the relevant registers before the transfer is resumed. If there is only one block to

send when the transfer is resumed, the MSBSEL and BCEN bits of the Transfer Type register are set as well as the AC12EN bit. However, the uSDHC will automatically send a CMD12 to mark the end of the multi-block transfer.

67.5.3.2 Block Read

Information about Normal Read, DDR Read, Read with Pause, and DLL (Delay Line) in Read Path are detailed in the sections below.

67.5.3.2.1 Normal Read

For block reads, the basic unit of data transfer is a block whose maximum size is stored in areas defined by the corresponding card specification. A CRC is appended to the end of each block, ensuring data transfer integrity. The CMD17, CMD18, CMD53, CMD60, CMD61, and so on, can initiate a block read. After completing the transfer, the card returns to the Transfer State. For multi blocks read, data blocks will be continuously transferred until a stop command is issued.

The software flow to read from a card incorporates the internal DMA and the read operation is a multi-block read with the Auto CMD12 enabled. For the other two methods (by means of external DMA or CPU polling status) with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to read from a card is described below:

1. Check the card status, wait until card is ready for data.
2. Set the card block length/size:
   - For SD/MMC, use SET_BLOCKLEN (CMD16)
   - For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)
3. Set the uSDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the uSDHC number block register (NOB), nob is 5 (for instance).
5. Disable the buffer read ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a read CRC error occurred, or some another error, occurred during the auto12 command sending and response receiving.

67.5.3.2.2 DDR Read

uSDHC supports dual data rate mode.

The software flow to write to a card in ddr mode is described below:

1. Check the card status, wait until the card is ready for data.
2. For eMMC4.4 card, block length only can be set to 512byte.
3. Set the uSDHC number block register (NOB), nob is 5 (for instance).
4. Set eMMC4.4 card to high speed mode, use SWITCH(CMD6).
5. Set eMMC4.4 card bus with (4-bit /8-bit ddr mode), use SWITCH(CMD6).
6. Disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer.The DDR_EN bit should be set. The AC12EN bit should also be set.
7. Wait for the Transfer Complete interrupt.
8. Check the status bit to see if a write CRC error occurred, or some another error, that occurred during the auto12 command sending and response receiving.

67.5.3.2.3 Read with Pause

The read operation is not generally able to pause. Only the SDIO card (and SDCombo card working under I/O mode) supporting the Read Wait feature can pause during the read operation. If the SDIO card supports Read Wait (SRW bit in CCCR register is 1), the Driver can set the SABGREQ bit in the Protocol Control register to pause the transfer between the data blocks.

Before setting the SABGREQ bit, make sure the RWCTL bit in the Protocol Control register is set, otherwise the uSDHC will not assert the Read Wait signal during the block gap and data corruption occurs. It is recommended to set the RWCTL bit once the Read Wait capability of the SDIO card is recognized.

Like in the flow described in Normal Read, the read with pause is shown with the same kind of read operation:

1. Check the SRW bit in the CCR register on the SDIO card to confirm the card supports Read Wait.
2. Set the RWCTL bit.
3. Check the card status and wait until the card is ready for data.
4. Set the card block length/size:
   - For SD/MMC, use SET_BLOCKLEN (CMD16)
   - For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)

5. Set the uSDHC block length register to be the same as the block length set for the card in Step 2.
6. Set the uSDHC number block register (NOB), nob is 5 (for instance).
7. Disable the buffer read ready interrupt, configure the DMA setting and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set
8. Set the SABGREQ bit.
9. Wait for the Transfer Complete interrupt.
10. Clear the SABGREQ bit.
11. Check the status bit to see if read CRC error occurred.
12. Set the CREQ bit to continue the read operation.
13. Wait for the Transfer Complete interrupt.
14. Check the status bit to see if a read CRC error occurred, or some another error, occurred during the auto12 command sending and response receiving.

Like the write operation, it is possible to meet the ending block of the transfer when paused. In this case, the uSDHC will ignore the Stop At Block Gap Request and treat it as a command read operation.

Unlike the write operation, there is no remaining data inside the buffer when the transfer is paused. All data received before the pause will be transferred to the Host System. No matter if the Suspend Command is sent or not, the internal data buffer is not flushed.

If the Suspend Command is sent and the transfer is later resumed by means of a Resume Command, the uSDHC takes the command as a normal one accompanied with data transfer. It is left for the Driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, the MSBSEL and BCEN bits of the Transfer Type register are set, as well as the AC12EN bit. However, the uSDHC will automatically send the CMD12 to mark the end of multi-block transfer.

67.5.3.2.4 DLL (Delay Line) in Read Path

The DLL(Delay Line) is newly added to assist in sampling read data. The DLL provides the ability to programmatically select a quantized delay (in fractions of the clock period) regardless of on-chip variations such as process, voltage and temperature (PVT).

The reasons why the DLL is needed for uSDHC are 1.) the path of read data traveling from card to host varies. 2.) in SD/MMC DDR mode the minimum input setup and hold time are both at 2.5 ns. The data sampling window is so small that the delay of loopback clock needs to be accurate and consistent regardless of PVT. The DLL takes the divided card_clk as the reference clock and loopback clock as the input clock. It then generates a delayed version of the input clock according to the programmed target delay.

The DLL can be disabled or bypassed, and it can also be manually set for a fixed delay in override mode. The override value set is the number of delay cells. In override mode, there is no need to set the DLL_enable. Another DLL mode is target value mode. In this mode, the DLL will automatically adjust the number of delay cells according to target value set by user and PVT changes. Be aware that target value is in units of 1/32 of the clock reference period. If the card_clk is 100Mhz, then the reference clock period is 10ns, setting target vaule of 16 means 5ns =(16/32)*10ns. Software can disable automatic update by setting dll_gate_update bit. Please refer to Figure 67-20.

Since the user may change the frequency of the card_clk from time to time by changing SDCLKFS[7:0]/DVS[3:0], the software must adjust the delay value to ensure it works correctly when the reference clock (card_clk) is changed. The following is the correct flow, which should be ignored if DLL_CTRL_ENABLE is not set.

Step 1: Set DLL_CTRL_RESET bit

Step 2: Configure the SDCLKFS[7:0] and DVS[3:0]

Step 3: Wait until SDSTB is asserted

Step 4: clear DLL_CTRL_RESET bit

Step 5: Wait until both DLL_STS_SLV_LOCK and DLL_STS_REF_LOCK are asserted

Step 6: set DLL_CTRL_SLV_FORCE_UPD

Step 7: clear DLL_CTRL_SLV_FORCE_UPD

NOTE:Software should make sure the DLL_CTRL_SLV_FORCE_UPD is lasted for at least one card_clk. So software may need to add some delay between step6 and step7.

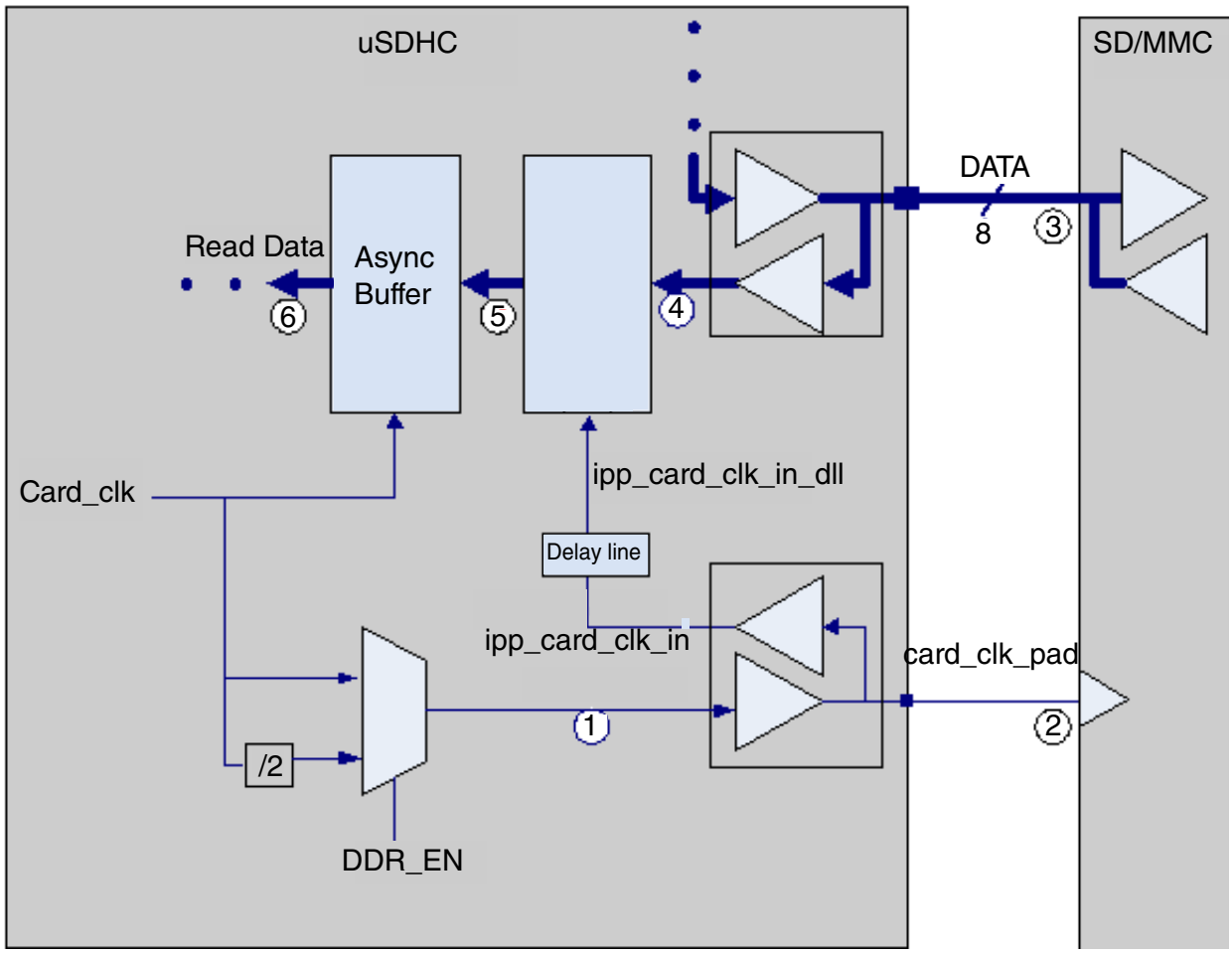


Figure 67-20. DLL(Delay Line) in Read Path

67.5.3.3 Suspend Resume

The uSDHC supports the Suspend Resume operations of SDIO cards, although slightly differently than the suggested implementation of Suspend in the SDIO card specification.

67.5.3.3.1 Suspend

After setting the SABGREQ bit, the Host Driver may send a Suspend command to switch to another function of the SDIO card. The uSDHC does not monitor the content of the response, so it doesn't know if the Suspend command succeeded or not. Accordingly, it doesn't de-assert Read Wait for read pause. To solve this problem, the Driver shall not mark the Suspend command as a "Suspend", (i.e. setting the CMDTYP bits to 01). Instead, the Driver shall send this command as if it were a normal command, and only

when the command succeeds, and the BS bit is set in the response, can the Driver send another command marked as "Suspend" to inform the uSDHC that the current transfer is suspended. As shown in the following sequence for Suspend operation:

1. Set the SABREQ bit to pause the current data transfer at block gap.
2. After the BGE bit is set, send the Suspend command to suspend the active function. The CMDTYP bit field must be 2'b00.
3. Check the BS bit of the CCCR in the response. If it is 1, repeat this step until the BS bit is cleared or abandon the suspend operation according to the Driver strategy.
4. Send another normal I/O command to the suspended function. The CMDTYP of this command must be 2'b01, so the uSDHC can detect this special setting and be informed that the paused operation has successfully suspended. If the paused transfer is a read operation, the uSDHC stops driving DATA2 and goes to the idle state.
5. Save the context registers in the system memory for later use, including the DMA System Address Register (for internal DMA operation), and the Block Attribute Register.
6. Begin operation for another function on the SDIO card.

67.5.3.3.2 Resume

To resume the data transfer, a Resume command shall be issued:

1. To resume the suspended function, restore the context register with the saved value in step #5 of the Suspend operation above.
2. Send the Resume command. In the Transfer Type register, all bit fields are set to the value as if this were another ordinary data transfer, instead of a transfer resume (except the CMDTYP is set to 2'b10).
3. If the Resume command has responded, the data transfer will be resumed.

67.5.3.4 ADMA Usage

To use the ADMA in a data transfer, the Host Driver must prepare the correct descriptor chain prior to sending the read/write command.

The steps to prepare the correct descriptor chain are:

1. Create a descriptor to set the data length that the current descriptor group is about to transfer. The data length should be even numbers of the block size.
2. Create another descriptor to transfer the data from the address setting in this descriptor. The data address must be at a page boundary (4kB address aligned).
3. If necessary, create a Link descriptor containing the address of the next descriptor. The descriptor group is created in steps 1 ~ 3.

4. Repeat steps 1 ~ 3 until all descriptors are created.
5. In the last descriptor, set the End flag to 1 and make sure the total length of all descriptors match the product of the block size and block number configured in the Block Attribute Register.
6. Set the ADMA System Address Register to the address of the first descriptor and set the DMAS field in the Protocol Control Register to 01 to select the ADMA.
7. Issue a write or read command with the DMAEN bit set to 1 in the Transfer Type Register.

Steps 1 ~ 5 are independent of step 6, so step 6 can finish before steps 1 ~ 5. Regarding the descriptor configuration, it is recommended not to use the Link descriptor as it requires extra system memory access.

67.5.3.5 Transfer Error

Information about CRC, Internal DMA, Transfer ADMA, and Auto CMD12 Errors are detailed in the sections below.

67.5.3.5.1 CRC Error

It is possible at the end of a block transfer, that a write CRC status error or read CRC error occurs. For this type of error the latest block received shall be discarded. This is because the integrity of the data block is not guaranteed. It is recommended to discard the following data blocks and re-transfer the block from the corrupted one.

For a multi-block transfer, the Host Driver shall issue a CMD12 to abort the current process and start the transfer by a new data command. In this scenario, even when the AC12EN and BCEND bits are set, the uSDHC does not automatically send a CMD12 because the last block is not transferred. On the other hand, if it is within the last block that the CRC error occurs, an Auto CMD12 will be sent by the uSDHC. In this case, the Driver shall re-send or re-obtain the last block with a single block transfer.

67.5.3.5.2 Internal DMA Error

During the data transfer with internal Simple DMA, if the DMA engine encounters some error on the AHB bus, the DMA operation is aborted and DMA Error interrupt is sent to the Host System. When acknowledged by such an interrupt, the Driver shall calculate the start address of data block in which the error occurs.

The start address can be calculated by either:

1. Reading the DMA System Address register. The error occurs during the previous burst. Taking the block size, the previous burst length and the start address of the

next burst transfer into account, it is straight forward to obtain the start address of the corrupted block.
2. Reading the BLKCNT field of the Block Attribute register. By the number of blocks left, the total number to transfer, the start address of transfer, and the size of each block, the start address of corrupted block can be determined. When the BCEN bit is not set, the contents of the Block Attribute register does not change, so this method does not work.

When a DMA error occurs, it is recommended to abort the current transfer by means of a CMD12 (for multi block transfer), apply a reset for data, and re-start the transfer from the corrupted block to recover from the error.

67.5.3.5.3 Transfer ADMA Error

There are 3 kinds of possible ADMA errors. The AHB transfer, invalid descriptor, and data-length mismatch errors. Whenever these errors occur, the DMA transfer stops and the corresponding error status bit is set.

For acknowledging the status, the Host Driver should recover the error as shown below and re-transfer from the place of interruption.

1. AHB transfer error: Such errors may occur during data transfer or descriptor fetch. For either scenario, it is recommended to retrieve the transfer context, reset for the data part and re-transfer the block that was corrupted, or the next block if no block is corrupted.
2. Invalid descriptor error: For such errors, it is recommended to retrieve the transfer context, reset for the data part and re-create the descriptor chain from the invalid descriptor and issue a new transfer. As the data to transfer now may be less than the previous setting, the data length configured in the new descriptor chain should match the new value.
3. Data-length mismatch error: It is similar to recover from this error. The Host Driver polls relating registers to retrieve the transfer context, apply a reset for the data part, configure a new descriptor chain, and make another transfer if there is data left. Like the previous scenario of the invalid descriptor error, the data length must match the new transfer.

67.5.3.5.4 Auto CMD12 Error

After the last block of the multi block transfer is sent or received, and the AC12EN bit is set when the data transfer is initiated by the data command, the uSDHC automatically sends a CMD12 to the card to stop the transfer.

When errors with this command occur, it is recommended to the Driver to deal with the situations in the following manner:

1. Auto CMD12 response time-out. It is not certain whether the command is accepted by the card or not. The Driver should clear the Auto CMD12 error status bits and re-send the CMD12 until it is accepted by the card.
2. Auto CMD12 response CRC error. Since card responds to the CMD12, the card will abort the transfer. The Driver may ignore the error and clear the error status bit.
3. Auto CMD12 conflict error or not sent. The command is not sent, so the Driver shall send a CMD12 manually.

67.5.3.6 Card Interrupt

The external cards can inform the Host Controller by means of some special signals. For the SDIO card, it can be the low level on the DATA1 line during some special period. The uSDHC only monitors the DATA1 line and supports the SDIO interrupt.

When the SDIO interrupt is captured by the uSDHC, and the Host System is informed by the uSDHC asserting the uSDHC interrupt line, the interrupt service from the Host Driver is called.

As the interrupt source is controlled by the external card, the interrupt from the SDIO card must be serviced before the CINT bit is cleared by written. Refer to Card Interrupt Handling for the card interrupt handling flow.

67.5.4 Switch Function

A switch command must be issued by the Host Driver to enable new features added to the SD/MMC spec. SD/MMC cards can transfer data at bus widths other than 1-bit. Different speed mode are also defined. To enable these features, a switch command shall be issued by the Host Driver.

For SDIO cards, the high speed mode/DDR50/SDR50/SDR104 are enabled by writing the EHS bit in the CCCR register after the SHS bit is confirmed. For SD cards, the high speed mode/DDR50/SDR50/SDR104 are queried and enabled by a CMD6 (with the mnemonic symbol as SWITCH_FUNC). For MMC cards, the high speed mode/HS200 are queried by a CMD8 and enabled by a CMD6 (with the mnemonic symbol as SWITCH).

The SDR4-bit, SDR8-bit, DDR4-bit and DDR8-bit width of the MMC is also enabled by the SWITCH command, but with a different argument.

These new functions can also be disabled by a software reset. For SDIO cards it can be done by setting the RES bit in the CCCR register. For other cards, it can be accomplished by issuing a CMD0. This method of restoring to the normal mode is not recommended because a complete identification process is needed before the card is ready for data transfer.

For the sake of simplicity, the following pseudocode examples do not show current capability check, which is recommended in the function switch process.

67.5.4.1 Query, Enable and Disable SDIO High Speed Mode

```
enable_sdio_high_speed_mode(void)
{
send CMD52 to query bit SHS at address 0x13;
if (SHS bit is '0') report the SDIO card does not support high speed mode and return;
send CMD52 to set bit EHS at address 0x13 and read after write to confirm EHS bit is set;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of around 50MHz;
(data transactions like normal peers)
}
disable_sdio_high_speed_mode(void)
{
send CMD52 to clear bit EHS at address 0x13 and read after write to confirm EHS bit is
cleared;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

67.5.4.2 Query, Enable and Disable SD High Speed Mode/SDR50/ SDR104/DDR50

```
enable_sd_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0xFFFFFx and read 64 bytes of data accompanying the R1 response;
(high speed mode,x=1; SDR50,x=2; SDR104,x=3; DDR50,x=4;)
wait data transfer done bit is set;
check if the bit x of received 512 bits is set;
if (bit 401 is '0') report the SD card does not support high speed mode and return;
if (bit 402 is '0') report the SD card does not support SDR50 mode and return;
if (bit 403 is '0') report the SD card does not support SDR104 mode and return;
if (bit 404 is '0') report the SD card does not support DDR50 mode and return;
send CMD6, with argument 0x80FFFFFx and read 64 bytes of data accompanying the R1 response;
(high speed mode,x=1; SDR50,x=2; SDR104 x=3; DDR50 x=4;)
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of around 50MHz for high speed mode, 100Mhz for SDR50, 200Mhz for SDR104, 50Mhz for
DDR50;
(data transactions like normal peers)
}
disable_sd_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0x80FFFFF0 and read 64 bytes of data accompanying the R1 response;
check if the bit field 379~376 is 0xF;
```

```
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

67.5.4.3 Query, Enable and Disable MMC High Speed Mode

```
enable_mmc_high_speed_mode(void)
{
send CMD9 to get CSD value of MMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the MMC does not support high speed mode and
return;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of MMC;
extract the value of CARD_TYPE field to check the 'high speed mode' in this MMC is 26MHz or
52MHz;
send CMD6 with argument 0x1B90100;
send CMD13 to wait card ready (busy line released);
send CMD8 to get EXT_CSD value of MMC;
check if HS_TIMING byte (byte number 185) is 1;
if (HS_TIMING is not 1) report MMC switching to high speed mode failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of around 26MHz or 52MHz according to the CARD_TYPE;
(data transactions like normal peers)
}
disable_mmc_high_speed_mode(void)
{
send CMD6 with argument 0x2B90100;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of MMC;
check if HS_TIMING byte (byte number 185) is 0;
if (HS_TIMING is not 0) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 20MHz;
(data transactions like normal peers)
}
```

67.5.4.4 Set MMC Bus Width

```
change_mmc_bus_width(void)
{
send CMD9 to get CSD value of MMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the MMC does not support multiple bit width and
return;
send CMD6 with argument 0x3B70x00; (8-bit(dual data rate), x=6; 4-bit(dual data rate), x=5;8-
bit, x=2; 4-bit, x=1; 1-bit, x=0)
send CMD13 to wait card ready (busy line released);
(data transactions like normal peers)
}
```

67.5.5 ADMA Operation

Code for ADMA1 Operation and ADMA2 Operation can be found here.

67.5.5.1 ADMA1 Operation

```
Set_adma1_descriptor
{
if (to start data transfer) {
// Make sure the address is 4KB align.
Set 'Set' type descriptor;
{
Set Act bits to 01;
Set [31:12] bits data length (byte unit);
}
Set 'Tran' type descriptor;
{
Set Act bits to 10;
Set [31:12] bits address (4KB align);
}
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to 11;
Set [31:12] bits the next descriptor address (4KB aligned);
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set End bit to 1;
}
if (to generate interrupt for this descriptor) {
Set Int bit to 1;
}
Set Valid bit to 1;
}
```

67.5.5.2 ADMA2 Operation

```
Set_adma2_descriptor
{
if (to start data transfer) {
// Make sure the address is a 32-bit boundary (lower 2-bit are always '00').
Set higher 32-bit of descriptor for this data transfer initial address;
Set [31:16] bits data length (byte unit);
Set Act bits to '10';
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to '11';
// Make sure the address is 32-bit boundary (lower 2-bit are always set to '00').
Set higher 32-bit of descriptor for the next descriptor address;
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set 'End' bit '1';
}
if (to generate interrupt for this descriptor) {
Set 'Int' bit '1';
}
Set the 'Valid' bit to '1';
}
```

67.5.6 Fast Boot Operation

67.5.6.1 Normal fast boot flow

1. Software must configure init_active bit (system control register bit 27) to make sure 74 card clocks are finished.
2. Software must configure the MMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 0 (normal fast boot), and bit 4 to select the ack mode or not. If the data will be sent through DMA mode, the software should configure bit 7 to enable the automatic stop at block gap feature, and configure bit 3-bit 0 to select the ack timeout value according to the SD CLK frequency.
3. Software then needs to configure the Block Attributes Register to set the block size and count. If in DDR fast boot mode, the block size only can be configured to 512 bytes.
4. Software must configure the Protocol control register to set DTW (data transfer width). If in DDR fast boot mode, DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software needs to configure the Command Argument Register to set argument if needed (no need in normal fast boot).
6. Software must configure the Transfer Type Register to start the boot process. In normal boot mode, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN and DMAEN are kept at the default value. DPSEL bit is set to 1, DTDSEL is set to 1, MSBSEL is set to 1.
7. DMAEN should be configured as 0 in polling mode. And if BCEN is configured as 1, it is recommended to configure the number of blocks in the Block Attributes Register to the maximum value. If in DDR fast boot mode, DDR_EN needs to be set to 1.
8. When the step 6 is configured, the boot process will begin. Software needs to poll the data buffer ready status to read the data from the buffer in time. If a boot timeout happens (ack times out or the first data read times out), an interrupt will be triggered, and software must configure MMC Boot Register to bit 6 to 0 to disable boot. This makes CMD high, then after at least 56 clocks, it is ready to begin a normal initialization process.
9. If there is no timeout, software needs to determine when the data read is finished and then configure MMC Boot Register bit 6 to 0 to disable boot. This will make CMD line high and command completed asserted. After at least 56 clocks, it is ready to begin normal initialization process.
10. Reset the host and then can begin the normal process.

67.5.6.2 Alternative fast boot flow

1. Software needs to configure init_active bit (system control register bit 27) to make sure 74 card clocks are finished.
2. Software needs to configure MMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 1 (alternative boot), and bit 4 to select the ack mode or not. If data needs to be sent through DMA mode, then configure bit 7 to enable the automatic stop at block gap feature. Software should also configure bit 3-bit 0 to select the ack timeout value according to the SD clock frequency.
3. Software then needs to configure Block Attributes Register to set the block size and count. If in DDR fast boot mode, the block size only can be configured to 512 bytes.
4. Software needs to configure the Protocol control register to set the DTW (data transfer width). If in ddr fast boot mode, DTW only can be configure to 4-bit/8-bit dataline mode.
5. Software needs to configure Command Argument Register to set argument to 0xFFFFFFFA.
6. Software needs to configure the Transfer Type Register to start the boot process by CMD0 with 0xFFFFFFFA argument . In alternative boot, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN and DMAEN are kept default value. DPSEL bit is set to 1, DTDSEL is set to 1, MSBSEL is set to 1. Note DMAEN should be configured as 0 in polling mode. And if BCEN is configured as 1 in polling mode, it is recommended to configure the block count in the Block Attributes Register to the maximum value. If in DDR fast boot mode, DDR_EN needs to be set to 1.
7. When the step 6 is configured, the boot process will begin. Software needs to poll the data buffer ready status to read the data from the buffer in time. If there is a boot timeout (ack data timeout in 50ms or data timeout in 1s), the host will send out the interrupt and software needs to send CMD0 with reset and then configure the boot enable bit to 0 to stop this process..
8. If there is no time out, software needs to decide when to stop the boot process, and send out the CMD0 with reset and then after the command is completed, configure the MMC Boot Register bit 6 to stop the process. After 8 clocks from the command completion, the slave (card) is ready for the identification step.
9. Reset the host and then begin the normal process.

67.5.6.3 Fast boot application case (in DMA mode)

In the boot application case, because the image destination and the image size are contained in the beginning of the image, it is necessary to switch DMA parameters on the fly during MMC fast boot.

In fast boot, the host can use ADMA2 (Advanced DMA2) with two destinations.

The detail flow is described below:

1. Software needs to configure INIT_ACTIVE bit (system control register bit 27) to make sure 74 card clocks are finished.
2. Software needs to configure the MMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot); and bit 5 to 0 (normal fast boot) or 1 (alternative boot); and bit 4 to select the ack mode or not. In DMA mode, configure bit 7 to 1 to enable the automatic stop at block gap feature. Also configure bits[31-16] to set the (BLK_CNT - VALUE1).Here VALUE1 is the value of the block count that needs to transfer the first time, so that that the hostwill stop at the block gap when the uSDHC controller gets VAULE1 blocks from the device. Also configure bits[3-0] to select the ack timeout value according to the SD clock frequency.
3. Software then needs to configure the Block Attributes Register to set block size and count. If in DDR fast boot mode, the block size only can be configured to 512 bytes. In DMA mode, it is recommended to set the block count (BLK_CNT) to the max value (16'hffff).
4. Software needs to configure Protocol Control Register to set DTW (data transfer width). If in DDR fast boot mode, the DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software enbable ADMA2 by configuring Protocol Control Register bits [9-8].
6. Software need to set at least three pairs ADMA2 descriptor in boot memory (ie, in IRAM, at least 6 word). The first pair descriptor define the start address (ie, IRAM)and data length(ie,512byte*VALUE1) of fisrt part boot code. Software also need to set the second pair descriptor, the second start address (any value that is writeable), data length is suggest to set 1~2word(record as VALUE2). Note: the second couple desc also transfer useful data even at lease 1 word. Because our ADMA2 can't support 0 data_length data transfer descriptor.
7. Software needs to configure Command Argument Register to set argument to 0xFFFFFFFA in alternative fast boot, and don't need set in normal fast boot.
8. Software needs to configure Transfer Type Register to start the boot process . CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN and DMAEN are kept default value. DPSEL bit is set to 1, DTDSEL is set to 1, MSBSEL is set to 1. DMAEN is configured as 1 in DMA mode. And if BCEN is configured as 1, better to configure blk no in Bock Attributes Register to the max value. And if in ddr fast boot mode, DDR_EN need to be set to 1.
9. When the step 8 is configured, boot process will begin, the first VALUE1 block number data has transfer. Software need to polling TC bit (bit1 in Interrupt Status Register) to determine first transfer is end. Also software need to polling BGE bit (bit2 in Interrupt Status Register) to determine if first transfer stop at block gap.

10. When TC, BGE bit is 1, . SW can analyzes the first code of VALUE1 block, initializes the new memory device, if required, and sets the third pair of descriptors to define the start address and length of the remaining part of boot code(VALUE3 the remain boot code block). Remeber set the last descriptor with END.
11. Software needs to configure MMC Boot Register (offset 0xc4) again. Set bit 6 to 1(enable boot); and bit 5 to 0(normal fast boot), to 1(alternative boot); and bit 4 to select the ack mode or not. In DMA mode, configure bit 7 to 1 for enable automatically stop at block gap feature. Also configure bit31-bit16 to set the *(BLK_CNT - (VALUE1+1+VALUE3))*, that host will stop at block gap when *the uSDHC controller gets (VALUE1+1+VALUE3)) blocks from device totally include the blocks received in step 9*. And need to configure bit 3-bit0 to select the ack timeout value according to the sd clk frequence. Please note, Software doesn't need to configure the *BLK_CNT* again, because it's counted down automatically by the uSDHC controller.
12. Software needs to clear TC and BGE bit. And software needs to clear SABGREQ(bit 16 in Protocol control register), and set CREQ(bit17 Protocol control register) to 1 to resume the data transfer. Host will transfer the VALUE2 and VALUE3 data to the destination that is set by descriptor.
13. Software need to polling BGE bit to determine if the fast boot is over.

Note:

1. When ADMA boot flow is started, for uSDHC, it is like a normal ADMA read operation. So setting ADMA2 descriptor as the normal ADMA2 transfer.
2. Need a few words length memory to keep descriptor.
3. For the 1~2 word data in second descriptor setting, it is the useful data, so software need to deal the data due to the application case.

67.6 Commands for MMC/SD/SDIO

A table containing the list of commands for the MMC/SD/SDIO cards can be found here.

Refer to the corresponding specifications for more details about the command information.

There are four kinds of commands defined to control the MultiMediaCard:

1. broadcast commands (bc), no response.
2. broadcast commands with response (bcr), response from all cards simultaneously.
3. addressed (point-to-point) commands (ac), no data transfer on the DATA.
4. addressed (point-to-point) data transfer commands (adtc).

Response: a response is a token which is sent from the card to the host as an answner to a previously received command. A response is transferred serially on the CMD line.

Table 67-6. Commands for MMC/SD/SDIO Cards

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	Resets all MMC and SD memory cards to idle state.
CMD1	bcr	[31:0] OCR without busy	R3	SEND_OP_COND	Asks all MMC and SD Memory cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line.
CMD3[1]	ac	[31:6] RCA [15:0] stuff bits	R1 R6 (SDIO)	SET/ SEND_RELATIVE_ADDR	Assigns relative address to the card.
CMD4	bc	[31:0] DSR [15:0] stuff bits	-	SET_DSR	Programs the DSR of all cards.
CMD5	bc	[31:0] OCR without busy	R4	IO_SEND_OP_COND	Asks all SDIO cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD6[2]	adtc	[31] Mode 0: Check function 1: Switch function [30:8] Reserved for function groups 6 ~ 3 (All 0 or 0xFFFF) [7:4] Function group1 for command system [3:0] Function group2 for access mode	R1	SWITCH_FUNC	Checks switch ability (mode 0) and switch card function (mode 1). Refer to "SD Physical Specification V1.1" for more details.
CMD6[3]	ac	[31:26] Set to 0 [25:24] Access [23:16] Index [15:8] Value [7:3] Set to 0 [2:0] Cmd Set	R1b	SWITCH	Switches the mode of operation of the selected card or modifies the EXT_CSD registers. Refer to "The MultiMediaCard System Specification Version 4.0 Final draft 2" for more details.
CMD7	ac	[31:6] RCA [15:0] stuff bits	R1b	SELECT/ DESELECT_CARD	Toggles a card between the stand-by and transfer states or between the programming and disconnect states. In both cases, the card is selected by its own relative address and gets deselected by any other address. Address 0 deselects all.

Table continues on the next page...

Table 67-6. Commands for MMC/SD/SDIO Cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD8	adtc	[31:0] stuff bits	R1	SEND_EXT_CSD	The card sends its EXT_CSD register as a block of data, with a block size of 512 bytes.
CMD9	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card-identification (CID) on the CMD line.
CMD11	adtc	[31:0] data address	R1	READ_DAT_UNTIL_STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD12	ac	[31:0] stuff bits	R1b	STOP_TRANSMISSION	Forces the card to stop transmission.
CMD13	ac	[31:6] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	ac	[31:6] RCA [15:0] stuff bits	-	GO_INACTIVE_STATE	Sets the card to inactive state in order to protect the card stack against communication breakdowns.
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Sets the block length (in bytes) for all following block commands (read and write). Default block length is specified in the CSD.
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a stop command.
CMD19	Reserved				
CMD20	adtc	[31:0] data address	R1	WRITE_DAT_UNTIL_STOP	Writes data stream from the host, starting at the given address, until a STOP_TRANSMISION follows.
CMD21-23	Reserved				
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows.
CMD26	adtc	[31:0] stuff bits	R1	PROGRAM_CID	Programming of the card identification register. This command shall be issued only once per card. The card contains hardware to prevent this operation

Table continues on the next page...

Table 67-6. Commands for MMC/SD/SDIO Cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
					after the first programming. Normally this command is reserved for the manufacturer.
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits.
CMD31	Reserved				
CMD32	ac	[31:0] data address	R1	TAG_SECTOR_START	Sets the address of the first sector of the erase group.
CMD33	ac	[31:0] data address	R1	TAG_SECTOR_END	Sets the address of the last sector in a continuous range within the selection of a single sector to be selected for erase.
CMD34	ac	[31:0] data address	R1	UNTAG_SECTOR	Removes one previously selected sector from the erase selection.
CMD35	ac	[31:0] data address	R1	TAG_ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase.
CMD36	ac	[31:0] data address	R1	TAG_ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be selected for erase.
CMD37	ac	[31:0] data address	R1	UNTAG_ERASE_GROUP	Removes one previously selected erase group from the erase selection.
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erase all previously selected sectors.
CMD39	ac	[31:0] RCA [15] register write flag [14:8] register address [7:0] register data	R4	FAST_IO	Used to write and read 8-bit (register) data fields. The command addresses a card, and a register, and provides the data for writing if the write flag is set. The R4 response contains data read from the address register. This

Table continues on the next page...

Table 67-6. Commands for MMC/SD/SDIO Cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
					command accesses application dependent registers which are not defined in the MMC standard.
CMD40	bcr	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode.
CMD41	Reserved				
CDM42	adtc	[31:0] stuff bits	R1b	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.
CMD43~51	Reserved				
CMD52	ac	[31:0] stuff bits	R5	IO_RW_DIRECT	Access a single register within the total 128k of register space in any I/O function.
CMD53	ac	[31:0] stuff bits	R5	IO_RW_EXTENDED	Accesses a multiple I/O register with a single command. Allows the reading or writing of a large number of I/O registers.
CMD54	Reserved				
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather that a standard command.
CMD56	adtc	[31:1] stuff bits [0]: RD/WR	R1b	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose / application specific commands. The size of the data block is set by the SET_BLOCK_LEN command.
CMD57-59	Reserved				
CMD60	adtc	[31] WR [30:24] stuff bits [23:16] address [15:8] stuff bits [7:0] byte count	R1b	RW_MULTIPLE_REGISTER	These registers are used to control the behavior of the device and to retrieve status information regarding the operation of the device. All Status and Control registers are WORD (32-bit) in size and are WORD aligned. CMD60 shall be used to read and write these registers.
CMD61	adtc	[31] WR [30:16] stuff bits [15:0] data unit count	R1b	RW_MULTIPLE_BLOCK	The host issues a RW_MULTIPLE_BLOCK (CMD61) to begin the data transfer.
CMD62-63	Reserved				
ACMD6[4]	ac	[31:2] stuff bits [1:0] bus width	R1	SET_BUS_WIDTH	Defines the data bus width ('00'=1bit or '10'=4bit bus) to be used for data transfer. The allowed data bus widths are given in SCR register.

Table continues on the next page...

Table 67-6. Commands for MMC/SD/SDIO Cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
ACMD13[4]	adtc	[31:0] stuff bits	R1	SD_STATUS	Send the SD Memory Card status.
ACMD22[4]	adtc	[31:0] stuff bits	R1	SEND_NUM_WR_SECTORS	Send the number of the written sectors (without errors). Responds with 32-bit plus the CRC data block.
ACMD23[4]	ac	[31:23] stuff bits [22:0] Number of blocks	R1	SET_WR_BLK_ERASE_COUNT	Set the number of write blocks to be pre-erased before writing (to be used for fast Multiple Block WR command). "1"=default(one write block).
ACMD41[4]	bcr	[31:0] OCR	R3	SD_APP_OP_COND	Asks the accessed card to send its operating condition register (OCR) contents in the response on the CMD line.
ACMD42[4]	ac	[31:1] stuff bits [0] set_cd	R1	SET_CLR_CARD_DETECT	Connect(1)/Disconnect(0) the 50KOhm pull-up resistor on CD_B/DATA3 of the card.
ACMD51[4]	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

1. CMD3 differs for MMC and SD cards. For MMC cards, it is referred to as SET_RELATIVE_ADDR, with a response type of R1. For SD cards, it is referred to as SEND_RELATIVE_ADDR, with a response type of R6 (with RCA inside).
2. CMD6 differs completely between high speed MMC cards and high speed SD cards. Command SWITCH_FUNC is for high speed SD cards.
3. Command SWITCH is for high speed MMC cards . The Index field can contain any value from 0-255, but only values 0-191 are valid. If the Index value is in the 192-255 range the card does not perform any modification and the SWITCH_ERROR status bit in the EXT_CSD register is set. The Access Bits are shown in Table 2.
4. ACMDs shall be preceded with the APP_CMD command. (Commands listed are used for SD only, other SD commands not listed are not supported on this module).

The Access Bits for the EXT_CSD Access Modes are shown below.

Table 67-7. EXT_CSD Access Modes

Bits	Access Name	Operation
00	Command Set	The command set is changed according to the Cmd Set field of the argument
01	Set Bits	The bits in the pointed byte are set, according to the 1 bits in the Value field.
10	Clear Bits	The bits in the pointed byte are cleared, according to the 1 bits in the Value field.
11	Write Byte	The Value field is written into the pointed byte.

67.7 Software Restrictions

67.7.1 Initialization Active

The driver cannot set INITA bit in System Control register when any of the command line or data lines is active, so the driver must ensure both CDIHB and CIHB bits are cleared.

67.7.2 Software Polling Procedure

For polling read or write, once the software begins a buffer read or write, it must access exactly the number of times as the values set in the Watermark Level Register; moreover, if the block size is not a multiple of the value in Watermark Level Register (read and write respectively), the software must access exactly the remaining number of words at the end of each block.

For example, for a read operation, if the RD_WML is 4, indicating the watermark level is 16 bytes, block size is 40 bytes, and the block number is 2, then the access times for the burst sequence in the whole transfer process must be 4, 4, 2, 4, 4, 2.

67.7.3 Suspend Operation

In order to suspend the data transfer, the software must inform uSDHC that the suspend command is successfully accepted. To achieve this, after the Suspend command is accepted by the SDIO card, software must send another normal command marked as suspend command (CMDTYP bits set as '01') to inform uSDHC that the transfer is suspended.

If software needs to resume the suspended trasnfer, it should read the value in BLKCNT register to save the remaining number of blocks before sending the normal command marked as suspend, otherwise on sending such 'suspend' command, uSDHC will regard the current transfer is aborted and change BLKCNT register to its original value, instead of keeping the remained number of blocks.

67.7.4 Data Length Setting

For either ADMA (ADMA1 or ADMA2) transfer, the data in the data buffer must be word aligned, so the data length set in the descriptor must be a multiple of 4.

67.7.5 (A)DMA Address Setting

To configure ADMA1/ADMA2/DMA address register, when TC bit is set, the register will always update itself with the internal address value to support dynamic address synchronization, so the software must ensure that the TC bit is cleared prior to configuring ADMA1/ADMA2/DMA address register.

67.7.6 Data Port Access

Data Port does not support parallel access. For example, during an external DMA access, it is not allowed to write any data to the Data Port by CPU; or during a CPU read operation, it is also prohibited to write any data to the Data Port, by either CPU or external DMA. Otherwise the data would be corrupted inside the uSDHC buffer.

67.7.7 Change Clock Frequency

uSDHC does not automatically gate off the card clock when the Host Driver changes the clock frequency. To prevent possible glitch on the card clock, clear the FRC_SDCLK_ON bit when changing clock divisor value(SDCLKFS or DVS in System Control Register) or setting RSTA bit.

Also before changing the clock divisor value, Host Driver should make sure the SDSTB bit is high.

67.7.8 Multi-block Read

For pre-defined multi-block read operation, i.e., the number of blocks to read has been defined by previous CMD23 for MMC, or pre-defined number of blocks in CMD53 for SDIO/SDCombo, or whatever multi-block read without abort command at card side, an abort command, either automatic or manual CMD12/CMD52, is still required by uSDHC after the pre-defined number of blocks are done, to drive the internal state machine to idle mode.

In this case, the card may not respond to this extra abort command and uSDHC will get Response Timeout. It is recommended to manually send an abort command with RSPTYP[1:0] both bits cleared.

67.8 uSDHC Memory Map/Register Definition

This section includes the module memory map and detailed descriptions of all registers.

See the table below for the register memory map for the uSDHC. All these registers only support 32-bit accesses.

NOTE

The uSDHC registers are 32-bit wide and only support 32-bit access.

uSDHC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
219_0000	DMA System Address (uSDHC1_DS_ADDR)	32	R/W	0000_0000h	67.8.1/5600
219_0004	Block Attributes (uSDHC1_BLK_ATT)	32	R/W	0000_0000h	67.8.2/5600
219_0008	Command Argument (uSDHC1_CMD_ARG)	32	R/W	0000_0000h	67.8.3/5602
219_000C	Command Transfer Type (uSDHC1_CMD_XFR_TYP)	32	R/W	0000_0000h	67.8.4/5602
219_0010	Command Response0 (uSDHC1_CMD_RSP0)	32	R	0000_0000h	67.8.5/5606
219_0014	Command Response1 (uSDHC1_CMD_RSP1)	32	R	0000_0000h	67.8.6/5606
219_0018	Command Response2 (uSDHC1_CMD_RSP2)	32	R	0000_0000h	67.8.7/5607
219_001C	Command Response3 (uSDHC1_CMD_RSP3)	32	R	0000_0000h	67.8.8/5607
219_0020	Data Buffer Access Port (uSDHC1_DATA_BUFF_ACC_PORT)	32	R/W	0000_0000h	67.8.9/5609
219_0024	Present State (uSDHC1_PRES_STATE)	32	R	0000_0000h	67.8.10/ 5609
219_0028	Protocol Control (uSDHC1_PROT_CTRL)	32	R/W	0880_0020h	67.8.11/ 5615
219_002C	System Control (uSDHC1_SYS_CTRL)	32	R/W	0080_800Fh	67.8.12/ 5620
219_0030	Interrupt Status (uSDHC1_INT_STATUS)	32	w1c	0000_0000h	67.8.13/ 5623
219_0034	Interrupt Status Enable (uSDHC1_INT_STATUS_EN)	32	R/W	0000_0000h	67.8.14/ 5629
219_0038	Interrupt Signal Enable (uSDHC1_INT_SIGNAL_EN)	32	R/W	0000_0000h	67.8.15/ 5632
219_003C	Auto CMD12 Error Status (uSDHC1_AUTOCMD12_ERR_STATUS)	32	R	0000_0000h	67.8.16/ 5634
219_0040	Host Controller Capabilities (uSDHC1_HOST_CTRL_CAP)	32	R	07F3_0000h	67.8.17/ 5638
219_0044	Watermark Level (uSDHC1_WTMK_LVL)	32	R/W	0810_0810h	67.8.18/ 5640
219_0048	Mixer Control (uSDHC1_MIX_CTRL)	32	R/W	8000_0000h	67.8.19/ 5641

Table continues on the next page...

uSDHC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
219_0050	Force Event (uSDHC1_FORCE_EVENT)	32	W (always reads 0)	0000_0000h	67.8.20/ 5643
219_0054	ADMA Error Status Register (uSDHC1_ADMA_ERR_STATUS)	32	R	0000_0000h	67.8.21/ 5646
219_0058	ADMA System Address (uSDHC1_ADMA_SYS_ADDR)	32	R/W	0000_0000h	67.8.22/ 5648
219_0060	DLL (Delay Line) Control (uSDHC1_DLL_CTRL)	32	R/W	0000_0200h	67.8.23/ 5649
219_0064	DLL Status (uSDHC1_DLL_STATUS)	32	R	0000_0000h	67.8.24/ 5651
219_0068	CLK Tuning Control and Status (uSDHC1_CLK_TUNE_CTRL_STATUS)	32	R/W	0000_0000h	67.8.25/ 5652
219_00C0	Vendor Specific Register (uSDHC1_VEND_SPEC)	32	R/W	2000_7809h	67.8.26/ 5654
219_00C4	MMC Boot Register (uSDHC1_MMC_BOOT)	32	R/W	0000_0000h	67.8.27/ 5657
219_00C8	Vendor Specific 2 Register (uSDHC1_VEND_SPEC2)	32	R/W	0000_0006h	67.8.28/ 5658
219_4000	DMA System Address (uSDHC2_DS_ADDR)	32	R/W	0000_0000h	67.8.1/5600
219_4004	Block Attributes (uSDHC2_BLK_ATT)	32	R/W	0000_0000h	67.8.2/5600
219_4008	Command Argument (uSDHC2_CMD_ARG)	32	R/W	0000_0000h	67.8.3/5602
219_400C	Command Transfer Type (uSDHC2_CMD_XFR_TYP)	32	R/W	0000_0000h	67.8.4/5602
219_4010	Command Response0 (uSDHC2_CMD_RSP0)	32	R	0000_0000h	67.8.5/5606
219_4014	Command Response1 (uSDHC2_CMD_RSP1)	32	R	0000_0000h	67.8.6/5606
219_4018	Command Response2 (uSDHC2_CMD_RSP2)	32	R	0000_0000h	67.8.7/5607
219_401C	Command Response3 (uSDHC2_CMD_RSP3)	32	R	0000_0000h	67.8.8/5607
219_4020	Data Buffer Access Port (uSDHC2_DATA_BUFF_ACC_PORT)	32	R/W	0000_0000h	67.8.9/5609
219_4024	Present State (uSDHC2_PRES_STATE)	32	R	0000_0000h	67.8.10/ 5609
219_4028	Protocol Control (uSDHC2_PROT_CTRL)	32	R/W	0880_0020h	67.8.11/ 5615
219_402C	System Control (uSDHC2_SYS_CTRL)	32	R/W	0080_800Fh	67.8.12/ 5620
219_4030	Interrupt Status (uSDHC2_INT_STATUS)	32	w1c	0000_0000h	67.8.13/ 5623
219_4034	Interrupt Status Enable (uSDHC2_INT_STATUS_EN)	32	R/W	0000_0000h	67.8.14/ 5629
219_4038	Interrupt Signal Enable (uSDHC2_INT_SIGNAL_EN)	32	R/W	0000_0000h	67.8.15/ 5632
219_403C	Auto CMD12 Error Status (uSDHC2_AUTOCMD12_ERR_STATUS)	32	R	0000_0000h	67.8.16/ 5634

Table continues on the next page...

uSDHC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
219_4040	Host Controller Capabilities (uSDHC2_HOST_CTRL_CAP)	32	R	07F3_0000h	67.8.17/ 5638
219_4044	Watermark Level (uSDHC2_WTMK_LVL)	32	R/W	0810_0810h	67.8.18/ 5640
219_4048	Mixer Control (uSDHC2_MIX_CTRL)	32	R/W	8000_0000h	67.8.19/ 5641
219_4050	Force Event (uSDHC2_FORCE_EVENT)	32	W (always reads 0)	0000_0000h	67.8.20/ 5643
219_4054	ADMA Error Status Register (uSDHC2_ADMA_ERR_STATUS)	32	R	0000_0000h	67.8.21/ 5646
219_4058	ADMA System Address (uSDHC2_ADMA_SYS_ADDR)	32	R/W	0000_0000h	67.8.22/ 5648
219_4060	DLL (Delay Line) Control (uSDHC2_DLL_CTRL)	32	R/W	0000_0200h	67.8.23/ 5649
219_4064	DLL Status (uSDHC2_DLL_STATUS)	32	R	0000_0000h	67.8.24/ 5651
219_4068	CLK Tuning Control and Status (uSDHC2_CLK_TUNE_CTRL_STATUS)	32	R/W	0000_0000h	67.8.25/ 5652
219_40C0	Vendor Specific Register (uSDHC2_VEND_SPEC)	32	R/W	2000_7809h	67.8.26/ 5654
219_40C4	MMC Boot Register (uSDHC2_MMC_BOOT)	32	R/W	0000_0000h	67.8.27/ 5657
219_40C8	Vendor Specific 2 Register (uSDHC2_VEND_SPEC2)	32	R/W	0000_0006h	67.8.28/ 5658
219_8000	DMA System Address (uSDHC3_DS_ADDR)	32	R/W	0000_0000h	67.8.1/5600
219_8004	Block Attributes (uSDHC3_BLK_ATT)	32	R/W	0000_0000h	67.8.2/5600
219_8008	Command Argument (uSDHC3_CMD_ARG)	32	R/W	0000_0000h	67.8.3/5602
219_800C	Command Transfer Type (uSDHC3_CMD_XFR_TYP)	32	R/W	0000_0000h	67.8.4/5602
219_8010	Command Response0 (uSDHC3_CMD_RSP0)	32	R	0000_0000h	67.8.5/5606
219_8014	Command Response1 (uSDHC3_CMD_RSP1)	32	R	0000_0000h	67.8.6/5606
219_8018	Command Response2 (uSDHC3_CMD_RSP2)	32	R	0000_0000h	67.8.7/5607
219_801C	Command Response3 (uSDHC3_CMD_RSP3)	32	R	0000_0000h	67.8.8/5607
219_8020	Data Buffer Access Port (uSDHC3_DATA_BUFF_ACC_PORT)	32	R/W	0000_0000h	67.8.9/5609
219_8024	Present State (uSDHC3_PRES_STATE)	32	R	0000_0000h	67.8.10/ 5609
219_8028	Protocol Control (uSDHC3_PROT_CTRL)	32	R/W	0880_0020h	67.8.11/ 5615
219_802C	System Control (uSDHC3_SYS_CTRL)	32	R/W	0080_800Fh	67.8.12/ 5620
219_8030	Interrupt Status (uSDHC3_INT_STATUS)	32	w1c	0000_0000h	67.8.13/ 5623

Table continues on the next page...

uSDHC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
219_8034	Interrupt Status Enable (uSDHC3_INT_STATUS_EN)	32	R/W	0000_0000h	67.8.14/ 5629
219_8038	Interrupt Signal Enable (uSDHC3_INT_SIGNAL_EN)	32	R/W	0000_0000h	67.8.15/ 5632
219_803C	Auto CMD12 Error Status (uSDHC3_AUTOCMD12_ERR_STATUS)	32	R	0000_0000h	67.8.16/ 5634
219_8040	Host Controller Capabilities (uSDHC3_HOST_CTRL_CAP)	32	R	07F3_0000h	67.8.17/ 5638
219_8044	Watermark Level (uSDHC3_WTMK_LVL)	32	R/W	0810_0810h	67.8.18/ 5640
219_8048	Mixer Control (uSDHC3_MIX_CTRL)	32	R/W	8000_0000h	67.8.19/ 5641
219_8050	Force Event (uSDHC3_FORCE_EVENT)	32	W (always reads 0)	0000_0000h	67.8.20/ 5643
219_8054	ADMA Error Status Register (uSDHC3_ADMA_ERR_STATUS)	32	R	0000_0000h	67.8.21/ 5646
219_8058	ADMA System Address (uSDHC3_ADMA_SYS_ADDR)	32	R/W	0000_0000h	67.8.22/ 5648
219_8060	DLL (Delay Line) Control (uSDHC3_DLL_CTRL)	32	R/W	0000_0200h	67.8.23/ 5649
219_8064	DLL Status (uSDHC3_DLL_STATUS)	32	R	0000_0000h	67.8.24/ 5651
219_8068	CLK Tuning Control and Status (uSDHC3_CLK_TUNE_CTRL_STATUS)	32	R/W	0000_0000h	67.8.25/ 5652
219_80C0	Vendor Specific Register (uSDHC3_VEND_SPEC)	32	R/W	2000_7809h	67.8.26/ 5654
219_80C4	MMC Boot Register (uSDHC3_MMC_BOOT)	32	R/W	0000_0000h	67.8.27/ 5657
219_80C8	Vendor Specific 2 Register (uSDHC3_VEND_SPEC2)	32	R/W	0000_0006h	67.8.28/ 5658
219_C000	DMA System Address (uSDHC4_DS_ADDR)	32	R/W	0000_0000h	67.8.1/5600
219_C004	Block Attributes (uSDHC4_BLK_ATT)	32	R/W	0000_0000h	67.8.2/5600
219_C008	Command Argument (uSDHC4_CMD_ARG)	32	R/W	0000_0000h	67.8.3/5602
219_C00C	Command Transfer Type (uSDHC4_CMD_XFR_TYP)	32	R/W	0000_0000h	67.8.4/5602
219_C010	Command Response0 (uSDHC4_CMD_RSP0)	32	R	0000_0000h	67.8.5/5606
219_C014	Command Response1 (uSDHC4_CMD_RSP1)	32	R	0000_0000h	67.8.6/5606
219_C018	Command Response2 (uSDHC4_CMD_RSP2)	32	R	0000_0000h	67.8.7/5607
219_C01C	Command Response3 (uSDHC4_CMD_RSP3)	32	R	0000_0000h	67.8.8/5607
219_C020	Data Buffer Access Port (uSDHC4_DATA_BUFF_ACC_PORT)	32	R/W	0000_0000h	67.8.9/5609
219_C024	Present State (uSDHC4_PRES_STATE)	32	R	0000_0000h	67.8.10/ 5609

Table continues on the next page...

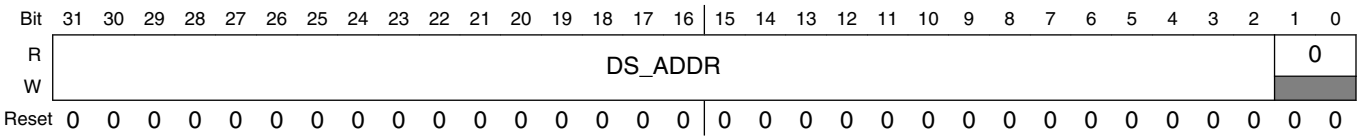
uSDHC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
219_C028	Protocol Control (uSDHC4_PROT_CTRL)	32	R/W	0880_0020h	67.8.11/ 5615
219_C02C	System Control (uSDHC4_SYS_CTRL)	32	R/W	0080_800Fh	67.8.12/ 5620
219_C030	Interrupt Status (uSDHC4_INT_STATUS)	32	w1c	0000_0000h	67.8.13/ 5623
219_C034	Interrupt Status Enable (uSDHC4_INT_STATUS_EN)	32	R/W	0000_0000h	67.8.14/ 5629
219_C038	Interrupt Signal Enable (uSDHC4_INT_SIGNAL_EN)	32	R/W	0000_0000h	67.8.15/ 5632
219_C03C	Auto CMD12 Error Status (uSDHC4_AUTOCMD12_ERR_STATUS)	32	R	0000_0000h	67.8.16/ 5634
219_C040	Host Controller Capabilities (uSDHC4_HOST_CTRL_CAP)	32	R	07F3_0000h	67.8.17/ 5638
219_C044	Watermark Level (uSDHC4_WTMK_LVL)	32	R/W	0810_0810h	67.8.18/ 5640
219_C048	Mixer Control (uSDHC4_MIX_CTRL)	32	R/W	8000_0000h	67.8.19/ 5641
219_C050	Force Event (uSDHC4_FORCE_EVENT)	32	W (always reads 0)	0000_0000h	67.8.20/ 5643
219_C054	ADMA Error Status Register (uSDHC4_ADMA_ERR_STATUS)	32	R	0000_0000h	67.8.21/ 5646
219_C058	ADMA System Address (uSDHC4_ADMA_SYS_ADDR)	32	R/W	0000_0000h	67.8.22/ 5648
219_C060	DLL (Delay Line) Control (uSDHC4_DLL_CTRL)	32	R/W	0000_0200h	67.8.23/ 5649
219_C064	DLL Status (uSDHC4_DLL_STATUS)	32	R	0000_0000h	67.8.24/ 5651
219_C068	CLK Tuning Control and Status (uSDHC4_CLK_TUNE_CTRL_STATUS)	32	R/W	0000_0000h	67.8.25/ 5652
219_C0C0	Vendor Specific Register (uSDHC4_VEND_SPEC)	32	R/W	2000_7809h	67.8.26/ 5654
219_C0C4	MMC Boot Register (uSDHC4_MMC_BOOT)	32	R/W	0000_0000h	67.8.27/ 5657
219_C0C8	Vendor Specific 2 Register (uSDHC4_VEND_SPEC2)	32	R/W	0000_0006h	67.8.28/ 5658

67.8.1 DMA System Address (uSDHCx_DS_ADDR)

This register contains the physical system memory address used for DMA transfers.

Address: Base address + 0h offset



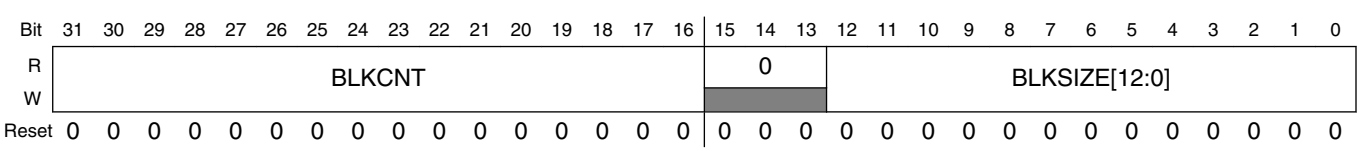
uSDHCx_DS_ADDR field descriptions

Field	Description
31–2 DS_ADDR	DMA System Address: This register contains the 32-bit system memory address for a DMA transfer. Since the address must be word (4 bytes) aligned, the least 2 bits are reserved, always 0. When the uSDHC stops a DMA transfer, this register points to the system address of the next contiguous data position. It can be accessed only when no transaction is executing (i.e. after a transaction has stopped). Read operation during transfers may return an invalid value. The Host Driver shall initialize this register before starting a DMA transaction. After DMA has stopped, the system address of the next contiguous data position can be read from this register. This register is protected during a data transfer. When data lines are active, write to this register is ignored. The Host driver shall wait, until the DLA bit in the Present State register is cleared, before writing to this register. The uSDHC internal DMA does not support a virtual memory system. It only supports continuous physical memory access. And due to AHB burst limitations, if the burst must cross the 1 KB boundary, uSDHC will automatically change SEQ burst type to NSEQ. Since this register supports dynamic address reflecting, when TC bit is set, it automatically alters the value of internal address counter, so SW cannot change this register when TC bit is set. Such restriction is also listed in Software Restrictions .
Reserved	This read-only field is reserved and always has the value 0.

67.8.2 Block Attributes (uSDHCx_BLK_ATT)

This register is used to configure the number of data blocks and the number of bytes in each block.

Address: Base address + 4h offset



uSDHCx_BLK_ATT field descriptions

Field	Description
31–16 BLKCNT	Blocks Count For Current Transfer: This register is enabled when the Block Count Enable bit in the Transfer Mode register is set to 1 and is valid only for multiple block transfers. For single block transfer, this register will always read as 1. The Host Driver shall set this register to a value between 1 and the maximum block count. The uSDHC decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to 0 results in no data blocks being transferred. This register should be accessed only when no transaction is executing (i.e. after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored. When saving transfer content as a result of a Suspend command, the number of blocks yet to be transferred can be determined by reading this register. The reading of this register should be applied after transfer is paused by stop at block gap operation and before sending the command marked as suspend. This is because when Suspend command is sent out, uSDHC will regard the current transfer is aborted and change BLKCNT register back to its original value instead of keeping the dynamical indicator of remained block count. When restoring transfer content prior to issuing a Resume command, the Host Driver shall restore the previously saved block count. **NOTE:** Although the BLKCNT field is 0 after reset, the read of reset value is 0x1. This is because when MSBSEL bit is indicating a single block transfer, the read value of BLKCNT is always 1. FFFF 65535 blocks 0002 2 blocks 0001 1 block 0000 Stop Count
15–13 Reserved	This read-only field is reserved and always has the value 0.
BLKSIZE[12:0]	Transfer Block Size: This register specifies the block size for block data transfers. Values ranging from 1 byte up to the maximum buffer size can be set. It can be accessed only when no transaction is executing (i.e. after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations will be ignored. 1000 4096 Bytes 800 2048 Bytes 200 512 Bytes 1FF 511 Bytes 004 4 Bytes 003 3 Bytes 002 2 Bytes 001 1 Byte 000 No data transfer

67.8.3 Command Argument (uSDHCx_CMD_ARG)

This register contains the SD / MMC Command Argument.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMDARG[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

uSDHCx_CMD_ARG field descriptions

Field	Description
CMDARG[31:0]	Command Argument The SD / MMC Command Argument is specified as bits 39-8 of the Command Format in the SD or MMC Specification.This register is write protected when the Command Inhibit (CMD) bit in the Present State register is set.

67.8.4 Command Transfer Type (uSDHCx_CMD_XFR_TYP)

This register is used to control the operation of data transfers. The Host Driver shall set this register before issuing a command followed by a data transfer, or before issuing a Resume command. To prevent data loss, the uSDHC prevents writing to the bits, that are involved in the data transfer of this register, when data transfer is active. These bits are DPSEL, MBSEL, DTDSEL, AC12EN, BCEN and DMAEN.

The Host Driver shall check the Command Inhibit DAT bit (CDIHB) and the Command Inhibit CMD bit (CIHB) in the Present State register before writing to this register. When the CDIHB bit in the Present State register is set, any attempt to send a command with data by writing to this register is ignored; when the CIHB bit is set, any write to this register is ignored.

On sending commands with data transfer invovled, it is mandatory that the block size is non-zero. Block count must also be non-zero, or indicated as single block transfer (bit 5 of this register is '0' when written), or block count is disabled (bit 1 of this register is '0' when written), otherwise uSDHC will ignore the sending of this command and do nothing. For write command, with all above restrictions, it is also mandatory that the write protect switch is not active (WPSPL bit of Present State Register is '1'), otherwise uSDHC will also ignore the command.

If the commands with data transfer does not receive the response in 64 clock cycles, i.e., response time-out, uSDHC will regard the external device does not accept the command and abort the data transfer. In this scenario, the driver should issue the command again to re-try the transfer. It is also possible that for some reason the card responds the command but uSDHC does not receive the response, and if it is internal DMA (either simple DMA or ADMA) read operation, the external system memory is over-written by the internal DMA with data sent back from the card.

The table below shows the summary of how register settings determine the type of data transfer.

Table 67-8. Transfer Type Register Setting for Various Transfer Types

Multi/Single Block Select	Block Count Enable	Block Count	Function
0	Don't Care	Don't Care	Single Transfer
1	0	Don't Care	Infinite Transfer
1	1	Positive Number	Multiple Transfer
1	1	Zero	No Data Transfer

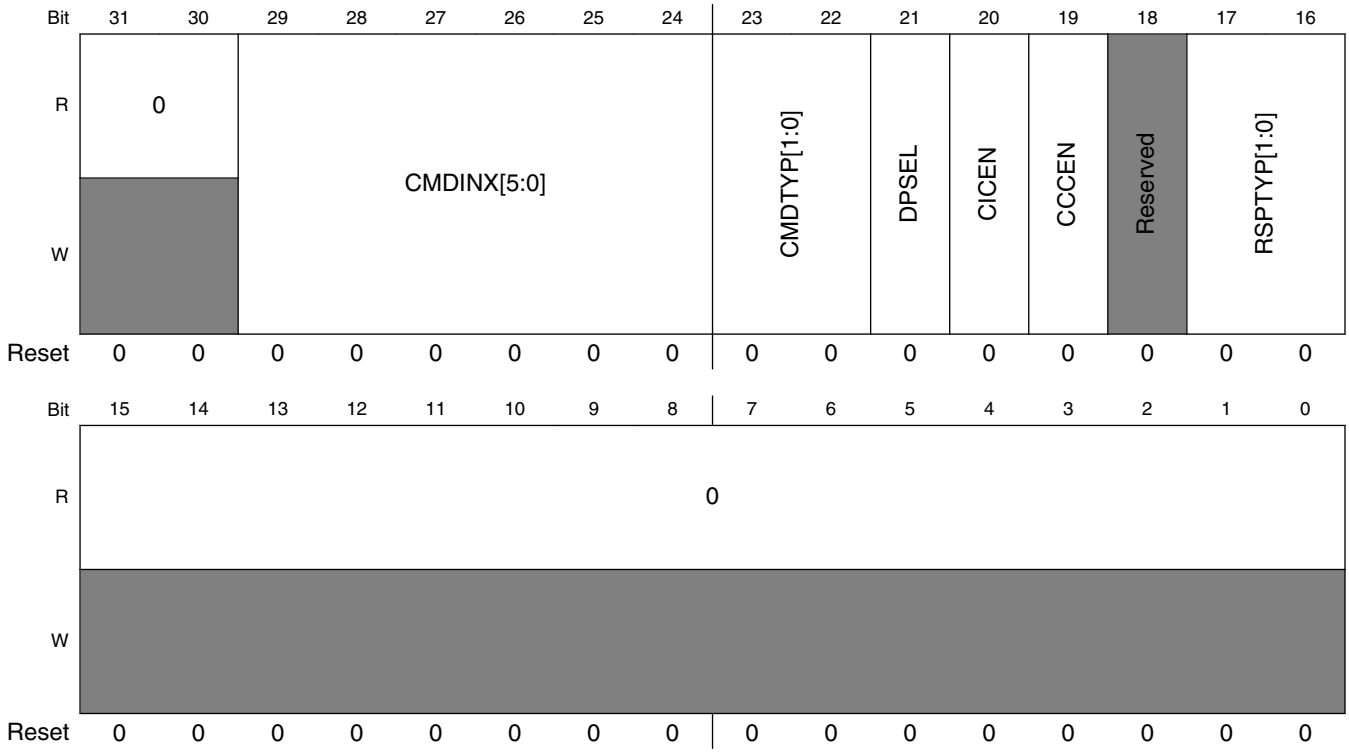
The table below shows the relationship between the Command Index Check Enable and the Command CRC Check Enable, in regards to the Response Type bits as well as the name of the response type.

Table 67-9. Relationship Between Parameters and the Name of the Response Type

Response Type	Index Check Enable	CRC Check Enable	Name of Response Type
00	0	0	No Response
01	0	1	R2
10	0	0	R3,R4
10	1	1	R1,R5,R6
11	1	1	R1b,R5b

- In the SDIO specification, response type notation for R5b is not defined. R5 includes R5b in the SDIO specification. But R5b is defined in this specification to specify that the uSDHC will check the busy status after receiving a response. For example, usually CMD52 is used with R5, but the I/O abort command shall be used with R5b.
- The CRC field for R3 and R4 is expected to be all 1 bits. The CRC check shall be disabled for these response types.

Address: Base address + Ch offset



uSDHCx_CMD_XFR_TYP field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value 0.
29–24 CMDINX[5:0]	Command Index These bits shall be set to the command number that is specified in bits 45-40 of the Command-Format in the SD Memory Card Physical Layer Specification and SDIO Card Specification.
23–22 CMDTYP[1:0]	Command Type There are three types of special commands: Suspend, Resume and Abort. These bits shall be set to 00b for all other commands. • Suspend Command: If the Suspend command succeeds, the uSDHC shall assume that the card bus has been released and that it is possible to issue the next command which uses the DATA line. Since the uSDHC does not monitor the content of command response, it does not know if the Suspend command succeeded or not. It is the Host Driver's responsibility to check the status of the Suspend command and send another command marked as Suspend to inform the uSDHC that a Suspend command was successfully issued. Refer to Suspend Resume for more details. After the end bit of command is sent, the uSDHC de-asserts Read Wait for read transactions and stops checking busy for write transactions. In 4-bit mode, the interrupt cycle starts. If the Suspend command fails, the uSDHC will maintain its current state, and the Host Driver shall restart the transfer by setting the Continue Request bit in the Protocol Control register. • Resume Command: The Host Driver re-starts the data transfer by restoring the registers saved before sending the Suspend Command and then sends the Resume Command. The uSDHC will check for a pending busy state before starting write transfers. • Abort Command: If this command is set when executing a read transfer, the uSDHC will stop reads to the buffer. If this command is set when executing a write transfer, the uSDHC will stop driving the DATA line. After issuing the Abort command, the Host Driver should issue a software reset (Abort Transaction).

Table continues on the next page...

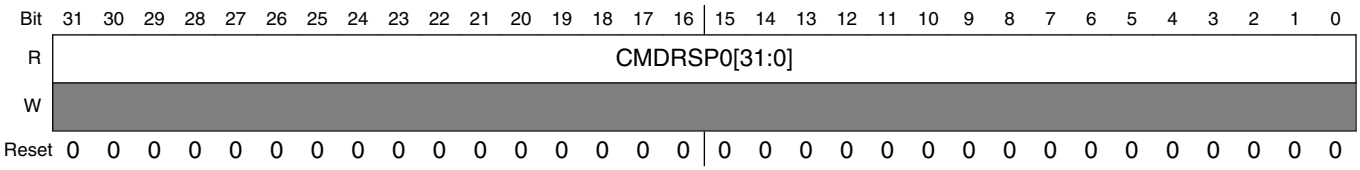
uSDHCx_CMD_XFR_TYP field descriptions (continued)

Field	Description
	11 Abort CMD12, CMD52 for writing I/O Abort in CCCR 10 Resume CMD52 for writing Function Select in CCCR 01 Suspend CMD52 for writing Bus Suspend in CCCR 00 Normal Other commands
21 DPSEL	Data Present Select This bit is set to 1 to indicate that data is present and shall be transferred using the DATA line. It is set to 0 for the following: • Commands using only the CMD line (e.g. CMD52). • Commands with no data transfer, but using the busy signal on DATA0 line (R1b or R5b e.g. CMD38) **NOTE:** In resume command, this bit shall be set, and other bits in this register shall be set the same as when the transfer was initially launched. When the Write Protect switch is on, (i.e. the WPSPL bit is active as '0'), any command with a write operation will be ignored. That is to say, when this bit is set, while the DTDSEL bit is 0, writes to the register Transfer Type are ignored. 1 Data Present 0 No Data Present
20 CICEN	Command Index Check Enable If this bit is set to 1, the uSDHC will check the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this bit is set to 0, the Index field is not checked. 1 Enable 0 Disable
19 CCCEN	Command CRC Check Enable If this bit is set to 1, the uSDHC shall check the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this bit is set to 0, the CRC field is not checked. The number of bits checked by the CRC field value changes according to the length of the response. (Refer to RSPTYP[1:0] and Command Transfer Type (uSDHC_CMD_XFR_TYP) .) 1 Enable 0 Disable
18 -	This field is reserved. Reserved
17–16 RSPTYP[1:0]	Response Type Select 00 No Response 01 Response Length 136 10 Response Length 48 11 Response Length 48, check Busy after response
Reserved	This read-only field is reserved and always has the value 0.

67.8.5 Command Response0 (uSDHCx_CMD_RSP0)

This register is used to store part 0 of the response bits from the card.

Address: Base address + 10h offset



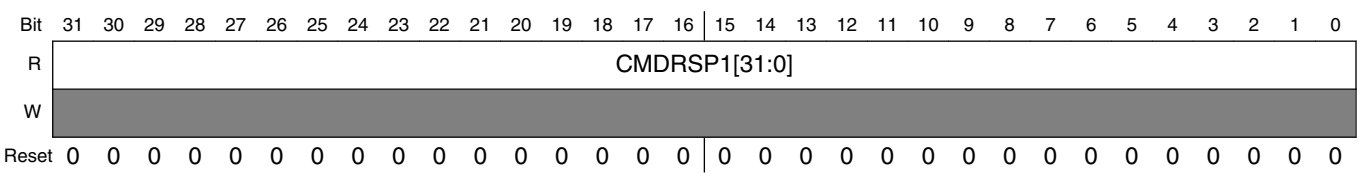
uSDHCx_CMD_RSP0 field descriptions

Field	Description
CMDRSP0[31:0]	Command Response 0 Refer to Command Response3 (uSDHC_CMD_RSP3) for the mapping of command responses from the SD Bus to this register for each response type.

67.8.6 Command Response1 (uSDHCx_CMD_RSP1)

This register is used to store part 1 of the response bits from the card.

Address: Base address + 14h offset



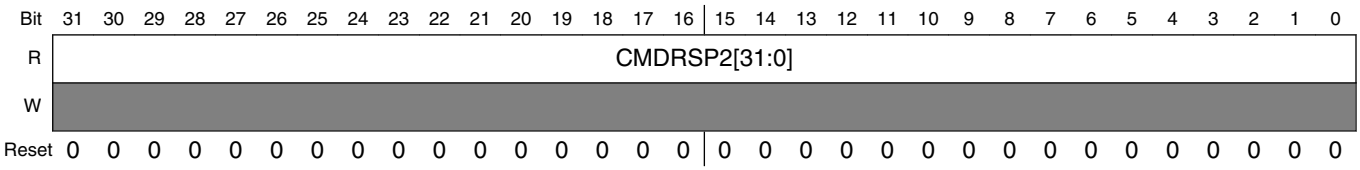
uSDHCx_CMD_RSP1 field descriptions

Field	Description
CMDRSP1[31:0]	Command Response 1 Refer to Command Response3 (uSDHC_CMD_RSP3) for the mapping of command responses from the SD Bus to this register for each response type.

67.8.7 Command Response2 (uSDHCx_CMD_RSP2)

This register is used to store part 2 of the response bits from the card.

Address: Base address + 18h offset



uSDHCx_CMD_RSP2 field descriptions

Field	Description
CMDRSP2[31:0]	Command Response 2 Refer to Command Response3 (uSDHC_CMD_RSP3) for the mapping of command responses from the SD Bus to this register for each response type.

67.8.8 Command Response3 (uSDHCx_CMD_RSP3)

This register is used to store part 3 of the response bits from the card.

The table below describes the mapping of command responses from the SD Bus to Command Response registers for each response type. In the table, R[ ] refers to a bit range within the response data as transmitted on the SD Bus.

Table 67-10. Response Bit Definition for Each Response Type

Response Type	Meaning of Response	Response Field	Response Register
R1,R1b (normal response)	Card Status	R[39:8]	CMDRSP0
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R[39:8]	CMDRSP3
R2 (CID, CSD register)	CID/CSD register [127:8]	R[127:8]	{CMDRSP3[23:0], CMDRSP2, CMDRSP1, CMDRSP0}
R3 (OCR register)	OCR register for memory	R[39:8]	CMDRSP0
R4 (OCR register)	OCR register for I/O etc.	R[39:8]	CMDRSP0
R5, R5b	SDIO response	R[39:8]	CMDRSP0
R6 (Publish RCA)	New Published RCA[31:16] and card status[15:0]	R[39:9]	CMDRSP0

This table shows that most responses with a length of 48 (R[47:0]) have 32-bits of the response data (R[39:8]) stored in the CMDRSP0 register. Responses of type R1b (Auto CMD12 responses) have response data bits (R[39:8]) stored in the CMDRSP3 register. Responses with length 136 (R[135:0]) have 120-bits of the response data (R[127:8]) stored in the CMDRSP0, 1, 2, and 3 registers.

To be able to read the response status efficiently, the uSDHC only stores part of the response data in the Command Response registers. This enables the Host Driver to efficiently read 32-bits of response data in one read cycle on a 32-bit bus system. Parts of the response, the Index field and the CRC, are checked by the uSDHC (as specified by the Command Index Check Enable and the Command CRC Check Enable bits in the Transfer Type register) and generate an error interrupt if any error is detected. The bit range for the CRC check depends on the response length. If the response length is 48, the uSDHC will check R[47:1], and if the response length is 136 the uSDHC will check R[119:1].

Since the uSDHC may have a multiple block data transfer executing concurrently with a CMD_wo_DAT command, the uSDHC stores the Auto CMD12 response in the CMDRSP3 register. The CMD_wo_DAT response is stored in CMDRSP0. This allows the uSDHC to avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa. When the uSDHC modifies part of the Command Response registers, as shown in the table above, it preserves the unmodified bits.

Address: Base address + 1Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP3[31:0]																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

uSDHCx_CMD_RSP3 field descriptions

Field	Description
CMDRSP3[31:0]	Command Response 3 Refer to Command Response3 (uSDHC_CMD_RSP3) for the mapping of command responses from the SD Bus to this register for each response type.

67.8.9 Data Buffer Access Port (uSDHCx_DATA_BUFF_ACC_PORT)

This is a 32-bit data port register used to access the internal buffer.

Address: Base address + 20h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATCONT[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

uSDHCx_DATA_BUFF_ACC_PORT field descriptions

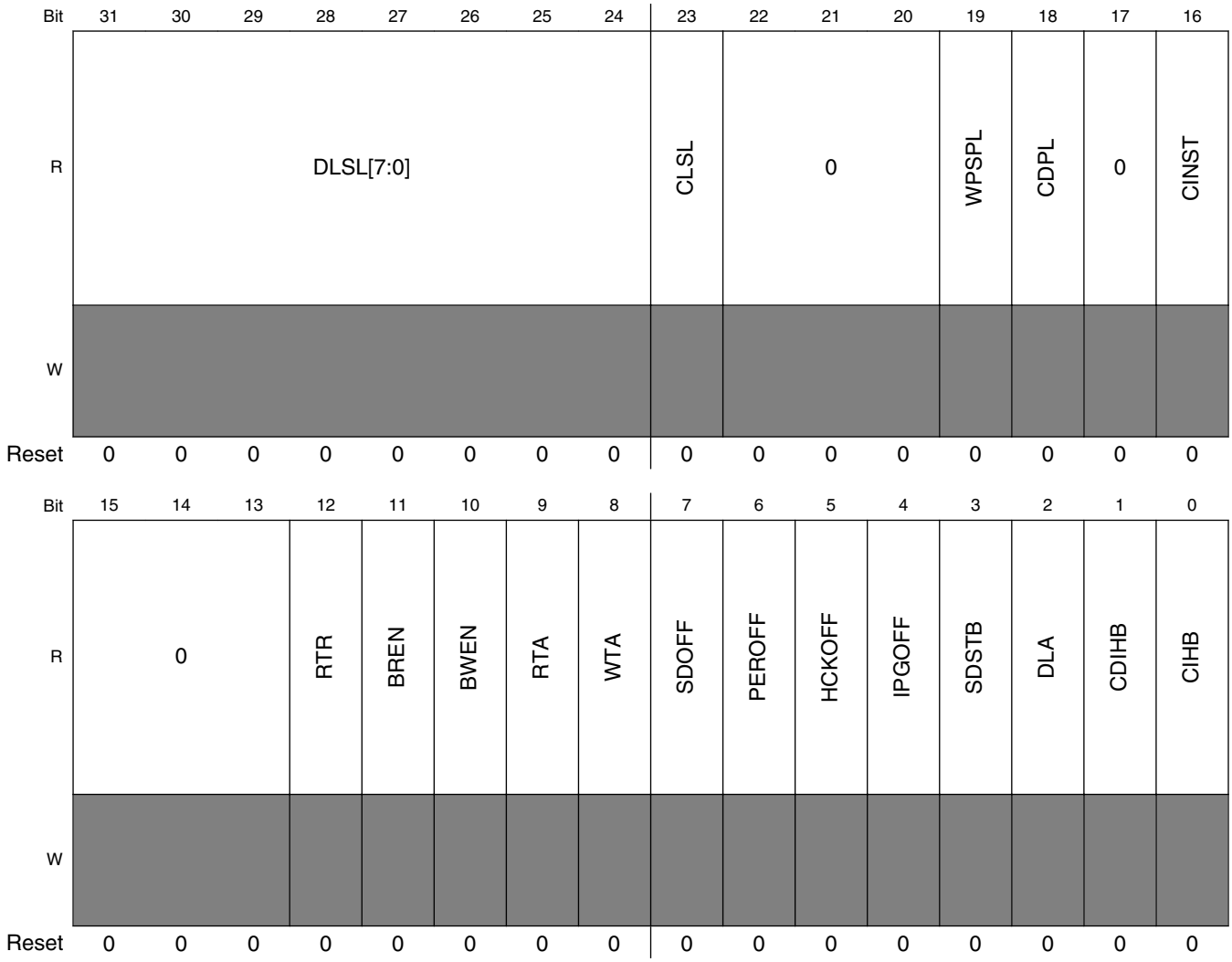
Field	Description
DATCONT[31:0]	Data Content The Buffer Data Port register is for 32-bit data access by the Arm platform or the external DMA. When the internal DMA is enabled, any write to this register is ignored, and any read from this register will always yield 0s.

67.8.10 Present State (uSDHCx_PRES_STATE)

The Host Driver can get status of the uSDHC from this 32-bit read only register.

- The Host Driver can issue CMD0, CMD12, CMD13 (for memory) and CMD52 (for SDIO) when the DATA lines are busy during a data transfer. These commands can be issued when Command Inhibit (CMD) is set to zero. Other commands shall be issued when Command Inhibit (DATA) is set to zero. Possible changes to the SD Physical Specification may add other commands to this list in the future.
- Note: the reset value of Present State Register depend on testbench connectivity.

Address: Base address + 24h offset



uSDHCx_PRES_STATE field descriptions

Field	Description
31–24 DLSL[7:0]	DATA[7:0] Line Signal Level This status is used to check the DATA line level to recover from errors, and for debugging.This is especially useful in detecting the busy signal level from DATA0. The reset value is affected by the external pull-up / pull-down resistors. By default, the read value of this bit field after reset is 8'b11110111, when DATA3 is pulled down and the other lines are pulled up. DATA7 Data 7 line signal level DATA6 Data 6 line signal level DATA5 Data 5 line signal level DATA4 Data 4 line signal level DATA3 Data 3 line signal level DATA2 Data 2 line signal level DATA1 Data 1 line signal level DATA0 Data 0 line signal level

Table continues on the next page...

uSDHCx_PRES_STATE field descriptions (continued)

Field	Description
23 CLSL	CMD Line Signal Level This status is used to check the CMD line level to recover from errors, and for debugging. The reset value is affected by the external pull-up / pull-down resistor, by default, the read value of this bit after reset is 1'b1, when the command line is pulled up.
22–20 Reserved	This read-only field is reserved and always has the value 0.
19 WPSPL	Write Protect Switch Pin Level The Write Protect Switch is supported for memory and combo cards.This bit reflects the inverted value of the WP pin of the card socket. A software reset does not affect this bit. The reset value is effected by the external write protect switch. If the WP pin is not used, it should be tied low, so that the reset value of this bit is high and write is enabled. 1 Write enabled (WP = 0) 0 Write protected (WP = 1)
18 CDPL	Card Detect Pin Level This bit reflects the inverse value of the CD_B pin for the card socket. Debouncing is not performed on this bit. This bit may be valid, but is not guaranteed, because of propagation delay. Use of this bit is limited to testing since it must be debounced by software. A software reset does not effect this bit. A write to the Force Event Register does not effect this bit. The reset value is effected by the external card detection pin. This bit shows the value on the CD_B pin (i.e. when a card is inserted in the socket, it is 0 on the CD_B input, and consequently the CDPL reads 1.) 1 Card present (CD_B = 0) 0 No card present (CD_B = 1)
17 Reserved	This read-only field is reserved and always has the value 0.
16 CINST	Card Inserted This bit indicates whether a card has been inserted. The uSDHC debounces this signal so that the Host Driver will not need to wait for it to stabilize. Changing from a 0 to 1 generates a Card Insertion interrupt in the Interrupt Status register. Changing from a 1 to 0 generates a Card Removal interrupt in the Interrupt Status register. A write to the Force Event Register does not effect this bit. The Software Reset For All in the System Control register does not effect this bit. A software reset does not effect this bit. 1 Card Inserted 0 Power on Reset or No Card
15–13 Reserved	This read-only field is reserved and always has the value 0.
12 RTR	Re-Tuning Request (only for SD3.0 SDR104 mode) Host Controller may request Host Driver to execute re-tuning sequence by setting this bit when the data window is shifted by temperature drift and a tuned sampling point does not have a good margin to receive correct data. This bit is cleared when a command is issued with setting Execute Tuning bit in MIXER_CTRL register. Changing of this bit from 0 to 1 generates Re-Tuning Event. Refer to Interrupt status registers for more detail.

Table continues on the next page...

uSDHCx_PRES_STATE field descriptions (continued)

Field	Description
	This bit isn't set to 1 if Sampling Clock Select in the MIXER_CTRL register is set to 0 (using fixed sampling clock). 1 Sampling clock needs re-tuning 0 Fixed or well tuned sampling clock
11 BREN	Buffer Read Enable This status bit is used for non-DMA read transfers. The uSDHC implements an internal buffer to transfer data efficiently. This read only flag indicates that valid data exists in the host side buffer. If this bit is high, valid data greater than the watermark level exist in the buffer. A change of this bit from 1 to 0 occurs when some reads from the buffer(read DATPORT (Base + 0x20)) are made and the buffer hasn't valid data greater than the watermark level. A change of this bit from 0 to1 occurs when there is enough valid data ready in the buffer and the Buffer Read Ready interrupt has been generated and enabled. 1 Read enable 0 Read disable
10 BWEN	Buffer Write Enable This status bit is used for non-DMA write transfers. The uSDHC implements an internal buffer to transfer data efficiently. This read only flag indicates if space is available for write data. If this bit is 1, valid data greater than the watermark level can be written to the buffer. A change of this bit from 1 to 0 occurs when some writes to the buffer(write DATPORT(Base + 0x20)) are made and the buffer hasn't valid space greater than the watermark level. A change of this bit from 0 to 1 occurs when the buffer can hold valid data greater than the write watermark level and the Buffer Write Ready interrupt is generated and enabled. 1 Write enable 0 Write disable
9 RTA	Read Transfer Active This status bit is used for detecting completion of a read transfer. This bit is set for either of the following conditions: • After the end bit of the read command. • When writing a 1 to the Continue Request bit in the Protocol Control register to restart a read transfer. A Transfer Complete interrupt is generated when this bit changes to 0. This bit is cleared for either of the following conditions: • When the last data block as specified by block length is transferred to the System, i.e. all data are read away from uSDHC internal buffer. • When all valid data blocks have been transferred from uSDHC internal buffer to the System and no current block transfers are being sent as a result of the Stop At Block Gap Request being set to 1. 1 Transferring data 0 No valid data
8 WTA	Write Transfer Active This status bit indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the uSDHC. This bit is set in either of the following cases: • After the end bit of the write command. • When writing 1 to the Continue Request bit in the Protocol Control register to restart a write transfer.

Table continues on the next page...

uSDHCx_PRES_STATE field descriptions (continued)

Field	Description
	This bit is cleared in either of the following cases: • After getting the CRC status of the last data block as specified by the transfer count (Single and Multiple). • After getting the CRC status of any block where data transmission is about to be stopped by a Stop At Block Gap Request. During a write transaction, a Block Gap Event interrupt is generated when this bit is changed to 0, as result of the Stop At Block Gap Request being set. This status is useful for the Host Driver in determining when to issue commands during Write Busy state. 1 Transferring data 0 No valid data
7 SDOFF	SD Clock Gated Off Internally This status bit indicates that the SD Clock is internally gated off, because of buffer over / under-run or read pause without read wait assertion, or the driver set FRC_SDCLK_ON bit is 0 to stop the SD clock in idle status. Set IPG_PERCLK_SOFT_EN and CARD_CLK_SOFT_EN to 0 also gate off SD clock. This bit is for the Host Driver to debug data transaction on the SD bus. 1 SD Clock is gated off. 0 SD Clock is active.
6 PEROFF	IPG_PERCLK Gated Off Internally This status bit indicates that the IPG_PERCLK is internally gated off. This bit is for the Host Driver to debug transaction on the SD bus. When IPG_CLK_SOFT_EN is cleared, IPG_PERCLK will be gated off, otherwise IPG_PERCLK will be always active. 1 IPG_PERCLK is gated off. 0 IPG_PERCLK is active.
5 HCKOFF	HCLK Gated Off Internally This status bit indicates that the HCLK is internally gated off. This bit is for the Host Driver to debug during a data transfer. 1 HCLK is gated off. 0 HCLK is active.
4 IPGOFF	IPG_CLK Gated Off Internally This status bit indicates that the ipg_clk is internally gated off. This bit is for the Host Driver to debug. 1 IPG_CLK is gated off. 0 IPG_CLK is active.
3 SDSTB	SD Clock Stable This status bit indicates that the internal card clock is stable. This bit is for the Host Driver to poll clock status when changing the clock frequency. It is recommended to clear FRC_SDCLK_ON bit in System Control register to remove glitches on the card clock when the frequency is changing. *Before changing clock divisor value(SDCLKFS or DVS), Host Driver should make sure the SDSTB bit is high.* 1 Clock is stable. 0 Clock is changing frequency and not stable.

Table continues on the next page...

uSDHCx_PRES_STATE field descriptions (continued)

Field	Description
2 DLA	Data Line Active This status bit indicates whether one of the DATA lines on the SD Bus is in use. In the case of read transactions: This status indicates if a read transfer is executing on the SD Bus. Changes in this value from 1 to 0, between data blocks, generates a Block Gap Event interrupt in the Interrupt Status register. This bit will be set in either of the following cases: • After the end bit of the read command. • When writing a 1 to the Continue Request bit in the Protocol Control register to restart a read transfer. This bit will be cleared in either of the following cases: (1) When the end bit of the last data block is sent from the SD Bus to the uSDHC. (2) When the Read Wait state is stopped by a Suspend command and the DATA2 line is released. The uSDHC will wait at the next block gap by driving Read Wait at the start of the interrupt cycle. If the Read Wait signal is already driven (data buffer cannot receive data), the uSDHC can wait for a current block gap by continuing to drive the Read Wait signal. It is necessary to support Read Wait in order to use the suspend / resume function. This bit will remain 1 during Read Wait. In the case of write transactions: This status indicates that a write transfer is executing on the SD Bus. Changes in this value from 1 to 0 generate a Transfer Complete interrupt in the Interrupt Status register. This bit will be set in either of the following cases: • After the end bit of the write command. • When writing to 1 to the Continue Request bit in the Protocol Control register to continue a write transfer. This bit will be cleared in either of the following cases: • When the SD card releases Write Busy of the last data block, the uSDHC will also detect if the output is not busy. If the SD card does not drive the busy signal after the CRC status is received, the uSDHC shall assume the card drive "Not Busy". • When the SD card releases write busy, prior to waiting for write transfer, and as a result of a Stop At Block Gap Request. In the case of command with busy pending: This status indicates that a busy state follows the command and the data line is in use. This bit will be cleared when the DATA0 line is released. 1 DATA Line Active 0 DATA Line Inactive
1 CDIHB	Command Inhibit (DATA) This status bit is generated if either the DAT Line Active or the Read Transfer Active is set to 1. If this bit is 0, it indicates that the uSDHC can issue the next SD / MMC Command. Commands with a busy signal belong to Command Inhibit (DATA) (for example. R1b, R5b type). Changing from 1 to 0 generates a Transfer Complete interrupt in the Interrupt Status register. **NOTE:** The SD Host Driver can save registers for a suspend transaction after this bit has changed from 1 to 0.

Table continues on the next page...

uSDHCx_PRES_STATE field descriptions (continued)

Field	Description
	1 Cannot issue command which uses the DATA line 0 Can issue command which uses the DATA line
0 CIHB	Command Inhibit (CMD) If this status bit is 0, it indicates that the CMD line is not in use and the uSDHC can issue a SD / MMC Command using the CMD line. This bit is set also immediately after the Transfer Type register is written. This bit is cleared when the command response is received. Even if the Command Inhibit (DATA) is set to 1, Commands using only the CMD line can be issued if this bit is 0. Changing from 1 to 0 generates a Command Complete interrupt in the Interrupt Status register. If the uSDHC cannot issue the command because of a command conflict error (Refer to Command CRC Error) or because of a Command Not Issued By Auto CMD12 Error, this bit will remain 1 and the Command Complete is not set. The Status of issuing an Auto CMD12 does not show on this bit. 1 Cannot issue command 0 Can issue command using only CMD line

67.8.11 Protocol Control (uSDHCx_PROT_CTRL)

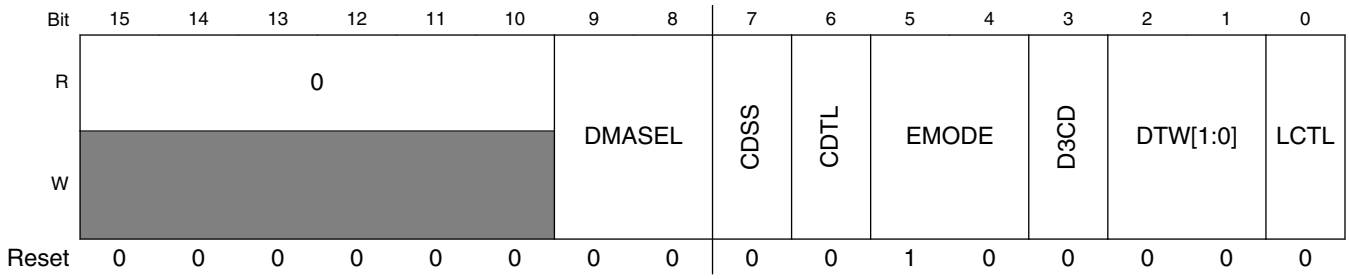
There are three cases to restart the transfer after stop at the block gap. Which case is appropriate depends on whether the uSDHC issues a Suspend command or the SD card accepts the Suspend command.

1. If the Host Driver does not issue a Suspend command, the Continue Request shall be used to restart the transfer.
2. If the Host Driver issues a Suspend command and the SD card accepts it, a Resume command shall be used to restart the transfer.
3. If the Host Driver issues a Suspend command and the SD card does not accept it, the Continue Request shall be used to restart the transfer.

Any time Stop At Block Gap Request stops the data transfer, the Host Driver shall wait for a Transfer Complete (in the Interrupt Status register), before attempting to restart the transfer. When restarting the data transfer by Continue Request, the Host Driver shall clear the Stop At Block Gap Request before or simultaneously.

Address: Base address + 28h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	-	NON_EXACT_BLK_RD	BURST_LEN_EN			WECRM	WECINS	WECINT	-			RD_DONE_NO_8CLK	IABG	RWCTL	CREQ	SABGREQ
Reset	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0



uSDHCx_PROT_CTRL field descriptions

Field	Description
31 -	Reserved. Always write as 0
30 NON_EXACT_BLK_RD	Current block read is non-exact block read. It is only used for SDIO. 1 The block read is non-exact block read. Host driver needs to issue abort command to terminate this multi-block read. 0 The block read is exact block read. Host driver doesn't need to issue abort command to terminate this multi-block read.
29–27 BURST_LEN_EN	BURST length enable for INCR, INCR4 / INCR8 / INCR16, INCR4-WRAP / INCR8-WRAP / INCR16-WRAP This is used to enable / disable the burst length for the external AHB2AXI bridge. It is useful especially for INCR transfer because without burst length indicator, the AHB2AXI bridge does not know the burst length in advance. Without burst length indicator, AHB INCR transfers can only be converted to SINGLEs on the AXI side. xx1 Burst length is enabled for INCR x1x Burst length is enabled for INCR4 / INCR8 / INCR16 1xx Burst length is enabled for INCR4-WRAP / INCR8-WRAP / INCR16-WRAP
26 WECRM	Wakeup Event Enable On SD Card Removal This bit enables a wakeup event, via a Card Removal, in the Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not effect this bit. When this bit is set, the Card Removal Status and the uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active in order to assert the Card Removal Status and the uSDHC interrupt. 1 Enable 0 Disable
25 WECINS	Wakeup Event Enable On SD Card Insertion This bit enables a wakeup event, via a Card Insertion, in the Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not effect this bit. When this bit is set, the Card Insertion Status and the uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active in order to assert the Card Insertion Status and the uSDHC interrupt. 1 Enable 0 Disable
24 WECINT	Wakeup Event Enable On Card Interrupt This bit enables a wakeup event, via a Card Interrupt, in the Interrupt Status register. This bit can be set to 1 if FN_WUS (Wake Up Support) in CIS is set to 1. When this bit is set, the Card Interrupt Status and the uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active in order to assert the Card Interrupt Status and the uSDHC interrupt.

Table continues on the next page...

uSDHCx_PROT_CTRL field descriptions (continued)

Field	Description
	1 Enable 0 Disable
23–21 -	Reserved. Always write as 3'b100
20 RD_DONE_NO_8CLK	*Read done no 8 clock:* *According to the SD/MMC spec, for read data transaction, 8 clocks are needed after the end bit of the last data block. So, by default(RD_DONE_NO_8CLK=0), 8 clocks will be active after the end bit of the last read data transaction.* *However, this 8 clocks should not be active if user wants to use stop at block gap(include the auto stop at block gap in boot mode) feature for read and the RWCTL bit(bit18) is not enabled. In this case, software should set RD_DONE_NO_8CLK to avoid this 8 clocks. Otherwise, the device may send extra data to uSDHC while uSDHC ignores these data.* *In a summary, this bit should be set only if the use case needs to use stop at block gap feature while the device can't support the read wait feature.*
19 IABG	Interrupt At Block Gap This bit is valid only in 4-bit mode, of the SDIO card, and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SDIO card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0 to avoid an inadvertent interrupt. When the Host Driver detects an SDIO card insertion, it shall set this bit according to the CCCR of the card. 1 Enabled 0 Disabled
18 RWCTL	Read Wait Control The read wait function is optional for SDIO cards. If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using the DATA2 line. Otherwise the uSDHC has to stop the SD Clock to hold read data, which restricts commands generation. When the Host Driver detects an SDIO card insertion, it shall set this bit according to the CCCR of the card. If the card does not support read wait, this bit shall never be set to 1, otherwise DATA line conflicts may occur. If this bit is set to 0, stop at block gap during read operation is also supported, but the uSDHC will stop the SD Clock to pause reading operation. 1 Enable Read Wait Control, and assert Read Wait without stopping SD Clock at block gap when SABGREQ bit is set 0 Disable Read Wait Control, and stop SD Clock at block gap when SABGREQ bit is set
17 CREQ	Continue Request This bit is used to restart a transaction which was stopped using the Stop At Block Gap Request. When a Suspend operation is not accepted by the card, it is also by setting this bit to restart the paused transfer. To cancel stop at the block gap, set Stop At Block Gap Request to 0 and set this bit to 1 to restart the transfer. The uSDHC automatically clears this bit, therefore it is not necessary for the Host Driver to set this bit to 0. If both Stop At Block Gap Request and this bit are 1, the continue request is ignored. 1 Restart 0 No effect
16 SABGREQ	Stop At Block Gap Request This bit is used to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the Transfer Complete is set to 1, indicating a transfer completion, the Host Driver shall

Table continues on the next page...

uSDHCx_PROT_CTRL field descriptions (continued)

Field	Description
	leave this bit set to 1. Clearing both the Stop At Block Gap Request and Continue Request does not cause the transaction to restart. Read Wait is used to stop the read transaction at the block gap. The uSDHC will honor the Stop At Block Gap Request for write transfers, but for read transfers it requires that the SDIO card support Read Wait. Therefore, the Host Driver shall not set this bit during read transfers unless the SDIO card supports Read Wait and has set the Read Wait Control to 1, otherwise the uSDHC will stop the SD bus clock to pause the read operation during block gap. In the case of write transfers in which the Host Driver writes data to the Data Port register, the Host Driver shall set this bit after all block data is written. If this bit is set to 1, the Host Driver shall not write data to the Data Port register after a block is sent. Once this bit is set, the Host Driver shall not clear this bit before the Transfer Complete bit in Interrupt Status Register is set, otherwise the uSDHCs behavior is undefined. This bit effects Read Transfer Active, Write Transfer Active, DATA Line Active and Command Inhibit (DATA) in the Present State register. 1 Stop 0 Transfer
15–10 Reserved	This read-only field is reserved and always has the value 0.
9–8 DMASEL	DMA Select This field is valid while DMA (SDMA or ADMA) is enabled and selects the DMA operation. 00 No DMA or Simple DMA is selected 01 ADMA1 is selected 10 ADMA2 is selected 11 reserved
7 CDSS	Card Detect Signal Selection This bit selects the source for the card detection. 1 Card Detection Test Level is selected (for test purpose). 0 Card Detection Level is selected (for normal purpose).
6 CDTL	Card Detect Test Level This is bit is enabled while the Card Detection Signal Selection is set to 1 and it indicates card insertion. 1 Card Detect Test Level is 1, card inserted 0 Card Detect Test Level is 0, no card inserted
5–4 EMODE	Endian Mode The uSDHC supports all three endian modes in data transfer. Refer to Data Buffer for more details. 00 Big Endian Mode 01 Half Word Big Endian Mode 10 Little Endian Mode 11 Reserved
3 D3CD	DATA3 as Card Detection Pin If this bit is set, DATA3 should be pulled down to act as a card detection pin. Be cautious when using this feature, because DATA3 is also a chip-select for the SPI mode. A pull-down on this pin and CMD0 may set the card into the SPI mode, which the uSDHC does not support.

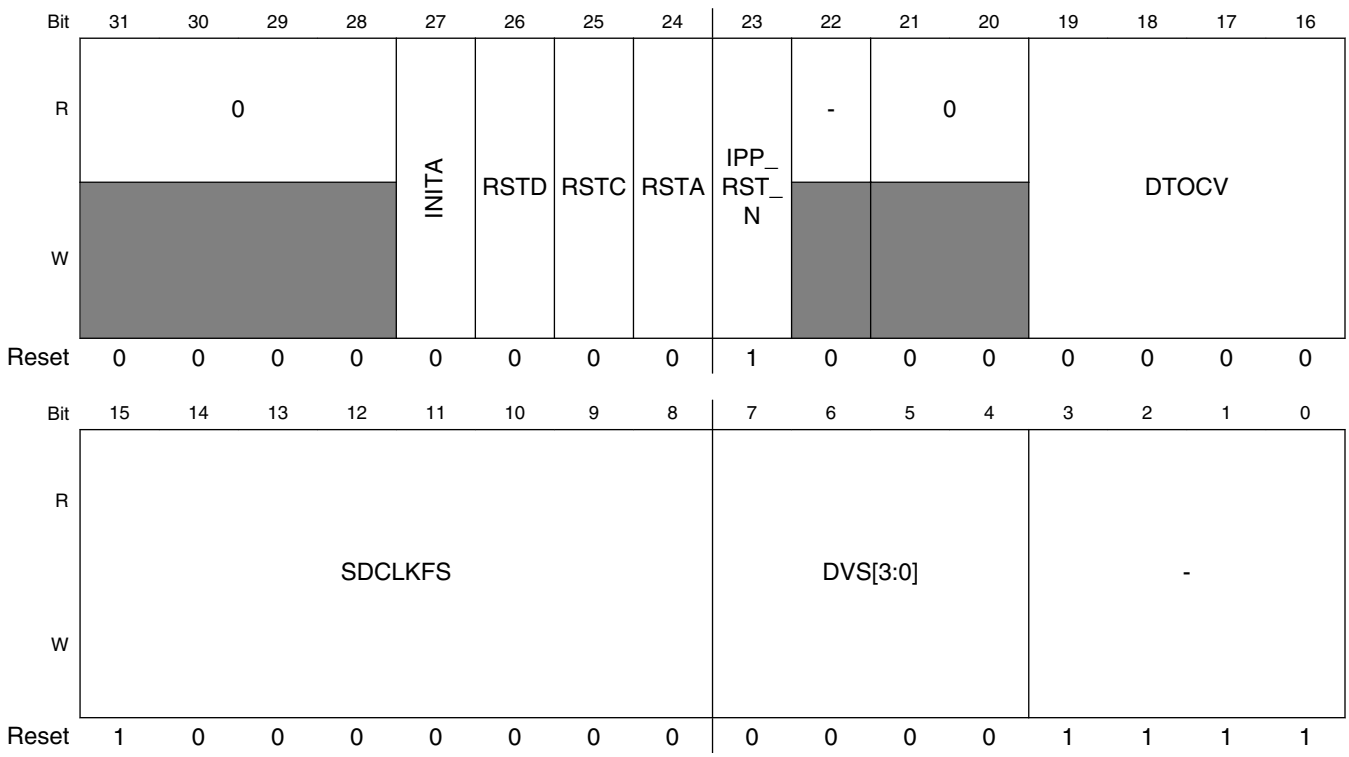
Table continues on the next page...

uSDHCx_PROT_CTRL field descriptions (continued)

Field	Description
	1 DATA3 as Card Detection Pin 0 DATA3 does not monitor Card Insertion
2–1 DTW[1:0]	Data Transfer Width This bit selects the data width of the SD bus for a data transfer. The Host Driver shall set it to match the data width of the card. Possible Data transfer Width is 1-bit, 4-bits or 8-bits. 10 8-bit mode 01 4-bit mode 00 1-bit mode 11 Reserved
0 LCTL	LED Control This bit, fully controlled by the Host Driver, is used to caution the user not to remove the card while the card is being accessed. If the software is going to issue multiple SD commands, this bit can be set during all these transactions. It is not necessary to change for each transaction. When the software issues multiple SD commands, setting the bit once before the first command is sufficient: it is not necessary to reset the bit between commands. 1 LED on 0 LED off

67.8.12 System Control (uSDHCx_SYS_CTRL)

Address: Base address + 2Ch offset



uSDHCx_SYS_CTRL field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27 INITA	Initialization Active When this bit is set, 80 SD-Clocks are sent to the card. After the 80 clocks are sent, this bit is self cleared. This bit is very useful during the card power-up period when 74 SD-Clocks are needed and the clock auto gating feature is enabled. Writing 1 to this bit when this bit is already 1 has no effect. Writing 0 to this bit at any time has no effect. When either of the CIHB and CDIHB bits in the Present State Register are set, writing 1 to this bit is ignored (i.e. when command line or data lines are active, write to this bit is not allowed). On the otherhand, when this bit is set, i.e., during intialization active period, it is allowed to issue command, and the command bit stream will appear on the CMD pad after all 80 clock cycles are done. So when this command ends, the driver can make sure the 80 clock cycles are sent out. This is very useful when the driver needs send 80 cycles to the card and does not want to wait till this bit is self cleared.
26 RSTD	Software Reset For DATA Line Only part of the data circuit is reset. DMA circuit is also reset. After this bit is set, SW waits for self-clear. The following registers and bits are cleared by this bit: • Data Port register

Table continues on the next page...

uSDHCx_SYS_CTRL field descriptions (continued)

Field	Description
	• Buffer is cleared and initialized. • Present State register • Buffer Read Enable • Buffer Write Enable • Read Transfer Active • Write Transfer Active • DATA Line Active • Command Inhibit (DATA) Protocol Control register • Continue Request • Stop At Block Gap Request Interrupt Status register • Buffer Read Ready • Buffer Write Ready • DMA Interrupt • Block Gap Event • Transfer Complete **NOTE:** When reset, SW must make sure there is no incomplete data transferring. If there is data transfer going on, SW need wait TC or DC uSDHC_INT_STATUS register is set. 1 Reset 0 No Reset
25 RSTC	Software Reset For CMD Line Only part of the command circuit is reset. After this bit is set, SW waits for self-clear. The following registers and bits are cleared by this bit: • Present State register Command Inhibit (CMD) • Interrupt Status register Command Complete 1 Reset 0 No Reset
24 RSTA	Software Reset For ALL This reset effects the entire Host Controller except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC are cleared. During its initialization, the Host Driver shall set this bit to 1 to reset the uSDHC. The uSDHC shall reset this bit to 0 when the capabilities registers are valid and the Host Driver can read them. Additional use of Software Reset For All does not affect the value of the Capabilities registers. After this bit is set, it is recommended that the Host Driver reset the external card and re-initialize it. After this bit is set, SW should wait for self-clear. **NOTE:** When reset, SW must make sure there is no incomplete data transferring. If there is data transfer going on, SW need wait TC or DC uSDHC_INT_STATUS register is set. 1 Reset 0 No Reset
23 IPP_RST_N	This register's value will be output to CARD from pad directly for hardware reset of the card if the card supports this feature.
22 -	Reserved
21–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 DTOCV	Data Timeout Counter Value

Table continues on the next page...

uSDHCx_SYS_CTRL field descriptions (continued)

Field	Description
	This value determines the interval by which DAT line timeouts are detected. Refer to the Data Timeout Error bit in the Interrupt Status register for information on factors that dictate time-out generation. Time-out clock frequency will be generated by dividing the base clock SDCLK value by this value. The Host Driver can clear the Data Timeout Error Status Enable (in the Interrupt Status Enable register) to prevent inadvertent time-out events. 1111b SDCLK x 2^{29} 1110b SDCLK x 2^{28} ... 0001b SDCLK x 2^{15} 0000b SDCLK x 2^{14}
15–8 SDCLKFS	SDCLK Frequency Select This register is used to select the frequency of the SDCLK pin.This register field selects the divide value of the second divisor stage. *In Single Data Rate mode(DDR_EN bit of MIXERCTRL is '0')* Only the following settings are allowed: 80h) Base clock divided by 256 40h) Base clock divided by 128 20h) Base clock divided by 64 10h) Base clock divided by 32 08h) Base clock divided by 16 04h) Base clock divided by 8 02h) Base clock divided by 4 01h) Base clock divided by 2 00h) Base clock divided by 1 *While in Dual Data Rate mode(DDR_EN bit of MIXERCTRL is '1')* *Only the following settings are allowed:* *80h) Base clock divided by 512* *40h) Base clock divided by 256* *20h) Base clock divided by 128* *10h) Base clock divided by 64* *08h) Base clock divided by 32* *04h) Base clock divided by 16* *02h) Base clock divided by 8* *01h) Base clock divided by 4* *00h) Base clock divided by 2* *When S/W changes the DDR_EN bit, SDCLKFS may need to be changed also!* In Single Data Rate mode, setting 00h bypasses the frequency prescaler of the SD Clock. Multiple bits must not be set, or the behavior of this prescaler is undefined. The two default divider values can be calculated by the frequency of ipg_perclk and the following Divisor bits.

Table continues on the next page...

uSDHCx_SYS_CTRL field descriptions (continued)

Field	Description
	The frequency of SDCLK is set by the following formula: Clock Frequency = (Base Clock) / (prescaler x divisor) For example, in Single Data Rate mode, if the Base Clock Frequency is 96 MHz, and the target frequency is 25 MHz, then choosing the prescaler value of 01h and divisor value of 1h will yield 24 MHz, which is the nearest frequency less than or equal to the target. Similarly, to approach a clock value of 400 kHz, the prescaler value of 08h and divisor value of eh yields the exact clock value of 400 kHz. The reset value of this bit field is 80h, so if the input Base Clock (ipg_perclk) is about 96 MHz, the default SD Clock after reset is 375 kHz. According to the SD Physical Specification Version 1.1 and the SDIO Card Specification Version 1.2, the maximum SD Clock frequency is 50 MHz and shall never exceed this limit. *Before changing clock divisor value(SDCLKFS or DVS), Host Driver should make sure the SDSTB bit is high.* *If setting SDCLKFS and DVS can generate same clock frequency,(For example, in SDR mode, SDCLKFS = 01h is same as DVS = 01h.) SDCLKFS is highly recommended.*
7–4 DVS[3:0]	Divisor This register is used to select the frequency of the SDCLK pin. This register field selects the divide value of the first divisor stage. *Before changing clock divisor value(SDCLKFS or DVS), Host Driver should make sure the SDSTB bit is high.* The setting are as following: 0000 Divide-by-1 0001 Divide-by-2 ..... 1110 Divide-by-15 1111 Divide-by-16
-	Reserved. Always write as 1.

67.8.13 Interrupt Status (uSDHCx_INT_STATUS)

An interrupt is generated when the Normal Interrupt Signal Enable is enabled and at least one of the status bits is set to 1. For all bits, writing 1 to a bit clears it; writing to 0 keeps the bit unchanged. More than one status can be cleared with a single register write. For Card Interrupt, before writing 1 to clear, it is required that the card stops asserting the interrupt, meaning that when the Card Driver services the interrupt condition, otherwise the CINT bit will be asserted again.

The table below shows the relationship between the Command Timeout Error and the Command Complete.

Table 67-11. uSDHC Status for Command Timeout Error/Command Complete Bit Combinations

Command Complete	Command Timeout Error	Meaning of the Status
0	0	X
X	1	Response not received within 64 SDCLK cycles
1	0	Response received

The table below shows the relationship between the Transfer Complete and the Data Timeout Error.

Table 67-12. uSDHC Status for Data Timeout Error/Transfer Complete Bit Combinations

Transfer Complete	Data Timeout Error	Meaning of the Status
0	0	X
0	1	Timeout occurred during transfer
1	X	Data Transfer Complete

The table below shows the relationship between the Command CRC Error and Command Timeout Error.

Table 67-13. uSDHC Status for Command CRC Error/Command Timeout Error Bit Combinations

Command Complete	Command Timeout Error	Meaning of the Status
0	0	No error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

Address: Base address + 30h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			DMAE	0	TNE	0	AC12E	0	DEBE	DCE	DTOE	CIE	CEBE	CCE	CTOE
W				w1c		w1c		w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	TP	0	RTE	0			CINT	CRM	CINS	BRR	BWR	DINT	BGE	TC	CC
W		w1c		w1c				w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

uSDHCx_INT_STATUS field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28 DMAE	DMA Error Occurs when an Internal DMA transfer has failed. This bit is set to 1, when some error occurs in the data transfer. This error can be caused by either Simple DMA or ADMA, depending on which DMA is in use. The value in DMA System Address register is the next fetch address where the error occurs. Since any error corrupts the whole data block, the Host Driver shall re-start the transfer from the corrupted block boundary. The address of the block boundary can be calculated either from the current DS_ADDR value or from the remaining number of blocks and the block size. 1 Error 0 No Error
27 Reserved	This read-only field is reserved and always has the value 0.
26 TNE	Tuning Error: (only for SD3.0 SDR104 mode) This bit is set when an unrecoverable error is detected in a tuning circuit. By detecting Tuning Error, Host Driver needs to abort a command executing and perform tuning.
25 Reserved	This read-only field is reserved and always has the value 0.
24 AC12E	Auto CMD12 Error Occurs when detecting that one of the bits in the Auto CMD12 Error Status register has changed from 0 to 1. This bit is set to 1, not only when the errors in Auto CMD12 occur, but also when the Auto CMD12 is not executed due to the previous command error. 1 Error 0 No Error
23 Reserved	This read-only field is reserved and always has the value 0.
22 DEBE	Data End Bit Error Occurs either when detecting 0 at the end bit position of read data, which uses the DATA line, or at the end bit position of the CRC. 1 Error 0 No Error
21 DCE	Data CRC Error

Table continues on the next page...

uSDHCx_INT_STATUS field descriptions (continued)

Field	Description
	Occurs when detecting a CRC error when transferring read data, which uses the DATA line, or when detecting the Write CRC status having a value other than 010. 1 Error 0 No Error
20 DTOE	Data Timeout Error Occurs when detecting one of following time-out conditions. • Busy time-out for R1b, R5b type • Busy time-out after Write CRC status • Read Data time-out. 1 Time out 0 No Error
19 CIE	Command Index Error Occurs if a Command Index error occurs in the command response. 1 Error 0 No Error
18 CEBE	Command End Bit Error Occurs when detecting that the end bit of a command response is 0. 1 End Bit Error Generated 0 No Error
17 CCE	Command CRC Error Command CRC Error is generated in two cases. • If a response is returned and the Command Timeout Error is set to 0 (indicating no time-out), this bit is set when detecting a CRC error in the command response. • The uSDHC detects a CMD line conflict by monitoring the CMD line when a command is issued. If the uSDHC drives the CMD line to 1, but detects 0 on the CMD line at the next SDCLK edge, then the uSDHC shall abort the command (Stop driving CMD line) and set this bit to 1. The Command Timeout Error shall also be set to 1 to distinguish CMD line conflict. 1 CRC Error Generated. 0 No Error
16 CTOE	Command Timeout Error Occurs only if no response is returned within 64 SDCLK cycles from the end bit of the command. If the uSDHC detects a CMD line conflict, in which case a Command CRC Error shall also be set (as shown in Interrupt Status (uSDHC_INT_STATUS)), this bit shall be set without waiting for 64 SDCLK cycles. This is because the command will be aborted by the uSDHC. 1 Time out 0 No Error
15 Reserved	This read-only field is reserved and always has the value 0.
14 TP	Tuning Pass:(only for SD3.0 SDR104 mode) Current CMD19 transfer is done successfully. That is, current sampling point is correct.

Table continues on the next page...

uSDHCx_INT_STATUS field descriptions (continued)

Field	Description
13 Reserved	This read-only field is reserved and always has the value 0.
12 RTE	Re-Tuning Event: (only for SD3.0 SDR104 mode) This status is set if Re-Tuning Request in the Present State register changes from 0 to 1. Host Controller requests Host Driver to perform re-tuning for next data transfer. Current data transfer (not large block count) can be completed without re-tuning. 1 Re-Tuning should be performed 0 Re-Tuning is not required
11–9 Reserved	This read-only field is reserved and always has the value 0.
8 CINT	Card Interrupt This status bit is set when an interrupt signal is detected from the external card. In 1-bit mode, the uSDHC will detect the Card Interrupt without the SD Clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so the interrupt from card can only be sampled during interrupt cycle, introducing some delay between the interrupt signal from the SDIO card and the interrupt to the Host System. Writing this bit to 1 can clear this bit, but as the interrupt source from the SDIO card does not clear, this bit is set again. In order to clear this bit, it is required to reset the interrupt source from the external card followed by a writing 1 to this bit. When this status has been set, and the Host Driver needs to service this interrupt, the Card Interrupt Signal Enable in the Interrupt Signal Enable register should be 0 to stop driving the interrupt signal to the Host System. After completion of the card interrupt service (It should reset the interrupt sources in the SDIO card and the interrupt signal may not be asserted), write 1 to clear this bit, set the Card Interrupt Signal Enable to 1, and start sampling the interrupt signal again. 1 Generate Card Interrupt 0 No Card Interrupt
7 CRM	Card Removal This status bit is set if the Card Inserted bit in the Present State register changes from 1 to 0. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card state may possibly be changed when the Host Driver clears this bit and the interrupt event may not be generated. When this bit is cleared, it will be set again if no card is inserted. In order to leave it cleared, clear the Card Removal Status Enable bit in Interrupt Status Enable register. 1 Card removed 0 Card state unstable or inserted
6 CINS	Card Insertion This status bit is set if the Card Inserted bit in the Present State register changes from 0 to 1. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card state may possibly be changed when the Host Driver clears this bit and the interrupt event may not be generated. When this bit is cleared, it will be set again if a card is inserted. In order to leave it cleared, clear the Card Inserted Status Enable bit in Interrupt Status Enable register. 1 Card inserted 0 Card state unstable or removed
5 BRR	Buffer Read Ready

Table continues on the next page...

uSDHCx_INT_STATUS field descriptions (continued)

Field	Description
	This status bit is set if the Buffer Read Enable bit, in the Present State register, changes from 0 to 1. Refer to the Buffer Read Enable bit in the Present State register for additional information. 1 Ready to read buffer 0 Not ready to read buffer
4 BWR	Buffer Write Ready This status bit is set if the Buffer Write Enable bit, in the Present State register, changes from 0 to 1. Refer to the Buffer Write Enable bit in the Present State register for additional information. 1 Ready to write buffer: 0 Not ready to write buffer
3 DINT	DMA Interrupt Occurs only when the internal DMA finishes the data transfer successfully. Whenever errors occur during data transfer, this bit will not be set. Instead, the DMAE bit will be set. Either Simple DMA or ADMA finishes data transferring, this bit will be set. 1 DMA Interrupt is generated 0 No DMA Interrupt
2 BGE	Block Gap Event If the Stop At Block Gap Request bit in the Protocol Control register is set, this bit is set when a read or write transaction is stopped at a block gap. If Stop At Block Gap Request is not set to 1, this bit is not set to 1. In the case of a Read Transaction: This bit is set at the falling edge of the DATA Line Active Status (When the transaction is stopped at SD Bus timing). The Read Wait must be supported in order to use this function. In the case of Write Transaction: This bit is set at the falling edge of Write Transfer Active Status (After getting CRC status at SD Bus timing). 1 Transaction stopped at block gap 0 No block gap event
1 TC	Transfer Complete This bit is set when a read or write transfer is completed. In the case of a Read Transaction: This bit is set at the falling edge of the Read Transfer Active Status. There are two cases in which this interrupt is generated. The first is when a data transfer is completed as specified by the data length (after the last data has been read to the Host System). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request bit in the Protocol Control register (after valid data has been read to the Host System). In the case of a Write Transaction: This bit is set at the falling edge of the DATA Line Active Status. There are two cases in which this interrupt is generated. The first is when the last data is written to the SD card as specified by the data length and the busy signal is released. The second is when data transfers are stopped at the block gap, by setting the Stop At Block Gap Request bit in the Protocol Control register, and the data transfers are completed. (after valid data is written to the SD card and the busy signal released). In the case of a command with busy, this bit is set when busy is deasserted. 1 Transfer complete 0 Transfer not complete

Table continues on the next page...

uSDHCx_INT_STATUS field descriptions (continued)

Field	Description
0 CC	Command Complete This bit is set when you receive the end bit of the command response (except Auto CMD12). Refer to the Command Inhibit (CMD) in the Present State register. 1 Command complete 0 Command not complete

67.8.14 Interrupt Status Enable (uSDHCx_INT_STATUS_EN)

Setting the bits in this register to 1 enables the corresponding Interrupt Status to be set by the specified event. If any bit is cleared, the corresponding Interrupt Status bit is also cleared (i.e. when the bit in this register is cleared, the corresponding bit in Interrupt Status Register is always 0).

- Depending on IABG bit setting, uSDHC may be programmed to sample the card interrupt signal during the interrupt period and hold its value in the flip-flop. There will be some delays on the Card Interrupt, asserted from the card, to the time the Host System is informed.
- To detect a CMD line conflict, the Host Driver must set both Command Timeout Error Status Enable and Command CRC Error Status Enable to 1.

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			0		0		0								
W				DMAESEN	TNESEN		AC12ESEN		DEBESEN	DCESEN	DTOESEN	CIESEN	CEBESEN	CCESEN	CTOESEN	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0		0											
W		TPSEN		RTESEN				CINTSEN	CRMSEN	CINSSEN	BRRSEN	BWRSEN	DINTSEN	BGESEN	TCSEN	CCSEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

uSDHCx_INT_STATUS_EN field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

uSDHCx_INT_STATUS_EN field descriptions (continued)

Field	Description
28 DMAESEN	DMA Error Status Enable 1 Enabled 0 Masked
27 Reserved	This read-only field is reserved and always has the value 0.
26 TNESEN	Tuning Error Status Enable 1 Enabled 0 Masked
25 Reserved	This read-only field is reserved and always has the value 0.
24 AC12ESEN	Auto CMD12 Error Status Enable 1 Enabled 0 Masked
23 Reserved	This read-only field is reserved and always has the value 0.
22 DEBESEN	Data End Bit Error Status Enable 1 Enabled 0 Masked
21 DCESEN	Data CRC Error Status Enable 1 Enabled 0 Masked
20 DTOESEN	Data Timeout Error Status Enable 1 Enabled 0 Masked
19 CIESEN	Command Index Error Status Enable 1 Enabled 0 Masked
18 CEBESEN	Command End Bit Error Status Enable 1 Enabled 0 Masked
17 CCESEN	Command CRC Error Status Enable 1 Enabled 0 Masked
16 CTOESEN	Command Timeout Error Status Enable 1 Enabled 0 Masked
15 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

uSDHCx_INT_STATUS_EN field descriptions (continued)

Field	Description
14 TPSEN	Tuning Pass Status Enable 1 Enabled 0 Masked
13 Reserved	This read-only field is reserved and always has the value 0.
12 RTESEN	Re-Tuning Event Status Enable 1 Enabled 0 Masked
11–9 Reserved	This read-only field is reserved and always has the value 0.
8 CINTSEN	Card Interrupt Status Enable If this bit is set to 0, the uSDHC will clear the interrupt request to the system. The Card Interrupt detection is stopped when this bit is cleared and restarted when this bit is set to 1. The Host Driver should clear the Card Interrupt Status Enable before servicing the Card Interrupt and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. 1 Enabled 0 Masked
7 CRMSEN	Card Removal Status Enable 1 Enabled 0 Masked
6 CINSSEN	Card Insertion Status Enable 1 Enabled 0 Masked
5 BRRSEN	Buffer Read Ready Status Enable 1 Enabled 0 Masked
4 BWRSEN	Buffer Write Ready Status Enable 1 Enabled 0 Masked
3 DINTSEN	DMA Interrupt Status Enable 1 Enabled 0 Masked
2 BGESEN	Block Gap Event Status Enable 1 Enabled 0 Masked
1 TCSEN	Transfer Complete Status Enable 1 Enabled 0 Masked

Table continues on the next page...

uSDHCx_INT_STATUS_EN field descriptions (continued)

Field	Description
0 CCSEN	Command Complete Status Enable 1 Enabled 0 Masked

67.8.15 Interrupt Signal Enable (uSDHCx_INT_SIGNAL_EN)

This register is used to select which interrupt status is indicated to the Host System as the interrupt. These status bits all share the same interrupt line. Setting any of these bits to 1 enables interrupt generation. The corresponding Status register bit will generate an interrupt when the corresponding interrupt signal enable bit is set.

Address: Base address + 38h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				0		0		0							
W				DMAEIEN		TNEIEN		AC12EIEN		DEBEIEN	DCEIEN	DTOEIEN	CIEIEN	CEBEIEN	CCEIEN	CTOEIEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0		0											
W		TPIEN		RTEIEN				CINTIEN	CRMIEN	CINSIEN	BRRIEN	BWRIEN	DINTIEN	BGEIEN	TCIEN	CCIEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

uSDHCx_INT_SIGNAL_EN field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28 DMAEIEN	DMA Error Interrupt Enable 1 Enable 0 Masked
27 Reserved	This read-only field is reserved and always has the value 0.
26 TNEIEN	Tuning Error Interrupt Enable 1 Enabled 0 Masked

Table continues on the next page...

uSDHCx_INT_SIGNAL_EN field descriptions (continued)

Field	Description
25 Reserved	This read-only field is reserved and always has the value 0.
24 AC12EIEN	Auto CMD12 Error Interrupt Enable 1 Enabled 0 Masked
23 Reserved	This read-only field is reserved and always has the value 0.
22 DEBEIEN	Data End Bit Error Interrupt Enable 1 Enabled 0 Masked
21 DCEIEN	Data CRC Error Interrupt Enable 1 Enabled 0 Masked
20 DTOEIEN	Data Timeout Error Interrupt Enable 1 Enabled 0 Masked
19 CIEIEN	Command Index Error Interrupt Enable 1 Enabled 0 Masked
18 CEBEIEN	Command End Bit Error Interrupt Enable 1 Enabled 0 Masked
17 CCEIEN	Command CRC Error Interrupt Enable 1 Enabled 0 Masked
16 CTOEIEN	Command Timeout Error Interrupt Enable 1 Enabled 0 Masked
15 Reserved	This read-only field is reserved and always has the value 0.
14 TPIEN	Tuning Pass Interrupt Enable 1 Enabled 0 Masked
13 Reserved	This read-only field is reserved and always has the value 0.
12 RTEIEN	Re-Tuning Event Interrupt Enable 1 Enabled 0 Masked

Table continues on the next page...

uSDHCx_INT_SIGNAL_EN field descriptions (continued)

Field	Description
11–9 Reserved	This read-only field is reserved and always has the value 0.
8 CINTIEN	Card Interrupt Interrupt Enable 1 Enabled 0 Masked
7 CRMIEN	Card Removal Interrupt Enable 1 Enabled 0 Masked
6 CINSIEN	Card Insertion Interrupt Enable 1 Enabled 0 Masked
5 BRRIEN	Buffer Read Ready Interrupt Enable 1 Enabled 0 Masked
4 BWRIEN	Buffer Write Ready Interrupt Enable 1 Enabled 0 Masked
3 DINTIEN	DMA Interrupt Enable 1 Enabled 0 Masked
2 BGEIEN	Block Gap Event Interrupt Enable 1 Enabled 0 Masked
1 TCIEN	Transfer Complete Interrupt Enable 1 Enabled 0 Masked
0 CCIEN	Command Complete Interrupt Enable 1 Enabled 0 Masked

67.8.16 Auto CMD12 Error Status (uSDHCx_AUTOCMD12_ERR_STATUS)

When the Auto CMD12 Error Status bit in the Status register is set, the Host Driver shall check this register to identify what kind of error the Auto CMD12 / CMD 23 indicated. Auto CMD23 errors are indicated in bit 04-01.This register is valid only when the Auto CMD12 Error status bit is set.

The table bwlow shows the relationship between the Auto CMGD12 CRC Error and the Auto CMD12 Command Timeout Error.

Table 67-14. Relationship Between Command CRC Error and Command Timeout Error for Auto CMD12

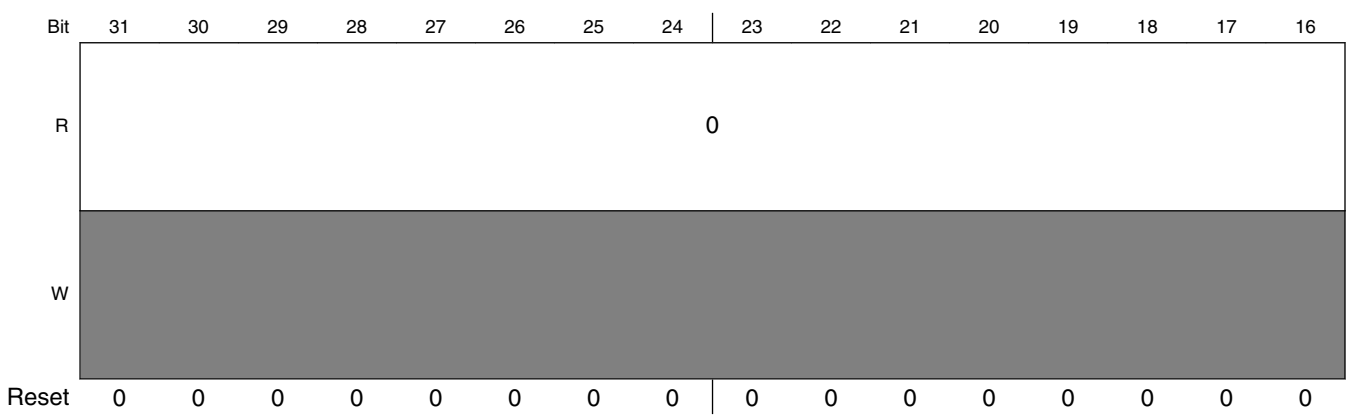
Auto CMD12 CRC Error	Auto CMD12 Timeout Error	Type of Error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

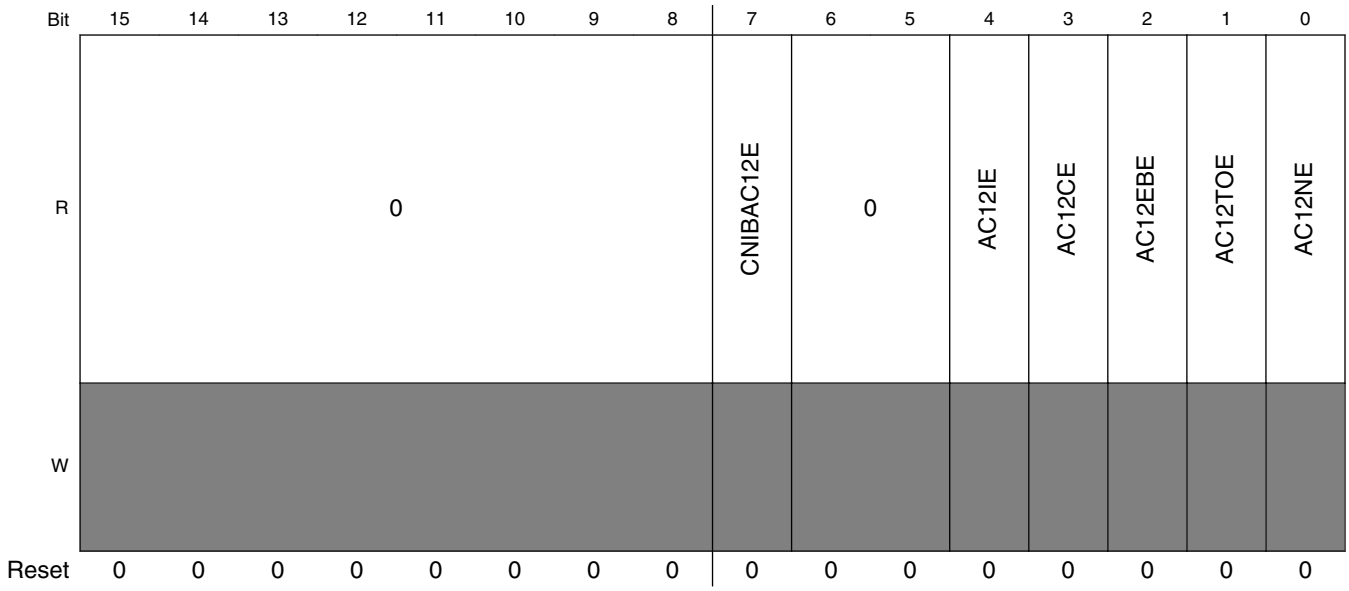
Changes in Auto CMD12 Error Status register can be classified in three scenarios:

1. When the uSDHC is going to issue an Auto CMD12.
   - Set bit 0 to 1 if the Auto CMD12 can't be issued due to an error in the previous command
   - Set bit 0 to 0 if the Auto CMD12 is issued
2. At the end bit of an Auto CMD12 response.
   - Check errors correspond to bits 1-4.
   - Set bits 1-4 corresponding to detected errors.
   - Clear bits 1-4 corresponding to detected errors
3. Before reading the Auto CMD12 Error Status bit 7.
   - Set bit 7 to 1 if there is a command that can't be issued
   - Clear bit 7 if there is no command to issue

The timing for generating the Auto CMD12 Error and writing to the Command register are asynchronous. After that, bit 7 shall be sampled when the driver is not writing to the Command register. So it is suggested to read this register only when the AC12E bit in Interrupt Status register is set. An Auto CMD12 Error Interrupt is generated when one of the error bits (0-4) is set to 1. The Command Not Issued By Auto CMD12 Error does not generate an interrupt.

Address: Base address + 3Ch offset





uSDHCx_AUTOCMD12_ERR_STATUS field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.
7 CNIBAC12E	Command Not Issued By Auto CMD12 Error Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 Error (D04-D01) in this register. 1 Not Issued 0 No error
6–5 Reserved	This read-only field is reserved and always has the value 0.
4 AC12IE	Auto CMD12 / 23 Index Error Occurs if the Command Index error occurs in response to a command. 1 Error, the CMD index in response is not CMD12/23 0 No error
3 AC12CE	Auto CMD12 / 23 CRC Error Occurs when detecting a CRC error in the command response. 1 CRC Error Met in Auto CMD12/23 Response 0 No CRC error
2 AC12EBE	Auto CMD12 / 23 End Bit Error Occurs when detecting that the end bit of command response is 0 which should be 1. 1 End Bit Error Generated 0 No error
1 AC12TOE	Auto CMD12 / 23 Timeout Error

Table continues on the next page...

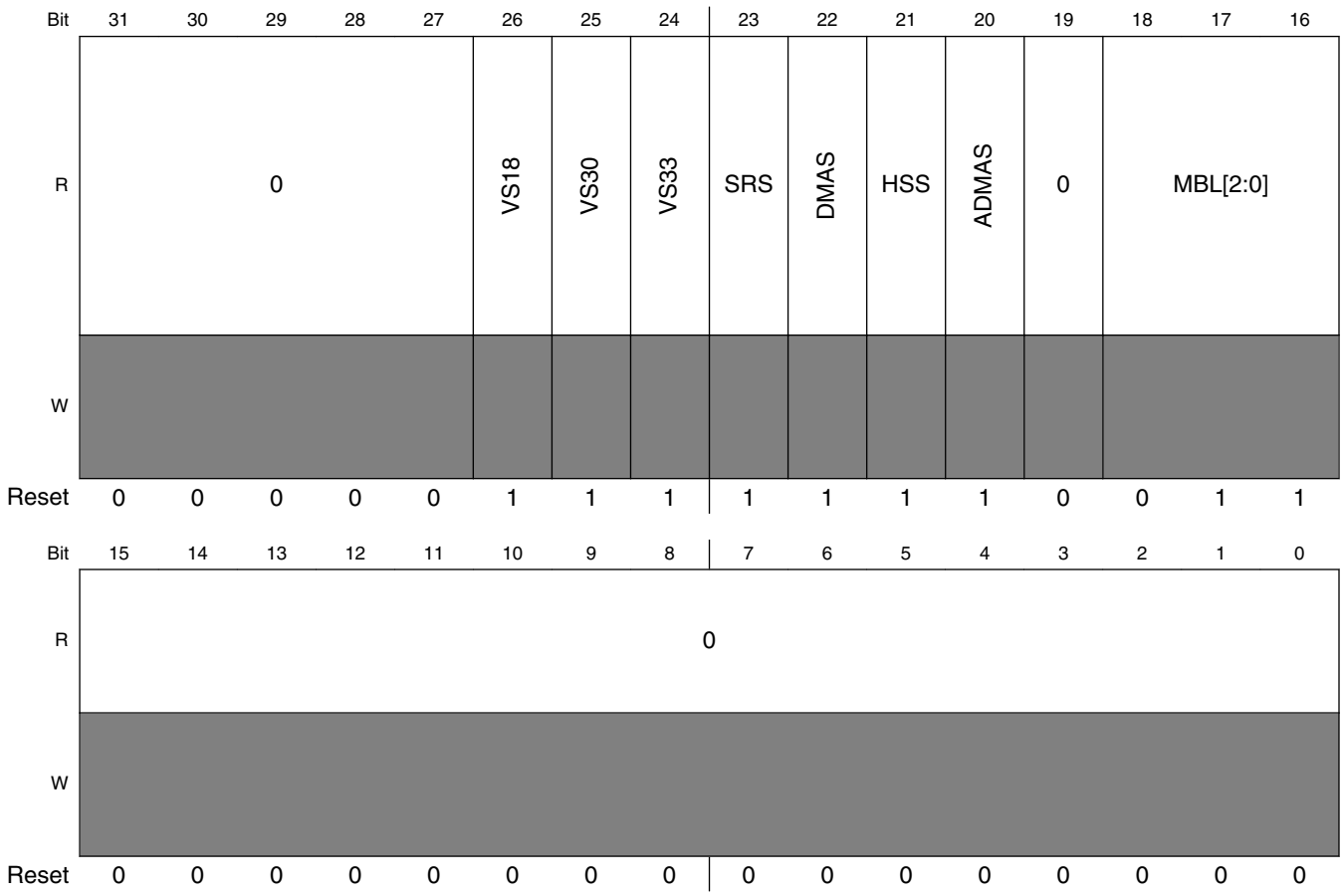
uSDHCx_AUTOCMD12_ERR_STATUS field descriptions (continued)

Field	Description
	Occurs if no response is returned within 64 SDCLK cycles from the end bit of the command. If this bit is set to1, the other error status bits (2-4) have no meaning. 1 Time out 0 No error
0 AC12NE	Auto CMD12 Not Executed If memory multiple block data transfer is not started, due to a command error, this bit is not set because it is not necessary to issue an Auto CMD12. Setting this bit to 1 means the uSDHC cannot issue the Auto CMD12 to stop a memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (1-4) have no meaning. 1 Not executed 0 Executed

67.8.17 Host Controller Capabilities (uSDHCx_HOST_CTRL_CAP)

This register provides the Host Driver with information specific to the uSDHC implementation. The value in this register is the power-on-reset value, and does not change with a software reset.

Address: Base address + 40h offset



uSDHCx_HOST_CTRL_CAP field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26 VS18	Voltage Support 1.8 V This bit shall depend on the Host System ability. 1 1.8V supported 0 1.8V not supported

Table continues on the next page...

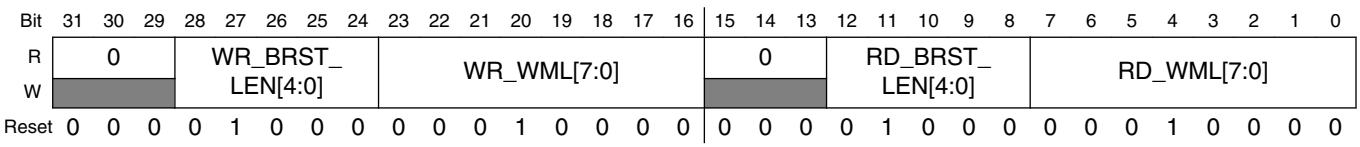
uSDHCx_HOST_CTRL_CAP field descriptions (continued)

Field	Description
25 VS30	Voltage Support 3.0 V This bit shall depend on the Host System ability. 1 3.0V supported 0 3.0V not supported
24 VS33	Voltage Support 3.3V This bit shall depend on the Host System ability. 1 3.3V supported 0 3.3V not supported
23 SRS	Suspend / Resume Support This bit indicates whether the uSDHC supports Suspend / Resume functionality. If this bit is 0, the Suspend and Resume mechanism, as well as the Read Wait, are not supported, and the Host Driver shall not issue either Suspend or Resume commands. 1 Supported 0 Not supported
22 DMAS	DMA Support This bit indicates whether the uSDHC is capable of using the internal DMA to transfer data between system memory and the data buffer directly. 1 DMA Supported 0 DMA not supported
21 HSS	High Speed Support This bit indicates whether the uSDHC supports High Speed mode and the Host System can supply a SD Clock frequency from 25 MHz to 50 MHz. 1 High Speed Supported 0 High Speed Not Supported
20 ADMAS	ADMA Support This bit indicates whether the uSDHC supports the ADMA feature. 1 Advanced DMA Supported 0 Advanced DMA Not supported
19 Reserved	This read-only field is reserved and always has the value 0.
18–16 MBL[2:0]	Max Block Length This value indicates the maximum block size that the Host Driver can read and write to the buffer in the uSDHC. The buffer shall transfer block size without wait cycles. 000 512 bytes 001 1024 bytes 010 2048 bytes 011 4096 bytes
Reserved	This read-only field is reserved and always has the value 0.

67.8.18 Watermark Level (uSDHCx_WTMK_LVL)

Both write and read watermark levels (FIFO threshold) are configurable. There value can range from 1 to 128 words. Both write and read burst lengths are also Configurable. There value can range from 1 to 31 words.

Address: Base address + 44h offset



uSDHCx_WTMK_LVL field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–24 WR_BRST_ LEN[4:0]	Write Burst Length [1] The number of words the uSDHC writes in a single burst. The write burst length must be less than or equal to the write watermark level, and all bursts within a watermark level transfer will be in back-to-back mode. On reset, this field will be 8. Writing 0 to this field will result in '01000' (i.e. it is not able to clear this field).
23–16 WR_WML[7:0]	Write Watermark Level The number of words used as the watermark level (FIFO threshold) in a DMA write operation. Also the number of words as a sequence of write bursts in back-to-back mode. The maximum legal value for the write watermark level is 128.
15–13 Reserved	This read-only field is reserved and always has the value 0.
12–8 RD_BRST_ LEN[4:0]	Read Burst Length [2] The number of words the uSDHC reads in a single burst. The read burst length must be less than or equal to the read watermark level, and all bursts within a watermark level transfer will be in back-to-back mode. On reset, this field will be 8. Writing 0 to this field will result in '01000' (i.e. it is not able to clear this field).
RD_WML[7:0]	Read Watermark Level The number of words used as the watermark level (FIFO threshold) in a DMA read operation. Also the number of words as a sequence of read bursts in back-to-back mode. The maximum legal value for the read water mark level is 128.

1. Due to system restriction, the actual burst length may not exceed 16.

2. Due to system restriction, the actual burst length may not exceed 16.

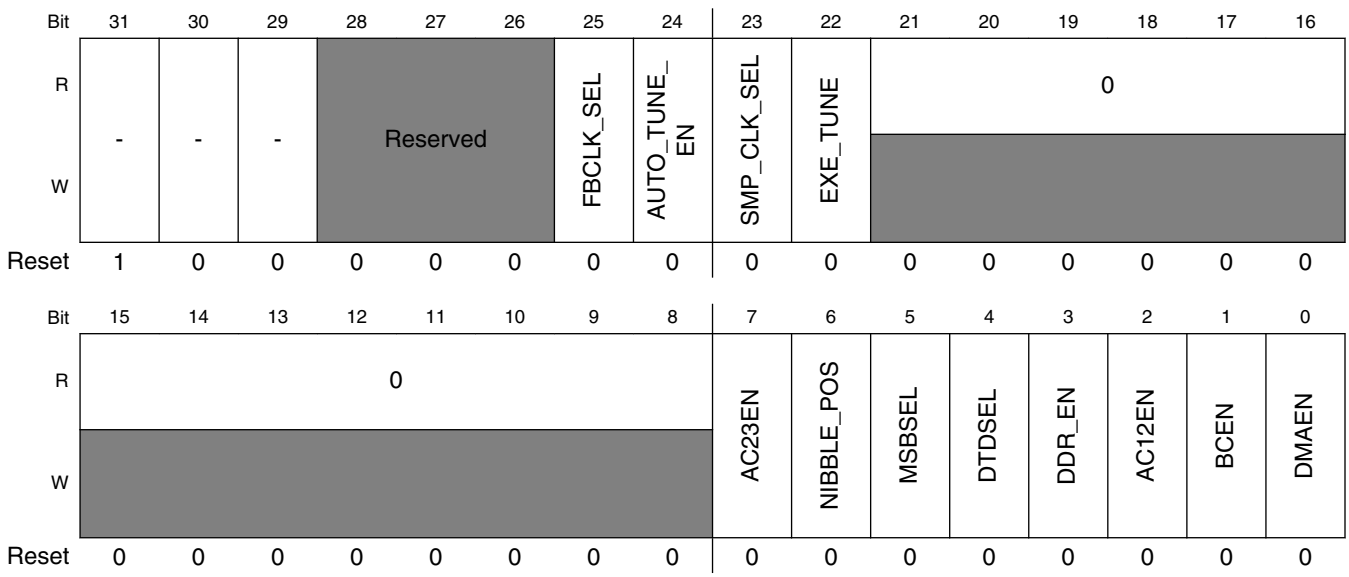
67.8.19 Mixer Control (uSDHCx_MIX_CTRL)

This register is used to DMA and data transfer. To prevent data loss, The software should check if data transfer is active before writing this register. These bits are DPSEL, MBSEL, DTDSEL, AC12EN, BCEN, and DMAEN.

Table 67-15. Transfer Type Register Setting for Various Transfer Types

Multi/Single Block Select	Block Count Enable	Block Count	Function
0	Don't Care	Don't Care	Single Transfer
1	0	Don't Care	Infinite Transfer
1	1	Positive Number	Multiple Transfer
1	1	Zero	No Data Transfer

Address: Base address + 48h offset



uSDHCx_MIX_CTRL field descriptions

Field	Description
31 -	Reserved. Always write as 1
30 -	Reserved. Always write as 0.
29 -	Reserved. Always write as 0.
28–26 -	This field is reserved. Reserved
25 FBCLK_SEL	Feedback Clock Source Selection (Only used for SD3.0, SDR104 mode)

Table continues on the next page...

uSDHCx_MIX_CTRL field descriptions (continued)

Field	Description
	1 Feedback clock comes from the ipp_card_clk_out 0 Feedback clock comes from the loopback CLK
24 AUTO_TUNE_EN	Auto Tuning Enable (Only used for SD3.0, SDR104 mode) 1 Enable auto tuning 0 Disable auto tuning
23 SMP_CLK_SEL	Tuned clock or Fixed clock is used to sample data / cmd (Only used for SD3.0, SDR104 mode) 1 Tuned clock is used to sample data / cmd 0 Fixed clock is used to sample data / cmd
22 EXE_TUNE	Execute Tuning: (Only used for SD3.0, SDR104 mode) This bit is set to 1 to indicate the Host Driver is starting tuning procedure. Tuning procedure is aborted by writing 0. 1 Execute Tuning 0 Not Tuned or Tuning Completed
21–8 Reserved	This read-only field is reserved and always has the value 0.
7 AC23EN	Auto CMD23 Enable When this bit is set to 1, the Host Controller issues a CMD23 automatically before issuing a command specified in the Command Register.
6 NIBBLE_POS	In DDR 4-bit mode nibble position indictation. 0- the sequence is 'odd high nibble -> even high nibble -> odd low nibble -> even low nibble'; 1- the sequence is 'odd high nibble -> odd low nibble -> even high nibble -> even low nibble'.
5 MSBSEL	Multi / Single Block Select This bit enables multiple block DATA line data transfers. For any other commands, this bit can be set to 0. If this bit is 0, it is not necessary to set the Block Count register. (Refer to Command Transfer Type (uSDHC_CMD_XFR_TYP)). 1 Multiple Blocks 0 Single Block
4 DTDSEL	Data Transfer Direction Select This bit defines the direction of DATA line data transfers. The bit is set to 1 by the Host Driver to transfer data from the SD card to the uSDHC and is set to 0 for all other commands. 1 Read (Card to Host) 0 Write (Host to Card)
3 DDR_EN	Dual Data Rate mode selection
2 AC12EN	Auto CMD12 Enable Multiple block transfers for memory require a CMD12 to stop the transaction. When this bit is set to 1, the uSDHC will issue a CMD12 automatically when the last block transfer has completed. The Host Driver shall not set this bit to issue commands that do not require CMD12 to stop a multiple block data transfer. In particular, secure commands defined in File Security Specification (see reference list) do not require CMD12. In single block transfer, the uSDHC will ignore this bit no matter it is set or not.

Table continues on the next page...

uSDHCx_MIX_CTRL field descriptions (continued)

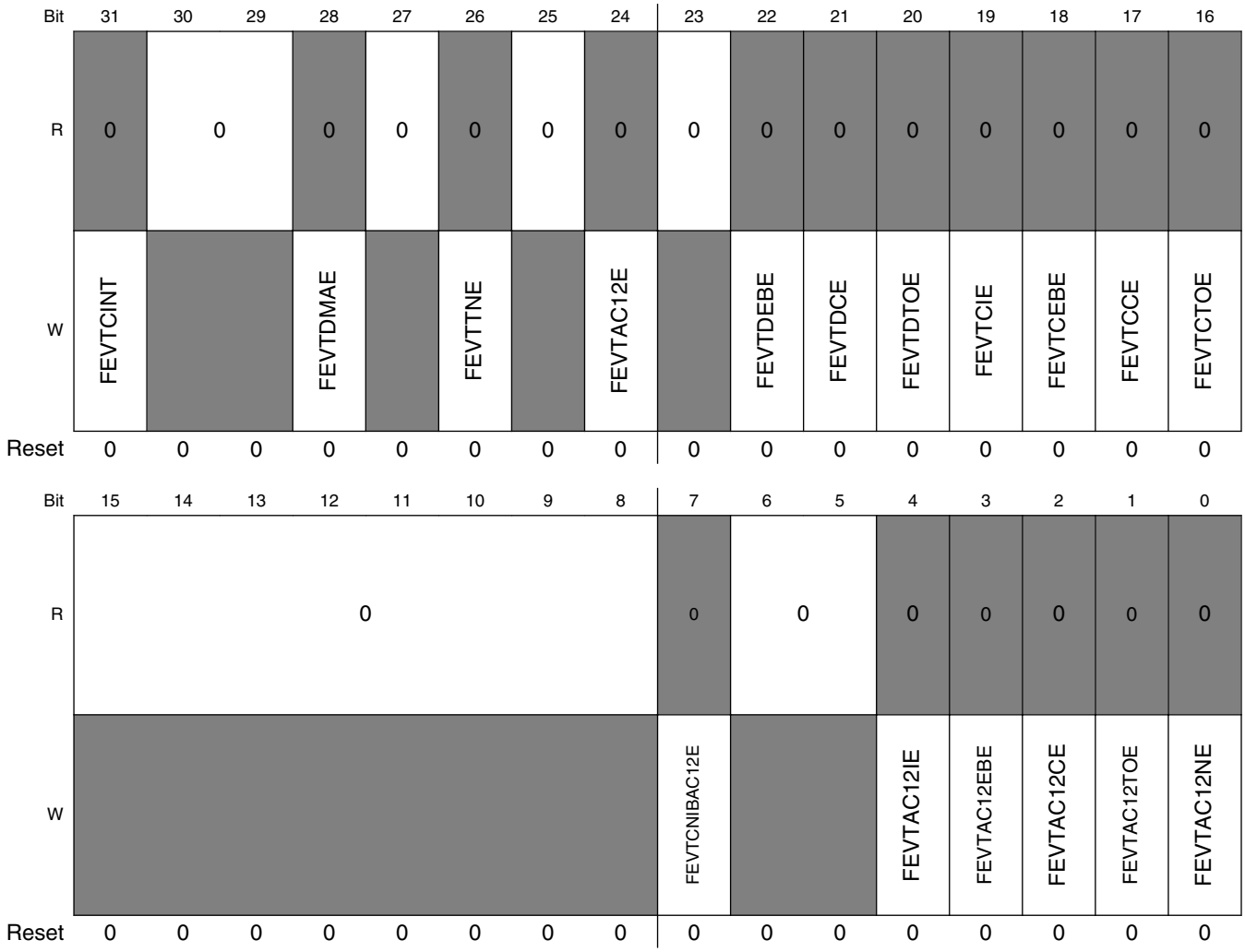
Field	Description
	1 Enable 0 Disable
1 BCEN	Block Count Enable This bit is used to enable the Block Count register, which is only relevant for multiple block transfers. When this bit is 0, the internal counter for block is disabled, which is useful in executing an infinite transfer. 1 Enable 0 Disable
0 DMAEN	DMA Enable This bit enables DMA functionality. If this bit is set to 1, a DMA operation shall begin when the Host Driver sets the DPSEL bit of this register. Whether the Simple DMA or the Advanced DMA is active depends on the DMA Select field of the Protocol Control register. 1 Enable 0 Disable

67.8.20 Force Event (uSDHCx_FORCE_EVENT)

The Force Event Register is not a physically implemented register. Rather, it is an address at which the Interrupt Status Register can be written if the corresponding bit of the Interrupt Status Enable Register is set. This register is a write only register and writing 0 to it has no effect. Writing 1 to this register actually sets the corresponding bit of Interrupt Status Register. A read from this register always results in 0's. In order to change the corresponding status bits in the Interrupt Status Register, make sure to set IPGEN bit in System Control Register so that IPG_CLK is always active.

Forcing a card interrupt will generate a short pulse on the DATA1 line, and the driver may treat this interrupt as a normal interrupt. The interrupt service routine may skip polling the card interrupt factor as the interrupt is self cleared.

Address: Base address + 50h offset



uSDHC*x*_FORCE_EVENT field descriptions

Field	Description
31 FEVTCINT	Force Event Card Interrupt Writing 1 to this bit generates a short low-level pulse on the internal DATA1 line, as if a self clearing interrupt was received from the external card. If enabled, the CINT bit will be set and the interrupt service routine may treat this interrupt as a normal interrupt from the external card.
30–29 Reserved	This read-only field is reserved and always has the value 0.
28 FEVTDMAE	Force Event DMA Error Forces the DMAE bit of Interrupt Status Register to be set.
27 Reserved	This read-only field is reserved and always has the value 0.
26 FEVTTNE	Force Tuning Error Forces the TNE bit of Interrupt Status Register to be set.
25 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

uSDHCx_FORCE_EVENT field descriptions (continued)

Field	Description
24 FEVTAC12E	Force Event Auto Command 12 Error Forces the AC12E bit of Interrupt Status Register to be set.
23 Reserved	This read-only field is reserved and always has the value 0.
22 FEVTDEBE	Force Event Data End Bit Error Forces the DEBE bit of Interrupt Status Register to be set.
21 FEVTDCE	Force Event Data CRC Error Forces the DCE bit of Interrupt Status Register to be set.
20 FEVTDTOE	Force Event Data Time Out Error Force the DTOE bit of Interrupt Status Register to be set.
19 FEVTCIE	Force Event Command Index Error Forces the CCE bit of Interrupt Status Register to be set.
18 FEVTCEBE	Force Event Command End Bit Error Forces the CEBE bit of Interrupt Status Register to be set.
17 FEVTCCE	Force Event Command CRC Error Forces the CCE bit of Interrupt Status Register to be set.
16 FEVTCTOE	Force Event Command Time Out Error Forces the CTOE bit of Interrupt Status Register to be set.
15–8 Reserved	This read-only field is reserved and always has the value 0.
7 FEVTCNIBAC12E	Force Event Command Not Executed By Auto Command 12 Error Forces the CNIBAC12E bit in the Auto Command12 Error Status Register to be set.
6–5 Reserved	This read-only field is reserved and always has the value 0.
4 FEVTAC12IE	Force Event Auto Command 12 Index Error Forces the AC12IE bit in the Auto Command12 Error Status Register to be set.
3 FEVTAC12EBE	Force Event Auto Command 12 End Bit Error Forces the AC12EBE bit in the Auto Command12 Error Status Register to be set.
2 FEVTAC12CE	Force Event Auto Command 12 CRC Error Forces the AC12CE bit in the Auto Command12 Error Status Register to be set.
1 FEVTAC12TOE	Force Event Auto Command 12 Time Out Error Forces the AC12TOE bit in the Auto Command12 Error Status Register to be set.
0 FEVTAC12NE	Force Event Auto Command 12 Not Executed Forces the AC12NE bit in the Auto Command12 Error Status Register to be set.

67.8.21 ADMA Error Status Register (uSDHCx_ADMA_ERR_STATUS)

When an ADMA Error Interrupt has occurred, the ADMA Error States field in this register holds the ADMA state and the ADMA System Address register holds the address around the error descriptor.

For recovering from this error, the Host Driver requires the ADMA state to identify the error descriptor address as follows:

- ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address.
- ST_FDS: Current location set in the ADMA System Address register is the error descriptor address.
- ST_CADR: This state is never set because it only increments the descriptor pointer and doesn't generate an ADMA error.
- ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address.

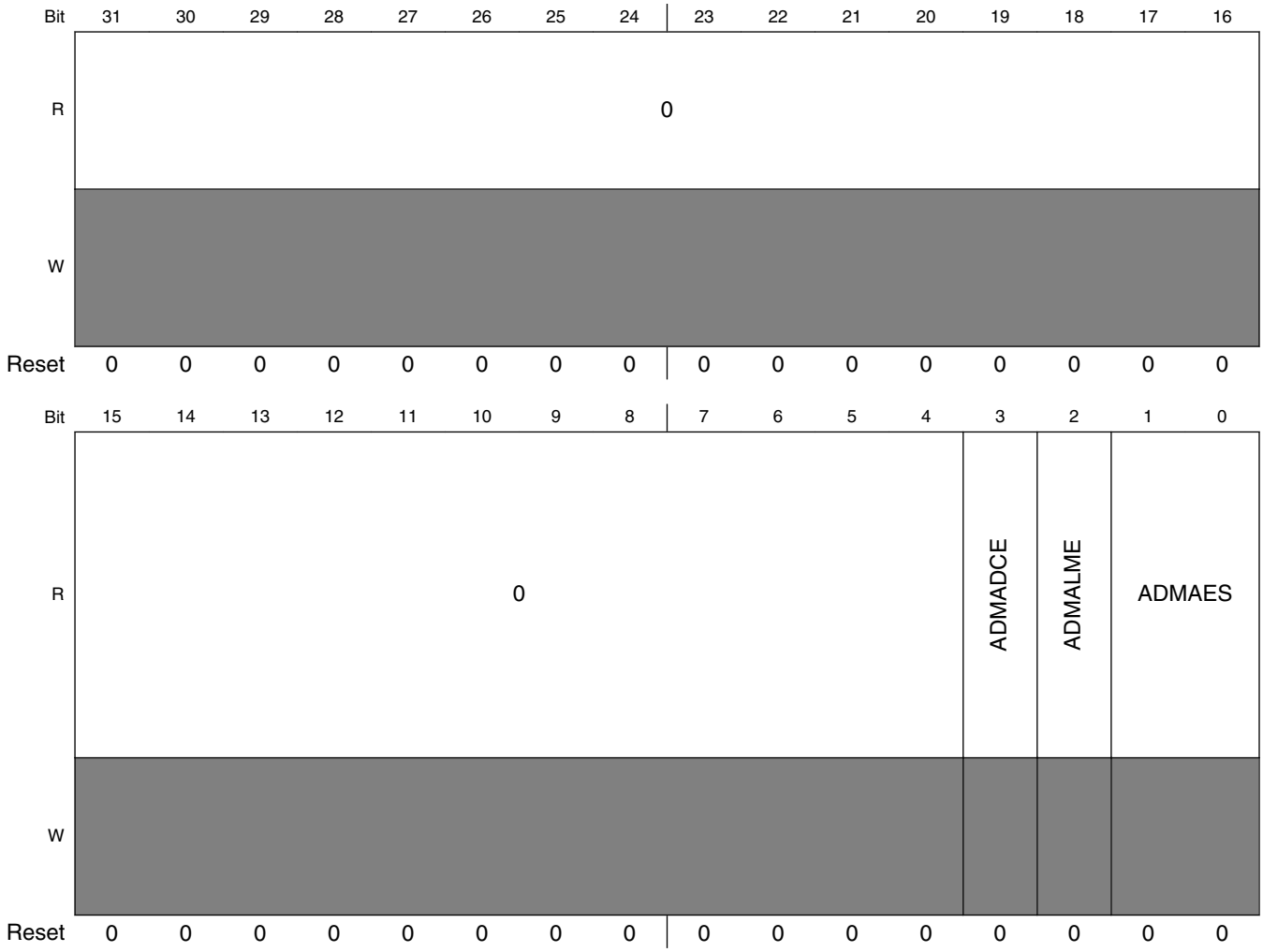
In case of a write operation, the Host Driver should use the ACMD22 to get the number of the written block, rather than using this information, since unwritten data may exist in the Host Controller.

The Host Controller generates the ADMA Error Interrupt when it detects invalid descriptor data (Valid=0) in the ST_FDS state. The Host Driver can distinguish this error by reading the Valid bit of the error descriptor.

Table 67-16. ADMA Error State Coding

D01-D00	ADMA Error State (when error has occurred)	Contents of ADMA System Address Register
00	ST_STOP (Stop DMA)	Holds the address of the next executable Descriptor command
01	ST_FDS (Fetch Descriptor)	Holds the valid Descriptor address
10	ST_CADR (Change Address)	No ADMA Error is generated
11	ST_TFR (Transfer Data)	Holds the address of the next executable Descriptor command

Address: Base address + 54h offset



uSDHC*x*_ADMA_ERR_STATUS field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value 0.
3 ADMADCE	ADMA Descritor Error This error occurs when invalid descriptor fetched by ADMA. 1 Error 0 No Error
2 ADMALME	ADMA Length Mismatch Error This error occurs in the following 2 cases: • While the Block Count Enable is being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length. • Total data length cannot be divided by the block length. 1 Error 0 No Error
ADMAES	ADMA Error State (when ADMA Error is occurred)

Table continues on the next page...

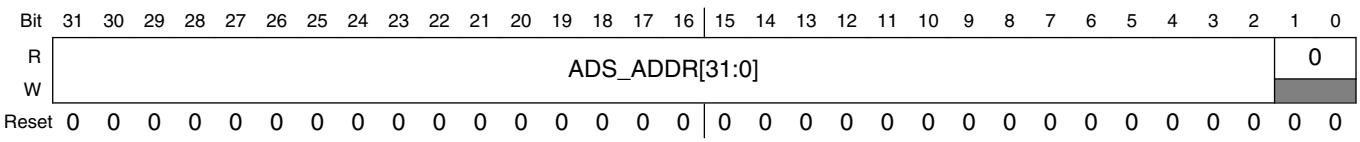
uSDHCx_ADMA_ERR_STATUS field descriptions (continued)

Field	Description
	This field indicates the state of the ADMA when an error has occurred during an ADMA data transfer. Refer to ADMA Error Status Register (uSDHC_ADMA_ERR_STATUS) for more details.

67.8.22 ADMA System Address (uSDHCx_ADMA_SYS_ADDR)

This register contains the physical system memory address used for ADMA transfers.

Address: Base address + 58h offset



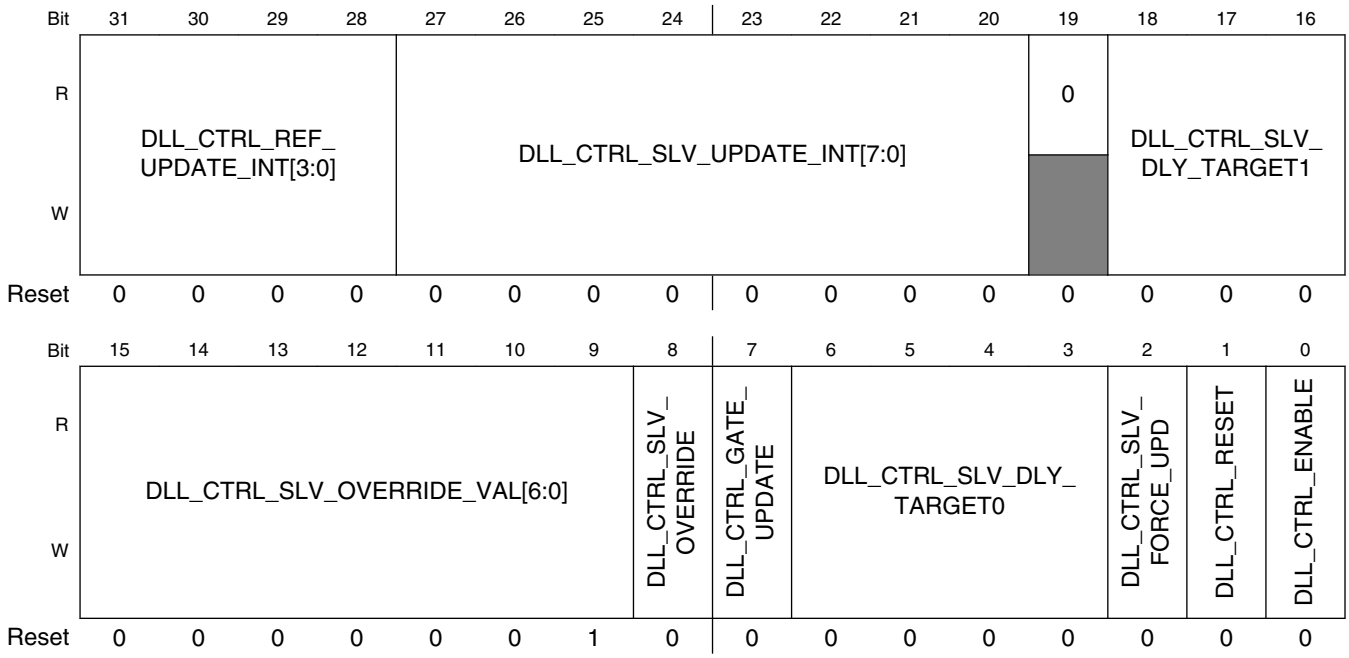
uSDHCx_ADMA_SYS_ADDR field descriptions

Field	Description
31–2 ADS_ ADDR[31:0]	ADMA System Address This register holds the word address of the executing command in the Descriptor table. At the start of ADMA, the Host Driver shall set the start address of the Descriptor table. The ADMA engine increments this register address whenever fetching a Descriptor command. When the ADMA is stopped at the Block Gap, this register indicates the address of the next executable Descriptor command. When the ADMA Error Interrupt is generated, this register shall hold the valid Descriptor address depending on the ADMA state. The lower 2 bits of this register is tied to '0' so the ADMA address is always word aligned. Since this register supports dynamic address reflecting, when TC bit is set, it automatically alters the value of internal address counter, so SW cannot change this register when TC bit is set. Such restriction is also listed in Software Restrictions .
Reserved	This read-only field is reserved and always has the value 0.

67.8.23 DLL (Delay Line) Control (uSDHCx_DLL_CTRL)

This register contains control bits for DLL.

Address: Base address + 60h offset



uSDHCx_DLL_CTRL field descriptions

Field	Description
31–28 DLL_CTRL_REF_UPDATE_INT[3:0]	DLL control loop update interval. The interval cycle is (2 + REF_UPDATE_INT) * REF_CLOCK. By default, the DLL control loop shall update every two REF_CLOCK cycles. It should be noted that increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that may effect the delay (such as voltage and temperature)
27–20 DLL_CTRL_SLV_UPDATE_INT[7:0]	Slave delay line update interval. If default 0 is used, it means 256 cycles of REF_CLOCK. A value of 0x0f results in 15 cycles and so on. Note that software can always cause an update of the slave-delay line using the SLV_FORCE_UPDATE register. Note that the slave delay line will also update automatically when the reference DLL transitions to a locked state (from an un-locked state).
19 Reserved	This read-only field is reserved and always has the value 0.
18–16 DLL_CTRL_SLV_DLY_TARGET1	Refer to DLL_CTRL_SLV_DLY_TARGET0 below.
15–9 DLL_CTRL_SLV_OVERRIDE_VAL[6:0]	When SLV_OVERRIDE = 1 This field is used to select 1 of 128 physical taps manually. A value of 0 selects tap 1, and a value of 0x7f selects tap 128.

Table continues on the next page...

uSDHC*x*_DLL_CTRL field descriptions (continued)

Field	Description
8 DLL_CTRL_ SLV_OVERRIDE	Set this bit to 1 to Enable manual override for slave delay chain using SLV_OVERRIDE_VAL; to set 0 to disable manual override. This feature does not require the DLL to be enabled using the ENABLE bit. In fact to reduce power, if SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE = 0
7 DLL_CTRL_ GATE_UPDATE	Set this bit to 1 to prevent the DLL from updating (since when clock_in exists, glitches may appear during DLL updates). This bit may be used by software if such a condition occurs. Clear the bit to 0 to allow the DLL to update automatically.
6–3 DLL_CTRL_ SLV_DLY_ TARGET0	The delay target for the uSDHC loopback read clock can be programmed in 1/16th increments of an ref_clock half-period. The delay is ((DLL_CTRL_SLV_DLY_TARGET1 +1)* REF_CLOCK / 2) / 16 So the input read-clock can be delayed relative input data from (REF_CLOCK / 2) / 16 to REF_CLOCK * 4.
2 DLL_CTRL_ SLV_FORCE_ UPD	Setting this bit to 1, forces the slave delay line to update to the DLL calibrated value immediately. The slave delay line shall update automatically based on the SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the slave-line update can only occur if SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered). Be sure to use it when uSDHC is idle. This function may not work when uSDHC is working on data / cmd / response.
1 DLL_CTRL_ RESET	Setting this bit to 1 force a reset on DLL. This will cause the DLL to lose lock and re-calibrate to detect an REF_CLOCK half period phase shift. This signal is used by the DLL as edge-sensitive, so in order to create a subsequent reset, RESET must be taken low and then asserted again.
0 DLL_CTRL_ ENABLE	Set this bit to 1 to enable the DLL and delay chain; otherwise; set to 0 to bypasses DLL. Note that using the slave delay line override feature with SLV_OVERRIDE and SLV_OVERRIDE VAL, the DLL does not need to be enabled.

67.8.24 DLL Status (uSDHC*x*_DLL_STATUS)

This register contains the DLL status information. All bits are read only and will read the same as the power-reset value.

Address: Base address + 64h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DLL_STS_REF_SEL[6:0]								DLL_STS_SLV_SEL[6:0]						DLL_STS_REF_LOCK	DLL_STS_SLV_LOCK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

uSDHCx_DLL_STATUS field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–9 DLL_STS_REF_SEL[6:0]	Reference delay line select taps. This is encoded by 7 bits for 127 taps.
8–2 DLL_STS_SLV_SEL[6:0]	Slave delay line select status. This is the instant value generated from reference chain. Since the reference chain can only be updated when REF_CLOCK is detected, this value should be the right value to be updated when the reference is locked.
1 DLL_STS_REF_LOCK	Reference DLL lock status. This signifies that the DLL has detected and locked to a half-phase ref_clock shift, allowing the slave delay-line to perform programmed clock delays
0 DLL_STS_SLV_LOCK	Slave delay-line lock status. This signifies that a valid calibration has been set to the slave-delay line and that the slave-delay line is implementing the programmed delay value

67.8.25 CLK Tuning Control and Status (uSDHCx_CLK_TUNE_CTRL_STATUS)

This register contains the Clock Tuning Control status information. All bits are read only and will read the same as the power-reset value. This register is added to support SD3.0 UHS-I SDR104 mode.

Address: Base address + 68h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PRE_ERR	TAP_SEL_PRE[6:0]							TAP_SEL_OUT[3:0]				TAP_SEL_POST[3:0]			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NXT_ERR	DLY_CELL_SET_PRE[6:0]							DLY_CELL_SET_OUT[6:0]				DLY_CELL_SET_POST[6:0]			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

uSDHCx_CLK_TUNE_CTRL_STATUS field descriptions

Field	Description
31 PRE_ERR	PRE error which means the number of delay cells added on the feedback clock is too small. It is valid only when SMP_CLK_SEL of Mix control register (bit23 of 0x48) is enabled.

Table continues on the next page...

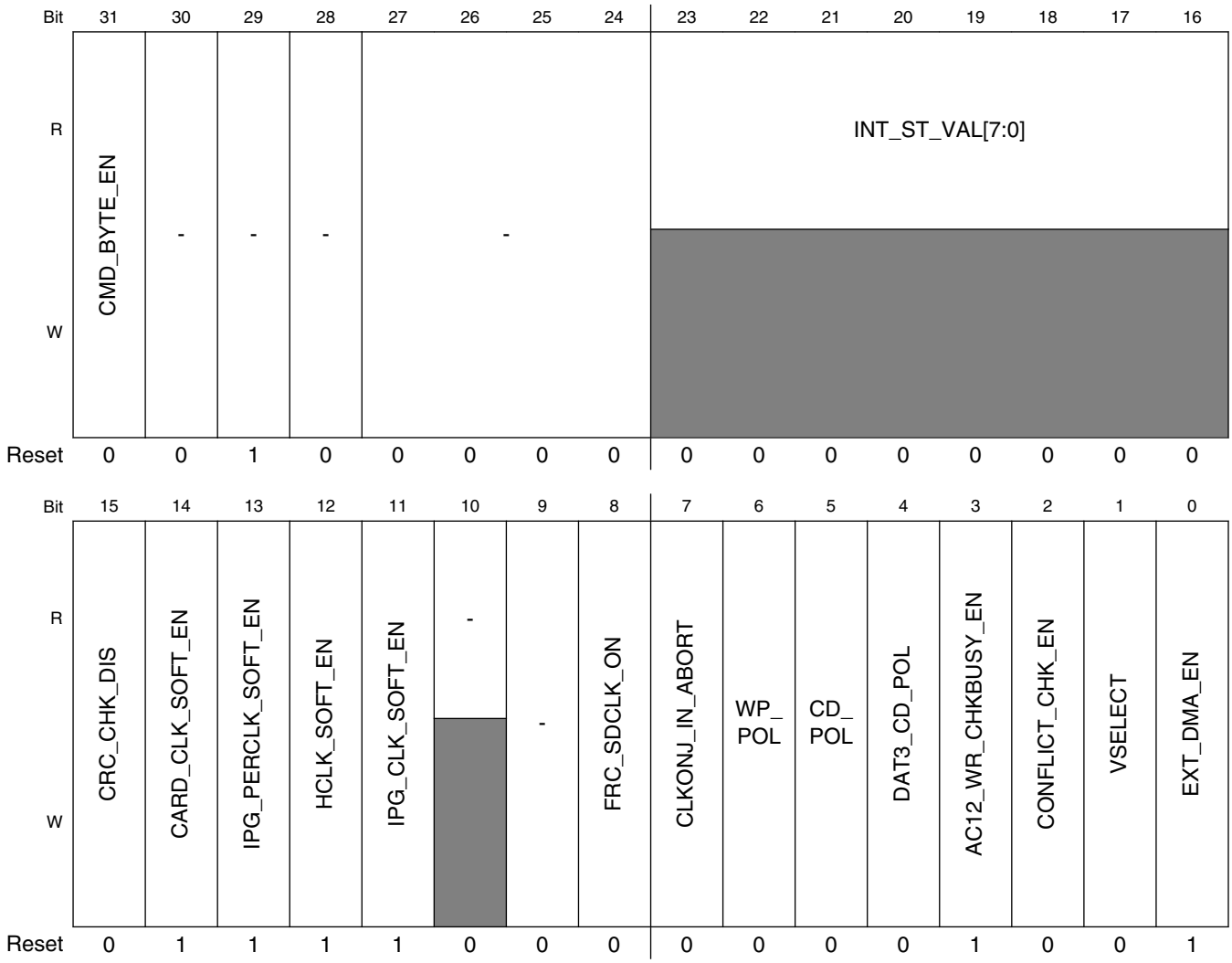
uSDHCx_CLK_TUNE_CTRL_STATUS field descriptions (continued)

Field	Description
30–24 TAP_SEL_ PRE[6:0]	Reflects the number of delay cells added on the feedback clock between the feedback clock and CLK_PRE. When AUTO_TUNE_EN (bit24 of 0x48) is disabled, TAP_SEL_PRE is always equal to DLY_CELL_SET_PRE. When AUTO_TUNE_EN (bit24 of 0x48) is enabled, TAP_SEL_PRE will be updated automatically according to the status of the auto tuning circuit to adjust the sample clock phase.
23–20 TAP_SEL_ OUT[3:0]	Reflect the number of delay cells added on the feedback clock between CLK_PRE and CLK_OUT.
19–16 TAP_SEL_ POST[3:0]	Reflect the number of delay cells added on the feedback clock between CLK_OUT and CLK_POST.
15 NXT_ERR	NXT error which means the number of delay cells added on the feedback clock is too large. It's valid only when SMP_CLK_SEL of Mix control register (bit23 of 0x48) is enabled.
14–8 DLY_CELL_ SET_PRE[6:0]	Set the number of delay cells on the feedback clock between the feedback clock and CLK_PRE.
7–4 DLY_CELL_ SET_OUT[6:0]	Set the number of delay cells on the feedback clock between CLK_PRE and CLK_OUT.
DLY_CELL_ SET_POST[6:0]	Set the number of delay cells on the feedback clock between CLK_OUT and CLK_POST.

67.8.26 Vendor Specific Register (uSDHCx_VEND_SPEC)

This register contains the vendor specific control / status register.

Address: Base address + C0h offset



uSDHCx_VEND_SPEC field descriptions

Field	Description
31 CMD_BYTE_EN	Byte access 0 Disable 1 Enable
30 -	Reserved. Always write as 0.

Table continues on the next page...

uSDHC*x*_VEND_SPEC field descriptions (continued)

Field	Description
29 -	Reserved. Always write as 1.
28 -	Reserved. Always write as 0.
27–24 -	Reserved. Always write as 4'b0000.
23–16 INT_ST_VAL[7:0]	Internal State Value Internal state value, reflecting the corresponding state value selected by Debug Select field. This field is read-only and write to this field does not have effect.
15 CRC_CHK_DIS	CRC Check Disable 0 Check CRC16 for every read data packet and check CRC bits for every write data packet 1 Ignore CRC16 check for every read data packet and ignore CRC bits check for every write data packet
14 CARD_CLK_ SOFT_EN	Card Clock Software Enable 0 Gate off the sd_clk 1 Enable the sd_clk
13 IPG_PERCLK_ SOFT_EN	IPG_PERCLK Software Enable 0 Gate off the IPG_PERCLK 1 Enable the IPG_PERCLK
12 HCLK_SOFT_EN	AHB Clock Software Enable **NOTE:** Hardware auto-enables the AHB clock when the internal DMA is enabled even if HCLK_SOFT_EN is 0. 0 Gate off the AHB clock. 1 Enable the AHB clock.
11 IPG_CLK_ SOFT_EN	IPG_CLK Software Enable 0 Gate off the IPG_CLK 1 Enable the IPG_CLK
10 -	Reserved. Always write as 0.
9 -	Reserved. Always write as 0.
8 FRC_SDCLK_ ON	Force CLK output active 0 CLK active or inactive is fully controlled by the hardware. 1 Force CLK active.
7 CLKONJ_IN_ ABORT	Only for debug. Force CLK output active when sending Abort command:

Table continues on the next page...

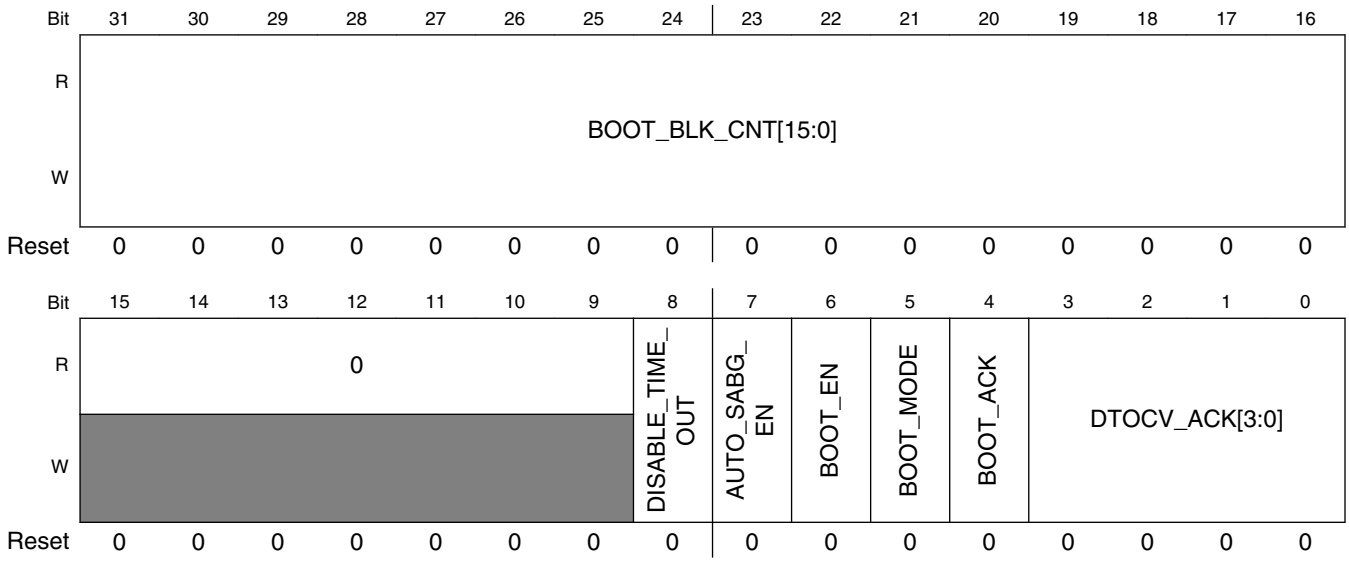
uSDHC*x*_VEND_SPEC field descriptions (continued)

Field	Description
	0 The CLK output is active when sending abort command while data is transmitting even if the internal FIFO is full (for read) or empty (for write). 1 The CLK output is inactive when sending abort command while data is transmitting if the internal FIFO is full (for read) or empty (for write).
6 WP_POL	Only for debug. Polarity of the WP pin: 0 WP pin is high active. 1 WP pin is low active.
5 CD_POL	Only for debug. Polarity of the CD_B pin: 0 CD_B pin is low active. 1 CD_B pin is high active.
4 DAT3_CD_POL	Only for debug. Polarity of DATA3 pin when it is used as card detection. 0 Card detected when DATA3 is high. 1 Card detected when DATA3 is low.
3 AC12_WR_ CHKBUSY_EN	Check busy enable after auto CMD12 for write data packet 0 Do not check busy after auto CMD12 for write data packet 1 Check busy after auto CMD12 for write data packet
2 CONFLICT_ CHK_EN	Conflict check enable. It is not implemented in uSDHC IP. 0 Conflict check disable 1 Conflict check enable
1 VSELECT	Voltage Selection Change the value of output signal VSELECT, to control the voltage on pads for external card. There must be a control circuit out of uSDHC to change the voltage on pads. 1 Change the voltage to low voltage range, around 1.8 V 0 Change the voltage to high voltage range, around 3.0 V
0 EXT_DMA_EN	External DMA Request Enable Enable the request to external DMA. When the internal DMA (either Simple DMA or Advanced DMA) is not in use, and this bit is set, uSDHC will send out DMA request when the internal buffer is ready. This bit is particularly useful when transferring data by Arm platform polling mode, and it is not allowed to send out the external DMA request. By default, this bit is set. 0 In any scenario, uSDHC does not send out external DMA request. 1 When internal DMA is not active, the external DMA request will be sent out.

67.8.27 MMC Boot Register (uSDHC*x*_MMC_BOOT)

This register contains the MMC Fast Boot control register.

Address: Base address + C4h offset



uSDHC*x*_MMC_BOOT field descriptions

Field	Description
31–16 BOOT_BLK_CNT[15:0]	The value defines the Stop At Block Gap value of automatic mode. When received card block cnt is equal to (BLK_CNT - BOOT_BLK_CNT) and AUTO_SABG_EN is 1, then Stop At Block Gap. Here, BLK_CNT is defined in the Block Atrributes Register, bit31 - 16 of 0x04.
15–9 Reserved	This read-only field is reserved and always has the value 0.
8 DISABLE_TIME_OUT	Disable Time Out **NOTE:** When this bit is set, there is no timeout check no matter whether BOOT_EN is set or not. 0 Enable time out 1 Disable time out
7 AUTO_SABG_EN	During boot, enable auto stop at block gap function. This function will be triggered, and host will stop at block gap when received card block cnt is equal to (BLK_CNT - BOOT_BLK_CNT).
6 BOOT_EN	Boot mode enable 0 Fast boot disable 1 Fast boot enable
5 BOOT_MODE	Boot mode select

Table continues on the next page...

uSDHCx_MMC_BOOT field descriptions (continued)

Field	Description
	0 Normal boot 1 Alternative boot
4 BOOT_ACK	Boot ACK mode select 0 No ack 1 Ack
DTOCV_ ACK[3:0]	Boot ACK time out counter value. 0000 SDCLK x 2^13 0001 SDCLK x 2^14 0010 SDCLK x 2^15 0011 SDCLK x 2^16 0100 SDCLK x 2^17 0101 SDCLK x 2^18 0110 SDCLK x 2^19 0111 SDCLK x 2^20 1110 SDCLK x 2^27 1111 SDCLK x 2^28

67.8.28 Vendor Specific 2 Register (uSDHCx_VEND_SPEC2)

This register contains the vendor specific control 2 register.

Address: Base address + C8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				Reserved		0	0	CARD_INT_AUTO_CLR_DIS	TUNING_CMD_EN	TUNING_1bit_EN	TUNING_8bit_EN	CARD_INT_D3_TEST	SDR104_NSD_DIS	SDR104_OE_DIS	SDR104_TIMING_DIS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

uSDHCx_VEND_SPEC2 field descriptions

Field	Description
31–12 -	This field is reserved. Reserved
11–10 -	This field is reserved. Reserved
9 Reserved	This read-only field is reserved and always has the value 0.
8 Reserved	This read-only field is reserved and always has the value 0.
7 CARD_INT_AUTO_CLR_DIS	Disable the feature to clear the Card interrupt status bit when Card Interrupt status enable bit is cleared. Only for debug. 0 Card interrupt status bit (CINT) can be cleared when Card Interrupt status enable bit is 0. 1 Card interrupt status bit (CINT) can only be cleared by writting a 1 to CINT bit.
6 TUNING_CMD_EN	Enable the auto tuning circuit to check the CMD line. 0 Auto tuning circuit does not check the CMD line. 1 Auto tuning circuit checks the CMD line.
5 TUNING_1bit_EN	Enable the auto tuning circuit to check the DATA0 only. It is used with the TUNING_8bit_EN together.
4 TUNING_8bit_EN	Enable the auto tuning circuit to check the DATA[7:0]. It is used with the TUNING_1bit_EN together. **NOTE:** The format of these two bits are [TUNNING_8bit_EN:TUNNING_1bit_EN]. 00 Tuning circuit only checks the DATA[3:0]. 01 Tuning circuit only checks the DATA0. 10 Tuning circuit checks the whole DATA[7:0]. 11 Invalid.
3 CARD_INT_D3_TEST	Card Interrupt Detection Test This bit only uses for debugging. 0 Check the card interrupt only when DATA3 is high. 1 Check the card interrupt by ignoring the status of DATA3.
2 SDR104_NSD_DIS	Interrupt window after abort command is sent. This bit only uses for debugging. 0 Enable the interrupt window 9 cycles later after the end of the I/O abort command (or CMD12) is sent. 1 Enable the interrupt window 5 cycles later after the end of the I/O abort command (or CMD12) is sent.
1 SDR104_OE_DIS	CMD_OE / DATA_OE logic generation test. This bit only uses for debugging. 0 Drive the CMD_OE / DATA_OE for one more clock cycle after the end bit. 1 Stop to drive the CMD_OE / DATA_OE at once after driving the end bit.
0 SDR104_TIMING_DIS	Timeout counter test. This bit only uses for debugging. 0 The timeout counter for Ncr changes to 80, Ncrc changes to 21. 1 The timeout counter for Ncr changes to 72, Ncrc changes to 15.

uSDHCx_VEND_SPEC2 field descriptions (continued)

Field	Description

Chapter 68
Video Data Order Adapter (VDOA)

68.1 Overview

The function of the VDOA is to reorder video data from the "tiled" order used by the VPU to the conventional raster-scan order needed by the IPU .

68.1.1 Block Diagram

The following figure shows the VDOA top level block diagram.

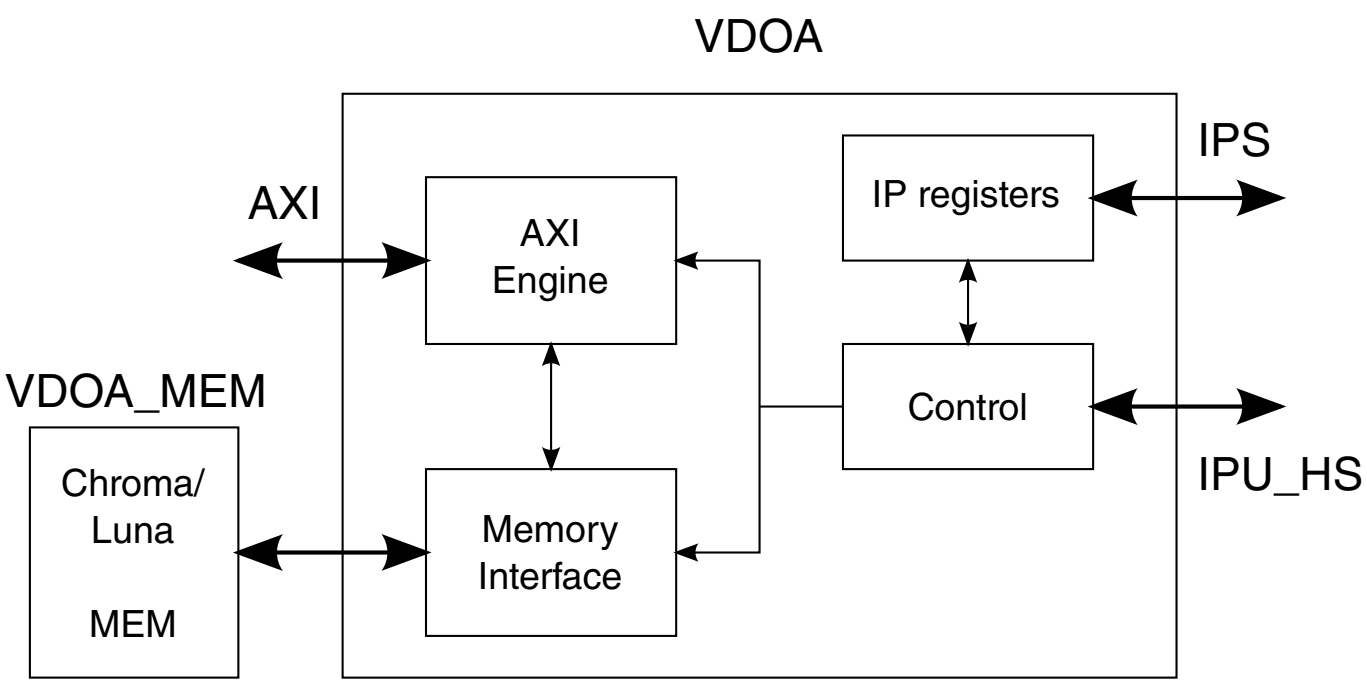


Figure 68-1. VDOA Block Diagram

68.1.2 Features

- Works with 266-MHz clock
- Converts tiled macro-blocks to raster-scan order
- Output formats of IPU data 4:2:0 partially interleaved and 4:2:2 interleaved, format of VPU input data 4:2:0 partially interleaved
- Total rate of 3 or 2.4 pixel/clk, depending on IPU data format (see Data Rate)
- Frame width: even; up to 8192 pixels
- Frame height: even; up to 4096 pixels
- Addresses (bytes)
  - Base address: 32-bit integer, multiple of 8
  - Offset of Chroma buffer from the Lumma buffer : 27-bit integer, multiple of 8
  - IPU stride line: 14-bit integer, multiple of 8
  - VPU stride line: 13-bit integer, multiple of 8
- Each VDOA task is activated individually by the CPU and consists of reordering a single buffer of pixels; or reordering concurrently three buffers of pixels (needed for the three input fields used for de-interlacing in the IPU).
- All task parameters are double buffered to allow software-controlled frame-by-frame task switching without reconfiguration overhead.
- Ability to decode full frame continuously or a band (a few lines) at a time, synchronized with the IPU using double buffer for output data
- Ability to work with one of two IPU units
- Interrupt on task completion and on AXI transfer error

68.2 Clocks

The table found here describes the clock sources for VDOA.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 68-1. VDOA Clocks

Clock name	Clock Root	Description
ipg_clk_s	vdo_axi_clk_root	IP bus clock domain. Controls VDOA registers read/write function.
vdoa_clk	vdo_axi_clk_root	VDOA functional clock. Maximum frequency of 266MHz. The actual value of the core clock is dependent upon the maximum use case. Clock frequency scaling will be done through the VPU API.

68.3 Functional Description

This section provides a complete functional description of the block.

68.3.1 Memory organization

68.3.1.1 Frame organization - VPU

The data of the VPU will always be in YUV 4:2:0 partially interleaved format.

Data is organized in macro blocks which correspond to 16x16 pixel blocks in the frame. Each macro block is divided into two 8x16 blocks.

A cluster of four pixels in 4:2:0 format is composed of four luma values (Y, 1 byte) and one chroma value (Cr, Cb, 2 bytes). In partially interleaved format, the luma and chroma exist on sperate buffers. The chroma buffer is 16x8 bytes in size.

The following table shows an example of how data is arranged in a frame and in blocks:

Table 68-2. FW*FH frame Luma buffer Example

0/0-7	0/8-15	0/16-23	0/24-31	...	0/16x-16x+7	0/16x+8-16x+15	...	0/FW-16-FW-9	0/FW-8-FW-1
1/0-7	1/8-15	1/16-23	1/24-31	...	1/16x-16x+7	1/16x+8-16x+15	...	1/FW-16-FW-9	1/FW-8-FW-1
...	...	...	...	...	...	...	...	...	...
15/0-7	15/8-15	15/16-23	15/24-32	...	15/16x-16x+7	15/16x+8-16x+15	...	15/FW-16-FW-9	15/FW-8-FW-1
16/0-7	16/8-15	16/16-23	16/24-31	...	16/16x-16x+7	16/16x+8-16x+15	...	16/FW-16-FW-9	16/FW-8-FW-1
17/0-7	17/8-15	17/16-23	17/24-31	...	17/16x-16x+7	17/16x+8-16x+15	...	17/FW-16-FW-9	17/FW-8-FW-1
...	...	...	...	...	...	...	...	...	...
31/0-7	31/8-15	31/16-23	31/24-32	...	31/16x-16x+7	31/16x+8-16x+15	...	31/FW-16-FW-9	31/FW-8-FW-1
...	...	...	...	...	...	...	...	...	...
FH-16/0-7	FH-16/8-15	FH-16/16-23	FH-16/24-31	...	FH-16/16x-16x+7	FH-16/16x+8-16x+15	...	FH-16/FW-16-FW-9	FH-16/FW-8-FW-1
FH-15/0-7	FH-15/8-15	FH-15/16-23	FH-15/24-31	...	FH-15/16x-16x+7	FH-15/16x+8-16x+15	...	FH-15/FW-16-FW-9	FH-15/FW-8-FW-1
...	...	...	...	...	...	...	...	...	...
FH-1/0-7	FH-1/8-15	FH-1/16-23	FH-1/24-31	...	FH-1/16x-16x+7	FH-1/16x+8-16x+15	...	FH-1/FW-16-FW-9	FH-1/FW-8-FW-1

Legend -> row/Start pixel - End pixel

68.3.1.2 Frame organization - IPU

The IPU support various frame organization (or pixel format). VDOA supports only two of these formats: 4:2:0 partially interleaved (similar to VPU) and 4:2:2 interleaved.

68.3.2 Conversion types

The VDOA supports two types of data conversion:

1. VPU 4:2:0 partially interleaved -> IPU 4:2:0 partially interleaved
2. VPU 4:2:0 partially interleaved -> IPU 4:2:2 interleaved

NOTE

For all conversions the VPU format data is in tiled order and the IPU data is in raster order.

68.3.3 IPU synchronization

The synchronization with the IPU is described in /ipu/map_ipu.xml.

The synchronization between VDOA and IPU executed on band resolution (a band consists of several rows, depending on IPU's IDMAC configuration). Up to two band buffers are supported for this synchronization.

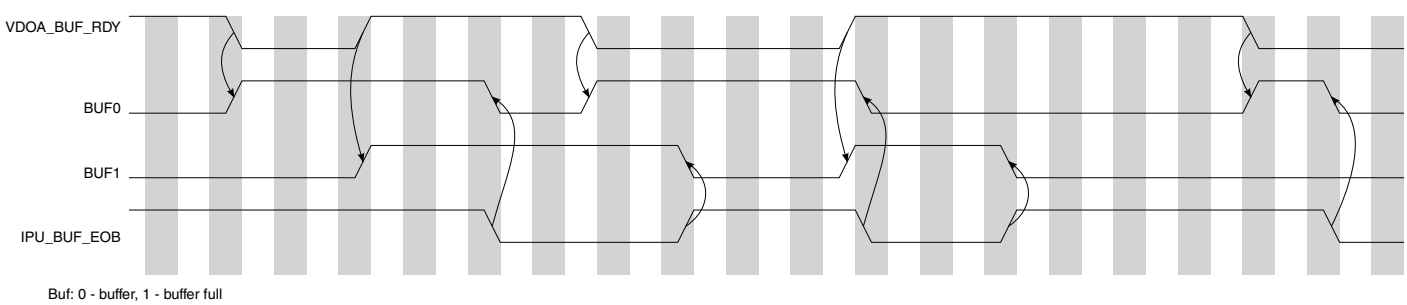


Figure 68-2. Toggling mechanism and buffer indication

68.3.4 Double buffering

All configuration registers (all registers excluding IVDOAE, VDOAIS, VDOAST, and VDOATD) are double buffered. This means that on start, their content is copied to internal registers.

Writing new settings to VDOA's configuration registers will affect the following transfer.

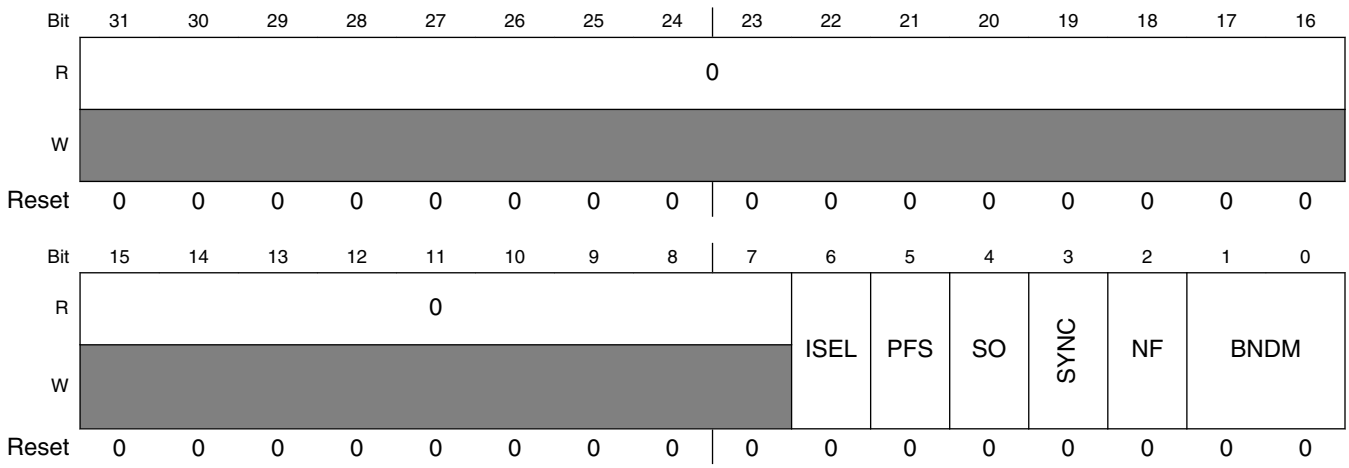
68.4 VDOA Memory Map/Register Definition

VDOA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
21E_4000	VDOA Control Register (VDOA_VDOAC)	32	R/W	0000_0000h	68.4.1/5666
21E_4004	VDOA Start and Reset (VDOA_VDOASRR)	32	R/W	0000_0000h	68.4.2/5667
21E_4008	VDOA Interrupt Enable Register (VDOA_VDOAIE)	32	R/W	0000_0000h	68.4.3/5668
21E_400C	VDOA Interrupt Status Register (VDOA_VDOAIST)	32	w1c	0000_0000h	68.4.4/5668
21E_4010	VDOA Frame Parameters Register (VDOA_VDOAFP)	32	R/W	0000_0000h	68.4.5/5669
21E_4014	VDOA IPU External Buffer 0 Frame 0 Address Register (VDOA_VDOAIEBA00)	32	R/W	0000_0000h	68.4.6/5669
21E_4018	VDOA IPU External Buffer 0 Frame 1 Address Register (VDOA_VDOAIEBA01)	32	R/W	0000_0000h	68.4.7/5670
21E_401C	VDOA IPU External Buffer 0 Frame 2 Address Register (VDOA_VDOAIEBA02)	32	R/W	0000_0000h	68.4.8/5670
21E_4020	VDOA IPU External Buffer 1 Frame 0 Address Register (VDOA_VDOAIEBA10)	32	R/W	0000_0000h	68.4.9/5670
21E_4024	VDOA IPU External Buffer 1 Frame 1 Address Register (VDOA_VDOAIEBA11)	32	R/W	0000_0000h	68.4.10/ 5671
21E_4028	VDOA IPU External Buffer 1 Frame 2 Address Register (VDOA_VDOAIEBA12)	32	R/W	0000_0000h	68.4.11/ 5671
21E_402C	VDOA IPU Stride Line Register (VDOA_VDOASL)	32	R/W	0000_0000h	68.4.12/ 5672
21E_4030	VDOA IPU U (Chroma) Buffer Offset Register (VDOA_VDOAIUBO)	32	R/W	0000_0000h	68.4.13/ 5672
21E_4034	VDOA VPU External Buffer 0 Address Register (VDOA_VDOAVEBA0)	32	R/W	0000_0000h	68.4.14/ 5673
21E_4038	VDOA VPU External Buffer 1 Address Register (VDOA_VDOAVEBA1)	32	R/W	0000_0000h	68.4.15/ 5673
21E_403C	VDOA VPU External Buffer 2 Address Register (VDOA_VDOAVEBA2)	32	R/W	0000_0000h	68.4.16/ 5673
21E_4040	VDOA VPU U (Chroma) Buffer Offset Register (VDOA_VDOAVUBO)	32	R/W	0000_0000h	68.4.17/ 5674
21E_4044	VDOA Status Register (VDOA_VDOASR)	32	R	0000_0000h	68.4.18/ 5675

68.4.1 VDOA Control Register (VDOA_VDOAC)

Address: 21E_4000h base + 0h offset = 21E_4000h



VDOA_VDOAC field descriptions

Field	Description
31–7 Reserved	This read-only field is reserved and always has the value 0.
6 ISEL	IPU SELECT - determines in sync mode which of the two sets of hand shake pins is used 0 **vdoa_buf_rdy_and_ipu_buf_eob_0** — Use vdoa_buf_rdy[0] and ipu_buf_eob[0] 1 **vdoa_buf_rdy_and_ipu_buf_eob_1** — Use vdoa_buf_rdy[1] and ipu_buf_eob[1]
5 PFS	Pixel Format Select - Pixel format of data written to / read from IPU. Note Data from VPU is always assumed to have partial interleaved 4:2:0 format 0 **4_2_0** — partialy interleaved 4:2:0 1 **4_2_2** — interleaved 4:2:2 Y1U1Y2V1
4 SO	Scan Order 0 **PROGRESSIVE** — Scan order is progressive 1 **INTERLACED** — Scan order is interlaced
3 SYNC	SYNC MODE - defines whether the VDOA will transfer a full frame (or 2 frames) continuously or will transfer a band at a time and wait for IPU signal to continue 0 **NO_SYNC_MODE** — None SYNC mode (default) 1 **SYNC_MODE** — Sync mode
2 NF	Number of frames - Determines whether to transfer 1 frame or three frames 0 **1_FRAME** — One frame (default) 1 **3_FRAMES** — Three frames
BNDM	BNDM Band Size 00 **BAND_HEIGHT_8** — Band height = 8 lines.- Supported only for interlaced scan (SO=1) 01 **BAND_HEIGHT_16** — Band height = 16 lines.

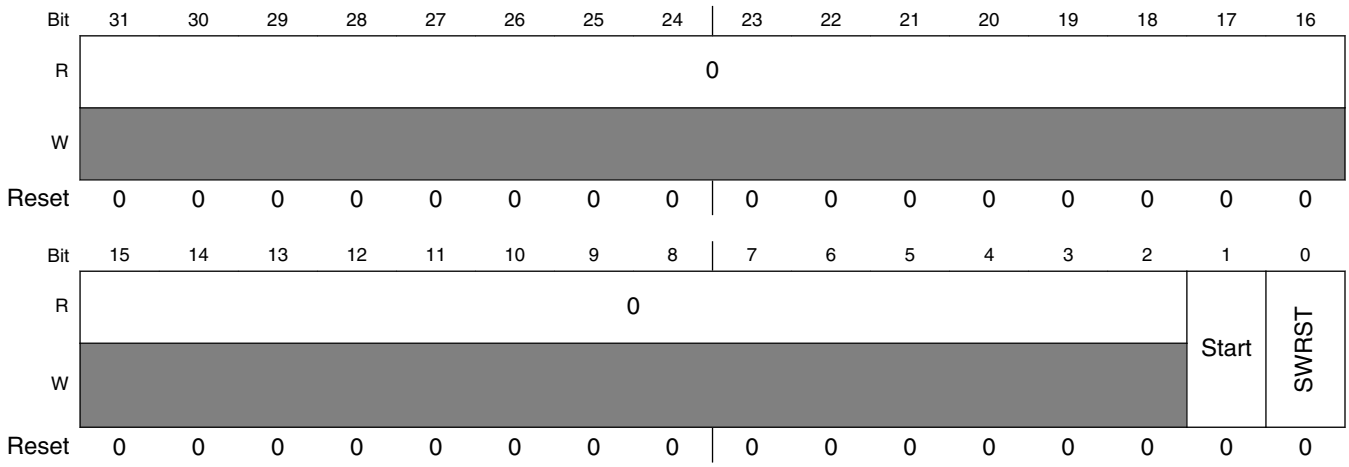
Table continues on the next page...

VDOA_VDOAC field descriptions (continued)

Field	Description
	10 **BAND_HEIGHT_32** — Band height = 32 lines. 11 reserved

68.4.2 VDOA Start and Reset (VDOA_VDOASRR)

Address: 21E_4000h base + 4h offset = 21E_4004h

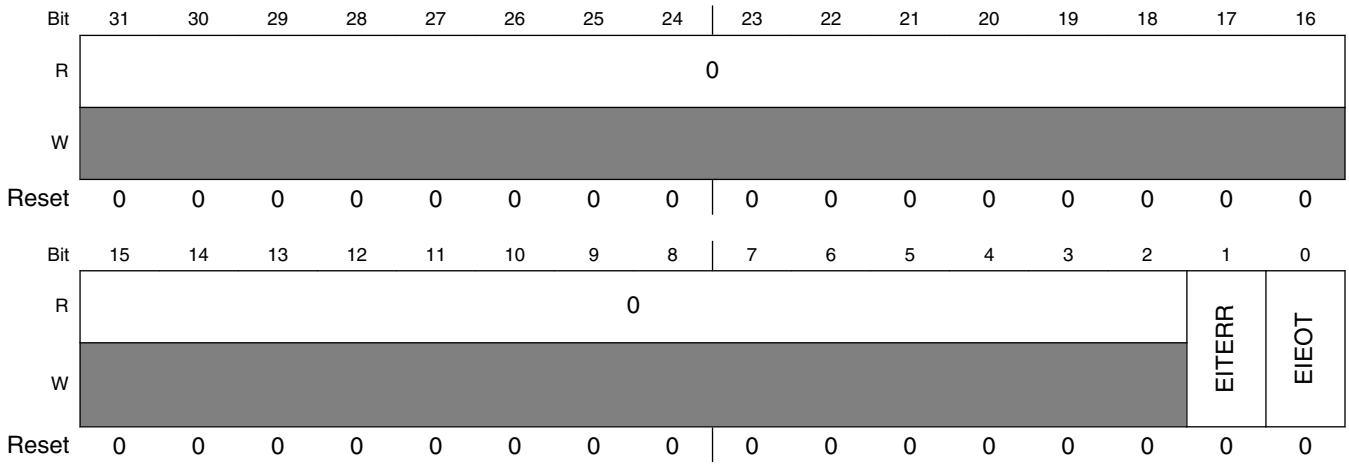


VDOA_VDOASRR field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
1 Start	Start Transfer - Strat a VDOA data transfer according to all parameters. **Note:** During run this bit is read only In IDLE - 0 Ignored 1 **START_TRANSFER** — Start a new transfer All registers we copied internally so any write to them will take place only in next transfer (double buffer)
0 SWRST	Software reset - Finish outstanding AXI transfer and reset all internal registers the configuration registers are mnot cleared

68.4.3 VDOA Interrupt Enable Register (VDOA_VDOAIE)

Address: 21E_4000h base + 8h offset = 21E_4008h

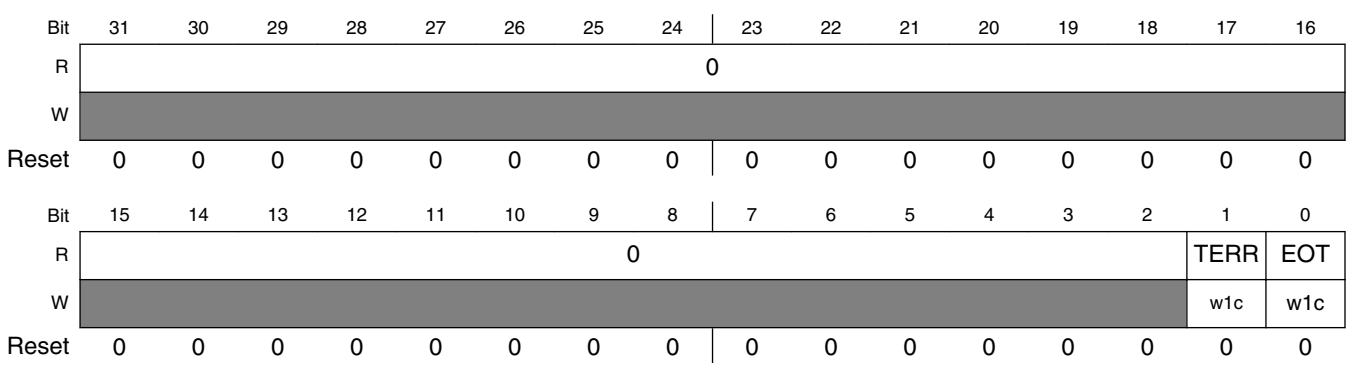


VDOA_VDOAIE field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
1 EITERR	EITERR - Enable Interrupt Transfer access Error - Enables Interrupt on AXI access Error 0 **IRQ_DISABLED** — interrupt disable (default) 1 **IRQ_ENABLED** — Interrupt Enabled
0 EIEOT	EIEOT - Enable Interrupt End Of Transfer- Enables Interrupt on end of transfer 0 **IRQ_DISABLED** — interrupt disable (default) 1 **IRQ_ENABLED** — Interrupt Enabled

68.4.4 VDOA Interrupt Status Register (VDOA_VDOAIST)

Address: 21E_4000h base + Ch offset = 21E_400Ch

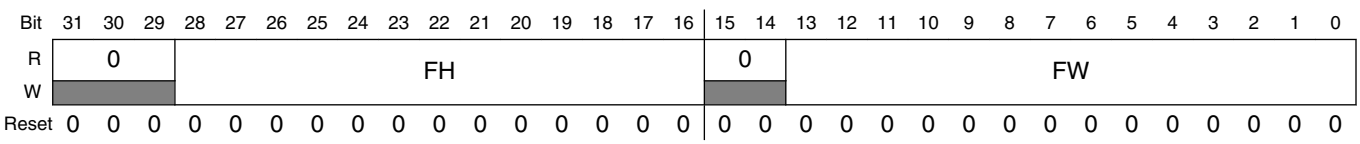


VDOA_VDOAIST field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
1 TERR	Axi Access had an access error see ERRW bit in 0XBASE_0044 (VDOASR) for type of access (read write) if EITERR is set an interrupt will be generated
0 EOT	End Of transfer - Transfer was completed if EIEOT is set an interrupt will be generated

68.4.5 VDOA Frame Parameters Register (VDOA_VDOAFP)

Address: 21E_4000h base + 10h offset = 21E_4010h

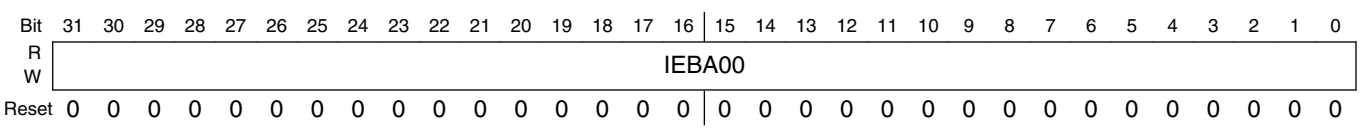


VDOA_VDOAFP field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28–16 FH	Number of pixels in one column, of the frame. Note the 3 LSB are RO and will always be 0 (multiply of 8)
15–14 Reserved	This read-only field is reserved and always has the value 0.
FW	Number of pixels in one row, of the frame. Note the3 LSB are RO and will always be 0 (multiply of 8)

68.4.6 VDOA IPU External Buffer 0 Frame 0 Address Register (VDOA_VDOAIEBA00)

Address: 21E_4000h base + 14h offset = 21E_4014h



VDOA_VDOAIEBA00 field descriptions

Field	Description
IEBA00	External Address of Frame 0 output (IPU) buffer 0 - Note that the 3 LSB are always 0 (aligned to 8 address) Used for all transfer types

68.4.7 VDOA IPU External Buffer 0 Frame 1 Address Register (VDOA_VDOAIEBA01)

Address: 21E_4000h base + 18h offset = 21E_4018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IEBA01																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDOA_VDOAIEBA01 field descriptions

Field	Description
IEBA01	External Address of Frame 1 output (IPU) buffer 0 - Note that the 3 LSB are always 0 (aligned to 8 address) Used when transfering 3 frames (NF=1) only

68.4.8 VDOA IPU External Buffer 0 Frame 2 Address Register (VDOA_VDOAIEBA02)

Address: 21E_4000h base + 1Ch offset = 21E_401Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IEBA02																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDOA_VDOAIEBA02 field descriptions

Field	Description
IEBA02	External Address of Frame 2 output (IPU) buffer 0 - Note that the 3 LSB are always 0 (aligned to 8 address) Used when transfering 3 frames (NF=1) only

68.4.9 VDOA IPU External Buffer 1 Frame 0 Address Register (VDOA_VDOAIEBA10)

Address: 21E_4000h base + 20h offset = 21E_4020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IEBA10																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDOA_VDOAIEBA10 field descriptions

Field	Description
IEBA10	External Address of Frame 0 output (IPU) buffer 1 - Note that the 3 LSB are always 0 (aligned to 8 address) Used in sync mode (SYNC=1) only

68.4.10 VDOA IPU External Buffer 1 Frame 1 Address Register (VDOA_VDOAIEBA11)

Address: 21E_4000h base + 24h offset = 21E_4024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IEBA11																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDOA_VDOAIEBA11 field descriptions

Field	Description
IEBA11	External Address of Frame 1 output (IPU) buffer 1 - Note that the 3 LSB are always 0 (aligned to 8 address) This register is used only in sync mode (SYNC=1), 3 frames transfer (NF=1)

68.4.11 VDOA IPU External Buffer 1 Frame 2 Address Register (VDOA_VDOAIEBA12)

Address: 21E_4000h base + 28h offset = 21E_4028h

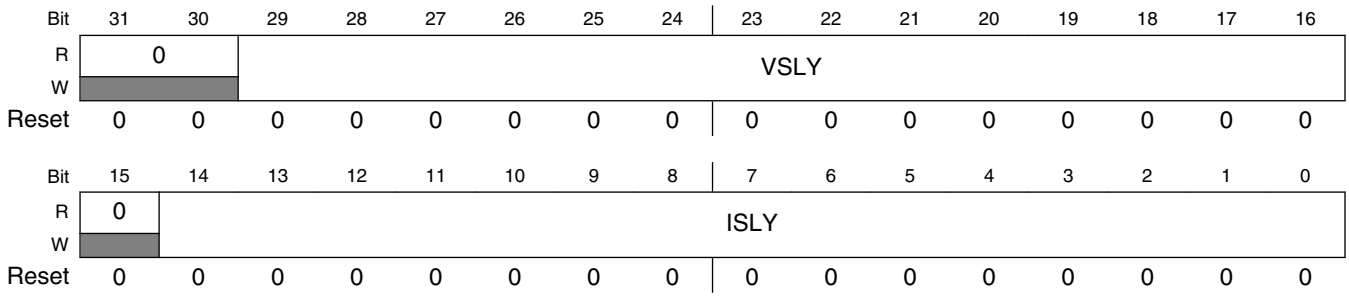
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IEBA12																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDOA_VDOAIEBA12 field descriptions

Field	Description
IEBA12	External Address of Frame 2 output (IPU) buffer 1 - Note that the 3 LSB are always 0 (aligned to 8 address) This register is used only in sync mode (SYNC=1), 3 frames transfer (NF=1)

68.4.12 VDOA IPU Stride Line Register (VDOA_VDOASL)

Address: 21E_4000h base + 2Ch offset = 21E_402Ch

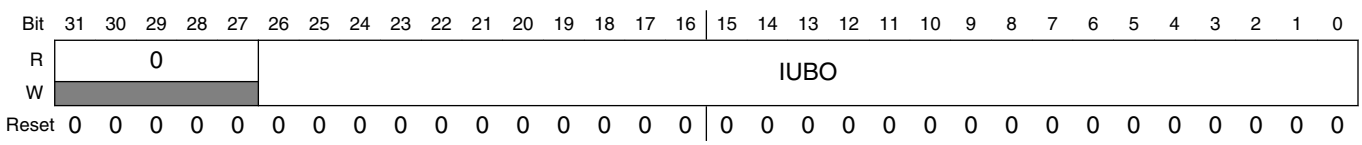


VDOA_VDOASL field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value 0.
29–16 VSLY	VPU Stride Line - Address vertical scaling factor in bytes for memory access. Also number of maximum bytes in the "Y" component row according to memory limitations.
15 Reserved	This read-only field is reserved and always has the value 0.
ISLY	IPU Stride Line - Address vertical scaling factor in bytes for memory access; also number of maximum bytes in the "Y" component row.- Note for 4:2:2 format ISLY will be doubled since each pixel consists of two bytes.

68.4.13 VDOA IPU U (Chroma) Buffer Offset Register (VDOA_VDOAIUBO)

Address: 21E_4000h base + 30h offset = 21E_4030h



VDOA_VDOAIUBO field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
IUBO	The offset of Chroma (UV) Buffer for all IPU output frames i.e Buffer Chroma address will be VDOAIEBAnm+VDOAIUBO - Note that the 3 LSB are always 0 (aligned to 8 address). This parameter is used only for PFL = 4:2:0

68.4.14 VDOA VPU External Buffer 0 Address Register (VDOA_VDOAVEBA0)

Address: 21E_4000h base + 34h offset = 21E_4034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VEBA0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDOA_VDOAVEBA0 field descriptions

Field	Description
VEBA0	Address of Frame 0 VPU buffers - Note that the 3 LSB are always 0 (aligned to 8 address) Used for all transfers

68.4.15 VDOA VPU External Buffer 1 Address Register (VDOA_VDOAVEBA1)

Address: 21E_4000h base + 38h offset = 21E_4038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VEBA1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDOA_VDOAVEBA1 field descriptions

Field	Description
VEBA1	Address of Frame 1 VPU buffers - Note that the 3 LSB are always 0 (aligned to 8 address) Used when transfering three frame (NF=1) only

68.4.16 VDOA VPU External Buffer 2 Address Register (VDOA_VDOAVEBA2)

Address: 21E_4000h base + 3Ch offset = 21E_403Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VEBA2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDOA_VDOAVEBA2 field descriptions

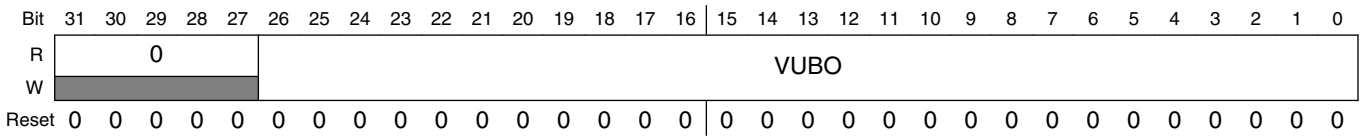
Field	Description
VEBA2	Address of Frame 2 VPU buffers - Note that the 3 LSB are always 0 (aligned to 8 address)

VDOA_VDOAVEBA2 field descriptions (continued)

Field	Description
	Used when transfering three frame (NF=1) only

68.4.17 VDOA VPU U (Chroma) Buffer Offset Register (VDOA_VDOAVUBO)

Address: 21E_4000h base + 40h offset = 21E_4040h

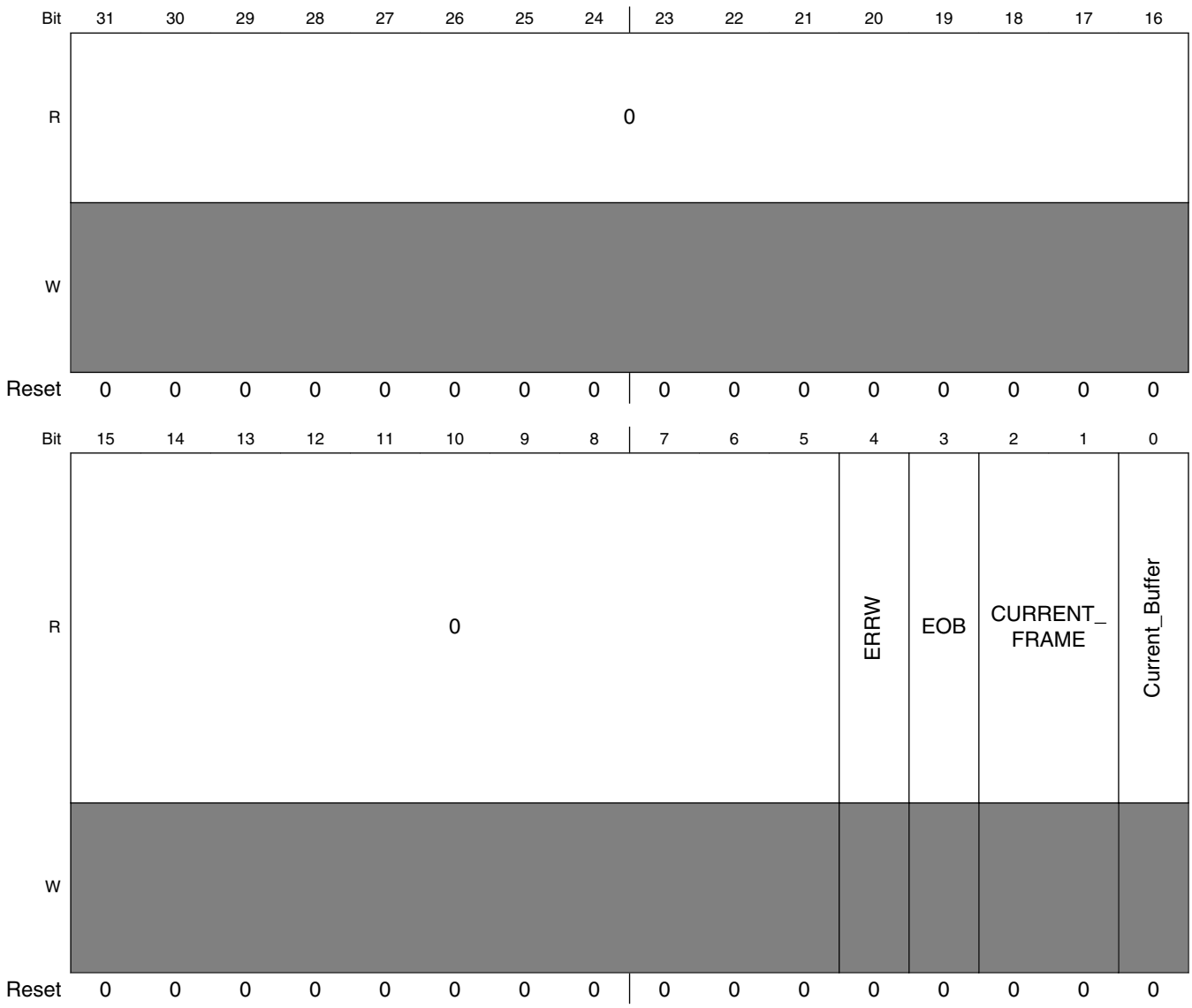


VDOA_VDOAVUBO field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
VUBO	The offset of Chroma (UV) Buffer for all VPU input frames i.e Chroma Buffer address will be VDOAVEBAm+VDOAVUBO - Note that the 3 LSB are always 0 (aligned to 8 address)

68.4.18 VDOA Status Register (VDOA_VDOASR)

Address: 21E_4000h base + 44h offset = 21E_4044h



VDOA_VDOASR field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
4 ERRW	Error Write - Indicates that the last access that failed was a read or a write.This field is valid only when TERR bit is set in VDOA Interrupt Status Register - VDOAIST 0 **READ_ERROR** — Read Error 1 **WRITE_ERROR** — Write Error

Table continues on the next page...

VDOA_VDOASR field descriptions (continued)

Field	Description
3 EOB	End of Band- Indicates that the VDOA has finished transferring a band in SYNC mode and is waiting for IPU to continue.
2–1 CURRENT_ FRAME	Current Frame - When working on 3 frames the index of the frame currently transferred
0 Current_Buffer	Current Buffer - for Double buffer shows the index of the buffer currently transferred

Chapter 69
Video Processing Unit (VPU)

69.1 Overview

Video Processing Unit of i.MX is a high performance multi-standard video codec

which can decode H.264 BP/MP/HP, VC-1 SP/MP/AP, MPEG-4 SP/ASP, H.263 P0/P3, MPEG-1/2 MP, Divx (Xvid) HP/PP/HTP/HDP, RV8/9/10, VP8 (1280x720), AVS, H.264-MVC (1280x720), MJPEG BP (max. 8192x8192) up to full-HD 1920x1088 @30fps. It can also perform H.264 BP/CBP, MPEG-4 SP, H.263 P0/P3, MJPEG BP (max. 8192x8192) encoding up to @30fps. VPU can support encode or decode multiple video clips with multiple standards simultaneously.

NOTE

RealNetworks video codec is disabled by default on i.MX 6 series processors. Contact your NXP sales representative for more details.

VPU has two bus interface: IP bus for register access controlling and AXI bus for data throughput.

VPU makes use of external memory space for bitstream buffer, parameter buffer, bit code buffer, working buffer and frame buffer.

VPU has an embedded BIT processor controlling internal video processing sub-blocks and communicating with host processor through host interface. Very low resources of the ARM platform is required to support VPU. Host processor only need to access VPU registers for initializing VPU or setting parameters during frame gap. In application, it is done through VPU API called by host processor. Large part of video codec (except BIT processor) is optimally shared which enables to achieve low power and low gate count.

VPU combined with processing capabilities of Image Processing Unit (IPU) can support high quality video processing applications for the chip.

The following figure shows VPU top-level diagram.

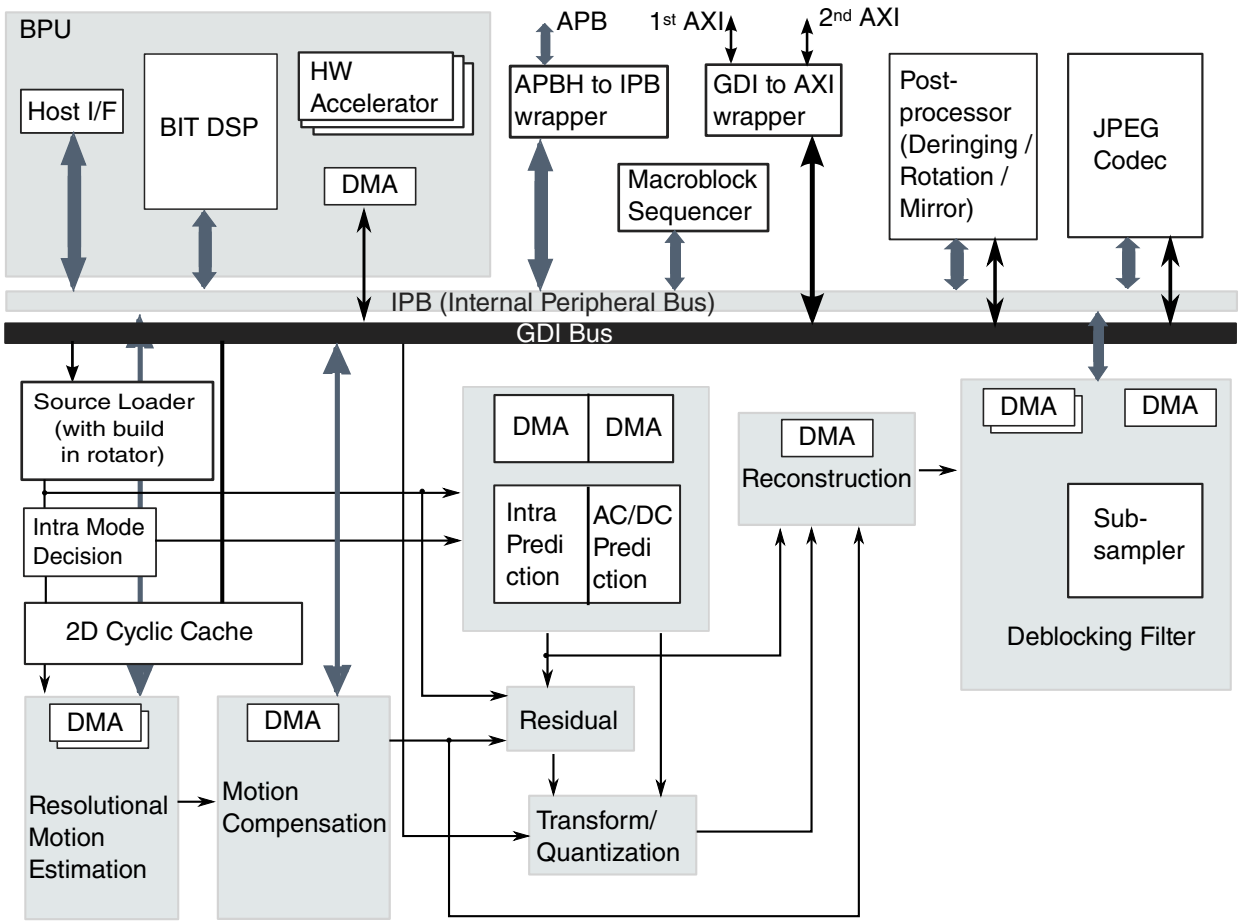


Figure 69-1. VPU Block Diagram

69.1.1 Features

VPU supports H.264 BP/MP/HP, VC-1 SP/MP/AP, MPEG-4 SP/ASP, H.263 P0/P3, MPEG-1/2 MP/HP, DivX (Xvid) HP/PP/HTP/HDP, RV8/9/10, MJPEG BP decoding and H.264 BP/CBP, MPEG-4 SP, H.263 P0/P3, MJPEG BP encoding. The supported resolution is up to full-HD 1920x1088@30fps for decoding and @30fps for encoding except MJPEG codec (up to 8192x8192). VPU can support various error resilience tools and also support multiple decoding and full duplex multi-party-call simultaneously.

And the VPU provides programmability, flexibility and ease of upgrade in decoding/encoding or host interface because all the controls in decoding/encoding process and host interface are implemented as firmware in a programmable BIT processor.

NOTE

The capabilities of VPU listed here is measured with a nearly ideal bus and memory system. But in real SOC, the capabilities of VPU are also depends on the bus and memory system of SOC.

The detailed feature of VPU is as follows:

Table 69-1. Supported Decoding/Encoding Standards

	Standard	Profile	Level	Resolution
Encoder	H.264	BP/CBP	4.0	1920x1088
	MPEG-4	SP	5/6	1920x1088
	H.263	P0/P3	70	1920x1088
	MJPEG	BP		8192x8192
Decoder	H.264	BP/MP/HP	4.1	1920x1088
	MPEG-4	SP/ASP		1920x1088
	H.263		70	1920x1088
	VC-1	SP/MP/AP		1920x1088
	MPEG-2	MP/HP		1920x1088
	DivX/Xvid	HP/PP/HTP/HDP		1920x1088
	RV	8/9/10		1920x1088
	H.264-MVC	BP/MP/HP		1280x720
	MJPEG	BP		8192x8192

- Multi-standard video codec
  - H.264/AVC encoder for baseline profile
  - MPEG-4 encoder for simple profile
  - H.263 encoder for baseline profile
  - MJPEG encoder for baseline profile
  - H.264/AVC decoder for baseline profile, main profile and high profile
  - VC-1 decoder for simple profile, main profile and advanced profile
  - MPEG-4 decoder for simple profile, advanced simple profile except GMC
  - H.263 decoder for baseline profile
  - DivX decoder for
  - MPEG-2 decoder for main profile @ main and high level
  - RV decoder for profile 8/9/10
  - MJPEG decoder for Baseline profile
- Other features
  - Pre/Post rotator and mirror
  - Built-in de-ringing filter.
  - Built-in de-blocking filter for MPEG-2/MPEG-4/DivX.
  - MPEG-2 encoder partial acceleration.

- VPU supports decoding/encoding processes simultaneously. Each process can have a different format.
- Robust error detection/concealment.

- Programmability
  - Embeds 16-bit BIT processor that is dedicated to processing bitstream and controlling the hardware.
  - General purpose registers and interrupt generation for communication between host processor and VPU.

69.2 Modes of Operation

The VPU has two kinds of operating modes: normal operating mode and low power mode.

69.2.1 Normal Operating Mode

Normal operating mode is the video codec processing mode, which can be MPEG-4 encoding/decoding, DivX decoding, H.264 encoding/decoding, VC-1 decoding, or multiple bitstream encoding/decoding in multiple standards simultaneously.

VPU supports decoding/encoding processes simultaneously. Each process can have a different format. The host processor creates and executes the specified process by sending parameters and commands to the BIT processor through the VPU API.

69.2.2 Low Power Mode

When VPU is in idle state, VPU interface signal vpu_idle is asserted and system can gate off its clock source according to this signal.

If the clock source is gated off and system wants to wake it up, it just re-enable the clock and send command to VPU. If power is off when the system wants to wake it up, VPU has to be re-initialized. Refer to section Detailed Signal Descriptions for detailed description about vpu_idle signal.

69.3 Clocks

There are four clock domains in VPU.

The following table describes the clock sources for VPU.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 69-2. VPU Clocks

Clock name	Clock Root	Description
aclk	vpu_axi_clk_root	AXI bus clock domain. Controls all AXI bus related functions. Maximum frequency is 266 MHz @ TSMC 40nm process. The aclk derives from the same clock as other blocks performing AXI bus access in the chip, the frequency is determined by the whole system performance requirement.
cclk	vpu_axi_clk_root	VPU functional clock. Controls all VPU decoding functionality, with a maximum frequency of 328 MHz. The actual value of the core clock is dependent upon the maximum use case. Clock frequency scaling will be done through the VPU API.
ipg_clk_s	ipg_clk_root	IP bus clock domain. Controls VPU registers read/write function, with a maximum frequency of 66.5 MHz. ipg_clk_s is a gated clock of IP green-line clock (ipg_clk) with ips_module_en, it is turned off when there is no registers read/write, for power saving.
rclk	video_27m_clk_root	Reference clock domain. Used as reference clock for vpu_idle related logic.

VPU has all four clock domains because it involves in AXI signal generation, functional calculation, IP bus configuration, vpu_idle control logic. Only positive edge clocks is used in VPU design. Each clock is fully asynchronous with other clocks and separated from other clocks.

All clocks can be gated off when VPU is in the idle state to reduce power consumption. There is no restriction about the value of aclk and cclk. The actual frequency of aclk and cclk affects bus bandwidth and video decoding processing capability.

69.4 Functional Description

VPU is a high-performance multi-standard video processing block that supports up to full HD 1920 x 1088 at 30 fps decoding and at 30 fps encoding.

69.4.1 VPU Architecture

VPU is a high performance multi-standard video codec that can decode

- H.264 BP/MP/HP
- VC-1 SP/MP/AP
- MPEG-4 SP/ASP
- H.263 P0/P3

- MPEG-1/2 MP/HP
- Divx (Xvid) HP/PP/HTP/HDP
- RV8/9/10
- MJPEG BP (max. 8192x8192) up to full-HD 1920 x 1088 at 30 fps

It can also perform H.264 BP, MPEG-4 SP, H.263 P0/P3, and MJPEG BP (max. 8192x8192) encoding up to . VPU can support encode or decode multiple video clips with multiple standards simultaneously.

It connects with the system via the 32-bit IP bus for system control and 64-bit AMBA3 AXI for data throughput. It uses on-chip memories to achieve high performance.

VPU has a 16-bit DSP called a BIT processor. The BIT processor controls the internal video codec sub blocks and communicates with a host processor through the host interface. The required resource for the host ARM platform to control the VPU is very low, under 1 MIPS, because all functions such as rate control, FMO, ASO, video codec control and error resilience are implemented in the BIT processor. The majority of sub-blocks in VPU are optimally shared, which enables the ultra low power and low gate count.

VPU mainly consists of an embedded 16-bit BIT processor, video codec hardware, and bus arbiter/interface. Refer to Figure 1 for the block diagram of VPU.

69.4.1.1 Embedded BIT processor

The embedded BIT processor is used for parsing or forming bitstreams. It includes some hardware accelerators to speed up bitstream processing.

In addition to handling bitstreams, the BIT processor controls the video decoding hardware and communicates with host processor through IP bus and AXI bus interface.

69.4.1.2 Video CODEC Hardware

All video decoding processing, except handling coefficients for VLD, are implemented in hardware.

The video codec hardware is designed to reduce logic gate count by sharing parts of sub-blocks for multi-standard video decoding.

It mainly includes the following hardware components.

69.4.1.2.1 Inter-Predictor

The Inter-Predictor sub-block uses a reconstructed motion vector that represents the displacement between the block currently being decoded and the corresponding location in the reference frame, to calculate interpolated pixel data for motion compensation.

69.4.1.2.2 AC/DC and Intra-Predictor

There are two intra-prediction sub-blocks in the video decoding hardware. One is for the MPEG-4/H.263 P3/VC-1 AC/DC prediction and another for the H.264 intra-prediction.

In case of VC-1 decoding, the hardware prediction mode decision is used. It brings high performance and low power consumption. The coefficient data of decoding is re-ordered automatically based on the detected prediction mode. For H.264 intra-prediction mode in decoding, hardware mode decision or software based made decision is used.

69.4.1.2.3 Inverse transform/Inverse quantization

VPU has only one transform/quantization sub-block. It operates in several types of inverse transform/quantization for MPEG-2, VC-1 and H.264.

Each inverse transform/quantization for decoding uses different coefficients and different data processes, but it uses same control and data flow in many cases.

69.4.1.2.4 De-blocking/Overlap-smoothing filter

>De-blocking filter removes blocking artifacts resulted from quantization, different motion vectors. The filter processing is applied in both decoder and encoder.

The VPU de-blocking filter supports H.264/H.263/MPEG4/VC-1. Overlap-smoothing filter supports VC-1 overlap-smoothing feature. For H.264, H.263 and VC-1, the de-blocking filter operates within coding loop. Filtered frames are used as reference frames for motion compensation of subsequent coded frames. But for MPEG4, the de-blocking filter operates outside coding loop for only display.

69.4.1.2.5 Coefficient buffer interface

The coefficient buffer interface provides a channel for BIT processor to read quantized coefficients resulted from encoding process or to send variable-length-decoded coefficients to the video codec for decoding process.

69.4.1.2.6 Macroblock controller

VPU has a complex and large number of pipeline for high-performance. To manage it wholly by BIT processor is not suitable, so VPU embeds the macroblock controller to control all sub-blocks of the video codec based on the configuration of pipelining by BIT processor. This scheme reduces the load on BIT processor and guarantees the programmability of the block. Before the video codec encodes or decodes a macroblock, BIT processor configures how the pipeline of the codec is structured. If all processes are completed for encoding/decoding a macroblock, the macro-block controller indicates its completion.

In summary, BIT processor configures which sub-blocks are enabled for current macroblock processing, and the macroblock controller manages corresponding sub-blocks based on the configuration.

The figure below shows the connectivity of macroblock controller.

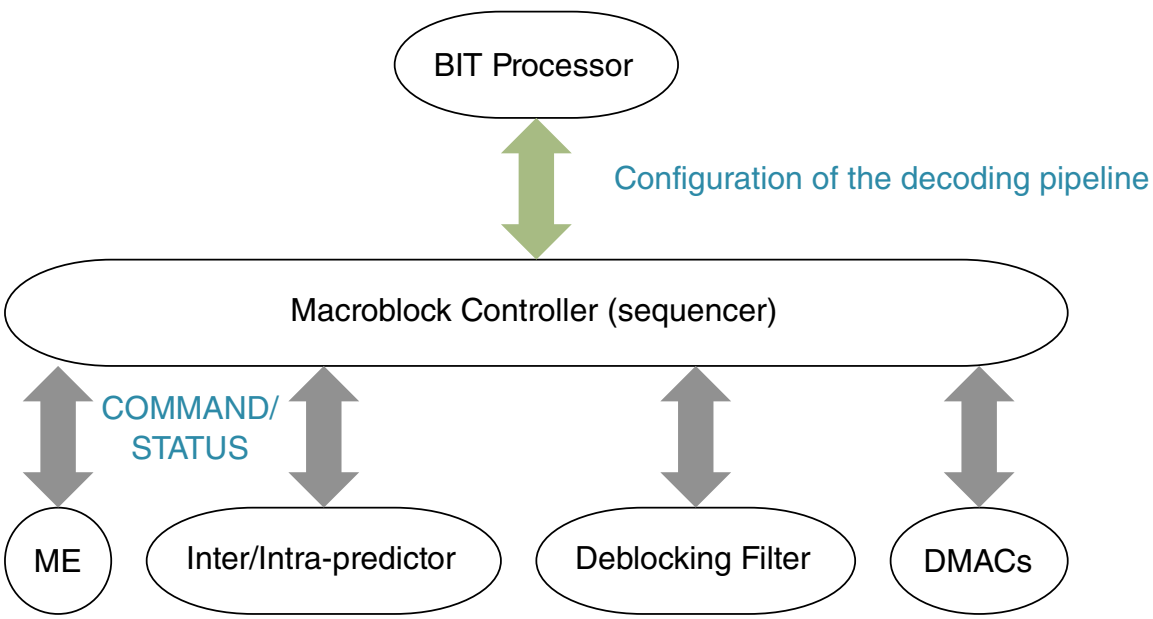


Figure 69-2. The connectivity of macroblock controller

69.4.1.2.7 Rotation/Mirroring

VPU supports rotation together with mirroring function for decoding output image to display. It is done by rotator/mirror sub-block.

The rotation/mirroring process in decoding process requires additional bandwidth because VPU has to re-use the un-rotated image for decoding the next image. So the rotated image is written to other memory space. In this scheme, the display I/F has not to change memory space for displaying the decoded image because subsequent rotated image is written to the same space.

The rotator sub-blocks support 8-types mode of 90 x n degree(n=0, 1, 2, 3) rotating and mirroring. The figure below gives architecture diagram of post rotation/mirroring sub-block.

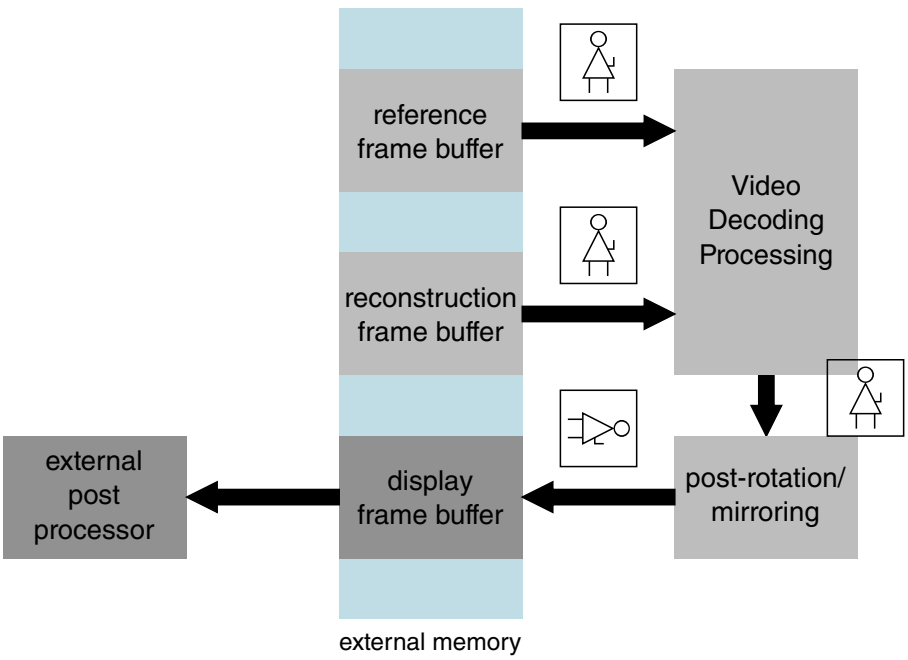


Figure 69-3. Post rotation/mirroring

69.4.2 Reset

There are four reset signals going into VPU, corresponding to four clock domains, active low.

- areset_n: used in AXI bus interface. Corresponding clock is aclk.
- creset_n: used in BIT processor and video codec hardware. Corresponding clock is cclk.
- ipg_hard_pos_async_reset_b: used in IP bus interface. Corresponding clock is ipg_clk_s.
- rreset_n: used in reference counter for vpu_idle or vpu_underrun. Corresponding clock is rclk.

The number of cycles for each reset signal must be at least 16 cycles.

VPU embeds an internal reset controller for feature of the software reset from the BIT processor. If any of the VPU blocks except the BIT processor is needed to reset (software reset), the host processor can enable this software reset by setting the software reset register through VPU API. But, the BIT processor cannot be reset by this software reset scheme, because the reset signal of the BIT processor is connected directly from external reset signal.

If reset fires when VPU is processing a transaction through the AXI bus, there is no guarantee that the AXI will complete the transaction normally, because VPU will be reset. If there are any corrupted data in memory, it can be discarded by software. Basically, if the host processor needs to issue a reset, it must check that there is no transaction on AXI bus between the VPU and external memory interface. In general, the AXI bus is free of VPU transactions when one frame decoding is completed. The start of next frame processing need software configuration.

69.4.3 Interrupts

There interrupt signal output from VPU. Basically, used to indicate completion of decoding or encoding one frame.

generated when VPU interrupt is enabled and as well as the interrupt condition is met. The interrupt signal is active high and retains untill the host processor clears it by writing "1" to the interrupt clear register. This signal is synchronized to the positive edge of cclk.

When getting frame completion interrupt, the software can set the parameters for next frame processing and start the BIT processor again. The parameters are mainly the source/destination frame buffer base address. It can be different from the previous frame buffer because the previous completed frame of data may be needed for other image block like display block or IPU in the system.

Basically, the following operation responding to interrupt is dependent upon the application. For example, the software can send the decoded frame to the Image Processing Unit for post-processing and display, at the same time the software should prepare the next bitstream to be processed to the external memory before starting a new processing. This can be done through VPU driver.

69.4.4 Endianness

VPU supports both little and big endian memory system.

User should specify the endianness of the bitstream buffer and frame buffer corresponding to the application scenario. The endianness configuration can be done through VPU API.

69.5 Initialization Information

VPU embedded BIT processor is highly optimized to handle bitstream data.

In addition to processing bitstreams, the BIT processor also controls the communication between VPU and host processor.

The BIT processor firmware to drive video decoding is divided into 2 parts.

One is boot code, which is downloaded by host processor through the IP bus to BIT processor internal memory, it is loaded only once at the VPU initialization stage. The size of the boot code is 1Kbyte. This boot code has also to be written to a region of external memory, and the base address of the firmware region is written through VPU API.

The other is a package of firmware for driving decoding processing. Before starting decoding, firmware has to be written to the continuous region of external memory following the boot code. At run-time, the BIT processor will self-download firmware into internal memory corresponding to the activated decoding standard. It is loaded through the AXI bus.

The VPU initialization process is shown in the figure below.

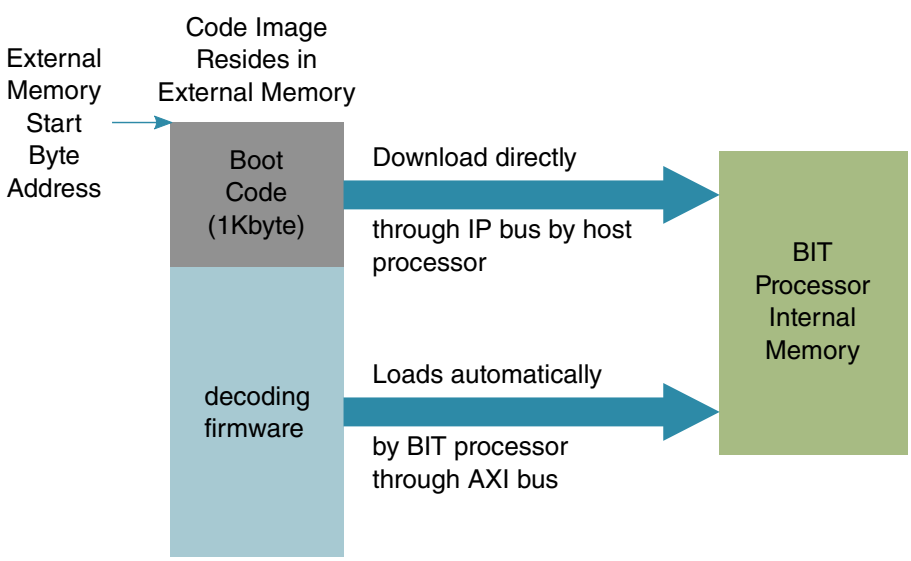


Figure 69-4. VPU initialization process

69.6 Application Information

The figure found here shows roles of the BIT processor and VPU video processing core sub-block and how to interface with application software.

Basically, at the frame level, host processor communicates with VPU through the provided API's. To give the video decoding more flexibility and debugging capability, all processes related to the bitstream are assigned to the BIT processor.

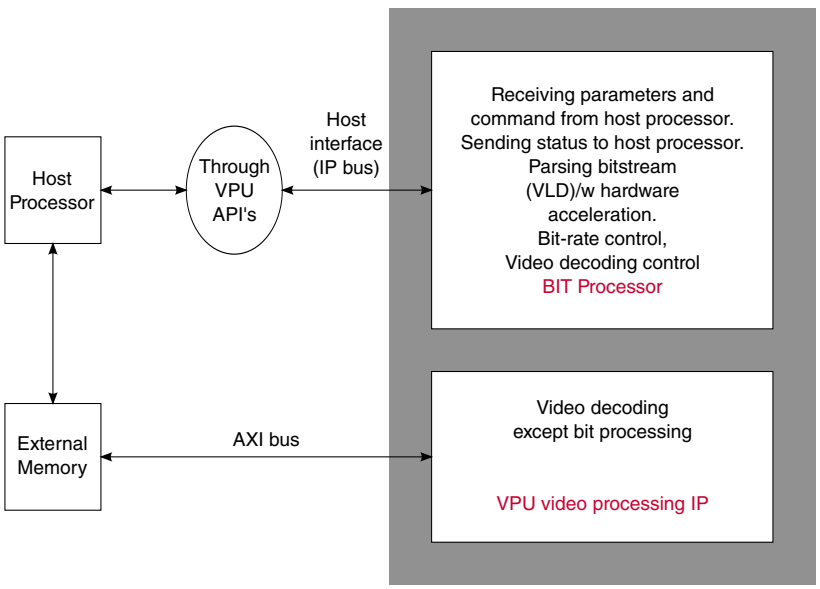


Figure 69-5. VPU interface with application software diagram

69.6.1 Video Decoding Processing Control

This section describes how BIT processor controls the video decoding and communicates with the host.

69.6.1.1 Video Decoding Process Flow

VPU can handle processes simultaneously. Each process can have a different format - MPEG-4, MPEG-2, H.264, VC-1 and etc. The figure below shows a simplified state diagram for running decoding process.

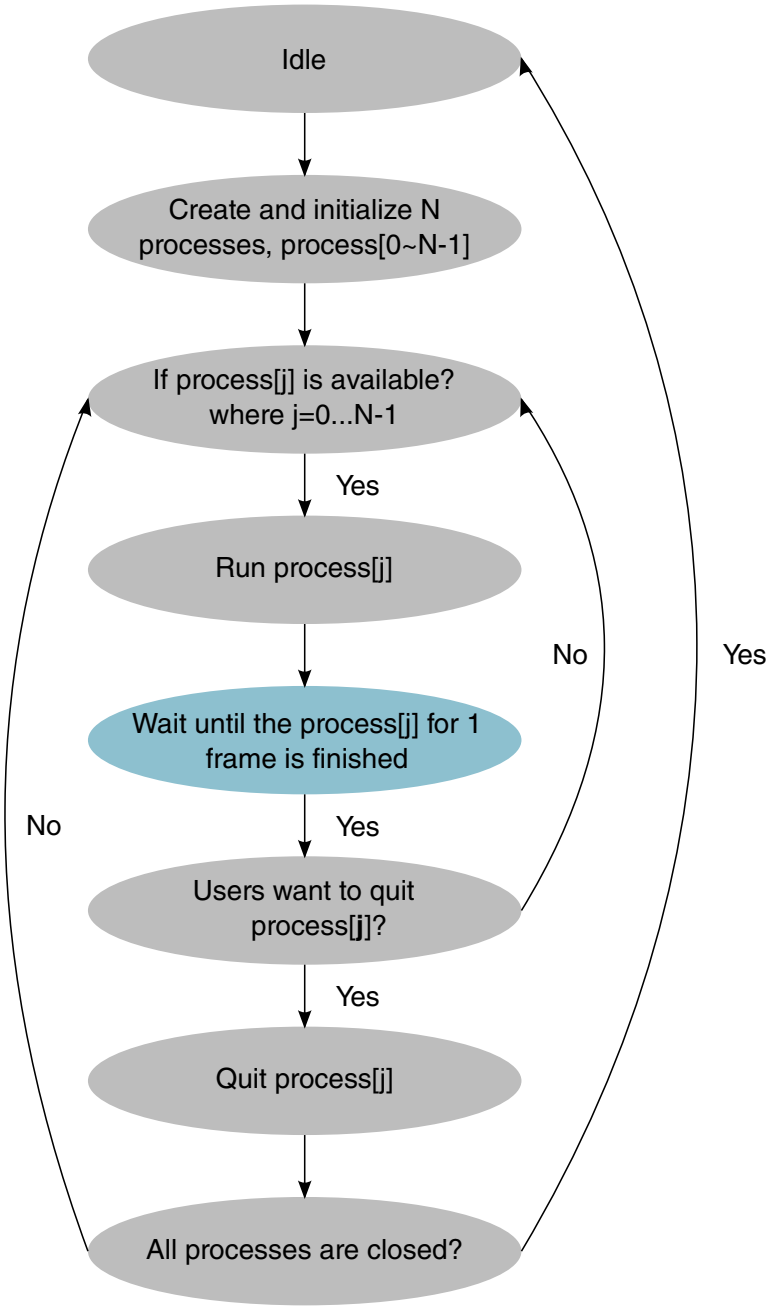


Figure 69-6. Decoding process state diagram

Each decoding process consists of three categories:

- Create processes- Software creates and configures processes.
- Running processes- At a proper time instance, software will begin a specific process. The proper time instance means when the decoding is in idle state and bitstream to be decoded is ready in the external memory.
- Quit Processes- Software can quit a specific process

If more than one process are ready to run, each process must be assigned to different process ID - RunIndex, which is range from 0 to 3. Basically, the ID is assigned based on the order of creation. For example, when 1 MPEG-4 Decoding + 1 H.264 Decoding + 1 MPEG-2 Decoding + 1 VC1 Decoding are running simultaneously, MPEG-4 Decoding is assigned to process index "0", H.264 Decoding is assigned to process index "1", MPEG-2 Decoding is assigned to index "2", and VC1 decoding is assigned to process index "3".

There is no priority rules for executing processes, after creating all processes at the initialization step, the host enables the BIT processor to execute process specified with the RunIndex. All processes are executed in time-division like mechanism, after one process finishes decoding or encoding a frame, another process then can be executed.

In conjunction with the process ID, the RunCodStd needs to be set, to define which coding standard is used with the created process and whether the created process will decode a bitstream. The table below shows the dedicated RunCodStd value for each decoding standard. All this can be done through VPU API.

Table 69-3. RuncCodStd Register Value for Coding Standard

Coding standard	RunCodStd
decoding	0
	1
decoding	2
encoding	3
decoding	4

69.6.1.2 Video Decoding Process Command

There are 7 execution commands to initialize, run, quit, set frame and set parameter processes. A command is sent by writing the command value to the RunCommand register through VPU API.

- DEC_SEQ_INIT- This command is to initiate a decoding process. At this command, BIT processor finds sequence header and parses the header to extract bitstream information such as picture size then the information is reported to DEC_SEQ_INIT return registers. API should set following configuration parameters before sending this command to VPU.
  - Bitstream buffer base address and size
  - Frame buffers base addresses and stride lines
- DEC_SEQ_END- This command is to terminate a decoding process.
  - After this command, no more PICTURE_RUN commands will be accepted for this process.

- DEC_PIC_RUN- This command is to decode one picture.
  - In decoding case, frame destination address should be set before executing.
- SET_FRAME_BUF- This command informs frame buffer addresses to be used as a decoding/reconstructing image to the BIT processor. Total 63 frame buffers may be used for decoding/reconstructing.
  - The decoding image must be reserved for motion compensation reference until it is not used for reference. So the decoded frame buffer is re-used carefully. The BIT processor receives the whole frame buffer address by this command before picture decoding is started. Then the BIT processor manages the frame buffer allocation for next storage area for decoding.
  - The frame buffer addresses are stored to external memory address. The luminance and one/two chrominance buffer addresses for each frame index must be stored.
- DEC_PARA_SET- This command adds a sequence parameter set or a picture parameter set in H.264 decoder case.
  - The sequence parameter set or the picture parameter set may be conveyed via "out-of-band". In that case, the host must transfer the sequence parameter set or picture parameter set to decoder by this command.
  - The sequence parameter set or picture parameter set must be written to the Parameter buffer of the BIT processor in RBSP format prior to executing this command. The BIT processor decodes the transferred sequence/the picture parameter and stores decoded contents. The decoded sequence/picture parameter set will be activated at decoding slice header with the matched sequence/picture parameter set id.
  - Multiple sequence/picture parameter sets may be delivered to decoder. They are distinguished by different sequence/picture parameter set id. BIT processor can process 32 sequence parameter sets and 256 picture parameter sets.
  - The type (sequence or picture) and size (byte count) of conveyed sequence/ picture parameter set must be delivered to BIT processor by command argument register.
- DEC BUF FLUSH- Flush data in bitstream buffer.
  - After this command finished, bitstream buffer read pointer will be 0. So host must set bitstream buffer write pointer as 0.
- GET F/W VERSION- Get firmware version.

There is a busy status register in the VPU to show whether the BIT processor is executing a command. The busy status keeps "1" until the command is finished, then the BIT processor can accept a new command.

69.6.1.3 Video Decoding Process Finish Detection

VPU raises the interrupt signal or busy state is asserted when a process finishes decoding a frame. So there are three ways of detecting whether the process is finished:

- Polling VPU interrupt status register. When interrupt is generated, the interrupt status register can indicate that.
- Polling VPU busy status register. When a DEC_PIC_RUN command is issued to BIT processor and the process is under operation, the busy status keeps "1", as soon as the busy status becomes "0", the decoding process is finished.
- Capture the interrupt signal from system level, respond to interrupt request within system interrupt service routine.

There is a single busy status register for all processes, user knows which process is finished by the specified running process ID - RunIndex. Interrupt status can be cleared by writing 1 to interrupt clear register. Busy state can be self cleared after read out.

69.6.1.4 Video Decoding Process Flow Example

The figure below shows a process flow example for decoding an H.264 bitstream and decoding an MPEG4 bitstream simultaneously. At first both decoding process are created and initialized, then each process is executed with DEC_PIC_RUN command alternately. More details are described as below.

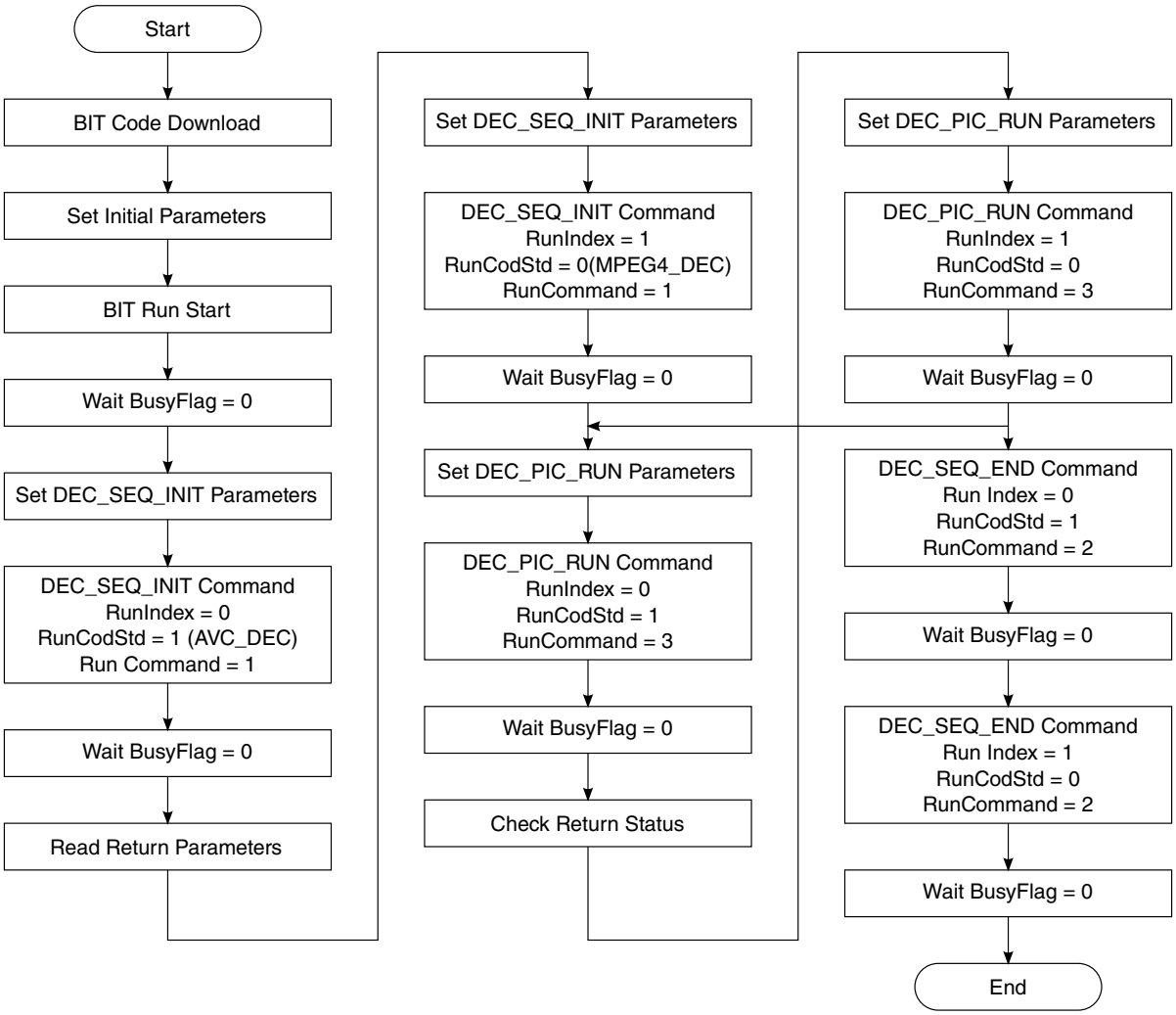


Figure 69-7. H.264 decoding and MPEG4 decoding process flow

NOTE

*RunCommand = 1 (DEC_SEQ_INIT); RunCommand = 2 (DEC_SEQ_END); RunCommand = 3 (DEC_PIC_RUN)

1. Initialize VPU
   - BIT Code Download: Load BIT Processor firmware to memory.
   - Set Initial Parameters: General configuration for BIT processor, setting working buffer base address, BIT Code memory address, bitstream buffer control and so on.
   - BIT Run Start: Run BIT processor to initialize VPU.
2. Create and initialize an H.264 decoding process
   - Set DEC_SEQ_INIT parameters: Configure base address and size of bitstream buffer, base address of frame buffers and so on.
   - Run DEC_SEQ_INIT command: Initiate an H.264 decoding process.

- Wait BusyFlag=0: Wait BIT processor completes DEC_SEQ_INIT command execution.
- Read Return Parameters: Read the features of decoded bitstream, such as the picture resolution and number of reference frames, this can be done by reading the RetSrcFormat and Ret264Info register through VPU API. In this way, the host can prepare the required frame buffers.

3. Create and initialize an MPEG4 decoding process
   - The flow is similar to the H.264 decoding process except the run standard setting.
4. Run the H.264 decoding process
   - Set DEC_PIC_RUN Parameters: Configure the frame destination address.
   - Run DEC_PIC_RUN command: Start the H.264 decoding process.
   - Wait BusyFlag=0: Wait the BIT processor completes DEC_PIC_RUN command execution. It also means one frame process is finished. (The finish detection can be implemented in other way described in Video Decoding Process Finish Detection.) The decoded frame can be sent to the Image Processing Unit for post-processing and display. The actual operation is dependant on the application.
5. Run the MPEG4 decoding process
   - The flow is similar to the H.264 decoding process. The decoding process should configure frame source address in addition to destination address.
6. Execute step 4 and step 5 alternately.
   - Before running decoding process, the host should load new bitstream to the bitstream buffer if it is empty, and update the frame destination address according to the application.
7. Stop the decoding processes
   - Run DEC_SEQ_END command to each process to terminate it.

Basically, the process flow for decoding is similar, though it may have minor change for different firmware version. The detailed implementation is done in the VPU driver.

69.6.2 Video Encoding Processing Control

This section describes how BIT processor controls the video encoding and communicates with the host.

69.6.2.1 The Pipeline for Encoding

The following figure shows the 11 pipeline stages for H.264 encoding.

In encoding, there are two separate data paths. The nine stages on the right form the encoding path, and the four stages on the left form the reconstruction data path.

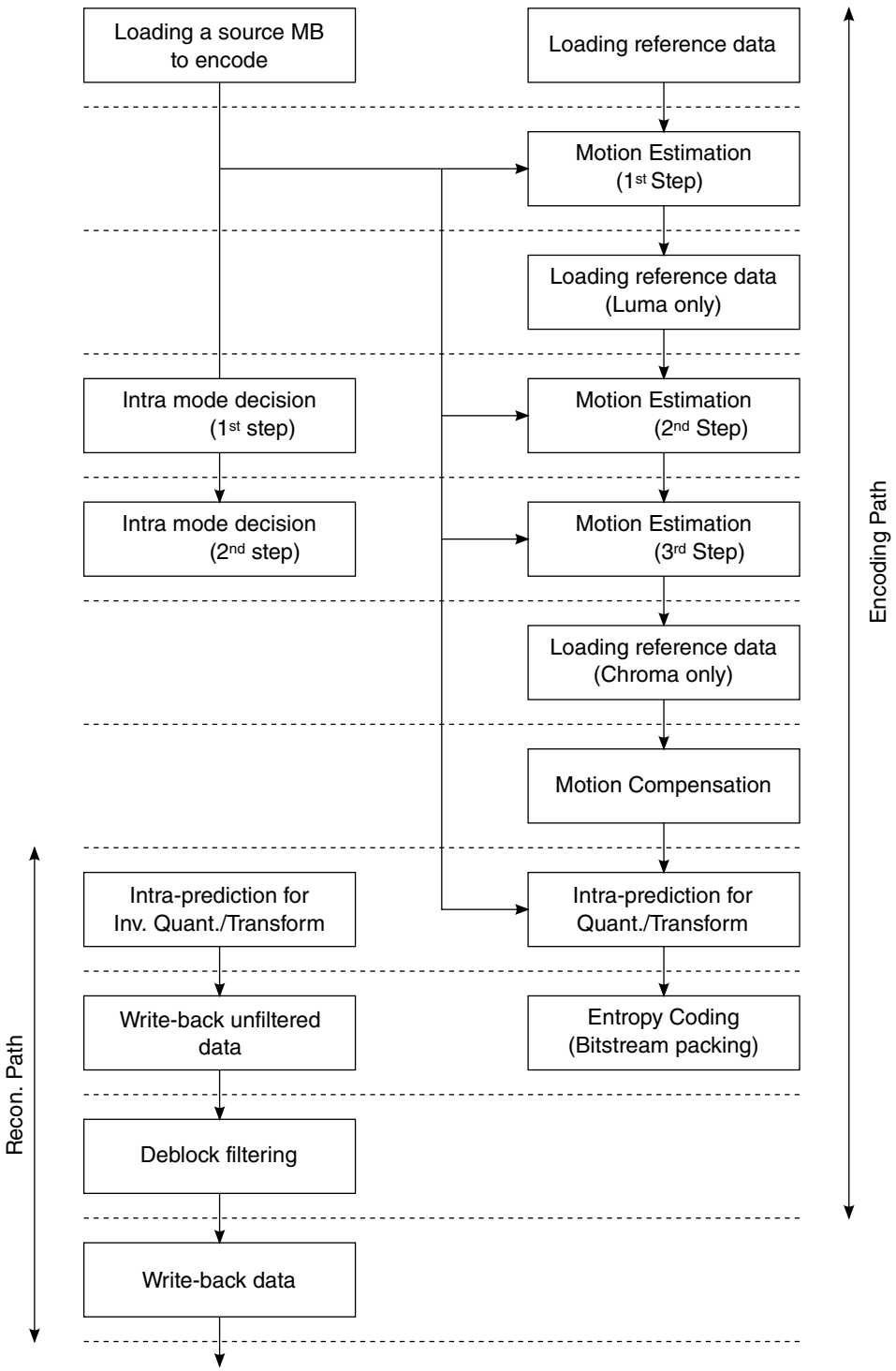


Figure 69-8. Encoding pipeline stage for H.264

The sub-block in each stage is controlled by BIT processor and main controller (macroblock sequencer). The detailed H.264 encoding pipeline stage is as follows.

- Pipeline Stage 1
  a. Load source and reference data through a sub-sampler.
  b. Loaded data passes through a sub-sampler.
- Pipeline Stage 2
  a. The motion estimation (ME) searches coarse motion vectors.
- Pipeline Stage 3
  a. Load data for refine motion estimation.
- Pipeline Stage 4
  a. Excute refine ME and fast intra-mode decision
- Pipeline Stage 5
  a. Excute fractional ME and fine intra-mode decision.
- Pipeline Stage 6
  a. Load reconstructed data for motion compensation.
- Pipeline Stage 7
  a. The MC block writes predicted data for the ME motion vector.
  b. The intra predictor calculates Intra SAD and cost for mode decision.
  c. The Inter/Intra prediction block writes predicted data local memory for subtraction in the next pipleline stage.
- Pipeline Stage 8
  a. Execute Intra Prediction and transfer the result to Quantizer/Transform.
  b. The quantized and transformed data are written to residual buffer for bitstream packing.
  c. The quantized and transformed data are transferred to the Inverse Quantizer/ Inverse Transfrom.
  d. The inverse quantized and inverse transformed data are added(reconstructed) with prediction block data, to make reconstructed frame.
- Pipeline Stage 9
  a. The BIT processor reads the residual buffer and makes the bitstream.
  b. Write back the un-filtered pixel data.
- Pipeline Stage 10
  a. The de-blocking filter reads reconstructed data from its local memory and run de-blocking filter for H.264.
  b. The filtered pixel data, that is final reconstructed picture, is stored in local memory of de-blocking filter block for write-back.
- Pipeline Stage 11
  a. The write-back DMA writes decoded picture to external frame buffer and it is for a reference picture of next picture decoding.

Figure 69-9 shows the 10 pipeline stages for MPEG-4/H.263 encoding when the deblocking filter is disabled and Figure 69-10 shows the 11 pipeline stages for H.263 when the deblocking filter is enabled. The encoding data path in MPEG-4/H.263 is similar to the H.264 encoding data path, except for the addition of an AC/DC prediction step.

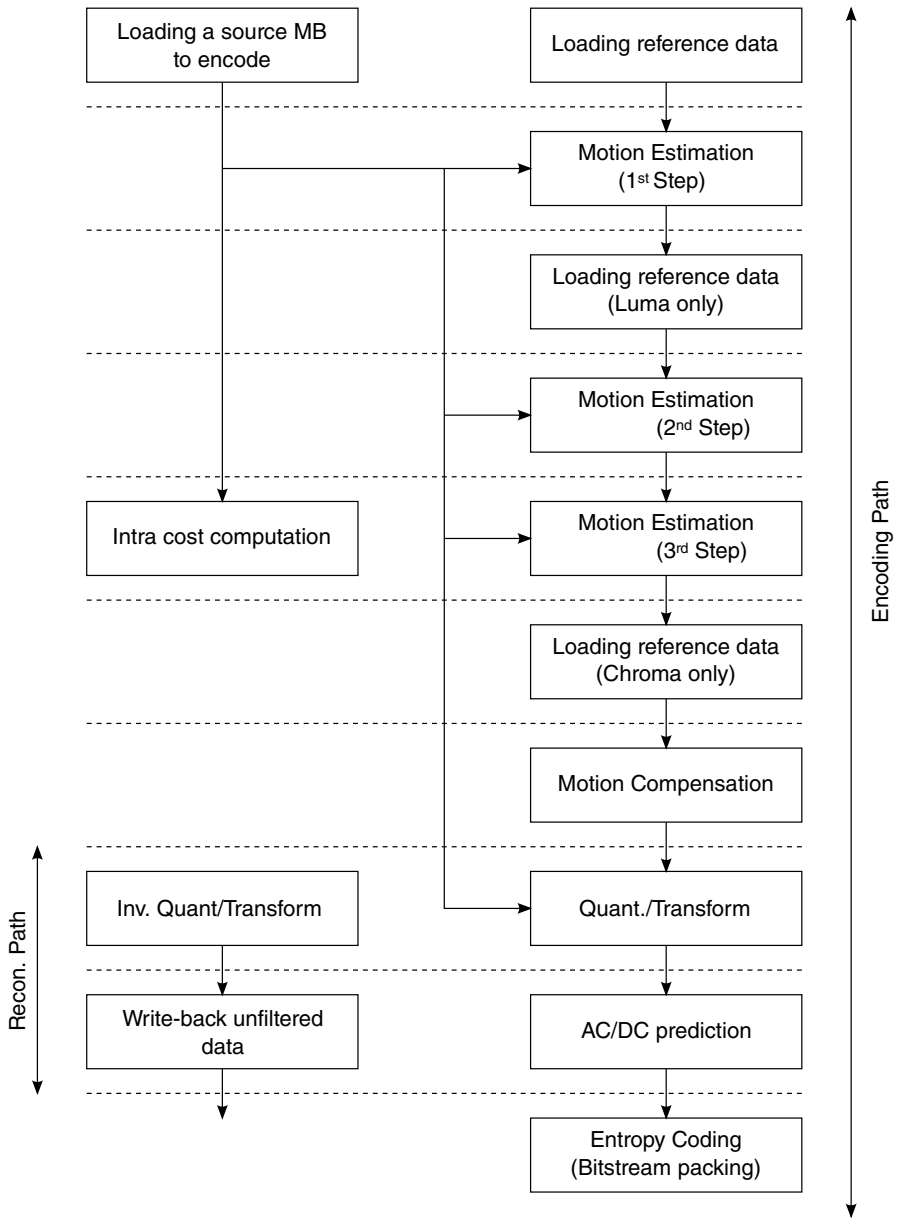


Figure 69-9. MPEG-4/H.263 encoding pipeline stage when deblocking filter is disabled

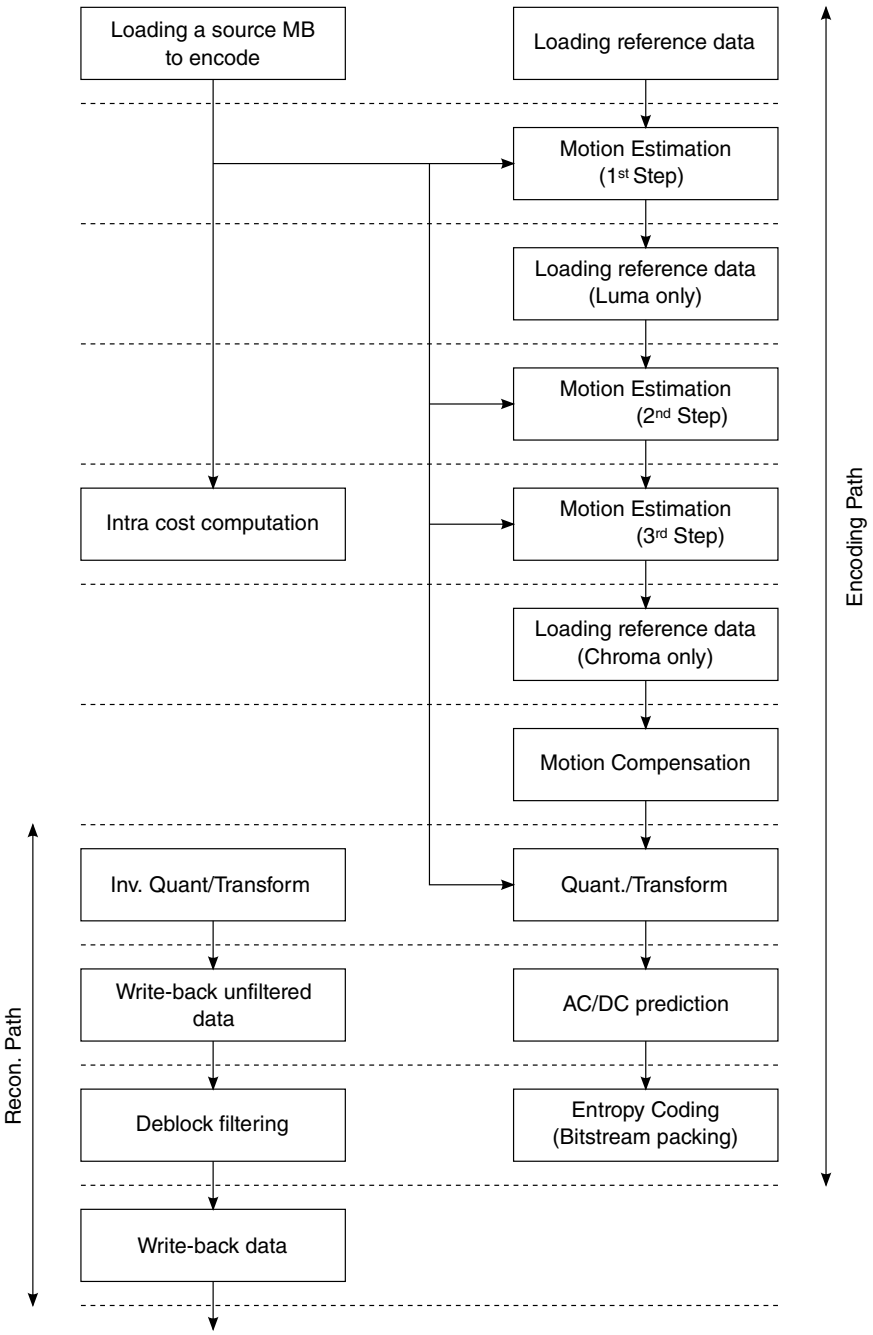


Figure 69-10. H.263 encoding pipeline stage when deblocking filter is enabled

69.6.3 Video Codec Processing Buffer Requirement

VPU has full access to the entire external memory. It uses external memory to load or store image frame, bitstream, program and data for the BIT processor.

AC/DC predication and de-blocking filtering also uses external memory. The buffer size requirement is dependent on the standard and target applications. For example, the H.264 decoding uses multiple reference frames up to 16. MPEG-4 and H.263 decoding uses only one reference frame. Each standard requires a different temporary memory size when it processes de-blocking or overlap-smoothing filtering.

VPU uses five kinds of buffers, as follows:

- Frame Buffer: for storing image frame.
- BIT processor program memory: for boot code and firmware.
- Working Buffer: for intermediate data from the BIT processor and the video decoding hardware.
- Bitstream Buffer: for loading bitstream.
- Parameter Buffer: for BIT processor command execution argument and return data.
- Search RAM: for the use of ME to reduce SDRAM bus loading.

Different buffers can be noncontiguous in external memory, though each buffer must be contiguous.

VPU also supports an optional secondary AXI bus which is connected to internal SRAM for storing temporal data of some sub-blocks, such as de-blocking filter and intra prediction. This decreases the total bandwidth to the external memory.

69.6.3.1 Memory Map Types of Frame Buffer

There are 7 map types of frame buffer:

- Type 0 : linear map
- Type 1 : Frame based tiled map, horizontal addressing
- Type 2 : Frame based tiled map, vertical addressing
- Type 3 : Field based tiled map, vertical addressing
- Type 4 : Frame/Field mixed tiled map, vertical addressing
- Type 5 : Tiled MB Raster Frame Map
- Type 6 : Tiled MB Raster Field Map

In the chip, only Type 0, 5 and 6 are supported.

In the linear map, a pixel is read out from or written to the frame buffer in a raster scan order for a frame. However, VPU requires pixels on an MB basis and such address access cannot be continuous on the physical memory. Meanwhile in the tiled map, the whole memory region is logically split into a specific number of tile-shaped segments. A tile can be an access unit in which addresses are sequential and have efficiently accessible

structure. The size of tile can be configurable according to applications. It might be an MB at least or a group of MBs whose addresses belong to the same row and bank address and are consecutive column address in SDRAM.

Meanwhile in the tiled map, the whole memory region is logically split into a specific number of tile-shaped segments. A tile can be an access unit in which addresses are sequential and have efficiently accessible structure. The size of tile can be configurable according to applications. It might be an MBat least or a group of MBs whose addresses belong to the same row and bank address and are consecutive column address in SDRAM. For more details, please refer to VPU API document.

69.6.3.2 Frame Buffer

This section describes the memory map of the frame buffer and the size requirement.

- For (M)JPEG codec: 4:0:0, 4:2:0, 4:2:2, and 4:4:4 are supported.
- For all other standards only 4:2:0 is supported.
- Luminance, Cb and Cr frame buffer base addresses in -byte alignment. In case of the interleaved Cb/Cr map, a base address for Cr is ignored because a base address for the Cb is used to store or load the interleaved Cb/Cr samples.
- Stride for the width of the luminance frame buffer should be equal or greater than the width of picture and multiple of .
- Endian of the frame buffers is configurable as little or big endian.
- Both the interleaved Cb/Cr map and the separate Cb/Cr map are supported.
- A field pair is stored in a single frame buffer.

As shown in Figure 69-11, a frame buffer is specified with the base address and the stride line. A complete image consists of Y, U, and V components. There are two kinds of configuration, separate Cb/Cr map and interleaved Cb/Cr map.

For the separate Cb/Cr configuration, each chroma component has its own buffer. Therefore, one frame buffer needs 3 buffers for Y, U, and V components, and these buffers can be noncontiguous in memory.

For the interleaved Cb/Cr configuration, only one buffer is needed for chroma components. Therefore, one frame buffer needs 2 buffers for Y, U, and V components, and these buffers can be noncontiguous in memory.

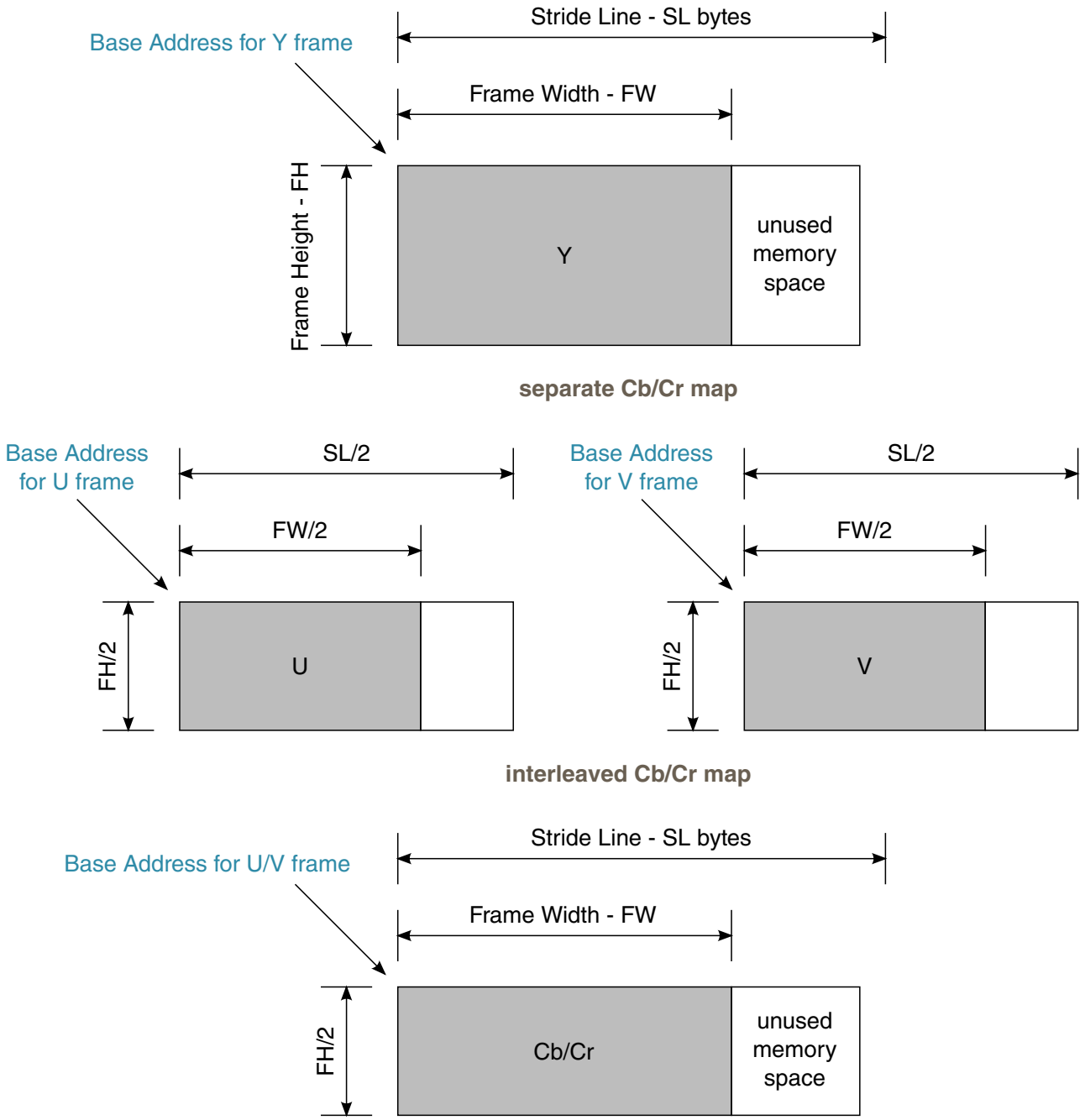


Figure 69-11. Frame Buffer configuration

Figure 69-11 shows the memory map of the frame buffer. In separate Cb/Cr map, for V frame buffer, the memory map is the same as the U frame buffer except for the base address.

VPU supports both little and big endian systems. Y(0,0) could be located in the bit[ ]. User can specify the endianness through VPU API.

The table below shows the frame buffer requirement for MPEG-4, H.264 and VC-1. The H.264 decoder requires multiple reference frames up to 16. In the case of VC-1 main profile, the decoder needs two reference frames to decode a B picture. Also, VC-1 requires two more frames that it stores for range reduction or multi-resolution.

The table below also shows the memory requirement in case of QCIF/CIF/VGA resolution image. In the case of H.264 decoding, the required size for the reference frame is dependent on the level being supported. The VPU supports up to H.264 decoding level 3.0, which the maximum decoded picture buffer size is defined as 3037.5Kbyte in the standard. To support H.264 CIF at level 3.0, 2524Kbyte is needed if 16 reference frames are used.

Table 69-4. Frame Buffer Requirement

		MEPG-4 Decoder	H.264 Decoder	VC-1 Decoder
QCIF	Reference Frames	2	16	2
	Instant Frames	1	1	2
	Display Frames	1	1	2
	Total Frames	4	18	6
	Picture Size[1]	37Kbyte	37Kbyte	37Kbyte
	Total Frame Size	148Kbyte	668Kbyte	222Kbyte
CIF	Reference Frames	2	16	2
	Instant Frames	1	1	2
	Display Frames	1	1	2
	Total Frames	4	18	6
	Picture Size	148Kbyte	148Kbyte	148Kbyte
	Total Frame Size	592Kbyte	2672Kbyte	891Kbyte
VGA	Reference Frames	2	5	2
	Instant Frames	1	1	2
	Display Frames	1	1	2
	Total Frames	4	7	6
	Picture Size	450Kbyte	450Kbyte	450Kbyte
	Total Frame Size	1800Kbyte	3150Kbyte	2700Kbyte

1. The Picture Size is the minimum size of one frame buffer with assumption that the picture is YUV 4:2:0 format and the stride line is equal to frame width.

69.6.3.3 BIT Processor Program Buffer

At the initialization stage of VPU, the host processor must download boot code to the BIT processor. After initialization, the BIT processor loads a program corresponding to the activated standard.

The program size of the boot code is 1Kbyte. There are total Kbyte sizes for the current version of BIT firmware to support multi-standards decoding.

69.6.3.4 Working Buffer

Besides buffers for frames and firmware, additional working buffer for intermediate data from the BIT processor and decoding is needed.

The buffers are such as the reconstructed pixel row buffer for MPEG-4 AC/DC prediction or H.264 intra-prediction, context saving buffer for running multiple processes and various temporal storage buffer for decoding process.

The required working buffer size varies according to decode size, decoding standard, decoding capability. For example, AC/DC prediction buffer size is determined by picture width and the maximum bitstream re-ordering buffer for data partition is determined by the maximum bitstream size of one picture. Working buffer size may change for different firmware version. The current version of firmware requires max Kbyte for working. Its size can be set through VPU API. The detailed working buffer is organized as indicated in the table below.

69.6.3.5 Bitstream Buffer

The host processor has to assign buffers for bitstreams on a per instance basis.

If VPU handles N-bitstreams simultaneously in an application, the host should assign N bitstream buffers, and specify the base address and size. The External bitstream buffer is "ring buffer" type. The start address of ring buffer and buffer size must be written by host to BIT processor. The current read or write address of ring buffer is automatically wrapped-around by firmware.

In decoding case, the host writes the bitstream to be decoded then BIT processor reads bitstream. In this case, the bitstream overwriting or underflow may occur and if it occurs, decoding will fail. To prevent overwriting or underflow, current bitstream read/write pointer must be exchanged between the host and the BIT processor.

The BIT processor writes current read pointer of ring buffer to internal register and the host must write current write pointer of ring buffer to internal register. The BIT processor checks the bit buffer empty (underflow) status by comparing current read pointer and write pointer. If no more bitstream data is available to be decoded (buffer empty status), the BIT processor stops bitstream decoding to prevent mis-reading the bitstream and

waits until the host writes more bitstream data and updates write pointer. The host must check the current read pointer and write pointer before writing more bitstream data to ring buffer to prevent overwriting bitstream data.

69.6.3.6 Parameter Buffer

Host processor must reserve parameter buffer in external memory for BIT processor command execution argument and return data.

69.6.3.7 Search RAM

VPU's motion estimation sub-block uses a search RAM to reduce the bandwidth on the external SDRAM.

Generally, motion estimation reads a reference pixel data several times. To avoid this B/W overhead, the motion estimation sub-block loads the reference pixel data form the external SDRAM one time and stores them to the search RAM through AXI bus. The stored reference pixel data is loaded several times to search the motion vector, but these operations are conducted through AMBA AXI bus. Therefore the B/W of loading reference pixel data will be reduced using search RAM in AXI bus.

69.6.3.8 Buffer Requirement Summary

69.7 VPU Memory Map/Register Definition

VPU registers are all 32-bit wide, support only 32bit aligned read/write operation. VPU registers are grouped into several regions corresponding to different decoding process step. They are used for decoding process configuration and control. They can only be accessed through IP bus interface.

Please note that there are some undefined address space in VPU memory map, any read/write accessing to this register address space is ignored in the VPU. Read accessing to write only register returns ZERO value. Write accessing to read only register is ignored.

The BIT processor registers' memory map in VPU is 0xBASE_0000~0xBASE_01FC. The BIT processor registers are divided into 2 categories.

- Address 0xBASE_0000~0xBASE_00FC (64 registers address space) are hardware registers. These registers have reset values and their functions are fixed (not configurable).
- Address 0xBASE_0100~0xBASE_01FC (64 registers address space) are software registers. They have no reset values and can be configured by internal BIT processor. Their definitions may change for different firmware version, so they are not provided here. This type of registers can be used as general parameter registers between host processor and BIT processor.
  - The first 32 parameter registers (address 0xBASE_0100~0xBASE_017C) are used as static parameters. Definition and functions of those registers are same for all kinds of run commands.
  - The second 32 parameter registers (address 0xBASE_0180~0xBASE_01FC) are used as temporal parameters. The definition and functions of those registers may differ in different run commands.

The memory map for the hardware registers of VPU is shown in the table below.

Please refer to the VPU API document for descriptions on software registers access.

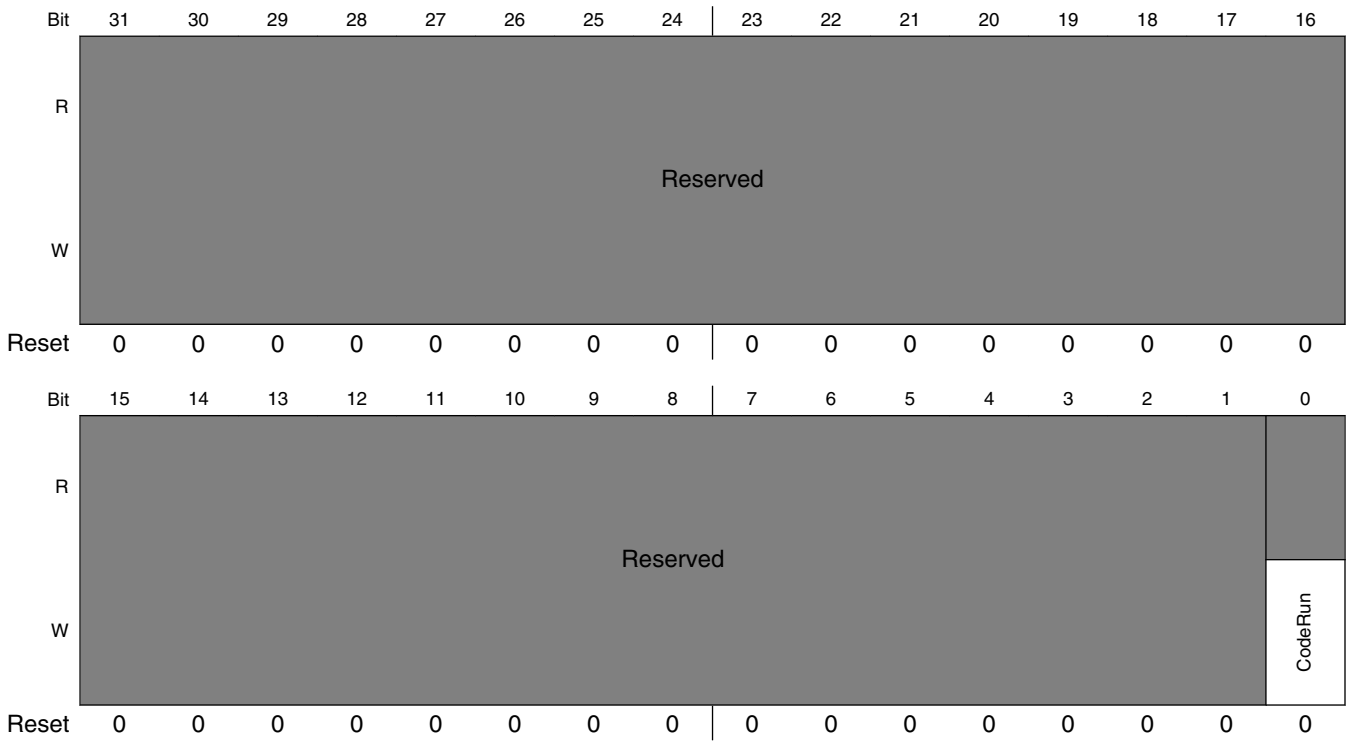
VPU memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
204_0000	BIT Processor run start (VPU_CodeRun)	32	W	0000_0000h	69.7.1/5706
204_0004	BIT Boot Code Download Data register (VPU_CodeDown)	32	W	0000_0000h	69.7.2/5706
204_0008	Host Interrupt Request to BIT (VPU_HostIntReq)	32	W	0000_0000h	69.7.3/5707
204_000C	BIT Interrupt Clear (VPU_BitIntClear)	32	W	0000_0000h	69.7.4/5708
204_0010	BIT Interrupt Status (VPU_BitIntSts)	32	R	0000_0000h	69.7.5/5709
204_0018	BIT Current PC (VPU_BitCurPc)	32	R	0000_0000h	69.7.6/5710
204_0020	BIT CODEC Busy (VPU_BitCodecBusy)	32	R	0000_0000h	69.7.7/5711

69.7.1 BIT Processor run start (VPU_CodeRun)

See the figure below for illustration of valid bits in VPU Code Run Register and the table below for description of the bit fields in the register.

Address: 204_0000h base + 0h offset = 204_0000h



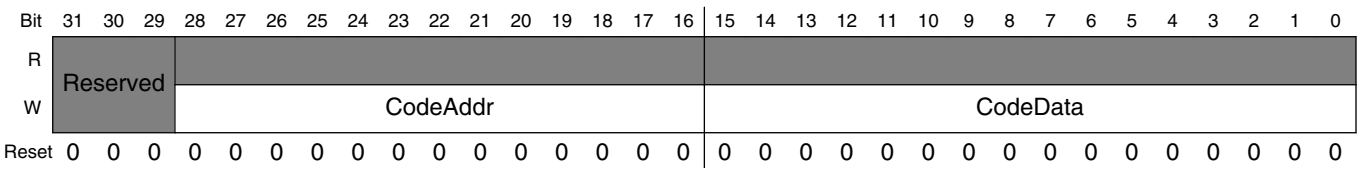
VPU_CodeRun field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 CodeRun	VPU_CodeRun. BIT processor run start bit. 0 BIT Processor stop execution. 1 BIT Processor start execution.

69.7.2 BIT Boot Code Download Data register (VPU_CodeDown)

See the figure below for illustration of valid bits in VPU BIT Boot Code Download Data Register and the following table for description of the bit fields in the register.

Address: 204_0000h base + 4h offset = 204_0004h



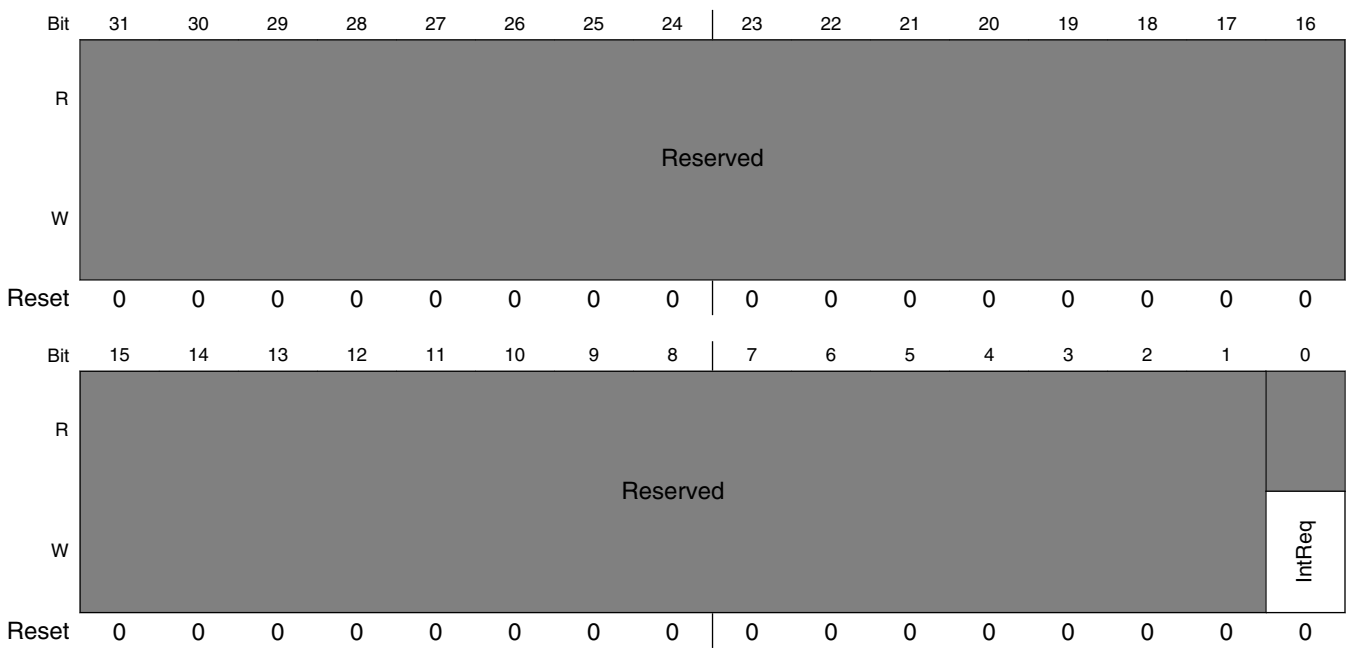
VPU_CodeDown field descriptions

Field	Description
31–29 -	This field is reserved. Reserved
28–16 CodeAddr	CodeAddr[12:0] Download address of VPU BIT boot code, which is VPU internal address of BIT processor.
CodeData	CodeData[15:0] Download data of VPU BIT boot code.

69.7.3 Host Interrupt Request to BIT (VPU_HostIntReq)

See the figure below for illustration of valid bits in VPU Host Interrupt Request Register and the following table for description of the bit fields in the register.

Address: 204_0000h base + 8h offset = 204_0008h



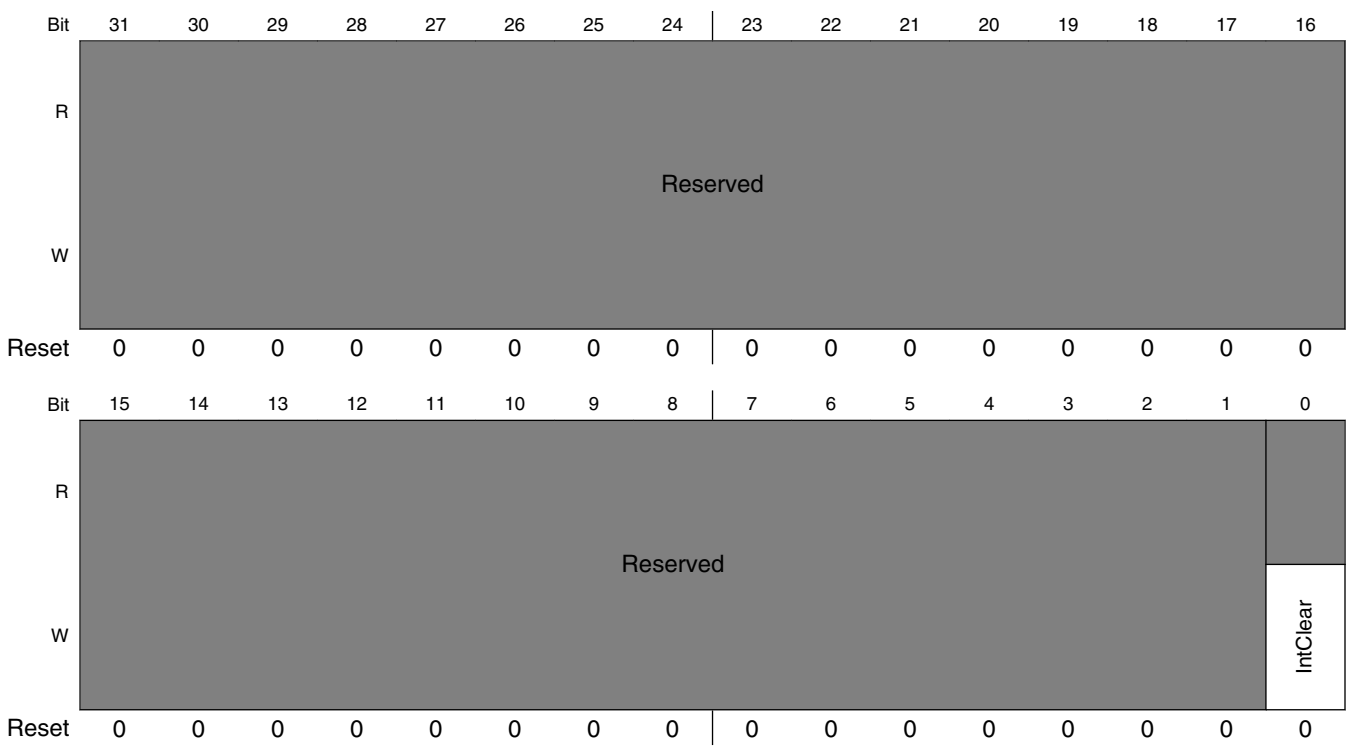
VPU_HostIntReq field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 IntReq	IntReq. The host interrupt request bit. 0 No host interrupt is requested. 1 The host processor request interrupt to the BIT processor.

69.7.4 BIT Interrupt Clear (VPU_BitIntClear)

See the figure below for illustration of valid bits in VPU BIT Interrupt Clear Register and the following table for description of the bit fields in the register.

Address: 204_0000h base + Ch offset = 204_000Ch



VPU_BitIntClear field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 IntClear	IntClear. BIT interrupt clear bit.

Table continues on the next page...

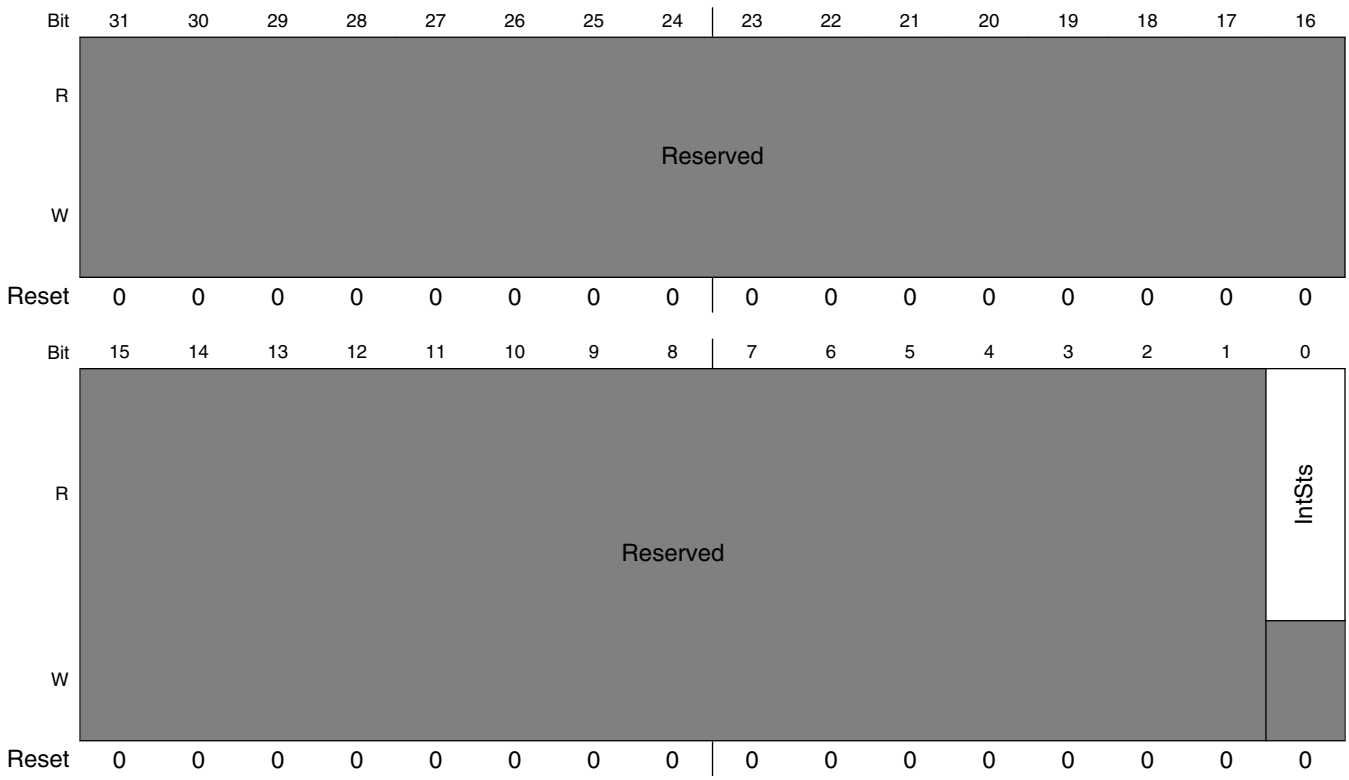
VPU_BitIntClear field descriptions (continued)

Field	Description
	0 No operation is issued. 1 Clear the BIT interrupt to the host.

69.7.5 BIT Interrupt Status (VPU_BitIntSts)

See the figure below for illustration of valid bits in VPU BIT Interrupt Status Register and the following table for description of the bit fields in the register.

Address: 204_0000h base + 10h offset = 204_0010h



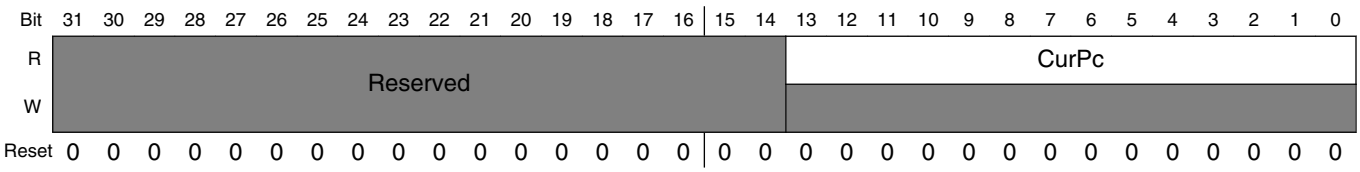
VPU_BitIntSts field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 IntSts	IntSts. BIT interrupt status bit. 0 No BIT interrupt is asserted. 1 The BIT interrupt is asserted to the host. It is cleared when the host processor write "1" to VPU_BitIntClear register.

69.7.6 BIT Current PC (VPU_BitCurPc)

See the figure below for illustration of valid bits in VPU BIT Current PC Register and the following table for description of the bit fields in the register.

Address: 204_0000h base + 18h offset = 204_0018h



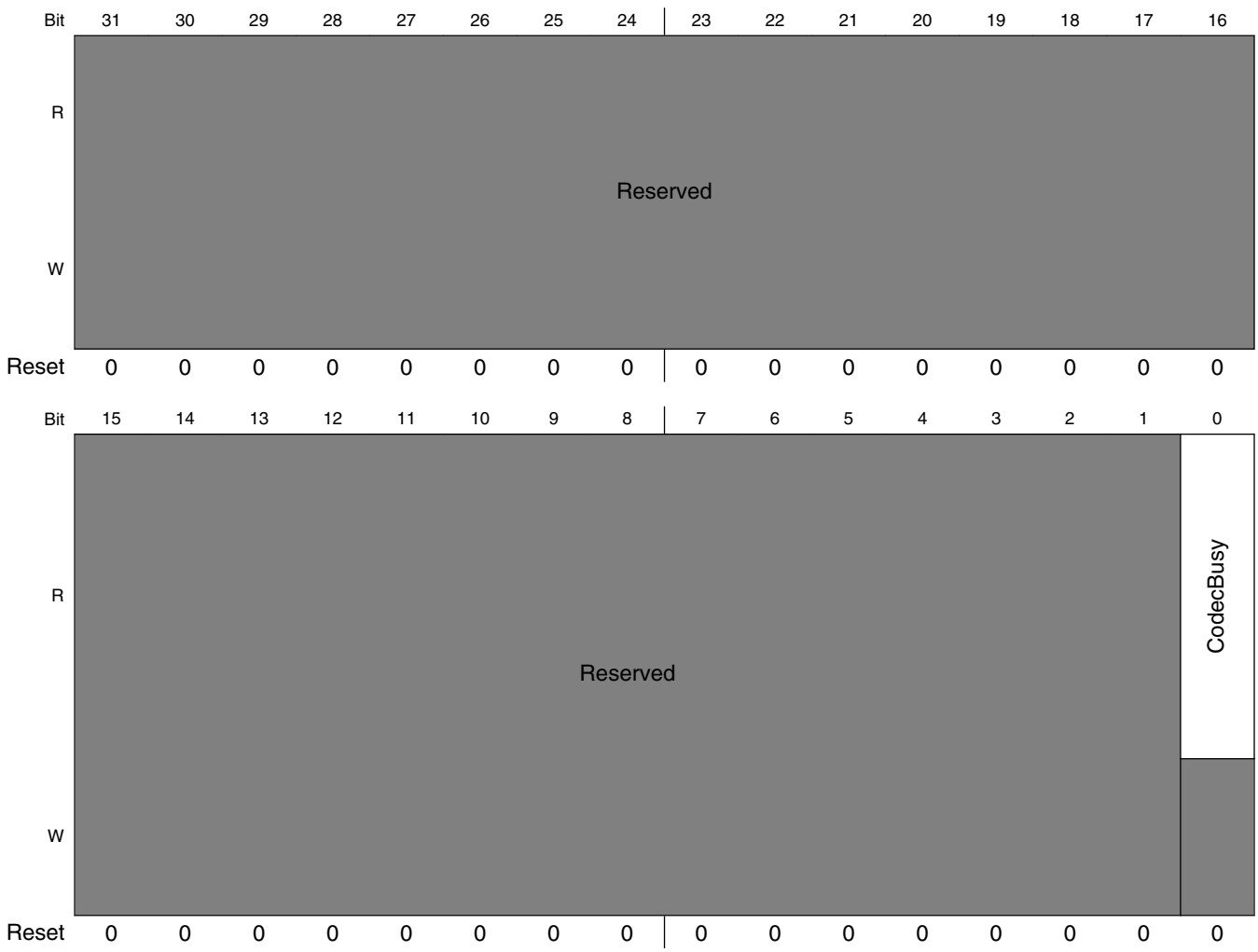
VPU_BitCurPc field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
CurPc	CurPc[13:0]. BIT current PC value. Returns the current program counter of BIT processor by reading this register.

69.7.7 BIT CODEC Busy (VPU_BitCodecBusy)

See the figure below for illustration of valid bits in VPU BIT Codec Busy Register and the following table for description of the bit fields in the register.

Address: 204_0000h base + 20h offset = 204_0020h



VPU_BitCodecBusy field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 CodecBusy	Codec busy flag for Bit processor.BIT processor write "1"to this register when the processor is running."0"means processor is waiting for a command.This value is connected to the o_vpu_idle.

Chapter 70
Watchdog Timer (WDOG)

70.1 Overview

The Watchdog Timer (WDOG) protects against system failures by providing a method by which to escape from unexpected events or programming errors.

Once the WDOG is activated, it must be serviced by the software on a periodic basis. If servicing does not take place, the timer times out. Upon timeout, the WDOG asserts the internal system reset signal, WDOG_RESET_B_DEB to the System Reset Controller (SRC).

There is also a provision for WDOG signal assertion by timeout counter expiration. There is an option of programmable interrupt generation before the counter actually times out. The time at which the interrupt needs to be generated prior to counter timeout is programmable. There is a power down counter which is enabled out of any reset (POR, Warm/Cold). This counter has a fixed timeout period of 16 seconds, upon which it asserts the WDOG signal.

Flow diagrams for the timeout counter, power down counter and interrupt operations are are shown in Flow Diagrams.

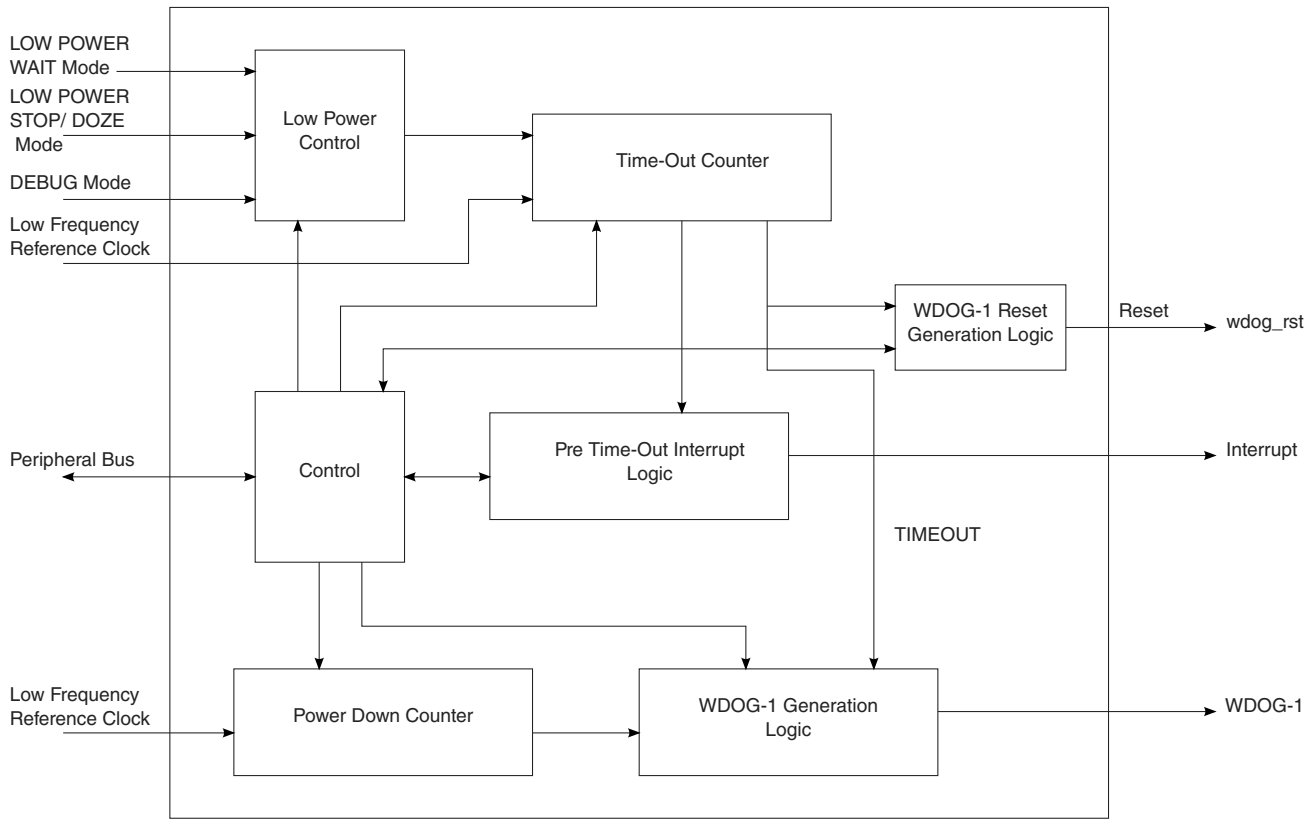


Figure 70-1. WDOG Diagram

70.1.1 Features

The WDOG features are listed below:

- Configurable timeout counter with timeout periods from 0.5 to 128 seconds which, after timeout expiration, result in the assertion of WDOG_RESET_B_DEB reset signal .
- Time resolution of 0.5 seconds
- Configurable timeout counter that can be programmed to run or stop during low-power modes
- Configurable timeout counter that can be programmed to run or stop during DEBUG mode
- Programmable interrupt generation prior to timeout
- The duration between interrupt and timeout events can be programmed from 0 to 127.5 seconds in steps of 0.5 seconds.
- Power down counter with fixed timeout period of 16 seconds, which if not disabled after reset will assert WDOG_B signal low

- Power down counter will be enabled out of any reset (POR, Warm / Cold reset) by default.

70.2 External signals

Table 70-1. WDOG External Signals

Signal	Description	Pad	Mode	Direction
WDOG1_B	This signal will power down the chip.	DISP0_DAT8	ALT3	I/O
		GPIO_9	ALT1	
		SD1_DAT2	ALT4	
WDOG1_RESET_B_DEB	This signal is a reset source for the chip.	SD1_DAT2	ALT6	O
WDOG2_B	This signal will power down the chip.	DISP0_DAT9	ALT3	I/O
		GPIO_1	ALT1	
		SD1_DAT3	ALT4	
WDOG2_RESET_B_DEB	This signal is a reset source for the chip.	SD1_DAT3	ALT6	O

70.3 Clocks

This section describes clocks and special clocking requirements of the block.

The WDOG uses the low frequency reference clock for its counter and control operations. The peripheral bus clock is used for register read/write operations.

The following table describes the clock sources for WDOG. Please see Introduction for clock setting, configuration and gating information.

Table 70-2. WDOG Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	IP Global functional clock. All functionality inside the WDOG module is synchronized to this clock.
ipg_clk_s	ipg_clk_root	IP slave bus clock. This clock is synchronized to ipg_clk and is only used for register read/write operations.
ipg_clk_32k	ckil_sync_clk_root	Low frequency (32.768 kHz) clock that continues to run in low-power mode. It is assumed that the Clock Controller will provide this clock signal synchronized to ipg_clk in the normal mode, and switch to a non-synchronized signal in low-power mode when the ipg_clk is off.

70.4 Watchdog mechanism and system integration

There are two WDOG modules, WDOG1 and WDOG2 (TZ) in the chip. The modules are disabled by default (after reset). WDOG1 will be configured during boot while WDOG2 is dedicated for secure world purposes and will be activated by TZ software if required. The TZ watchdog (TZ watchdog) module protects against TZ starvation by providing a method of escaping normal mode and forcing a switch to the TZ mode.TZ starvation is a situation where the normal OS prevents switching to the TZ mode. Such a situation is undesirable as it can compromise the system's security.

Once the TZ WDOG module is activated, it must be serviced by TZ on a periodic basis. If servicing does not take place, the timer times out. Upon a timeout, the TZ WDOG asserts a TZ-mapped interrupt that forces switching to the TZ mode. If it is still not serviced, the TZ WDOG asserts a security violation signal to the CSU. The TZ WDOG module cannot be programmed or de-activated by normal mode software.

The WDOG modules operate as follows:

- If servicing does not take place, the timer times out and the wdog_rst_b signal is activated (low)
- Interrupt can be generated before the counter actually times out
- The wdog_rst_b signal can be activated by software
- There is a power-down counter which gets enabled out of any reset. This counter has a fixed timeout period of 16 seconds upon which it will assert the ipp_wdog_b signal.

The following figure shows the WDOG1 and WDOG2 connectivity at the system level.

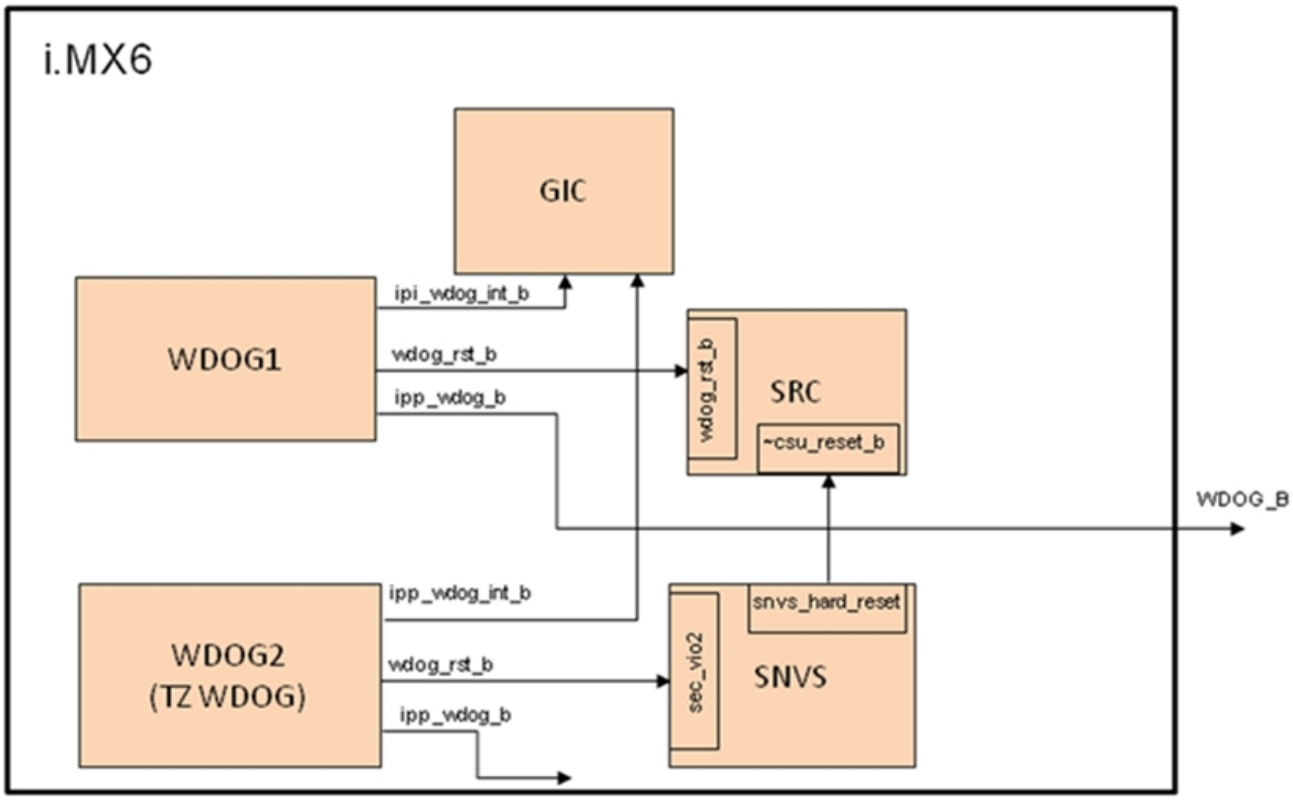


Figure 70-2. System integration

70.5 Functional description

This section provides a complete functional description of the block.

70.5.1 Timeout event

The WDOG provides timeout periods from 0.5 to 128 seconds with a time resolution of 0.5 seconds.

The user can determine the timeout period by writing to the WDOG timeout field (WT[7:0]) in the Watchdog Control Register (WDOG_WCR). The WDOG must be enabled by setting the WDE bit of Watchdog Control Register (WDOG_WCR) for the timeout counter to start running. After the WDOG is enabled, the counter is activated, loads the timeout value and begins to count down from this programmed value. The timer will time out when the counter reaches zero and the WDOG outputs a system reset signal, WDOG_RESET_B_DEB and asserts WDOG_B (WDT bit should be set in Watchdog Control Register (WDOG_WCR)).

However, the timeout condition can be prevented by reloading the counter with the new timeout value (WT[7:0] of WDOG_WCR) if a service routine (see Servicing WDOG to reload the counter) is performed before the counter reaches zero. If any system errors occur which prevent the software from servicing the Watchdog Service Register (WDOG_WSR), the timeout condition occurs. By performing the service routine, the WDOG reloads its counter to the timeout value indicated by bits WT[7:0] of the Watchdog Control Register (WDOG_WCR) and it restarts the countdown.

A system reset will reset the counter and place it in the idle state at any time during the countdown. The counter flow diagram is shown in Flow Diagrams.

NOTE

The timeout value is reloaded to the counter either at the time WDOG is enabled or after the service routine has been performed.

70.5.1.1 Servicing WDOG to reload the counter

To reload a timeout value to the counter the proper service sequence begins by writing 0x_5555 followed by 0x_AAAA to the Watchdog Service Register (WDOG_WSR). Any number of instructions can be executed between the two writes. If the WDOG_WSR is not loaded with 0x_5555 prior to writing 0x_AAAA to the WDOG_WSR, the counter is not reloaded. If any value other than 0x_AAAA is written to the WDOG_WSR after 0x_5555, the counter is not reloaded. This service sequence will reload the counter with the timeout value WT[7:0] of Watchdog Control Register (WDOG_WCR). The timeout value can be changed at any point; it is reloaded when WDOG is serviced by the core.

70.5.2 Interrupt event

Prior to timeout, the WDOG can generate an interrupt which can be considered a warning that timeout will occur shortly.

The duration between interrupt event and timeout event can be controlled by writing to the WICT field of Watchdog Interrupt Control Register (WDOG_WICR). It can vary between 0 and 127.5 seconds. If the WDOG is serviced (Servicing WDOG to reload the counter)before the interrupt generation, the counter will be reloaded with the timeout value WT[7:0] of Watchdog Control Register (WDOG_WCR) and the interrupt will not be triggered.

70.5.3 Power-down counter event

The power-down counter inside WDOG will be enabled out of reset. This counter has a fixed timeout value of 16 seconds, after which it will drive the WDOG_B signal low.

To prevent this, the software must disable this counter by clearing the PDE bit of Watchdog Miscellaneous Control Register (WDOG_WMCR) within 16 seconds of reset deassertion. Once disabled, this counter can't be enabled again until the next system reset occurs. This feature is intended to prevent the hanging up of cores after reset, as WDOG is not enabled out of reset.

70.5.4 Low power modes

70.5.4.1 STOP and DOZE mode

If the WDOG timer disable bit for low power STOP and DOZE mode (WDZST) bit in the Watchdog Control Register (WDOG_WCR), is cleared, the WDOG timer continues to operate using the low frequency reference clock. If the low power enable (WDZST) bit is set, the WDOG timer operation will be suspended in low power STOP or DOZE mode. Upon exiting low power STOP or DOZE mode, the WDOG operation returns to what it was prior to entering the STOP or DOZE mode.

70.5.4.2 WAIT mode

If the WDOG timer disable bit for low power WAIT mode (WDW) bit in the Watchdog Control Register (WDOG_WCR), is cleared, the WDOG timer continues to operate using the low frequency reference clock . If the low power WAIT enable (WDW) bit is set, the WDOG timer operation will be suspended. Upon exiting low power WAIT mode, the WDOG operation returns to what it was prior to entering the WAIT mode.

NOTE

The WDOG timer won't be able to detect events that happen for periods shorter than one low frequency reference clock cycle. For example, in repeated WAIT mode entry or exit, if the RUN mode time is less than one low frequency reference clock cycle and if the WDW bit is set, the WDOG timer may never time out, even though the system is in RUN mode for a finite duration; WDOG may not see a low frequency reference clock edge during its wake time.

70.5.5 Debug mode

The WDOG timer can be configured for continual operation, or for suspension during debug mode. If the WDOG debug enable (WDBG) bit is set in the Watchdog Control Register (WDOG_WCR), the WDOG timer operation is suspended in debug mode. If the WDBG bit is set and the debug mode is entered, WDOG timer operation is suspended after two low frequency reference clocks. Similarly, WDOG timer operation continues after two low frequency reference clocks of debug mode exit. Register read and write accesses in debug mode continue to function normally. Also, while in debug mode, the WDE bit of Watchdog Control Register (WDOG_WCR) can be enabled/disabled directly. If the WDOG debug enable (WDBG) bit is cleared then WDOG timer operation is not suspended. The power-down counter is not affected by debug mode entry/exit.

NOTE

If the WDE bit of Watchdog Control Register (WDOG_WCR) is set/cleared while in debug mode, it remains set/cleared even after exiting debug mode.

70.5.6 Operations

70.5.6.1 Watchdog reset generation

The WDOG generated reset signal WDOG_RESET_B_DEB is asserted by the following operations:

- A software write to the Software Reset Signal (SRS) bit of the Watchdog Control Register (WDOG_WCR).
- WDOG timeout. See Timeout event.

The $\overline{\text{wdog_rst}}$ will be asserted for one clock cycle of low frequency reference clock for both a timeout condition and a software write occurrence. It remains asserted for 1 clock cycle of low frequency reference clock even if a system reset is asserted in between. Figure 70-4 shows the timing diagram of this signal due to a timeout condition.

70.5.6.2 WDOG_B generation

The WDOG asserts WDOG_B in the following scenarios:

- Software write to WDA bit of Watchdog Control Register (WDOG_WCR). WDOG_B signal remains asserted as long as the WDA bit is "0".
- WDOG timeout condition, WDT bit of Watchdog Control Register (WDOG_WCR) must be set for this scenario. A description of the timeout condition can be found in the Timeout event. WDOG_B signal remains asserted until a power-on reset (POR) occurs. It gets cleared after the POR occurs (not due to any other system reset). Figure 70-5 shows the timing diagram of WDOG_B due to timeout condition.
- WDOG power-down counter timeout, PDE bit of Watchdog Miscellaneous Control Register (WDOG_WMCR) should not be cleared for this scenario. A description of this counter can be found in the Power-down counter event. WDOG_B signal remains asserted for one clock cycle of low frequency reference clock.

Figure 70-3 shows the scenarios under which WDOG_B gets asserted.

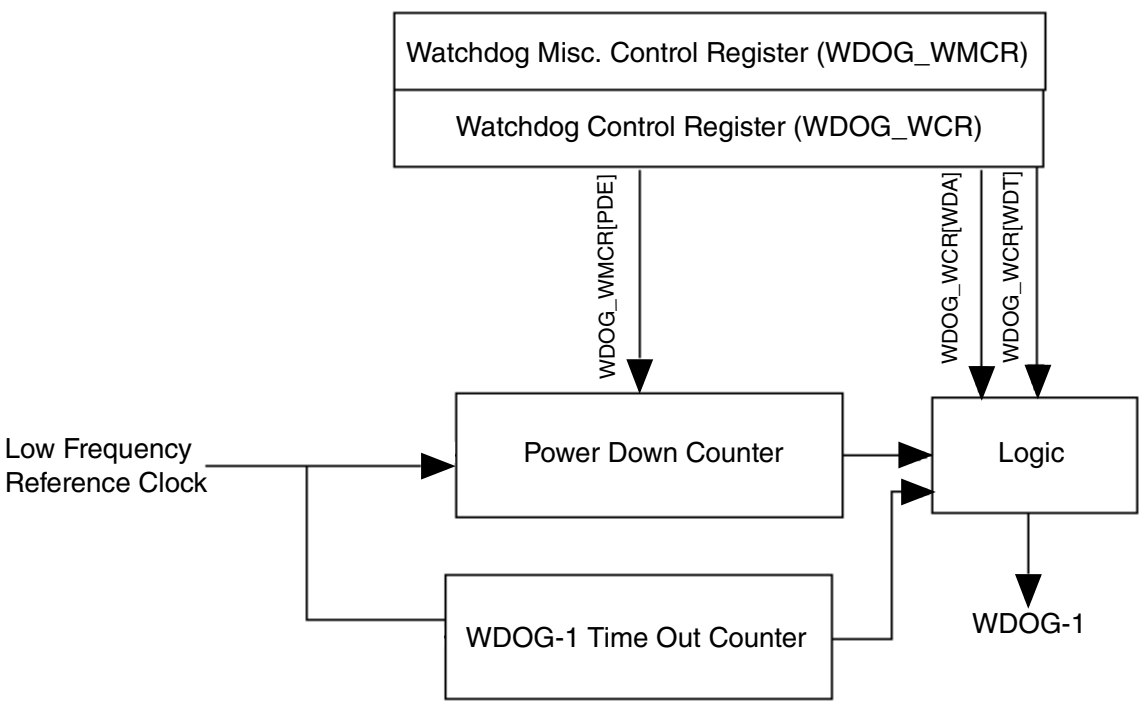


Figure 70-3. WDOG_B generation

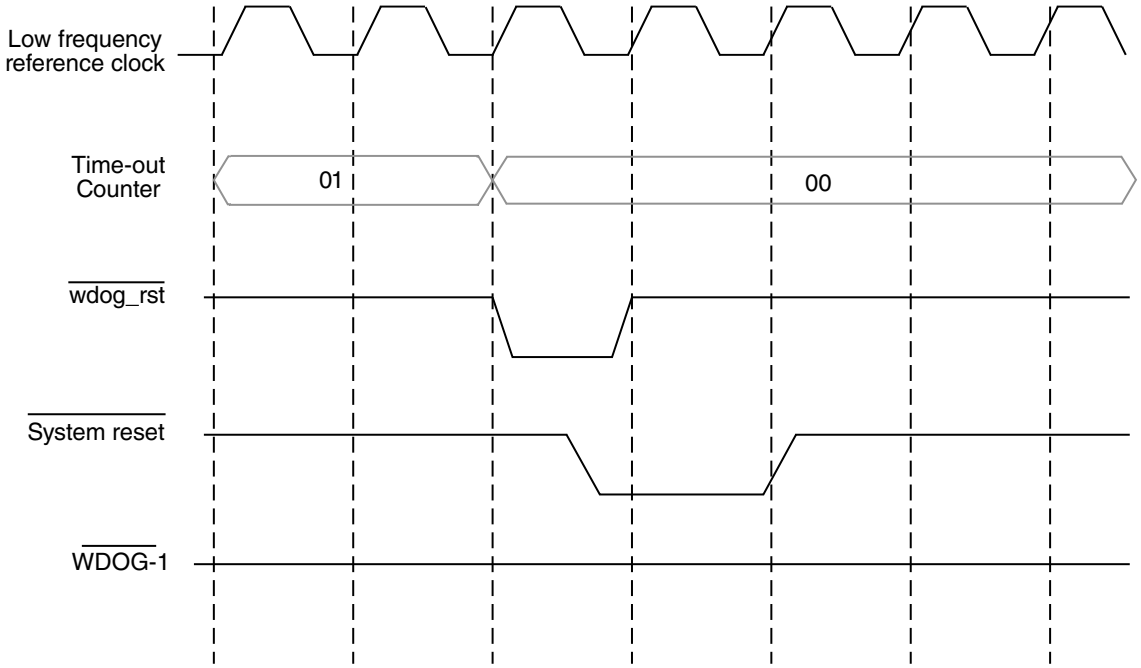


Figure 70-4. WDOG timeout condition/WDT bit is not set

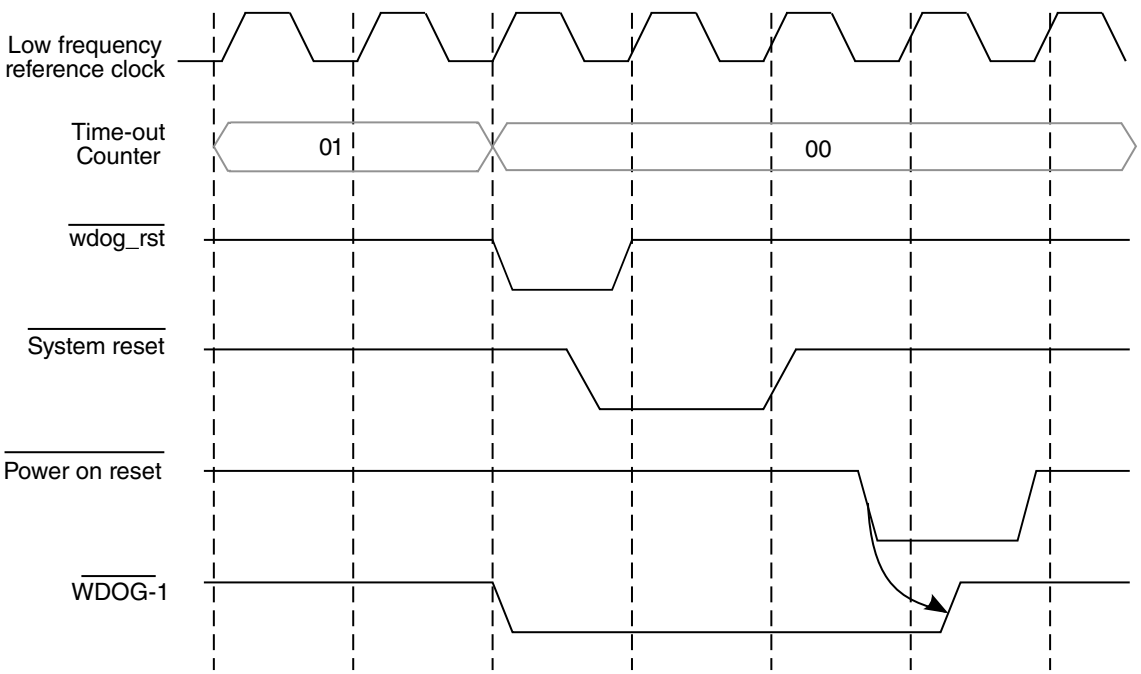


Figure 70-5. WDOG timeout condition/WDT bit is set

70.5.7 Reset

The block is reset by a system reset and the WDOG counter will be disabled. The power-down counter is enabled and starts counting.

70.5.8 Interrupt

The WDOG has the feature of Interrupt generation before timeout.

The interrupt will be generated only if the WIE bit in Watchdog Interrupt Control Register (WDOG_WICR) is set. The exact time at which the interrupt should occur (prior to timeout) depends on the value of WICT field of Watchdog Interrupt Control Register (WDOG_WICR). For example, if the WICT field has a value 0x04, then the interrupt will be generated two seconds prior to timeout. Once the interrupt is triggered the WTIS bit in Watchdog Interrupt Control Register (WDOG_WICR) will be set. The software needs to clear this bit to deassert the interrupt. If the WDOG is serviced before the interrupt generation then the counter will be reloaded with the timeout value WT[7:0] of Watchdog Control Register (WDOG_WCR) and interrupt would not be triggered.

70.5.9 Flow Diagrams

A flow diagram of WDOG operation is shown below.

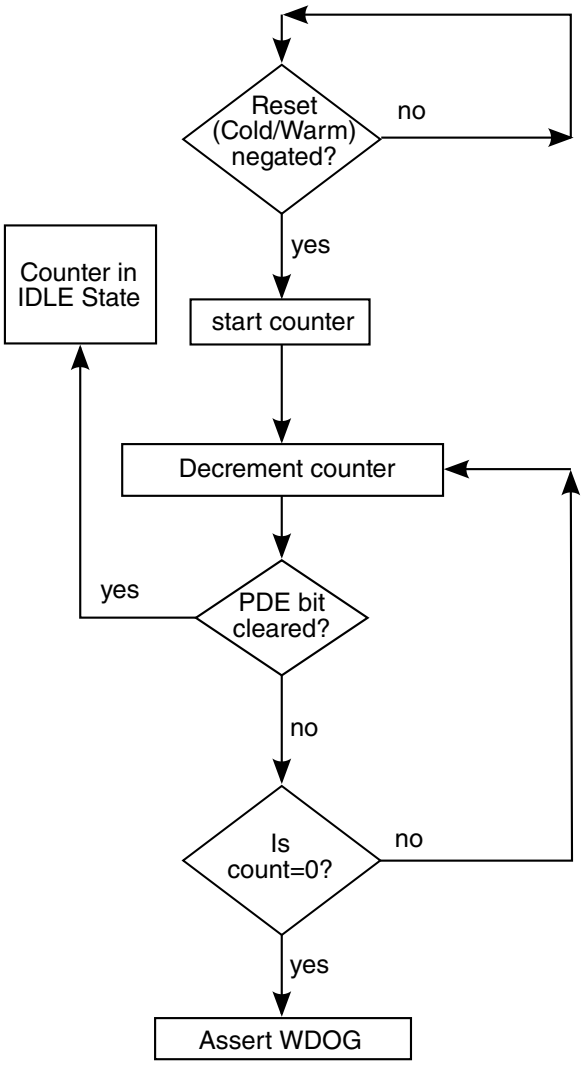


Figure 70-6. Power-Down Counter Flow Diagram

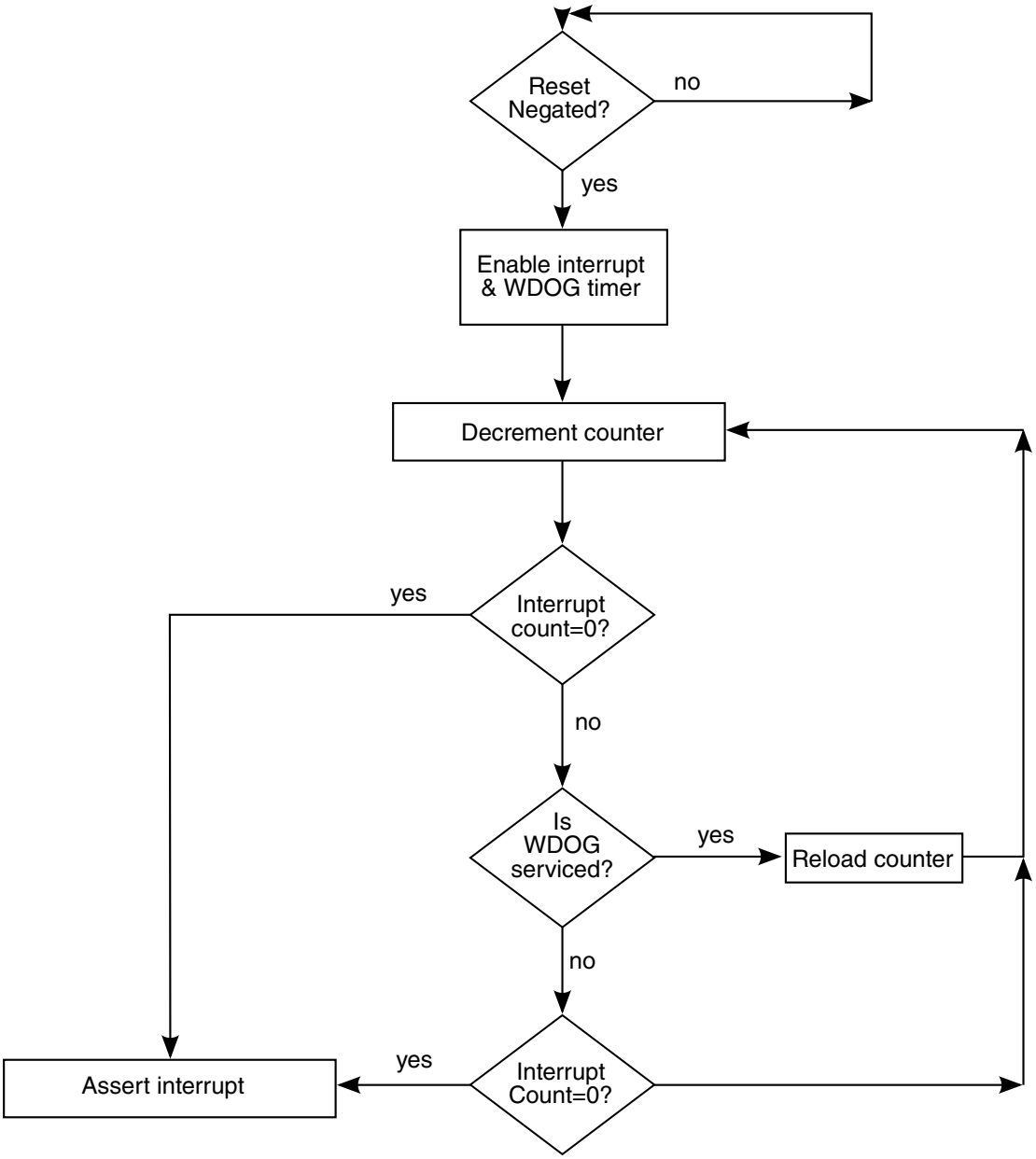


Figure 70-7. Interrupt Generation Flow Diagram

70.6 Initialization

The following sequence should be performed for WDOG initialization.

- PDE bit of Watchdog Miscellaneous Control Register (WDOG_WMCR) should be cleared to disable the power down counter.

- WT field of Watchdog Control Register (WDOG_WCR) should be programmed for sufficient timeout value.
- WDOG should be enabled by setting WDE bit of Watchdog Control Register (WDOG_WCR) so that the timeout counter loads the WT field value of Watchdog Control Register (WDOG_WCR) and starts counting.

70.7 WDOG Memory Map/Register Definition

The WDOG Memory Map/Register Definition can be found here.

The WDOG has user-accessible, 16-bit registers used to configure, operate, and monitor the state of the Watchdog Timer. Byte operations can be performed on these registers. If a 32-bit access is performed,the WDOG will not generate a peripheral bus error but will behave normally, like a 16-Bit access, making read/write possible. A 32-Bit access should be avoided, as the system may go to an unknown state.

WDOG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20B_C000	Watchdog Control Register (WDOG1_WCR)	16	R/W	0030h	70.7.1/5726
20B_C002	Watchdog Service Register (WDOG1_WSR)	16	R/W	0000h	70.7.2/5728
20B_C004	Watchdog Reset Status Register (WDOG1_WRSR)	16	R	0000h	70.7.3/5729
20B_C006	Watchdog Interrupt Control Register (WDOG1_WICR)	16	R/W	0004h	70.7.4/5730
20B_C008	Watchdog Miscellaneous Control Register (WDOG1_WMCR)	16	R/W	0001h	70.7.5/5731
20C_0000	Watchdog Control Register (WDOG2_WCR)	16	R/W	0030h	70.7.1/5726
20C_0002	Watchdog Service Register (WDOG2_WSR)	16	R/W	0000h	70.7.2/5728
20C_0004	Watchdog Reset Status Register (WDOG2_WRSR)	16	R	0000h	70.7.3/5729
20C_0006	Watchdog Interrupt Control Register (WDOG2_WICR)	16	R/W	0004h	70.7.4/5730
20C_0008	Watchdog Miscellaneous Control Register (WDOG2_WMCR)	16	R/W	0001h	70.7.5/5731

70.7.1 Watchdog Control Register (WDOGx_WCR)

The Watchdog Control Register (WDOG_WCR) controls the WDOG operation.

- WDZST, WDBG and WDW are write-once only bits. Once the software does a write access to these bits, they will be locked and cannot be reprogrammed until the next system reset assertion.
- WDE is a write one once only bit. Once software performs a write "1" operation to this bit it cannot be reset/cleared until the next system reset.
- WDT is also a write one once only bit. Once software performs a write "1" operation to this bit it cannot be reset/cleared until the next POR. This bit does not get reset/cleared due to any system reset.

Address: Base address + 0h offset

Bit	15	14	13	12	11	10	9	8
Read Write	WT							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	WDW	SRE	WDA	SRS	WDT	WDE	WDBG	WDZST
Reset	0	0	1	1	0	0	0	0

WDOGx_WCR field descriptions

Field	Description
15–8 WT	Watchdog Time-out Field. This 8-bit field contains the time-out value that is loaded into the Watchdog counter after the service routine has been performed or after the Watchdog is enabled. After reset, WT[7:0] must have a value written to it before enabling the Watchdog otherwise count value of zero which is 0.5 seconds is loaded into the counter. **NOTE:** The time-out value can be written at any point of time but it is loaded to the counter at the time when WDOG is enabled or after the service routine has been performed. For more information see Timeout event . 0x00 - 0.5 Seconds (Default). 0x01 - 1.0 Seconds. 0x02 - 1.5 Seconds. 0x03 - 2.0 Seconds. 0xff - 128 Seconds.
7 WDW	Watchdog Disable for Wait. This bit determines the operation of WDOG during Low Power WAIT mode. This is a write once only bit. 0 Continue WDOG timer operation (Default). 1 Suspend WDOG timer operation.
6 SRE	Software Reset Extension. Required to be set to 1 when used in conjunction with the Software Reset Signal (SRS). 0 Reserved 1 This bit must be set to 1.
5 WDA	WDOG_B assertion. Controls the software assertion of the WDOG_B signal. 0 Assert WDOG_B output. 1 No effect on system (Default).

Table continues on the next page...

WDOGx_WCR field descriptions (continued)

Field	Description
4 SRS	Software Reset Signal. Controls the software assertion of the WDOG-generated reset signal WDOG_RESET_B_DEB. This bit automatically resets to 1 after it has been asserted to 0. For proper operation of this function, the SRE bit in this register must be set to 1. **NOTE:** This bit does not generate the software reset to the block. 0 Assert system reset signal. 1 No effect on the system (Default).
3 WDT	WDOG_B Time-out assertion. Determines if the WDOG_B gets asserted upon a Watchdog Time-out Event. This is a write-one once only bit. **NOTE:** There is no effect on WDOG_RESET_B_DEB (WDOG Reset) upon writing on this bit. WDOG_B gets asserted along with WDOG_RESET_B_DEB if this bit is set. 0 No effect on WDOG_B (Default). 1 Assert WDOG_B upon a Watchdog Time-out event.
2 WDE	Watchdog Enable. Enables or disables the WDOG block. This is a write one once only bit. It is not possible to clear this bit by a software write, once the bit is set. **NOTE:** This bit can be set/reset in debug mode (exception). 0 Disable the Watchdog (Default). 1 Enable the Watchdog.
1 WDBG	Watchdog DEBUG Enable. Determines the operation of the WDOG during DEBUG mode. This bit is write once only. 0 Continue WDOG timer operation (Default). 1 Suspend the watchdog timer.
0 WDZST	Watchdog Low Power. Determines the operation of the WDOG during low-power modes. This bit is write once-only. **NOTE:** The WDOG can continue/suspend the timer operation in the low-power modes (STOP and DOZE mode). 0 Continue timer operation (Default). 1 Suspend the watchdog timer.

70.7.2 Watchdog Service Register (WDOGx_WSR)

When enabled, the WDOG requires that a service sequence be written to the Watchdog Service Register (WSR) to prevent the timeout condition.

NOTE

Executing the service sequence will reload the WDOG timeout counter.

Address: Base address + 2h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	WSR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOGx_WSR field descriptions

Field	Description
WSR	Watchdog Service Register. This 16-bit field contains the Watchdog service sequence. Both writes must occur in the order listed prior to the time-out, but any number of instructions can be executed between the two writes. The service sequence must be performed as follows: 0x5555 Write to the Watchdog Service Register (WDOG_WSR). 0xAAAA Write to the Watchdog Service Register (WDOG_WSR).

70.7.3 Watchdog Reset Status Register (WDOGx_WRSR)

The WRSR is a read-only register that records the source of the output reset assertion. It is not cleared by a hard reset. Therefore, only one bit in the WRSR will always be asserted high. The register will always indicate the source of the last reset generated due to WDOG. Read access to this register is with one wait state. Any write performed on this register will generate a Peripheral Bus Error .

A reset can be generated by the following sources, as listed in priority from highest to lowest:

- Watchdog Time-out
- Software Reset

Address: Base address + 4h offset

Bit	15	14	13	12	11	10	9	8
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	0			POR	0		TOUT	SFTW
Write								
Reset	0	0	0	0	0	0	0	0

WDOGx_WRSR field descriptions

Field	Description
15–5 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

WDOGx_WRSR field descriptions (continued)

Field	Description
4 POR	Power On Reset. Indicates whether the reset is the result of a power on reset. 0 Reset is not the result of a power on reset. 1 Reset is the result of a power on reset.
3–2 Reserved	This read-only field is reserved and always has the value 0.
1 TOUT	Timeout. Indicates whether the reset is the result of a WDOG timeout. 0 Reset is not the result of a WDOG timeout. 1 Reset is the result of a WDOG timeout.
0 SFTW	Software Reset. Indicates whether the reset is the result of a WDOG software reset by asserting SRS bit 0 Reset is not the result of a software reset. 1 Reset is the result of a software reset.

70.7.4 Watchdog Interrupt Control Register (WDOGx_WICR)

The WDOG_WICR controls the WDOG interrupt generation.

Address: Base address + 6h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	WIE	WTIS	0						WICT							
Write		w1c														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

WDOGx_WICR field descriptions

Field	Description
15 WIE	Watchdog Timer Interrupt enable bit. Reset value is 0. **NOTE:** This bit is a write once only bit. Once the software does a write access to this bit, it will get locked and cannot be reprogrammed until the next system reset assertion 0 Disable Interrupt (Default). 1 Enable Interrupt.
14 WTIS	Watchdog Timer Interrupt Status bit will reflect the timer interrupt status, whether interrupt has occurred or not. Once the interrupt has been triggered software must clear this bit by writing 1 to it. 0 No interrupt has occurred (Default). 1 Interrupt has occurred
13–8 Reserved	This read-only field is reserved and always has the value 0.
WICT	Watchdog Interrupt Count Time-out (WICT) field determines, how long before the counter time-out must the interrupt occur. The reset value is 0x04 implies interrupt will occur 2 seconds before time-out. The maximum value that can be programmed to WICT field is 127.5 seconds with a resolution of 0.5 seconds.

Table continues on the next page...

WDOGx_WICR field descriptions (continued)

Field	Description
	NOTE: This field is write once only. Once the software does a write access to this field, it will get locked and cannot be reprogrammed until the next system reset assertion. 0x00 WICT[7:0] = Time duration between interrupt and time-out is 0 seconds. 0x01 WICT[7:0] = Time duration between interrupt and time-out is 0.5 seconds. 0x04 WICT[7:0] = Time duration between interrupt and time-out is 2 seconds (Default). 0xff WICT[7:0] = Time duration between interrupt and time-out is 127.5 seconds.

70.7.5 Watchdog Miscellaneous Control Register (WDOGx_WMCR)

WDOG_WMCR Controls the Power Down counter operation.

Address: Base address + 8h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0															PDE
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

WDOGx_WMCR field descriptions

Field	Description
15–1 Reserved	This read-only field is reserved and always has the value 0.
0 PDE	Power Down Enable bit. Reset value of this bit is 1, which means the power down counter inside the WDOG is enabled after reset. The software must write 0 to this bit to disable the counter within 16 seconds of reset de-assertion. Once disabled this counter cannot be enabled again. See Power-down counter event for operation of this counter. **NOTE:** This bit is write-one once only bit. Once software sets this bit it cannot be reset until the next system reset. 0 Power Down Counter of WDOG is disabled. 1 Power Down Counter of WDOG is enabled (Default).

Chapter 71
Crystal Oscillator (XTALOSC)

71.1 Overview

This block comprises both the 24 MHz and 32 kHz implementation of a biased amplifier that when combined with a suitable external quartz crystal and external load capacitors, implements an oscillator.

The block includes means to:

- Accept an external clock source.
- Detect if the crystal frequency is close to 24 MHz or 32 kHz.
- Reduce the operating current via software after the oscillator has started (24 MHz specific feature)
- Supply another ~32 kHz clock source based off an independent internal oscillator if there is no oscillation sensed on the RTC_XTAL bumps(contacts) (32 kHz specific feature). The internal oscillator will provide clocks to the same on-chip modules as the external 32 kHz oscillator.
- Automatically switch to the external oscillation source when sensed on the RTC_XTAL bumps(contacts) (32 kHz specific feature).

71.2 External Signals

The table found here describes the external signals of XTALOSC:

Table 71-1. XTALOSC External Signals

Signal	Description	Pad	Mode	Direction
CLK1_N	Negative differential clock signal	CLK1_N	No Muxing	IO
CLK1_P	Positive differential clock signal	CLK1_P	No Muxing	IO
CLK2_N	Negative differential clock signal 2	CLK2_N	No Muxing	IO
CLK2_P	Positive differential clock signal 2	CLK2_P	No Muxing	IO

Table continues on the next page...

Table 71-1. XTALOSC External Signals (continued)

Signal	Description	Pad	Mode	Direction
REF_CLK_24M	24 MHz reference clock	GPIO_3	ALT3	O
		RGMII_TXC	ALT7	
REF_CLK_32K	32 KHz reference clock	GPIO_8	ALT1	O
		NVCC_SD2	No Muxing	
RTC_XTALI	Real-time clock crystal oscillator input	RTC_XTALI	No Muxing	I
RTC_XTALO	Real-time clock crystal oscillator output	RTC_XTALO	No Muxing	O
XTALI	Crystal oscillator input signal	XTALI	No Muxing	I
XTALO	Crystal oscillator output signal	XTALO	No Muxing	O

71.3 Crystal Oscillator 24 MHz

71.3.1 Oscillator Configuration (24 MHz)

The basic block diagram of the 24 MHz module configured as a crystal oscillator is shown below.

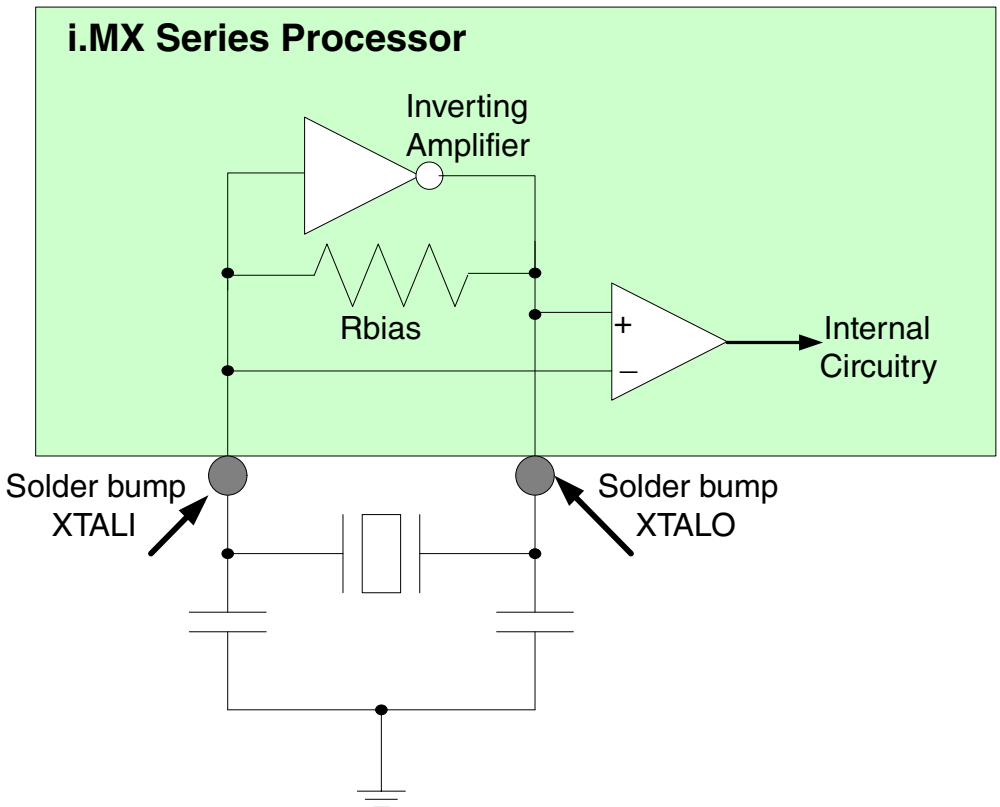


Figure 71-1. Oscillator Configuration (24 MHz)

This integrated biased amplifier can be used to create different frequency oscillators with different external component selection. However, care should be taken as many of the serial IO modules depend on the fixed frequency of 24 MHz. Please consult the sections of the document pertaining to the USB, ENET, PCIe interfaces, for example. After a healthy oscillation is established, then the bias current of the oscillator can generally be reduced to save power. This is accomplished through the XTALOSC24M_MISC0[OSC_I] bits, defined in the MISC0 register later in this chapter. Restore the XTALOSC24M_MISC0[OSC_I] bits before going into a power mode where the XTALOSC24 is powered down or oscillator startup may become an issue. The power down of the XTALOSC24 module is controlled by the CCM. See this section of the manual for more details.

71.3.2 Crystal Frequency Detection(24 MHz)

A submodule exists that gives a fairly crude (relative to the accuracy of a crystal) estimation of whether the clock frequency is correct.

This function may be enabled by setting the XTALOSC24M_MISC0[OSC_XTALOK_EN] bit. It is disabled at system reset. When the oscillator is stable and the correct frequency is detected, the XTALOSC24M_MISC0[OSC_XTALOK] bit will be set. Note that the correct frequency will be observed before the oscillator fully blooms(the oscillation waveform build-up is completed).

71.4 Crystal Oscillator 32 kHz

71.4.1 Oscillator Configuration (32 kHz)

The basic block diagram of the 32 kHz module configured as a crystal oscillator is shown below.

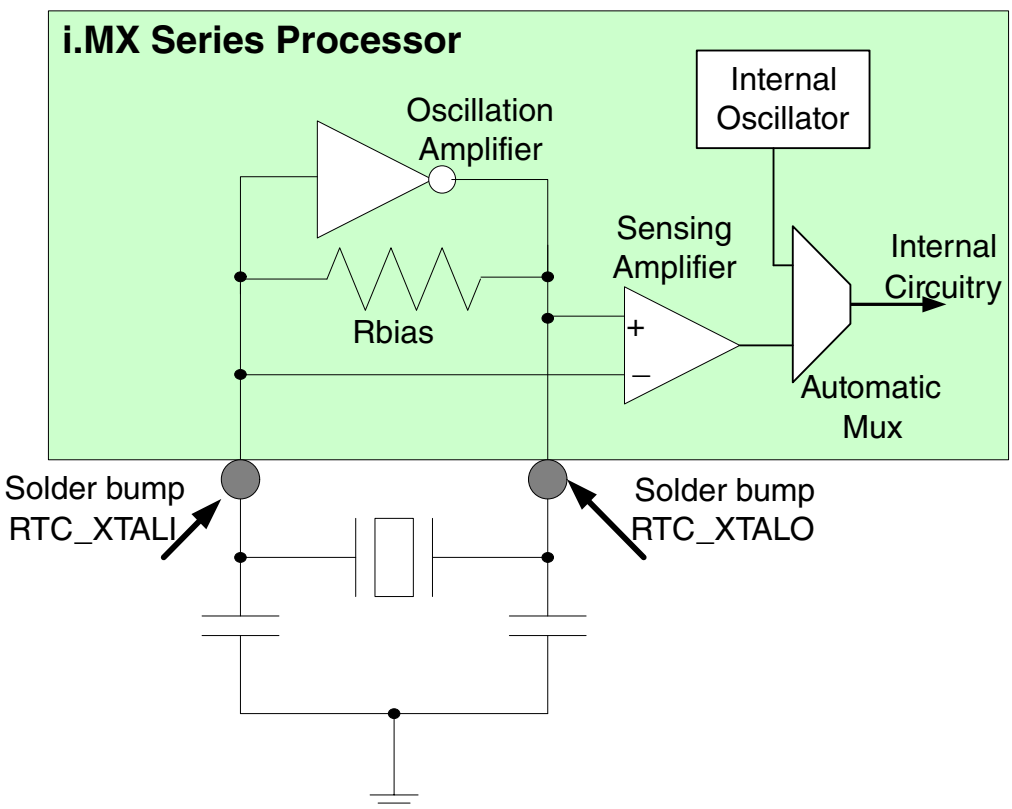


Figure 71-2. Oscillator Configuration (32 kHz)

This integrated biased amplifier can be used to create different frequency oscillators with different external component selection. Generally, RTC oscillators are either implemented with 32 kHz or 32.768 kHz crystals. Please consult the Security Reference Manual for appropriate frequency selection and configuration. Care must be taken to limit external leakage as this may debias the amplifier and degrade the gain.

The internal oscillator is automatically multiplexed in the clocking system when the system detects a loss of clock. The internal oscillator will provide clocks to the same on-chip modules as the external 32 kHz oscillator. The internal oscillator is not precise relative to a crystal. While it will provide a clock to the system, it generally will not be precise enough for long term time keeping. The internal oscillator is anticipated to be useful for quicker startup times and tampering prevention, but should not be used as the exclusive source for the 32 kHz clocks. An external 32 kHz clock source must be used for production systems.

71.4.2 Bypass Configuration (32 kHz)

If it is desired to drive the chip with an external clock source, then the 32 kHz oscillator could be driven in one of three configurations using a nominal 1.1V source.

1. A single ended external clock source can be used to overdrive the output of the amplifier (RTC_XTALO). Since the oscillation sensing amplifier is differential, the RTC_XTALI pin should be externally floating and capacitively loaded. The combination of the internal biasing resistor and the external capacitor will filter the signal applied to the RTC_XTALO pin and develop a rough reference for the sensing amplifier to compare to.
2. A single ended external clock source can be used to drive RTC_XTALI. In this configuration, RTC_XTALO should be left externally floating.
3. A differential external clock source can be used to drive both RTC_XTALI and RTC_XTALO.

Generally, configuration 2 is anticipated to be the most used configuration, but all three configurations may be utilized.

71.5 XTALOSC 24MHz Memory Map/Register Definition

NOTE

The register content is mixed with analog functions not related to the oscillator function. These bits are noted.

XTALOSC24M memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_8150	Miscellaneous Register 0 (XTALOSC24M_MISC0)	32	R/W	0400_0000h	71.5.1/5739
20C_8154	Miscellaneous Register 0 (XTALOSC24M_MISC0_SET)	32	R/W	0400_0000h	71.5.1/5739
20C_8158	Miscellaneous Register 0 (XTALOSC24M_MISC0_CLR)	32	R/W	0400_0000h	71.5.1/5739
20C_815C	Miscellaneous Register 0 (XTALOSC24M_MISC0_TOG)	32	R/W	0400_0000h	71.5.1/5739

71.5.1 Miscellaneous Register 0 (XTALOSC24M_MISC0*n*)

This register defines the control and status bits for miscellaneous analog blocks.

Address: 20C_8000h base + 150h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			CLKGATE_DELAY			CLKGATE_CTRL	Reserved					WBCP_VPW_THRESH		OSC_XTALOK_EN	OSC_XTALOK
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OSC_I		discon_high_snvs	STOP_MODE_CONFIG	Reserved				REFTOP_VBGUP	REFTOP_VBGADJ			REFTOP_SELFBIASOFF	Reserved		REFTOP_PWD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XTALOSC24M_MISC0*n* field descriptions

Field	Description
31–29 -	This field is reserved.
28–26 CLKGATE_ DELAY	This field specifies the delay between powering up the XTAL 24MHz clock and releasing the clock to the digital logic inside the analog block. **NOTE:** Do not change the field during a low power event. This is not a field that the user would normally need to modify. 000 0.5ms 001 1.0ms 010 2.0ms 011 3.0ms 100 4.0ms 101 5.0ms 110 6.0ms 111 7.0ms
25 CLKGATE_CTRL	This bit allows disabling the clock gate (always ungated) for the xtal 24MHz clock that clocks the digital logic in the analog block. **NOTE:** Do not change the field during a low power event. This is not a field that the user would normally need to modify. 0 **ALLOW_AUTO_GATE** — Allow the logic to automatically gate the clock when the XTAL is powered down. 1 **NO_AUTO_GATE** — Prevent the logic from ever gating off the clock.
24–20 -	This field is reserved. Always set to zero.
19–18 WBCP_VPW_ THRESH	This signal alters the voltage that the pwell is charged pumped to. **NOTE:** Not related to oscillator. 00 **NOMINAL_BIAS** — Nominal output pwell bias voltage. 01 **PLUS_25MV** — Increase pwell output voltage by 25mV. 10 **MINUS_25MV** — Decrease pwell output pwell voltage by 25mV. 11 **MINUS_50MV** — Decrease pwell output pwell voltage by 50mV.
17 OSC_XTALOK_ EN	This bit enables the detector that signals when the 24MHz crystal oscillator is stable.
16 OSC_XTALOK	Status bit that signals that the output of the 24-MHz crystal oscillator is stable. Generated from a timer and active detection of the actual frequency.
15–14 OSC_I	This field determines the bias current in the 24MHz oscillator. The aim is to start up with the highest bias current, which can be decreased after startup if it is determined to be acceptable. 00 **NOMINAL** — Nominal 01 **MINUS_12_5_PERCENT** — Decrease current by 12.5% 10 **MINUS_25_PERCENT** — Decrease current by 25.0% 11 **MINUS_37_5_PERCENT** — Decrease current by 37.5%

Table continues on the next page...

XTALOSC24M_MISC0*n* field descriptions (continued)

Field	Description
13 discon_high_snvs	This bit forces the short between VDDHIGH_IN and VSNVS_IN to open when asserted. This is useful in power cases where SNVS_IN > VDDHIGH_IN. **NOTE:** Not related to oscillator.
12 STOP_MODE_CONFIG	Configure the analog behavior in stop mode. **NOTE:** Not related to oscillator. 0x0 **DEEP** — Deep Stop Mode - 0x0 All analog except RTC powered down on Stop mode assertion 0x1 **LIGHT** — Light Stop Mode - 0x1 All the analog domain except the LDO_1P1, LDO_2P5, and PLL3 is powered down on STOP mode assertion. If required the CCM can be configured not to power down the oscillator (XTALOSC). PLL3 can be disabled with register settings if desired.
11–8 -	This field is reserved. Reserved
7 REFTOP_VBGUP	Status bit that signals the analog bandgap voltage is up and stable. 1 - Stable. **NOTE:** Not related to oscillator.
6–4 REFTOP_VBGADJ	**NOTE:** Not related to oscillator. 000 Nominal VBG 001 VBG+0.78% 010 VBG+1.56% 011 VBG+2.34% 100 VBG-0.78% 101 VBG-1.56% 110 VBG-2.34% 111 VBG-3.12%
3 REFTOP_SELFBIASOFF	Control bit to disable the self-bias circuit in the analog bandgap. The self-bias circuit is used by the bandgap during startup. This bit should be set after the bandgap has stabilized and is necessary for best noise performance of analog blocks using the outputs of the bandgap. **NOTE:** Value should be returned to zero before removing vddhigh_in or asserting bit 0 of this register (REFTOP_PWD) to assure proper restart of the circuit. **NOTE:** Not related to oscillator. 0 Uses coarse bias currents for startup 1 Uses bandgap-based bias currents for best performance.
2–1 -	This field is reserved.
0 REFTOP_PWD	Control bit to power-down the analog bandgap reference circuitry. **NOTE:** A note of caution, the bandgap is necessary for correct operation of most of the LDO, pll, and other analog functions on the die. **NOTE:** Not related to oscillator.

Appendix A
SDMA Scripts

A.1 Introduction

This appendix provides descriptions of scripts that may be used to perform data transfers using the Smart DMA (SDMA) block of the SoC.

The SDMA block supports data transfer from core memory space to core memory, from core memory space to peripherals, and vice versa.

A.1.1 SDMA Scripts Overview

Table A-1 gives an overview of the SDMA scripts.

Table A-1. SDMA Scripts Overview

Data Transfer	Script to Use	Use Case	Parameters Through Context	Parameters Through Buffer Descriptor
Memory	ap_2_ap	Memory Copy	None	Word length (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, first address is memory source address, second address is memory destination address
Shared UART	uartsh_2_mcu	Uart Rx	UART Rx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01), Counter, Memory destination address, Second address not used
UART	uart_2_mcu	Uart Rx	Uart Rx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01), Counter, Memory source address, Second address not used
SPDIF	mcu_2_spdif	Spdif Playback	SPDIF Tx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b10 or 2'b00), Counter, Memory source address, Second address not used

Table continues on the next page...

Table A-1. SDMA Scripts Overview (continued)

Data Transfer	Script to Use	Use Case	Parameters Through Context	Parameters Through Buffer Descriptor
	spdif_2_mcu	Spdif Record	SPDIF Rx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b10 or 2'b00), Counter, Memory source address, Second address not used
Shared Peripheral	mcu_2_shp	Peripheral Transmit	Tx fifo address (r6) Event_mask (r1) Event2_mask(r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory source address, Second address not used
	shp_2_mcu	Peripheral Receive	RX fifo address (r6) Event_mask (r1) Event2_mask(r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory destination address, Second address not used
Peripheral	mcu_2_app	Peripheral Transmit	Tx fifo address (r6) Event_mask (r1) Event2_mask(r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory source address, Second address not used
	app_2_mcu	Peripheral Receive	RX fifo address (r6) Event_mask (r1) Event2_mask(r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory destination address, Second address not used
SSI	mcu_2_ssiapp	Audio Playback	Tx FIFO 0 address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory source address, Second address not used
	ssiapp_2_mcu	Audio Record	Rx FIFO 0 address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory destination address, Second address not used
Shared SSI	mcu_2_ssish	Audio Playback	Tx FIFO 0 address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory source address, Second address not used
	ssish_2_mcu	Audio Record	Rx FIFO 0 address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Mem destination address, Second address not used
Peripheral	p_2_p	Peripheral Transfer	Destination address (r6) Event_mask (r1) Event2_mask (r0) Source address (r2) Information (r7)	Counter, First address and Second address not used
HDMI	hdmi_dma	HDMI playback	Buffer chain length (r4) Pointer address (r6)	None

In Table A-1, the data transfer column lists the possible types of DMA channels that involve the SDMA and a specific script is attached to each channel. It must be noted that some scripts cover several DMA channels. The scripts deal with the channels that have generic peripherals where only the watermark level is needed (no aging timer, no end_of_packet feature, and so on). The location refers to whether a script resides in ROM or in RAM and the size of the script is given in the instructions. The parameters columns are explained in the next section.

The following terms are used in Table A-1 which lists all the supported data transfers.

EMI or External Memory:

The external memories are connected to the External Memory Interface. Thus, a data transfer to EMI means a data transfer to any of the external memories (NAND Flash, NOR Flash, PSRAM, SDRAM, and so on).

Host mem or ARM internal memory:

The memory space accessible through the MAX of the ARM platform. It does not correspond to the peripherals, although they are accessible through the MAX as well. A data transfer from Host mem means data is read from the internal memory of the ARM or from the external memory. It is possible to point to an external memory through the MAX because it is also connected to EMI module. The next diagram replaces the SDMA and its DMA port in the hardware architecture. The Host mem is accessible through the Per DMA port of the SDMA, the EMI is accessible through the Burst DMA and the DSP mem through the DSP port.

Shared Peripheral:

A same peripheral can be connected to the shared peripheral bus (output of SPBA module) and to the ARM platform. This is the case for some UART, CSPI, and SSI. Therefore, "shared UART" indicates that the UART is connected to the shared peripheral, whereas "UART" means the UART is connected to the ARM platform.

Figure A-1 gives an overview of the SDMA in its hardware environment.

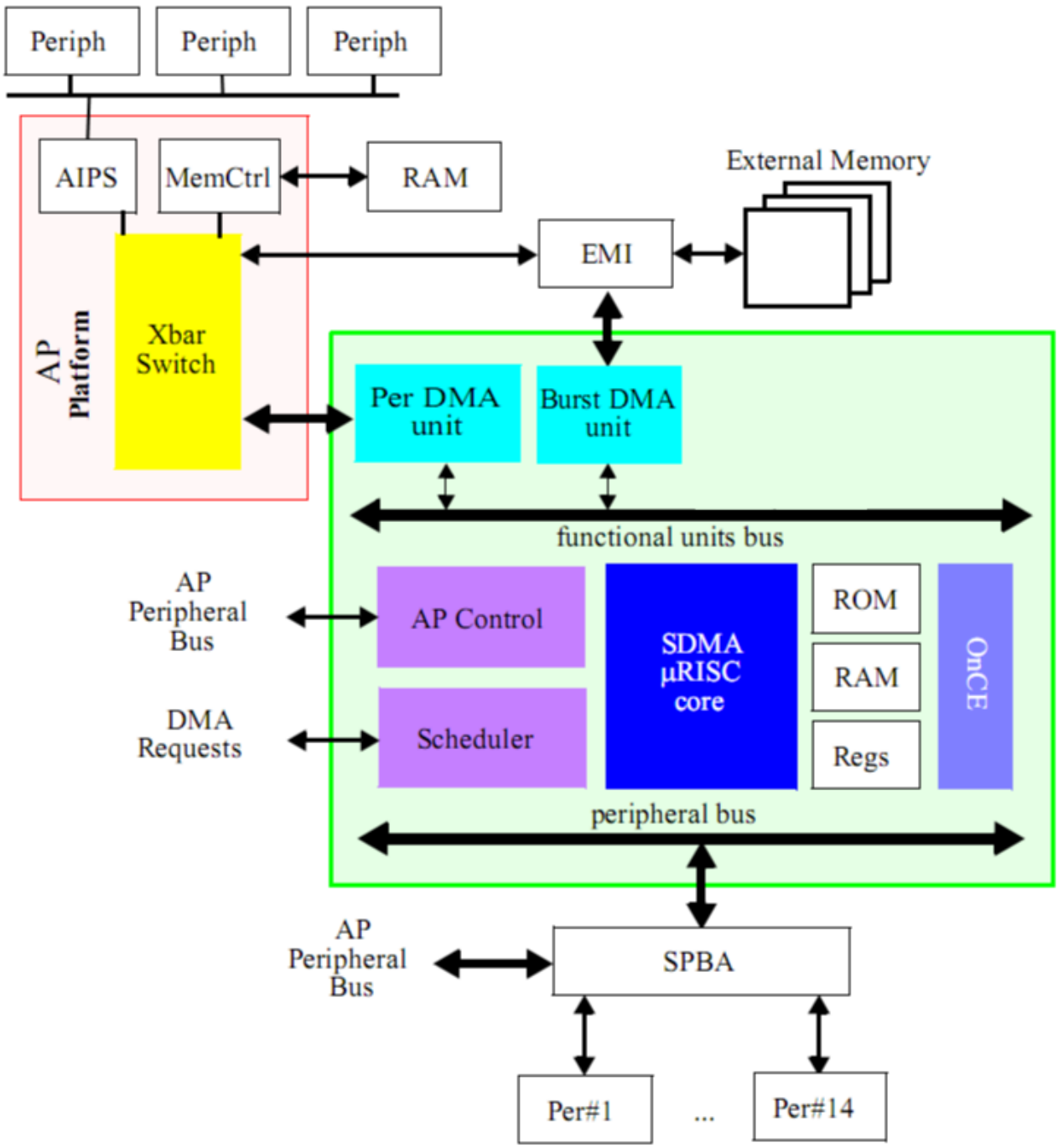


Figure A-1. SDMA Connections

A.2 Scripts Parameters: Definition

A.2.1 Parameters Definition

A.2.1.1 Parameters Required by Data Transfer Script Involving a Peripheral

The scripts that have a peripheral as a player in the transfer of data need to have the following input parameters:

A.2.1.1.1 Watermark Level (WML)

It is the upper or the lower threshold of the receive or transmit FIFO of the peripheral that triggers a DMA request to the SDMA.

The WML determines the data transfer loop size, meaning the number of bytes that will be read/write from/to the receive/transmit FIFO. This parameter must be a multiple of the peripheral FIFO data size. As an example, if UART1 is the data transfer destination, data must be written to the Tx FIFO of the peripheral. For instance, the watermark level can be set to 8 bytes (the UART is a 1 byte FIFO data size), meaning the UART_TX DMA request will be sent to the SDMA each time there are more than 8 free locations in the FIFO; therefore SDMA loop size will be set to 8 and 8 bytes will be written to the UART_TX FIFO. In fact, the loop size is the minimum between the WML and the count field of the buffer descriptor attached to the channel. This concept is explained later on in the chapter. Similarly, if the UART is the source of a data transfer and if the WML is set to 16, each time the DMA request is received by the SDMA, which triggers the associated channel, the data transfer loop size is set to 16. This means 16 bytes will be read from the UART_RX FIFO. The watermark level is a "long" (32-bit data) programmed by the ARM and is not supposed to evaluate a lot during application. The WML is always given in bytes.

NOTE

For the transmit FIFO, the WML = FIFO_SIZE - FIFOTX_THRESHOLD For the receive FIFO, the WML = FIFORX_THRESHOLD The FIFOTX_THRESHOLD/ FIFORX_THRESHOLD are the values of the peripheral transmit and receive FIFO level at which a DMA request is triggered. For instance, if TX FIFO is 32 bytes depth and if the DMA request is sent when the number of bytes present in the FIFO is below a threshold of 10, the WML will be 22. It means that when the SDMA will receive the DMA request from the

peripheral, it will be possible to store 22 bytes in the TX FIFO.
For the receive FIFO, the WML equals the threshold.

A.2.1.1.2 Event_mask and Event2_mask

As previously said, when the WML threshold is over or under passed, a DMA request is sent to the scheduler part of the SDMA.

The SDMA scheduler has 48 input pins, called "events", which are connected to the different DMA request.

The mapping between the DMA request name and the event number is SoC dependent and defined in the "Interrupts and SDMA Events" chapter. For instance, in SoC A, the UART_RX DMA request is connected to events pin 5; whereas for SoC B, it may be connected to events pin 15. The script reads the EVENTS(32 events requests) and EVENTS2(remaining 16 events requests) register to check if the received DMA request is compliant with the expected one and it guaranteed that a second DMA request sent during the data transfer is not lost.

For instance, while SDMA is reading the UART_RX FIFO (again, the number of data to be read is given by WML value), the peripheral may receive new data from its input ports. Therefore after the first data transfer, the number of data available in the UART received FIFO can be still higher than the WML threshold, so a second DMA request may be sent.

The event_mask value is generally a 1-bit high value, which means that if the script attached to the channel must be triggered by DMA request number I, event_mask[I] must be set to 1, if the script attached to the channel must be triggered by DMA request number 32+I, event2_mask[I] must be set to 1. Some scripts (like ata_2_mcu) may be triggered by several DMA requests; in that case several bits of the event_mask must be asserted. The event_mask and event2_mask are long (32-bit) data.

A.2.1.1.3 Peripheral Address

The scripts of the SDMA scripts library are SoC independent but as their goal is to address peripheral, they required the base address or the FIFO address of the peripheral.

Passing FIFO or peripheral base address depends on the scripts.

A.2.1.1.4 Data Length

The scripts whose name contains the key word "app" or "shp" are generic: they are used to transfer data from/to a 8, 16, 24 or 32-bit peripheral data size.

Some jumps to some routines of these scripts depend on this peripheral size. This parameter is required as input of these scripts. This parameter is passed through the command field of the buffer descriptor and is coded on bits 25 and 24 of the mode:

00: 32-bit data transfer.

01: 8-bit data transfer.

10: 16-bit datatransfer.

11: 24-bit data transfer.

A.3 Scripts

The following scripts are all based on buffer descriptor mechanism.

A.3.1 SDMA ROM Scripts

The ROM holds the generic memory to memory scripts (ap_2_ap, ap_2_bp, bp_2_ap, bp_2_bp) and the peripheral most useful scripts (app_2_mcu, mcu_2_app, uart_2_mcu, uartsh_2_mcu, mcu_2_shp, shp_2_mcu).

In these memory to memory scripts, the number of bytes to transfer from the source memory to the destination is divided by 4 to get the number of 32-bit words that can be transferred. Most of the time, the transfer will be a transfer of words. There will be a byte or half data transfer at the end of the transfer if the total number of bytes to transfer is not an multiple of 4.

A.3.1.1 ap_2_ap

This script is used to transfer data inside the application processor memory using BurstDMA.

Using this script there is no restriction on the number of bytes to transmit, the source address and destination address alignment, or on the source memory and destination memory type. But word alignment and source/destination memory type will impact performance. In case that the source and destination are in the same memory type, being word aligned allows the SDMA to use the copy capability of the DMA units. When it is not the case the SRAM of the SDMA is used as temporary buffer.

A.3.1.1.1 Parameters Transmitted Through the Context ap_2_ap

A.3.1.1.2 Parameters Transmitted Through the Buffer Descriptor ap_2_ap

The number of bytes to transmit is stored in the first Buffer descriptor word.

The source address is stored in the second Buffer descriptor word (address field).

The destination address is stored in the Extended Buffer address

A.3.1.1.3 Use of the General Register During the Execution ap_2_ap

- r0: nb of bytes to transfer, counter during loop
- r2: channel cb address, scratch
- r3: current buffer descriptor pointer, AP destination address, scratch
- r4: mode
- r5: source address loaded by getbd,
- r6: destination address loaded by getcb, scratch
- r7: pointer in the scratch buffer

A.3.1.1.4 Overview of Script Functionality ap_2_ap

The parameters of the transfer (number of bytes to transfer, source address and destination address) are retrieved from the buffer descriptor.

Depending on the address alignment and the DMA units involved, if possible, the transfer is performed using the copy capability of the DMA unit, or using the SDMA SRAM as a temporary buffer.

When the transfer is finished, this algorithm restarts (a new buffer descriptor is read).

The count in the BD mode word after the script is over is meaningful only if error was reported, else it has the same value as was fed as input to the script.

A.3.1.2 uartsh_2_mcu

This script is used to transfer data from the shared UART to the External memory. The UART is connected to the ARM platform.

The UART is an 8-bit peripheral, but when reading the FIFO, 16 bits are retrieved. In fact, the 8 MSB are the status bytes for the current data. At each access the value of this byte is checked to verify that a valid data was retrieved. If an error is detected the transfer is stopped and the information is sent back to the Host through the buffer descriptor with the byte count field updated.

A.3.1.2.1 Parameters Transmitted Through the Context uartsh_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the peripheral Rx FIFO (project dependent)

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.3.1.2.2 Parameters Transmitted Through the Buffer Descriptor uartsh_2_mcu

The number of bytes to transmit is stored in the first Buffer descriptor word.

The destination address is stored in the second Buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.1.2.3 Use of the General Register During the Execution uartsh_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = AP M3 destination address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.1.2.4 Overview of Script Functionality uartsh_2_mcu

When the UART sends a DMA request, it can be for three different reasons.

When the RX threshold is over passed. It deals with the most frequent event.

When the AGEING timer has reached its final value, meaning there is less data than the RX threshold in the RX FIFO, and they have been present for too long. It is called the aging DMA request.

When IDLE timer reached its final value; meaning the RX FIFO has been empty for too long. It is called the iddle dma request. The IDLE timer is not used, so only normal and AGEING DMA requests are supported by the script.

The first time the script is executed, a buffer descriptor is opened, the buffer destination address and its size are retrieved from the BD.

Then the script checks if the RRDY bit of USR1 is set. If yes, it means that the Watermark level has been reached, there are "WML" bytes in RX FIFO. If the count field of the opened BD is higher than WML-1, the SDMA script will perform WML-1 read accesses to the RX FIFO and WML write access to the destination buffer (a read access then a write access is call a data transfer loop). If the count field is lesser than the WML-1 value, the count field will be the data transfer loop size. So the data transfer loop size equals min(WML-1,BD count)

So there will be always one data remaining in the UART RX FIFO after every data transfer loop and the ageing timer will always trigger a DMA request. It means that the channel on which the script is run will be always closed on an AGEING DMA request.

Now assume the script is triggered by a DMA request, but RRDY is not set. It means that it deals with an AGEING DMA request; there is at least one data in the RX FIFO. The data is read and written into the destination buffer, and then the RDR is checked again. If it is set and if the destination data buffer is not full (BD count is not 0), a new data is read and so on until RDR is 0. When RDR is 0, the ageing timer interrupt flag is disabled by setting the AGTIM bit, the count field of BD is updated and the current BD is closed. Then the script tries to open the next BD if the Continuous bit of the current BD was set, if there is no more BD to open, the channel is stopped.

When reading a data from the UART RX FIFO, the MSB are check to verify that a valid data has been retrieved. If an error is detected the transfer is stopped, the error bit is set in the BD, the count is updated, the BD is closed and an interrupt is sent to the ARM.

When the buffer is full, this algorithm restarts (a new buffer descriptor is read, if it exists).

The next diagram illustrates the script algorithm, it is quite complex but things become more obvious when we notice that there is a loop for the normal DMA request (in sky-blue) and on when the AGEING DMA request was detected (in yellow).

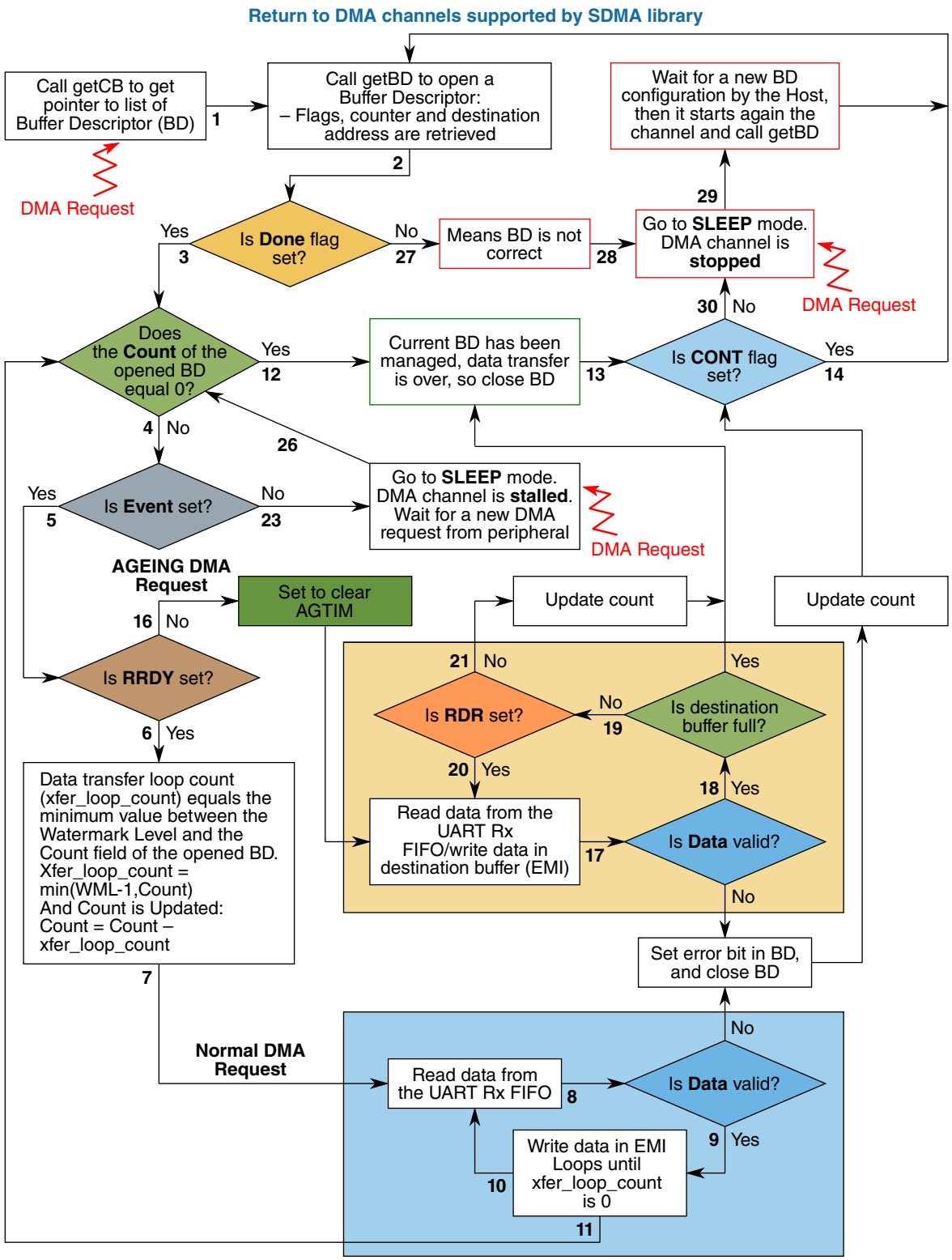


Figure A-2. Script Algorithm

A.3.1.3 shp_2_mcu

This generic script is used to transfer data from a 8, 16, 24 or 32-bit peripheral connected to the shared peripheral bus accessed through the SPBA to memories accessed by the BurstDMA (External memories).

It can be used for SSI (8, 16, 24 or 32-bit data size), for CSPI (32-bit data size) or for SDHC(MMC)/SIM (16-bit data size), these peripherals being connected to the shared peripheral bus.

A.3.1.3.1 Parameters Transmitted Through the Context shp_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the peripheral Rx FIFO (project dependent)

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.3.1.3.2 Parameters Transmitted Through the Buffer Descriptor shp_2_mcu

The peripheral size/data length is set in the command field of the first Buffer descriptor word, especially bits 24 and 25.

The number of bytes to transmit is stored in the first Buffer descriptor word (count field)

The destination address in the external memory is stored in the second Buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.1.3.3 Use of the General Register During the Execution shp_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = BD destination address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.1.3.4 Overview of Script Functionality shp_2_mcu

The parameters of the transfer (peripheral size, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the peripheral sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" data in the FIFO.

During this data transfer loop, the number of bytes that are transferred from the RX FIFO of the peripheral to the EMI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory write access is fixed by the parameter peripheral size.

The necessary number of transfer loops are executed in order to fill the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is filled, this algorithm restarts (a new buffer descriptor is read).

A.3.1.4 mcu_2_shp

This generic script is used to transfer data from memories accessed by the BurstDMA (External memories) to a 8, 16, 24 or 32-bit peripheral connected to the shared peripheral bus accessed through the SPBA

. It can be used for SSI (8, 16, 24 or 32-bit data size), for CSPI (32-bit data size), for SDHC(MMC)/SIM (16-bit data size) or for UART3,4 (8-bit data size), these peripherals being connected to the shared peripheral bus.

A.3.1.4.1 Parameters Transmitted Through the Context mcu_2_shp

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check events - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the peripheral Tx FIFO (project dependent)

r7: Watermark level - Used to determine the maximum of data that can be retrieved from the peripheral each time the channel is started.

A.3.1.4.2 Parameters Transmitted Through the Buffer Descriptor mcu_2_shp

The peripheral size/data length is set in the command field of the first Buffer descriptor word, especially bits 24 and 25.

The number of bytes to transmit is stored in the first Buffer descriptor word (count field)

The source address in the external memory is stored in the second Buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.1.4.3 Use of the General Register During the Execution mcu_2_shp

r0 = mask to check events2

r1 = mask to check events

r2 = EVENTS / EVENTS2

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.1.4.4 Overview of Script Functionality mcu_2_shp

The parameters of the transfer (peripheral size, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the peripheral sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" empty rooms in the FIFO.

During this data transfer loop, the number of bytes that are transferred from the EMI to the TX FIFO of the peripheral is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory read access is fixed by the parameter peripheral size.

The necessary number of transfer loops are executed in order to empty the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is empty, this algorithm restarts (a new buffer descriptor is read).

A.3.1.5 uart_2_mcu

This script is similar to uartsh_2_mcu script. The only difference between these scripts is the way SDMA access UART RX FIFO.

In case of uartsh_2_mcu, the UART RX FIFO can directly be accessed by SDMA as it is on Shared Peripheral Bus. In case of uart_2_mcu, SDMA script uses one of its DMA (Peripheral DMA) to access UART RX FIFO.

A.3.1.6 app_2_mcu

This generic script is used to transfer data from a 8, 16, 24 or 32-bit peripheral connected to the AIPS accessed through the Periphera DMA of SDMA, to memories accessed by the BurstDMA (External memories).

It can be used for SSI (8, 16, 24 or 32-bit data size), for CSPI (32-bit data size) or for SDHC(MMC)/SIM (16-bit data size).

A.3.1.6.1 Parameters Transmitted Through the Context app_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the peripheral Rx FIFO (project dependent)

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.3.1.6.2 Parameters Transmitted Through the Buffer Descriptor app_2_mcu

The Data Length peripheral size/data length is set in the command field of the first buffer descriptor word, specially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field).

The destination address in the external memory is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.1.6.3 Use of the General Register During the Execution app_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = AP M3 destination address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.1.6.4 Overview of Script Functionality app_2_mcu

The parameters of the transfer (peripheral size, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the peripheral sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" data in the FIFO.

During this data transfer loop, the number of bytes that are transferred from the RX FIFO of the peripheral to the EMI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory write access is fixed by the parameter peripheral size.

The necessary number of transfer loops are executed in order to fill the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is filled, this algorithm restarts (a new buffer descriptor is read).

A.3.1.7 mcu_2_app

This generic script is used to transfer data from memories accessed by the BurstDMA (External memories), to a 8, 16, 24 or 32-bit peripheral connected to the AIPS accessed through the Periphera DMA of SDMA.

It can be used for SSI (8, 16, 24 or 32-bit data size), for CSPI (32-bit data size) or for SDHC(MMC)/SIM (16-bit data size).

A.3.1.7.1 Parameters Transmitted Through the Context mcu_2_app

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the peripheral Tx FIFO (project dependent)

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.3.1.7.2 Parameters Transmitted Through the Buffer Descriptor mcu_2_app

The peripheral size/data length is set in the command field of the first buffer descriptor word, specially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field).

The source address in the external memory is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.1.7.3 Use of the General Register During the Execution mcu_2_app

r0 = mask to check events2

r1 = mask to check events

r2 = TX FIFO address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.1.7.4 Overview of Script Functionality mcu_2_app

The parameters of the transfer (peripheral size, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the peripheral sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" empty room in the FIFO.

During this data transfer loop, the number of bytes that are transferred from the EMI to the TX FIFO of the peripheral is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory read access is fixed by the parameter peripheral size.

The necessary number of transfer loops are executed in order to fill the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is filled, this algorithm restarts (a new buffer descriptor is read).

A.3.1.8 mcu_2_spdif

This script performs a data move from AP buffer to 24 bits FIFO of the SPDIF.

This peripheral is on the peripheral bus, the SDMA accesses it through the SPBA. The MCU buffer is accessed through the burst DMA.

A.3.1.8.1 Parameters Transmitted Through the Context mcu_2_spdif

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check events - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: Address of SPDIF STL register

r7: Watermark level - Used to determine the maximum data that can be sent to the peripheral each time the channel is started.

A.3.1.8.2 Parameters Transmitted Through the Buffer Descriptor mcu_2_spdif

The data length (Either 32-bit or 16-bit) is set in the command field of the first Buffer descriptor word, especially bits 24, 25.

The number of bytes to transmit is stored in the first Buffer descriptor word (count field)

The source address in the external memory is stored in the second Buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.1.8.3 Use of the General Register During the Execution mcu_2_spdif

r0 = mask to check events2

r1 = mask to check events

r2 = EVENTS / EVENTS2

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.1.8.4 Overview of Script Functionality mcu_2_spdif

The parameters of the transfer (data length, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the spdif sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" of empty room in the FIFO (Sum of left and right FIFO's).

During this data transfer loop, the number of bytes that are transferred from the EMI to the TX FIFO of the spdif is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory read access is fixed by the parameter data length.

The necessary numbers of transfer loops are executed in order to empty the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is empty, this algorithm restarts (a new buffer descriptor is read).

A.3.1.9 spdif_2_mcu

This script performs a data move from 24 bits FIFO of the SPDIF to AP buffer.

This peripheral is on the peripheral bus, the SDMA accesses it through the SPBA. The MCU buffer is accessed through the burst DMA.

A.3.1.9.1 Parameters Transmitted Through the Context spdif_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check events - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: Address of SPDIF SRL register

r7: Watermark level - Used to determine the maximum data that can be sent to the peripheral each time the channel is started.

A.3.1.9.2 Parameters Transmitted Through the Buffer Descriptor spdif_2_mcu

The data length (Either 32 bit or 16 bit) is set in the command field of the first Buffer descriptor word, especially bits 24, 25.

The number of bytes to transmit is stored in the first Buffer descriptor word (count field)

The destination address in the external memory is stored in the second Buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.1.9.3 Use of the General Register During the Execution spdif_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = EVENTS / EVENTS2

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.1.9.4 Overview of Script Functionality spdif_2_mcu

The parameters of the transfer (data length, number of bytes to transfer and destination address) are retrieved from the buffer descriptor.

When the spdif sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" date in the FIFO (Sum of left and right FIFO's).

During this data transfer loop, the number of bytes that are transferred from the RX FIFO of the spdif to the EMI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory write access is fixed by the parameter data length.

The necessary numbers of transfer loops are executed in order to empty the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is empty, this algorithm restarts (a new buffer descriptor is read).

A.3.2 SDMA RAM scripts

A.3.2.1 mcu_2_ssiapp

This script performs a data move from AP buffer to 8, 16, 24 or 32-bit FIFO of the SSI in the AP region with 2 FIFOs enabled.

The MCU buffer is accessed through the burst DMA, while SSI is accessed with Per DMA.

A.3.2.1.1 Parameters Transmitted Through the Context mcu_2_ssiapp

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check events - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: Address of SSI Transmit FIFO 0 register

r7: Watermark level - Used to determine the maximum data that can be sent to the peripheral each time the channel is started.

A.3.2.1.2 Parameters Transmitted Through the Buffer Descriptor mcu_2_ssiapp

The data length (8, 16, 24 or 32-bit) is set in the command field of the first buffer descriptor word, especially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field).

The source address in the memory is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.2.1.3 Use of the General Register During the Execution mcu_2_ssiapp

r0 = mask to check events2

r1 = mask to check events

r2 = Tx FIFO address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.2.1.4 Overview of Script Functionality mcu_2_ssiapp

The parameters of the transfer (data length, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the SSI sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" of empty room in the FIFO (Sum of left and right FIFO's).

During this data transfer loop, the number of bytes that are transferred from the EMI to the TX FIFO of the SSI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory read access is fixed by the parameter data length.

The necessary numbers of transfer loops are executed in order to empty the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is empty, this algorithm restarts (a new buffer descriptor is read).

A.3.2.2 ssiapp_2_mcu

This script performs a data move from the 8, 16, 24 or 32-bit FIFO of the SSI to the AP buffer.

The SSI is in the AP region and 2 FIFO enabled. The MCU buffer is accessed through the burst DMA while the SSI is accessed through Per DMA.

A.3.2.2.1 Parameters Transmitted Through the Context ssiapp_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the SSI RX FIFO 0

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.3.2.2.2 Parameters Transmitted Through the Buffer Descriptor ssiapp_2_mcu

The data length (8, 16, 24, 32 bits) is set in the command field of the first buffer descriptor word, especially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field)

The destination address is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used

A.3.2.2.3 Use of the General Register During the Execution ssiapp_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = AP M3 destination address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.2.2.4 Overview of Script Functionality ssiapp_2_mcu

When the SSI sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" of data in the FIFO(sum of left and right FIFOs).

During this data transfer loop, the number of bytes that are transferred from the RX FIFO of the SSI to the EMI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory write access is fixed by the parameter data length.

The necessary numbers of transfer loops are executed in order to fill the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is full, this algorithm restarts (a new buffer descriptor is read).

A.3.2.3 mcu_2_ssish

This script is similar to mcu_2_ssiapp.

The only difference is the way SDMA core accesses the SSI TX FIFO. For script of mcu_2_ssiapp, SDMA core accesses the TX FIFO through Functional Unit Bus and regards it as the peripheral DMA, whereas for the mcu_2_ssish, the access is through the Shared Peripheral Bus and SSI can be directly accessed by SDMA.

A.3.2.4 ssish_2_mcu

This script is similar to ssiapp_2_mcu. The only difference is the way SDMA core accesses the SSI RX FIFO.

For script of ssiapp_2_mcu, SDMA core accesses the RX FIFO through Functional Unit Bus and regards it as the peripheral DMA, whereas for the ssish_2_mcu, the access is through the Shared Peripheral Bus and SSI can be directly accessed by SDMA.

A.3.2.5 p_2_p

This script performs a data move of 32 bits from the peripheral's FIFO connected either on SPBA or AIPS bus to another.

If destination FIFO belongs to ASRC then pad adding is performed after every N samples from the source device. If the source FIFO belongs to ASRC then after transmitting N samples a pad is swallowed. In case there is a transaction between ASRC(source) and SPDIF (destination) no pad adding or swallowing is required because SPDIF expects even number of samples.

This script presently does not deal with the transactions involving two devices connected to AIPS bus.

A.3.2.5.1 Parameters Transmitted Through the Context p_2_p

r0: LWML event mask

r1: HWML event mask

r2: source address

r6: destination address

r7: INFO. See details in the following table.

p_2_p (r7 INFO)

Bits	Name	Description
0-7	Lower WML	Watermark Level
8	PS	1 : Pad Swallowing 0 : No Pad swallowing
9	PA	1 : Pad Adding 0 : No Pad swallowing
10	SPDIF	If this bit is set both source and destination are on SPBA
11	Source Bit(SP)	1 : Source on SPBA 0 : Source on AIPS
12	Destination Bit (DP)	1 : Destination on SPBA 0 : Destination on AIPS
13-15	----------	MUST BE 0
16-23	Higher WML	HWML
24-27	N	Total number of samples after which Pad adding/Swallowing must be done. It must be odd.
28	Lower WML Event (LWE)	SDMA events reg to check for LWML event mask 0 : LWE in EVENTS register 1 : LWE in EVENTS2 register
29	Higher WML Event(HWE)	SDMA events reg to check for HWML event mask 0 : HWE in EVENTS register 1 : HWE in EVENTS2 register
30	----------	MUST BE 0
31	CONT	1 : Amount of samples to be transferred is unknown and script will keep on transferring samples as long as both events are detected and script must be manually stopped by the application 0 : The amount of samples to be is equal to the count field of mode word

A.3.2.5.2 Parameters Transmitted Through the Buffer Descriptor p_2_p

In the first buffer descriptor word, the mode and number of types to transfer is included.

The Buffer address and Extended Buffer address are not used.

A.3.2.5.3 Use of the General Register During the Execution p_2_p

The register usage for each case(spba_2_spba, asrc_2_spba…) are different, so don't present here.

A.3.2.5.4 Overview of Script Functionality p_2_p

The script get the number of data to transfer per event according to the bit 31(CONT) of INFO.

If CONT is 0, each transfer loop will transmit the number of data as in the mode count; Or else, it means the number of samples to be transferred is not known and the transfer must take place as long as both events are detected and the script must be manually stopped by the application in case the transfer needs to be stopped.

When PS or PA is set, then after each N words transfer, a pad swallowing or pad adding will be done. This is to handle the ASRC involved transfer.

The transfer is executed only when the event is set. After each transfer for the event, the script will try to obtain a new BD.

A.3.2.6 hdmi_dma

This script sets up the HDMI DMA start and end address from the buffer chain, and launches an HDMI DMA transfer on the HDMI DMA done event.

A.3.2.6.1 Parameters Transmitted Through the Context hdmi_dma

r4: Number of DMA buffers in loop

r6: Address of data structure setup by software. Detailed data structure below:

Table A-2. SDMA Scripts Overview

Name	Offset	Width (bytes)	Description
CONTROL	0	4	Address of HDMI DMA control register
STATUS	4	4	Address of HDMI DMA status register
START	8	4	Address of HDMI DMA start address register
START1	12	4	Start address of buffer 1 for HDMI DMA transfer
END1	16	4	End address of buffer 1 for HDMI DMA transfer
START2	20	4	Start address of buffer 1 for HDMI DMA transfer. Available only when value of r4>1
END2	24	4	End address of buffer 1 for HDMI DMA transfer. Available only when value of r4>1

Table continues on the next page...

Table A-2. SDMA Scripts Overview (continued)

Name	Offset	Width (bytes)	Description
STARTn	4+8*n	4	Start address of buffer 1 for HDMI DMA transfer. n is the value of r4.
ENDn	8+8*n	4	End address of buffer 1 for HDMI DMA transfer. n is ghe value in r4.

A.3.2.6.2 Parameters Transmitted Through the Buffer Descriptor hdmi_dma

None

A.3.2.6.3 Use of the General Register During the Execution hdmi_dma

r0 = N/A

r1 = control register address

r2 = status register address

r3 = DMA start address register address

r4 = offset of DMA buffer chain end

r5 = offset of current DMA buffer

r6 = pointer of data structure that described above

r7 = N/A

A.3.2.6.4 Overview of Script Functionality hdmi_dma

The parameters of the data structure used in the script are retrieved from the channel context. No parameter is retrieved through the buffer descriptor as no real SDMA transfer performed.

When the HDMI DMA done request is captured, the script will clear the DMA done bit in the status register and set up the DMA start and end address to the next buffer (rewind if it reaches the end of the buffer chain). Next it will restart the HDMI DMA transfer, then yield and wait for the next DMA done request.

When an error is captured, it will stop the channel.

Appendix B
i.MX 6Solo/6DualLite Revision History

B.1 Substantive changes from revision 4 to revision 5

Substantive changes from revision 4 to revision 5 are as follows:

B.1.1 Reference Manual Revision History

No substantive changes

B.1.2 Introduction Revision History

No substantive changes

B.1.3 Memory Maps Revision History

No substantive changes

B.1.4 Interrupts and DMA Events Revision History

No substantive changes

B.1.5 External Signals Revision History

No substantive changes

B.1.6 Fusemap Revision History

No substantive changes

B.1.7 External Memory Controllers Revision History

No substantive changes

B.1.8 System Debug Revision History

No substantive changes

B.1.9 System Boot Revision History

Topic Cross-Reference	Change Description
Redundant boot support for expansion device	Redundant boot image offset updated
Using manufacture mode / serial download mode with eMMC	Added new topic

B.1.10 Multimedia Revision History

Topic Cross-Reference	Change Description
Image Converter (IC)	Changed Data Throughput bullets, For single/several tasks to include pixel/sec values.
Image Rotator (IRT)	Changed Data Throughput bullets, For single/more tasks to include pixel/sec values.
Video de-interlacer (VDIC)	Changed bullet from, Rate: up to pixels/sec, to Rate: up to 198M pixels/sec. (added value)

B.1.11 Power Management Revision History

No substantive changes

B.1.12 System Security Revision History

No substantive changes

B.1.13 ARM Cortex A9 Revision History

Topic Cross-Reference	Change Description
Debug Clocks	Clarified the difference among Debug Clocks listed.

B.1.14 AIPSTZ Revision History

Topic Cross-Reference	Change Description
-	Added Off-Platform Peripherals Index.

B.1.15 APBH Revision History

No substantive changes

B.1.16 ASRC Revision History

No substantive changes

B.1.17 AUDMUX Revision History

No substantive changes

B.1.18 BCH Revision History

No substantive changes

B.1.19 CCM Revision History

No substantive changes

B.1.20 CSI2IPU Revision History

No substantive changes

B.1.21 DCIC Revision History

No substantive changes

B.1.22 ECSPI Revision History

No substantive changes

B.1.23 EIM Revision History

No substantive changes

B.1.24 ENET Revision History

No substantive changes

B.1.25 EPDC Revision History

No substantive changes

B.1.26 EPIT Revision History

No substantive changes

B.1.27 ESAI Revision History

No substantive changes

B.1.28 FLEXCAN3 Revision History

Topic Cross-Reference	Change Description
Protocol Timing	Updated descriptions for "Timing Segment 1" and "Timing Segment 2", and added note.

B.1.29 GPC Revision History

No substantive changes

B.1.30 GPIO Revision History

No substantive changes

B.1.31 GPMI Revision History

No substantive changes

B.1.32 GPT Revision History

No substantive changes

B.1.33 GPU2D Revision History

No substantive changes

B.1.34 GPU3D Revision History

No substantive changes

B.1.35 HDMI Revision History

No substantive changes

B.1.36 HDMI PHY Revision History

No substantive changes

B.1.37 I2C Revision History

No substantive changes

B.1.38 IOMUXC Revision History

No substantive changes

B.1.39 IPU Revision History

No substantive changes

B.1.40 KPP Revision History

No substantive changes

B.1.41 LDB Revision History

No substantive changes

B.1.42 MIPI CSI Revision History

No substantive changes

B.1.43 MIPI DSI Revision History

No substantive changes

B.1.44 MIPI HSI Revision History

No substantive changes

B.1.45 MLB150 Revision History

Topic Cross-Reference	Change Description
Low Power Mode Channel Initialization	Added external MLB INIC note.

B.1.46 MMDC Revision History

Topic Cross-Reference	Change Description
Calibration Process	Modified NOTE to list specific features to disable before beginning write leveling. Clarified that ZQ calibration is allowed with these features enabled.
MMDC initialization MMDC Memory Map/Register Definition	Added PRECHARGE ALL note to the MMDC initialization and the MDCFG1[tRPA] and MDSCR[CMD] bitfields.

B.1.47 NIC Revision History

No substantive changes

B.1.48 OCOTP Revision History

Topic Cross-Reference	Change Description
OCOTP	Removed references to non-existent sections "Software Write Sequence" and "Software Read Sequence."
Write Postamble	Changed bullet from "Once BUSY ... use HW_DIGCTL_MICROSECONDS to wait 2 us" to "After BUSY ... wait an additional 2 us."

B.1.49 OCRAM Revision History

No substantive changes

B.1.50 PCIe Ctrl Revision History

No substantive changes

B.1.51 PCIe PHY Revision History

No substantive changes

B.1.52 PMU Revision History

No substantive changes

B.1.53 PWM Revision History

No substantive changes

B.1.54 PXP Revision History

No substantive changes

B.1.55 ROMCP Revision History

No substantive changes

B.1.56 SDMA Revision History

No substantive changes

B.1.57 SJC Revision History

No substantive changes

B.1.58 SNVS Revision History

No substantive changes

B.1.59 SPBA Revision History

No substantive changes

B.1.60 SPDIF Revision History

No substantive changes

B.1.61 SRC Revision History

Topic Cross-Reference	Change Description
SRC Memory Map/Register Definition	Added bit field description for SRC_SRSR[0]
Reset inputs and outputs POR (SRC_POR_B)	Removed references to srtc_rst_b as this function is not used in the i.MX 6 Series.

B.1.62 SSI Revision History

No substantive changes

B.1.63 TEMPMON Revision History

No substantive changes

B.1.64 TZASC Revision History

Topic Cross-Reference	Change Description
Address Mapping in various memory mapping modes	Updated description

B.1.65 UART Revision History

No substantive changes

B.1.66 USB Revision History

No substantive changes

B.1.67 USB PHY Revision History

No substantive changes

B.1.68 USDHC Revision History

No substantive changes

B.1.69 VDOA Revision History

No substantive changes

B.1.70 VPU Revision History

No substantive changes

B.1.71 WDOG Revision History

No substantive changes

B.1.72 XTALOSC Revision History

No substantive changes

B.1.73 SDMA Scripts Revision History

No substantive changes

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. Arm, AMBA, Arm Powered, Cortex, Jazelle, Thumb, and TrustZone are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. CoreLink, CoreSight, and NEON are trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2012-2018 NXP B.V.

Document Number IMX6SDLRM
Revision 5, 05/2020